Úvod do teoretické informatiky

Zdeněk Sawa

Katedra informatiky, FEI,
Vysoká škola báňská – Technická universita Ostrava
17. listopadu 15, Ostrava-Poruba 708 33
Česká republika

6. února 2014

Přednášející

Jméno: Ing. Zdeněk Sawa, Ph.D.

E-mail: zdenek.sawa@vsb.cz

Místnost: A1024

Web: http://www.cs.vsb.cz/sawa/uti

Na těchto stránkách najdete:

- Informace o předmětu
- Učební texty
- Slidy z přednášek
- Zadání příkladů na cvičení
- Aktuální informace
- Odkaz na stránku s animacemi

Požadavky

- **Zápočet** (22 bodů):
  - Zápočtová písemka (22 bodů) — bude se psát na cvičení

  Minimum pro získání zápočtu je 7 bodů.

  Možnost opravy za 14 bodů.

- **Zkouška** (78 bodů)
  - Písemná zkouška skládající se ze tří částí po 26 bodech, přičemž z každé části je nutné získat nejméně 10 bodů.

Teoretická informatika

Různé oblasti teoretické informatiky:

- algoritmy
- výpočetní složitost
- výpočetní modely
- teorie automatů
- formální jazyky
- syntaxe a sémantika programovacích jazyků
- teorie typů
- paralelní a distribuované systémy
- ...

Teoretická informatika

Překrývá se s mnoha dalšími oblastmi matematiky a informatiky:

- logika
- teorie grafů
- teorie čísel
- kryptografie
- výpočetní geometrie
- teorie her
- numerická matematika
- ...

Teoretická informatika

Některé důležité charakteristiky teoretické informatiky:

- formální matematický přístup k problémům
- používání matematických definic a důkazů
- rigorózní matematické důkazy

Pojem důkazu, axiomatický přístup.

Logika — obor zkoumající vyplývání, důkazy, formalismy

Konjunkce

Konjunkce: $A \wedge B$

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Poznámka: označuje se též & nebo **and**

Pravidla pro konjunkci

$$\wedge\mathrm{i}: \quad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

$$\wedge\mathrm{e}_1: \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad\qquad \wedge\mathrm{e}_2: \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

$$\mathrm{Assm}: \quad \frac{}{\Gamma, A, \Delta \vdash A}$$

Formálně zapsaný důkaz

$$A \wedge B, C \wedge D \vdash B \wedge C$$

1. $A \wedge B, C \wedge D \vdash A \wedge B$ (Assm)
2. $A \wedge B, C \wedge D \vdash B$ ($\wedge\mathrm{e}_2$ 1)
3. $A \wedge B, C \wedge D \vdash C \wedge D$ (Assm)
4. $A \wedge B, C \wedge D \vdash C$ ($\wedge\mathrm{e}_1$ 3)
5. $A \wedge B, C \wedge D \vdash B \wedge C$ ($\wedge\mathrm{i}$ 2,4)

Formálně zapsaný důkaz

$$A \wedge B, C \wedge D \vdash B \wedge C$$

1. $\Gamma \vdash A \wedge B$ (Assm)
2. $\Gamma \vdash C \wedge D$ (Assm)
3. $\Gamma \vdash C$ ($\wedge \mathrm{e}_1$ 2)
4. $\Gamma \vdash B$ ($\wedge \mathrm{e}_2$ 1)
5. $\Gamma \vdash B \wedge C$ ($\wedge \mathrm{i}$ 4,3)

Poznámka: $\Gamma := A \wedge B, C \wedge D$

Důkaz znázorněný jako strom

$$A \wedge B, C \wedge D \vdash B \wedge C$$

$$\dfrac{\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \qquad \dfrac{\Gamma \vdash C \wedge D}{\Gamma \vdash C}}{\Gamma \vdash B \wedge C}$$

Poznámka: $\Gamma := A \wedge B, C \wedge D$

Odvozené pravidlo

$$\frac{\Gamma \vdash (A \wedge B) \wedge C}{\Gamma \vdash A \wedge (B \wedge C)}$$

1.	$\Gamma \vdash (A \wedge B) \wedge C$	(premisa)
2.	$\Gamma \vdash A \wedge B$	$(\wedge \mathrm{e}_1\ 1)$
3.	$\Gamma \vdash A$	$(\wedge \mathrm{e}_1\ 2)$
4.	$\Gamma \vdash B$	$(\wedge \mathrm{e}_2\ 2)$
5.	$\Gamma \vdash C$	$(\wedge \mathrm{e}_2\ 1)$
6.	$\Gamma \vdash B \wedge C$	$(\wedge \mathrm{i}\ 4{,}5)$
7.	$\Gamma \vdash A \wedge (B \wedge C)$	$(\wedge \mathrm{i}\ 3{,}6)$

Předpoklady

permutace:

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}$$

zeslabení:

$$\frac{\Gamma, \Delta \vdash B}{\Gamma, A, \Delta \vdash B}$$

kontrakce:

$$\frac{\Gamma, A, A, \Delta \vdash B}{\Gamma, A, \Delta \vdash B}$$

$$\text{Ant:} \quad \frac{\Gamma \vdash A}{\Gamma' \vdash A}(\Gamma \subseteq \Gamma')$$

$$\text{Ch:} \quad \frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}$$

Disjunkce

Disjunkce: $A \vee B$

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Poznámka: označuje se též **or**

XOR

xor (exclusive or): $A \oplus B$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Pravidla pro disjunkci

$$\vee\mathrm{i}_1\text{:}\ \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \vee\mathrm{i}_2\text{:}\ \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

$$\vee\mathrm{e}\text{:}\ \frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}$$

Příklad důkazu

$$(A \vee B) \vee C \vdash A \vee (B \vee C)$$

1.	$\varphi \vdash (A \vee B) \vee C$	(Assm)
2.	$\varphi, A \vee B \vdash A \vee B$	(Assm)
3.	$\varphi, A \vee B, A \vdash A$	(Assm)
4.	$\varphi, A \vee B, A \vdash A \vee (B \vee C)$	($\vee \mathrm{i}_1$ 3)
5.	$\varphi, A \vee B, B \vdash B$	(Assm)
6.	$\varphi, A \vee B, B \vdash B \vee C$	($\vee \mathrm{i}_1$ 5)
7.	$\varphi, A \vee B, B \vdash A \vee (B \vee C)$	($\vee \mathrm{i}_2$ 6)
8.	$\varphi, A \vee B \vdash A \vee (B \vee C)$	($\vee$e 2,4,7)
9.	$\varphi, C \vdash C$	(Assm)
10.	$\varphi, C \vdash B \vee C$	($\vee \mathrm{i}_2$ 9)
11.	$\varphi, C \vdash A \vee (B \vee C)$	($\vee \mathrm{i}_2$ 10)
12.	$\varphi \vdash A \vee (B \vee C)$	($\vee$e 1,8,11)

Poznámka: $\varphi := (A \vee B) \vee C$

Negace

Negace: $\neg A$

A	$\neg A$
0	1
1	0

Poznámka: označuje se též $\sim$ nebo **not**

Důkazy sporem

$$\text{Ctr:}\ \frac{\Gamma, A \vdash B \qquad \Gamma, A \vdash \neg B}{\Gamma \vdash \neg A}$$

$$\text{CtrN:}\ \frac{\Gamma, \neg A \vdash B \qquad \Gamma, \neg A \vdash \neg B}{\Gamma \vdash A}$$

Dvojitá negace

$$\neg\neg\text{i:}\ \frac{\Gamma \vdash A}{\Gamma \vdash \neg\neg A} \qquad \neg\neg\text{e:}\ \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}$$

1.	$\Gamma \vdash A$	(premisa)
2.	$\Gamma, \neg A \vdash A$	(Ant 1)
3.	$\Gamma, \neg A \vdash \neg A$	(Assm)
4.	$\Gamma \vdash \neg\neg A$	(Ctr 2,3)

1.	$\Gamma \vdash \neg\neg A$	(premisa)
2.	$\Gamma, \neg A \vdash \neg A$	(Assm)
3.	$\Gamma, \neg A \vdash \neg\neg A$	(Ant 1)
4.	$\Gamma \vdash A$	(CtrN 2,3)

Dvojitá negace

Odvození CtrN pomocí Ctr a $\neg\neg\mathrm{e}$:

$$\mathrm{CtrN}: \quad \frac{\Gamma, \neg A \vdash B \qquad \Gamma, \neg A \vdash \neg B}{\Gamma \vdash A}$$

1. $\Gamma, \neg A \vdash B$ (premisa)
2. $\Gamma, \neg A \vdash \neg B$ (premisa)
3. $\Gamma \vdash \neg\neg A$ (Ctr 1,2)
4. $\Gamma \vdash A$ ($\neg\neg\mathrm{e}$ 3)

Ze sporných předpokladů plyne cokoliv

$$\text{CtrA:}\ \frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash B}$$

1. $\Gamma \vdash A$ (premisa)
2. $\Gamma \vdash \neg A$ (premisa)
3. $\Gamma, \neg B \vdash A$ (Ant 1)
4. $\Gamma, \neg B \vdash \neg A$ (Ant 2)
5. $\Gamma \vdash B$ (CtrN 3,4)

Kontrapozice

$$\text{Cp (a)}: \quad \frac{\Gamma, A \vdash B}{\Gamma, \neg B \vdash \neg A} \qquad \text{Cp (b)}: \quad \frac{\Gamma, A \vdash \neg B}{\Gamma, B \vdash \neg A}$$

$$\text{Cp (c)}: \quad \frac{\Gamma, \neg A \vdash B}{\Gamma, \neg B \vdash A} \qquad \text{Cp (d)}: \quad \frac{\Gamma, \neg A \vdash \neg B}{\Gamma, B \vdash A}$$

1.	$\Gamma, A \vdash B$	(premisa)	1.	$\Gamma, A \vdash \neg B$	(premisa)
2.	$\Gamma, \neg B, A \vdash B$	(Ant 1)	2.	$\Gamma, B, A \vdash B$	(Assm)
3.	$\Gamma, \neg B, A \vdash \neg B$	(Assm)	3.	$\Gamma, B, A \vdash \neg B$	(Ant 1)
4.	$\Gamma, \neg B \vdash \neg A$	(Ctr 2,3)	4.	$\Gamma, B \vdash \neg A$	(Ctr 2,3)

Řetězení argumentů

$$\text{Ch:}\quad \frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}$$

1. $\Gamma \vdash A$ (premisa)
2. $\Gamma, A \vdash B$ (premisa)
3. $\Gamma, \neg B \vdash A$ (Ant 1)
4. $\Gamma, \neg B \vdash \neg A$ (Cp (a) 2)
5. $\Gamma \vdash B$ (CtrN 3,4)

Rozbor případů (proof by cases)

$$\text{PC:}\quad \frac{\Gamma, A \vdash B \qquad \Gamma, \neg A \vdash B}{\Gamma \vdash B}$$

1. $\Gamma, A \vdash B$ (premisa)
2. $\Gamma, \neg A \vdash B$ (premisa)
3. $\Gamma, \neg B \vdash A$ (Cp (c) 2)
4. $\Gamma, \neg B \vdash \neg A$ (Cp (a) 1)
5. $\Gamma \vdash B$ (CtrN 3,4)

Zákon vyloučení třetího (tertium non datur)

$$\vdash A \vee \neg A$$

1.	$A \vdash A$	(Assm)
2.	$A \vdash A \vee \neg A$	($\vee \mathrm{i}_1$ 1)
3.	$\neg A \vdash \neg A$	(Assm)
4.	$\neg A \vdash A \vee \neg A$	($\vee \mathrm{i}_2$ 3)
5.	$\vdash A \vee \neg A$	(PC 2,4)

Další pravidla pro disjunkci

$$\frac{\Gamma, \neg A \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, \neg B \vdash A}{\Gamma \vdash A \vee B}$$

1. $\Gamma, A \vdash A$ (Assm)
2. $\Gamma, A \vdash A \vee B$ ($\vee \mathrm{i}_1$ 1)
3. $\Gamma, \neg A \vdash B$ (premisa)
4. $\Gamma, \neg A \vdash A \vee B$ ($\vee \mathrm{i}_2$ 3)
5. $\Gamma \vdash A \vee B$ (PC 2,4)

Implikace

Implikace: $A \to B$

A	B	$A \to B$
0	0	1
0	1	1
1	0	0
1	1	1

Poznámka: označuje se též $\Rightarrow$ nebo $\supset$

$$\to\mathrm{i}: \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad\qquad \to\mathrm{e}: \quad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

Poznámka: $\to\mathrm{e}$ je známější pod názvem *modus ponens*

Příklady důkazů s implikací

$$\frac{\Gamma \vdash A \to B}{\Gamma, A \vdash B}$$

$$\text{Ch:}\ \frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}$$

1. $\Gamma \vdash A \to B$ (premisa)
2. $\Gamma, A \vdash A \to B$ (Ant 1)
3. $\Gamma, A \vdash A$ (Assm)
4. $\Gamma, A \vdash B$ ($\to$e 2,3)

1. $\Gamma \vdash A$ (premisa)
2. $\Gamma, A \vdash B$ (premisa)
3. $\Gamma \vdash A \to B$ ($\to$i 2)
4. $\Gamma \vdash B$ ($\to$e 3,1)

Ekvivalence

Ekvivalence: $A \leftrightarrow B$

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Poznámka: označuje se též $\Leftrightarrow$ nebo $\equiv$ nebo **iff**

Pravidla pro ekvivalenci

$$\leftrightarrow\mathrm{i}: \quad \frac{\Gamma, A \vdash B \qquad \Gamma, B \vdash A}{\Gamma \vdash A \leftrightarrow B}$$

$$\leftrightarrow\mathrm{e}_1: \quad \frac{\Gamma \vdash A \leftrightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\leftrightarrow\mathrm{e}_2: \quad \frac{\Gamma \vdash A \leftrightarrow B \qquad \Gamma \vdash B}{\Gamma \vdash A}$$

Logické konstanty

Logické konstanty: $\bot$ (false), $\top$ (true)

Poznámka: označují se též 0, 1 nebo **ff**, **tt**

$$\bot\mathrm{e}: \frac{\Gamma \vdash \bot}{\Gamma \vdash A} \qquad \top\mathrm{i}: \frac{}{\vdash \top}$$

$\bot$ a negace

$$\neg\mathrm{i}: \quad \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}$$

1. $\Gamma, A \vdash \bot$ (premisa)
2. $\Gamma, A \vdash \neg\bot$ ($\bot$e 1)
3. $\Gamma \vdash \neg A$ (Ctr 1,2)

$$\neg\mathrm{e}: \quad \frac{\Gamma \vdash \neg A \qquad \Gamma \vdash A}{\Gamma \vdash \bot}$$

$\bot$ a negace

$$\text{Ctr:}\ \frac{\Gamma, A \vdash B \qquad \Gamma, A \vdash \neg B}{\Gamma \vdash \neg A}$$

1. $\Gamma, A \vdash B$ (premisa)
2. $\Gamma, A \vdash \neg B$ (premisa)
3. $\Gamma, A \vdash \bot$ ($\neg$e 2,1)
4. $\Gamma \vdash \neg A$ ($\neg$i 3)

$$\text{CtrA:}\ \frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash B}$$

1. $\Gamma \vdash A$ (premisa)
2. $\Gamma \vdash \neg A$ (premisa)
3. $\Gamma \vdash \bot$ ($\neg$e 2,1)
4. $\Gamma \vdash B$ ($\bot$e 3)

Syntaxe — logické spojky

- Pokud φ je dobře vytvořená formule, tak i $\neg\varphi$ je dobře vytvořená formule.
- Pokud φ a ψ jsou dobře vytvořené formule, tak i $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ a $(\varphi \leftrightarrow \psi)$ jsou dobře vytvořené formule.
- $\bot$ a $\top$ jsou dobře vytvořené formule.

Abstraktní syntaktický strom (abstract syntax tree).

Konvence pro vypouštění závorek.

Výroková logika — syntaxe

At — množina atomických výroků

Abeceda — množina symbolů:

- atomické výroky: všechny prvky z At
- logické spojky: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$, $\top$
- závorky:), (

Jazyk — které sekvence symbolů jsou formulemi:

- Pokud $p \in At$, tak p je formule.
- $\bot$ a $\top$ jsou formule.
- Pokud φ je formule, tak i $\neg\varphi$ je formule.
- Pokud φ a ψ jsou formule, tak i $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ a $(\varphi \leftrightarrow \psi)$ jsou formule.
- Neexistují žádné další formule než ty, vytvořené podle předchozích bodů.

Výroková logika — syntaxe

Stručnější popis syntaxe pomocí tzv. Backus-Naurovy formy:

$$\varphi \ ::= \ p \mid \bot \mid \top \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\varphi \leftrightarrow \varphi)$$

p reprezentuje libovolný atomický výrok z množiny At

V tzv. abstraktní syntaxi se abstrahuje od detailů jako jsou závorky, priority apod.:

$$\varphi \ ::= \ p \mid \bot \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \varphi \leftrightarrow \varphi$$

Alternativní způsob zápisu abstraktní syntaxe:

$$\varphi \ ::= \ p \mid \bot \mid \top \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \to \varphi_2 \mid \varphi_1 \leftrightarrow \varphi_2$$

Výroková logika — sémantika

Pravdivostní ohodnocení: $\nu : At \to \{0, 1\}$

- $\nu \models \varphi$ — formule φ platí (tj. je pravdivá) při ohodnocení ν
- $\nu \not\models \varphi$ — formule φ neplatí (tj. je nepravdivá) při ohodnocení ν

Definice toho, kdy je formule pravdivá při ohodnocení ν:

- $\nu \models p$, kde $p \in At$, platí právě tehdy, když $\nu(p) = 1$.
- $\nu \models \bot$ neplatí nikdy (tj. vždy platí $\nu \not\models \bot$).
- $\nu \models \top$ platí vždy.
- $\nu \models \neg\varphi$ platí právě tehdy, když $\nu \not\models \varphi$.
- $\nu \models \varphi \wedge \psi$ platí právě tehdy, když $\nu \models \varphi$ a $\nu \models \psi$.
- $\nu \models \varphi \vee \psi$ platí právě tehdy, když $\nu \models \varphi$ nebo $\nu \models \psi$.
- $\nu \models \varphi \to \psi$ platí právě tehdy, když $\nu \not\models \varphi$ nebo $\nu \models \psi$.
- $\nu \models \varphi \leftrightarrow \psi$ platí právě tehdy, když $\nu \models \varphi$ a $\nu \models \psi$, nebo když $\nu \not\models \varphi$ a $\nu \not\models \psi$.

Výroková logika — sémantika

$$h_\neg : \{0,1\} \to \{0,1\}$$

x	$h_\neg(x)$
0	1
1	0

$h_* : \{0,1\} \times \{0,1\} \to \{0,1\}$ pro $* \in \{\wedge, \vee, \to, \leftrightarrow\}$

x	y	$h_\wedge(x,y)$	$h_\vee(x,y)$	$h_\to(x,y)$	$h_\leftrightarrow(x,y)$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Výroková logika — sémantika

$\mathcal{L}_{At}$ — množina všech dobře vytvořených formulí

Každému pravdivostnímu ohodnocení $\nu : At \to \{0, 1\}$ je přiřazena funkce

$$\hat{\nu} : \mathcal{L}_{At} \to \{0, 1\}$$

- $\hat{\nu}(p) = \nu(p)$ pro $p \in At$
- $\hat{\nu}(\bot) = 0$
- $\hat{\nu}(\top) = 1$
- $\hat{\nu}(\neg\varphi) = h_\neg(\hat{\nu}(\varphi))$
- $\hat{\nu}(\varphi * \psi) = h_*(\hat{\nu}(\varphi), \hat{\nu}(\psi))$ pro $* \in \{\wedge, \vee, \to, \leftrightarrow\}$

- $\nu \models \varphi$ — pokud $\hat{\nu}(\varphi) = 1$
- $\nu \not\models \varphi$ — pokud $\hat{\nu}(\varphi) = 0$

Interpretace a modely

Obecně se při popisu sémantiky mluví o **interpretacích**.

- V případě výrokové logiky má interpretace podobu pravdivostního ohodnocení.
- V případě jiných logik mohou interpretace vypadat jinak.
- Pokud ν je interpretace, zápis $\nu \models \varphi$ znamená, že formule φ v interpretaci ν platí.
- Interpretaci, ve které platí formule φ, se říká **model** formule φ.
- Pokud Γ je množina formulí, **modelem** této množiny formulí je taková interpretace, která je modelem každé formule z Γ.

Logické vyplývání

φ – formule, Γ – množina formulí

Formule φ **logicky vyplývá** z formulí Γ, což se zapisuje

$$\Gamma \models \varphi,$$

jestliže $\nu \models \varphi$ platí pro každou takovou interpretaci ν, kde pro všechny formule $\psi \in \Gamma$ platí $\nu \models \psi$.

Poznámka: Množina Γ může být i nekonečná.

Místo $\{\psi_1, \psi_2, \ldots, \psi_n\} \models \varphi$ se obvykle píše $\psi_1, \psi_2, \ldots, \psi_n \models \varphi$.

Místo $\emptyset \models \varphi$ se obvykle píše $\models \varphi$.

Tautologie, kontradikce a splnitelné formule

Formule φ je:

- **tautologie** — pro každou interpretaci ν platí $\nu \models \varphi$
- **kontradikce** — pro každou interpretaci ν platí $\nu \not\models \varphi$
- **splnitelná** — existuje alespoň jedna interpretace ν, pro kterou platí $\nu \models \varphi$

$\models \varphi$ — označuje, že φ je tautologie

Poznámka:

$$A_1, A_2, \ldots, A_n \models B$$

právě tehdy, když

$$\models A_1 \to (A_2 \to (\ \cdots\ \to (A_n \to B) \cdots))$$

Tabulková metoda

$$p \wedge (q \rightarrow \neg r)$$

p	q	r	$\neg r$	$q \rightarrow \neg r$	$p \wedge (q \rightarrow \neg r)$
0	0	0	1	1	0
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	0	0	0

Tabulková metoda

$$((A \to B) \to A) \to A$$

A	B	$A \to B$	$(A \to B) \to A$	$((A \to B) \to A) \to A$
0	0	1	0	1
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1

Dedukční systémy

Dedukční neboli **odvozovací systém** — sada pravidel, která určují, jak vypadají důkazy, a jaké kroky je možno v rámci důkazů provádět .

Příklad: Dedukční systém pro výrokovou logiku založený na sekventech (sekventový kalkul), obsahující následující pravidla:

Assm	Ant
$\wedge\mathrm{i}$	$\wedge\mathrm{e}_1$, $\wedge\mathrm{e}_2$
$\vee\mathrm{i}_1$, $\vee\mathrm{i}_2$	$\vee\mathrm{e}$
$\rightarrow\mathrm{i}$	$\rightarrow\mathrm{e}$
$\leftrightarrow\mathrm{i}$	$\leftrightarrow\mathrm{e}_1$, $\leftrightarrow\mathrm{e}_2$
$\neg\mathrm{i}$	$\neg\mathrm{e}$
$\top\mathrm{i}$	$\bot\mathrm{e}$
	$\neg\neg\mathrm{e}$

Poznámka: Jedná se o jednu z variant tzv. **přirozené dedukce**.

Dedukční systémy

φ – formule, Γ – množina formulí (může být i nekonečná)

Pro daný dedukční systém $\mathcal{D}$ zápis

$$\Gamma \vdash_{\mathcal{D}} \varphi$$

označuje, že v rámci systému $\mathcal{D}$ existuje důkaz formule φ z předpokladů Γ.

Pro daný dedukční systém $\mathcal{D}$ musí být přesně specifikováno, kdy platí $\Gamma \vdash_{\mathcal{D}} \varphi$.

Systém založený na sekventech (sekventový kalkul):

- $\Gamma \vdash_{\mathcal{D}} \varphi$ platí právě tehdy, když existují nějaké formule $\psi_1, \psi_2, \ldots, \psi_n$ z množiny Γ takové, že pomocí pravidel daného systému se dá odvodit sekvent

  $$\psi_1, \psi_2, \ldots, \psi_n \vdash \varphi$$

  (tj. existuje důkaz tohoto sekventu).

Dedukční systémy

Dedukční systém $\mathcal{D}$ je:

- **korektní** — jestliže platí, že pokud $\Gamma \vdash_{\mathcal{D}} \varphi$, pak $\Gamma \models \varphi$

  Tj. v rámci systému není možné odvodit nekorektní závěr, co se odvodí, to také skutečně platí.

- **úplný** — jestliže platí, že pokud $\Gamma \models \varphi$, pak $\Gamma \vdash_{\mathcal{D}} \varphi$

  Tj. vše, co platí, se dá v rámci daného systému odvodit.

Dedukční systémy

Pro dokázání toho, že daný systém je korektní stačí dokázat, že každé jednotlivé pravidlo je korektní.

Příklad: Pro dokázání toho, že pravidlo

$$\wedge\text{i:}\quad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

je korektní, stačí dokázat, že:

- pokud $\Gamma \models A$ a $\Gamma \models B$, pak $\Gamma \models A \wedge B$.

Tabulková metoda v rámci dedukčního systému

Tabulkovou metodu je možné v rámci dedukčního systému „simulovat“:

- To, že φ má při daném ohodnocení pravdivostní hodnotu 1 je reprezentováno formulí φ.
- To, že φ má při daném ohodnocení pravdivostní hodnotu 0 je reprezentováno formulí $\neg\varphi$.

Tabulková metoda v rámci dedukčního systému

Příklad: Mějme atomické výroky p, q, r a formuli φ vytvořenou z těchto atomických výroků pomocí logických spojek.

- Pokud například $\nu_1 \models \varphi$ pro ohodnocení ν_1, kde
$$\nu_1(p) = 1, \qquad \nu_1(q) = 0, \qquad \nu_1(r) = 0,$$
v rámci odvozovacího systému se dá odvodit sekvent
$$p, \neg q, \neg r \vdash \varphi$$

- Pokud například $\nu_2 \not\models \varphi$ pro ohodnocení ν_2, kde
$$\nu_2(p) = 0, \qquad \nu_2(q) = 1, \qquad \nu_2(r) = 0,$$
v rámci odvozovacího systému se dá odvodit sekvent
$$\neg p, q, \neg r \vdash \neg\varphi$$

Tabulková metoda v rámci dedukčního systému

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Například pro konjunkci se dají odvodit následující pravidla:

$$\frac{\Gamma \vdash \neg A \qquad \Gamma \vdash \neg B}{\Gamma \vdash \neg(A \wedge B)} \qquad \frac{\Gamma \vdash \neg A \qquad \Gamma \vdash B}{\Gamma \vdash \neg(A \wedge B)}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash \neg B}{\Gamma \vdash \neg(A \wedge B)} \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

Tabulková metoda v rámci dedukčního systému

Na tabulkovou metodu lze nahlížet jako na příklad důkazu rozborem případů:

p_1	p_2	p_3	φ
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

1. $\neg p_1, \neg p_2, \neg p_3 \vdash \varphi$ (...)
2. $\neg p_1, \neg p_2, p_3 \vdash \varphi$ (...)
3. $\neg p_1, p_2, \neg p_3 \vdash \varphi$ (...)
4. $\neg p_1, p_2, p_3 \vdash \varphi$ (...)
5. $p_1, \neg p_2, \neg p_3 \vdash \varphi$ (...)
6. $p_1, \neg p_2, p_3 \vdash \varphi$ (...)
7. $p_1, p_2, \neg p_3 \vdash \varphi$ (...)
8. $p_1, p_2, p_3 \vdash \varphi$ (...)
9. $\neg p_1, \neg p_2 \vdash \varphi$ (PC 2,1)
10. $\neg p_1, p_2 \vdash \varphi$ (PC 4,3)
11. $p_1, \neg p_2 \vdash \varphi$ (PC 6,5)
12. $p_1, p_2 \vdash \varphi$ (PC 8,7)
13. $\neg p_1 \vdash \varphi$ (PC 10,9)
14. $p_1 \vdash \varphi$ (PC 12,11)
15. $\vdash \varphi$ (PC 14,13)

Ekvivalence formulí

Formule φ a ψ jsou **logicky ekvivalentní**, což se označuje $\varphi \Leftrightarrow \psi$, jestliže pro každou interpretaci ν platí

$$\nu \models \varphi \qquad \text{právě tehdy, když} \qquad \nu \models \psi.$$

Poznámka: $\varphi \Leftrightarrow \psi$ platí právě tehdy, když $\varphi \models \psi$ a $\psi \models \varphi$.

Formule φ a ψ jsou v rámci daného dedukčního systému **dokazatelně ekvivalentní**, což se označuje $\varphi \dashv\vdash \psi$, jestliže platí

$$\varphi \vdash \psi \qquad \text{a} \qquad \psi \vdash \varphi.$$

Poznámka: Pokud je dedukční systém korektní, tak z $\varphi \dashv\vdash \psi$ plyne $\varphi \Leftrightarrow \psi$, pokud je úplný, tak z $\varphi \Leftrightarrow \psi$ plyne $\varphi \dashv\vdash \psi$.

Ekvivalence formulí

$\varphi \Leftrightarrow \psi$ právě tehdy, když $\models \varphi \leftrightarrow \psi$.

$\varphi \dashv\vdash \psi$ právě tehdy, když $\vdash \varphi \leftrightarrow \psi$.

Pro libovolné formule φ, ψ a χ platí:

- $\varphi \Leftrightarrow \varphi$.
- Pokud $\varphi \Leftrightarrow \psi$, tak $\psi \Leftrightarrow \varphi$.
- Pokud $\varphi \Leftrightarrow \psi$ a $\psi \Leftrightarrow \chi$, tak $\varphi \Leftrightarrow \chi$.

Podobně:

- $\varphi \dashv\vdash \varphi$.
- Pokud $\varphi \dashv\vdash \psi$, tak $\psi \dashv\vdash \varphi$.
- Pokud $\varphi \dashv\vdash \psi$ a $\psi \dashv\vdash \chi$, tak $\varphi \dashv\vdash \chi$.

Ekvivalence formulí

Pokud platí $A \dashv\vdash A'$:

- Z $\Gamma \vdash A$ se dá dokázat $\Gamma \vdash A'$ (a naopak).
- Z $\Gamma, A, \Delta \vdash B$ se dá dokázat $\Gamma, A', \Delta \vdash B$ (a naopak).

Pokud platí $A \dashv\vdash A'$, tak platí i:

- $\neg A \dashv\vdash \neg A'$
- $A \wedge B \dashv\vdash A' \wedge B, \quad B \wedge A \dashv\vdash B \wedge A'$
- $A \vee B \dashv\vdash A' \vee B, \quad B \vee A \dashv\vdash B \vee A'$
- $A \to B \dashv\vdash A' \to B, \quad B \to A \dashv\vdash B \to A'$
- $A \leftrightarrow B \dashv\vdash A' \leftrightarrow B, \quad B \leftrightarrow A \dashv\vdash B \leftrightarrow A'$

(podobně to platí i pro $\Leftrightarrow$)

Některé důležité ekvivalence

Poznámka: Všechny níže uvedené ekvivalence jsou i dokazatelné ($\dashv\vdash$).

$$A \Leftrightarrow \neg\neg A\,,$$

Asociativita $\wedge$, $\vee$, $\leftrightarrow$:

$$\begin{gathered}(A \wedge B) \wedge C \;\Leftrightarrow\; A \wedge (B \wedge C)\\ (A \vee B) \vee C \;\Leftrightarrow\; A \vee (B \vee C)\\ (A \leftrightarrow B) \leftrightarrow C \;\Leftrightarrow\; A \leftrightarrow (B \leftrightarrow C)\end{gathered}$$

Komutativita $\wedge$, $\vee$, $\leftrightarrow$:

$$A \wedge B \Leftrightarrow B \wedge A \qquad A \vee B \Leftrightarrow B \vee A \qquad A \leftrightarrow B \Leftrightarrow B \leftrightarrow A$$

Idempotence $\wedge$ a $\vee$:

$$A \wedge A \Leftrightarrow A \qquad A \vee A \Leftrightarrow A$$

Některé důležité ekvivalence

Distributivní zákony pro $\wedge$ a $\vee$:

$$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C) \qquad A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$$

De Morganovy zákony:

$$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B \qquad \neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$$

Konjunkce, resp. disjunkce, formulí A a $\neg A$ je ekvivalentní formuli $\bot$, resp. $\top$:

$$A \wedge \neg A \Leftrightarrow \bot \qquad A \vee \neg A \Leftrightarrow \top$$

Některé důležité ekvivalence

Vztahy s $\bot$ a $\top$:

$$\neg\bot \Leftrightarrow \top \qquad \neg\top \Leftrightarrow \bot$$
$$A \wedge \bot \Leftrightarrow \bot \qquad A \wedge \top \Leftrightarrow A$$
$$A \vee \bot \Leftrightarrow A \qquad A \vee \top \Leftrightarrow \top$$
$$A \to \bot \Leftrightarrow \neg A \qquad A \to \top \Leftrightarrow \top$$
$$\bot \to A \Leftrightarrow \top \qquad \top \to A \Leftrightarrow A$$
$$A \leftrightarrow \bot \Leftrightarrow \neg A \qquad A \leftrightarrow \top \Leftrightarrow A$$

Vztahy pro nahrazení implikace:

$$A \to B \Leftrightarrow \neg A \vee B \qquad \neg(A \to B) \Leftrightarrow A \wedge \neg B$$

Vztahy pro nahrazení ekvivalence:

$$A \leftrightarrow B \Leftrightarrow (A \to B) \wedge (B \to A)$$
$$A \leftrightarrow B \Leftrightarrow (A \wedge B) \vee (\neg A \wedge \neg B)$$
$$A \leftrightarrow B \Leftrightarrow (A \vee \neg B) \wedge (\neg A \vee B)$$

Ekvivalentní úpravy

Místo $A_1 \Leftrightarrow A_2$, $A_2 \Leftrightarrow A_3$, $\ldots$, $A_{n-1} \Leftrightarrow A_n$ je možné psát

$$A_1 \Leftrightarrow A_2 \Leftrightarrow A_3 \Leftrightarrow \cdots \Leftrightarrow A_{n-1} \Leftrightarrow A_n$$

Pro libovolné A_i, A_j, kde $i, j \in \{1, 2, \ldots, n\}$, pak platí $A_i \Leftrightarrow A_j$.

Podobně místo $A_1 \dashv\vdash A_2$, $A_2 \dashv\vdash A_3$, $\ldots$, $A_{n-1} \dashv\vdash A_n$ je možné psát

$$A_1 \dashv\vdash A_2 \dashv\vdash A_3 \dashv\vdash \cdots \dashv\vdash A_{n-1} \dashv\vdash A_n$$

Pro libovolné A_i, A_j, kde $i, j \in \{1, 2, \ldots, n\}$, pak platí $A_i \dashv\vdash A_j$.

Normální formy formulí

Uvažujme nyní pouze formule výrokové logiky.

- **Literál** — atomický výrok nebo jeho negace, např. p nebo $\neg r$

$$L \ ::= \ p \ \mid \ \neg p$$

- **Elementární konjunkce** — konjunkce jednoho nebo více literálů, např. $p \wedge \neg q$, r, $q \wedge \neg r \wedge p$

$$C \ ::= \ L \ \mid \ L \wedge C$$

- **Elementární disjunkce** (**klauzule**) — disjunkce jednoho nebo více literálů, např. $p \vee \neg q$, r, $q \vee \neg r \vee p$

$$D \ ::= \ L \ \mid \ L \vee D$$

Normální formy formulí

- **Disjunktivní normální forma** (**DNF**) — disjunkce jedné nebo více elementárních konjunkcí, např. $(p \wedge \neg q) \vee (\neg r) \vee (\neg r \wedge \neg p \wedge \neg q)$

$$E \ ::= \ C \ \mid \ C \vee E$$

- **Konjunktivní normální forma** (**KNF**) — konjunkce jedné nebo více elementárních disjunkcí (klauzulí),
  např. $(p \vee \neg q) \wedge (\neg r) \wedge (\neg r \vee \neg p \vee \neg q)$

$$F \ ::= \ D \ \mid \ D \wedge F$$

Poznámka: Formule $\bot$ a $\top$ také budeme považovat za formule v DNF a KNF.

Normální formy formulí

- U formulí v KNF se snadno zjistí, jestli jde nebo nejde o tautologii:

  U tautologie v KNF musí každá klauzule obsahovat jako literál nějaký atomický výrok p a zároveň jeho negaci $\neg p$.

- U formulí v DNF se snadno zjistí, jestli jde nebo nejde o kontradikci:

  U kontradikce v DNF musí každá elementární konjunkce obsahovat nějaký atomický výrok p a zároveň jeho negaci $\neg p$.

Normální formy formulí

p	q	r	φ
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

DNF:

$(\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$

KNF:

$(p \vee q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$

Normální formy formulí

Pokud uvažujeme pevně danou **konečnou** množinu atomických výroků At:

- **Úplná disjunktivní normální forma** (**ÚDNF**) — formule v DNF, kde každá elementární konjunkce obsahuje každý atomický výrok z At právě jednou.

  Příklad: $(p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r)$

- **Úplná konjunktivní normální forma** (**ÚKNF**) — formule v KNF, kde každá klauzule obsahuje každý atomický výrok z At právě jednou.

  Příklad: $(p \vee \neg q \vee \neg r) \wedge (p \vee q \vee \neg r) \wedge (\neg p \vee q \vee \neg r)$

Poznámka: V příkladech je $At = \{p, q, r\}$.

Množiny

Množina — soubor prvků

Prvek a je prvkem množiny S:

$$a \in S$$

Prvek a není prvkem množiny S: $a \notin S$

Konečnou množinu je možné vyjádřit výčtem prvků, které obsahuje:

$$S = \{a, b, c\}$$

Podmnožina: $S \subseteq T$ — každý prvek množiny S je prvkem množiny T

Uspořádané n-tice

Uspořádaná n-tice: $(a_1, a_2, \ldots, a_n)$ nebo $\langle a_1, a_2, \ldots, a_n \rangle$

- Uspořádaná dvojice: (a, b) nebo $\langle a, b \rangle$
- Uspořádaná trojice: (a, b, c) nebo $\langle a, b, c \rangle$
- ...

Kartézský součin:

$$S_1 \times S_2 \times \cdots \times S_n$$

— množina všech uspořádaných n-tic

$$(a_1, a_2, \ldots, a_n)$$

kde $a_1 \in S_1$, $a_2 \in S_2$, ..., $a_n \in S_n$

Relace

Relace: $R \subseteq S_1 \times S_2 \times \cdots \times S_n$

- vyjádření vztahů mezi n-ticemi prvků
- n — arita relace
  - $n = 1$ — unární
  - $n = 2$ — binární
  - $n = 3$ — ternární

Příklad: Binární relace $R_1 \subseteq \mathbb{N} \times \mathbb{N}$ tvořená dvojicemi čísel (m, n), kde $m < n$

$$(m, n) \in R_1 \qquad \text{právě tehdy, když} \qquad m < n$$

Poznámka: $\mathbb{N} = \{0, 1, 2, \ldots\}$

Relace

Predikát — dané n-tici prvků $(a_1, a_2, \ldots, a_n)$ přiřadí pravdivostní hodnotu podle toho, zda je nebo není tato n-tice v dané relaci R:

$$R(a_1, a_2, \ldots, a_n)$$

Tj. $R(a_1, a_2, \ldots, a_n)$ platí právě tehdy, když $(a_1, a_2, \ldots, a_n) \in R$.

Příklad: Predikát R_1, kde

$$R_1(m, n)$$

platí právě tehdy, když $(m, n) \in R_1$, tj. když $m < n$

Relace

Unární relace — $R \subseteq S$ vyjařuje nějakou vlastnost prvků z množiny S

Příklad: Unární relace $R_2 \subseteq \mathbb{N}$ tvořená těmi čísly, která jsou prvočísly

Predikát R_2 vyjádřující, že n je prvočíslo

$$R_2(n)$$

Funkce

Funkce — binární relace $f \subseteq S \times T$ splňující:

- pokud $(x, y_1) \in f$ a $(x, y_2) \in f$, pak $y_1 = y_2$

Tj. ke každému prvku $x \in S$ existuje nejvýše jeden prvek $y \in T$ takový, že $(x, y) \in f$.

Tento prvek se označuje

$$f(x)$$

Funkce f — reprezentuje zobrazení, které prvkům z množiny S přiřazuje prvky z množiny T:

$$f : S \to T$$

Totální vs. parciální funkce

Funkce

Funkce $f : S \to T$, kde $S = A_1 \times A_2 \times \cdots \times A_n$:

$$f : A_1 \times A_2 \times \cdots \times A_n \to T$$

n — arita funkce f

Místo $f((a_1, a_2, \ldots, a_n))$ se píše $f(a_1, a_2, \ldots, a_n)$.

Příklad: Binární funkce $f_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, která dvojici přirozených čísel přiřadí jejich součet

$$f_1(x, y) = x + y$$

Struktury

Struktura — neprázdná množina prvků spolu s několika relacemi a funkcemi nad prvky této množiny

Příklad: Množina $\mathbb{N} = \{0, 1, 2, \ldots\}$ spolu s následujícími relacemi a funkcemi:

- unární funkce $f : \mathbb{N} \to \mathbb{N}$, kde $f(x) = x + 1$
- binární funkce $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, kde $g(x, y) = x + y$
- binární funkce $h : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, kde $h(x, y) = x \cdot y$
- binární relace $P \subseteq \mathbb{N} \times \mathbb{N}$, kde $(x, y) \in P$ právě tehdy, když $x = y$
- binární relace $Q \subseteq \mathbb{N} \times \mathbb{N}$, kde $(x, y) \in Q$ právě tehdy, když $x < y$

Struktury

Příklad: Množina $A = \{a, b, c\}$ spolu následující funkcemi f a g a relací R:

- unární funkce $f : A \to A$ a binární funkce $g : A \times A \to A$

x	$f(x)$
a	b
b	a
c	b

g	a	b	c
a	c	a	a
b	a	b	c
c	a	c	c

- binární relace $R \subseteq A \times A$, kde

$$R = \{(a, a), (a, c), (b, b), (c, a), (c, b)\}$$

Signatury

Signatura — čtveřice

$$(\mathcal{P}, \mathcal{F}, \mathcal{C}, \text{arity})$$

- $\mathcal{P}$ — množina **predikátových symbolů**
- $\mathcal{F}$ — množina **funkčních symbolů**
- $\mathcal{C}$ — množina **konstantních symbolů**
- $\text{arity} : \mathcal{P} \cup \mathcal{F} \to \mathbb{N}$ — funkce určující **aritu** jednotlivých predikátových a funkčních symbolů

Příklad: Signatura $S = (\mathcal{P}, \mathcal{F}, \mathcal{C}, \text{arity})$, kde $\mathcal{P} = \{P, Q, R\}$, $\mathcal{F} = \{f, g\}$, $\mathcal{C} = \{c, d\}$ a kde $\text{arity}(P) = 1$, $\text{arity}(Q) = 1$, $\text{arity}(R) = 2$, $\text{arity}(f) = 2$, $\text{arity}(g) = 1$

Interpretace

Signatura $S = (\mathcal{P}, \mathcal{F}, \mathcal{C}, \text{arity})$

Interpretace: $\mathfrak{A} = (A, \mathfrak{a})$

- A — **univerzum** — libovolná neprázdná množina
- $\mathfrak{a}$ — zobrazení přiřazující význam jednotlivým symbolům signatury S:
  - pro $P \in \mathcal{P}$ (kde $\text{arity}(P) = n$):
    $\mathfrak{a}(P) = P^{\mathfrak{A}}$, kde $P^{\mathfrak{A}} \subseteq A^n$ je libovolná n-ární relace nad množinou A
  - pro $f \in \mathcal{F}$ (kde $\text{arity}(f) = n$):
    $\mathfrak{a}(f) = f^{\mathfrak{A}}$, kde $f^{\mathfrak{A}} : A^n \to A$ je libovolná n-ární (totální) funkce nad množinou A
  - pro $c \in \mathcal{C}$:
    $\mathfrak{a}(c) = c^{\mathfrak{A}}$, kde $c^{\mathfrak{A}} \in A$ je libovolný prvek univerza A

Valuace

Proměnné: $Var = \{x_0, x_1, x_2, \ldots\}$

Poznámka: Proměnné budeme označovat x, y, z

Předpokládejme interpretaci $\mathfrak{A} = (A, \mathfrak{a})$.

Valuace — určuje hodnoty proměnných, přiřazuje proměnným prvky univerza:

$$v : Var \to A$$

Interpretace a valuace:

$$\mathfrak{I} = (\mathfrak{A}, v)$$

Termy

Term — výraz označující prvek univerza:

- x — kde $x \in \mathit{Var}$
- c — kde $c \in \mathcal{C}$
- $f(t_1, t_2, \ldots, t_n)$ — kde $f \in \mathcal{F}$, $\mathrm{arity}(f) = n$ a kde $t_1, t_2, \ldots, t_n$ jsou termy

Příklady termů: x c $f(x, y)$ $f(g(x), f(c, y))$

Syntaxe:

$$t \quad ::= \quad x \mid c \mid f(t, t, \ldots, t)$$

Termy

Interpretace a valuace $\mathfrak{I} = (\mathfrak{A}, v)$

$\mathfrak{I}(t)$ — hodnota, jaké nabývá term t v interpretaci $\mathfrak{A}$ při ohodnocení v

- pro $x \in Var$: $\mathfrak{I}(x) = v(x)$
- Pro $c \in \mathcal{C}$: $\mathfrak{I}(c) = c^{\mathfrak{A}}$
- Pro $f \in \mathcal{F}$ (kde $\text{arity}(f) = n$) a termy $t_1, t_2, \ldots, t_n$:
  $\mathfrak{I}(f(t_1, t_2, \ldots, t_n)) = f^{\mathfrak{A}}(\mathfrak{I}(t_1), \mathfrak{I}(t_2), \ldots, \mathfrak{I}(t_n))$

Atomické formule

Atomická formule:

$$P(t_1, t_2, \ldots, t_n)$$

kde $P \in \mathcal{P}$ (kde $\text{arity}(P) = n$) a $t_1, t_2, \ldots, t_n$ jsou termy

Příklady atomických formulí: $P(x)$ $Q(g(g(c)))$ $R(f(x, y), x)$

Interpretace a valuace $\mathfrak{I} = (\mathfrak{A}, v)$

Formule φ platí při interpretaci $\mathfrak{A}$ a valuaci v:

$$\mathfrak{I} \models \varphi$$

- $\mathfrak{I} \models P(t_1, t_2, \ldots, t_n)$ právě tehdy, když $(\mathfrak{I}(t_1), \mathfrak{I}(t_2), \ldots, \mathfrak{I}(t_n)) \in P^{\mathfrak{A}}$.

Rovnost

Rovnost (**identita**): označuje se symbolem $=$

Atomická formule:

$$t_1 = t_2$$

Příklady atomických formulí s rovností:

$$x = y \qquad f(f(x, y), z) = g(x)$$

- $\mathfrak{I} \models t_1 = t_2$ právě tehdy, když $\mathfrak{I}(t_1) = \mathfrak{I}(t_2)$

Kvantifikátory

Formule je možné vytvářet z menších podformulí pomocí logických spojek: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$, $\top$

Kromě toho je možné ve formulích používat **kvantifikátory**:

- univerzální kvantifikátor: $\forall$
- existenční kvantifikátor: $\exists$

Pokud φ je formule a x je proměnná ($x \in \mathit{Var}$), tak i:

- $\forall x \varphi$ je formule

  — reprezentuje tvrzení „pro každý prvek x platí φ“, „pro všechna x platí φ“, apod.

- $\exists x \varphi$ je formule

  — reprezentuje tvrzení „existuje prvek x, pro který platí φ“, „pro nějaké x platí φ“, apod.

Kvantifikátory

Interpretace $\mathfrak{A} = (A, \mathfrak{a})$, valuace $v : Var \to A$

$x \in Var$, $a \in A$

Zápis

$$v[x \mapsto a]$$

označuje valuaci $v' : Var \to A$, která se s valuací v shoduje v hodnotách všech proměnných jiných než x, a kde x nabývá hodnoty a

Tj. pro $y \in Var$ je

$$v'(y) = \begin{cases} a & \text{pokud } y = x \\ v(y) & \text{jinak} \end{cases}$$

$\mathfrak{I} = (\mathfrak{A}, v)$

$$\mathfrak{I}[x \mapsto a] = (\mathfrak{A}, v[x \mapsto a])$$

Kvantifikátory

- $\mathfrak{I} \models \forall x \varphi$ platí právě tehdy, když pro každé $a \in A$ platí $\mathfrak{I}[x \mapsto a] \models \varphi$
- $\mathfrak{I} \models \exists x \varphi$ platí právě tehdy, když existuje nějaké $a \in A$ takové, že $\mathfrak{I}[x \mapsto a] \models \varphi$

Příklady formulí:

- $\forall x \exists y R(x, y)$ — ke každému x existuje y takové, že x a y jsou v relaci R
- $\neg \exists x (P(x) \wedge Q(x))$ — neexistuje x, pro které by současně platilo $P(x)$ a $Q(x)$
- $\exists x P(x) \rightarrow \forall y Q(y)$ — pokud existuje x, pro které platí $P(x)$, pak pro každé y platí $Q(y)$

Predikátová logika 1. řádu — shrnutí

Symboly:

- Logické symboly:
  - **proměnné**: $x \in \mathit{Var}$, kde $\mathit{Var} = \{x_0, x_1, x_2, \ldots\}$
  - **logické spojky**: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
  - **kvantifikátory**: $\forall$, $\exists$
  - **závorky**:), (
  - **symbol pro rovnost**: $=$
- Mimologické symboly — dány signaturou $S = (\mathcal{P}, \mathcal{F}, \mathcal{C}, \text{arity})$:
  - **predikátové symboly**: $P \in \mathcal{P}$
  - **funkční symboly**: $f \in \mathcal{F}$
  - **konstantní symboly**: $c \in \mathcal{C}$

Syntaxe:

$$
\begin{aligned}
t &::= x \mid c \mid f(t, t, \ldots, t) \\
\varphi &::= P(t, t, \ldots, t) \mid t = t \mid \bot \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \\
&\quad \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi \mid \forall x \varphi \mid \exists x \varphi
\end{aligned}
$$

Predikátová logika 1. řádu — shrnutí

Sémantika:

Hodnota termu při $\mathfrak{I} = (\mathfrak{A}, v)$:

- pro $x \in Var$: $\mathfrak{I}(x) = v(x)$
- pro $c \in \mathcal{C}$: $\mathfrak{I}(c) = c^{\mathfrak{A}}$
- pro $f \in \mathcal{F}$ (kde $\text{arity}(f) = n$) a termy $t_1, t_2, \ldots, t_n$:
  $\mathfrak{I}(f(t_1, t_2, \ldots, t_n)) = f^{\mathfrak{A}}(\mathfrak{I}(t_1), \mathfrak{I}(t_2), \ldots, \mathfrak{I}(t_n))$

Predikátová logika 1. řádu — shrnutí

Pravdivost formule při $\mathfrak{I} = (\mathfrak{A}, \mathfrak{a})$:

- Pro $P \in \mathcal{P}$, kde $\text{arity}(P) = n$, a pro termy $t_1, t_2, \ldots, t_n$ platí $\mathfrak{I} \models P(t_1, t_2, \ldots, t_n)$ právě tehdy, když $(\mathfrak{I}(t_1), \mathfrak{I}(t_2), \ldots, \mathfrak{I}(t_n)) \in P^{\mathfrak{A}}$
- Pro termy t_1, t_2 platí $\mathfrak{I} \models t_1 = t_2$ právě tehdy, když $\mathfrak{I}(t_1) = \mathfrak{I}(t_2)$
- $\mathfrak{I} \models \bot$ neplatí nikdy, tj. vždy platí $\mathfrak{I} \not\models \bot$
- $\mathfrak{I} \models \top$ platí vždy
- $\mathfrak{I} \models \neg\varphi$ platí právě tehdy, když $\mathfrak{I} \not\models \varphi$
- $\mathfrak{I} \models \varphi \wedge \psi$ platí právě tehdy, když $\mathfrak{I} \models \varphi$ a $\mathfrak{I} \models \psi$
- $\mathfrak{I} \models \varphi \vee \psi$ platí právě tehdy, když $\mathfrak{I} \models \varphi$ nebo $\mathfrak{I} \models \psi$
- $\mathfrak{I} \models \varphi \to \psi$ platí právě tehdy, když $\mathfrak{I} \not\models \varphi$ nebo $\mathfrak{I} \models \psi$
- $\mathfrak{I} \models \varphi \leftrightarrow \psi$ platí právě tehdy, když $\mathfrak{I} \models \varphi$ a $\mathfrak{I} \models \psi$, nebo když $\mathfrak{I} \not\models \varphi$ a $\mathfrak{I} \not\models \psi$
- $\mathfrak{I} \models \forall x\varphi$ platí právě tehdy, když pro každé $a \in A$ platí $\mathfrak{I}[x \mapsto a] \models \varphi$
- $\mathfrak{I} \models \exists x\varphi$ platí právě tehdy, když existuje nějaké $a \in A$ takové, že $\mathfrak{I}[x \mapsto a] \models \varphi$

Volné a vázané výskyty proměnných

Každý výskyt proměnné x ve v podformuli tvaru $\exists x\varphi$ nebo $\forall x\varphi$ je **vázaný**.

Výskyt proměnné, který není vázaný, je **volný**.

$\mathrm{free}(t)$ — množina proměnných, které se vyskytují jako volné proměnné v termu t:

- $\mathrm{free}(x) = \{x\}$ pro $x \in \mathit{Var}$
- $\mathrm{free}(c) = \emptyset$ pro $c \in \mathcal{C}$
- $\mathrm{free}(f(t_1, t_2, \ldots, t_n)) = \mathrm{free}(t_1) \cup \mathrm{free}(t_2) \cup \ldots \cup \mathrm{free}(t_n)$ pro $f \in \mathcal{F}$

$\mathrm{free}(\varphi)$ — množina proměnných, které se vyskytují jako volné proměnné ve formuli φ:

- $\mathrm{free}(P(t_1, t_2, \ldots, t_n)) = \mathrm{free}(t_1) \cup \mathrm{free}(t_2) \cup \cdots \cup \mathrm{free}(t_n)$ pro $P \in \mathcal{P}$
- $\mathrm{free}(t_1 = t_2) = \mathrm{free}(t_1) \cup \mathrm{free}(t_2)$
- $\mathrm{free}(\bot) = \mathrm{free}(\top) = \emptyset$
- $\mathrm{free}(\neg\varphi) = \mathrm{free}(\varphi)$
- $\mathrm{free}(\varphi * \psi) = \mathrm{free}(\varphi) \cup \mathrm{free}(\psi)$ pro $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$
- $\mathrm{free}(\mathrm{Q}x\varphi) = \mathrm{free}(\varphi) - \{x\}$ pro $\mathrm{Q} \in \{\exists, \forall\}$ a $x \in \mathit{Var}$

Volné a vázané výskyty proměnných

Pravdivost formule φ při $\mathfrak{I} = (\mathfrak{A}, v)$ závisí pouze na $\mathfrak{A}$ a hodnotách, které v přiřazuje proměnným z množiny $\mathrm{free}(\varphi)$.

Speciálně v případě, kdy $\mathrm{free}(\varphi) = \emptyset$, tak pravdivost φ při $\mathfrak{I} = (\mathfrak{A}, v)$ závisí pouze na interpretaci $\mathfrak{A}$.

Pro formule φ, kde $\mathrm{free}(\varphi) = \emptyset$, můžeme psát

$$\mathfrak{A} \models \varphi \qquad \text{nebo} \qquad \mathfrak{A} \not\models \varphi$$

Formule φ, kde $\mathrm{free}(\varphi) = \emptyset$, se nazývá **uzavřená formule** neboli **sentence**.

Substituce

φ – formule, x – proměnná, t – term

Formule, kterou dostaneme, když dosadíme term t za **volné** výskyty proměnné x ve formuli φ:

$$\varphi[t/x]$$

Příklad: $\varphi := \forall x(P(x) \to R(f(x,z),y)), \quad t := g(f(y,w))$

Formule $\varphi[t/z]$:

$$\forall x(P(x) \to R(f(x,g(f(y,w))),y))$$

Poznámka: Je třeba dát pozor na to, aby se volný výskyt proměnné v termu t nestal po dosazení vázaným.
V takovém případě je potřeba vázanou proměnnou přejmenovat.

Dedukční systém pro predikátovou logiku 1. řádu

Dedukční systém obsahující stejná pravidla jako systém pro výrokovou logiku.

Navíc několik pravidel pro práci s kvantifikátory a s rovností.

$$\forall\mathrm{e}: \quad \frac{\Gamma \vdash \forall x A}{\Gamma \vdash A[t/x]}$$

Příklad:

$$\forall x(P(x) \to Q(x)),\ P(c) \vdash Q(c)$$

1.	$\Gamma \vdash \forall x(P(x) \to Q(x))$	(Assm)
2.	$\Gamma \vdash P(c) \to Q(c)$	($\forall$e 1)
3.	$\Gamma \vdash P(c)$	(Assm)
4.	$\Gamma \vdash Q(c)$	($\to$e 2,3)

$\Gamma := \forall x(P(x) \to Q(x)),\ P(c)$

Univerzální kvantifikátor

$$\forall \text{i:} \quad \frac{\Gamma \vdash A[y/x]}{\Gamma \vdash \forall x A} (y \notin \text{free}(\Gamma, \forall x A))$$

Speciální případ:

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x A} (x \notin \text{free}(\Gamma))$$

Univerzální kvantifikátor

Příklad: $\forall x(P(x) \vee Q(x)), \forall x(Q(x) \to R(x)) \vdash \forall x(P(x) \vee R(x))$

1.	$\Gamma \vdash \forall x(P(x) \vee Q(x))$	(Assm)
2.	$\Gamma \vdash P(x) \vee Q(x)$	($\forall$e 1)
3.	$\Gamma, P(x) \vdash P(x)$	(Assm)
4.	$\Gamma, P(x) \vdash P(x) \vee R(x)$	($\vee$i$_1$ 3)
5.	$\Gamma, Q(x) \vdash \forall x(Q(x) \to R(x))$	(Assm)
6.	$\Gamma, Q(x) \vdash Q(x) \to R(x)$	($\forall$e 5)
7.	$\Gamma, Q(x) \vdash Q(x)$	(Assm)
8.	$\Gamma, Q(x) \vdash R(x)$	($\to$e 6,7)
9.	$\Gamma, Q(x) \vdash P(x) \vee R(x)$	($\vee$i$_2$ 8)
10.	$\Gamma \vdash P(x) \vee R(x)$	($\vee$e 2,4,9)
11.	$\Gamma \vdash \forall x(P(x) \vee R(x))$	($\forall$i 10)

$\Gamma := \forall x(P(x) \vee Q(x)), \forall x(Q(x) \to R(x))$

Existenční kvantifikátor

$$\exists\mathrm{i}: \quad \frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x A}$$

$$\exists\mathrm{e}: \quad \frac{\Gamma \vdash \exists x A \qquad \Gamma, A[y/x] \vdash B}{\Gamma \vdash B}(y \notin \mathsf{free}(\Gamma, \exists x A, B))$$

Speciální případ:

$$\frac{\Gamma \vdash \exists x A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}(x \notin \mathsf{free}(\Gamma, B))$$

Rovnost

$$=\text{i:}\ \frac{}{\Gamma \vdash t = t}$$

$$=\text{e:}\ \frac{\Gamma \vdash t = t' \qquad \Gamma \vdash A[t/x]}{\Gamma \vdash A[t'/x]}$$

Příklad:

$$\frac{\Gamma \vdash t_1 = t_2}{\Gamma \vdash t_2 = t_1}$$

1.	$\Gamma \vdash t_1 = t_2$	(premisa)
2.	$\vdash t_1 = t_1$	$(=\text{i})$
3.	$\Gamma \vdash t_1 = t_1$	(Ant 2)
4.	$\Gamma \vdash (x = t_1)[t_1/x]$	(rep. 3, $x \notin \text{free}(t_1)$)
5.	$\Gamma \vdash (x = t_1)[t_2/x]$	$(=\text{e}\ 1{,}4)$
6.	$\Gamma \vdash t_2 = t_1$	(rep. 5)

Některé důležité ekvivalence

Poznámka: Všechny následující ekvivalence jsou i dokazatelné ($\dashv\vdash$)

$$\begin{aligned} \neg\forall x\varphi &\Leftrightarrow \exists x\neg\varphi \\ \neg\exists x\varphi &\Leftrightarrow \forall x\neg\varphi \end{aligned}$$

Pokud $x \notin \mathsf{free}(\psi)$:

$$\begin{aligned} (\forall x\varphi) \wedge \psi &\Leftrightarrow \forall x(\varphi \wedge \psi) \\ (\forall x\varphi) \vee \psi &\Leftrightarrow \forall x(\varphi \vee \psi) \\ (\exists x\varphi) \wedge \psi &\Leftrightarrow \exists x(\varphi \wedge \psi) \\ (\exists x\varphi) \vee \psi &\Leftrightarrow \exists x(\varphi \vee \psi) \end{aligned}$$

Některé důležité ekvivalence

$$(\forall x \varphi) \wedge (\forall x \psi) \Leftrightarrow \forall x (\varphi \wedge \psi)$$
$$(\exists x \varphi) \vee (\exists x \psi) \Leftrightarrow \exists x (\varphi \vee \psi)$$

$$\forall x \forall y \varphi \Leftrightarrow \forall y \forall x \varphi$$
$$\exists x \exists y \varphi \Leftrightarrow \exists y \exists x \varphi$$

Pokud $y \notin \mathsf{free}(\varphi)$:

$$\forall x \varphi \Leftrightarrow \forall y (\varphi[y/x])$$
$$\exists x \varphi \Leftrightarrow \exists y (\varphi[y/x])$$

Pokud $x \notin \mathsf{free}(\varphi)$:

$$\forall x \varphi \Leftrightarrow \varphi$$
$$\exists x \varphi \Leftrightarrow \varphi$$

Existuje právě jeden

Předpokládejme, že:

- φ je formule
- proměnná y je různá od proměnné x
- $y \in \mathrm{free}(\varphi)$

Tvrzení „pro právě jeden prvek x platí φ“ se dá zapsat formulí

$$\exists x(\varphi \wedge \forall y(\varphi[y/x] \rightarrow x = y))$$

Označuje se také:

$$\exists_{=1} x \varphi$$

Poznámka: Ve stejném významu jako $\exists_{=1}$ se používají například symboly $\exists_1$, $\exists^{=1}$, $\exists!$.

Přirozená čísla

Signatura $(\{<\}, \{\sigma, +, \cdot\}, \{0\}, \text{arity})$, kde $\text{arity}(<) = 2$, $\text{arity}(\sigma) = 1$, $\text{arity}(+) = 2$, $\text{arity}(\cdot) = 2$

Příklady formulí:

- $\forall x \exists y (x < y)$ — ke každému přirozenému číslu existuje větší přirozené číslo
- $\sigma(0) + \sigma(0) = \sigma(\sigma(0))$ — platí, že $1 + 1 = 2$
- $\sigma(0) < x \wedge \neg \exists y \exists z (\sigma(0) < y \wedge \sigma(0) < z \wedge y \cdot z = x)$ — formule s volnou proměnnou x reprezentující tvrzení, že x je prvočíslo

Přirozená čísla — Robinsonova a Peanova aritmetika

Axiomy:

- $\forall x(\sigma(x) \neq 0)$
- $\forall x \forall y(\sigma(x) = \sigma(y) \rightarrow x = y)$
- $\forall x(x = 0 \vee \exists y(\sigma(y) = x))$
- $\forall x(x + 0 = x)$
- $\forall x \forall y(x + \sigma(y) = \sigma(x + y))$
- $\forall x(x \cdot 0 = 0)$
- $\forall x \forall y(x \cdot \sigma(y) = (x \cdot y) + x)$
- $\forall x \forall y(x < y \leftrightarrow \exists z(\sigma(z) + x = y))$

Schéma axiomů indukce:

- $\forall y_1 \cdots \forall y_n(\varphi[0/x] \wedge \forall x(\varphi \rightarrow \varphi[\sigma(x)/x]) \rightarrow \forall x \varphi)$

— φ je libovolná formule, kde $\mathrm{free}(\varphi) \subseteq \{x, y_1, \ldots, y_n\}$

Definice

Předpokládejme, že je dána množina axiomů Γ.

- Definice predikátového symbolu P, kde $\text{arity}(P) = n$:

$$\forall x_1 \cdots \forall x_n \, (P(x_1, \ldots, x_n) \;\leftrightarrow\; \varphi)$$

  kde $\text{free}(\varphi) \subseteq \{x_1, \ldots, x_n\}$

- Definice funkčního symbolu f, kde $\text{arity}(f) = n$:

$$\forall x_1 \cdots \forall x_n \forall y \, (f(x_1, \ldots, x_n) = y \;\leftrightarrow\; \varphi)$$

  kde $\text{free}(\varphi) \subseteq \{x_1, \ldots, x_n, y\}$, přičemž $\Gamma \vdash \forall x_1 \cdots \forall x_n \exists_{=1} y(\varphi)$

- Definice konstantního symbolu c:

$$\varphi[c/x]$$

  kde $\text{free}(\varphi) \subseteq \{x\}$, přičemž $\Gamma \vdash \exists_{=1} x(\varphi)$

Teorie grup

Signatura $(\emptyset, \{\circ\}, \{e\}, \text{arity})$, kde $\text{arity}(\circ) = 2$

Axiomy:

- $\forall x \forall y \forall z((x \circ y) \circ z = x \circ (y \circ z))$
- $\forall x(x \circ e = x)$
- $\forall x \exists y(x \circ y = e)$

Teorie množin

Signatura $(\{\in\}, \emptyset, \emptyset, \text{arity})$, kde $\text{arity}(\in) = 2$

Příklady některých axiomů teorie množin:

- **Axiom extenzionality**:

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y)$$

- **Schéma axiomů vydělení**:

$$\forall y \exists z \forall x (x \in z \leftrightarrow (x \in y \wedge \varphi))$$

kde φ je libovolná formule, kde $z \notin \text{free}(\varphi)$

Množina tvořená těmi prvky x množiny y, pro které platí φ:

$$\{x \in y \mid \varphi\}$$

Často používané zkratky

- Tvrzení „existuje x z množiny A takové, že pro x platí φ“:

$$\exists x(x \in A \ \wedge \ \varphi)$$

Zkrácený zápis:

$$(\exists x \in A)(\varphi)$$

- Tvrzení „pro každé x z množiny A platí φ“:

$$\forall x(x \in A \ \rightarrow \ \varphi)$$

Zkrácený zápis:

$$(\forall x \in A)(\varphi)$$

Formální jazyky

Abeceda a slovo

Definice

Abeceda je libovolná neprázdná konečná množina **symbolů** (**znaků**).

Poznámka: Abeceda se často označuje řeckým písmenem Σ (velké sigma).

Definice

Slovo v dané abecedě je libovolná konečná posloupnost symbolů z této abecedy.

Příklad 1:

$\Sigma = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}, \mathtt{I}, \mathtt{J}, \mathtt{K}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{O}, \mathtt{P}, \mathtt{Q}, \mathtt{R}, \mathtt{S}, \mathtt{T}, \mathtt{U}, \mathtt{V}, \mathtt{W}, \mathtt{X}, \mathtt{Y}, \mathtt{Z}\}$

Slova v abecedě Σ: $\mathtt{AHOJ}$ $\mathtt{ABRACADABRA}$ $\mathtt{ERROR}$

Abeceda a slovo

Příklad 2:

$\Sigma_2 = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}, \mathtt{I}, \mathtt{J}, \mathtt{K}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{O}, \mathtt{P}, \mathtt{Q}, \mathtt{R}, \mathtt{S}, \mathtt{T}, \mathtt{U}, \mathtt{V}, \mathtt{W}, \mathtt{X}, \mathtt{Y}, \mathtt{Z}, \sqcup\}$

Slovo v abecedě Σ_2: $\mathtt{HELLO}\sqcup\mathtt{WORLD}$

Příklad 3:

$\Sigma_3 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Slova v abecedě Σ_3: 0, 31415926536, 65536

Příklad 4:

Slova v abecedě $\Sigma_4 = \{0, 1\}$: 011010001, 111, 1010101010101010

Příklad 5:

Slova v abecedě $\Sigma_5 = \{a, b\}$: *aababb*, *abbabbba*, *aaab*

Abeceda a slovo

Příklad 6:

Abeceda Σ_6 je množina všech ASCII znaků.

Příklad slova:

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

```
class␣HelloWorld␣{ ↩ ␣␣␣␣public␣static␣void␣main(Str···
```

Teorie formálních jazyků – motivace

Jazyk — množina (některých) slov tvořených symboly z dané abecedy

Příklady typů problémů, při jejichž řešení se využívá poznatků z teorie formálních jazyků:

- Tvorba překladačů:
  - lexikální analýza
  - syntaktická analýza
- Vyhledávání v textu:
  - hledání zadaného vzorku
  - hledání textu zadaného regulárním výrazem

Práce s formálními jazyky

Když chceme nějaký jazyk popsat, máme několik možností:

- Můžeme vyjmenovat všechna jeho slova (což je ale použitelné jen pro malé konečné jazyky).

  Příklad: $L = \{aab, babba, aaaaaa\}$

- Můžeme specifikovat nějakou vlastnost, kterou mají právě ta slova, která do tohoto jazyka patří:

  Příklad: Jazyk nad abecedou $\{0, 1\}$, obsahující všechna slova, ve kterých je počet výskytů symbolu 1 sudý.

Práce s formálními jazyky

V teorii formálních jazyků se používají především následující dva přístupy:

- Popsat (idealizovaný) stroj, zařízení, algoritmus, který rozpozná slova patřící do daného jazyka – vede k použití tzv. **automatů**.
- Popsat nějaký mechanismus umožňující generovat všechna možná slova patřící do daného jazyka – vede k tzv. **gramatikám** a **regulárním výrazům**.

Některé základní pojmy

Délka slova je počet znaků ve slově.

Například délka slova $abaab$ je 5.

Délku slova w označujeme $|w|$.

Pokud tedy např. $w = abaab$, pak $|w| = 5$.

Počet výskytů znaku a ve slově w označujeme $|w|_a$.

Pro slovo $w = ababb$ tedy platí $|w|_a = 2$ a $|w|_b = 3$.

Prázdné slovo je slovo délky 0, tj. neobsahující žádné znaky.

Prázdné slovo se označuje řeckým písmenem ε (epsilon).

(Pozn.: V literatuře se pro označení prázdného slova někdy používá místo symbolu ε řecké písmeno λ (lambda).)

$$|\varepsilon| = 0$$

Zřetězení slov

Se slovy je možné provádět operaci **zřetězení**:

Například zřetězením slov OST a RAVA vznikne slovo OSTRAVA.

Operace zřetězení se označuje symbolem $\cdot$ (podobně jako násobení). Tento symbol je možné vypouštět.

$$\mathtt{OST} \cdot \mathtt{RAVA} = \mathtt{OSTRAVA}$$

Zřetězení je **asociativní**, tj. pro libovolná tři slova u, v a w platí

$$(u \cdot v) \cdot w = u \cdot (v \cdot w)$$

což znamená, že při zápisu více zřetězení můžeme vypouštět závorky a psát například $w_1 \cdot w_2 \cdot w_3 \cdot w_4 \cdot w_5$ místo $(w_1 \cdot (w_2 \cdot w_3)) \cdot (w_4 \cdot w_5)$.

Zřetězení slov

Zřetězení není **komutativní**, tj. obecně pro dvojici slov u a v neplatí rovnost

$$u \cdot v = v \cdot u$$

Příklad:

$$\texttt{OST} \cdot \texttt{RAVA} \neq \texttt{RAVA} \cdot \texttt{OST}$$

Zjevně pro libovolná slova v a w platí:

$$|v \cdot w| = |v| + |w|$$

Pro libovolné slovo w také platí:

$$\varepsilon \cdot w = w \cdot \varepsilon = w$$

Prefixy, sufixy a podslova

Definice

Slovo x je **prefixem** slova y, jestliže existuje slovo v takové, že $y = xv$.

Slovo x je **sufixem** slova y, jestliže existuje slovo u takové, že $y = ux$.

Slovo x je **podslovem** slova y, jestliže existují slova u a v taková, že $y = uxv$.

Příklad:

- Prefixy slova abaab jsou ε, a, ab, aba, abaa, abaab.
- Sufixy slova abaab jsou ε, b, ab, aab, baab, abaab.
- Podslova slova abaab jsou ε, a, b, ab, ba, aba, baa, aab, abaa, baab, abaab.

Jazyk

Množinu všech slov tvořených symboly z abecedy Σ označujeme Σ^*.

Definice

(Formální) jazyk L v abecedě Σ je nějaká libovolná podmnožina množiny Σ^*, tj. $L \subseteq \Sigma^*$.

Příklad 1: Množina $\{00, 01001, 1101\}$ je jazyk v abecedě $\{0, 1\}$.

Příklad 2: Množina všech syntakticky správných programů v jazyce C je jazyk v abecedě tvořené množinou všech ASCII znaků.

Příklad 3: Množina všech textů obsahujících sekvenci znaků `ahoj` je jazyk v abecedě tvořené množinou všech ASCII znaků.

Množinové operace na jazycích

Vzhledem k tomu, že jazyky jsou množiny, můžeme s nimi provádět množinové operace:

Sjednocení – $L_1 \cup L_2$ je jazyk tvořený slovy, která patří buď do jazyka L_1 nebo do jazyka L_2 (nebo do obou).

Průnik – $L_1 \cap L_2$ je jazyk tvořený slovy, která patří současně do jazyka L_1 i do jazyka L_2.

Doplněk – $\overline{L_1}$ je jazyk tvořený těmi slovy ze Σ^*, která nepatří do L_1.

Rozdíl – $L_1 - L_2$ je jazyk tvořený slovy, která patří do L_1, ale nepatří do L_2.

Poznámka: Při operacích nad jazyky předpokládáme, že jazyky, se kterými operaci provádíme, používají tutéž abecedu Σ.

Množinové operace na jazycích

Formálně:

Sjednocení: $L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \vee w \in L_2\}$

Průnik: $L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \wedge w \in L_2\}$

Doplněk: $\overline{L_1} = \{w \in \Sigma^* \mid w \notin L_1\}$

Rozdíl: $L_1 - L_2 = \{w \in \Sigma^* \mid w \in L_1 \wedge w \notin L_2\}$

Poznámka: Předpokládáme, že $L_1, L_2 \subseteq \Sigma^*$ pro nějakou danou abecedu Σ.

Množinové operace na jazycích

Příklad:

Uvažujme jazyky nad abecedou $\{\mathtt{a}, \mathtt{b}\}$.

- L_1 — množina všech slov obsahujících podslovo $\mathtt{baa}$
- L_2 — množina všech slov se sudým počtem výskytů symbolu $\mathtt{b}$

Pak

- $L_1 \cup L_2$ — množina všech slov obsahujících podslovo $\mathtt{baa}$ nebo sudý počet symbolů $\mathtt{b}$
- $L_1 \cap L_2$ — množina všech slov obsahujících podslovo $\mathtt{baa}$ a sudý počet symbolů $\mathtt{b}$
- $\overline{L_1}$ — množina všech slov, která neobsahují podslovo $\mathtt{baa}$
- $L_1 - L_2$ — množina všech slov, ve kterých se vyskytuje podslovo $\mathtt{baa}$, ale kde počet symbolů $\mathtt{b}$ není sudý

Zřetězení jazyků

Definice

Zřetězení jazyků L_1 a L_2, kde $L_1, L_2 \subseteq \Sigma^*$, je jazyk $L \subseteq \Sigma^*$ takový, že pro každé $w \in \Sigma^*$ platí

$$w \in L \leftrightarrow (\exists u \in L_1)(\exists v \in L_2)(w = u \cdot v)$$

Zřetězení jazyků L_1 a L_2 označujeme $L_1 \cdot L_2$.

Příklad:

$$\begin{aligned} L_1 &= \{abb, ba\} \\ L_2 &= \{a, ab, bbb\} \end{aligned}$$

Jazyk $L_1 \cdot L_2$ obsahuje slova:

$$abba \qquad abbab \qquad abbbbb \qquad baa \qquad baab \qquad babbb$$

Iterace jazyka

Definice

Iterace jazyka L, označovaná zápisem L^*, je jazyk tvořený slovy vzniklými zřetězením libovolného počtu slov z jazyka L.
Tj. $w \in L^*$ právě tehdy, když

$$\exists n \in \mathbb{N} : \exists w_1, w_2, \ldots, w_n \in L : w = w_1 w_2 \cdots w_n$$

Příklad: $L = \{aa, b\}$

$$L^* = \{\varepsilon, aa, b, aaaa, aab, baa, bb, aaaaaa, aaaab, aabaa, aabb, \ldots\}$$

Poznámka: Počet slov, která zřetězujeme, může být i 0, což znamená, že vždy platí $\varepsilon \in L^*$ (bez ohledu na to, zda $\varepsilon \in L$ nebo ne).

Iterace jazyka — alternativní definice

Nejprve definujeme pro jazyk L a číslo $k \in \mathbb{N}$ jazyk L^k:

$$L^0 = \{\varepsilon\}, \qquad L^k = L^{k-1} \cdot L \quad \text{pro } k \geq 1$$

To znamená

$$\begin{aligned} L^0 &= \{\varepsilon\} \\ L^1 &= L \\ L^2 &= L \cdot L \\ L^3 &= L \cdot L \cdot L \\ L^4 &= L \cdot L \cdot L \cdot L \\ L^5 &= L \cdot L \cdot L \cdot L \cdot L \\ &\ldots \end{aligned}$$

Příklad: Pro $L = \{aa, b\}$ jazyk L^3 obsahuje následující slova:

$$aaaaaa \quad aaaab \quad aabaa \quad aabb \quad baaaa \quad baab \quad bbaa \quad bbb$$

Iterace jazyka — alternativní definice

Alternativní definice

Iterace jazyka L je jazyk

$$L^* = \bigcup_{k \geq 0} L^k$$

Poznámka:

$$\bigcup_{k \geq 0} L^k = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \cdots$$

Iterace jazyka

Poznámka: Používá se také zápis L^+ jako zkratka za $L \cdot L^*$, tj.

$$L^+ = \bigcup_{k \geq 1} L^k$$

Zrcadlový obraz

Zrcadlový obraz slova w je slovo w zapsané „pozpátku“.

Zrcadlový obraz slova w značíme w^R.

Příklad: $w = \texttt{AHOJ}$ $w^R = \texttt{JOHA}$

Formálně můžeme definovat, že pro $w = a_1 a_2 \cdots a_n$ (kde $a_i \in \Sigma$) je $w^R = a_n a_{n-1} \cdots a_1$.

Zrcadlový obraz

Zrcadlový obraz jazyka L je jazyk tvořený zrcadlovými obrazy všech slov z jazyka L.

Zrcadlový obraz jazyka L značíme L^R.

$$L^R = \{w^R \mid w \in L\}$$

Příklad: $L = \{ab, baaba, aaab\}$

$L^R = \{ba, abaab, baaa\}$

Uspořádání na slovech

Předpokládejme určité (lineární) uspořádání $<$ symbolů abecedy Σ, tj. pokud $\Sigma = \{a_1, a_2, \ldots, a_n\}$, tak platí

$$a_1 < a_2 < \ldots < a_n \,.$$

Příklad: $\Sigma = \{a, b, c\}$, přičemž $a < b < c$.

Na množině Σ^* můžeme definovat následující (lineární) uspořádání $<_L$: $x <_L y$ právě tehdy, když:

- $|x| < |y|$, nebo
- $|x| = |y|$ a existují slova $u, v, w \in \Sigma^*$ a symboly $a, b \in \Sigma$ takové, že platí

$$x = uav \qquad y = ubw \qquad a < b$$

Neformálně můžeme říct v uspořádání $<_L$ řadíme slova podle délky a v rámci stejné délky lexikograficky (podle abecedy).

Uspořádání na slovech

Všechna slova nad abecedou Σ můžeme pomocí uspořádání $<_L$ seřadit do posloupnosti

$$w_0, w_1, w_2, \ldots$$

ve které se každé slovo $w \in \Sigma^*$ vyskytuje právě jednou a kde pro libovolná $i, j \in \mathbb{N}$ platí, že $w_i <_L w_j$ právě tehdy, když $i < j$.

Příklad: Pro abecedu $\Sigma = \{a, b, c\}$ (kde $a < b < c$) bude začátek posloupnosti vypadat následovně:

$$\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, abb, abc, \ldots$$

Pokud budeme mluvit například o prvních deseti slovech jazyka $L \subseteq \Sigma^*$, máme tím na mysli deset slov, která patří do jazyka L a jsou mezi všemi slovy z jazyka L nejmenší vzhledem k uspořádání $<_L$.

Regulární výrazy

Regulární výrazy

Regulární výrazy popisující jazyky nad abecedou Σ:

- $\emptyset$, ε, a (kde $a \in \Sigma$) jsou regulární výrazy:

  $\emptyset$ ... označuje prázdný jazyk
  ε ... označuje jazyk $\{\varepsilon\}$
  a ... označuje jazyk $\{a\}$

- Jestliže α, β jsou regulární výrazy, pak i $(\alpha + \beta)$, $(\alpha \cdot \beta)$, (α^*) jsou regulární výrazy:

  $(\alpha + \beta)$ ... označuje sjednocení jazyků označených α a β
  $(\alpha \cdot \beta)$ ... označuje zřetězení jazyků označených α a β
  (α^*) ... označuje iteraci jazyka označeného α

- Neexistují žádné další regulární výrazy než ty definované podle předchozích dvou bodů.

Regulární výrazy

Příklad: abeceda $\Sigma = \{0, 1\}$

- Podle definice jsou 0 i 1 regulární výrazy.

Regulární výrazy

Příklad: abeceda $\Sigma = \{0, 1\}$

- Podle definice jsou 0 i 1 regulární výrazy.
- Protože 0 i 1 jsou regulární výrazy, je i $(0 + 1)$ regulární výraz.

Regulární výrazy

Příklad: abeceda $\Sigma = \{0, 1\}$

- Podle definice jsou 0 i 1 regulární výrazy.
- Protože 0 i 1 jsou regulární výrazy, je i $(0 + 1)$ regulární výraz.
- Protože 0 je regulární výraz, je i (0^*) regulární výraz.

Regulární výrazy

Příklad: abeceda $\Sigma = \{0, 1\}$

- Podle definice jsou 0 i 1 regulární výrazy.
- Protože 0 i 1 jsou regulární výrazy, je i $(0 + 1)$ regulární výraz.
- Protože 0 je regulární výraz, je i (0^*) regulární výraz.
- Protože $(0 + 1)$ i (0^*) jsou regulární výrazy, je i $((0 + 1) \cdot (0^*))$ regulární výraz.

Regulární výrazy

Příklad: abeceda $\Sigma = \{0, 1\}$

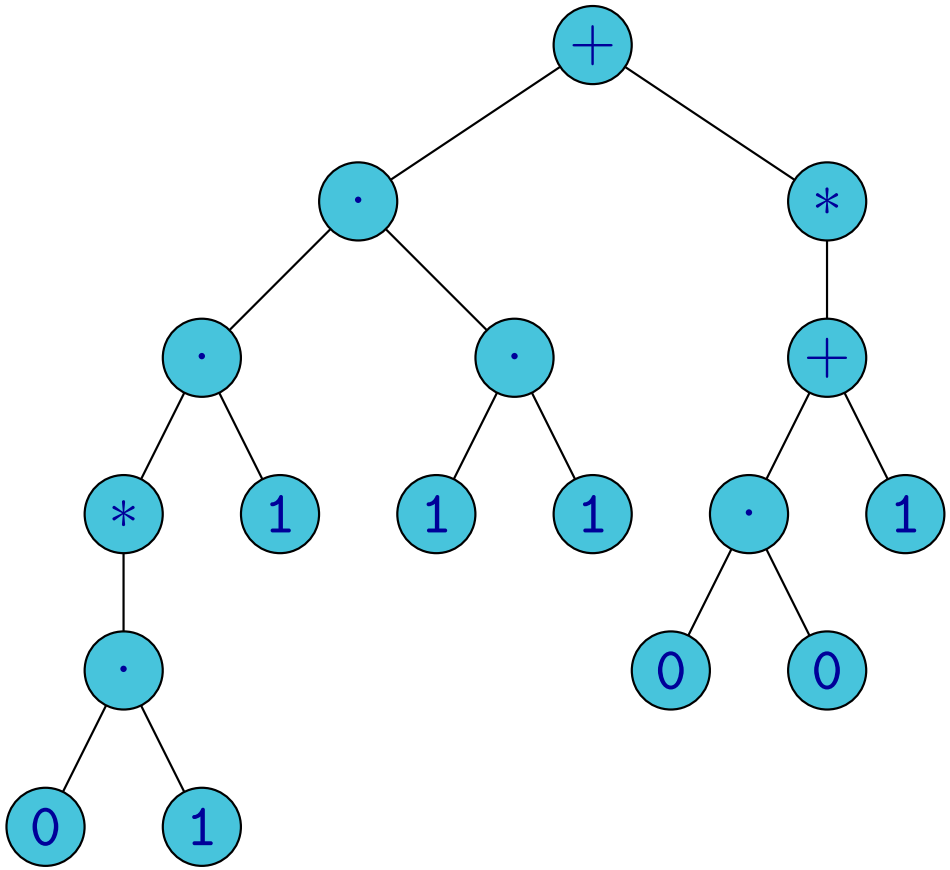
- Podle definice jsou 0 i 1 regulární výrazy.
- Protože 0 i 1 jsou regulární výrazy, je i $(0 + 1)$ regulární výraz.
- Protože 0 je regulární výraz, je i (0^*) regulární výraz.
- Protože $(0 + 1)$ i (0^*) jsou regulární výrazy, je i $((0 + 1) \cdot (0^*))$ regulární výraz.

Poznámka: Jestliže α je regulární výraz, zápisem $[\alpha]$ označujeme jazyk definovaný regulárním výrazem α.

$$[((0 + 1) \cdot (0^*))] = \{0,\ 1,\ 00,\ 10,\ 000,\ 100,\ 0000,\ 1000,\ 00000,\ \dots\}$$

Regulární výrazy

Strukturu regulárního výrazu si můžeme znázornit abstraktním syntaktickým stromem:



$$(((((0 \cdot 1)^*) \cdot 1) \cdot (1 \cdot 1)) + (((0 \cdot 0) + 1)^*))$$

Regulární výrazy

Popis (abstraktní) syntaxe regulárních výrazů pomocí Backus-Naurovy formy:

$$\alpha \ ::= \ \emptyset \ \mid \ \varepsilon \ \mid \ a \ \mid \ \alpha^* \ \mid \ \alpha \cdot \alpha \ \mid \ \alpha + \alpha$$

Formální definice sémantiky regulárních výrazů:

- $[\emptyset] = \emptyset$
- $[\varepsilon] = \{\varepsilon\}$
- $[a] = \{a\}$
- $[\alpha^*] = [\alpha]^*$
- $[\alpha \cdot \beta] = [\alpha] \cdot [\beta]$
- $[\alpha + \beta] = [\alpha] \cup [\beta]$

Regulární výrazy

Aby byl zápis regulárních výrazů přehlednější a stručnější, používáme následují pravidla:

- Vynecháváme vnější pár závorek.
- Vynecháváme závorky, které jsou zbytečné vzhledem k asociativitě operací sjednocení ($+$) a zřetězení ($\cdot$).
- Vynecháváme závorky, které jsou zbytečné vzhledem k prioritě operací (nejvyšší prioritu má iterace (*), menší zřetězení ($\cdot$) a nejmenší sjednocení ($+$)).
- Nepíšeme tečku pro zřetězení.

Příklad: Místo

$$(((((0 \cdot 1)^*) \cdot 1) \cdot (1 \cdot 1)) + (((0 \cdot 0) + 1)^*))$$

obvykle píšeme

$$(01)^*111 + (00 + 1)^*$$

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 . . . jazyk tvořený jediným slovem 0

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 . . . jazyk tvořený jediným slovem 0

01 . . . jazyk tvořený jediným slovem 01

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 ... jazyk tvořený jediným slovem 0

01 ... jazyk tvořený jediným slovem 01

$0 + 1$ ... jazyk tvořený dvěma slovy 0 a 1

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 ... jazyk tvořený jediným slovem 0

01 ... jazyk tvořený jediným slovem 01

$0 + 1$ ... jazyk tvořený dvěma slovy 0 a 1

0^* ... jazyk tvořený slovy ε, 0, 00, 000, ...

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 ... jazyk tvořený jediným slovem 0

01 ... jazyk tvořený jediným slovem 01

$0 + 1$ ... jazyk tvořený dvěma slovy 0 a 1

0^* ... jazyk tvořený slovy ε, 0, 00, 000, ...

$(01)^*$ ... jazyk tvořený slovy ε, 01, 0101, 010101, ...

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 ... jazyk tvořený jediným slovem 0

01 ... jazyk tvořený jediným slovem 01

$0 + 1$ ... jazyk tvořený dvěma slovy 0 a 1

0^* ... jazyk tvořený slovy ε, 0, 00, 000, ...

$(01)^*$ ... jazyk tvořený slovy ε, 01, 0101, 010101, ...

$(0 + 1)^*$ ... jazyk tvořený všemi slovy nad abecedou $\{0, 1\}$

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 ... jazyk tvořený jediným slovem 0

01 ... jazyk tvořený jediným slovem 01

$0 + 1$ ... jazyk tvořený dvěma slovy 0 a 1

0^* ... jazyk tvořený slovy ε, 0, 00, 000, ...

$(01)^*$ ... jazyk tvořený slovy ε, 01, 0101, 010101, ...

$(0 + 1)^*$ ... jazyk tvořený všemi slovy nad abecedou $\{0, 1\}$

$(0 + 1)^*00$ ... jazyk tvořený všemi slovy končícími 00

Regulární výrazy

Příklady: Ve všech případech $\Sigma = \{0, 1\}$.

0 ... jazyk tvořený jediným slovem 0

01 ... jazyk tvořený jediným slovem 01

$0 + 1$ ... jazyk tvořený dvěma slovy 0 a 1

0^* ... jazyk tvořený slovy ε, 0, 00, 000, ...

$(01)^*$ ... jazyk tvořený slovy ε, 01, 0101, 010101, ...

$(0 + 1)^*$ ... jazyk tvořený všemi slovy nad abecedou $\{0, 1\}$

$(0 + 1)^*00$ ... jazyk tvořený všemi slovy končícími 00

$(01)^*111(01)^*$ ... jazyk tvořený všemi slovy obsahujícími podslovo 111 předcházené i následované libovolným počtem slov 01

Regulární výrazy

$(0+1)^*00 + (01)^*111(01)^*$ ... jazyk tvořený všemi slovy, která buď končí 00 nebo obsahují podslovo 111 předcházené i následované libovolným počtem slov 01

Regulární výrazy

$(0+1)^*00+(01)^*111(01)^*$ ... jazyk tvořený všemi slovy, která buď končí 00 nebo obsahují podslovo 111 předcházené i následované libovolným počtem slov 01

$(0+1)^*1(0+1)^*$ ... jazyk tvořený všemi slovy obsahujícími alespoň jeden symbol 1

Regulární výrazy

$(0+1)^*00+(01)^*111(01)^*$ ... jazyk tvořený všemi slovy, která buď končí 00 nebo obsahují podslovo 111 předcházené i následované libovolným počtem slov 01

$(0+1)^*1(0+1)^*$ ... jazyk tvořený všemi slovy obsahujícími alespoň jeden symbol 1

$0^*(10^*10^*)^*$ ... jazyk tvořený všemi slovy obsahujícími sudý počet symbolů 1

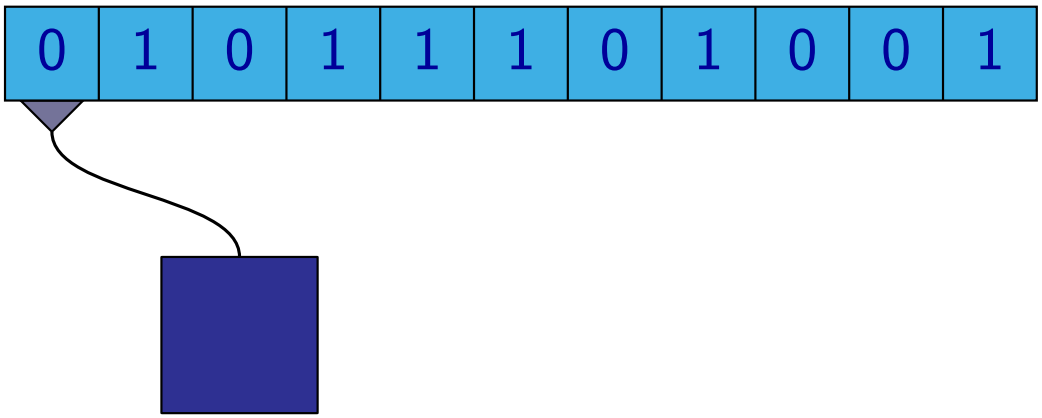
Konečné automaty

Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

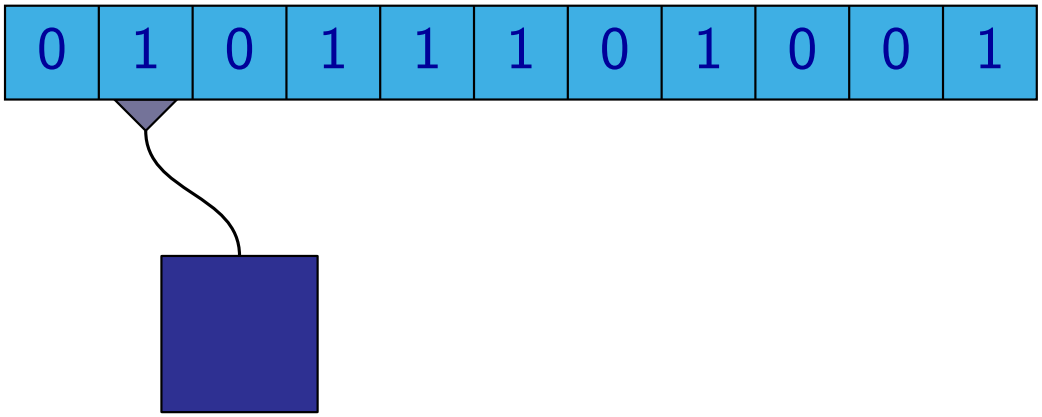


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

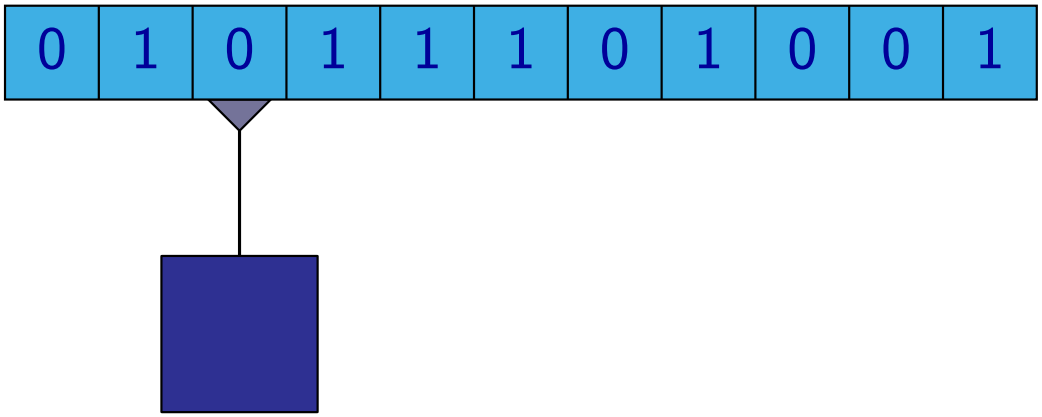


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

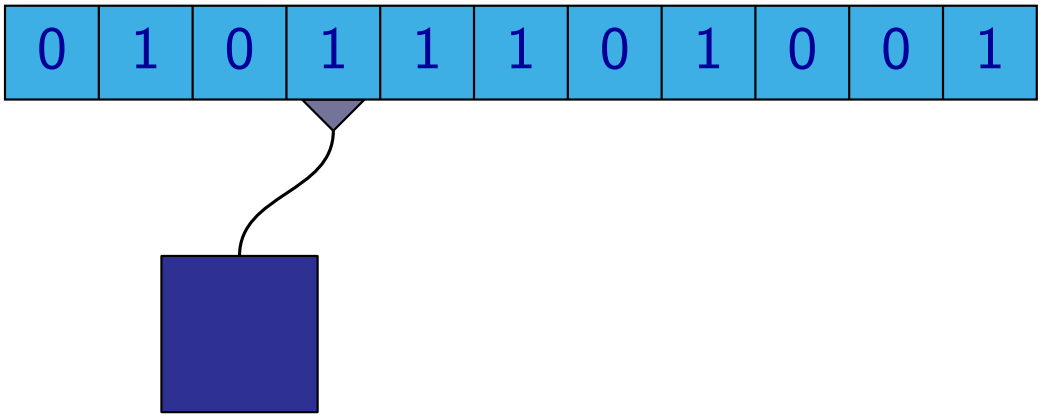


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

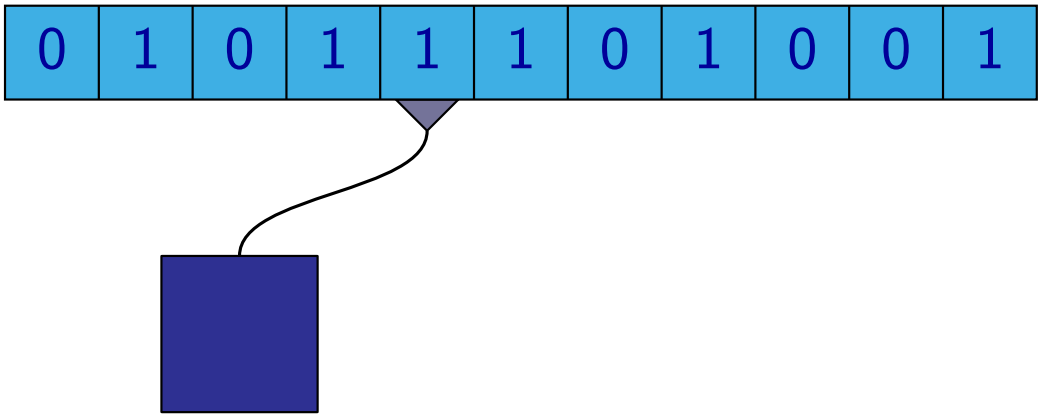


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

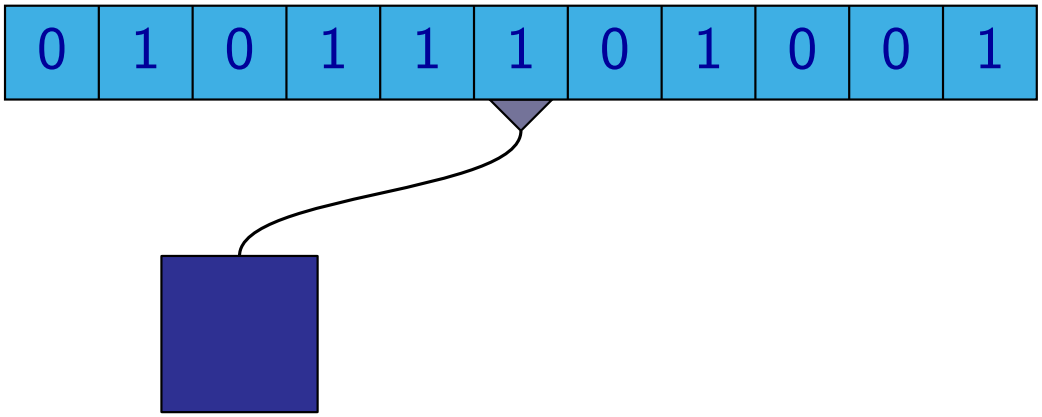


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

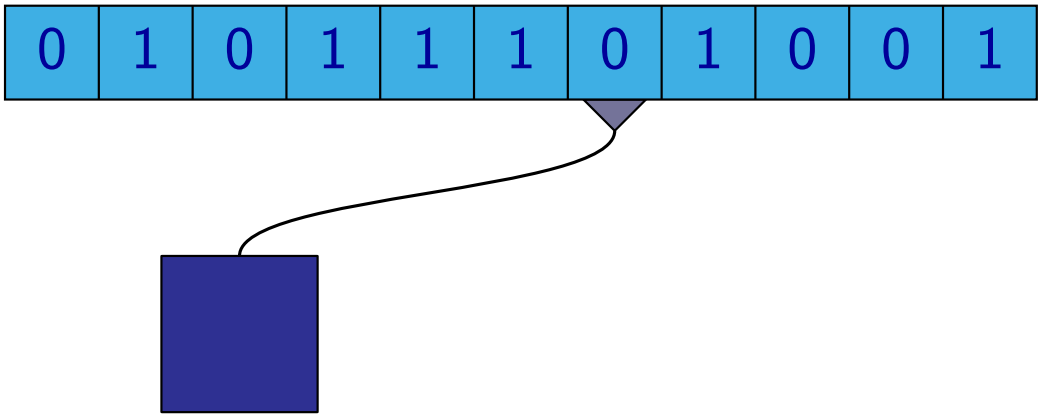


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

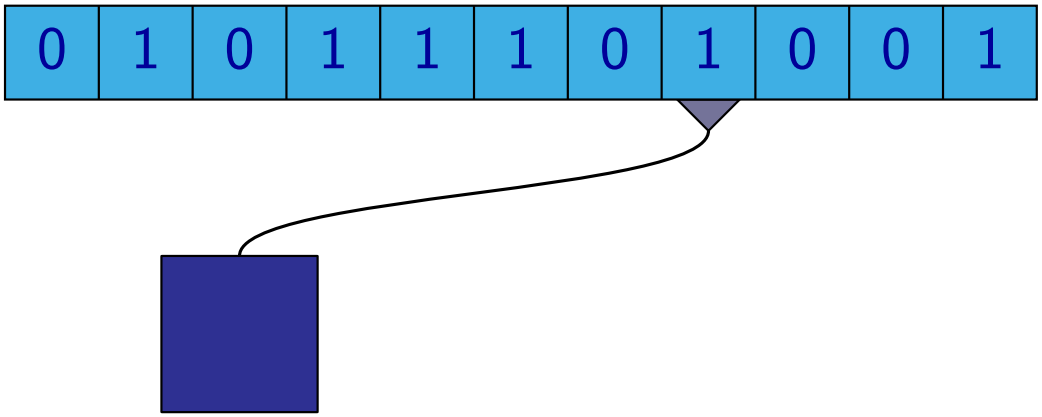


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

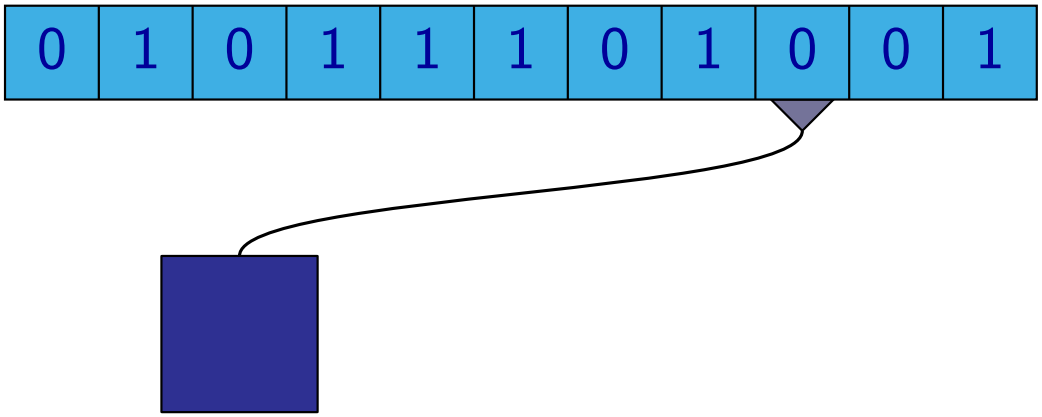


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

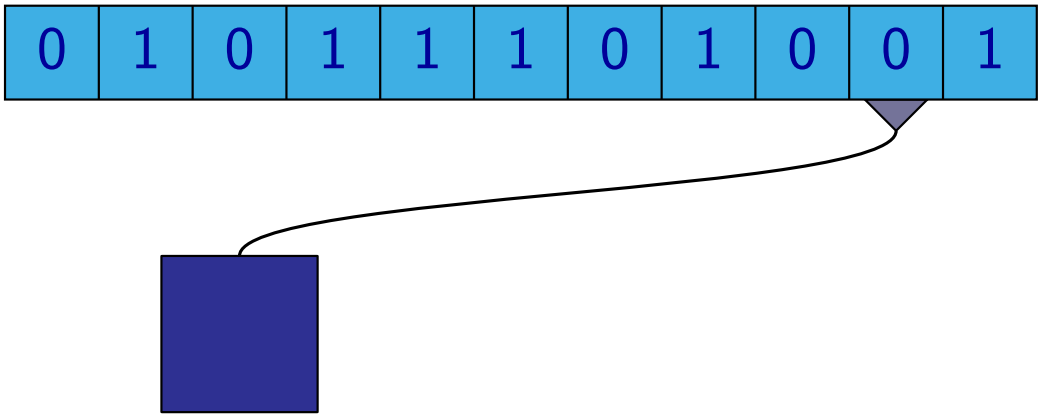


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

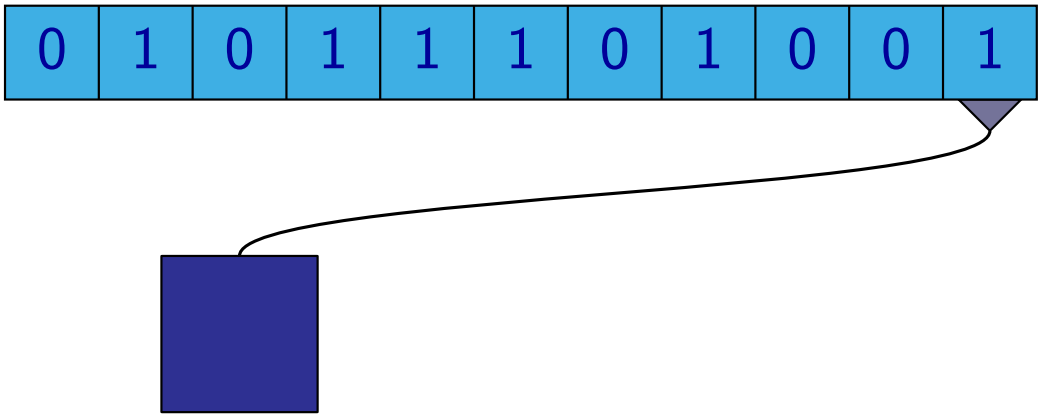


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.

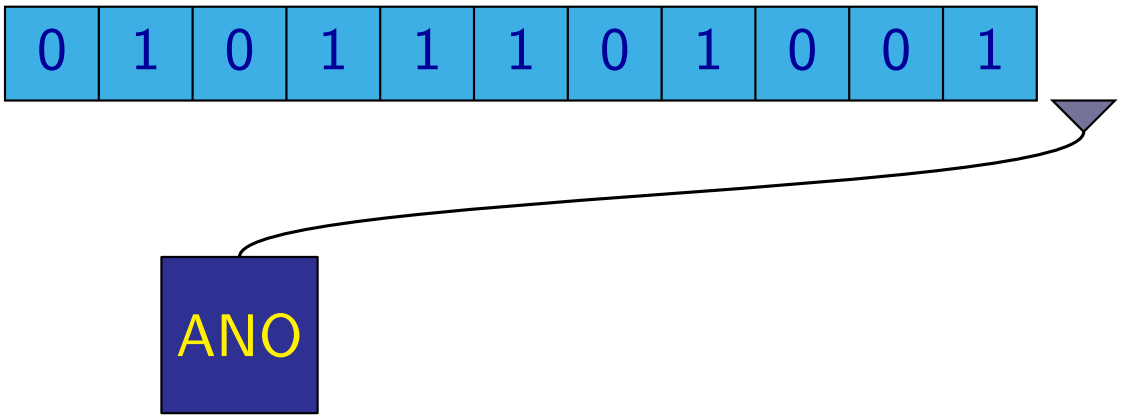


Rozpoznávání jazyka

Příklad: Uvažujme slova nad abecedou $\{0, 1\}$.

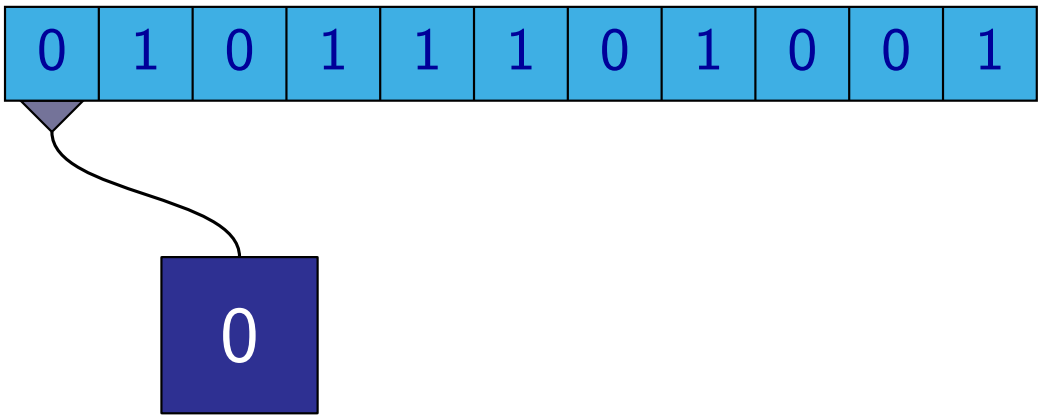
Chtěli bychom rozpoznávat jazyk L, který je tvořen slovy, ve kterých se vyskytuje sudý počet symbolů 1.

Chceme navrhnout zařízení, které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.



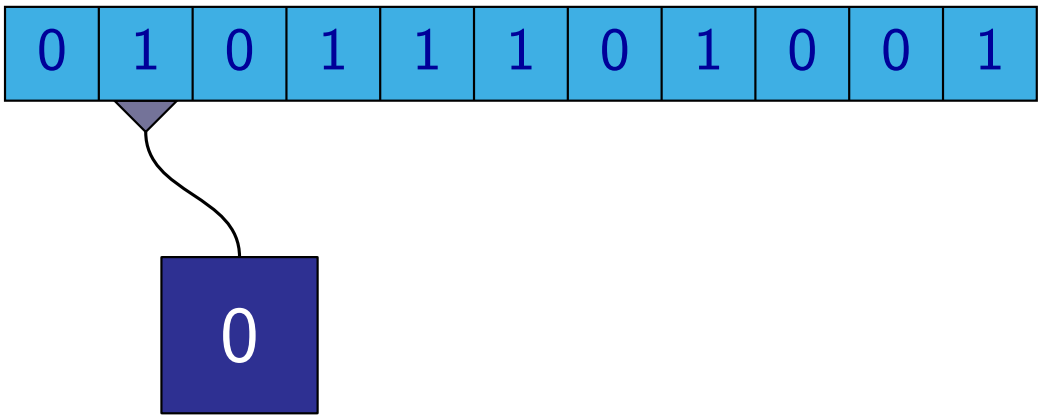
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



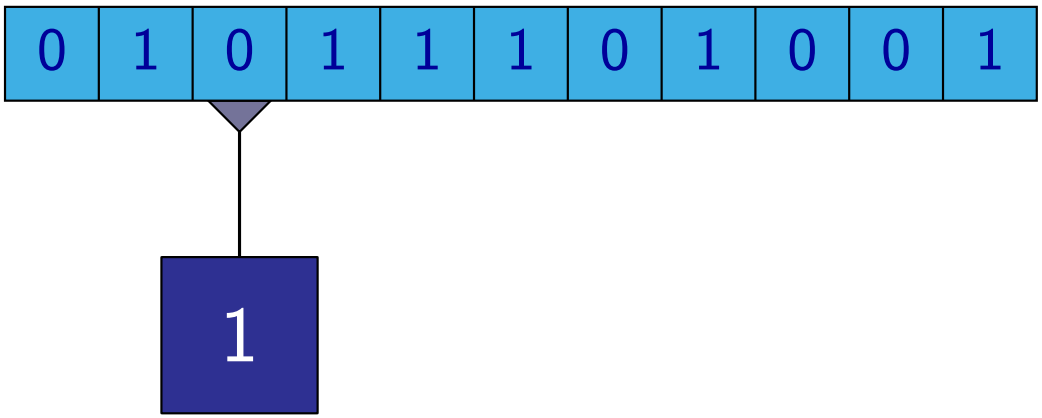
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



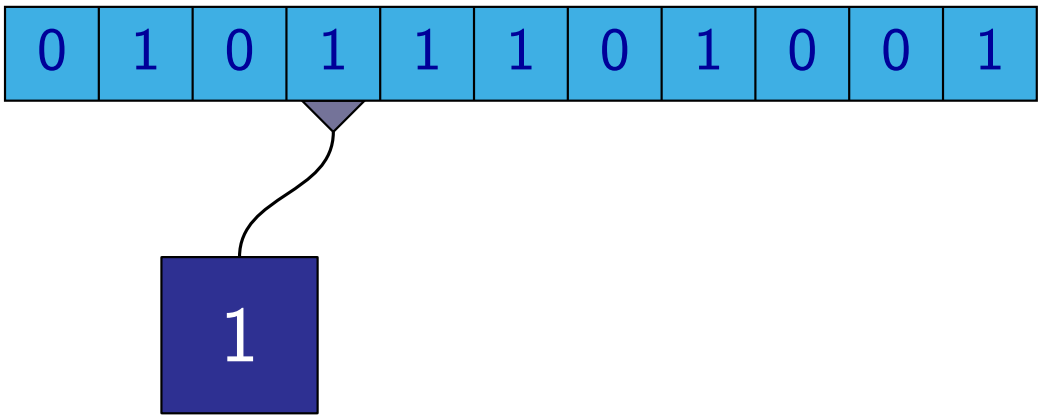
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



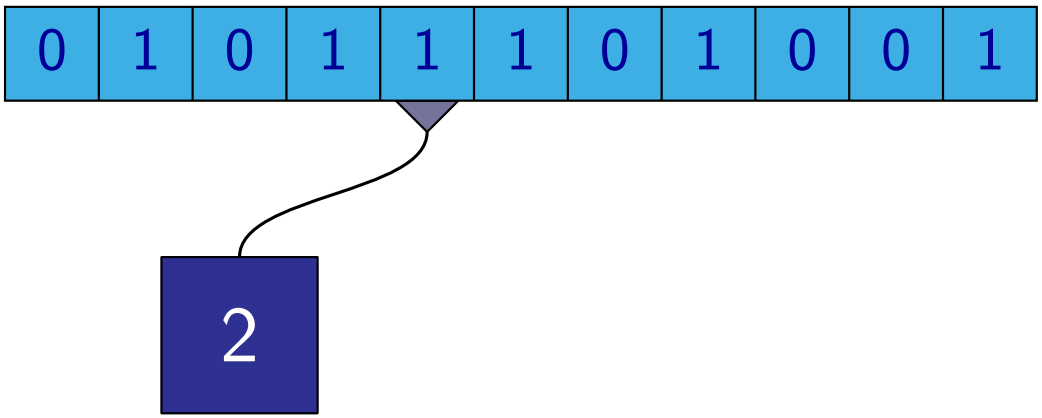
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



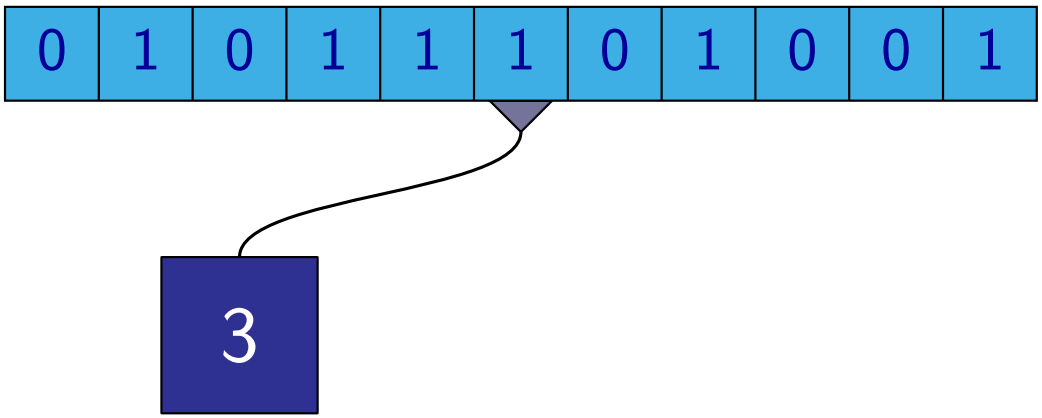
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



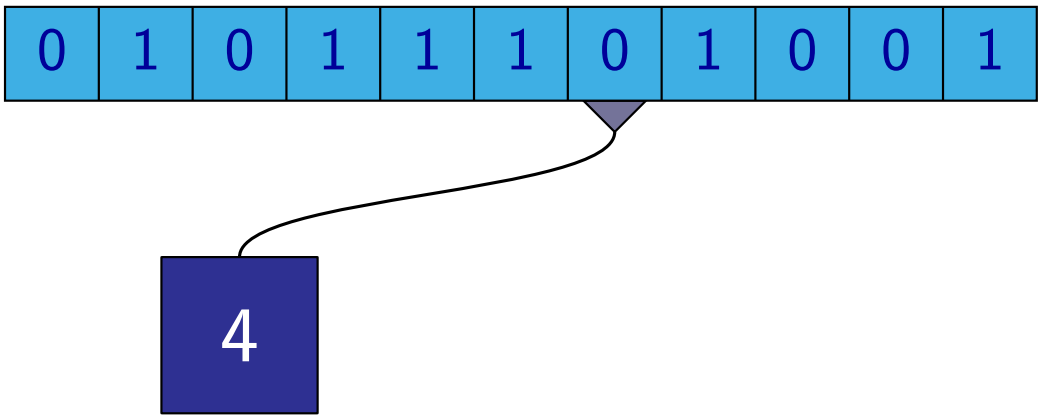
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



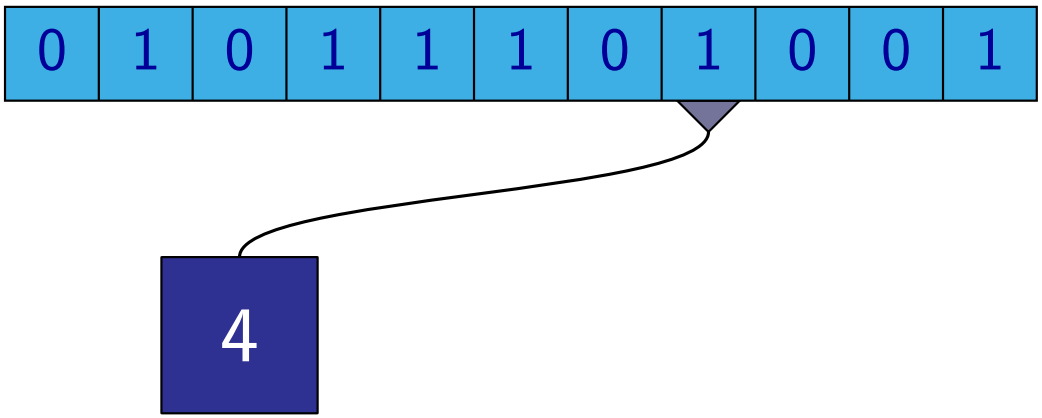
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



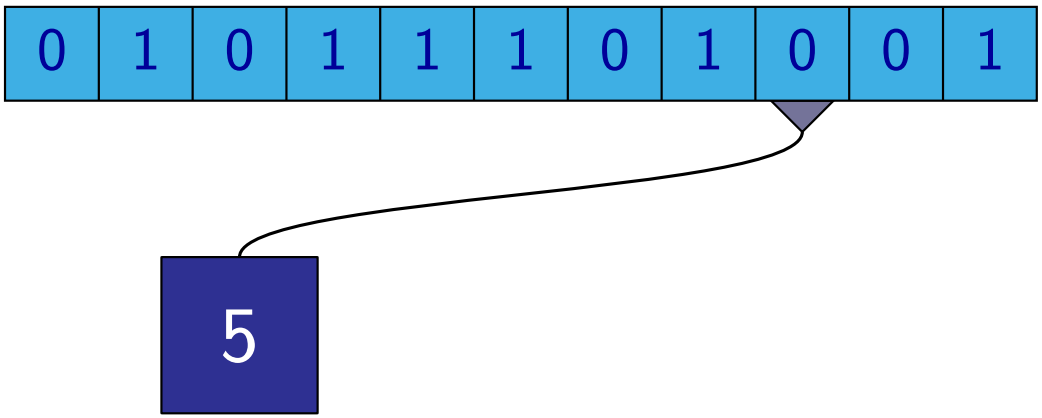
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



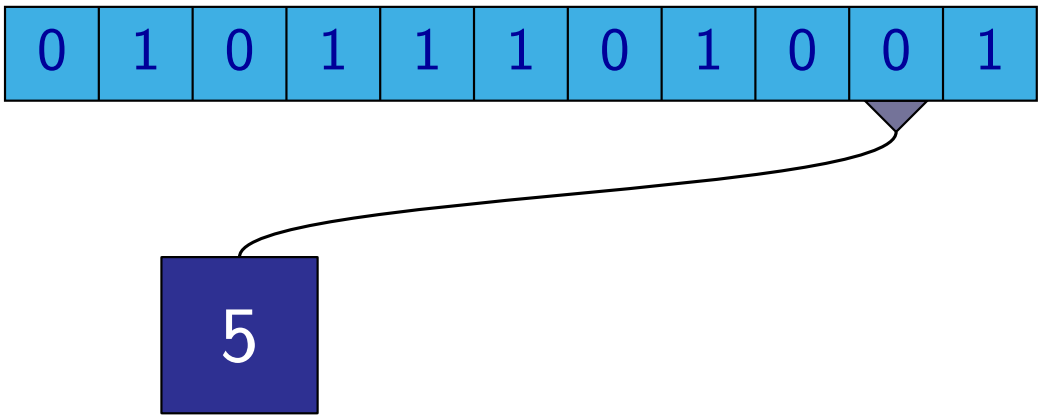
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



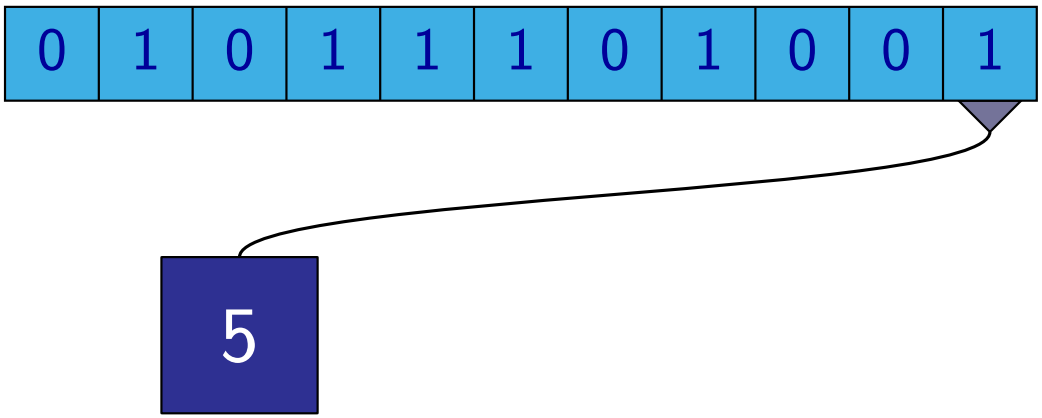
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



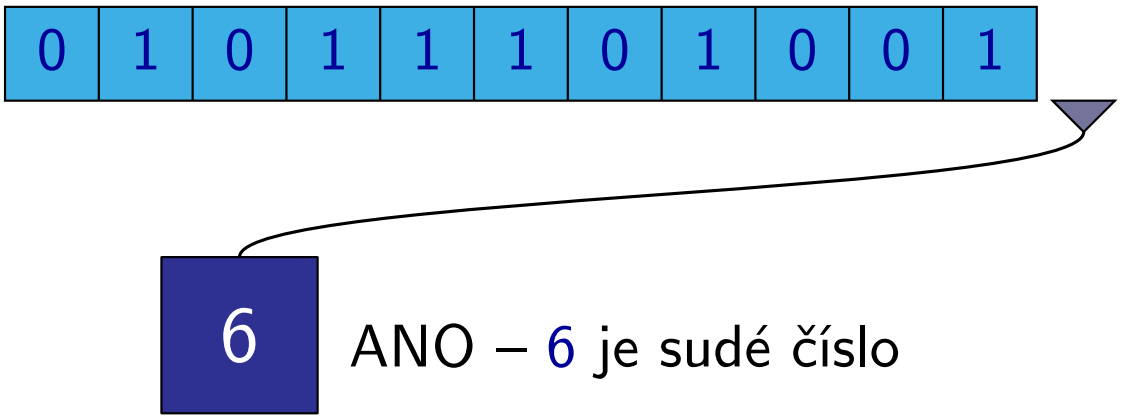
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



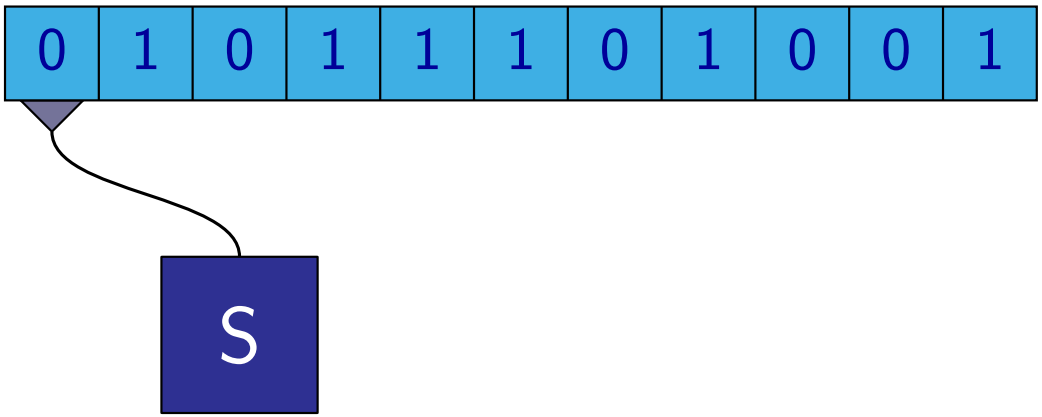
Rozpoznávání jazyka

První nápad: Počítat počet výskytů symbolů 1.



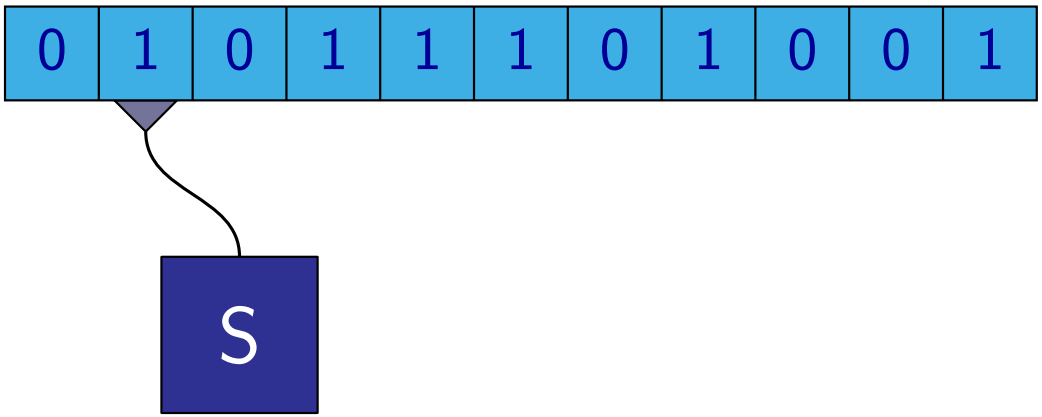
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



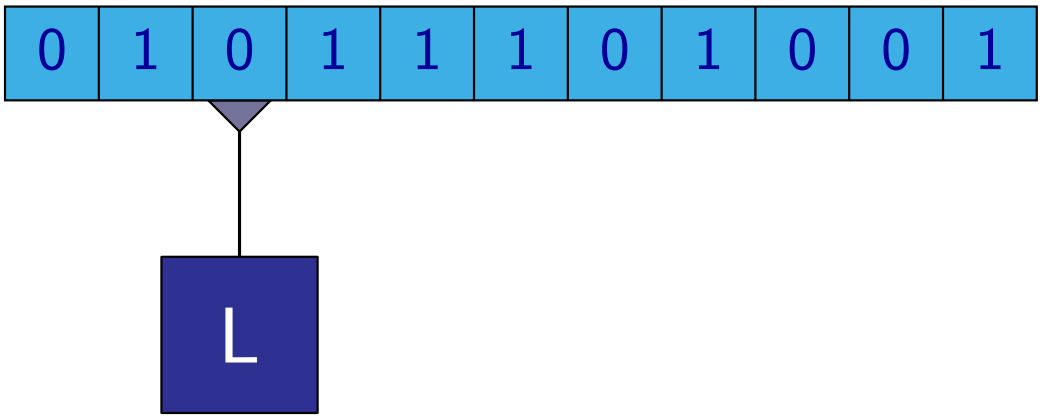
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



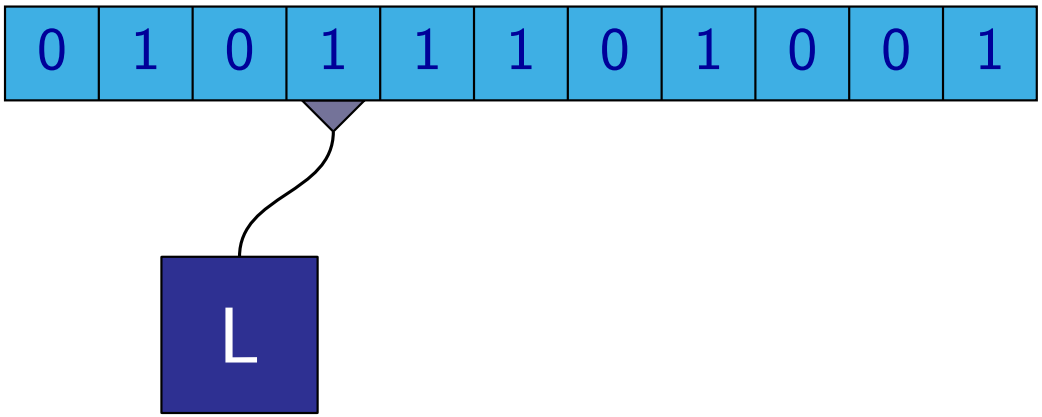
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



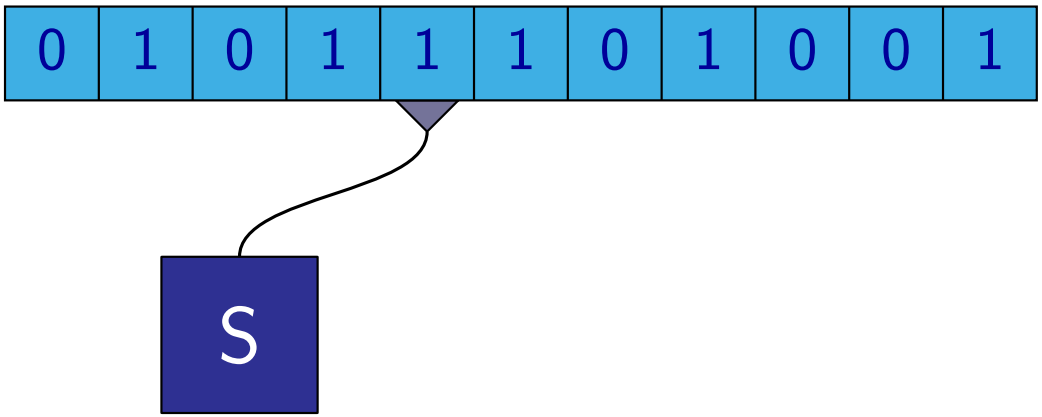
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



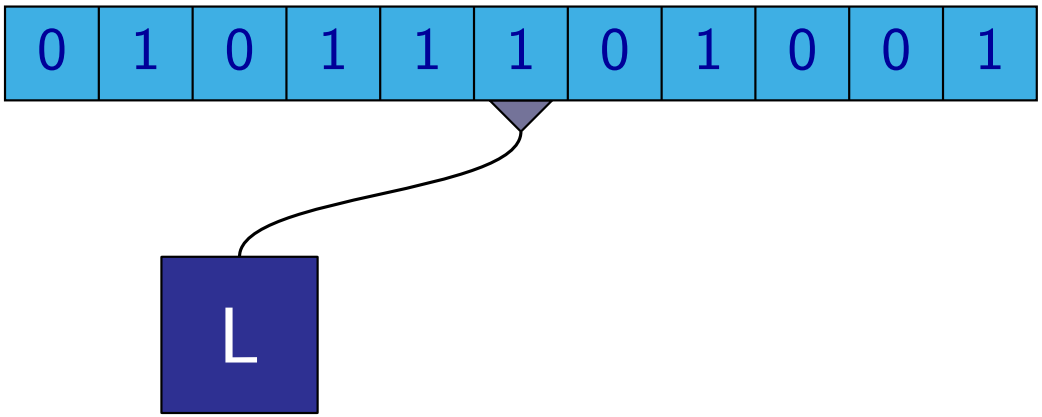
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



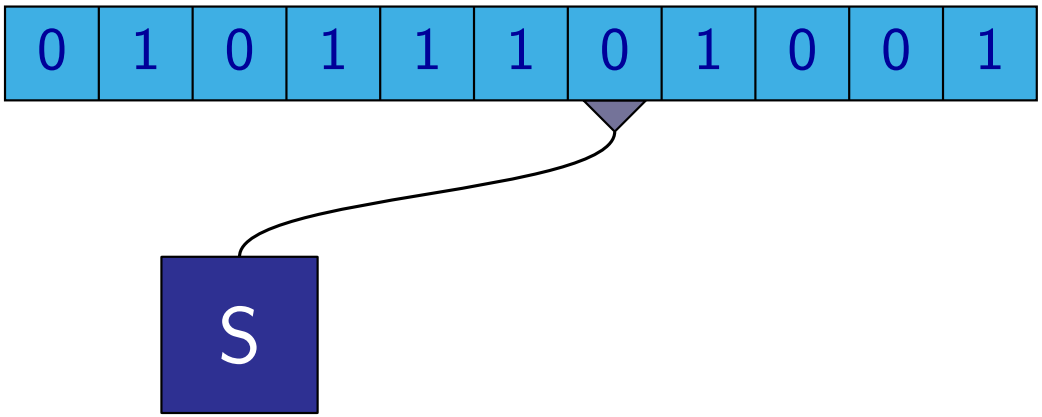
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



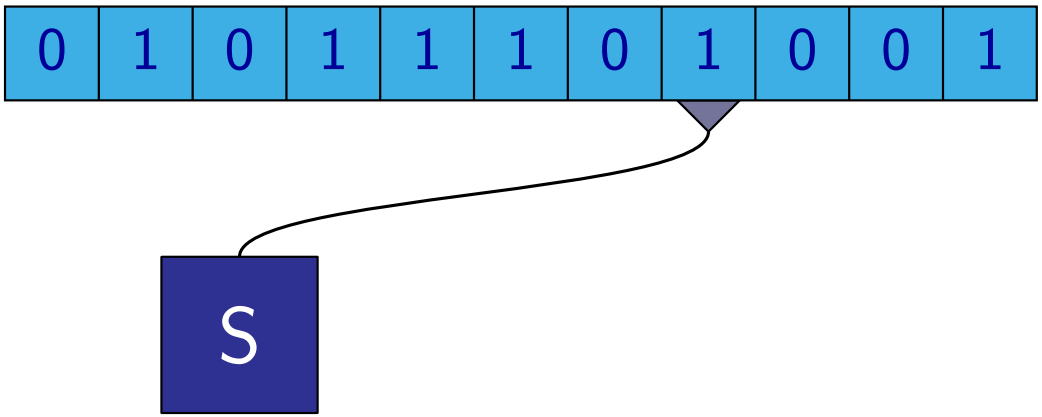
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



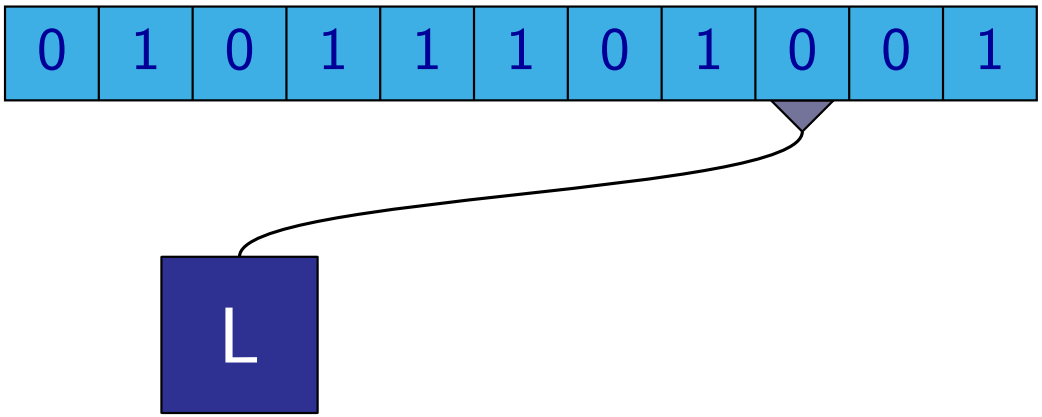
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



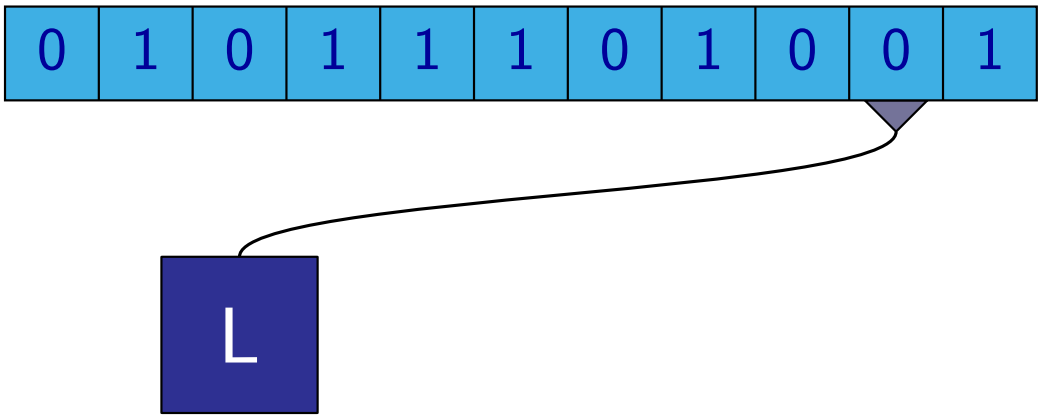
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



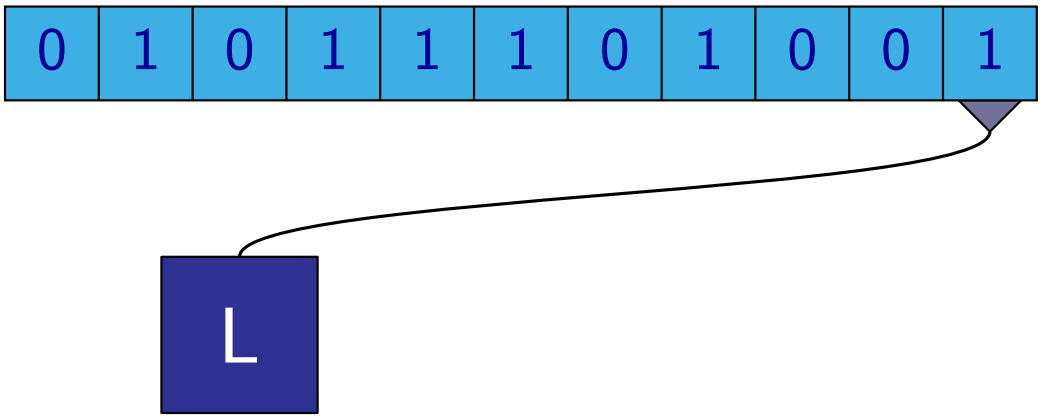
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



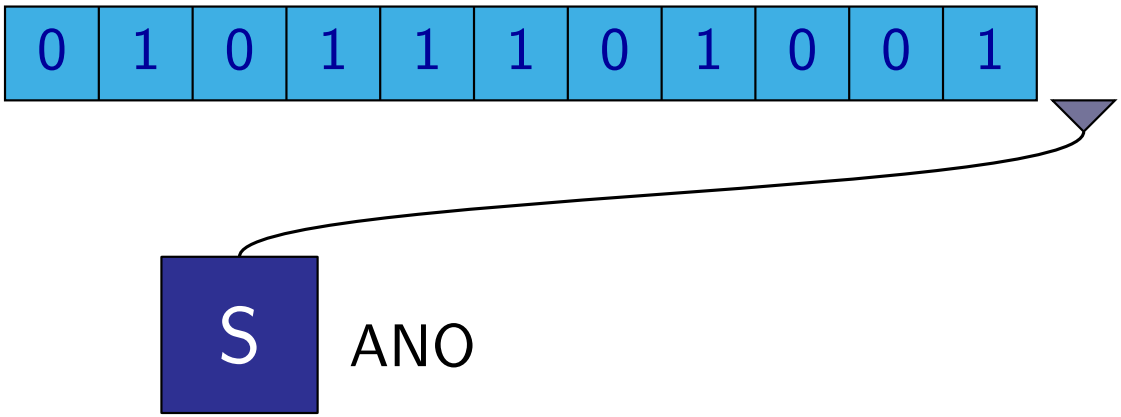
Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



Rozpoznávání jazyka

Druhý nápad: Ve skutečnosti nás zajímá pouze, zda počet dosud přečtených symbolů 1 je sudý nebo lichý (tj. místo čísla si stačí pamatovat jen jeho poslední bit).



Rozpoznávání jazyka

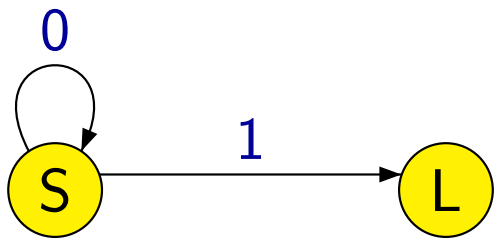
Chovaní tohoto zařízení můžeme popsat grafem:

Rozpoznávání jazyka

Chovaní tohoto zařízení můžeme popsat grafem:

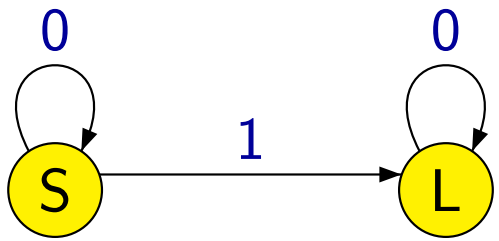
Rozpoznávání jazyka

Chovaní tohoto zařízení můžeme popsat grafem:



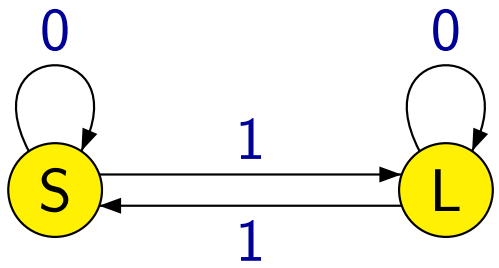
Rozpoznávání jazyka

Chovaní tohoto zařízení můžeme popsat grafem:



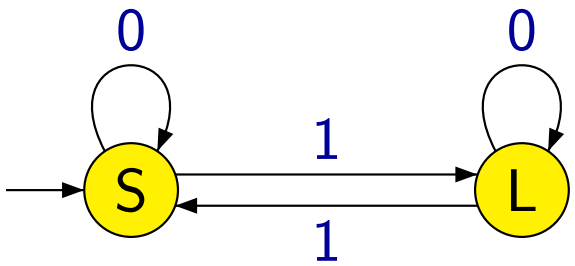
Rozpoznávání jazyka

Chovaní tohoto zařízení můžeme popsat grafem:



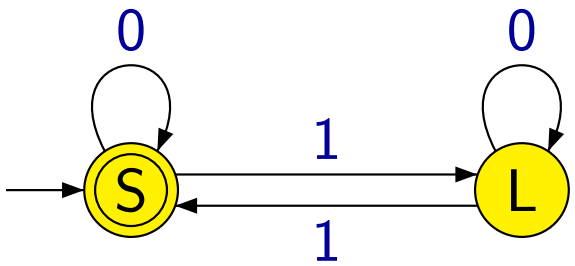
Rozpoznávání jazyka

Chovaní tohoto zařízení můžeme popsat grafem:



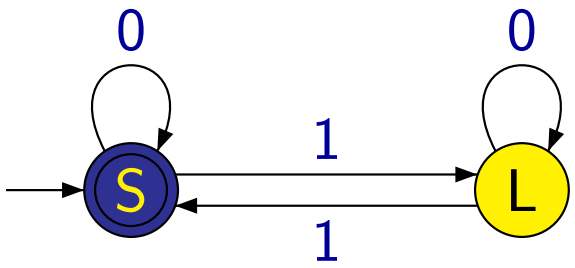
Rozpoznávání jazyka

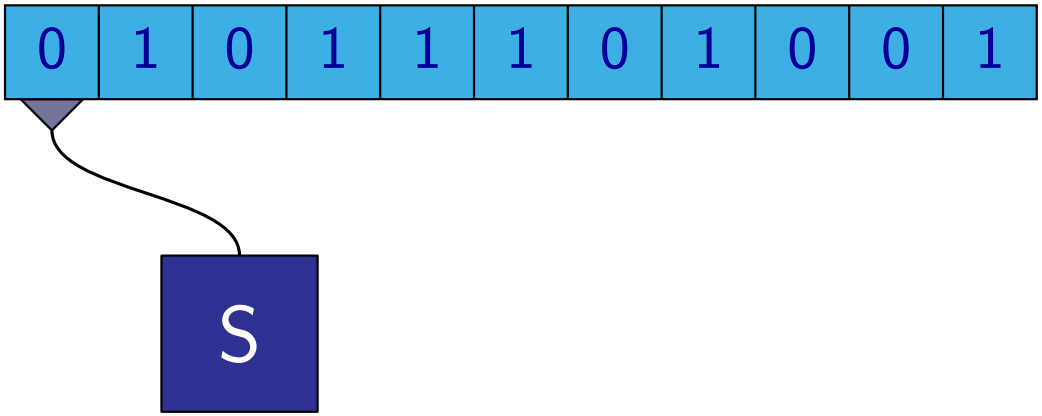
Chovaní tohoto zařízení můžeme popsat grafem:



Rozpoznávání jazyka

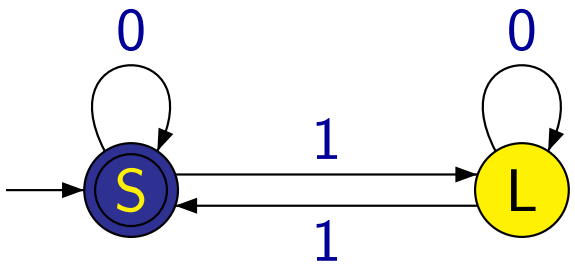
Chovaní tohoto zařízení můžeme popsat grafem:

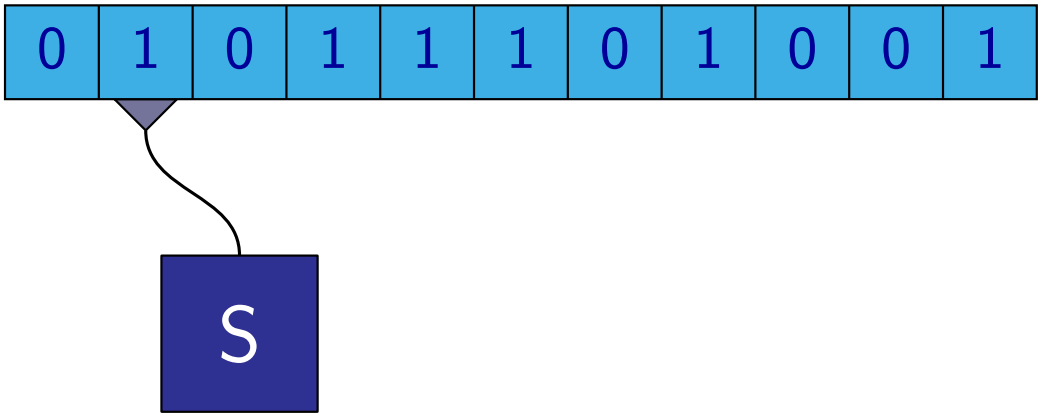




Rozpoznávání jazyka

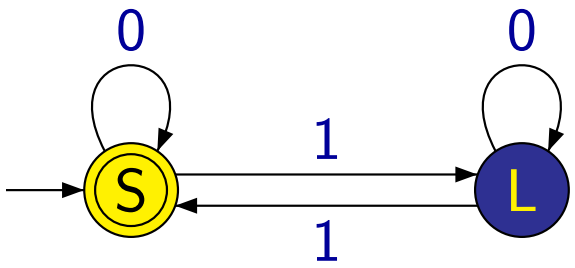
Chovaní tohoto zařízení můžeme popsat grafem:

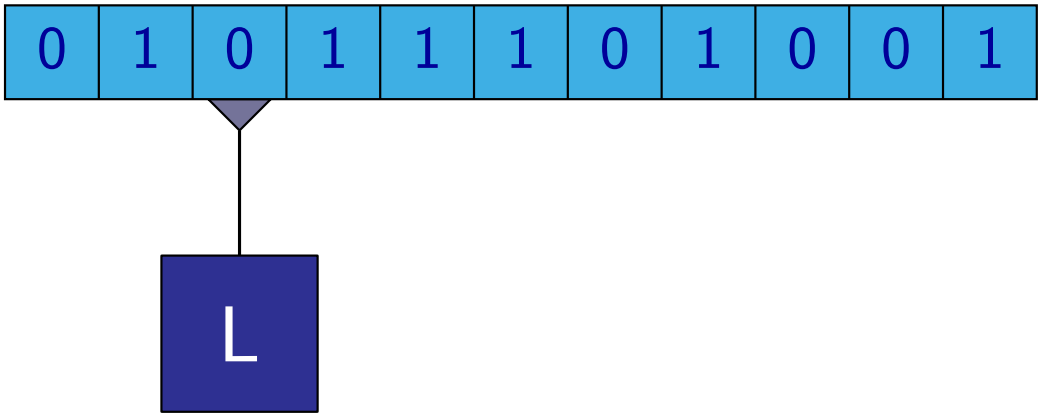




Rozpoznávání jazyka

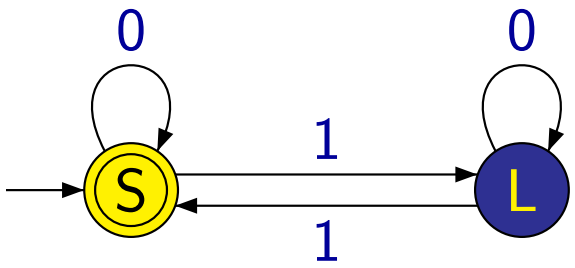
Chovaní tohoto zařízení můžeme popsat grafem:

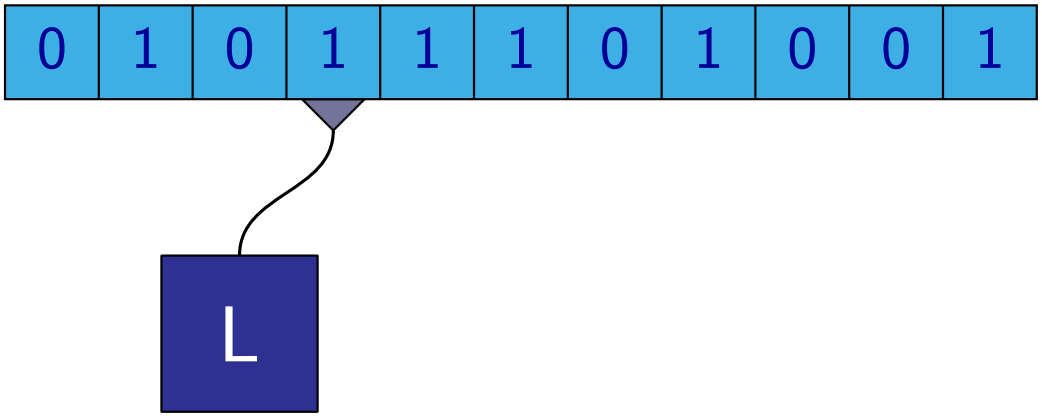




Rozpoznávání jazyka

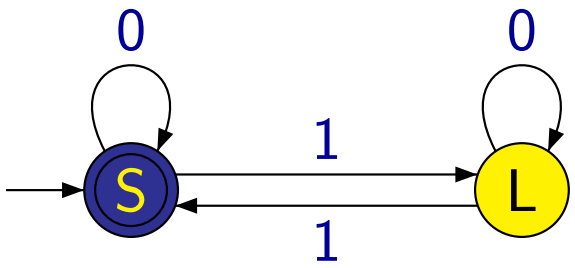
Chovaní tohoto zařízení můžeme popsat grafem:

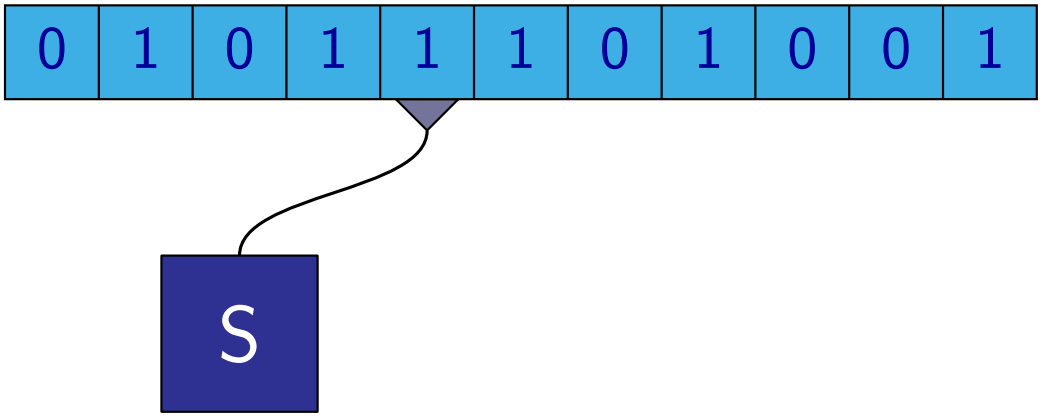




Rozpoznávání jazyka

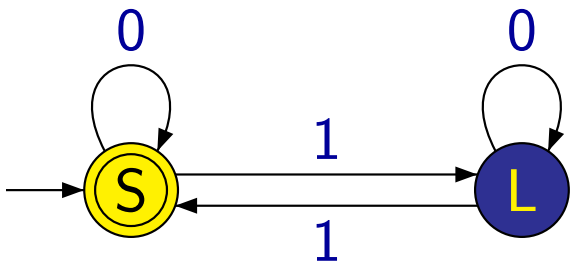
Chovaní tohoto zařízení můžeme popsat grafem:

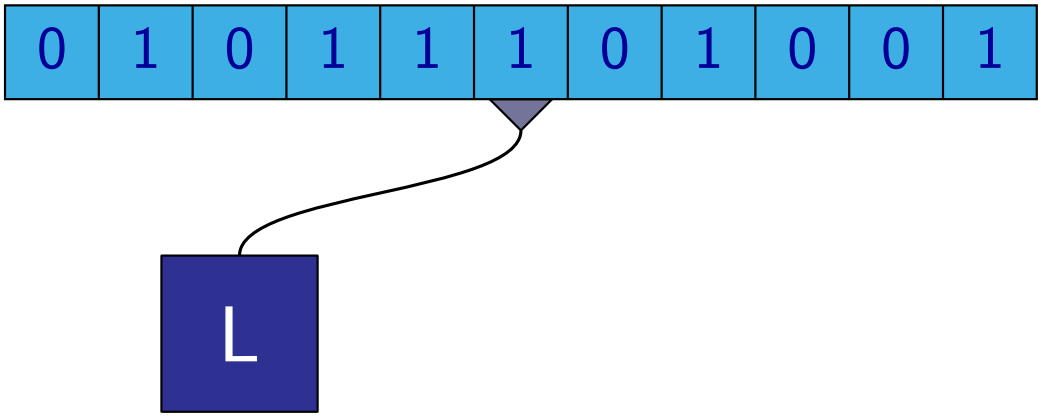




Rozpoznávání jazyka

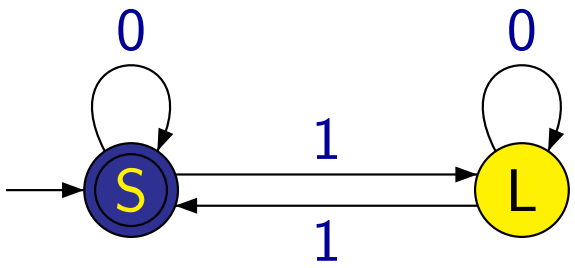
Chovaní tohoto zařízení můžeme popsat grafem:

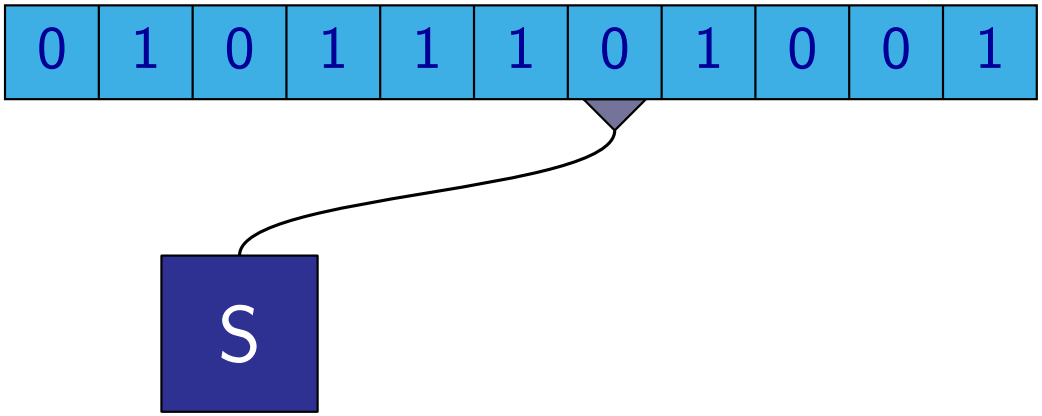




Rozpoznávání jazyka

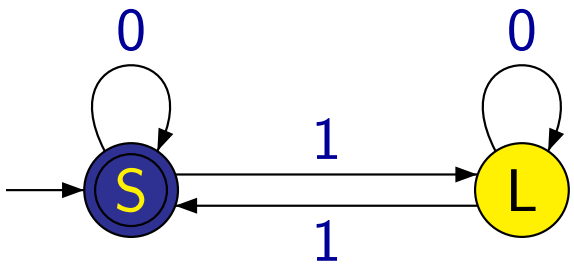
Chovaní tohoto zařízení můžeme popsat grafem:

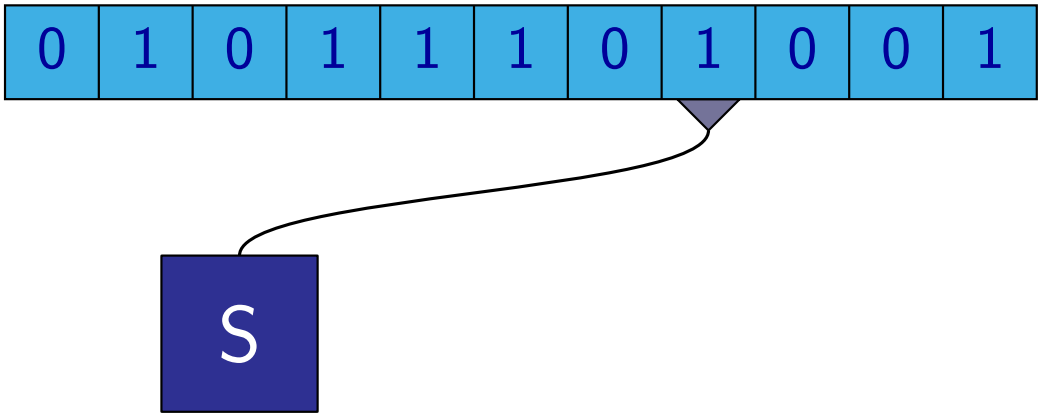




Rozpoznávání jazyka

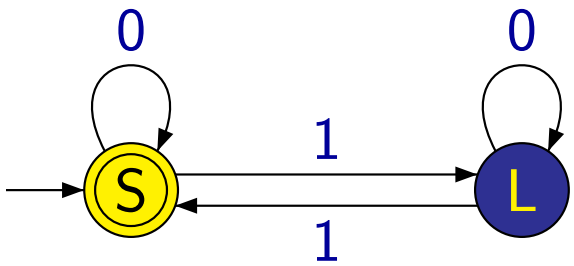
Chovaní tohoto zařízení můžeme popsat grafem:

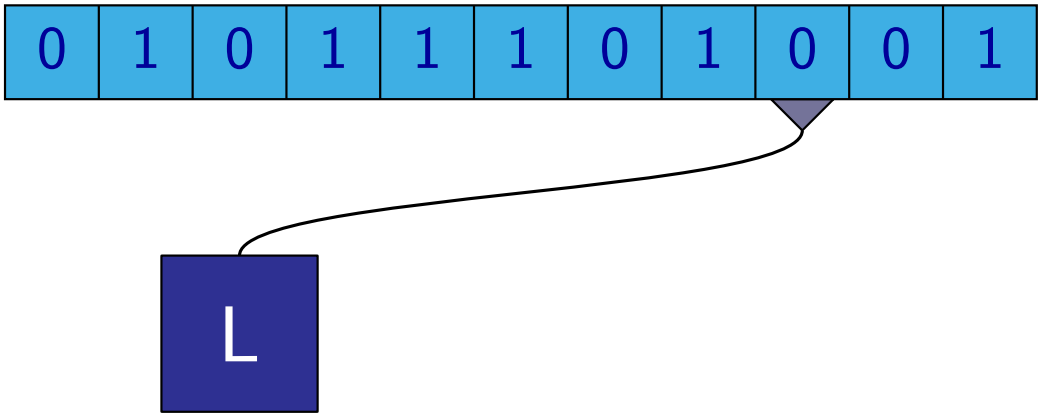




Rozpoznávání jazyka

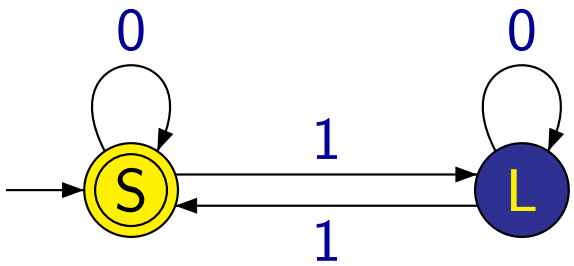
Chovaní tohoto zařízení můžeme popsat grafem:

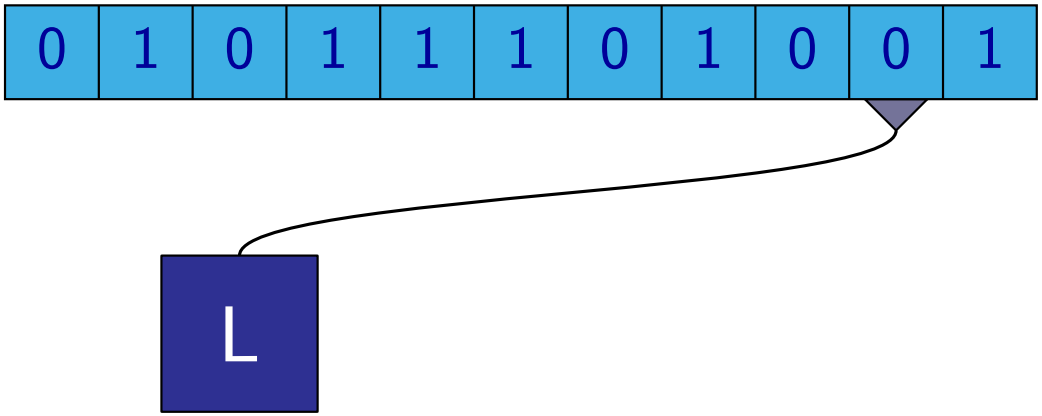




Rozpoznávání jazyka

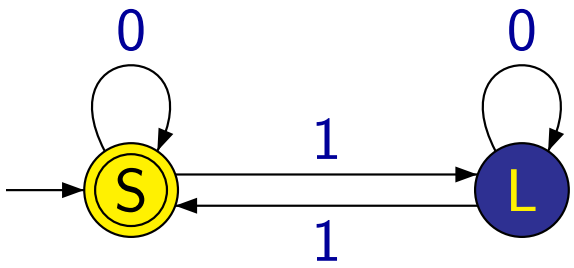
Chovaní tohoto zařízení můžeme popsat grafem:

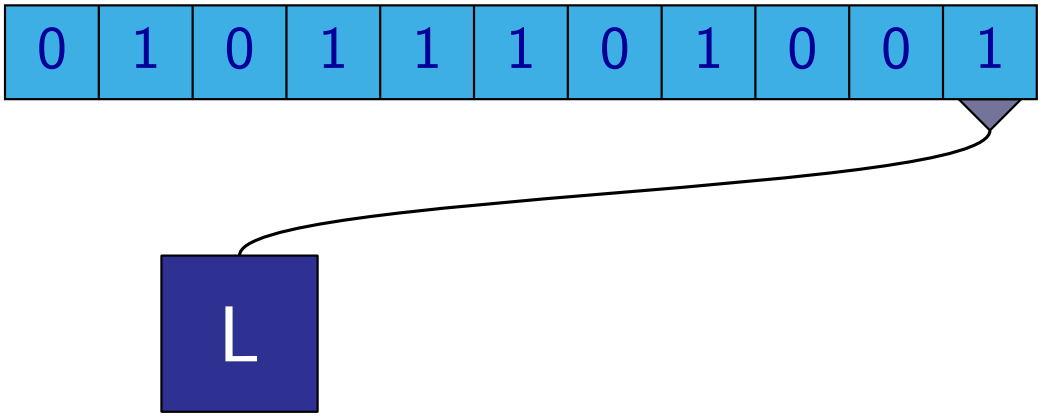




Rozpoznávání jazyka

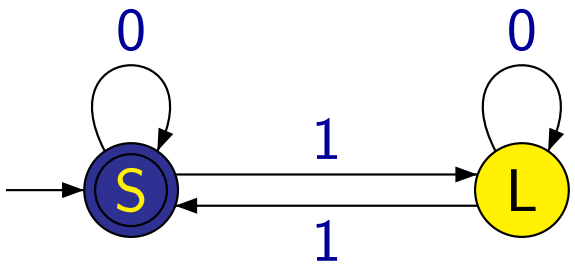
Chovaní tohoto zařízení můžeme popsat grafem:

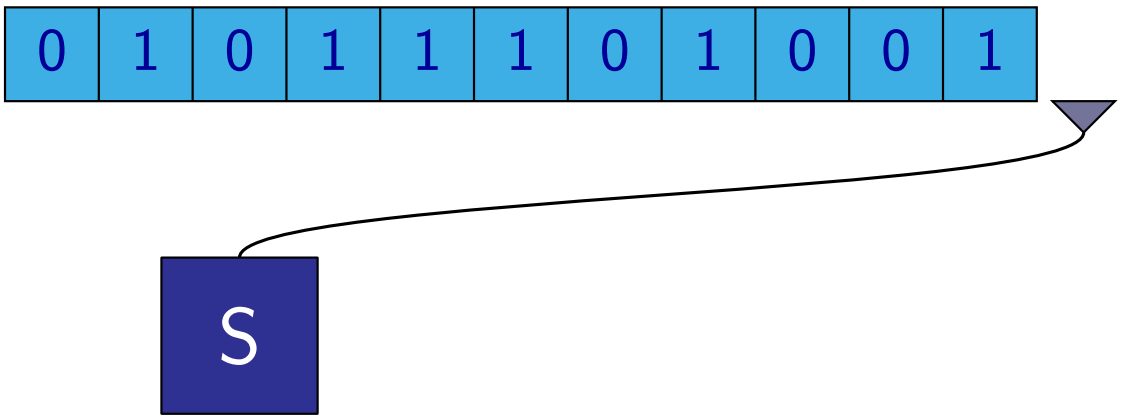




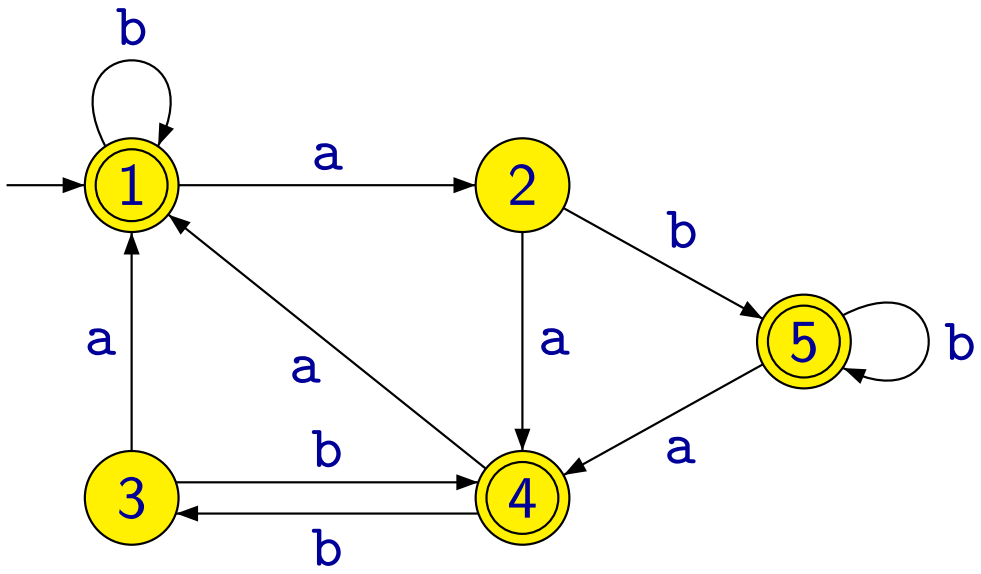
Rozpoznávání jazyka

Chovaní tohoto zařízení můžeme popsat grafem:





Deterministický konečný automat



Deterministický konečný automat se skládá ze **stavů** a **přechodů**. Jeden ze stavů je označen jako **počáteční stav** a některé ze stavů jsou označeny jako **přijímající**.

Deterministický konečný automat

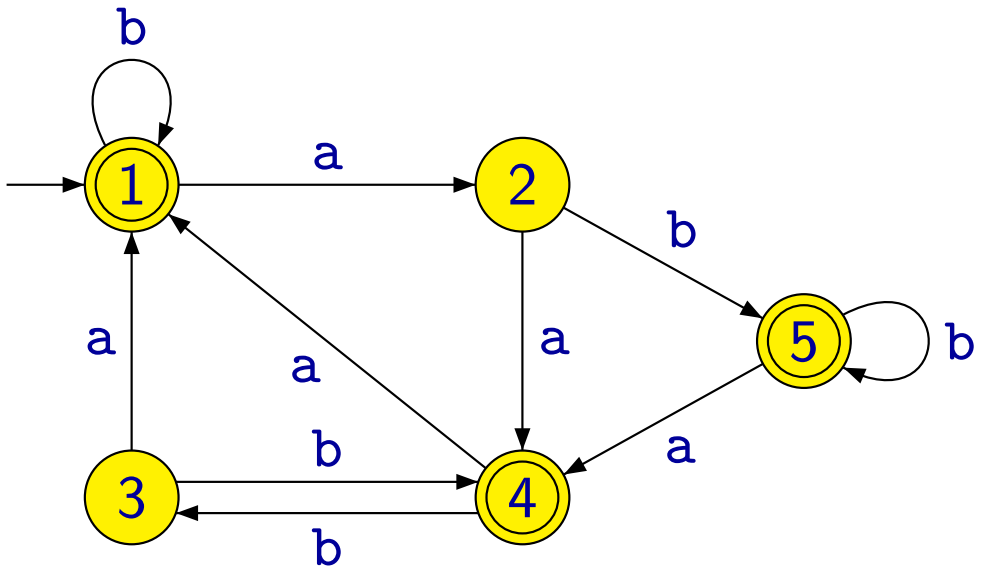
Formálně je **deterministický konečný automat** (**DKA**) definován jako pětice

$$(Q, \Sigma, \delta, q_0, F)$$

kde:

- Q je neprázdná konečná množina **stavů**
- Σ je **abeceda** (neprázdná konečná množina symbolů)
- $\delta : Q \times \Sigma \to Q$ je **přechodová funkce**
- $q_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **přijímajících stavů**

Deterministický konečný automat



- $Q = \{1, 2, 3, 4, 5\}$
- $\Sigma = \{\mathtt{a}, \mathtt{b}\}$
- $q_0 = 1$
- $F = \{1, 4, 5\}$

$$\delta(1, \mathtt{a}) = 2 \qquad \delta(1, \mathtt{b}) = 1$$
$$\delta(2, \mathtt{a}) = 4 \qquad \delta(2, \mathtt{b}) = 5$$
$$\delta(3, \mathtt{a}) = 1 \qquad \delta(3, \mathtt{b}) = 4$$
$$\delta(4, \mathtt{a}) = 1 \qquad \delta(4, \mathtt{b}) = 3$$
$$\delta(5, \mathtt{a}) = 4 \qquad \delta(5, \mathtt{b}) = 5$$

Deterministický konečný automat

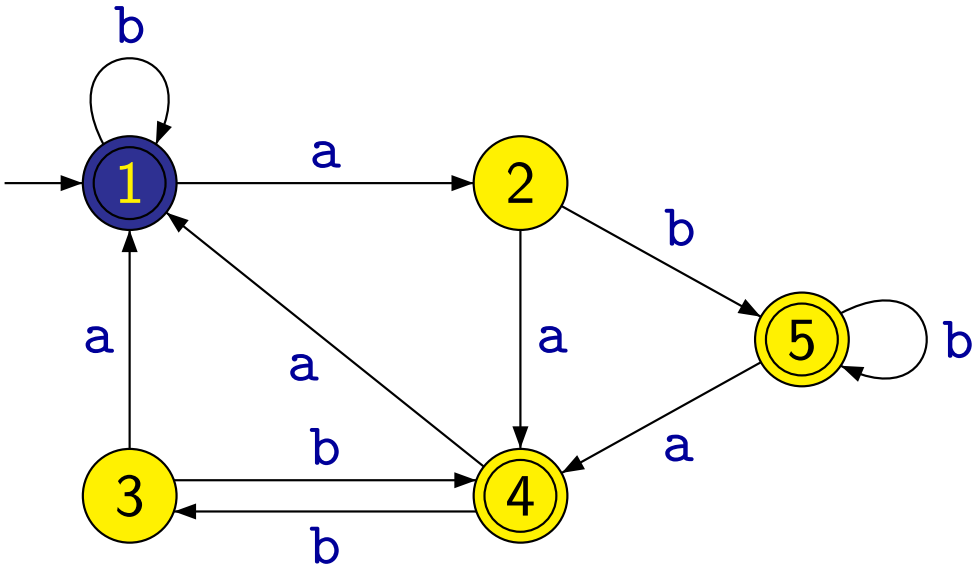
Místo zápisu

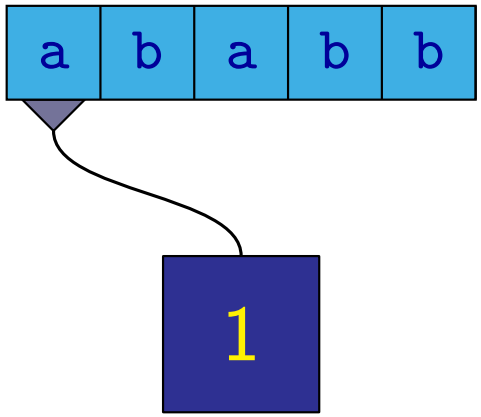
$$\begin{array}{ll} \delta(1,\mathtt{a}) = 2 & \delta(1,\mathtt{b}) = 1 \\ \delta(2,\mathtt{a}) = 4 & \delta(2,\mathtt{b}) = 5 \\ \delta(3,\mathtt{a}) = 1 & \delta(3,\mathtt{b}) = 4 \\ \delta(4,\mathtt{a}) = 1 & \delta(4,\mathtt{b}) = 3 \\ \delta(5,\mathtt{a}) = 4 & \delta(5,\mathtt{b}) = 5 \end{array}$$

budeme raději používat stručnější tabulku nebo grafické znázornění:

δ	a	b
$\leftrightarrow 1$	2	1
2	4	5
3	1	4
$\leftarrow 4$	1	3
$\leftarrow 5$	4	5

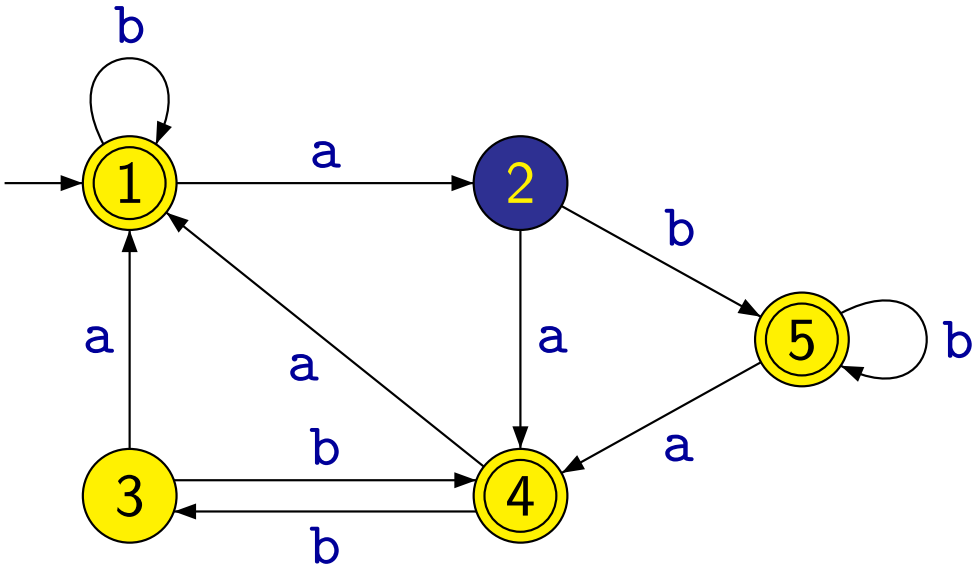
Deterministický konečný automat

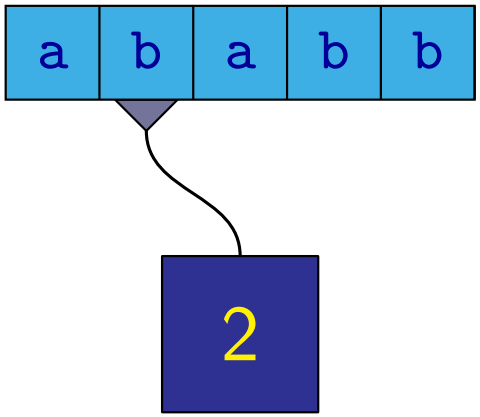




1

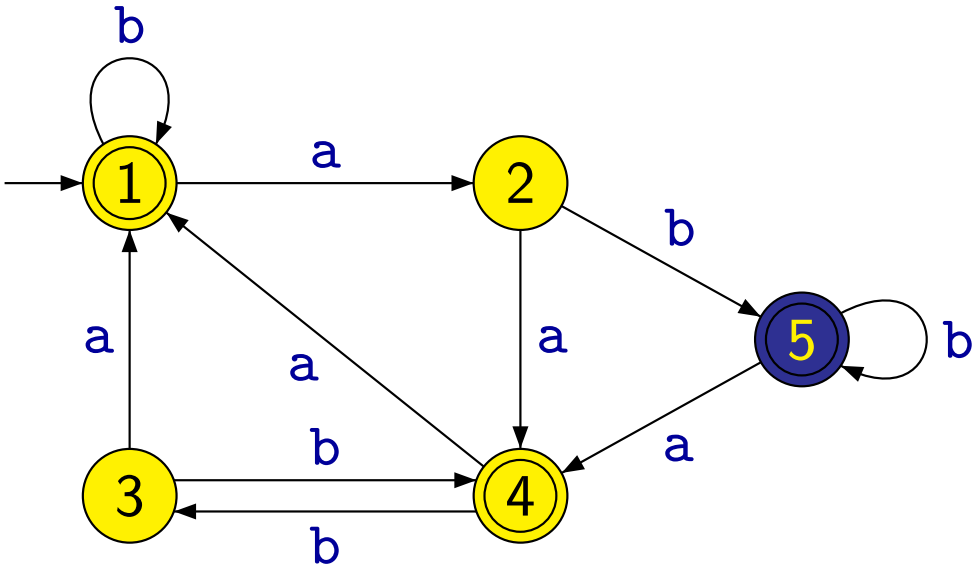
Deterministický konečný automat

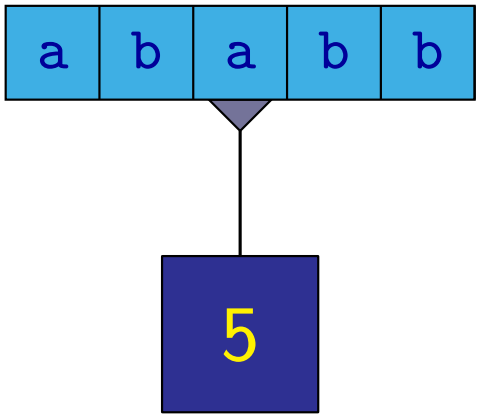




$1 \xrightarrow{a} 2$

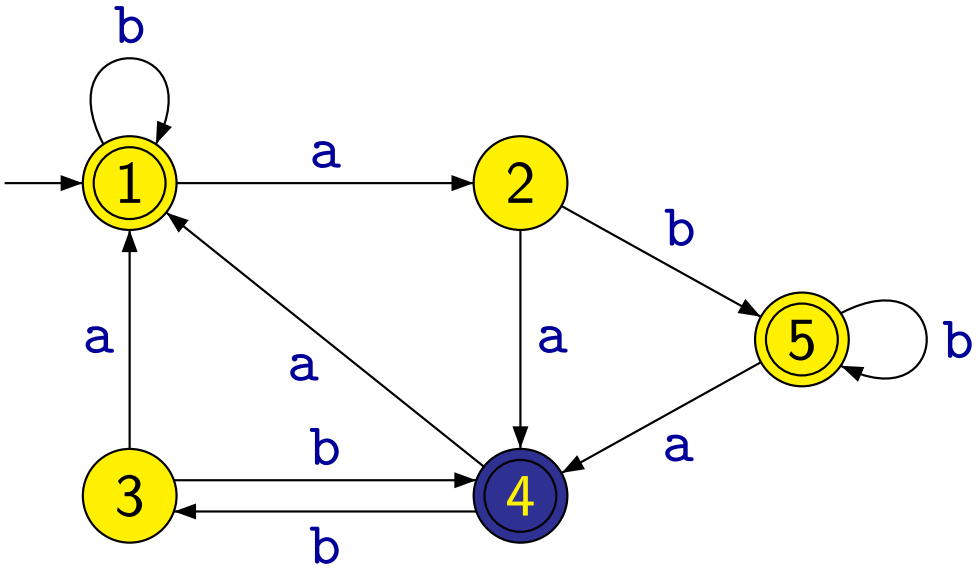
Deterministický konečný automat

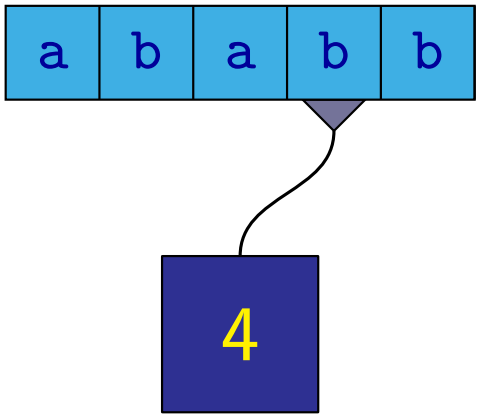




$1 \xrightarrow{a} 2 \xrightarrow{b} 5$

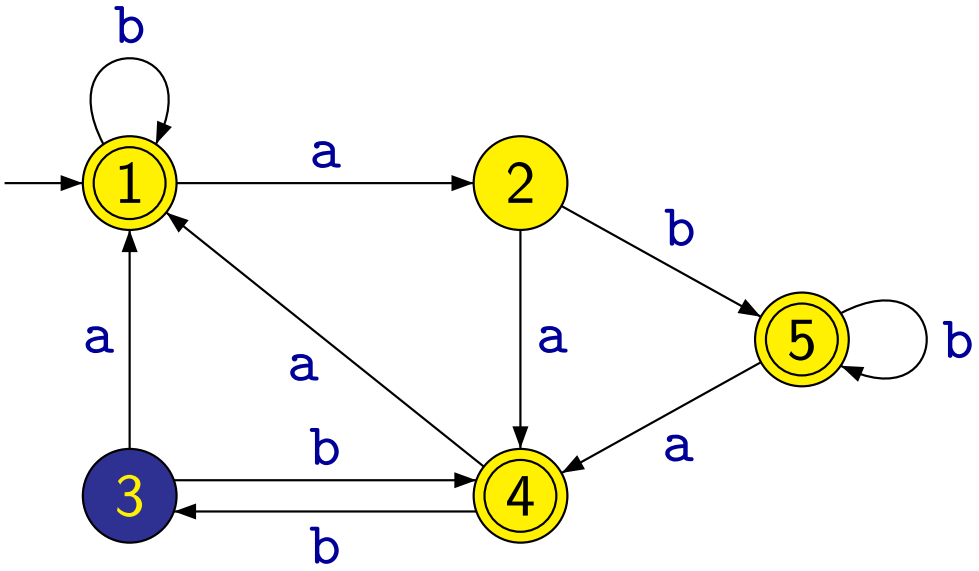
Deterministický konečný automat

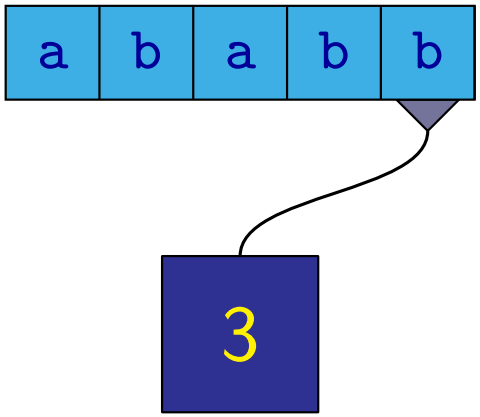




$1 \xrightarrow{a} 2 \xrightarrow{b} 5 \xrightarrow{a} 4$

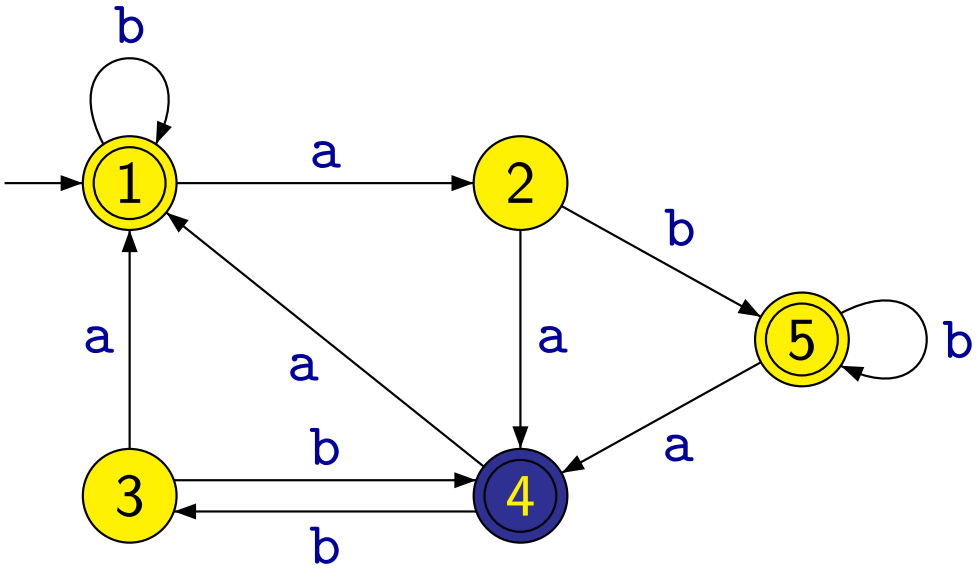
Deterministický konečný automat

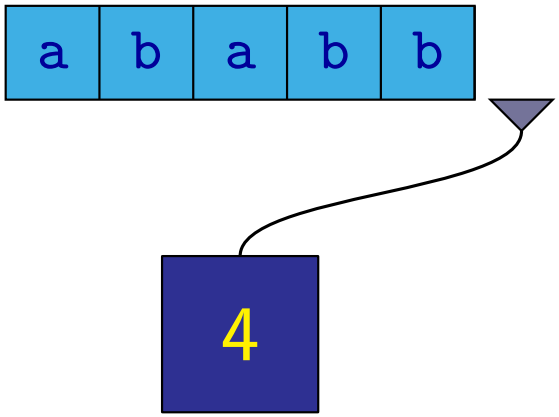




$$1 \xrightarrow{a} 2 \xrightarrow{b} 5 \xrightarrow{a} 4 \xrightarrow{b} 3$$

Deterministický konečný automat





$$1 \xrightarrow{a} 2 \xrightarrow{b} 5 \xrightarrow{a} 4 \xrightarrow{b} 3 \xrightarrow{b} 4$$

Deterministický konečný automat

Definice

Mějme DKA $A = (Q, \Sigma, \delta, q_0, F)$.

Zápisem $q \xrightarrow{w} q'$, kde $q, q' \in Q$ a $w \in \Sigma^*$, budeme označovat to, že pokud je automat ve stavu q, tak přečtením slova w přejde do stavu q'.

Poznámka: $\longrightarrow \subseteq Q \times \Sigma^* \times Q$ je ternární relace.

Místo $(q, w, q') \in \longrightarrow$ píšeme $q \xrightarrow{w} q'$.

Pro DKA platí, že pro libovolný stav q a libovolné slovo w existuje právě jeden stav q' takový, že $q \xrightarrow{w} q'$.

Deterministický konečný automat

Relaci $\longrightarrow$ můžeme formálně definovat následující induktivní definicí:

- $q \xrightarrow{\varepsilon} q$ pro libovolné $q \in Q$
- Pro $a \in \Sigma$ a $w \in \Sigma^*$:
  $q \xrightarrow{aw} q'$ právě tehdy, když existuje $q'' \in Q$ takové, že $\delta(q, a) = q''$ a $q'' \xrightarrow{w} q'$.

Deterministický konečný automat

Slovo $w \in \Sigma^*$ je **přijímáno** deterministickým konečným automatem $A = (Q, \Sigma, \delta, q_0, F)$ právě tehdy, když existuje stav $q \in F$ takový, že $q_0 \xrightarrow{w} q$.

Definice

Jazyk rozpoznávaný (přijímaný) daným deterministickým konečným automatem $A = (Q, \Sigma, \delta, q_0, F)$, označovaný $L(A)$, je množina všech slov přijímaných tímto automatem, tj.

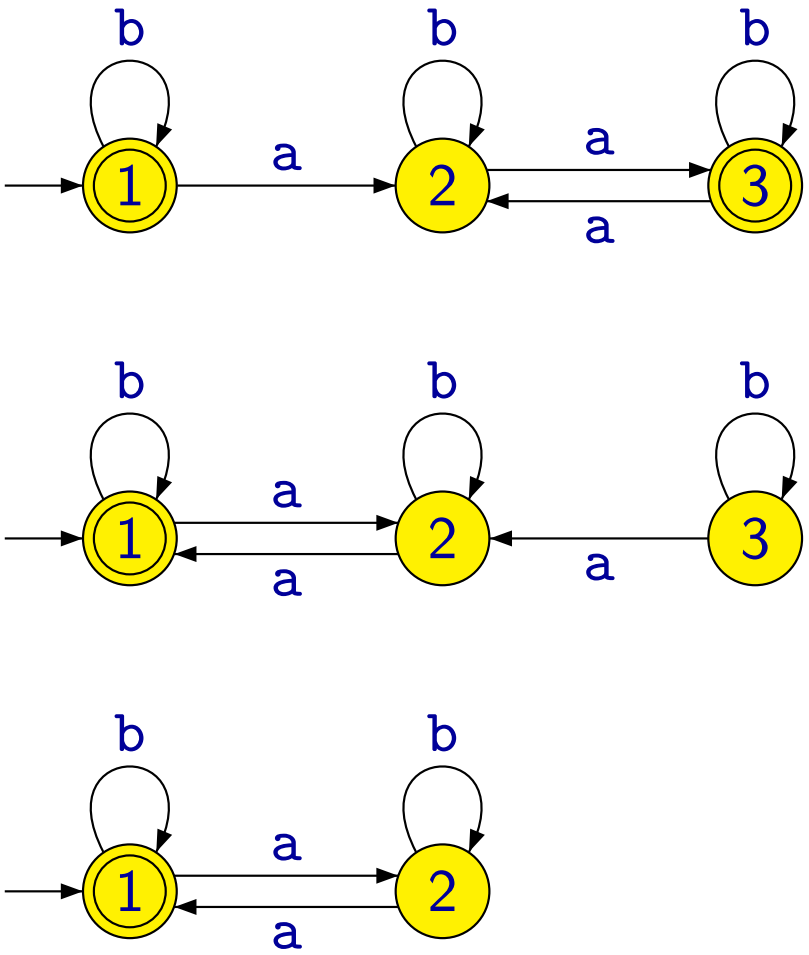
$$L(A) = \{w \in \Sigma^* \mid \exists q \in F : q_0 \xrightarrow{w} q\}$$

Regulární jazyky

Definice

Jazyk L je **regulární** právě tehdy, když existuje nějaký deterministicý konečný automat A, který jej přijímá, tj. DKA A, takový, že $L(A) = L$.

Ekvivalence automatů



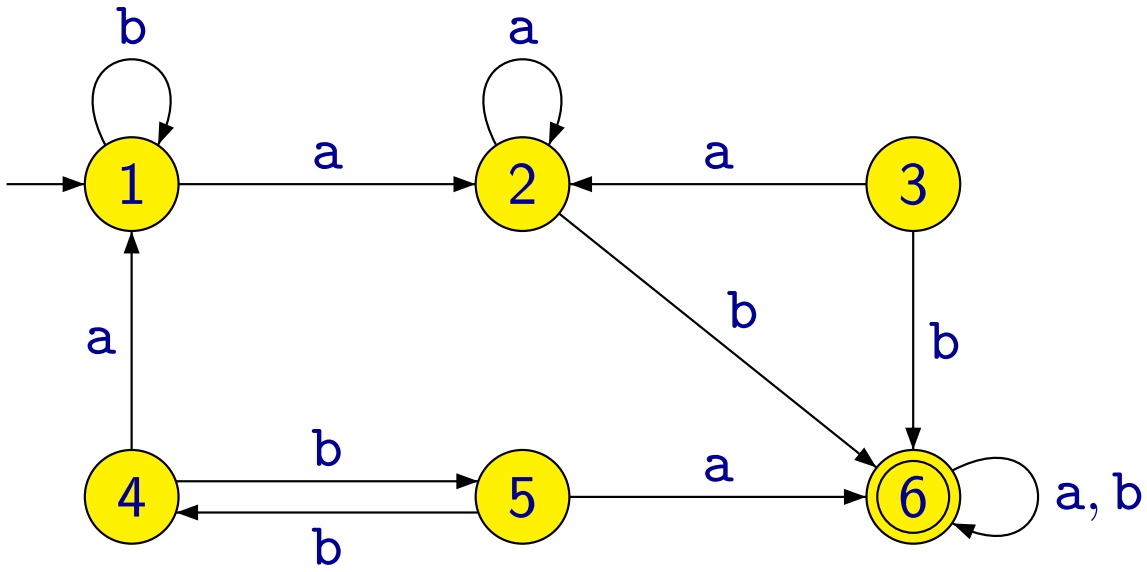
Všechny 3 automaty přijímají jazyk všech slov se sudým počtem *a*.

Ekvivalence automatů

Definice

O konečných automatech A_1, A_2 řekneme, že jsou **ekvivalentní**, jestliže $L(A_1) = L(A_2)$.

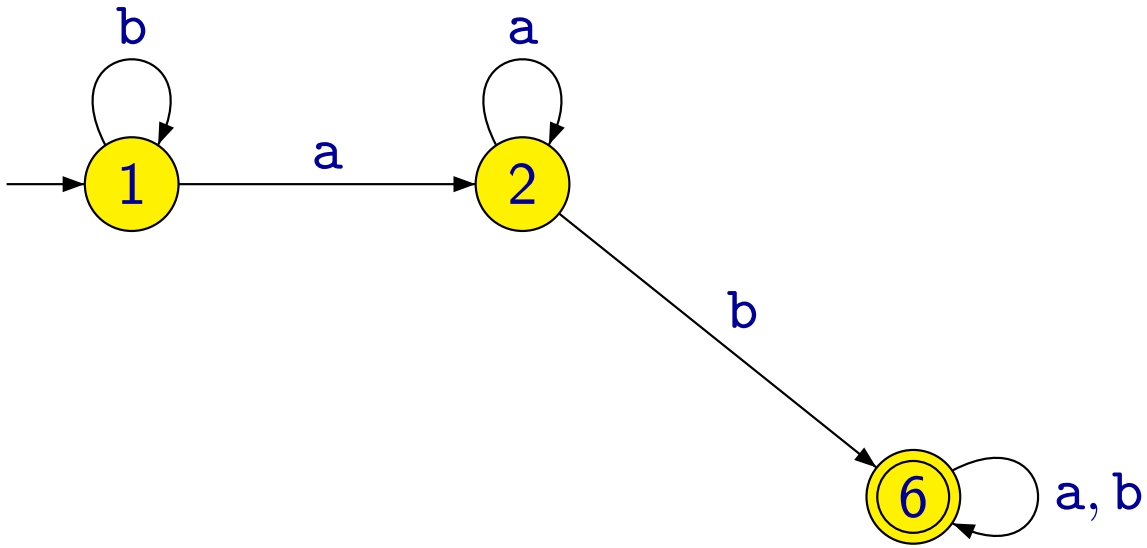
Nedosažitelné stavy automatu



- Automat přijímá jazyk $L = \{w \in \{a, b\}^* \mid w \text{ obsahuje podslovo } ab\}$
- Pro žádnou posloupnost vstupních symbolů se automat nedostane do stavů 3, 4 nebo 5.

.

Nedosažitelné stavy automatu



- Automat přijímá jazyk $L = \{w \in \{a, b\}^* \mid w \text{ obsahuje podslovo } ab\}$
- Pro žádnou posloupnost vstupních symbolů se automat nedostane do stavů 3, 4 nebo 5.
- Pokud tyto stavy odstraníme, pořád automat přijímá stejný jazyk L.

Nedosažitelné stavy automatu

Definice

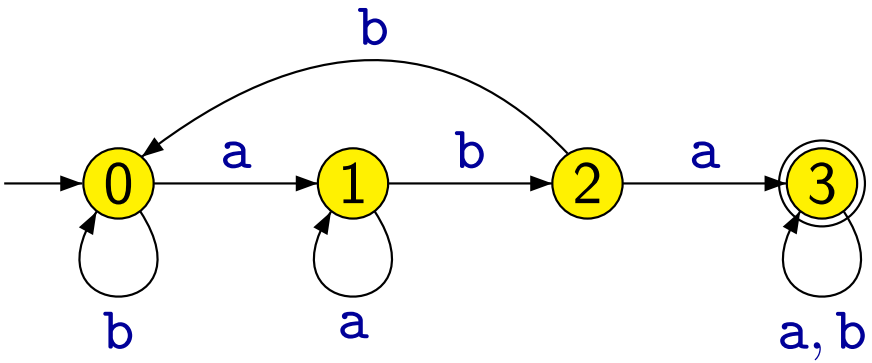
Stav q konečného automatu $A = (Q, \Sigma, \delta, q_0, F)$ je **dosažitelný** pokud existuje nějaké slovo w takové, že $q_0 \xrightarrow{w} q$.

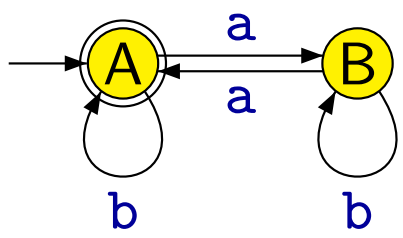
V opačném případě stav nazýváme **nedosažitelný**.

- Do nedosažitelných stavů nevede v grafu automatu žádná orientovaná cesta z počátečního stavu.
- Nedosažitelné stavy můžeme z automatu odstranit (spolu se všemi přechody vedoucími do nich a z nich). Jazyk přijímaný automatem se nezmění.

Automat pro průnik jazyků

Máme následující dva automaty:

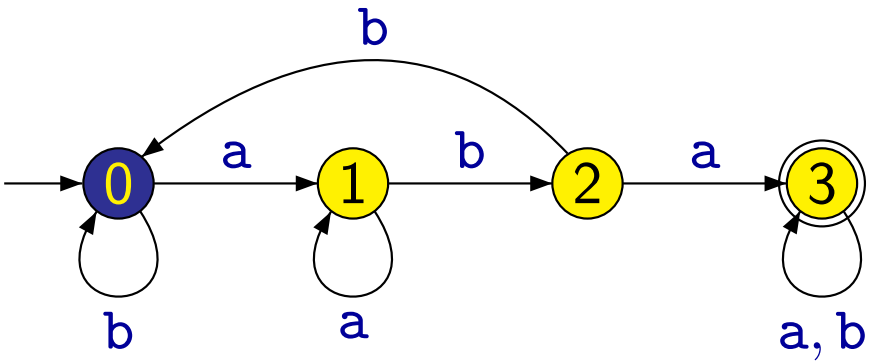


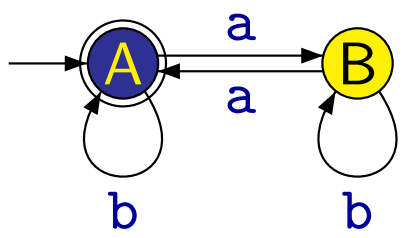


Příjmou oba slovo ababb?

Automat pro průnik jazyků

Máme následující dva automaty:

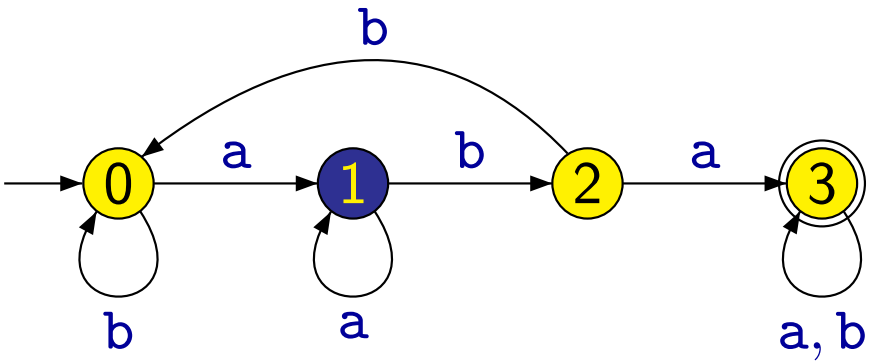


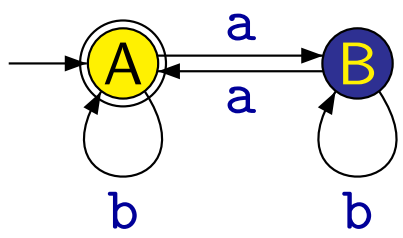


Příjmou oba slovo ababb?

Automat pro průnik jazyků

Máme následující dva automaty:

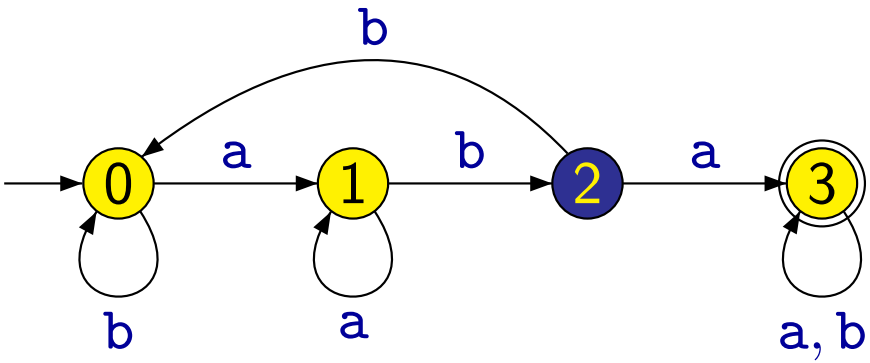


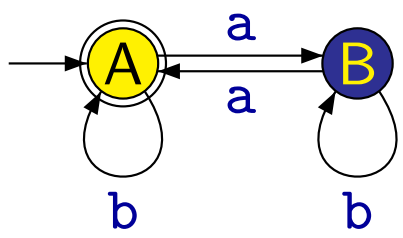


Příjmou oba slovo ababb?

Automat pro průnik jazyků

Máme následující dva automaty:

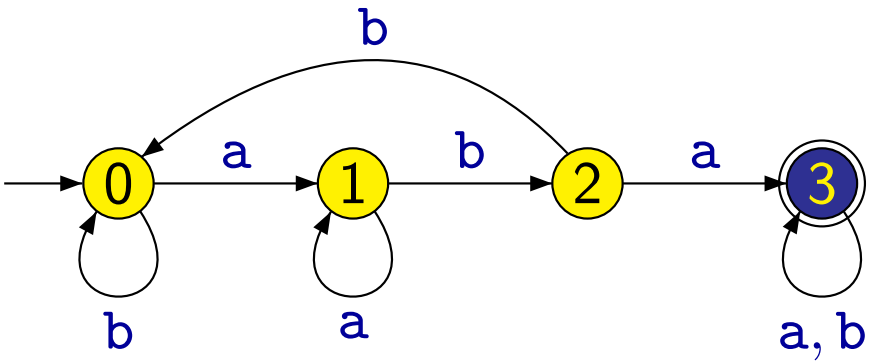


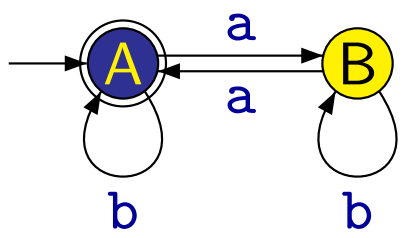


Příjmou oba slovo ababb?

Automat pro průnik jazyků

Máme následující dva automaty:

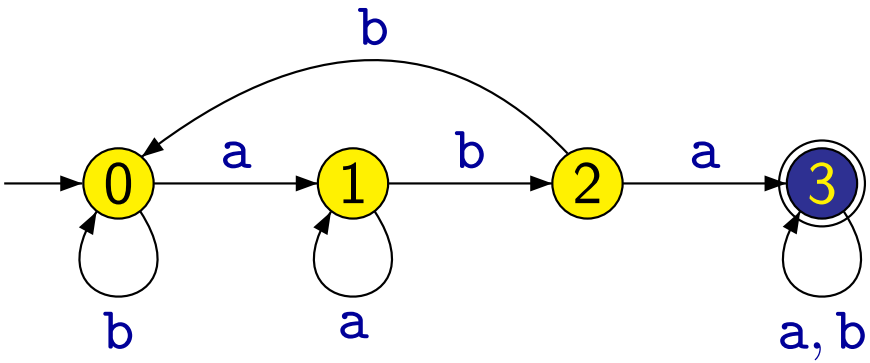


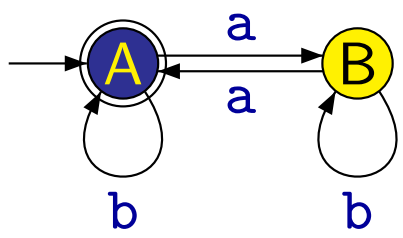


Příjmou oba slovo ababb?

Automat pro průnik jazyků

Máme následující dva automaty:

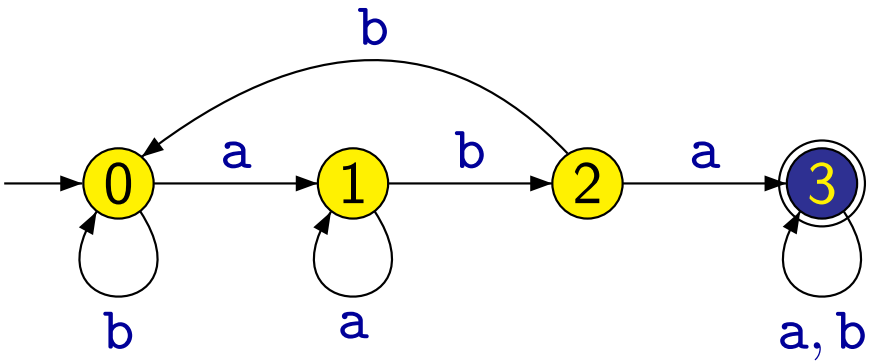


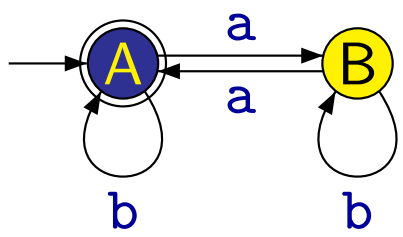


Příjmou oba slovo ababb?

Automat pro průnik jazyků

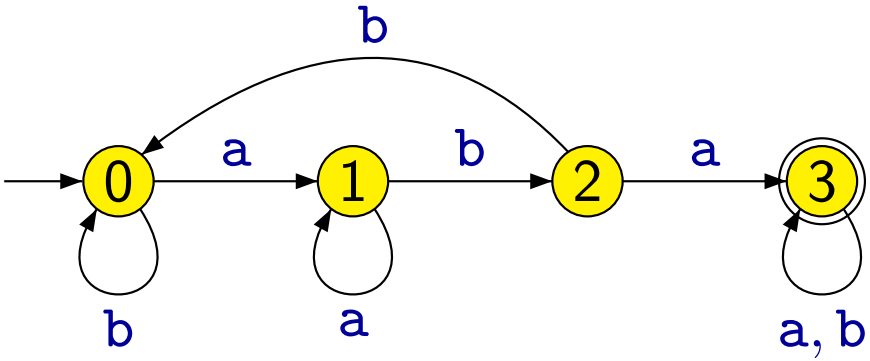
Máme následující dva automaty:

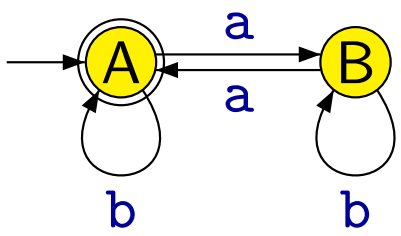




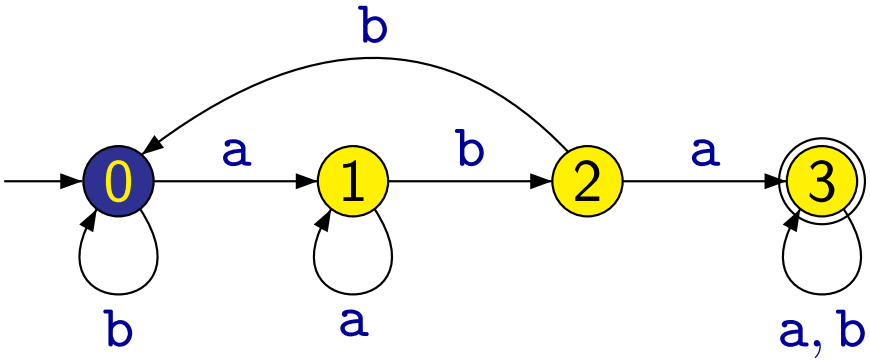
Příjmou oba slovo ababb?

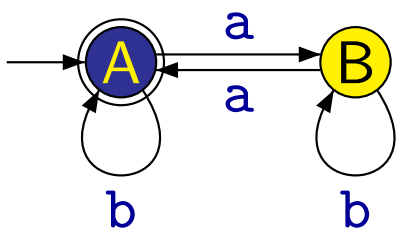
Automat pro průnik jazyků

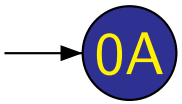




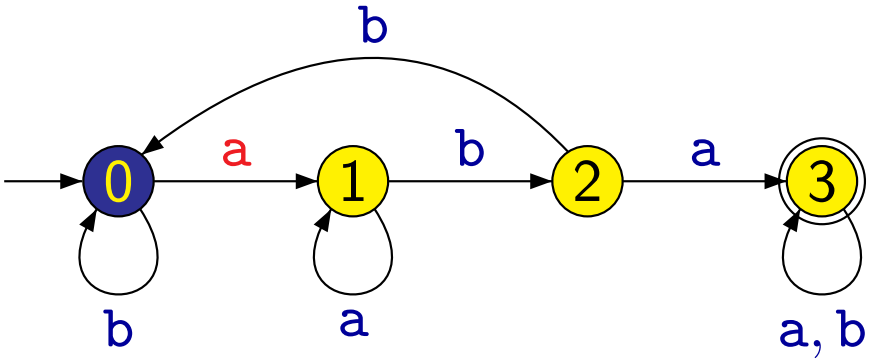
Automat pro průnik jazyků

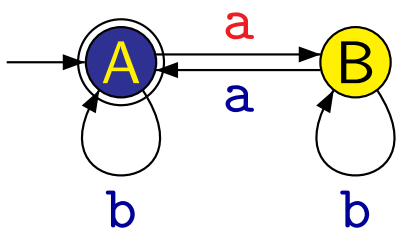




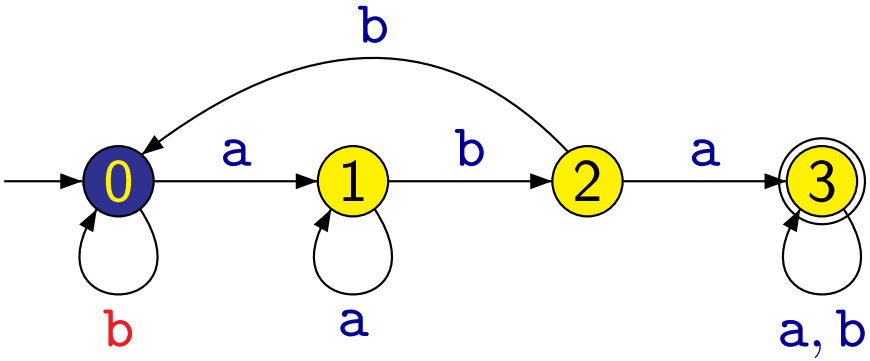


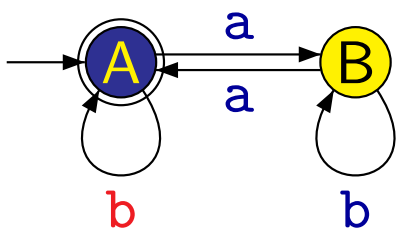
Automat pro průnik jazyků

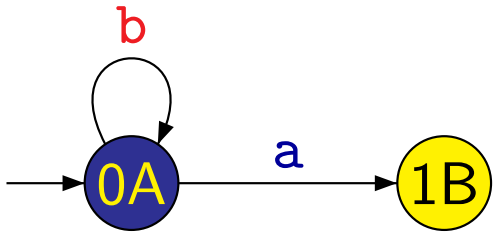




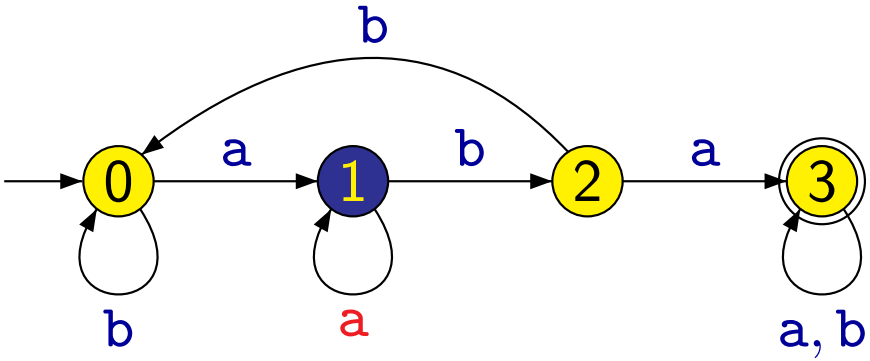
Automat pro průnik jazyků

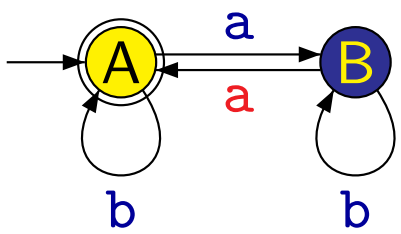


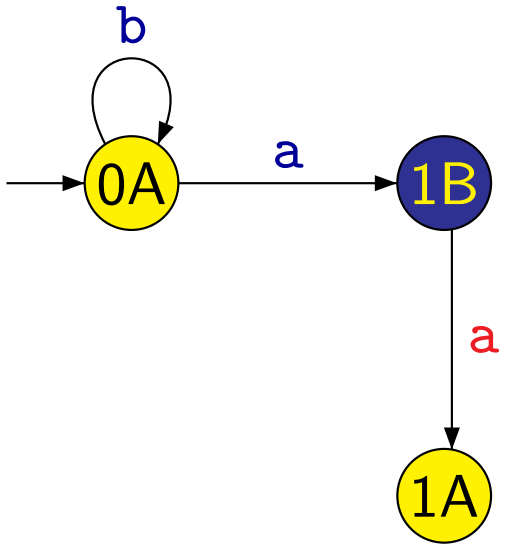




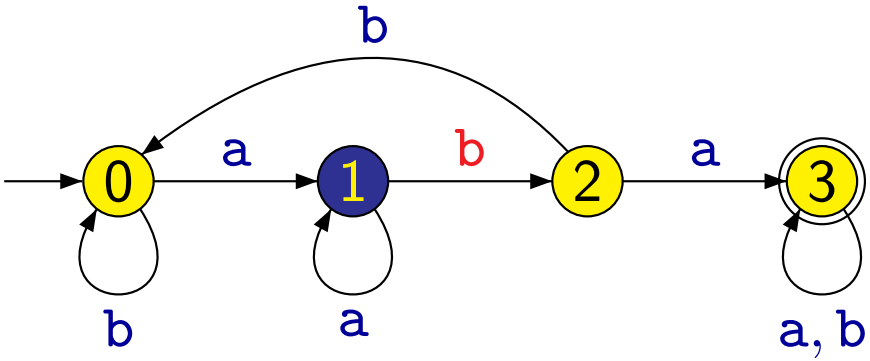
Automat pro průnik jazyků

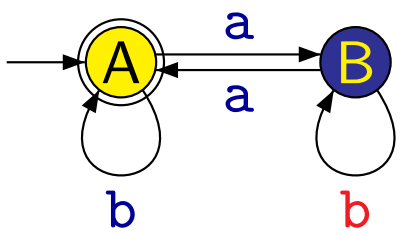


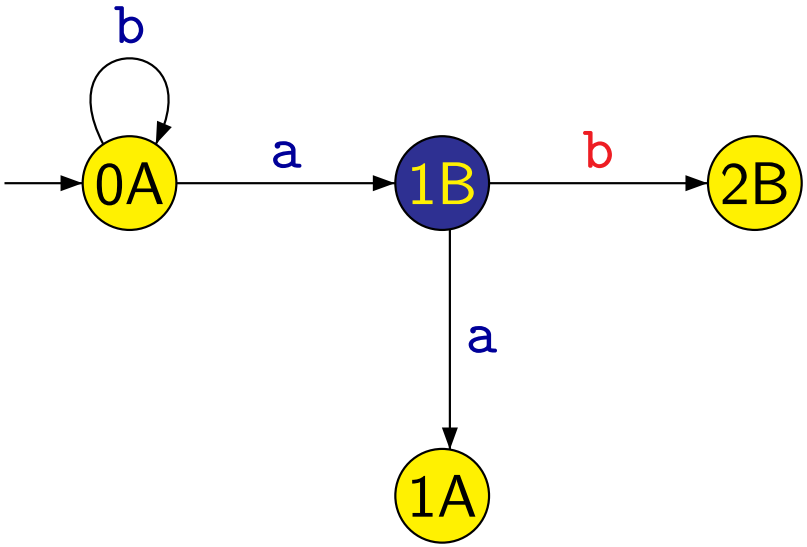




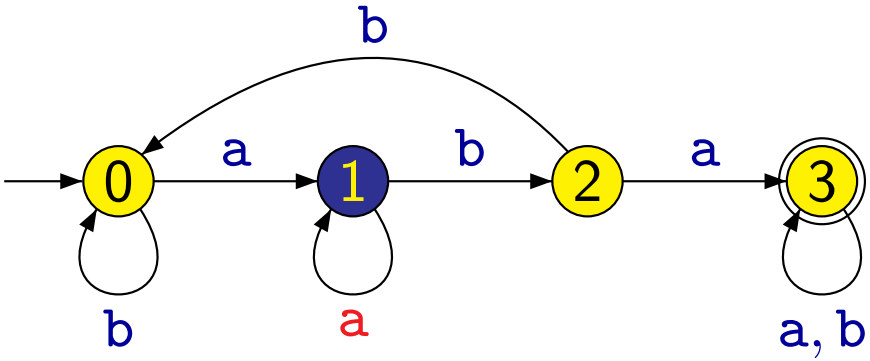
Automat pro průnik jazyků

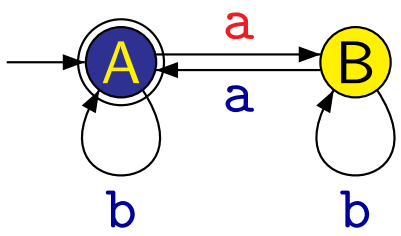


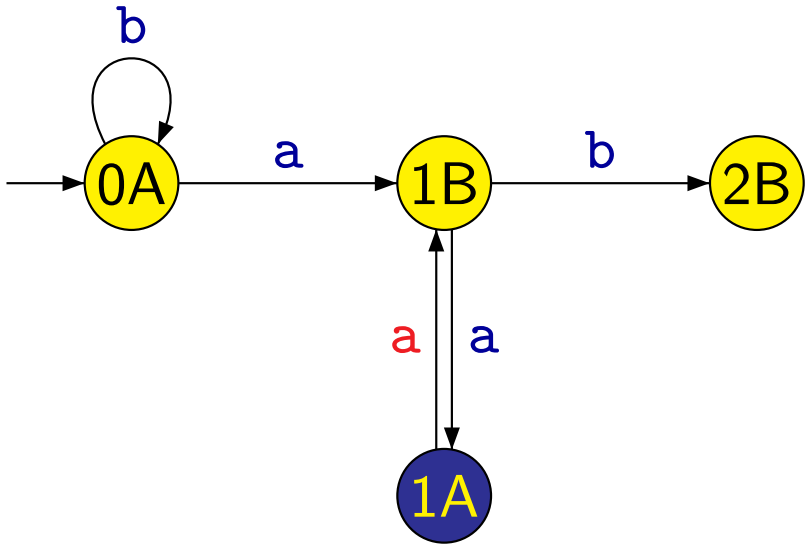




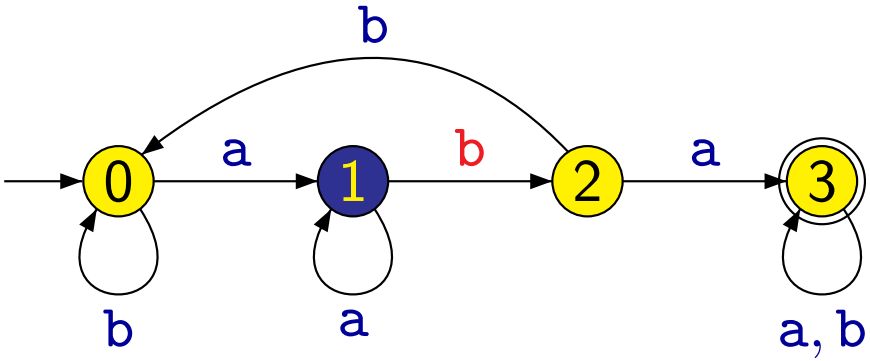
Automat pro průnik jazyků

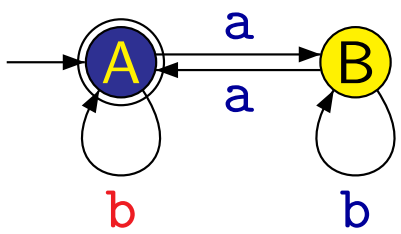


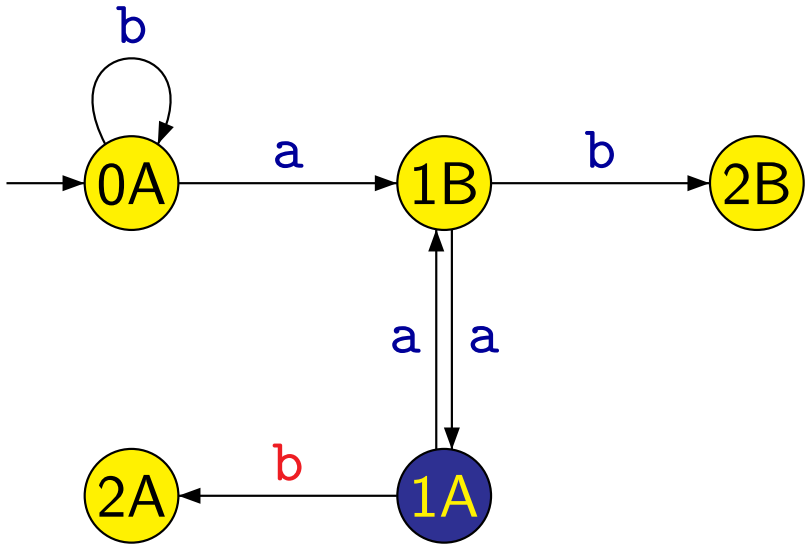




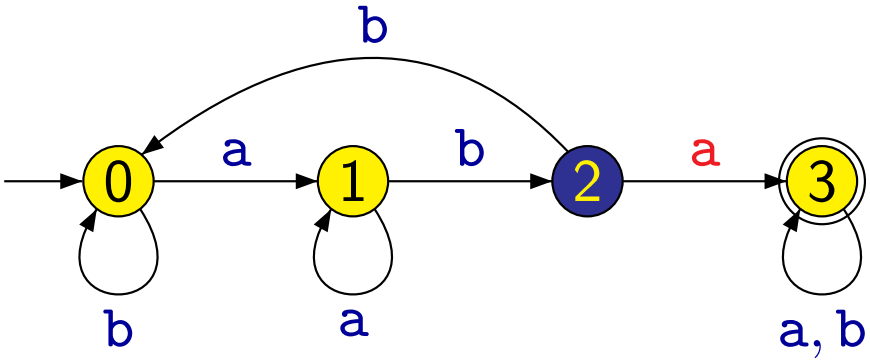
Automat pro průnik jazyků

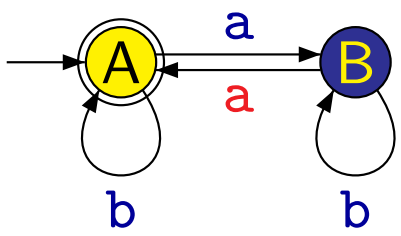


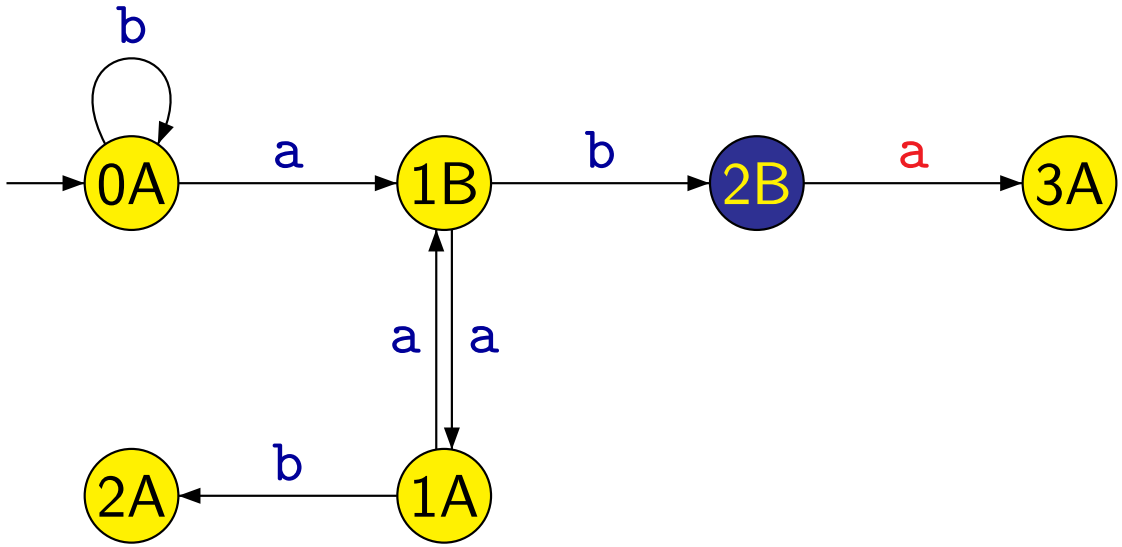




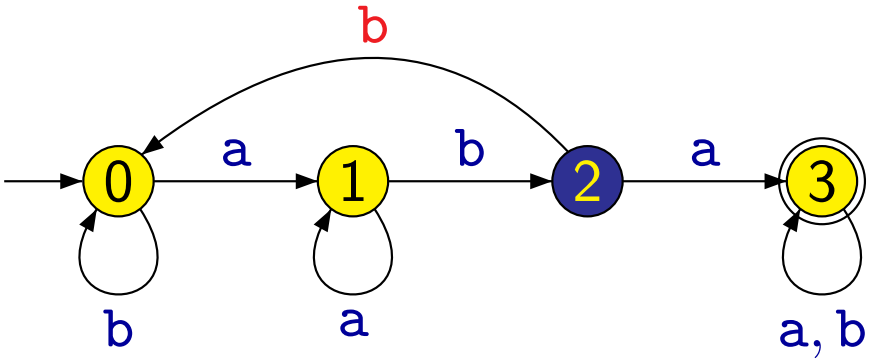
Automat pro průnik jazyků

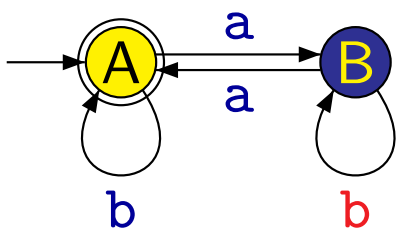


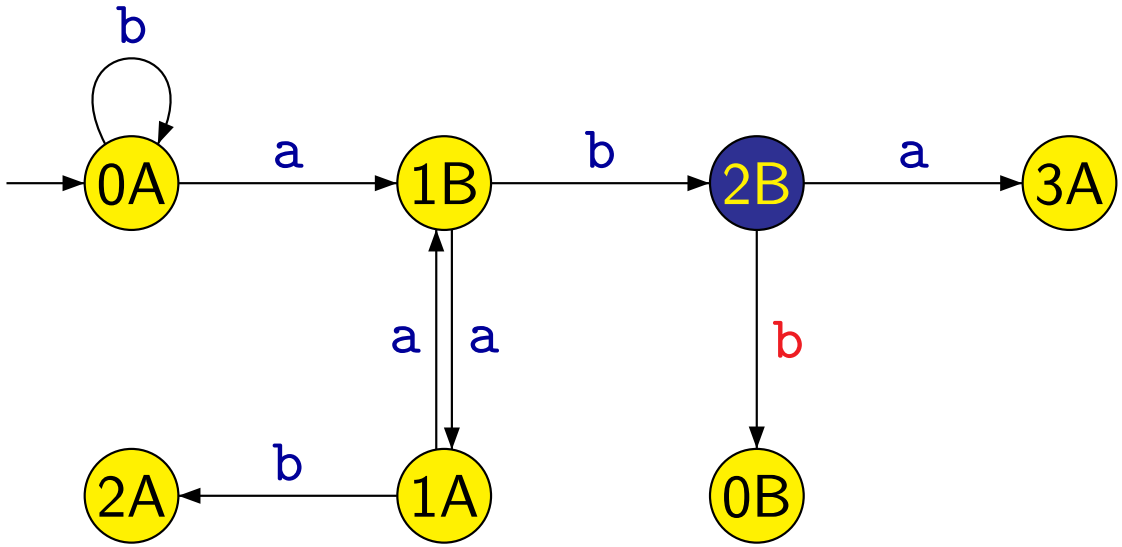




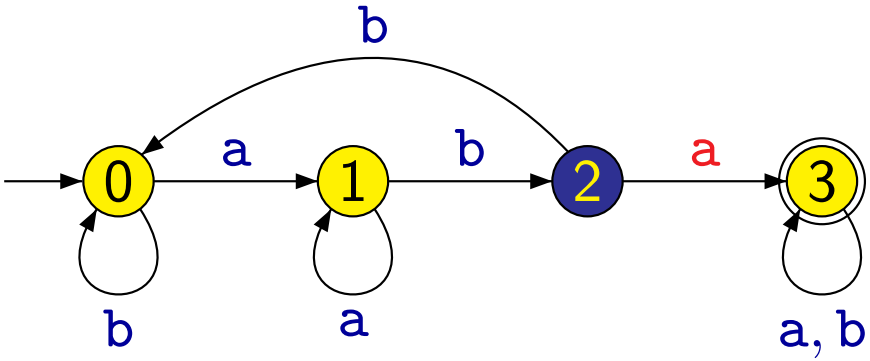
Automat pro průnik jazyků

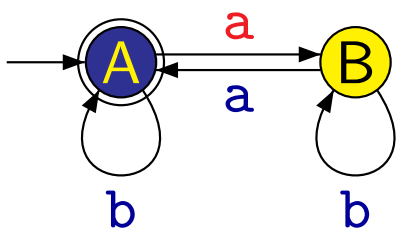


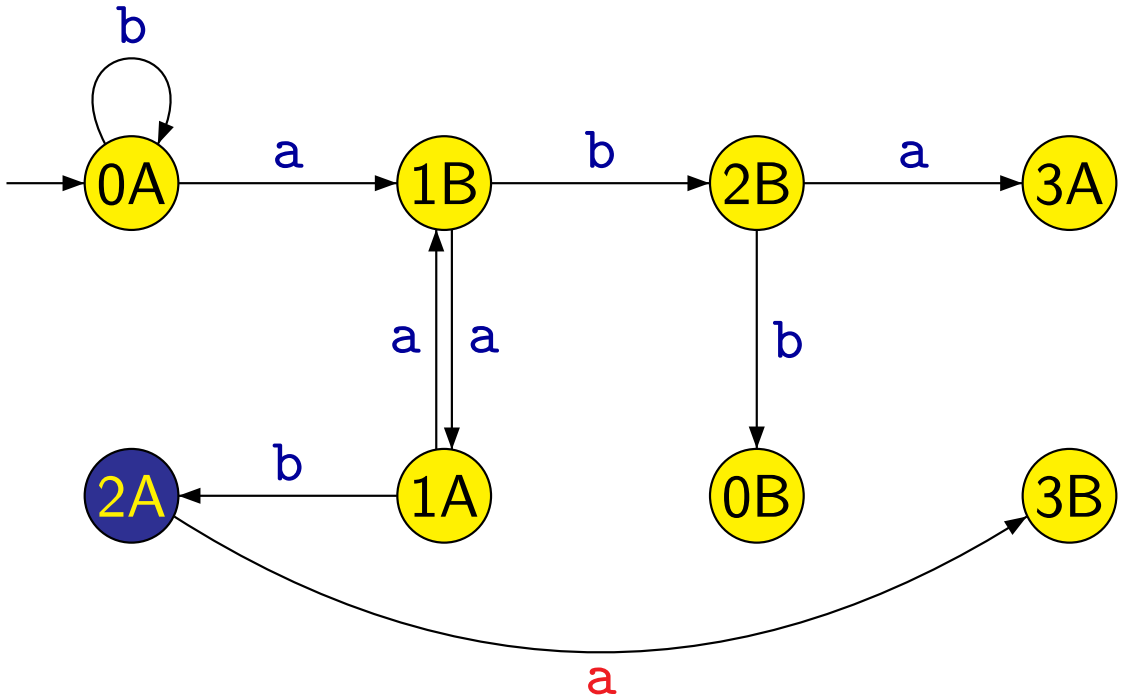




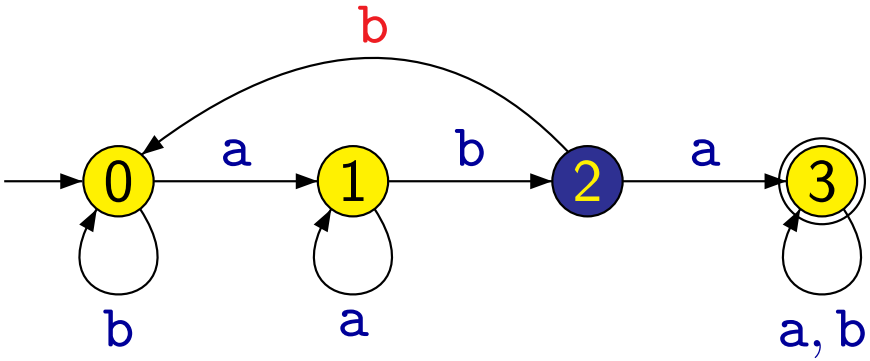
Automat pro průnik jazyků

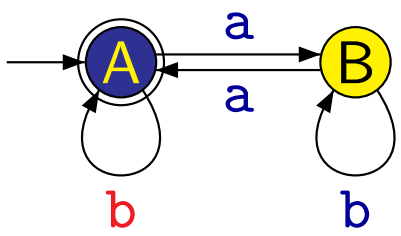


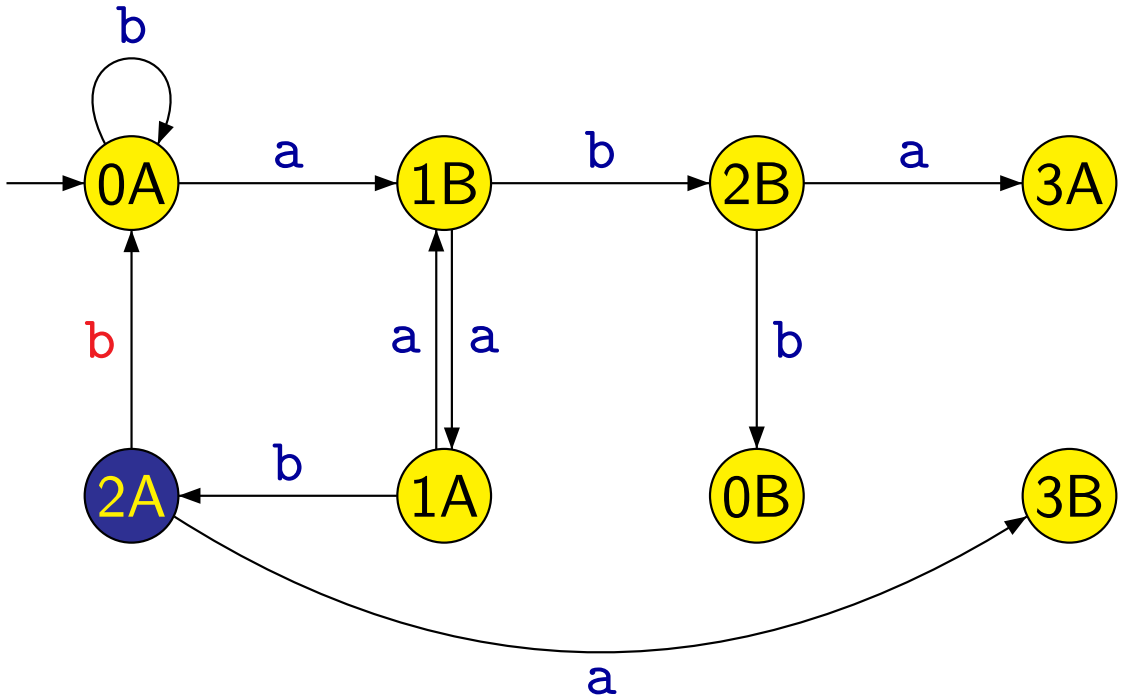




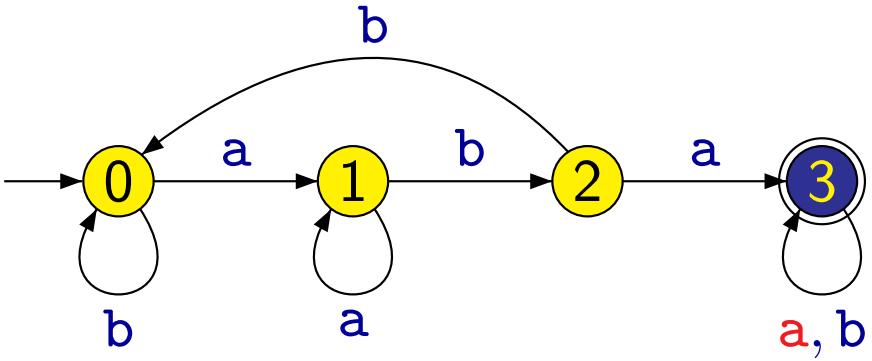
Automat pro průnik jazyků

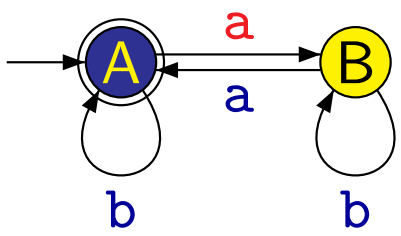


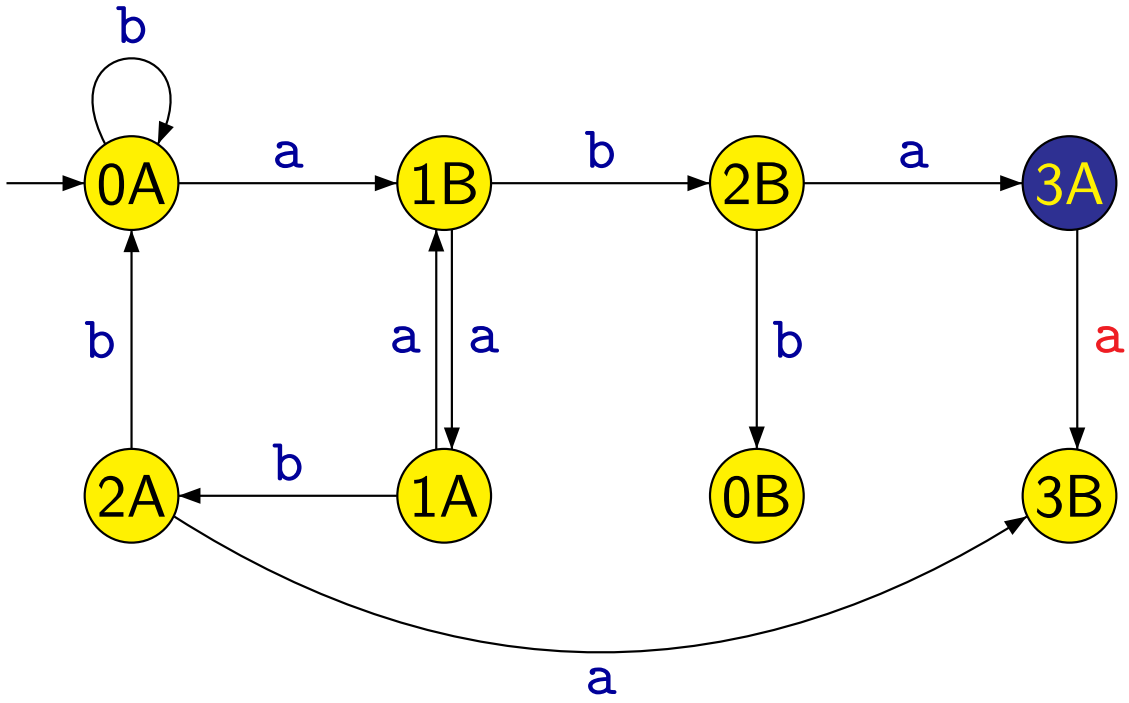




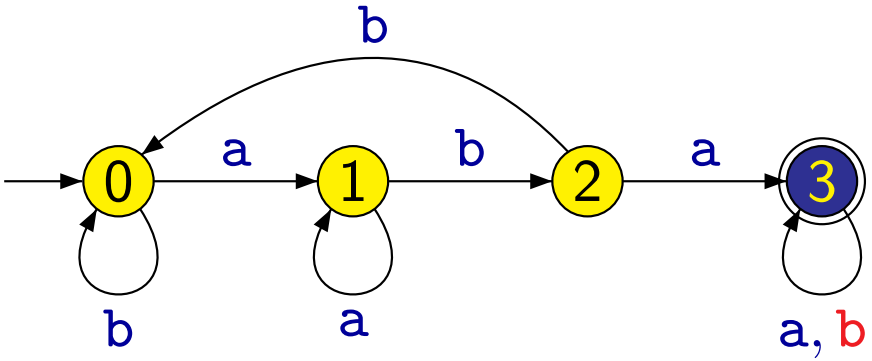
Automat pro průnik jazyků

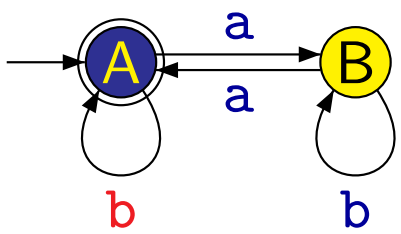


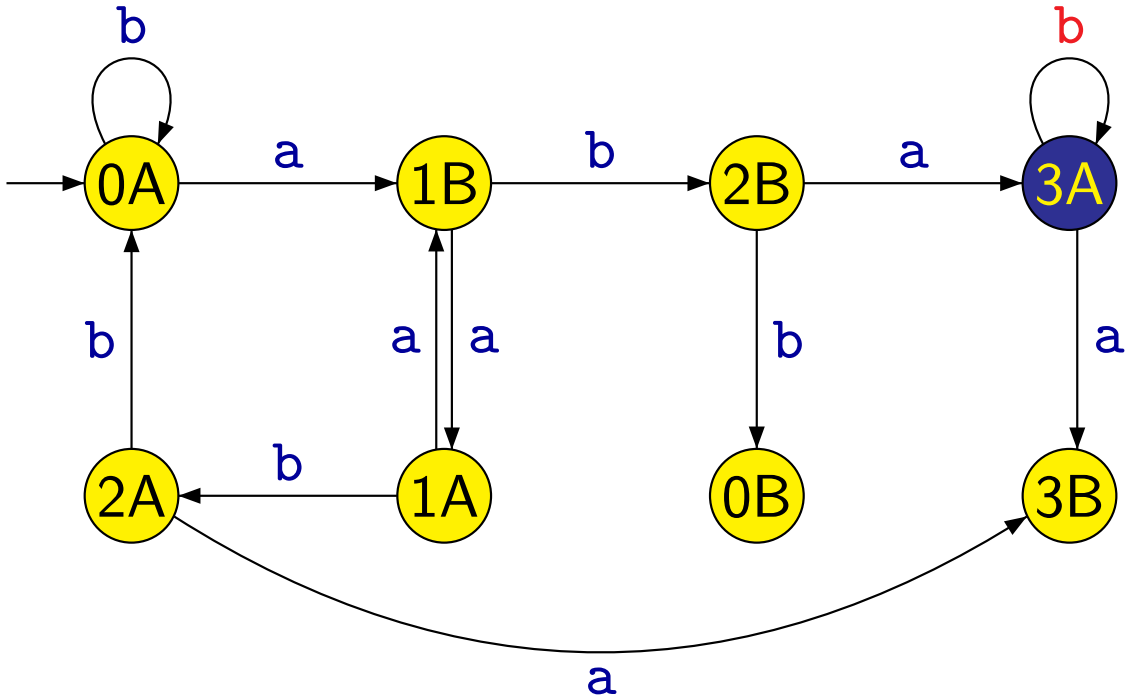




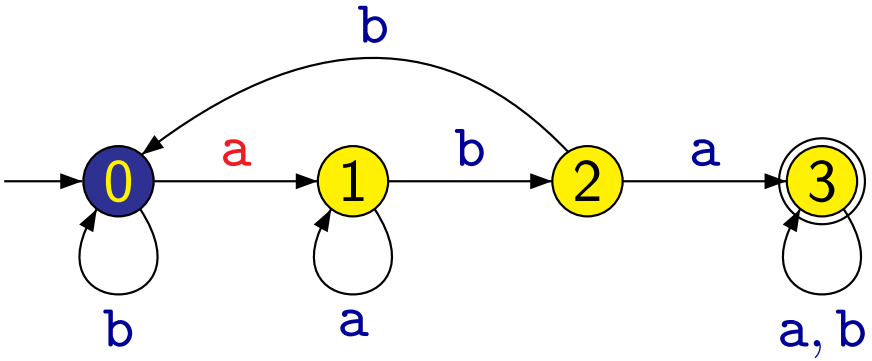
Automat pro průnik jazyků

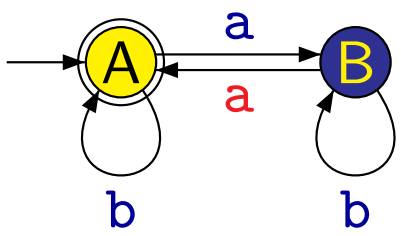


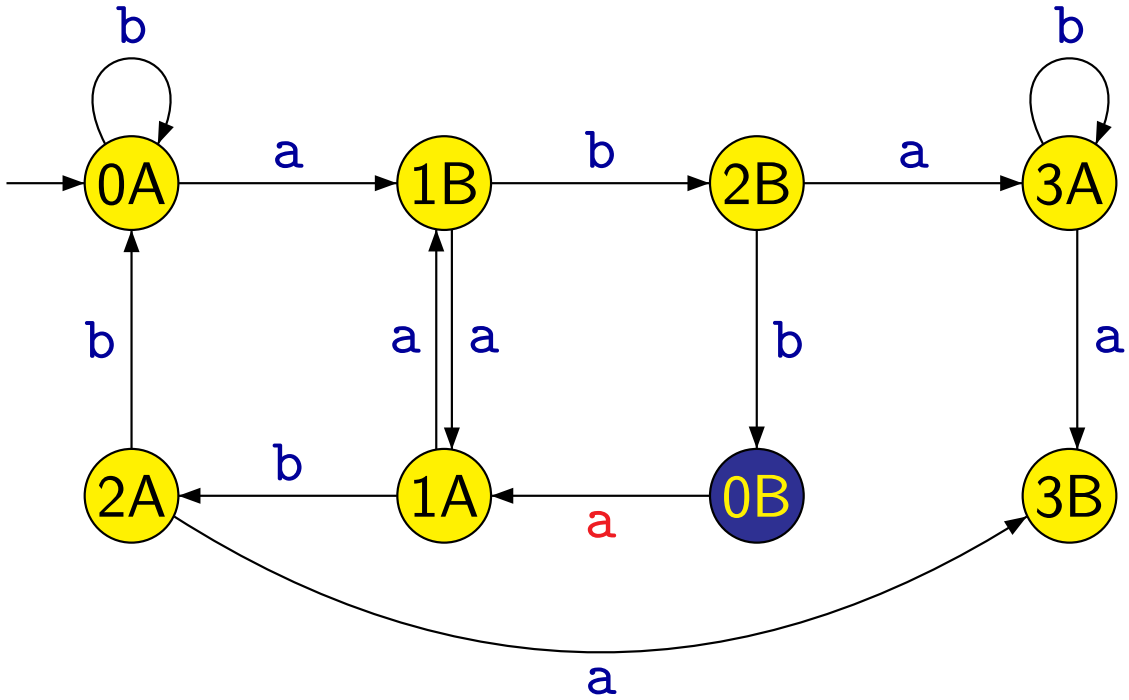




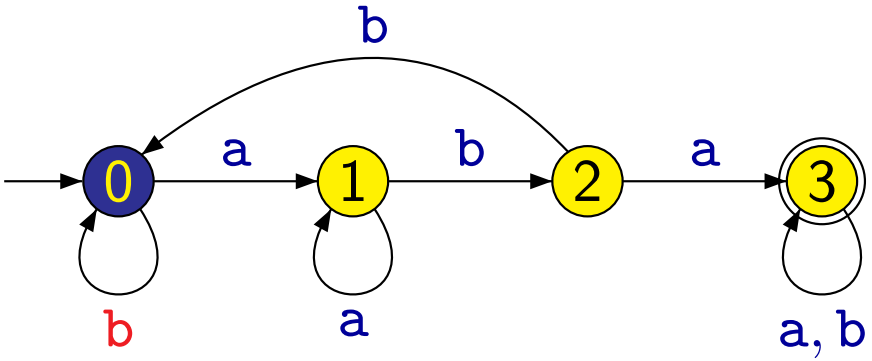
Automat pro průnik jazyků

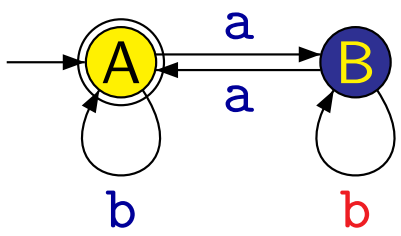


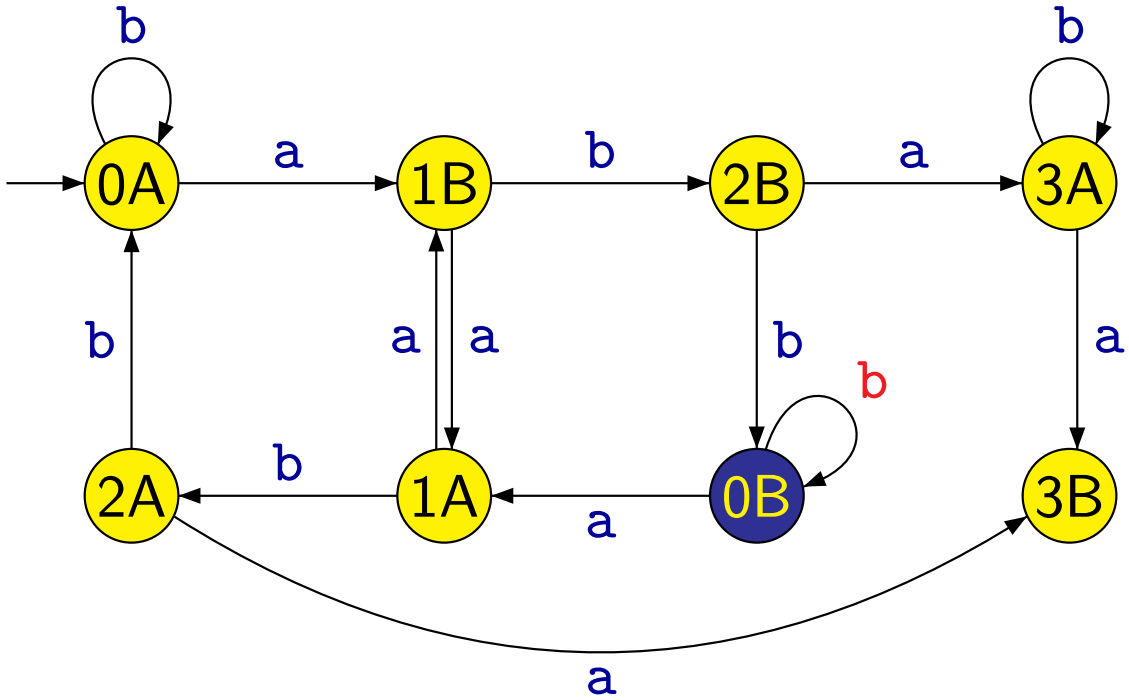




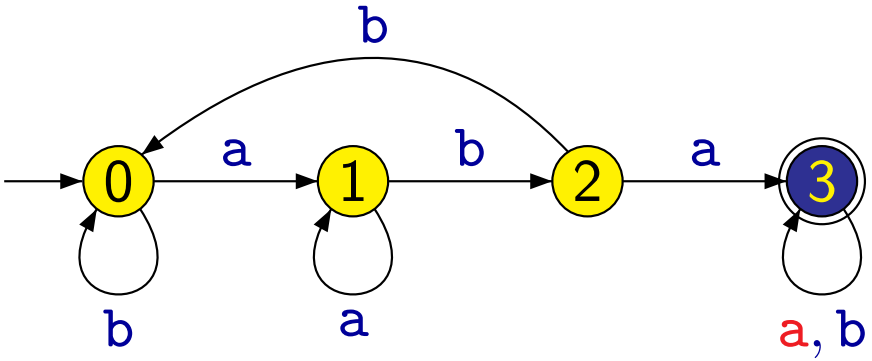
Automat pro průnik jazyků

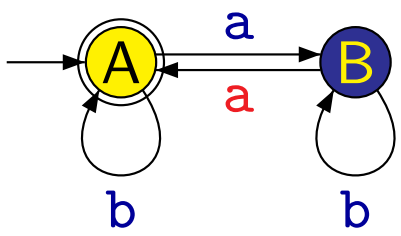


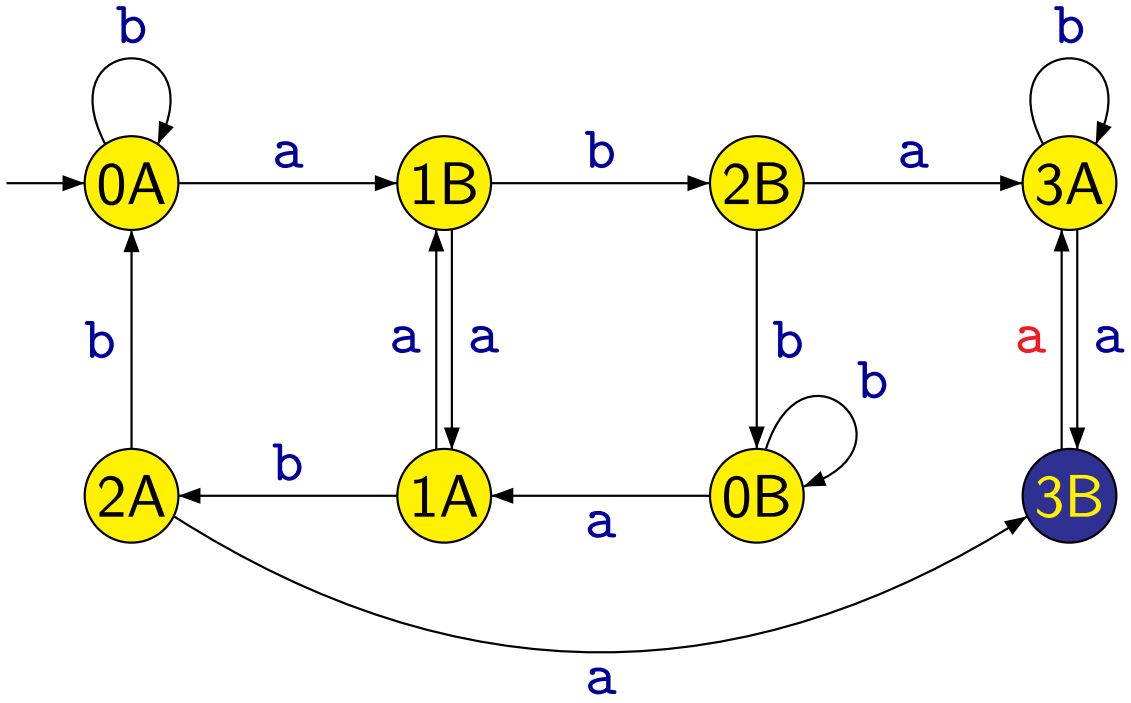




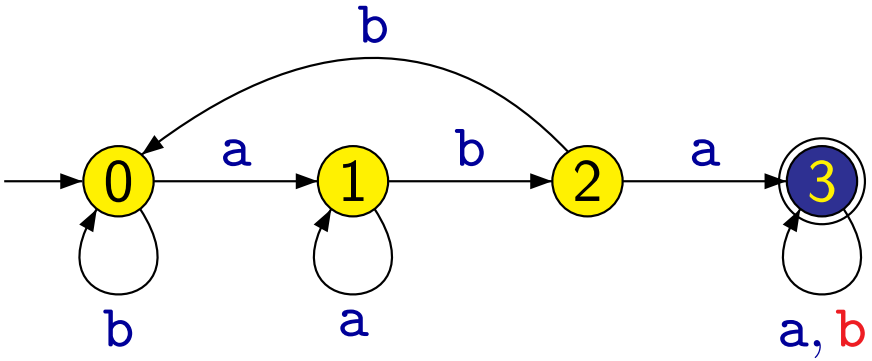
Automat pro průnik jazyků

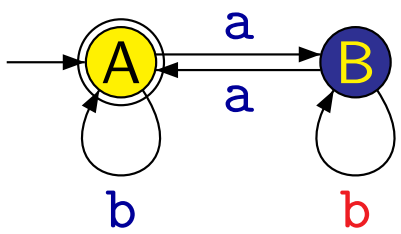


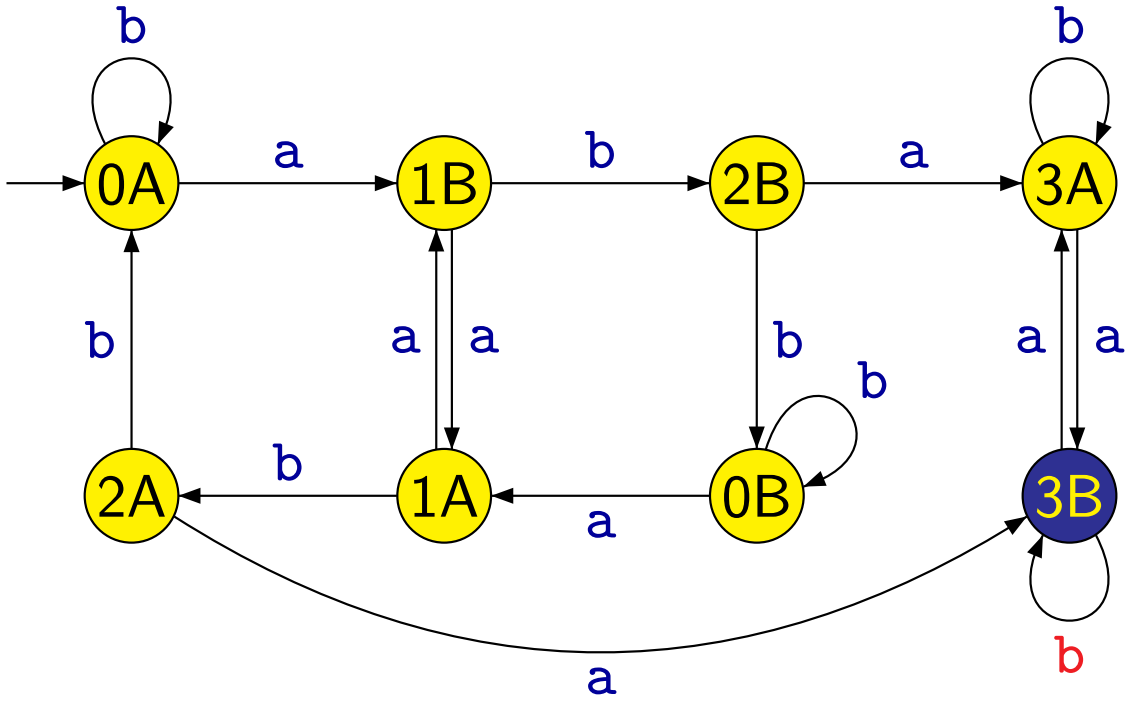




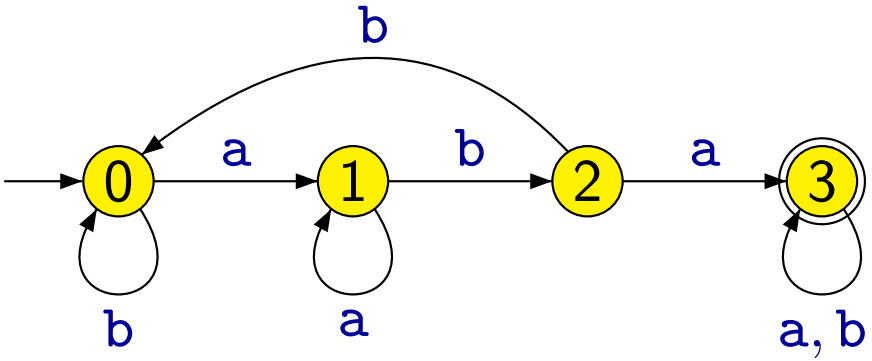
Automat pro průnik jazyků

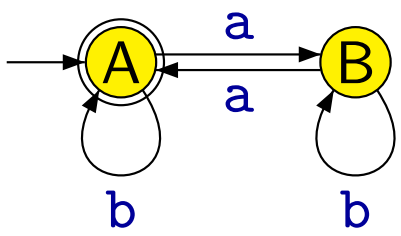


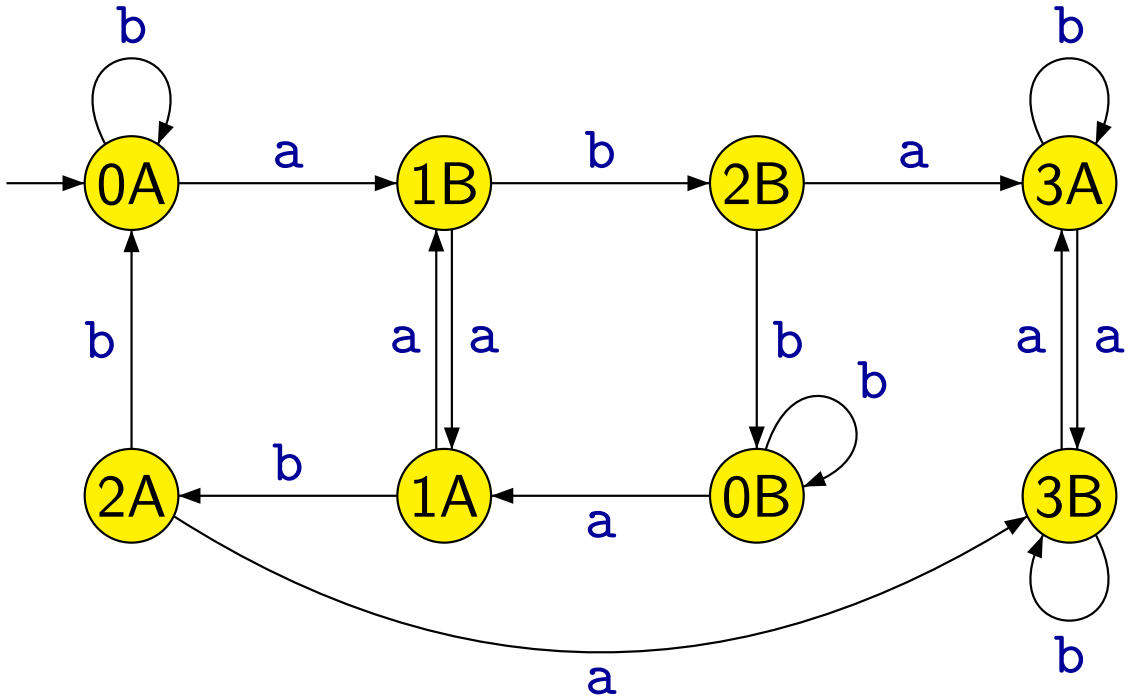




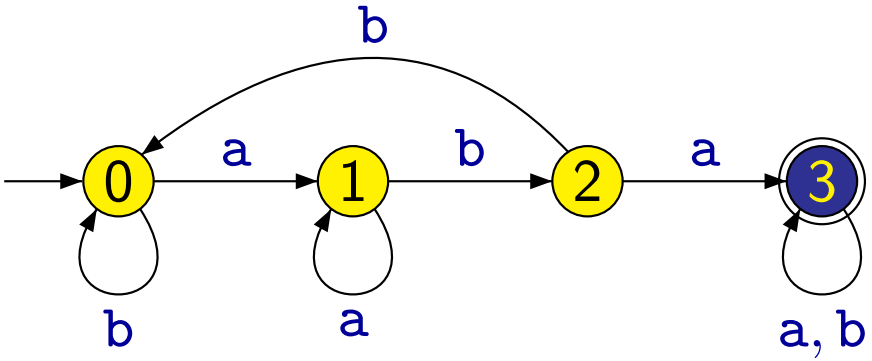
Automat pro průnik jazyků

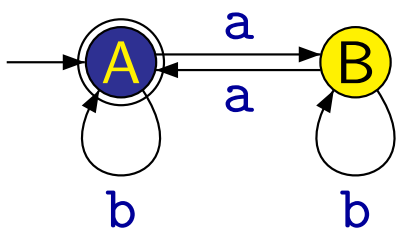


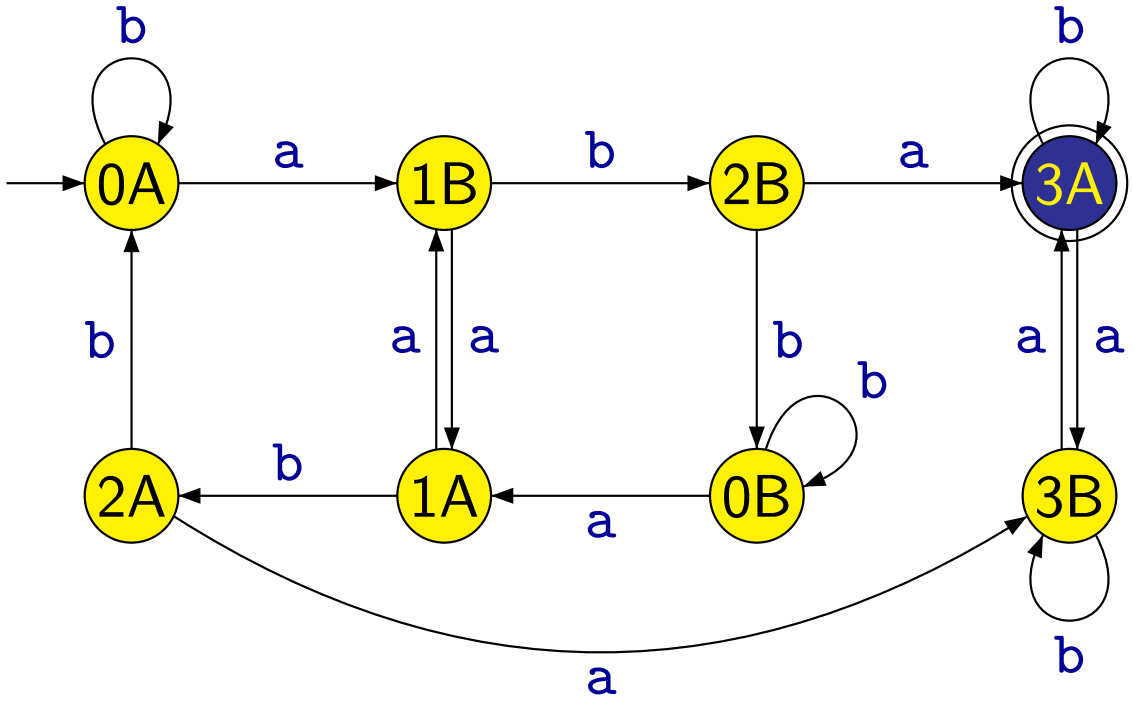




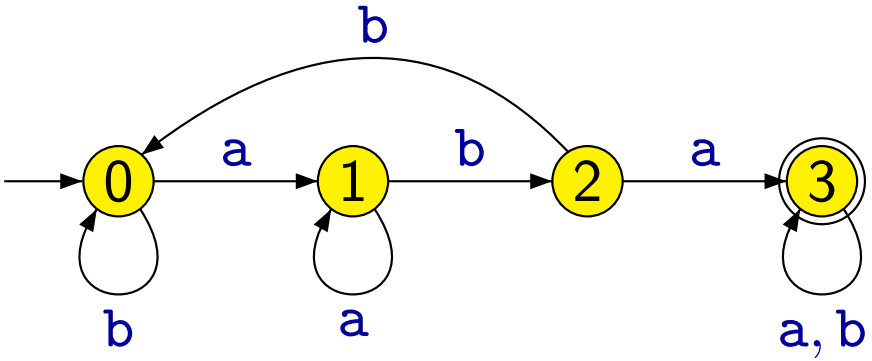
Automat pro průnik jazyků

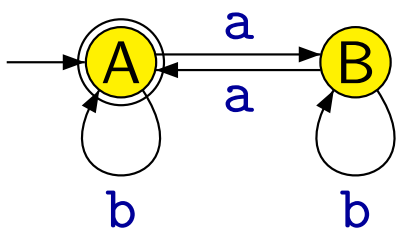


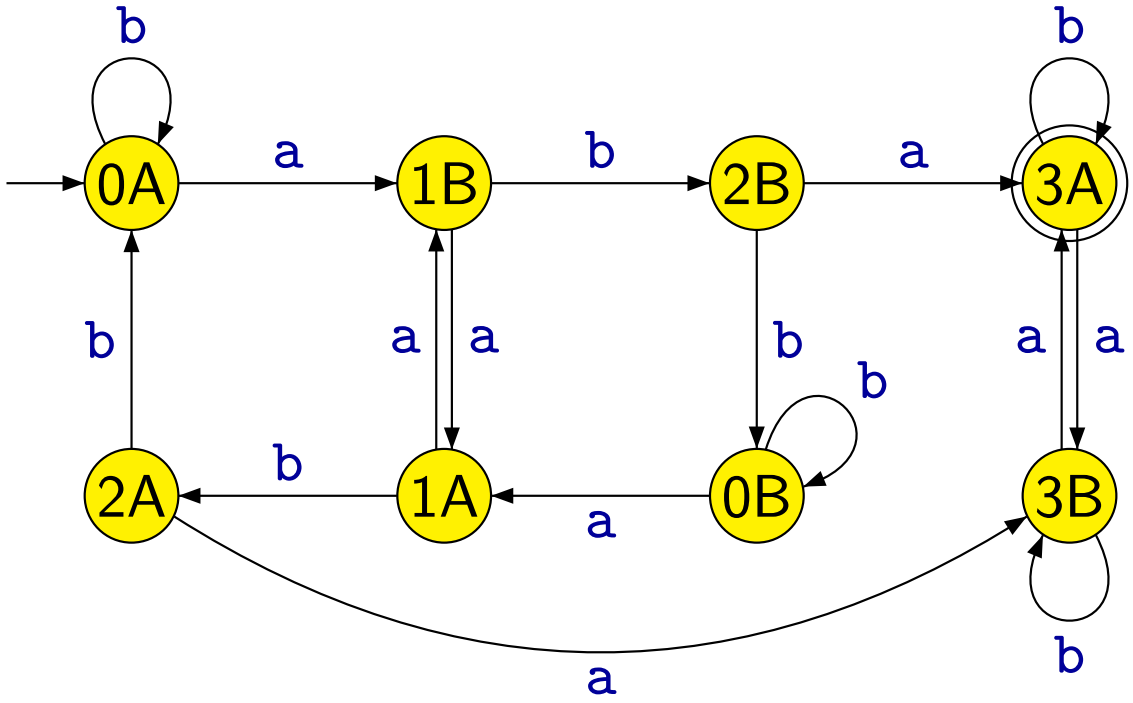




Automat pro průnik jazyků







Automat pro průnik jazyků

Formálně můžeme popsat tuto konstrukci následovně:

Předpokládáme, že máme dva deterministické konečné automaty $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ a $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.

K nim setrojíme DKA $A = (Q, \Sigma, \delta, q_0, F)$ kde:

- $Q = Q_1 \times Q_2$
- $\delta(\,(q_1, q_2), a\,) = (\,\delta_1(q_1, a), \delta_2(q_2, a)\,)$ pro všechna $q_1 \in Q_1$, $q_2 \in Q_2$, $a \in \Sigma$
- $q_0 = (q_{01}, q_{02})$
- $F = F_1 \times F_2$

Není težké ověřit, že pro libovolné slovo $w \in \Sigma^*$ platí, že $w \in L(A)$ právě tehdy, když $w \in L(A_1)$ a $w \in L(A_2)$, tj.

$$L(A) = L(A_1) \cap L(A_2)$$

Průnik regulárních jazyků

Věta

Jestliže jazyky $L_1, L_2 \subseteq \Sigma^*$ jsou regulární, pak také jazyk $L_1 \cap L_2$ je regulární.

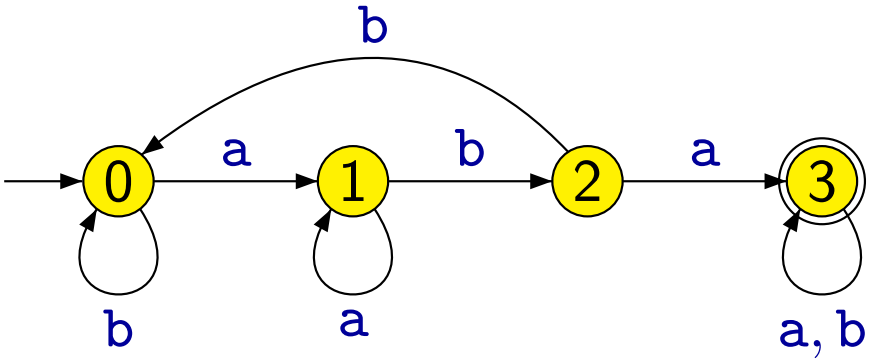
Důkaz: Předpokládejme, že A_1 a A_2 jsou deterministické konečné automaty takové, že

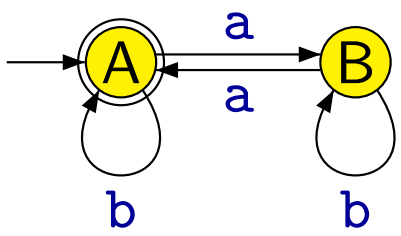
$$L_1 = L(A_1) \qquad L_2 = L(A_2)$$

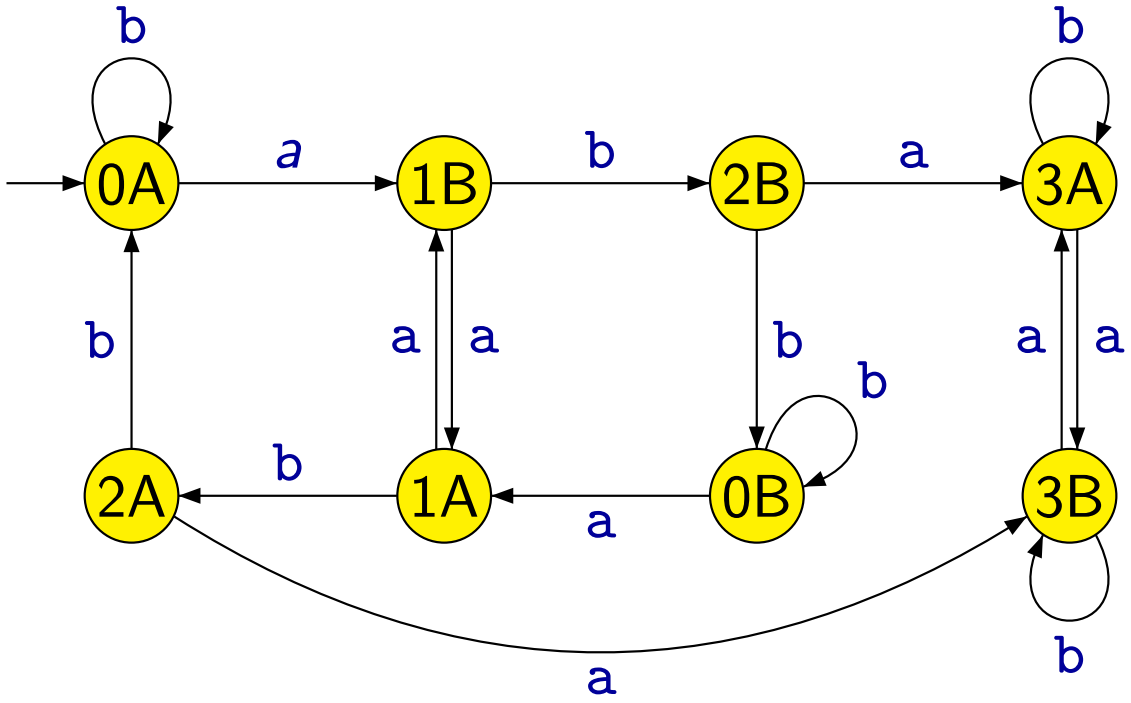
Popsanou konstrukcí k nim můžeme sestrojit deterministický konečný automat A takový, že

$$L(A) = L(A_1) \cap L(A_2) = L_1 \cap L_2$$

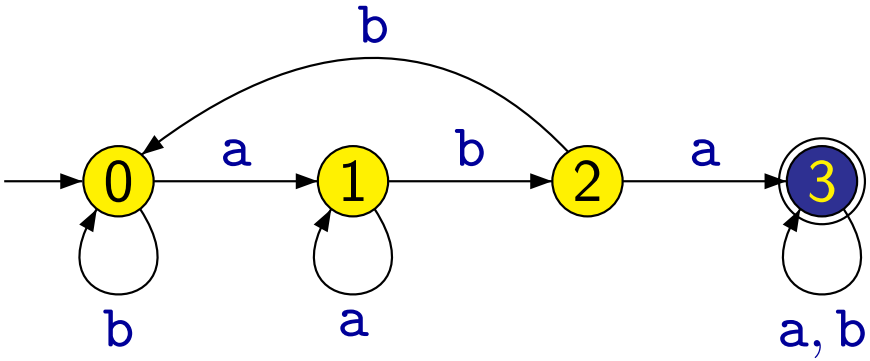
Automat pro sjednocení jazyků

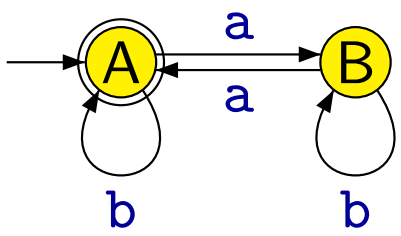


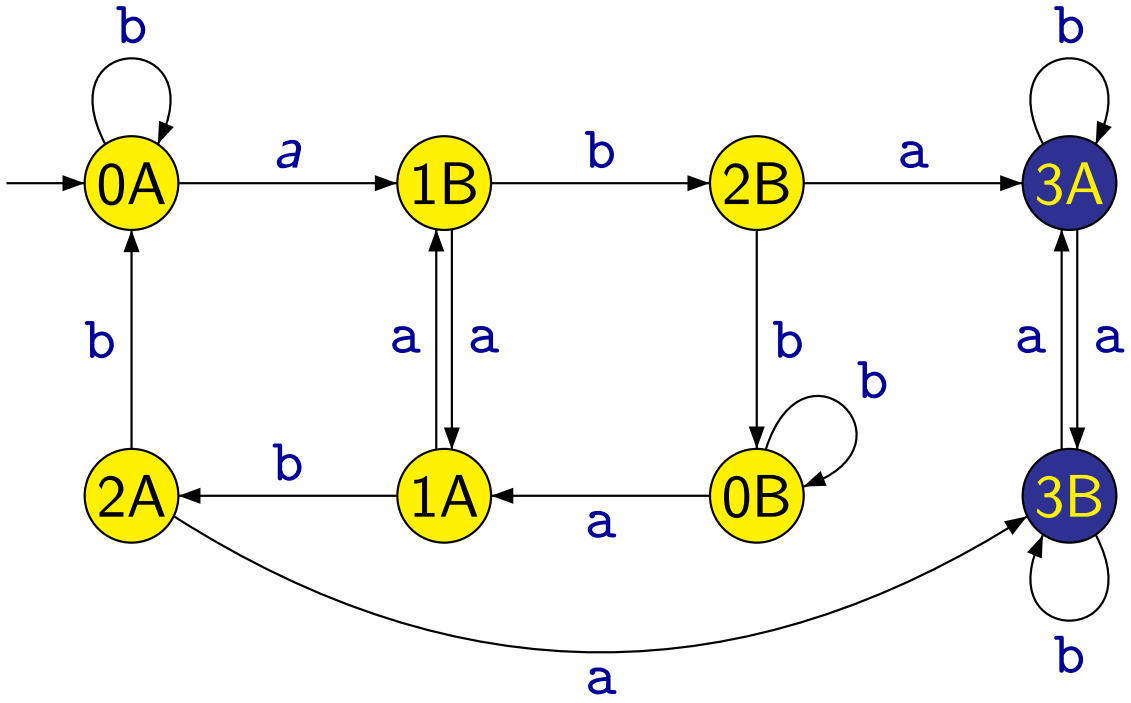




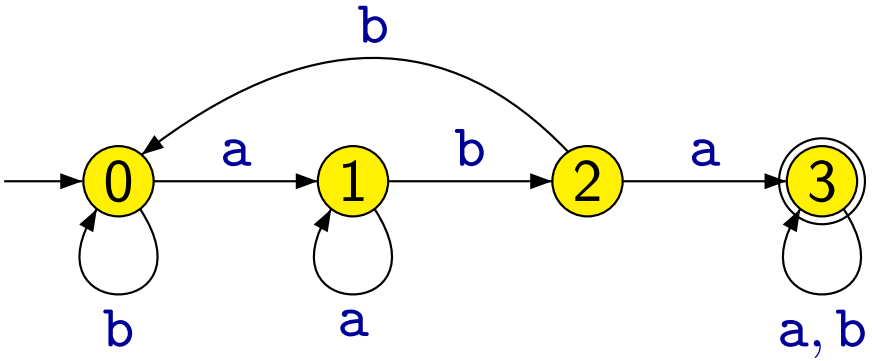
Automat pro sjednocení jazyků

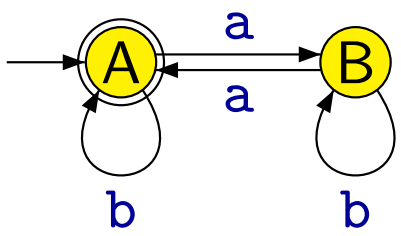


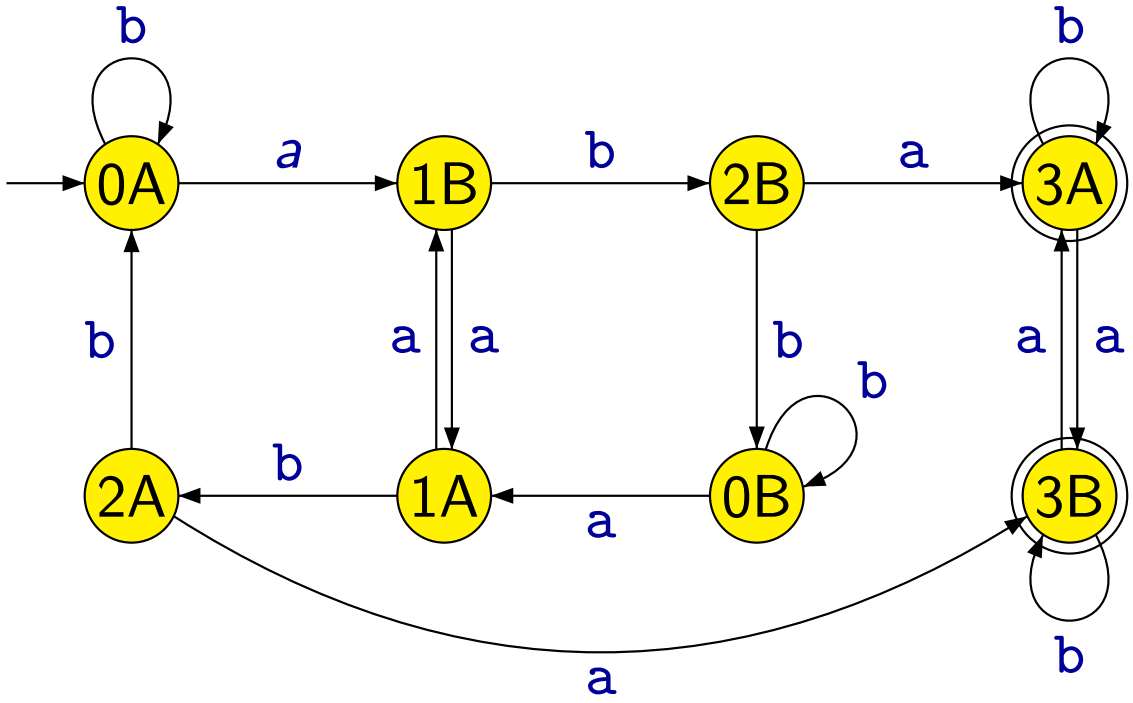




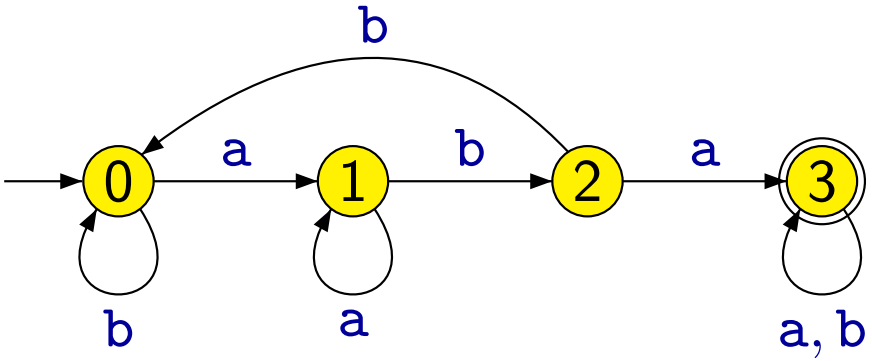
Automat pro sjednocení jazyků

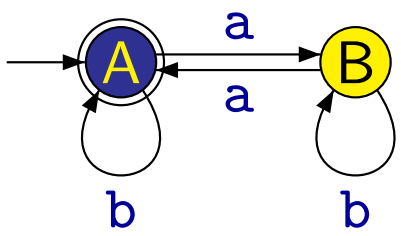


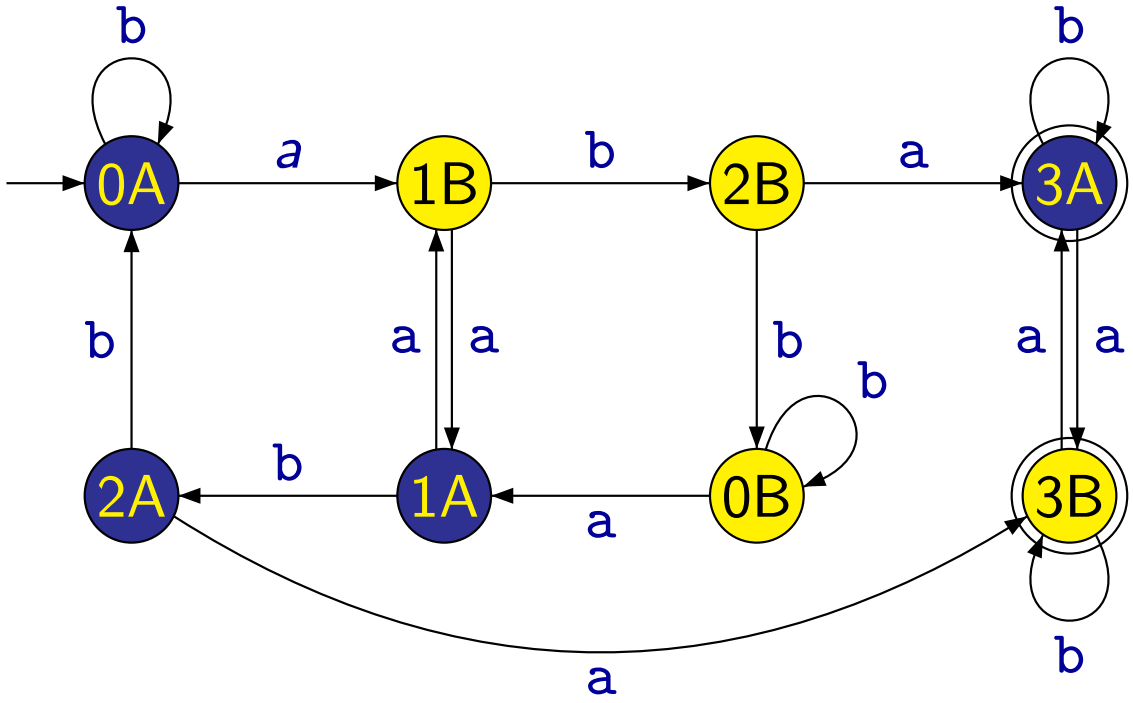




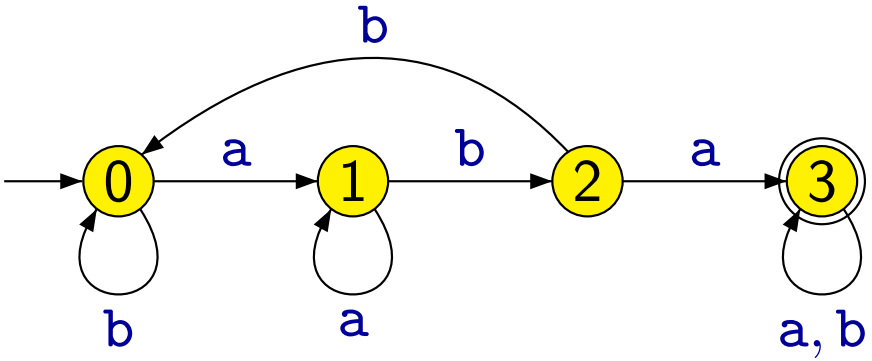
Automat pro sjednocení jazyků

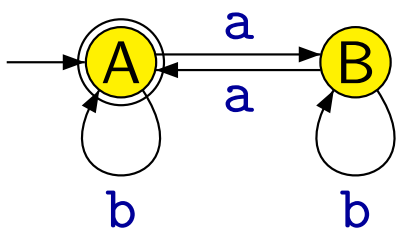


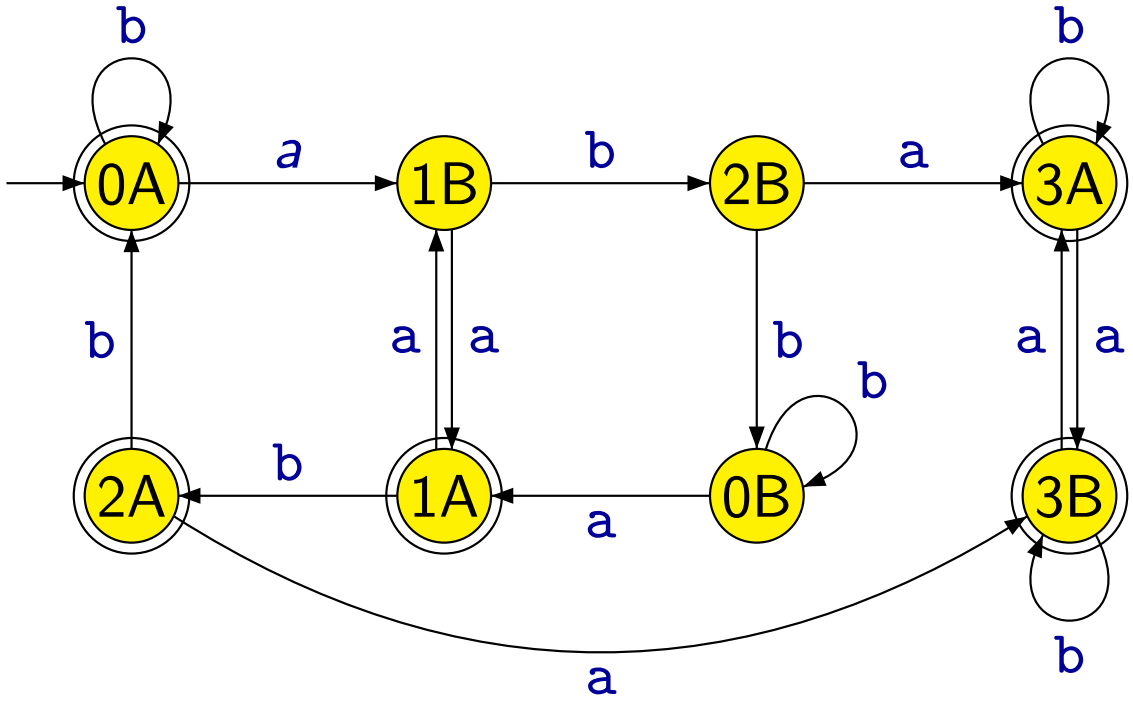




Automat pro sjednocení jazyků







Sjednocení regulárních jazyků

Konstukce automatu A, který přijímá **sjednocení** jazyků přijímaných automaty A_1 a A_2, tj. jazyk

$$L(A_1) \cup L(A_1)$$

je téměř stejná jako v případě automatu přijímajícího $L(A_1) \cap L(A_2)$.

Jediný rozdíl je v definici množiny přijímajících stavů:

- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

Sjednocení regulárních jazyků

Konstukce automatu A, který přijímá **sjednocení** jazyků přijímaných automaty A_1 a A_2, tj. jazyk

$$L(A_1) \cup L(A_1)$$

je téměř stejná jako v případě automatu přijímajícího $L(A_1) \cap L(A_2)$.

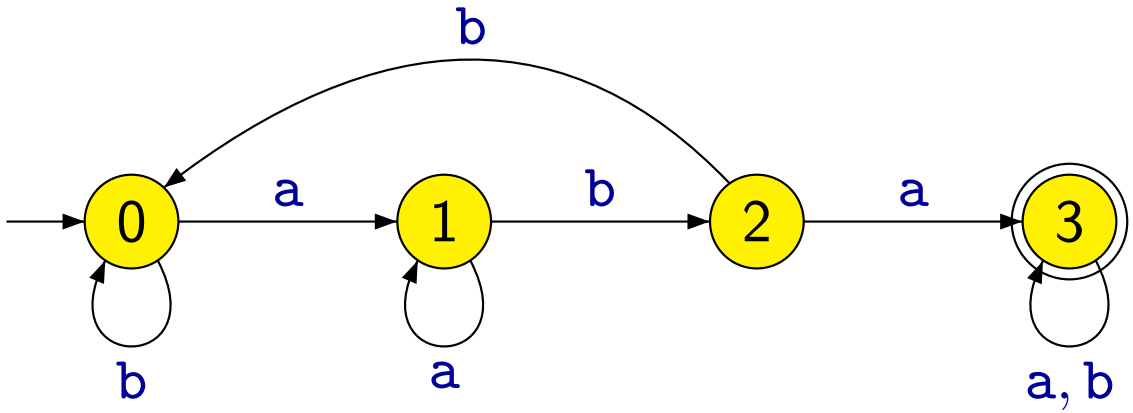
Jediný rozdíl je v definici množiny přijímajících stavů:

- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

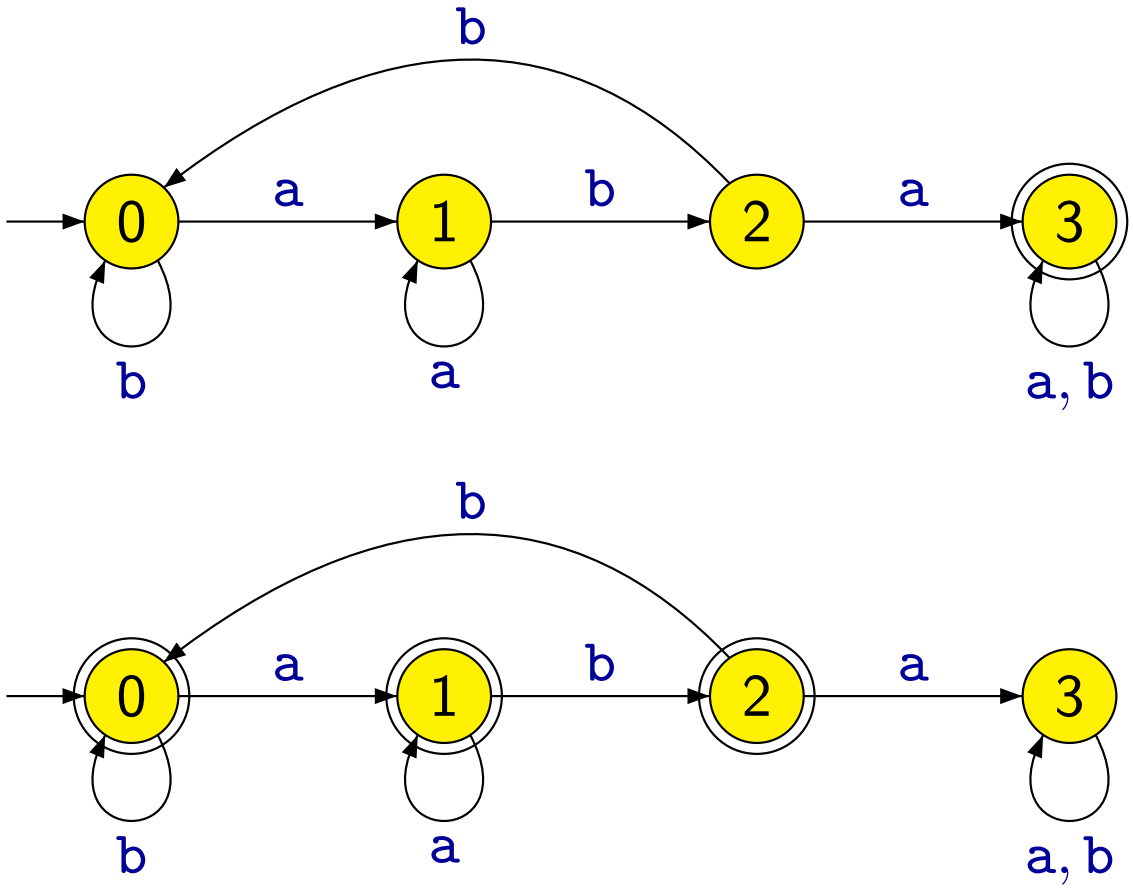
Věta

Jestliže jazyky $L_1, L_2 \subseteq \Sigma^*$ jsou regulární, pak také jazyk $L_1 \cup L_2$ je regulární.

Automat pro doplněk jazyka



Automat pro doplněk jazyka



Doplněk regulárního jazyka

K DKA $A = (Q, \Sigma, \delta, q_0, F)$ sestrojíme DKA $A' = (Q, \Sigma, \delta, q_0, Q - F)$.

Je očividné, že pro každé slovo $w \in \Sigma^*$ platí, že $w \in L(A')$ právě tehdy, když $w \notin L(A)$, tj.

$$L(A') = \overline{L(A)}$$

Doplněk regulárního jazyka

K DKA $A = (Q, \Sigma, \delta, q_0, F)$ sestrojíme DKA $A' = (Q, \Sigma, \delta, q_0, Q - F)$.

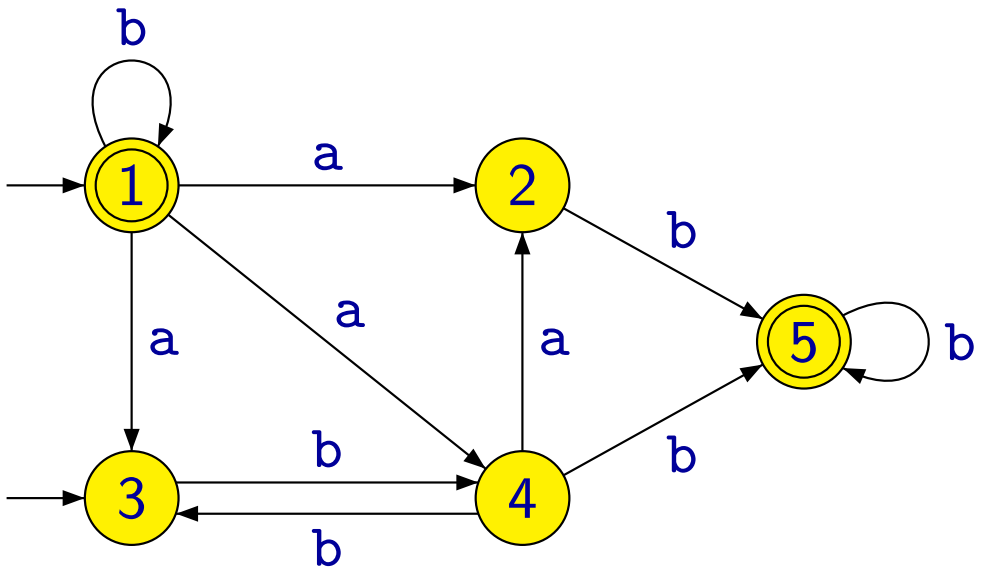
Je očividné, že pro každé slovo $w \in \Sigma^*$ platí, že $w \in L(A')$ právě tehdy, když $w \notin L(A)$, tj.

$$L(A') = \overline{L(A)}$$

Věta

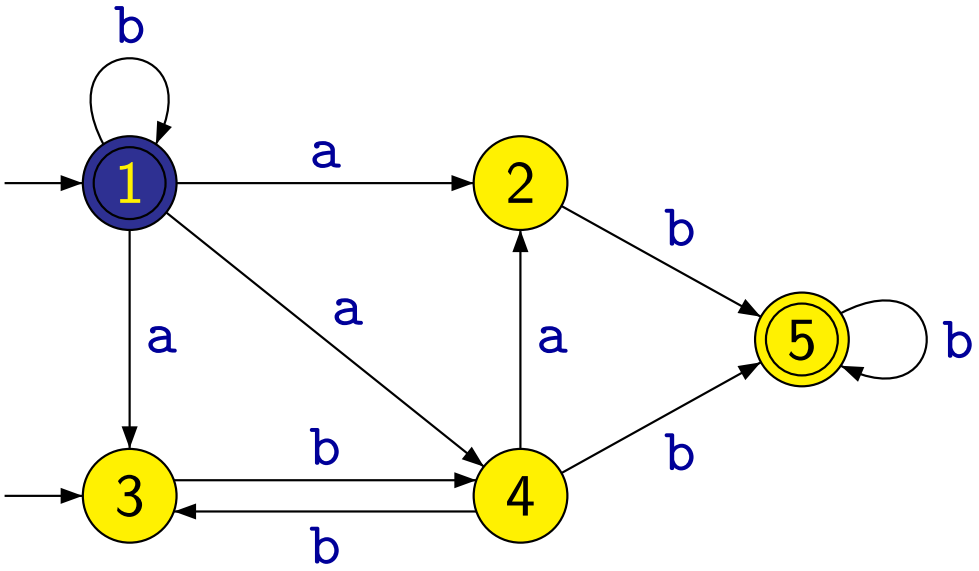
Jestliže jazyk L je regulární, pak také jeho doplněk $\overline{L}$ je regulární.

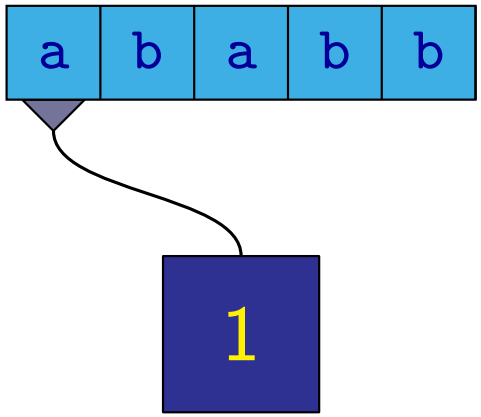
Nedeterministický konečný automat



- Z jednoho stavu může vézt libovolný (i nulový) počet přechodů označených stejným symbolem.
- V automatu může být víc než jeden počáteční stav.

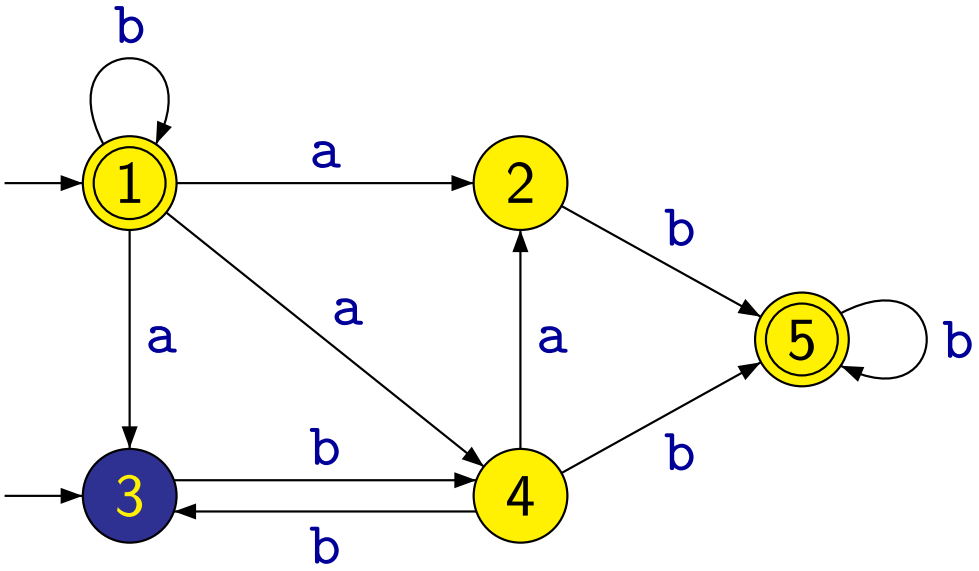
Nedeterministický konečný automat

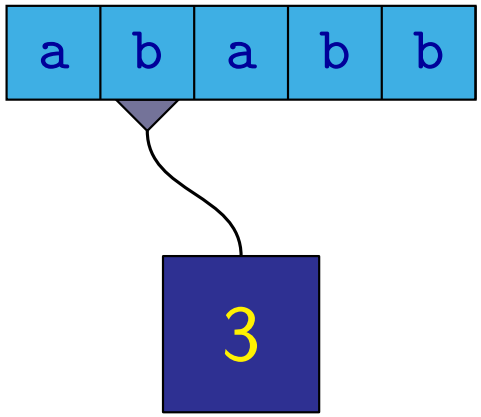




1

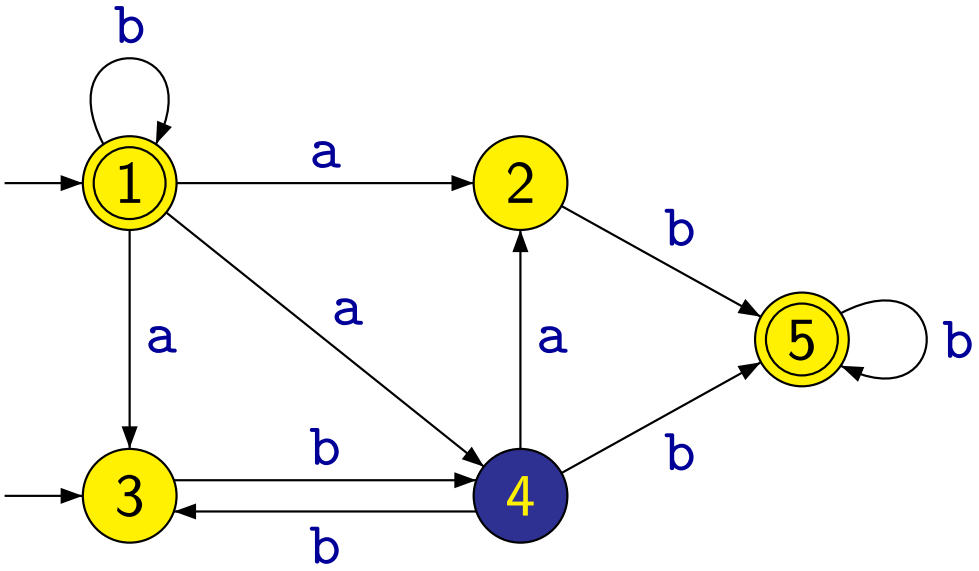
Nedeterministický konečný automat

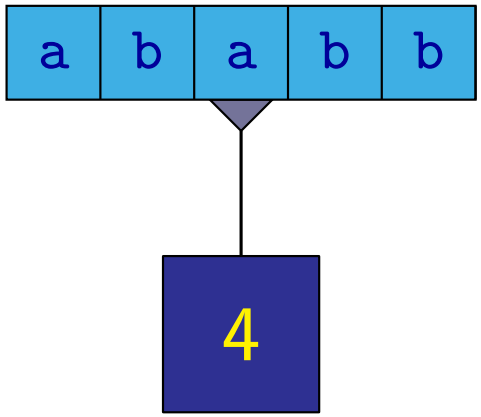




$1 \xrightarrow{a} 3$

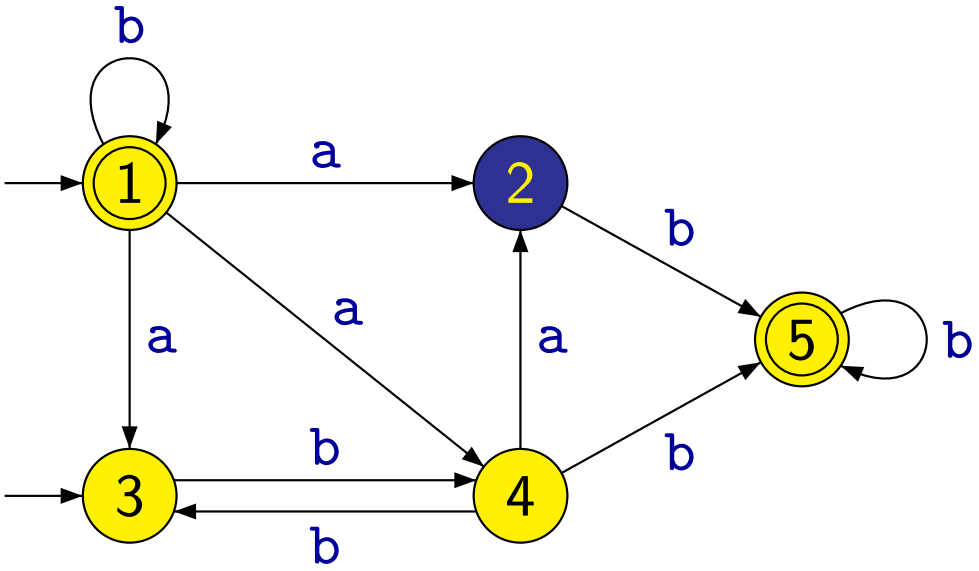
Nedeterministický konečný automat

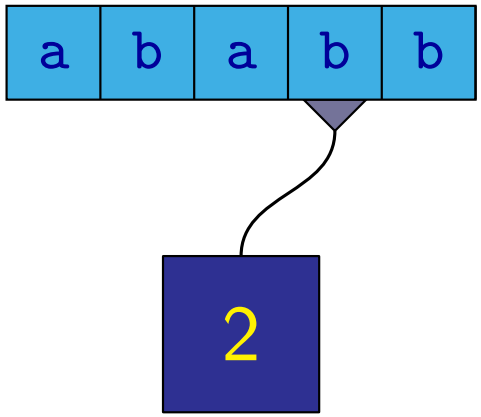




$1 \xrightarrow{a} 3 \xrightarrow{b} 4$

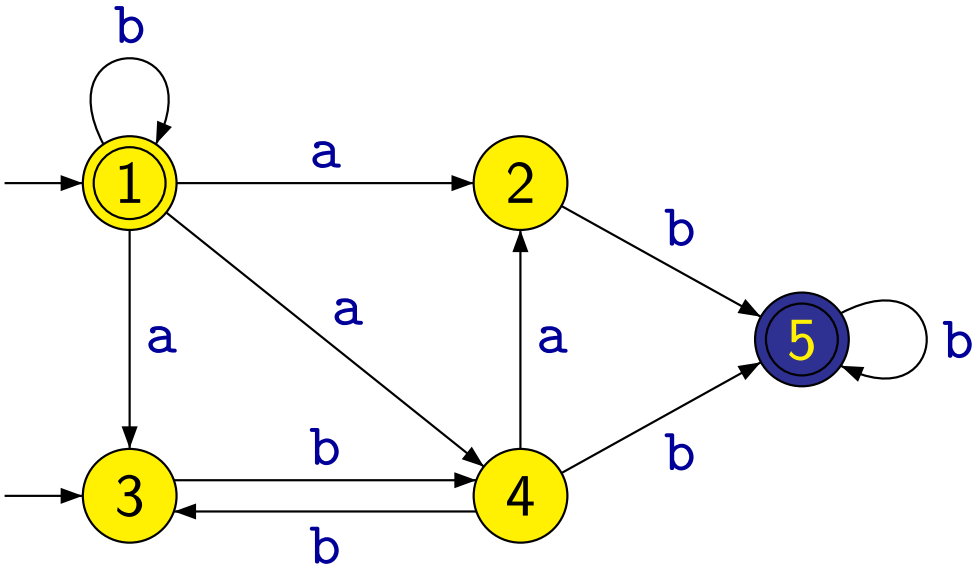
Nedeterministický konečný automat

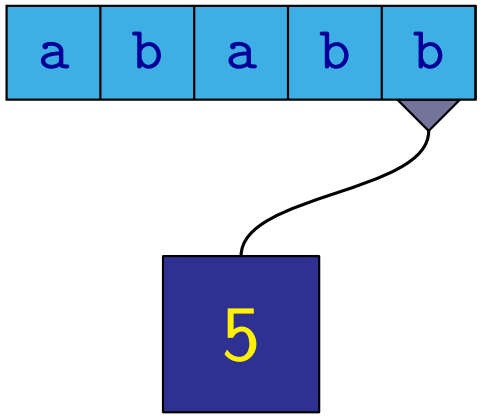




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{a} 2$$

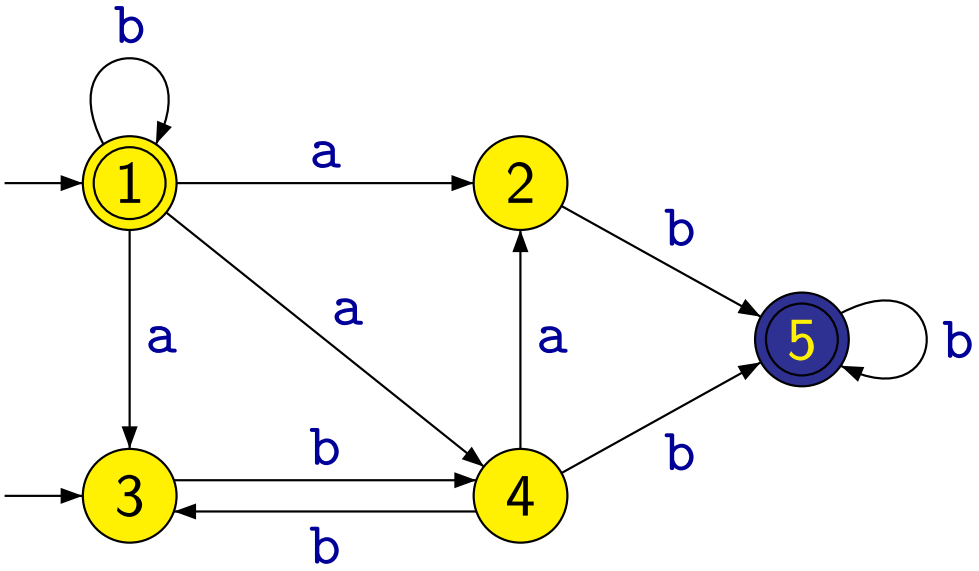
Nedeterministický konečný automat

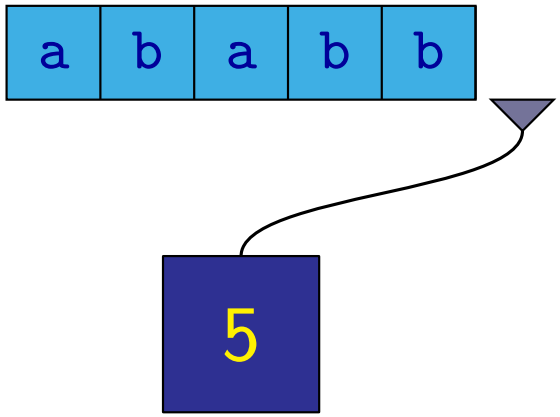




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{a} 2 \xrightarrow{b} 5$$

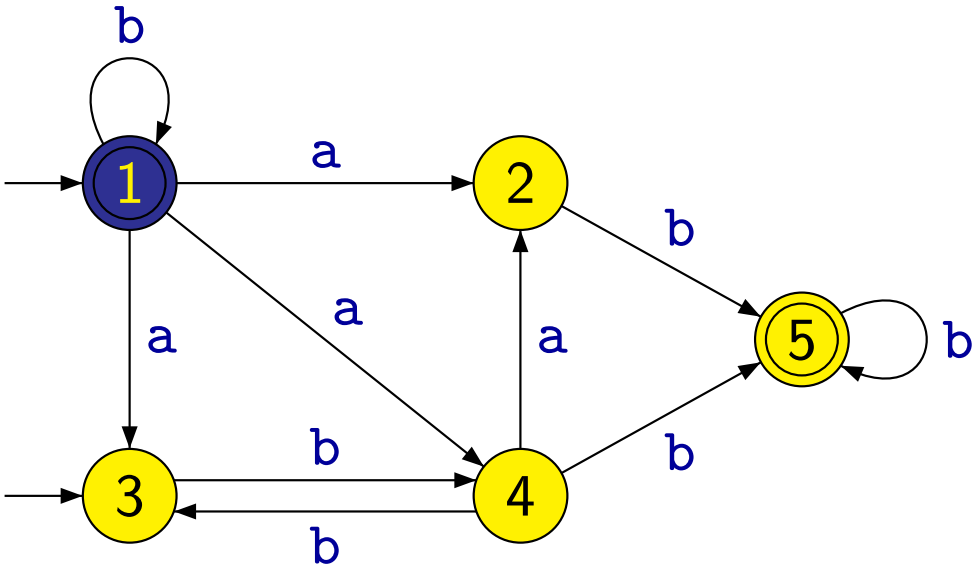
Nedeterministický konečný automat

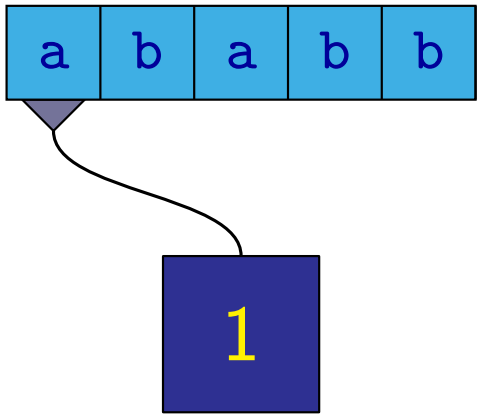




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{a} 2 \xrightarrow{b} 5 \xrightarrow{b} 5$$

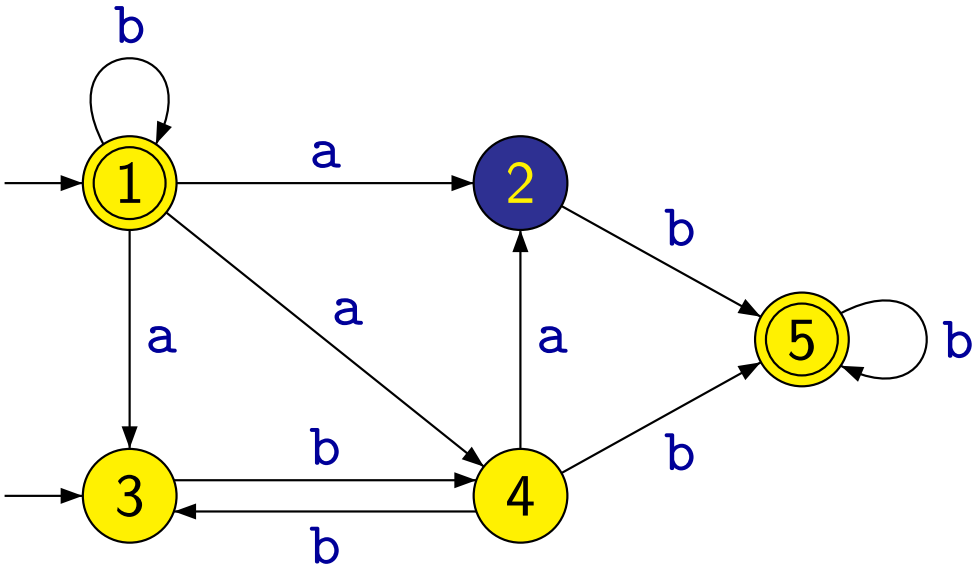
Nedeterministický konečný automat

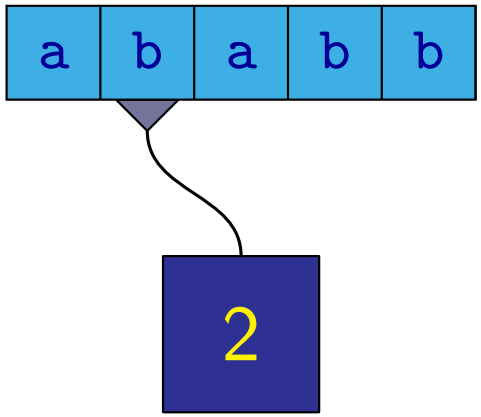




1

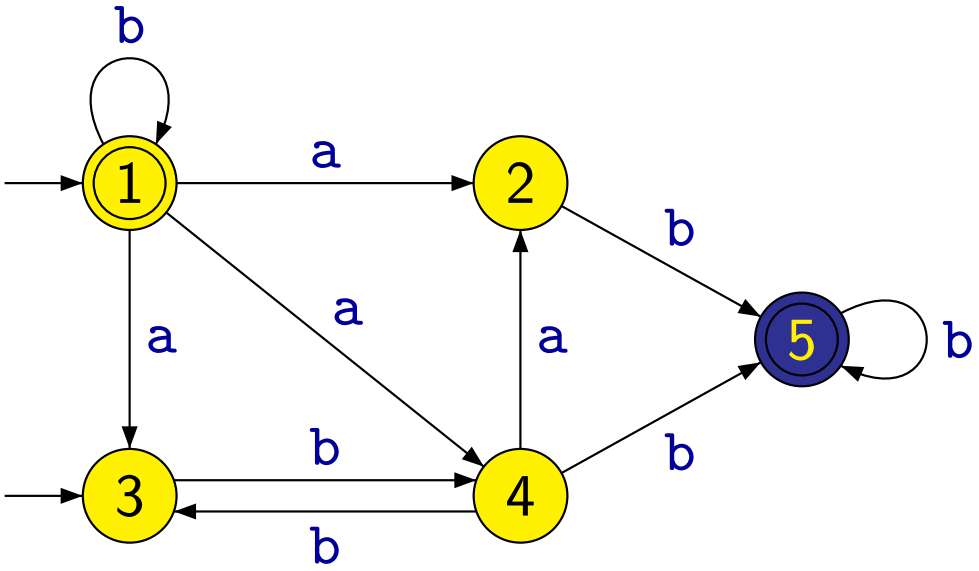
Nedeterministický konečný automat

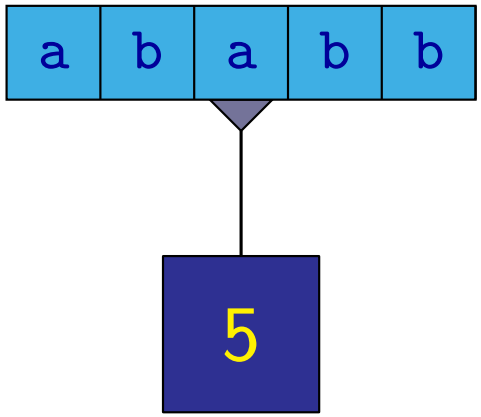




$1 \xrightarrow{a} 2$

Nedeterministický konečný automat

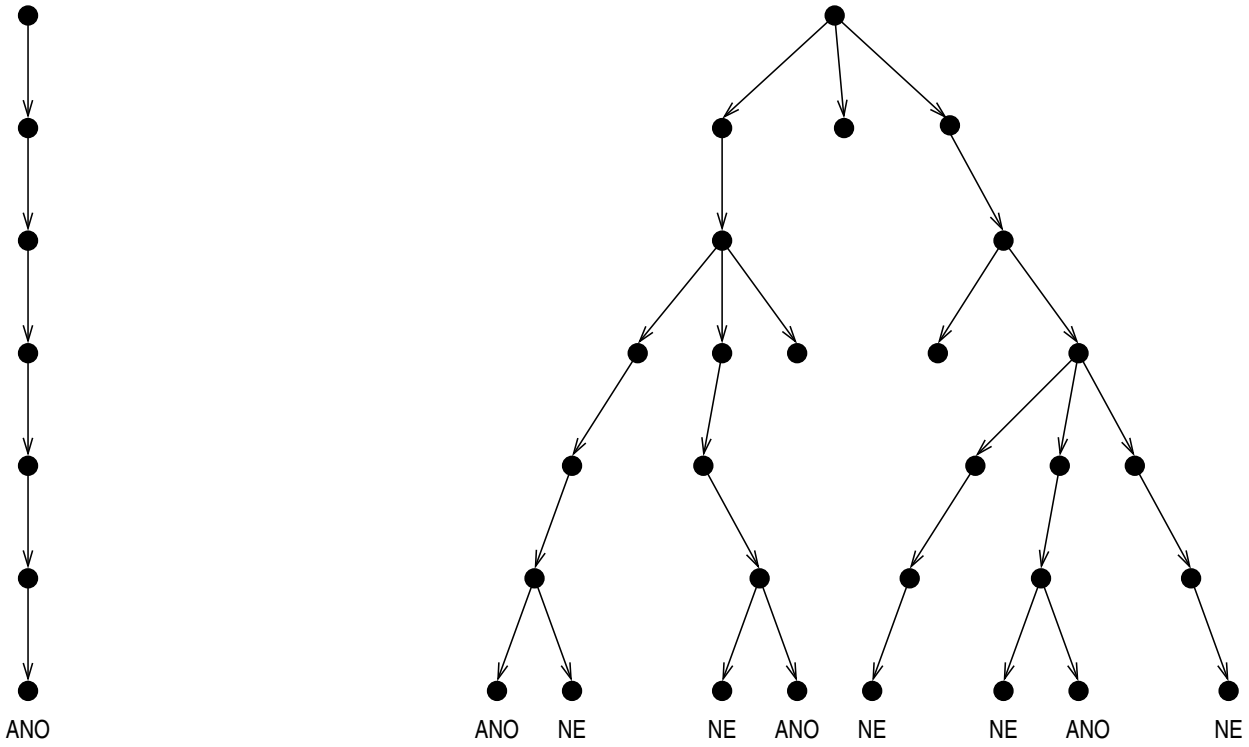




$1 \xrightarrow{a} 2 \xrightarrow{b} 5$

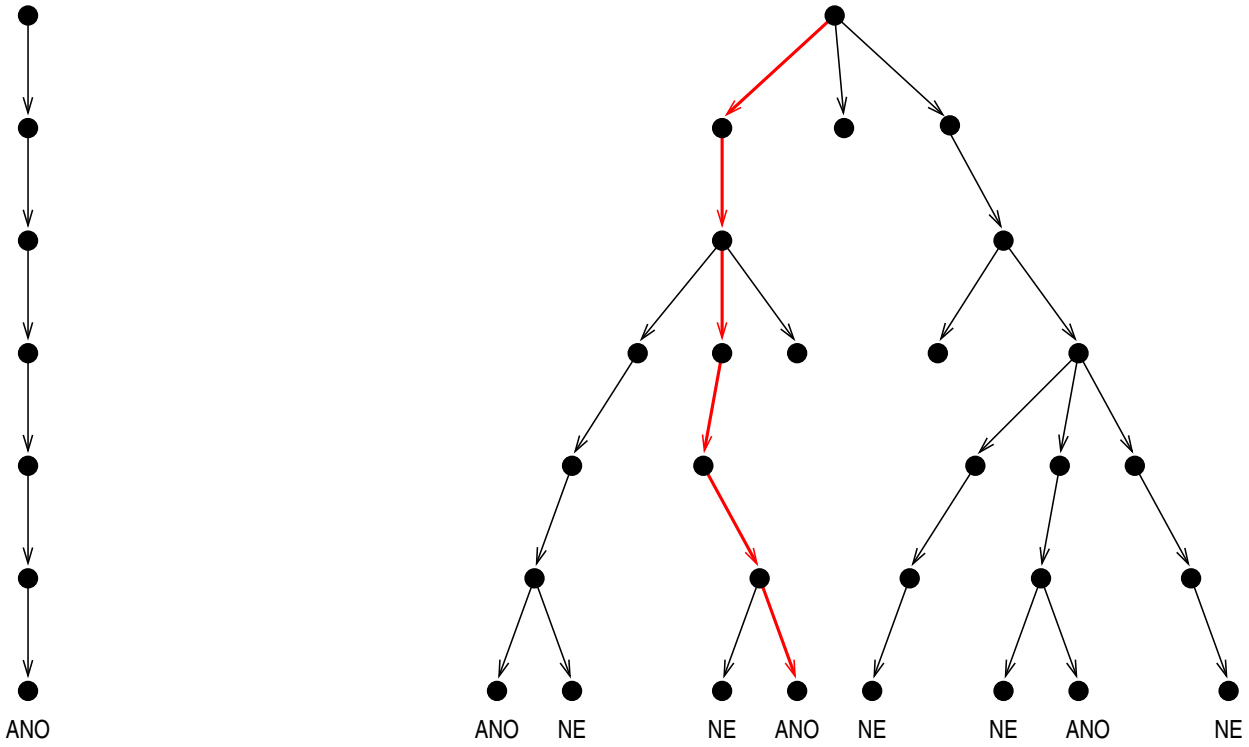
Nedeterministický konečný automat

Nedeterministický konečný automat přijímá dané slovo, jestliže **existuje** alespoň jeden jeho výpočet, který vede k přijetí tohoto slova.



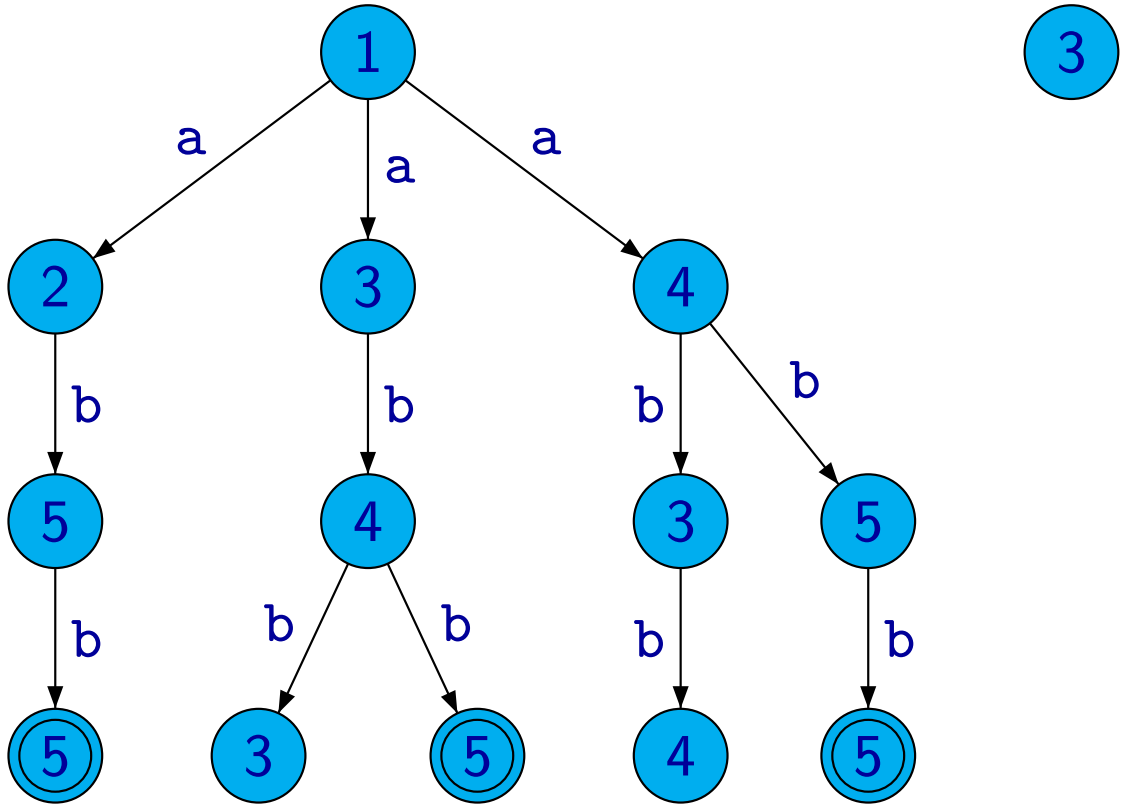
Nedeterministický konečný automat

Nedeterministický konečný automat přijímá dané slovo, jestliže **existuje** alespoň jeden jeho výpočet, který vede k přijetí tohoto slova.



Nedeterministický konečný automat

	a	b
$\leftrightarrow 1$	$2, 3, 4$	1
2	$-$	5
$\rightarrow 3$	$-$	4
4	2	$3, 5$
$\leftarrow 5$	$-$	5



Příklad: Les reprezentující všechny možné výpočty nad slovem abb.

Nedeterministický konečný automat

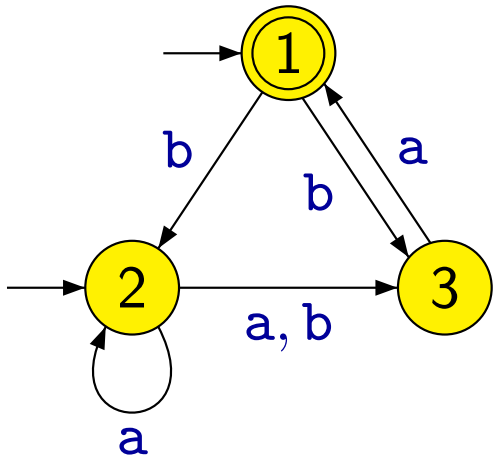
Formálně je **nedeterministický konečný automat** (**NKA**) definován jako pětice

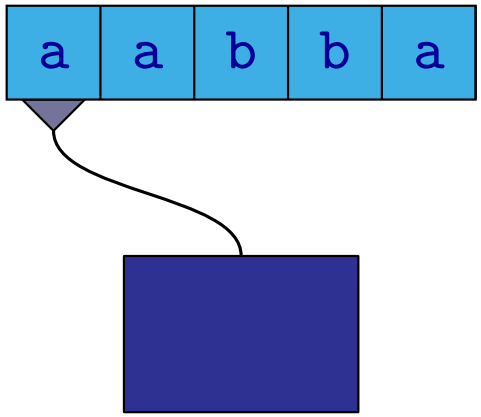
$$(Q, \Sigma, \delta, I, F)$$

kde:

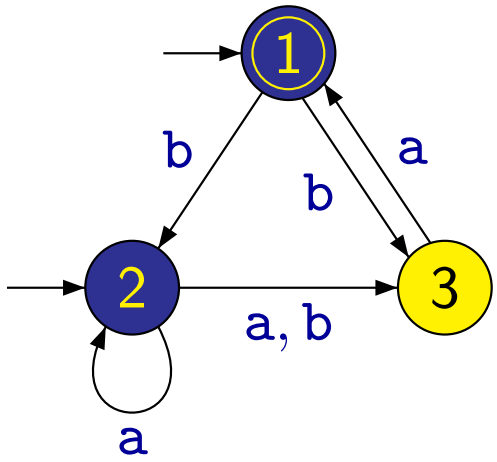
- Q je konečná množina **stavů**
- Σ je konečná **abeceda**
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ je **přechodová funkce**
- $I \subseteq Q$ je množina **počátečních stavů**
- $F \subseteq Q$ je množina **přijímajících stavů**

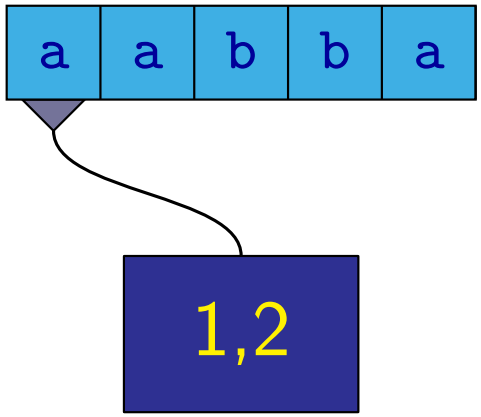
Převod NKA na DKA



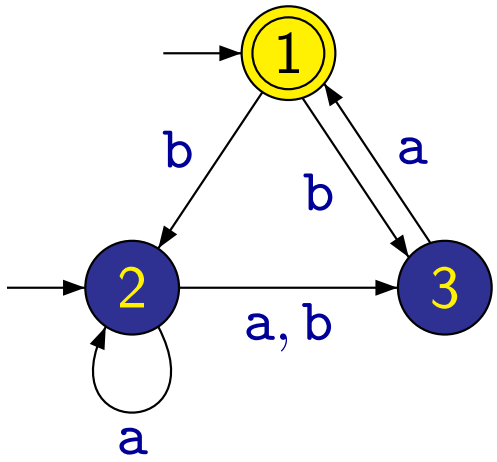


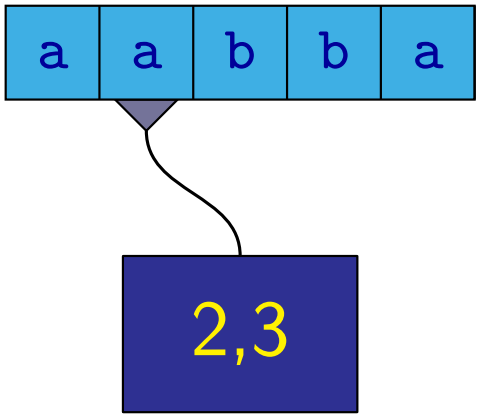
Převod NKA na DKA



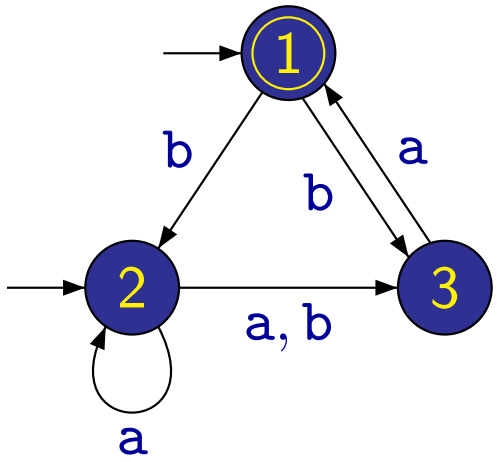


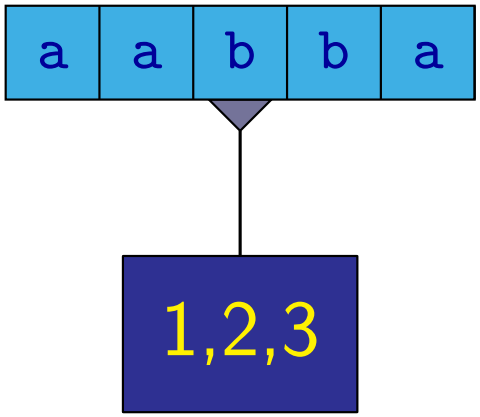
Převod NKA na DKA



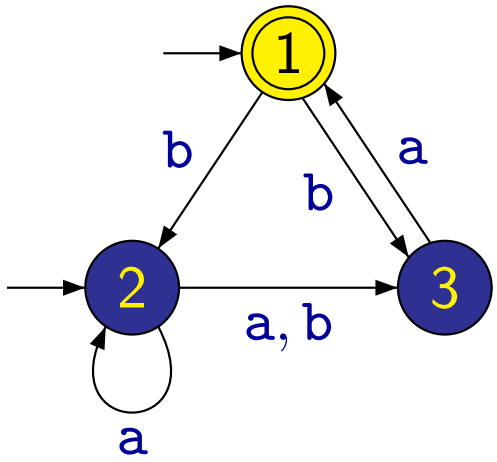


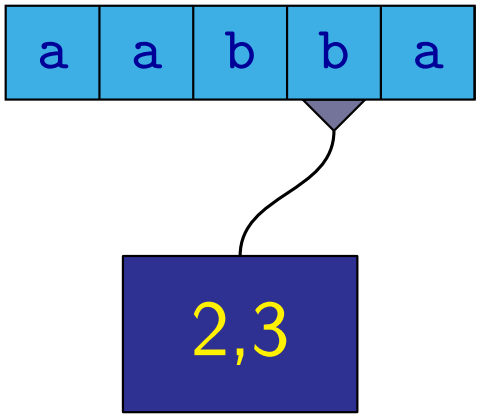
Převod NKA na DKA



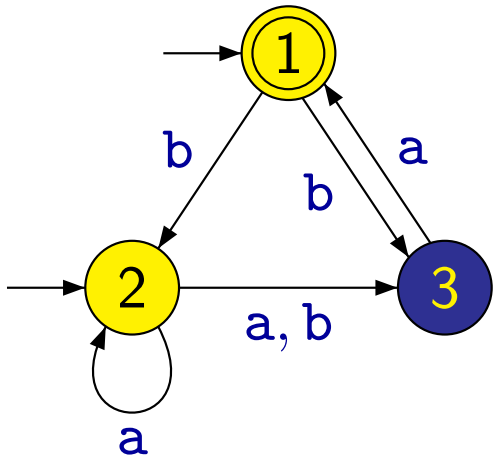


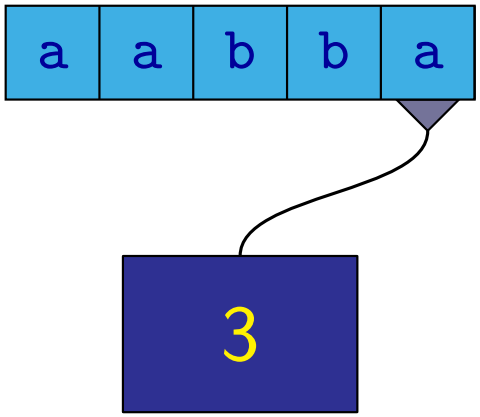
Převod NKA na DKA



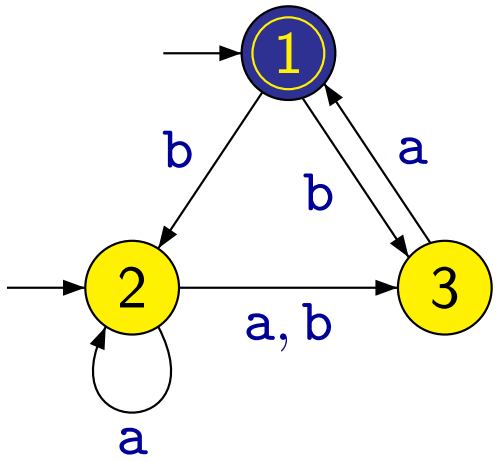


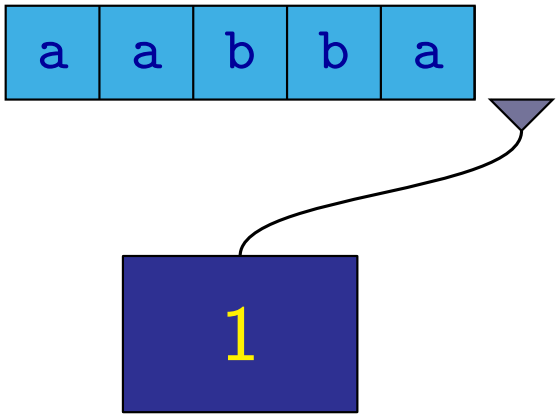
Převod NKA na DKA



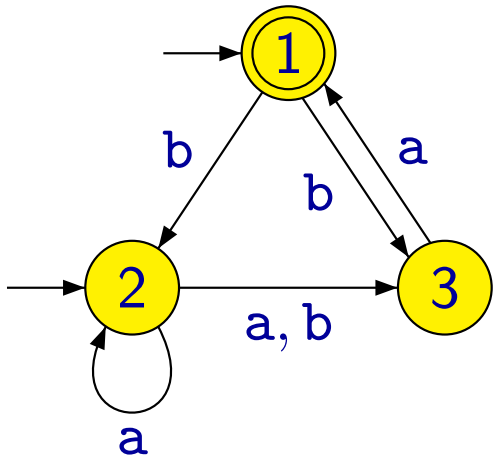


Převod NKA na DKA

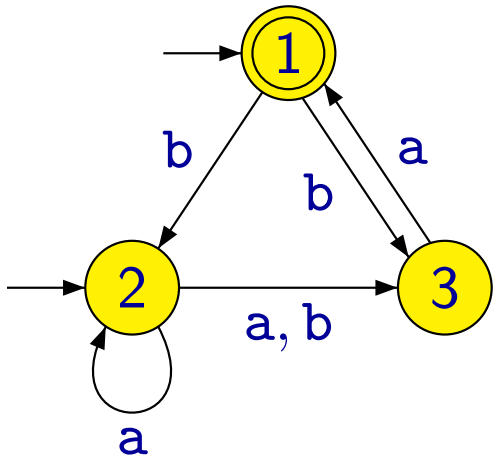




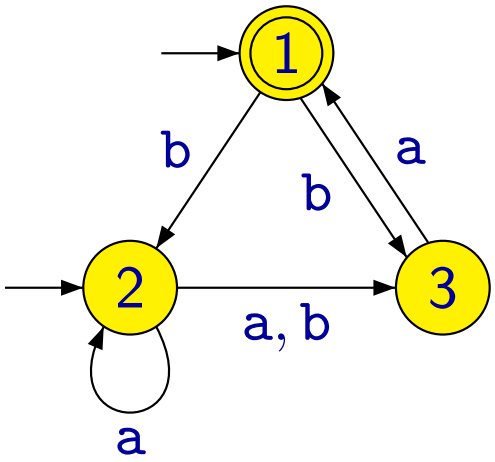
Převod NKA na DKA

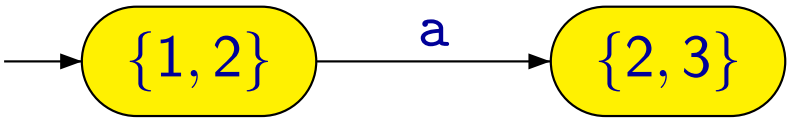


Převod NKA na DKA

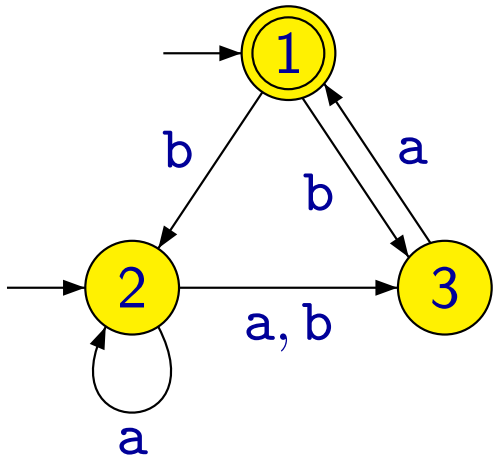


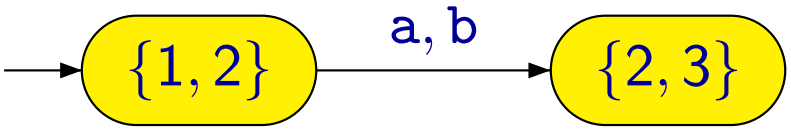
Převod NKA na DKA



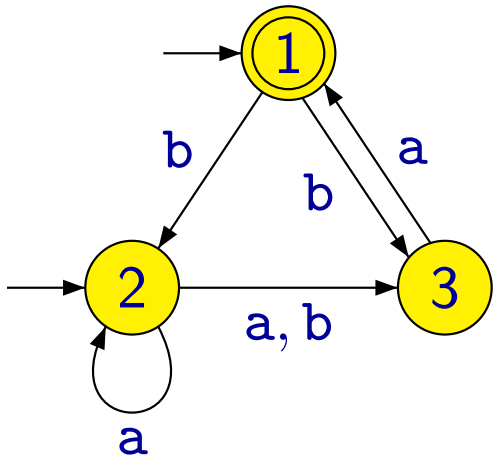


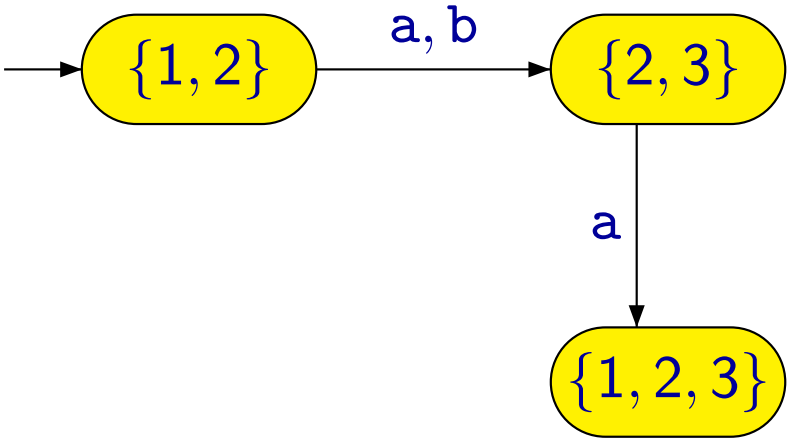
Převod NKA na DKA



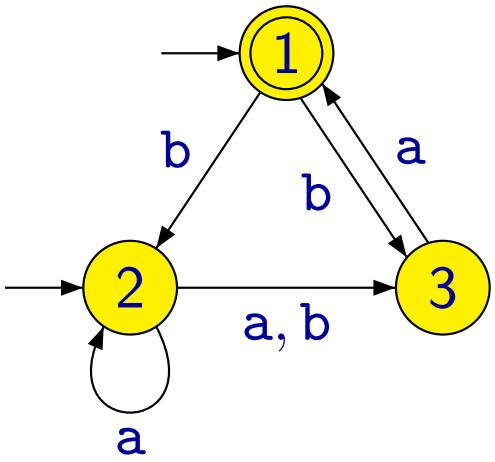


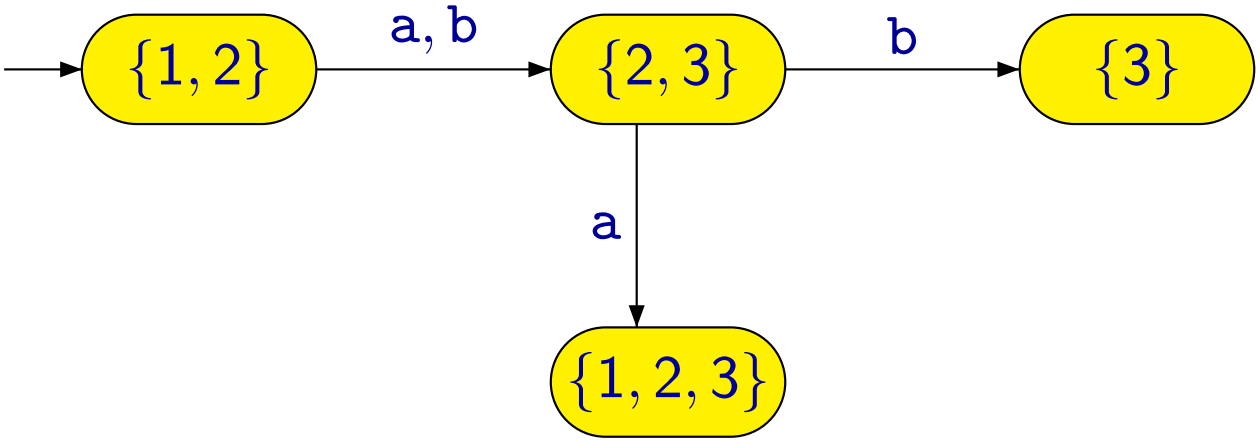
Převod NKA na DKA



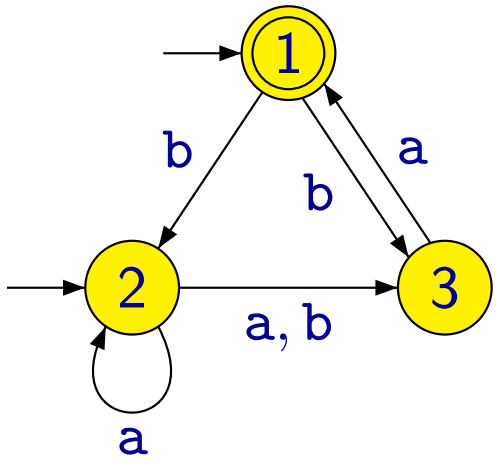


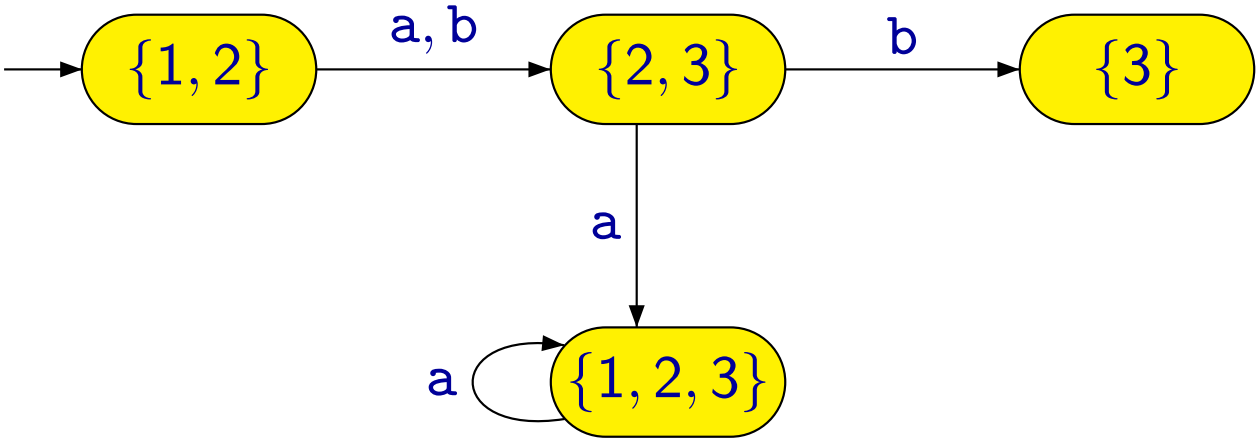
Převod NKA na DKA



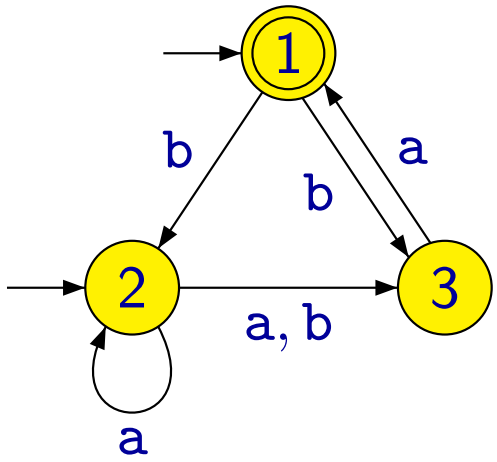


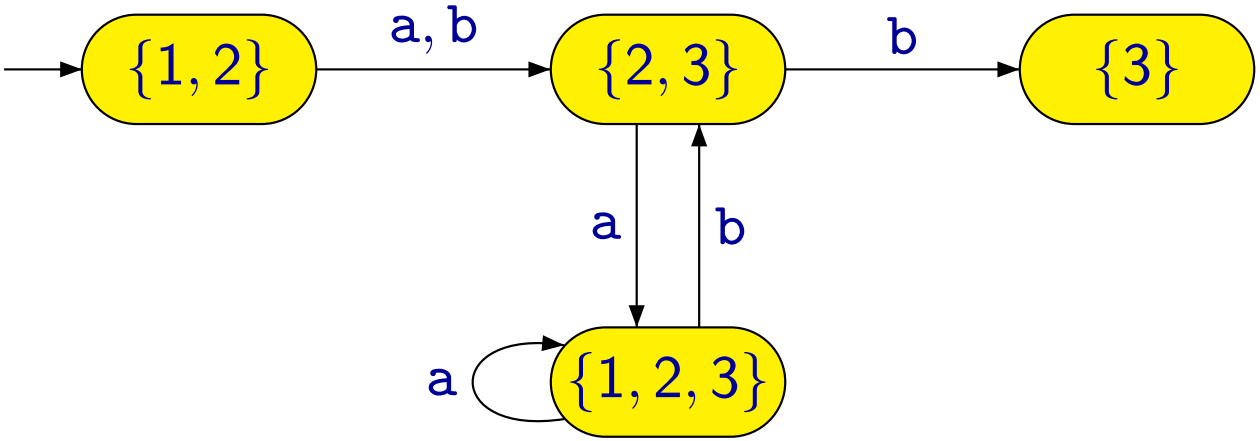
Převod NKA na DKA



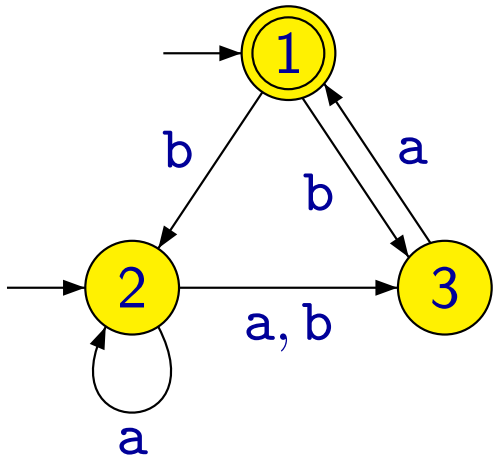


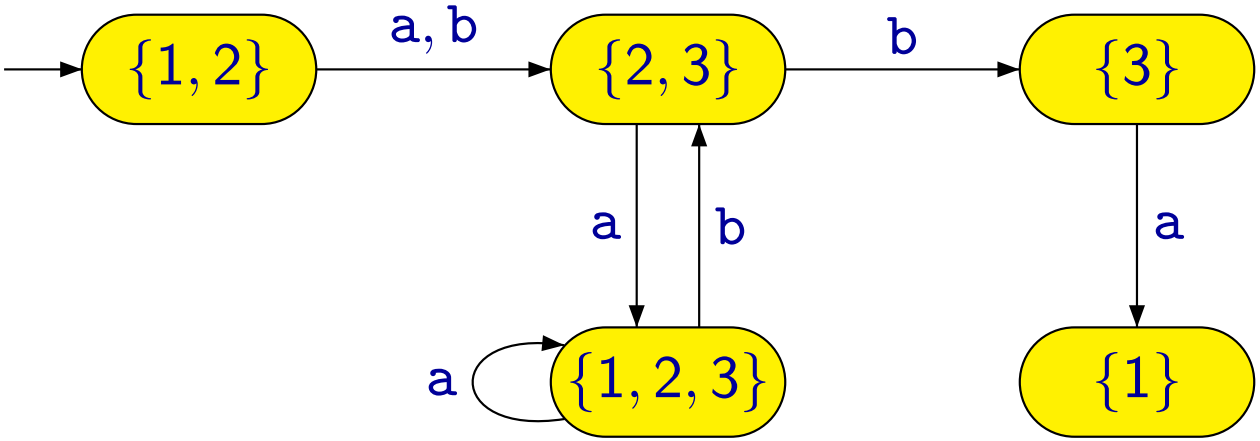
Převod NKA na DKA



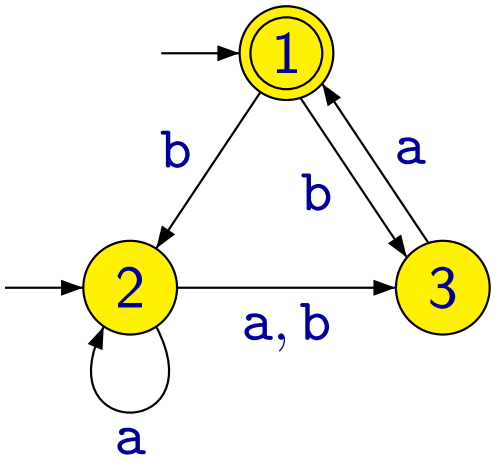


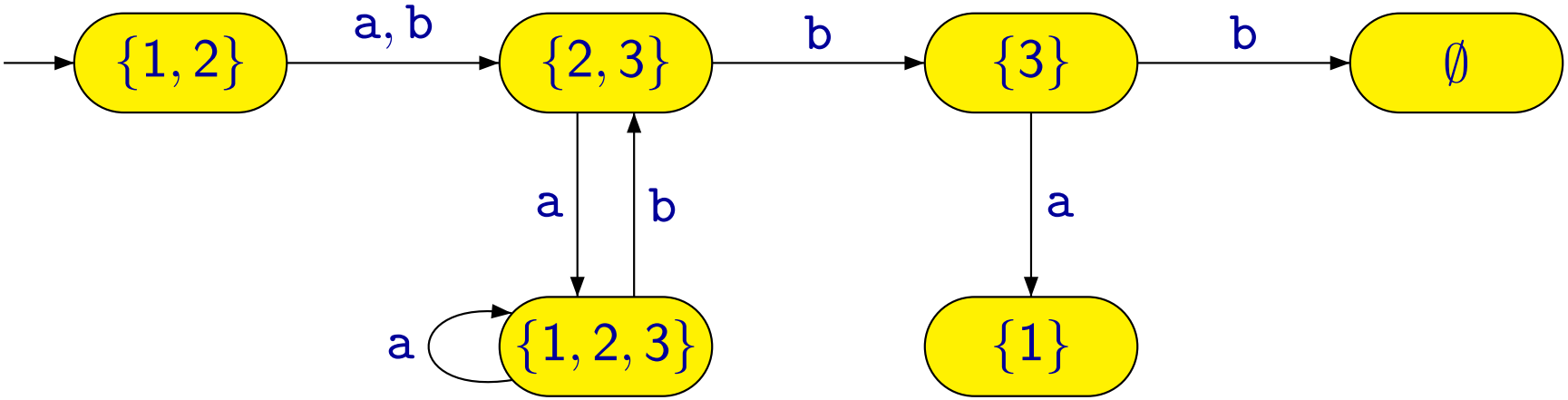
Převod NKA na DKA



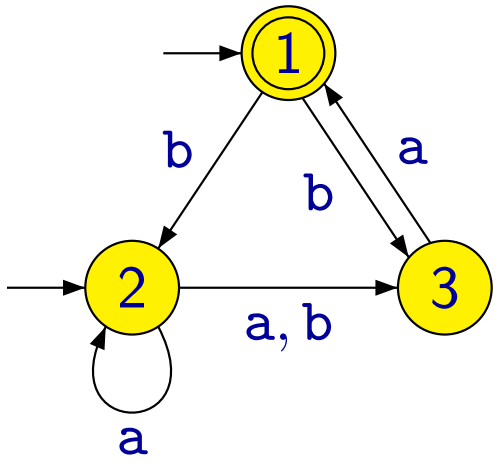


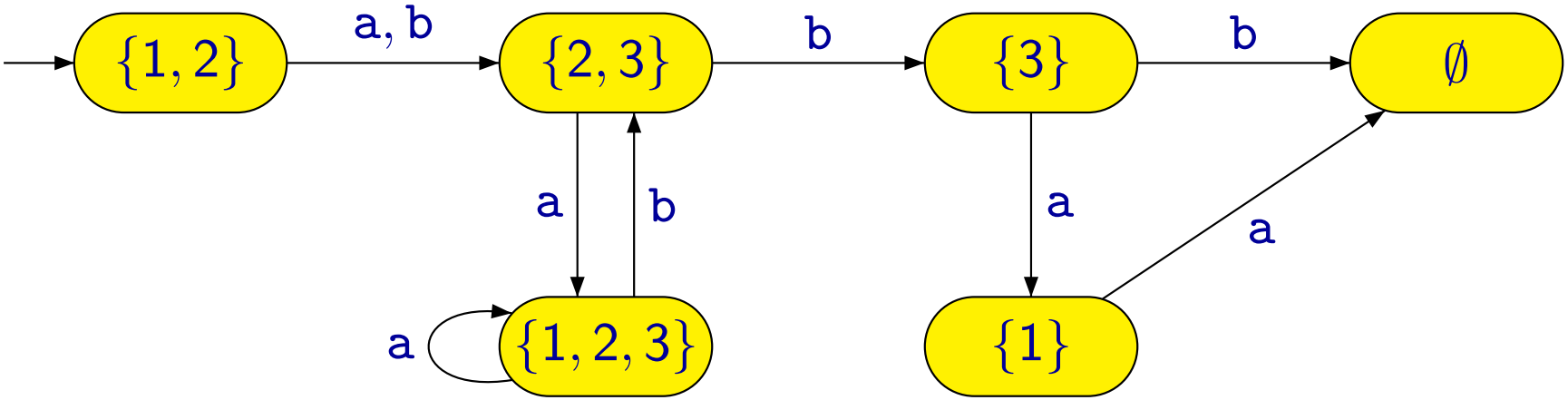
Převod NKA na DKA



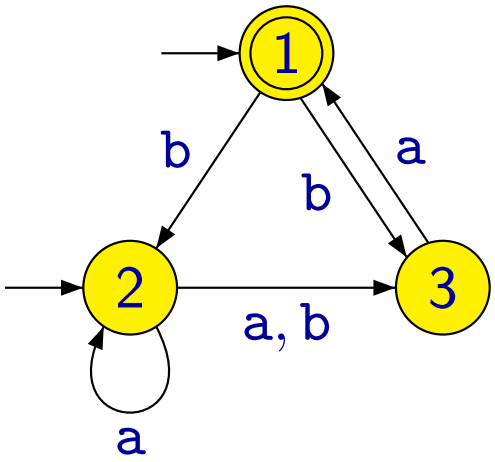


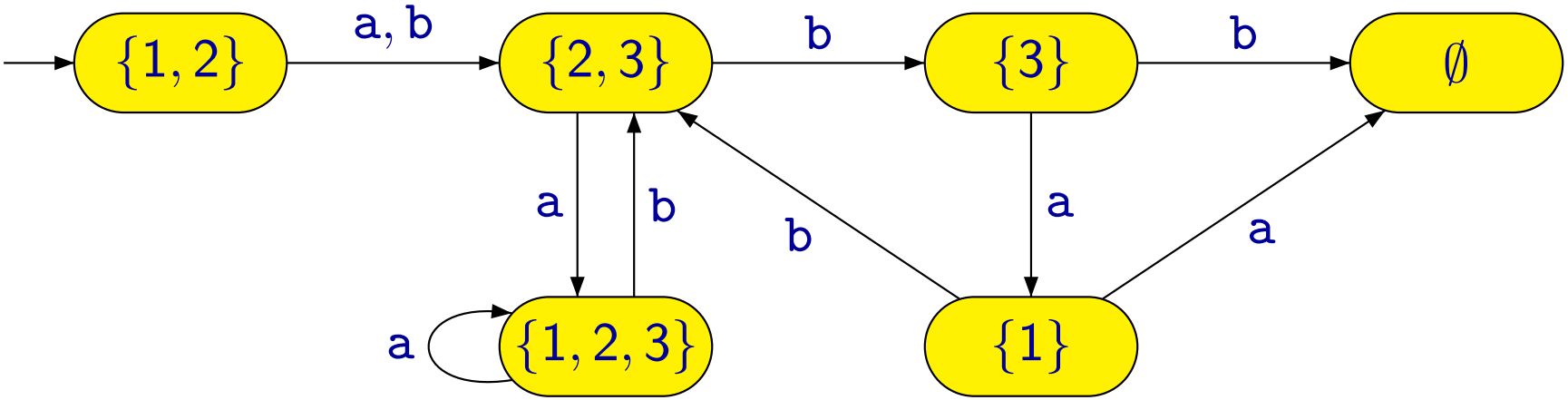
Převod NKA na DKA



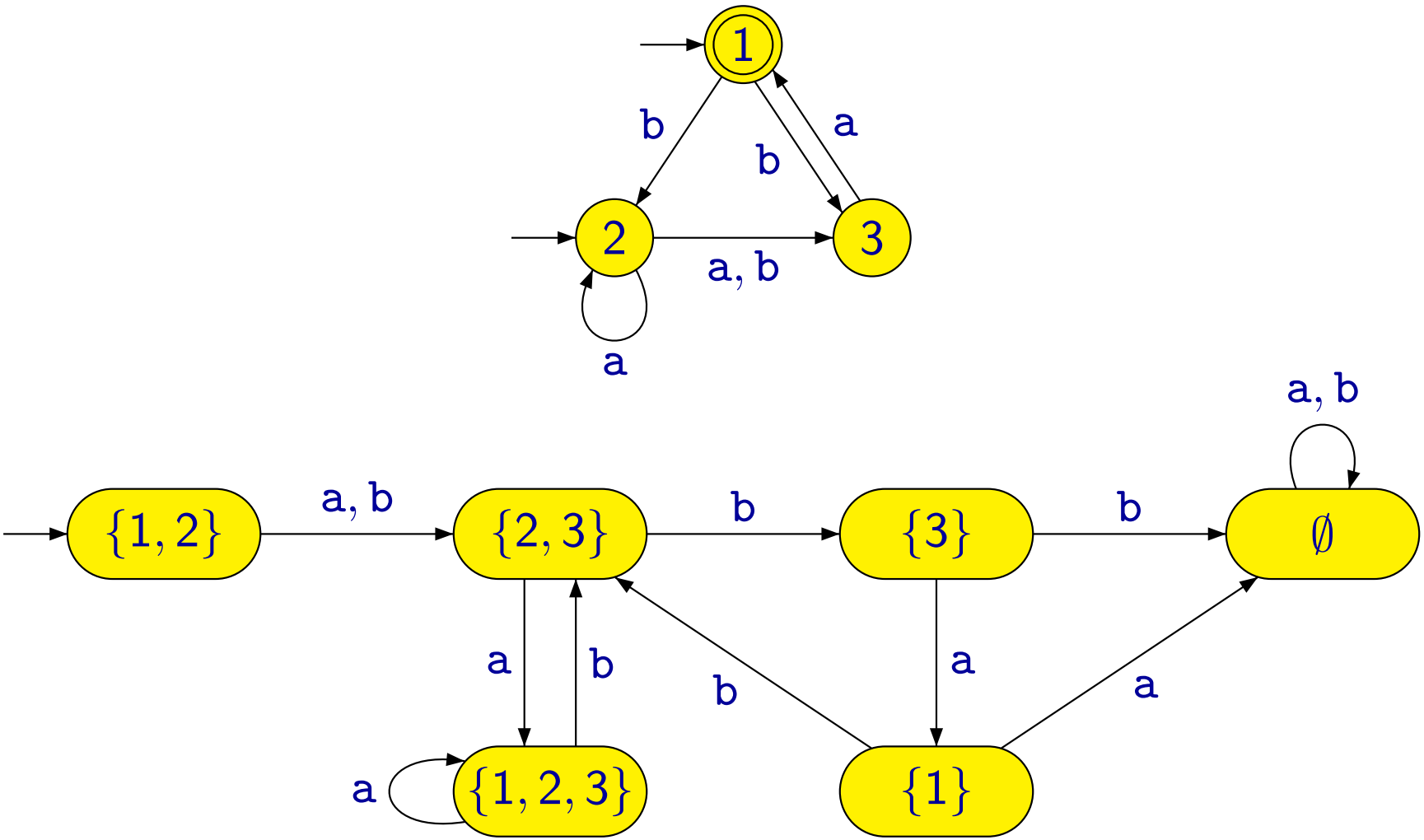


Převod NKA na DKA

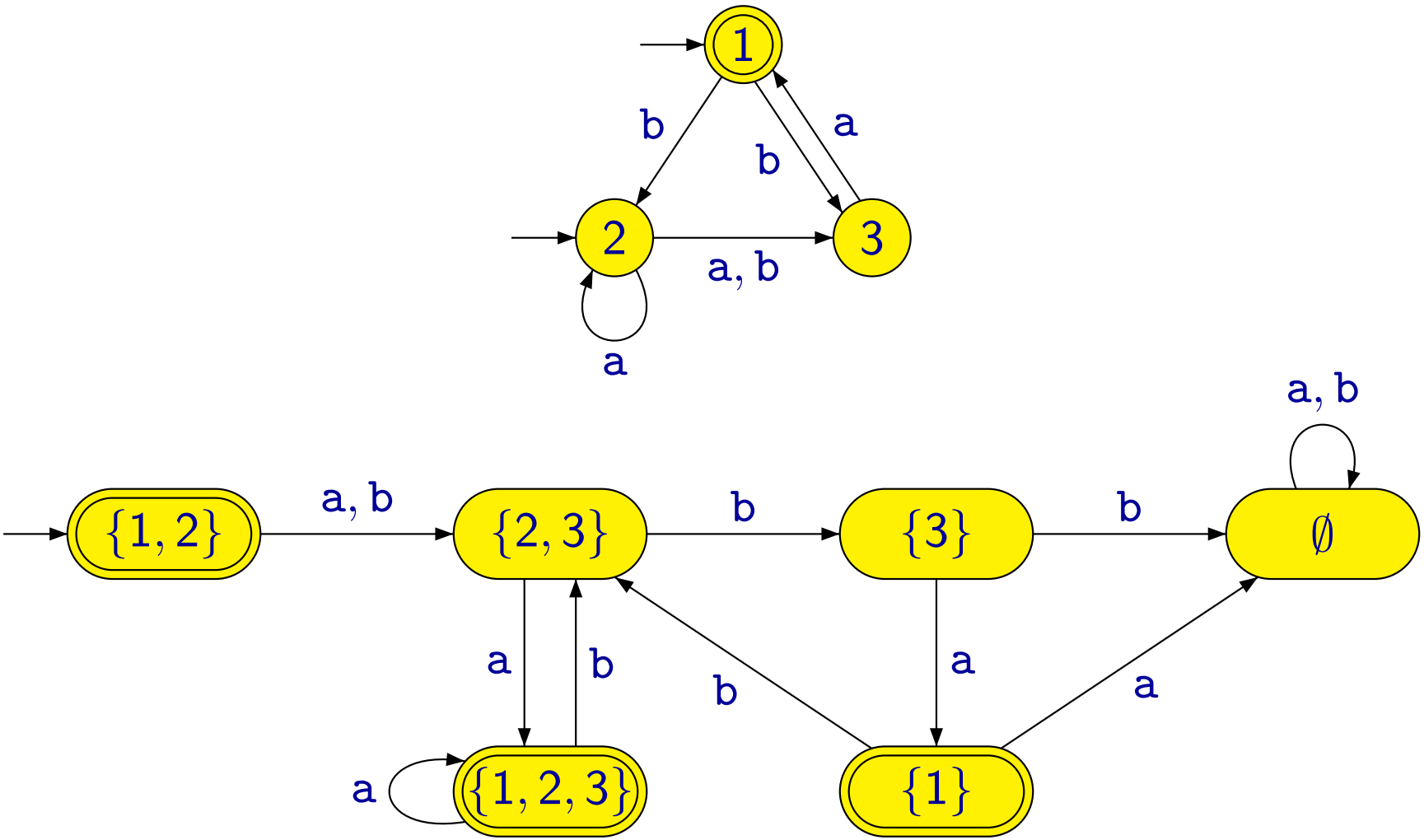




Převod NKA na DKA



Převod NKA na DKA



Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	
$\{2,3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	
$\leftarrow \{1,2,3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$		
$\{3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	
$\{3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	
$\leftarrow \{1\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$		
$\emptyset$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	–	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	–

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	
$\emptyset$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	–	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	–

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	$\{2,3\}$
$\emptyset$		

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	–	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	–

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	$\{2,3\}$
$\emptyset$	$\emptyset$	$\emptyset$

Převod NKA na DKA

	a	b
$\leftrightarrow 1$	–	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	–

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	$\{2,3\}$
$\emptyset$	$\emptyset$	$\emptyset$

	a	b
$\leftrightarrow 1$	2	2
2	3	4
$\leftarrow 3$	3	2
4	5	6
$\leftarrow 5$	6	2
6	6	6

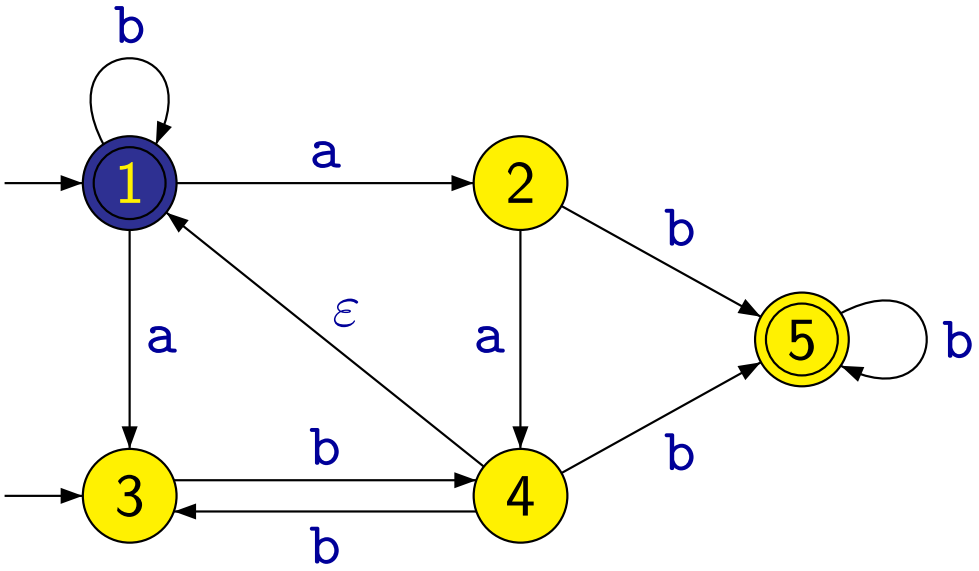
Převod NKA na DKA

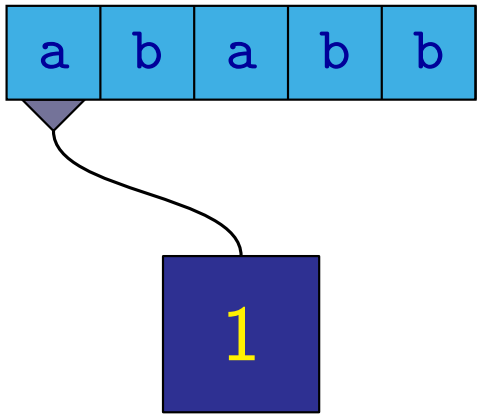
Poznámka: Při převodu nedeterministického automatu, který má n stavů, může mít výsledný deterministický automat až 2^n stavů.

Například při převodu automatu, který má 20 stavů, může vzniknout automat, který má $2^{20} = 1048576$ stavů.

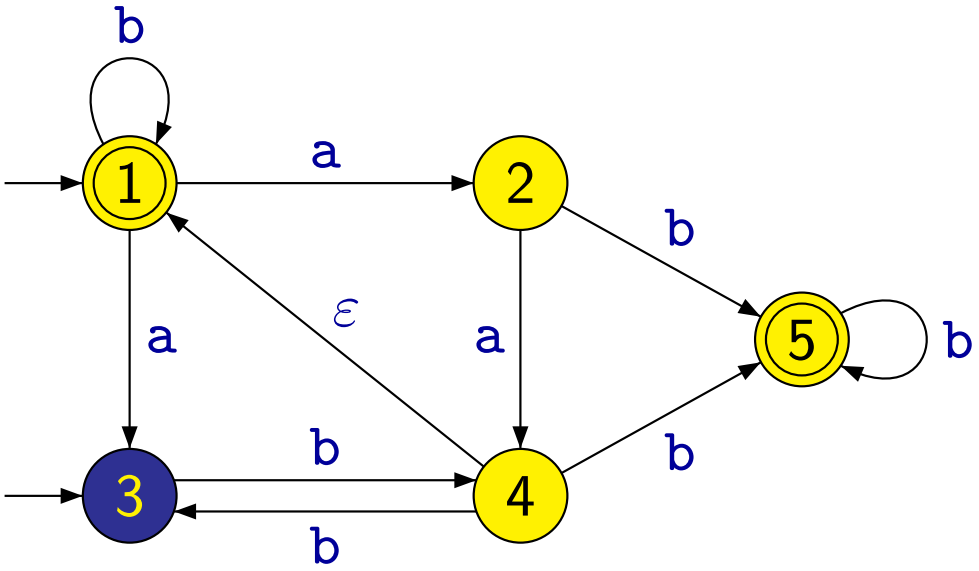
Často má sice výsledný automat podstatně méně než 2^n stavů, nicméně tyto nejhorší případy občas nastávají.

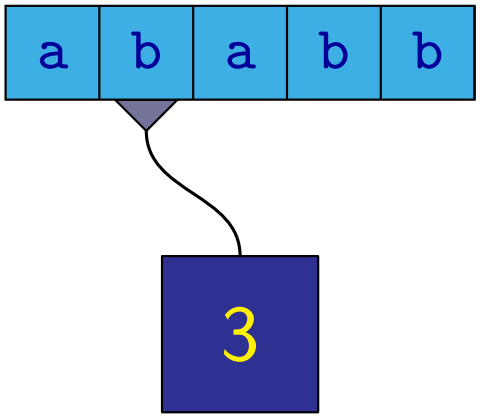
Zobecněný nedeterministický konečný automat





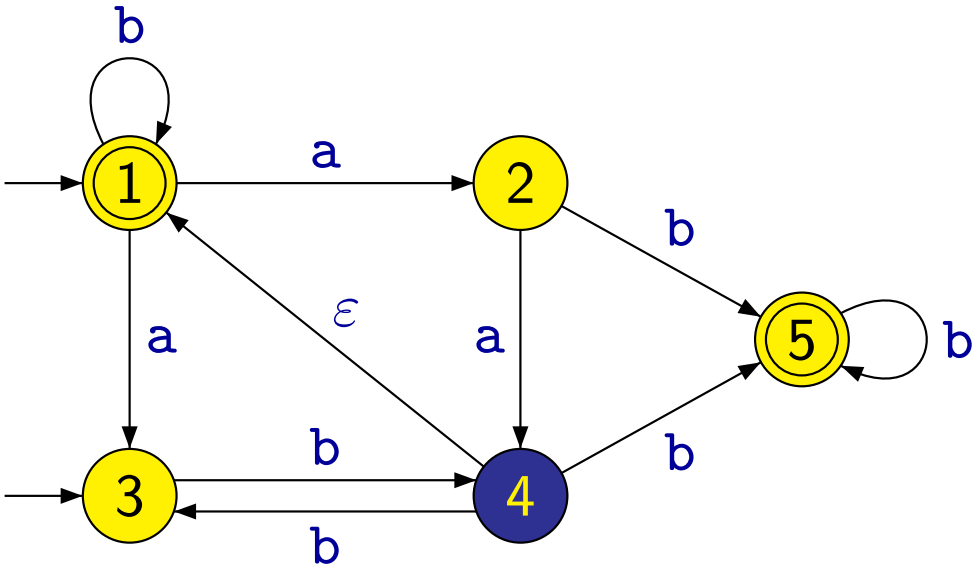
Zobecněný nedeterministický konečný automat

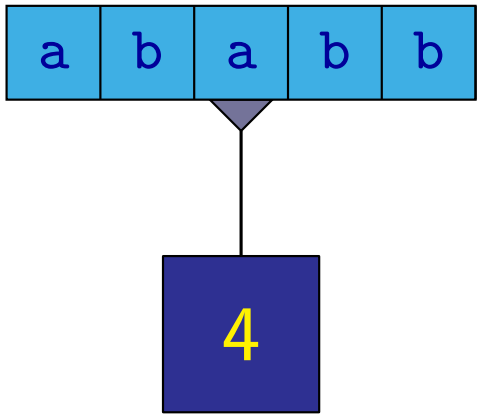




$1 \xrightarrow{a} 3$

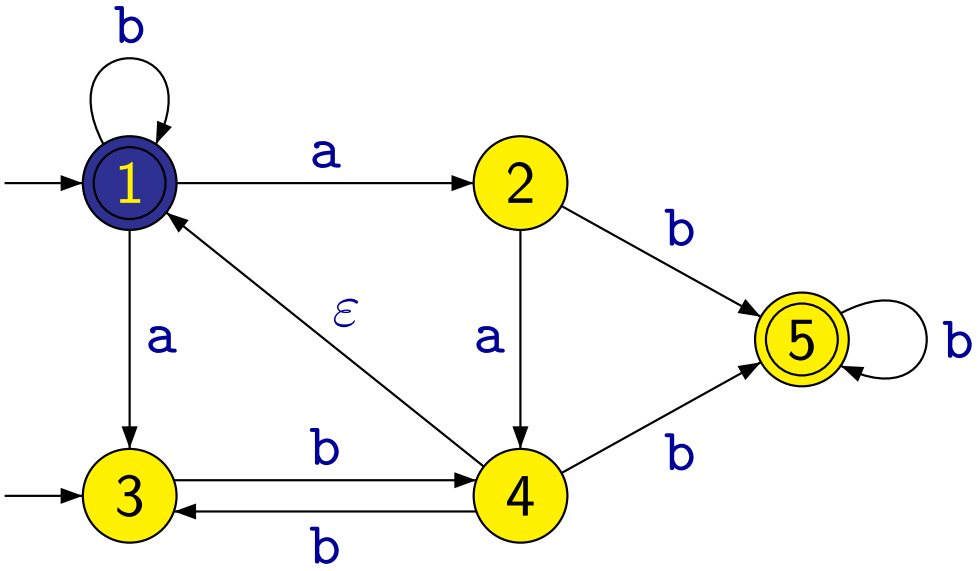
Zobecněný nedeterministický konečný automat

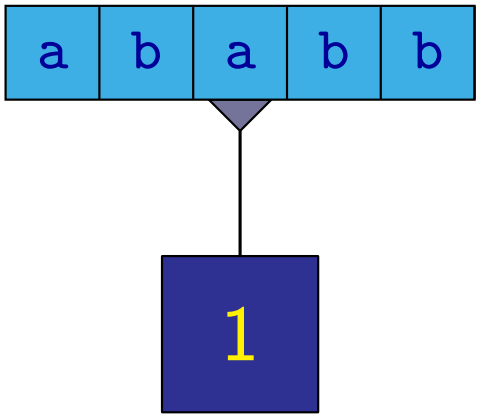




$1 \xrightarrow{a} 3 \xrightarrow{b} 4$

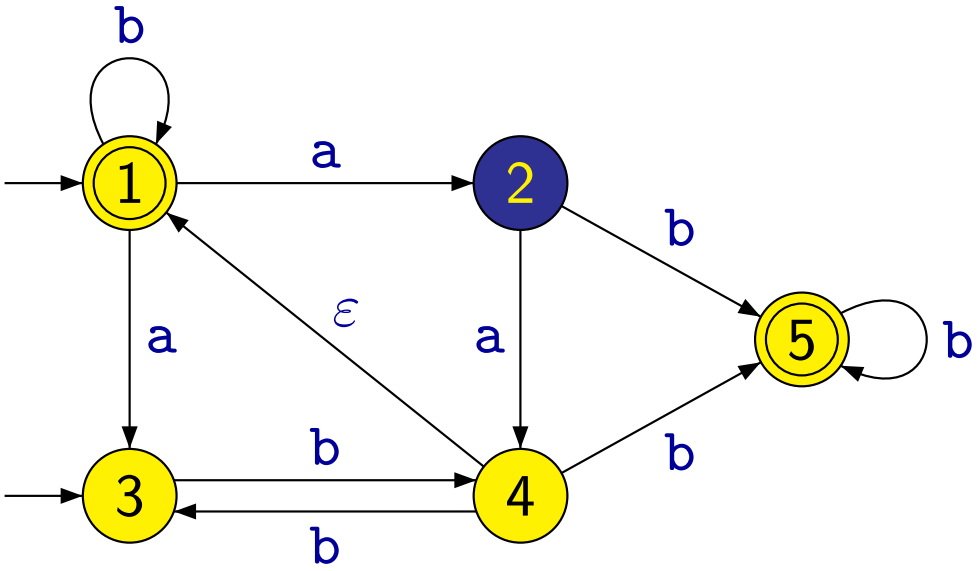
Zobecněný nedeterministický konečný automat

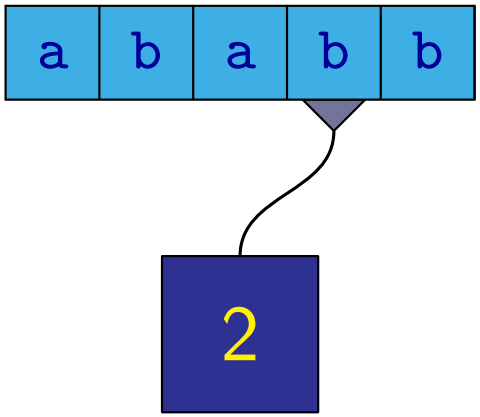




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{\varepsilon} 1$$

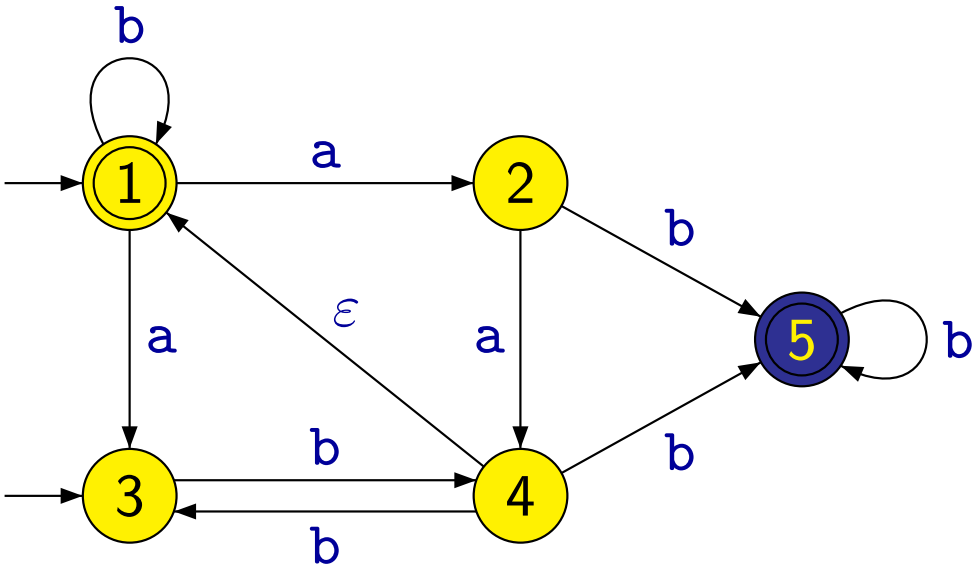
Zobecněný nedeterministický konečný automat

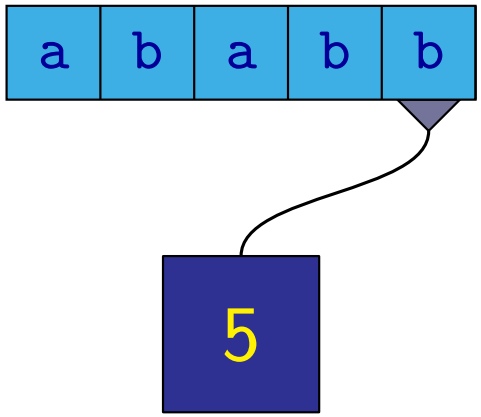




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{\varepsilon} 1 \xrightarrow{a} 2$$

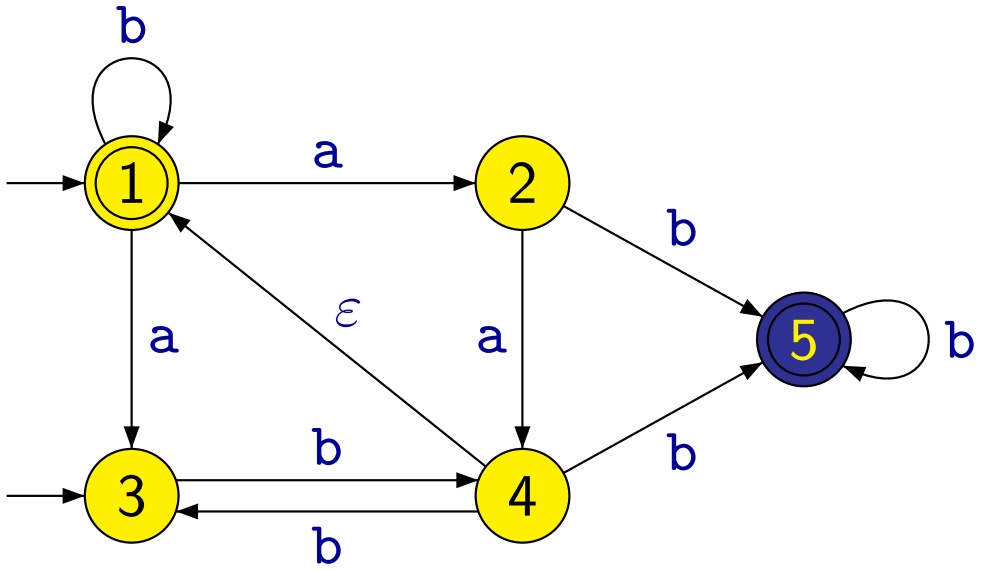
Zobecněný nedeterministický konečný automat

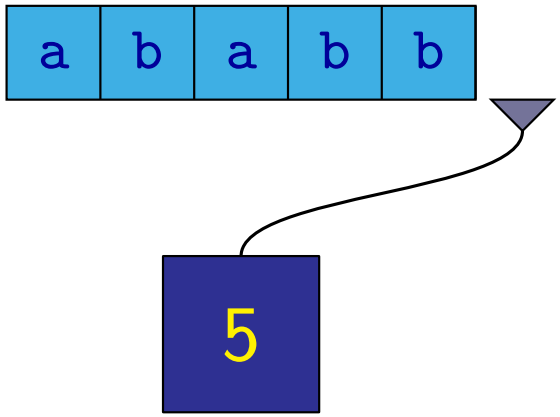




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{\varepsilon} 1 \xrightarrow{a} 2 \xrightarrow{b} 5$$

Zobecněný nedeterministický konečný automat





$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{\varepsilon} 1 \xrightarrow{a} 2 \xrightarrow{b} 5 \xrightarrow{b} 5$$

Zobecněný nedeterministický konečný automat

Oproti nedeterministickému konečnému automatu má **zobecněný nedeterministický konečný automat** tzv. ε-**přechody**, tj. přechody označené symbolem ε.

Při provádění ε-přechodu se mění pouze stav řídící jednotky, ale hlava na pásce se neposouvá.

Poznámka: Výpočty zobecněného nedeterministického automatu mohou být libovolně dlouhé a dokonce i nekonečné (pokud graf obsahuje cyklus tvořený ε-přechody) bez ohledu na délku slova na pásce.

Zobecněný nedeterministický konečný automat

Formálně je **zobecněný nedeterministický konečný automat** (**ZNKA**) definován jako pětice

$$(Q, \Sigma, \delta, I, F)$$

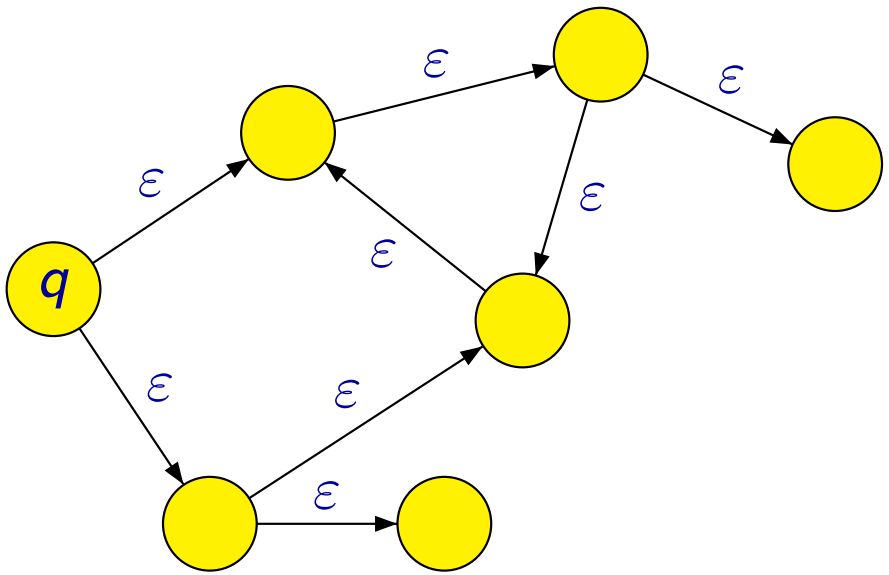
kde:

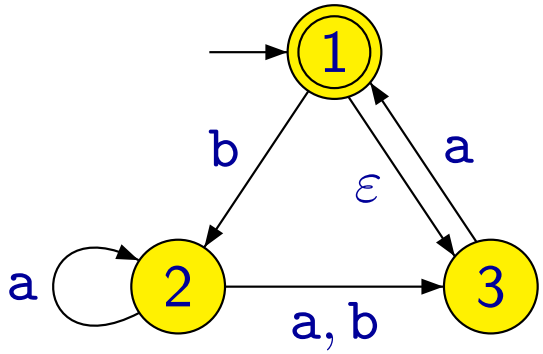
- Q je konečná množina **stavů**
- Σ je konečná **abeceda**
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ je **přechodová funkce**
- $I \subseteq Q$ je množina **počátečních stavů**
- $F \subseteq Q$ je množina **přijímajících stavů**

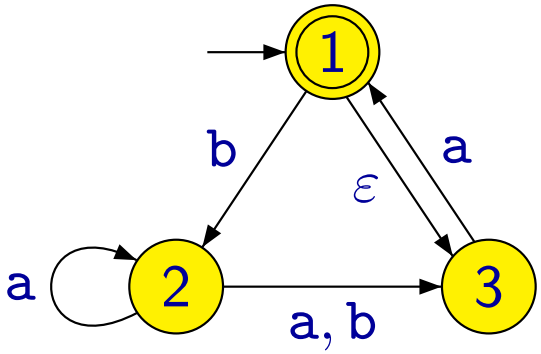
Poznámka: Na NKA můžeme nahlížet jako na speciální případ ZNKA, kde $\delta(q, \varepsilon) = \emptyset$ pro všechna $q \in Q$.

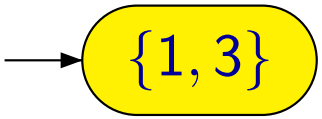
Převod na deterministický konečný automat

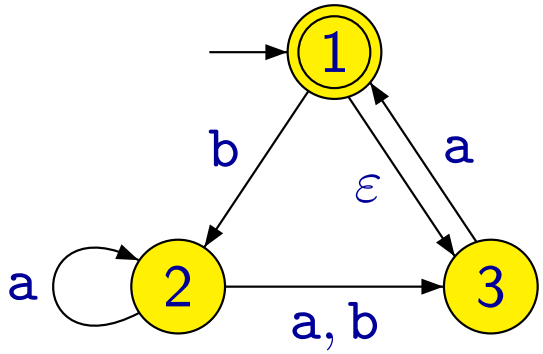
Zobecněný nedeterministický konečný automat je možné převést na deterministický podobnou konstrukcí jako nedeterministický konečný automat, s tím rozdílem, že do množin stavů musíme vždy přidat navíc i všechny stavy dosažitelné z již přidaných stavů nějakou sekvencí ε-přechodů.

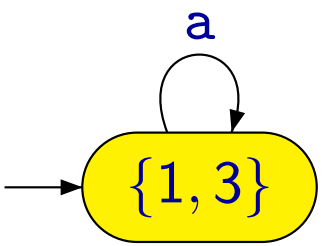


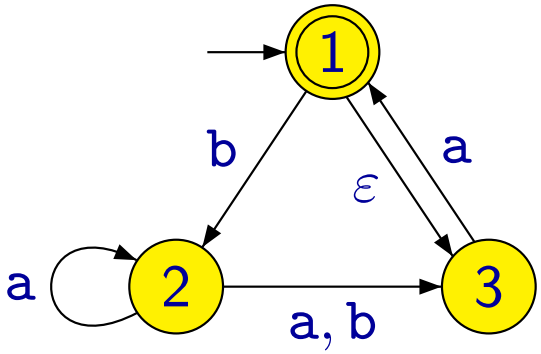

1
b
a
ε
a
2
3
a, b

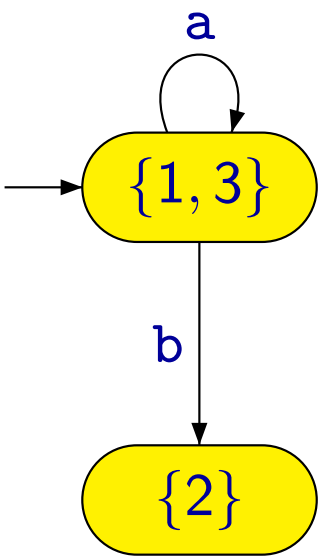

1
b
a
ε
a
2
3
a, b

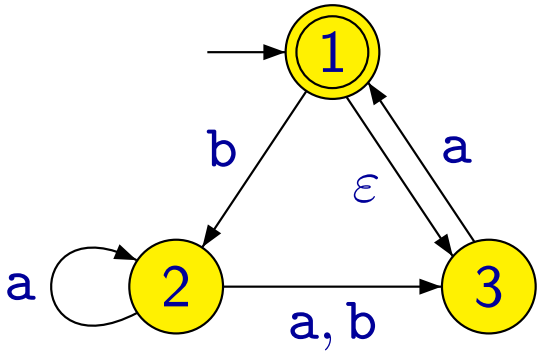

{1, 3}

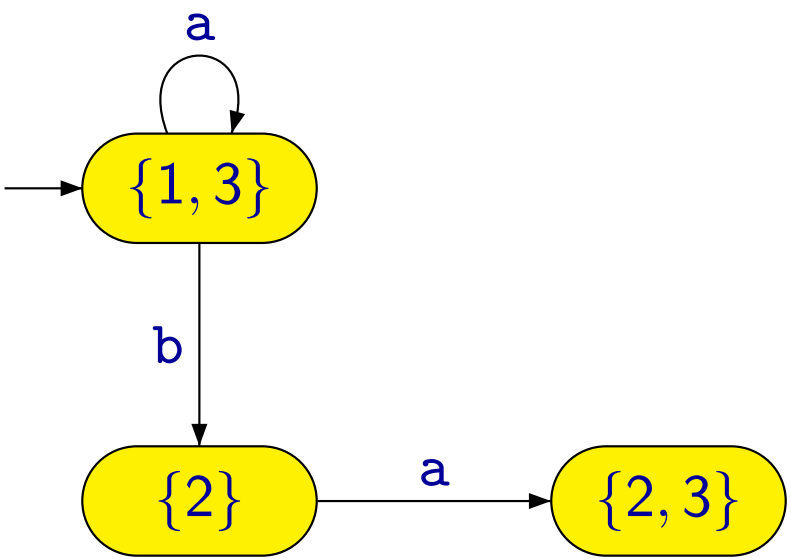

1
b
a
ε
a
2
3
a, b

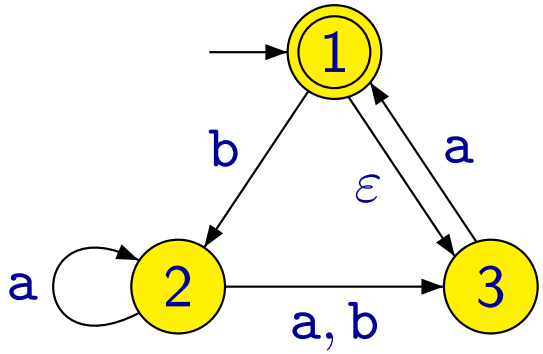

a
{1, 3}

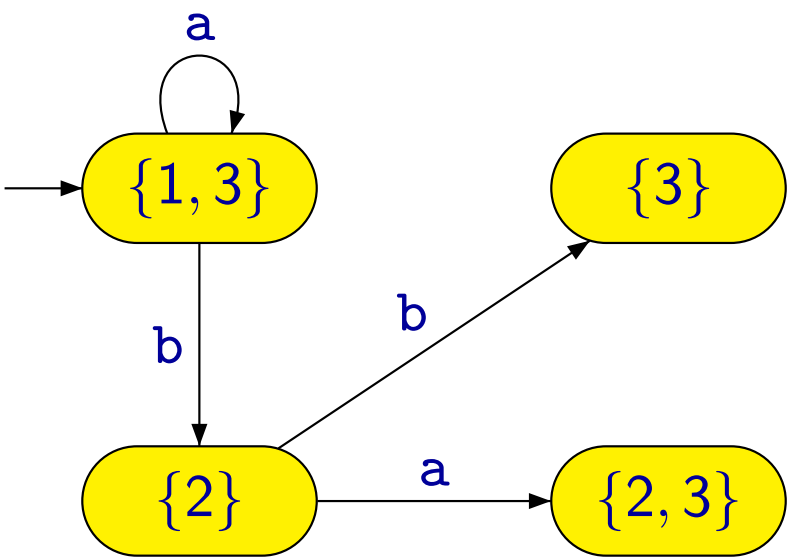

1
b
a
ε
a
2
3
a, b

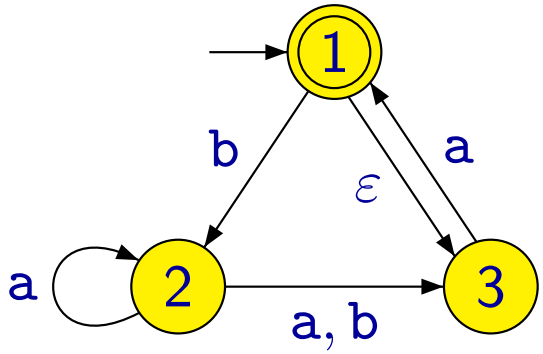

a
{1, 3}
b
{2}

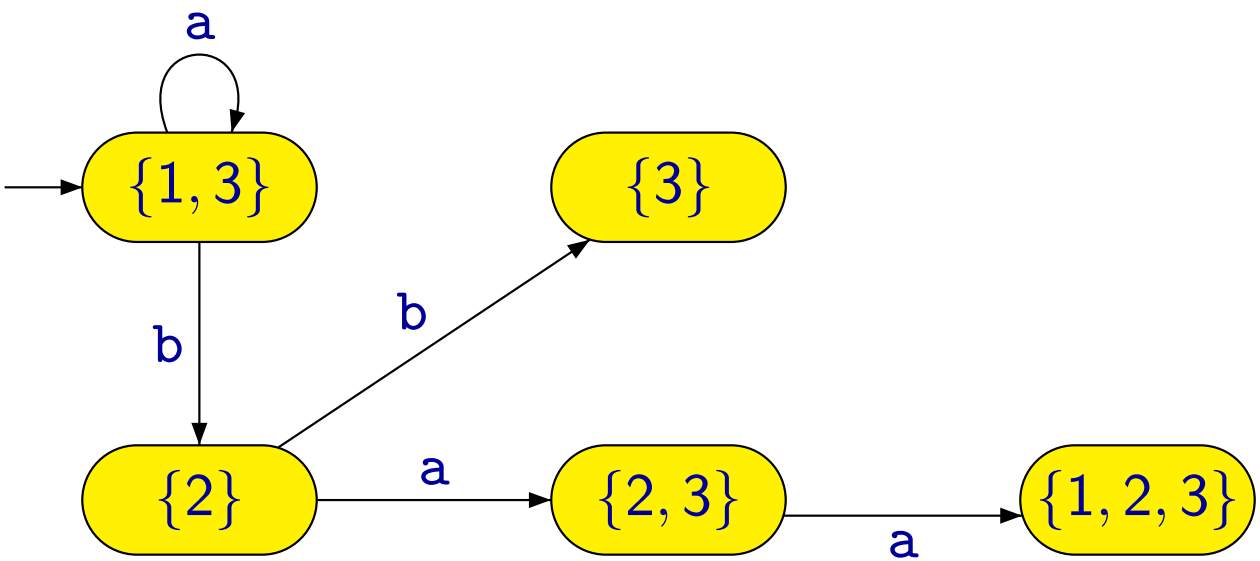

1
b
a
ε
a
2
3
a, b

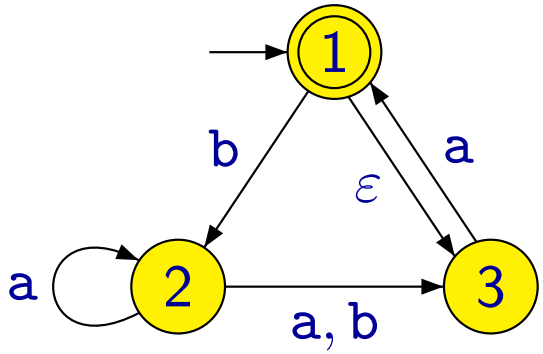

a
{1, 3}
b
{2}
a
{2, 3}

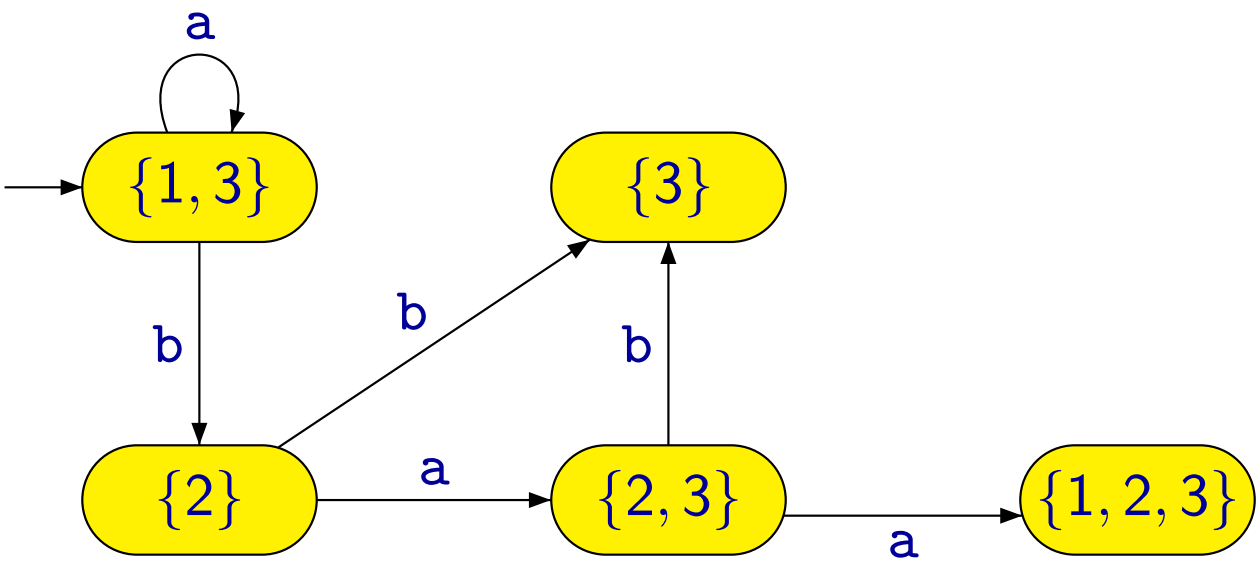

1
b
a
ε
a
2
3
a, b

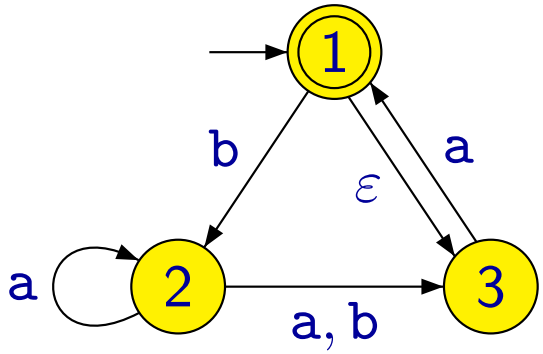

a
{1, 3}
{3}
b
b
{2}
a
{2, 3}

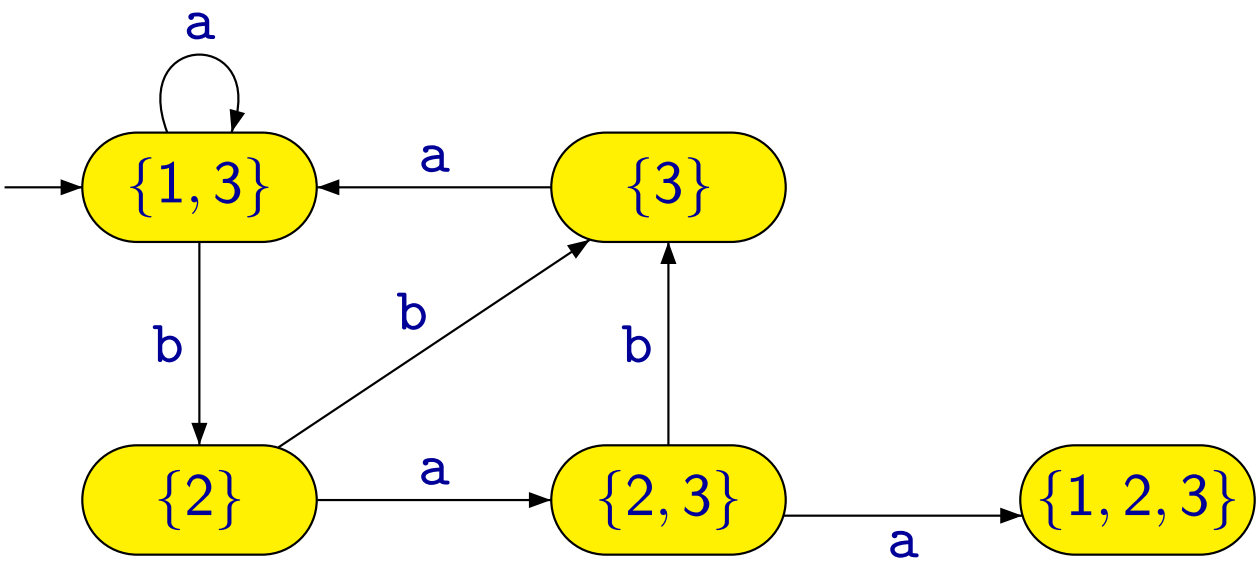

1
b
a
ε
a
2
3
a, b

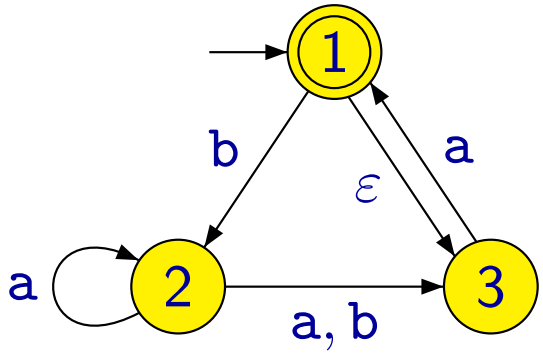

a
{1, 3}
{3}
b
b
{2}
a
{2, 3}
{1, 2, 3}
a

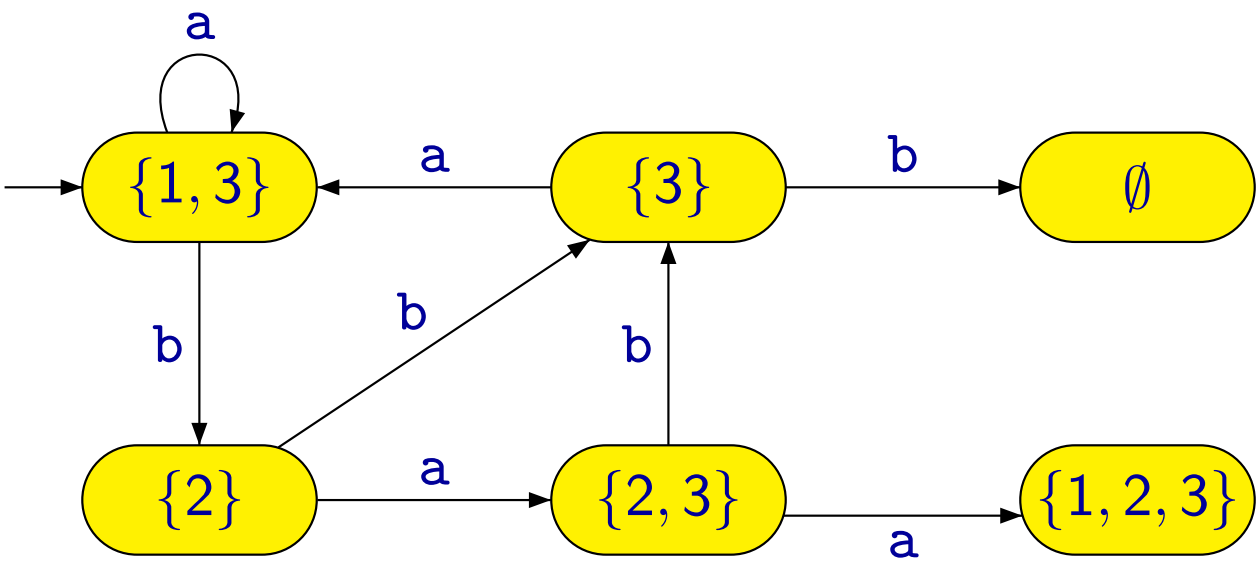

1
b
a
ε
a
2
3
a, b

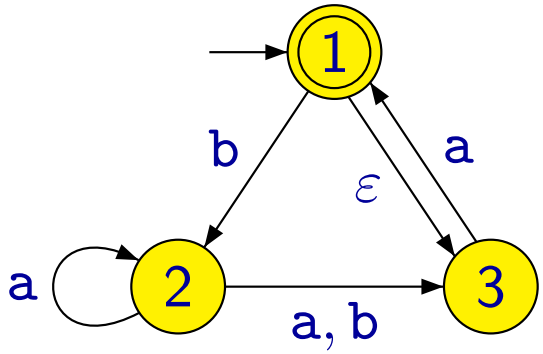

a
{1, 3}
{3}
b
b
b
{2}
a
{2, 3}
{1, 2, 3}
a

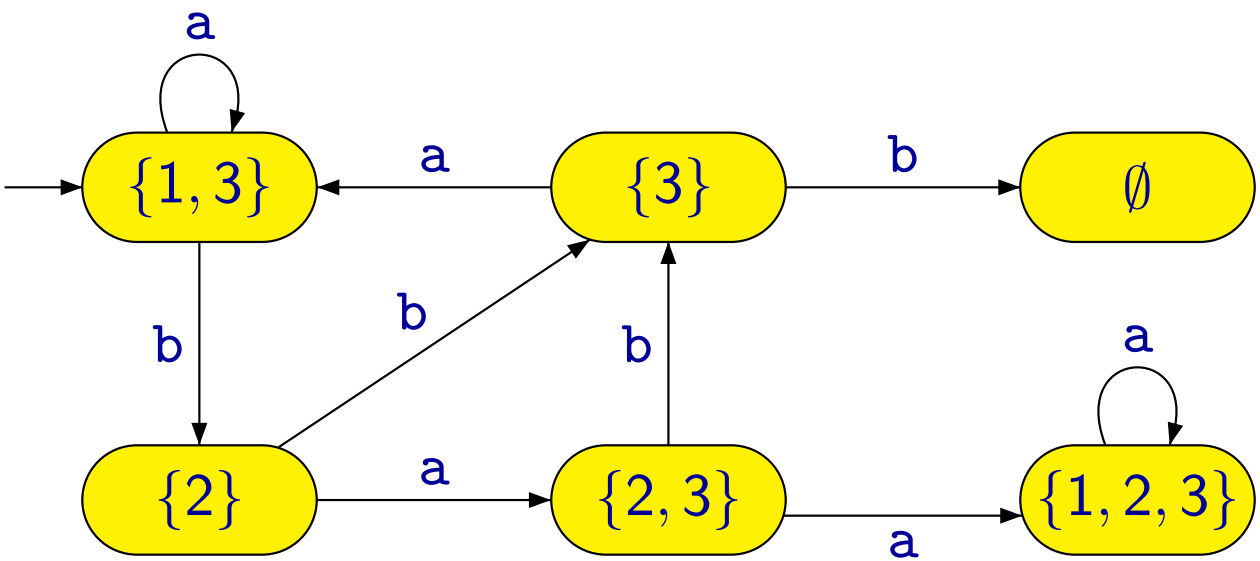

1
b
a
ε
a
2
3
a, b

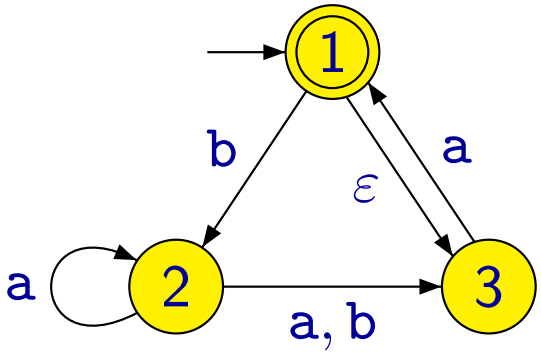

a
{1, 3}
a
{3}
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

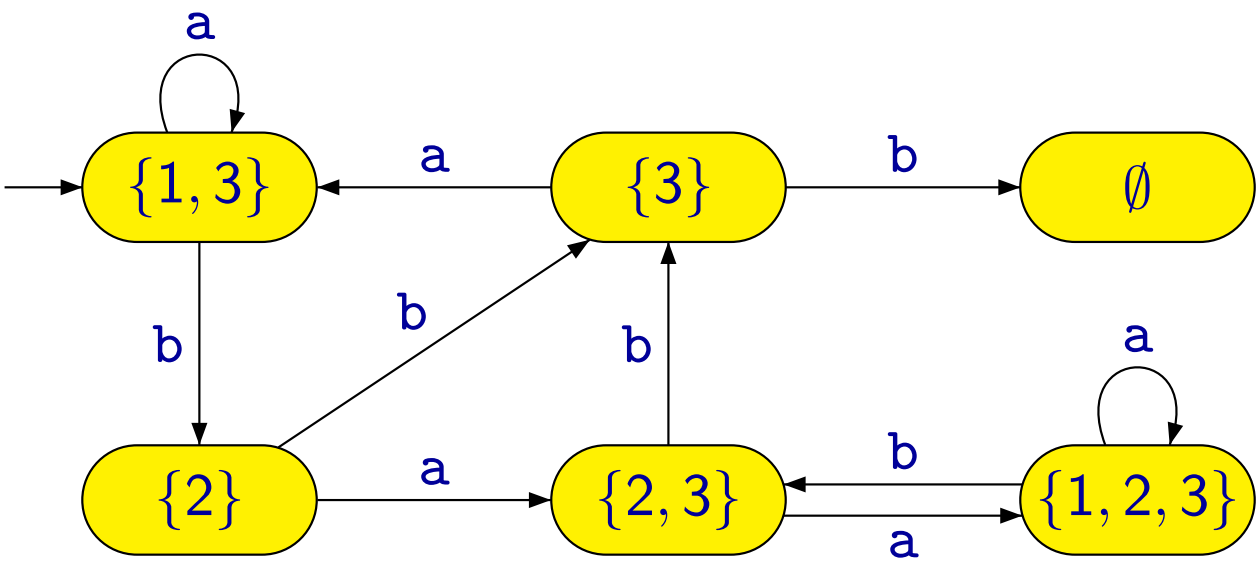

1
b
a
ε
a
2
3
a, b

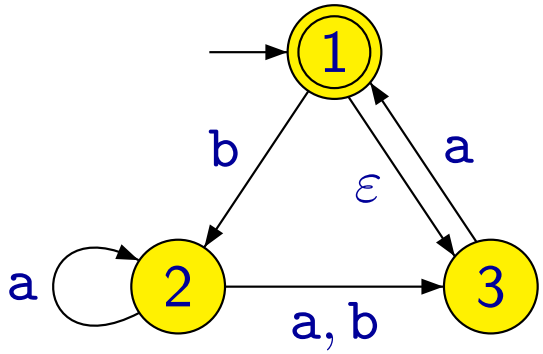

a
{1, 3}
a
{3}
b
∅
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

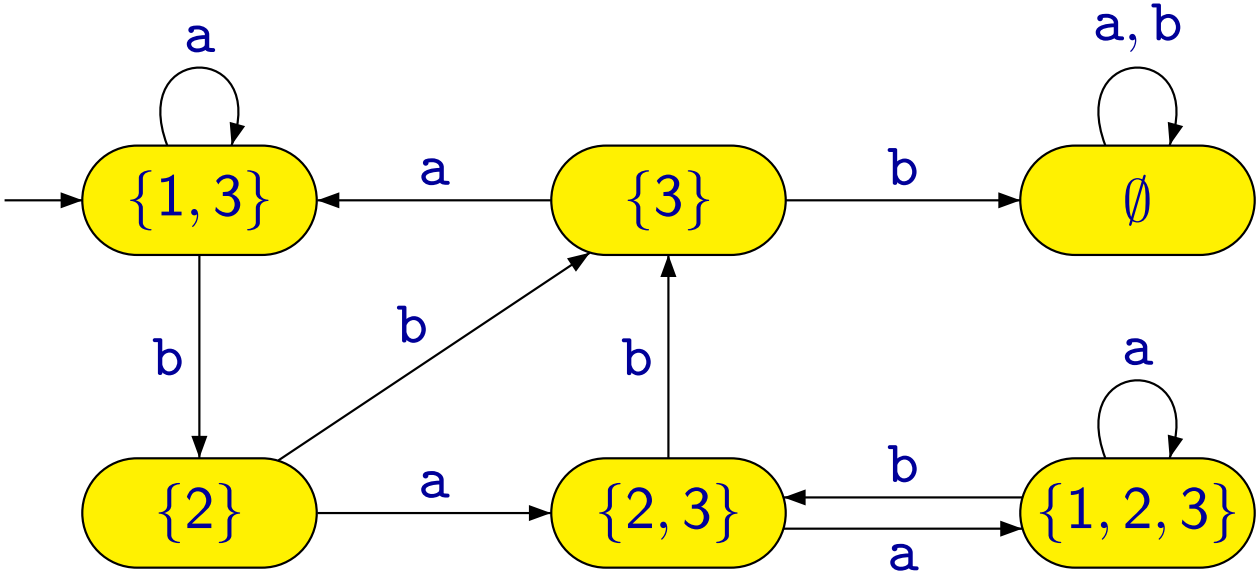

1
b
a
ε
a
2
3
a, b

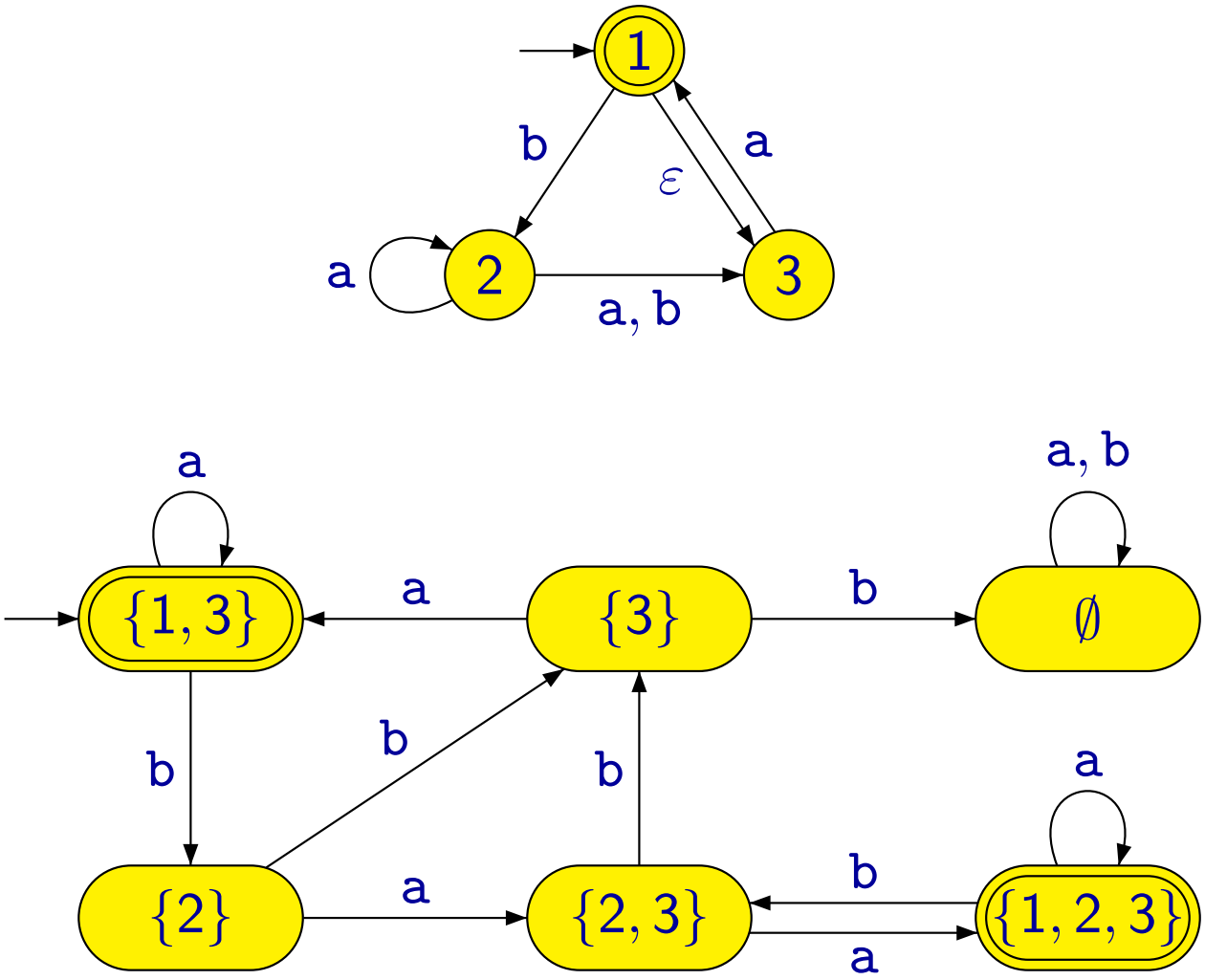

a
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
{1, 2, 3}
a


1
b
a
ε
a
2
3
a, b


a
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
b
{1, 2, 3}
a


1
b
a
ε
a
2
3
a, b


a
a, b
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
b
{1, 2, 3}
a


1
b
a
ε
a
2
3
a, b
a
a, b
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
b
{1, 2, 3}
a

Převod ZNKA na DKA

Předtím, než formálně popíšeme převod ZNKA na DKA, zaveďme si několik pomocných definic.

Předpokládejme nějaký daný ZNKA $A = (Q, \Sigma, \delta, I, F)$.

Definujme funkci $\hat{\delta} : \mathcal{P}(Q) \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ tak, že pro $K \subseteq Q$ a $a \in \Sigma \cup \{\varepsilon\}$ je

$$\hat{\delta}(K, a) = \bigcup_{q \in K} \delta(q, a)$$

Převod ZNKA na DKA

Pro $K \subseteq Q$ označme $Cl_\varepsilon(K)$ množinu všech stavů dosažitelných ze stavů z množiny K nějakou libovolnou sekvencí ε-přechodů.

To znamená, že funkce $Cl_\varepsilon : \mathcal{P}(Q) \to \mathcal{P}(Q)$ je definována tak, že pro $K \subseteq Q$ je $Cl_\varepsilon(K)$ nejmenší (vzledem k inkluzi) množina splňující následující dvě podmínky:

- $K \subseteq Cl_\varepsilon(K)$
- Pro každé $q \in Cl_\varepsilon(K)$ platí, že $\delta(q, \varepsilon) \subseteq Cl_\varepsilon(K)$.

Poznámka: Všimněme si, že pro libovolné K je $Cl_\varepsilon(Cl_\varepsilon(K)) = Cl_\varepsilon(K)$.

Všimněme si také, že v případě NKA (kde $\delta(q, \varepsilon) = \emptyset$ pro každé $q \in Q$) je $Cl_\varepsilon(K) = K$.

Převod ZNKA na DKA

K danému ZNKA $A = (Q, \Sigma, \delta, I, F)$ nyní můžeme sestrojit DKA $A' = (Q', \Sigma, \delta', q_0', F')$, kde:

- $Q' = \mathcal{P}(Q)$ ($K \in Q'$ tedy znamená, že $K \subseteq Q$)
- $\delta' : Q' \times \Sigma \to Q'$ je definová tak, že pro $K \in Q'$ a $a \in \Sigma$ je

$$\delta'(K, a) = Cl_\varepsilon(\hat{\delta}(Cl_\varepsilon(K), a))$$

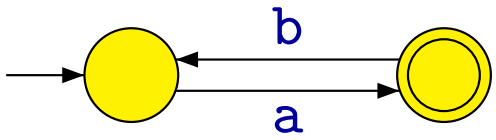
- $q_0' = Cl_\varepsilon(I)$
- $F' = \{K \in Q' \mid Cl_\varepsilon(K) \cap F \neq \emptyset\}$

Není težké ověřit, že $L(A) = L(A')$.

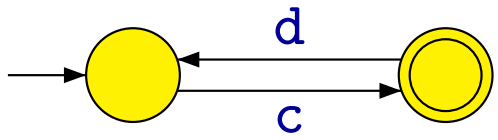
Zřetězení jazyků

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

A_1:



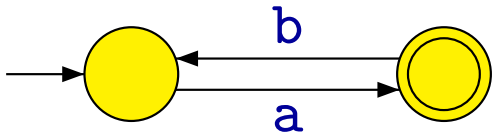
A_2:



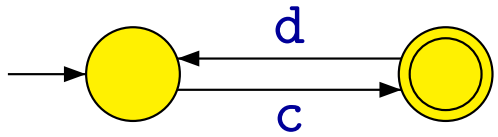
Zřetězení jazyků

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

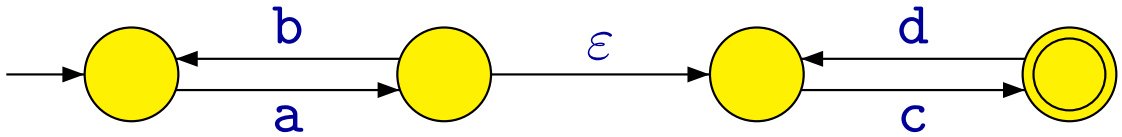
A_1:



A_2:



A:

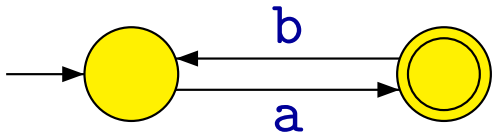


$L(A) = L(A_1) \cdot L(A_2)$

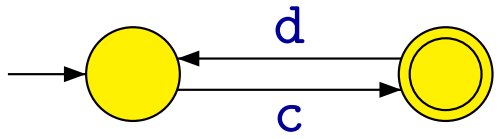
Zřetězení jazyků

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

A_1:

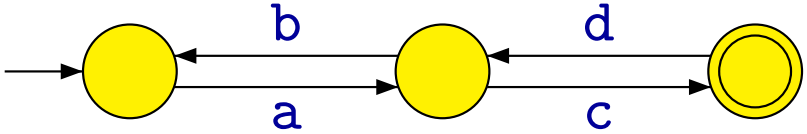


A_2:



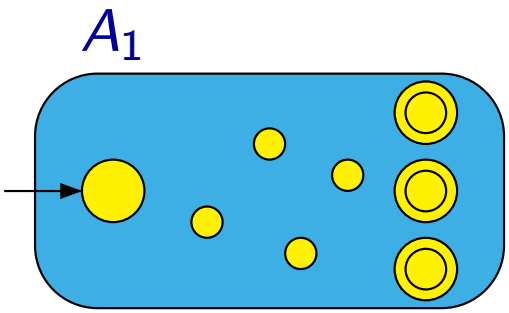
Chybná konstrukce:

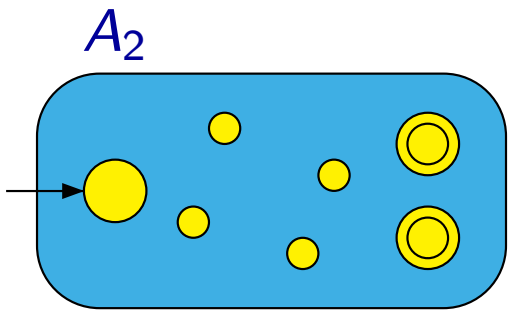
A:



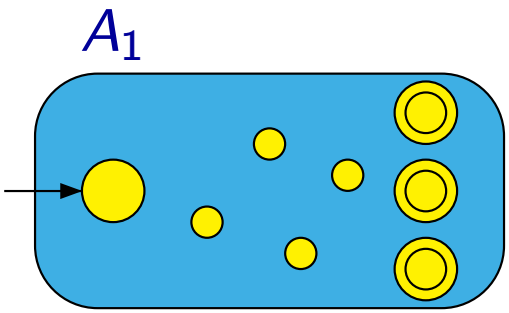
$\mathtt{acdbac} \in L(A)$, ale $\mathtt{acdbac} \notin L(A_1) \cdot L(A_2)$

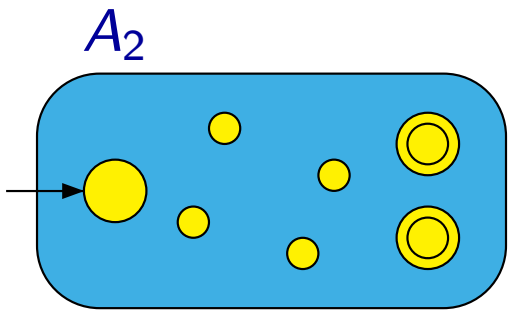
Zřetězení jazyků

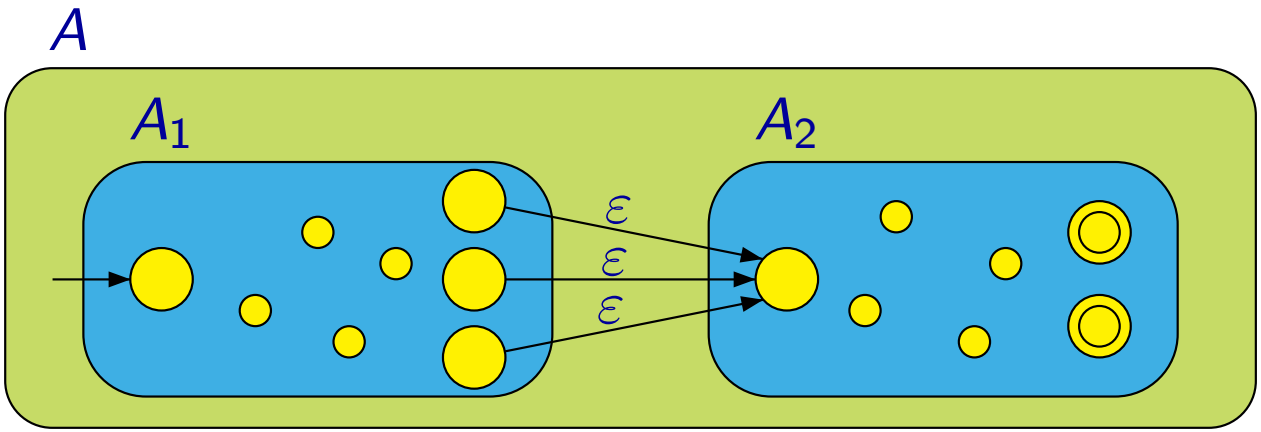




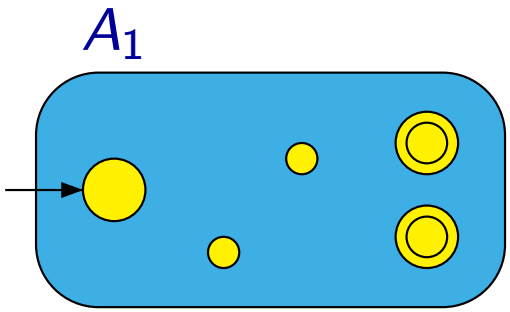
Zřetězení jazyků



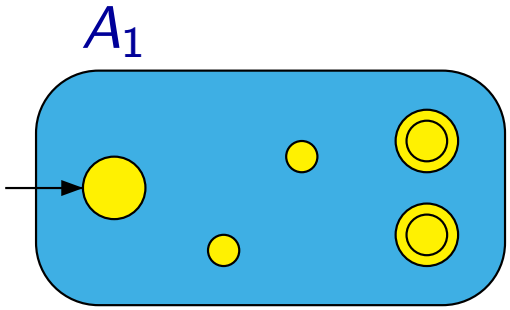


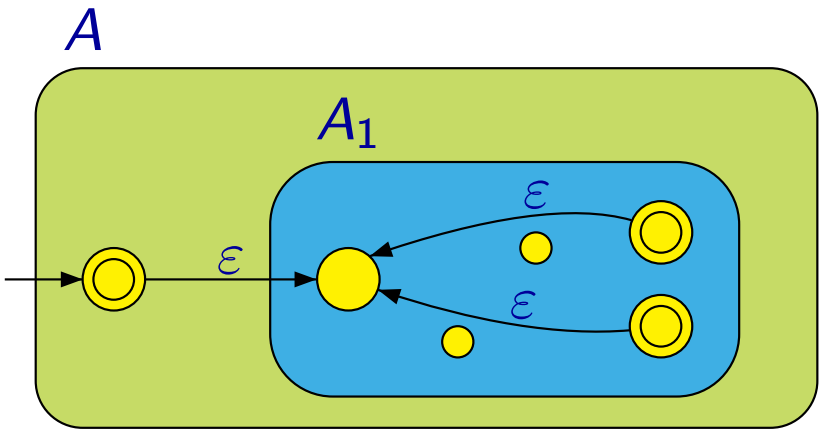


Iterace jazyka



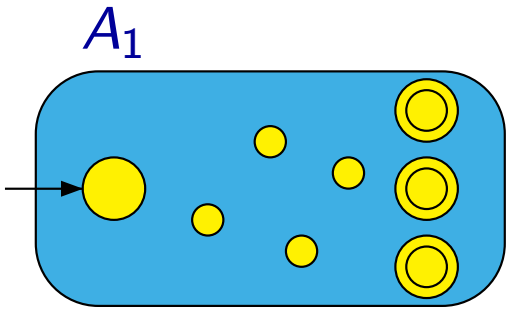
Iterace jazyka

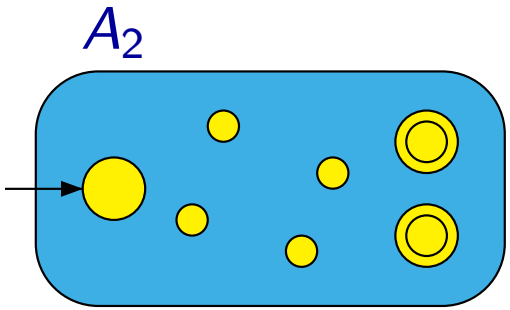




Sjednocení jazyků

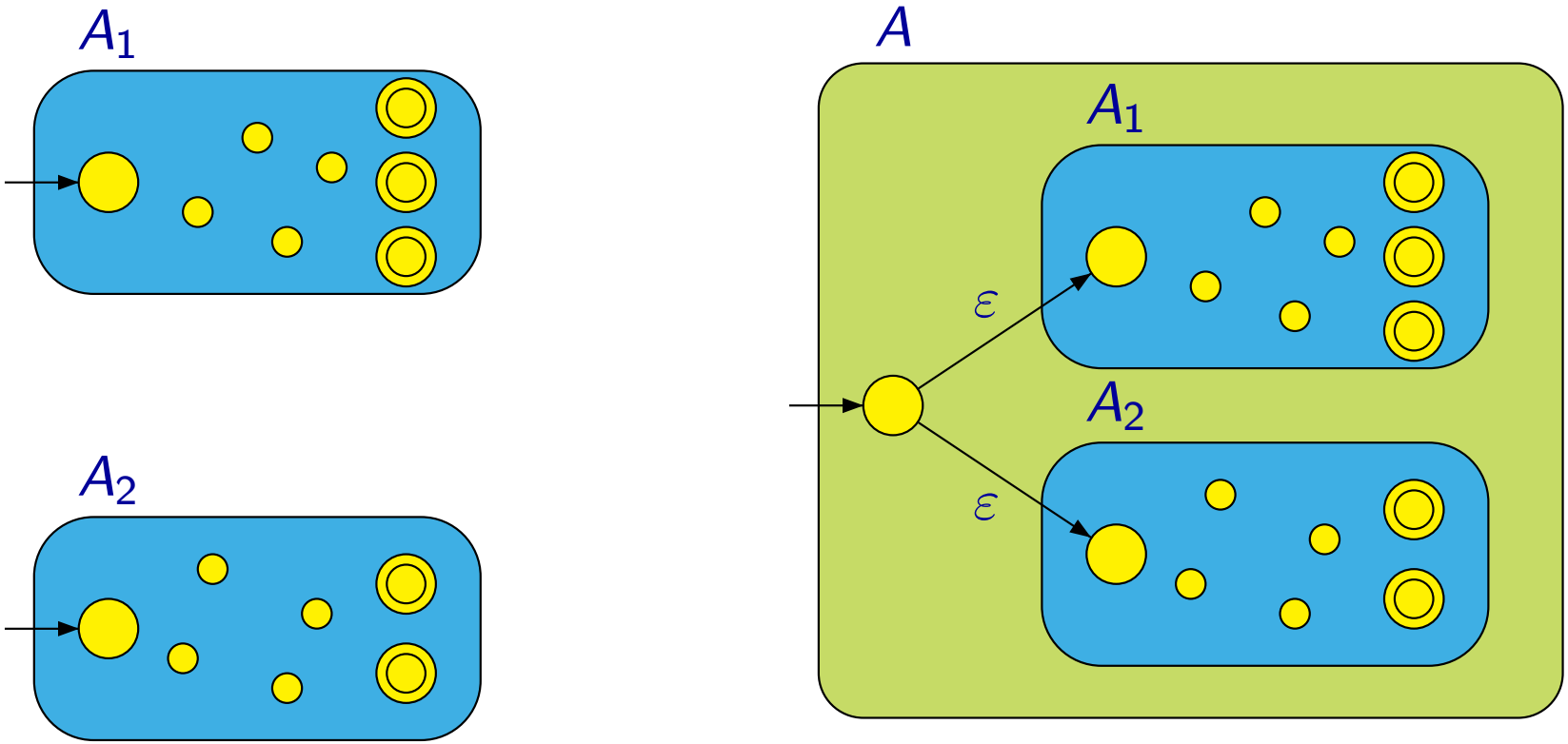
Alternativní konstrukce pro sjednocení jazyků:





Sjednocení jazyků

Alternativní konstrukce pro sjednocení jazyků:



Uzavřenost třídy regulárních jazyků

Množina (všech) regulárních jazyků je uzavřená vůči operacím:

- sjednocení
- průnik
- doplněk
- zřetězení
- iterace
- ...

Převod regulárního výrazu na konečný automat

Tvrzení

Každý jazyk, který je možné vyjádřit regulárním výrazem, je regulární (tj. rozpoznávaný nějakým konečným automatem).

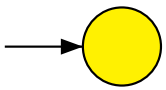
Důkaz: Stačí ukázat, jak k danému regulárnímu výrazu α zkonstruovat konečný automat, který rozpoznává jazyk $[\alpha]$.

Konstrukce je rekurzivní a postupuje podle struktury výrazu α:

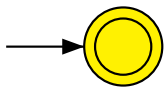
- Pokud je α elementární výraz (tj. $\emptyset$, ε nebo a):
  - Sestrojíme přímo odpovídající automat.
- Pokud je α tvaru $(\beta + \gamma)$, $(\beta \cdot \gamma)$ nebo (β^*):
  - Rekurzivně sestrojíme automaty rozpoznávající jazyky $[\beta]$ a $[\gamma]$.
  - Z nich sestrojíme automat rozpoznávající jazyk $[\alpha]$.

Převod regulárního výrazu na konečný automat

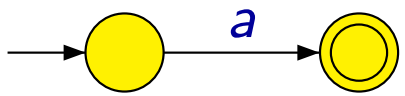
Automaty pro elementární výrazy:



$\emptyset$



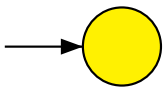
ε



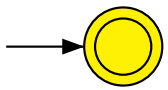
a

Převod regulárního výrazu na konečný automat

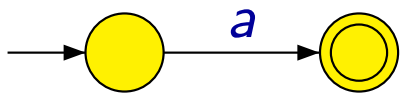
Automaty pro elementární výrazy:



$\emptyset$

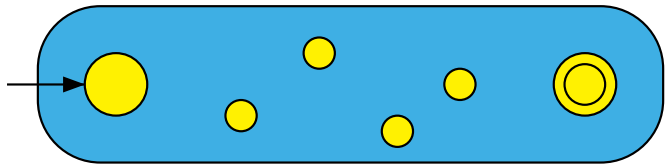


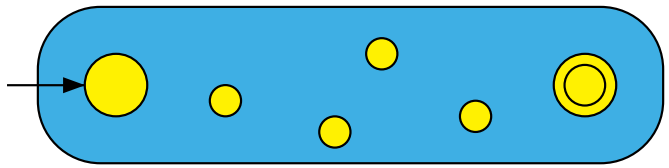
ε



a

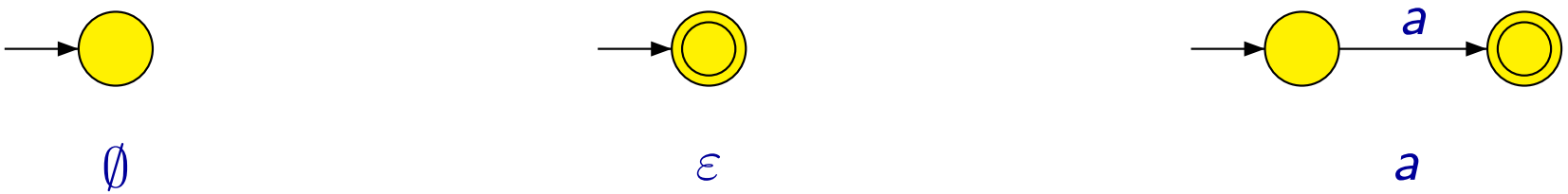
Konstrukce pro sjednocení:



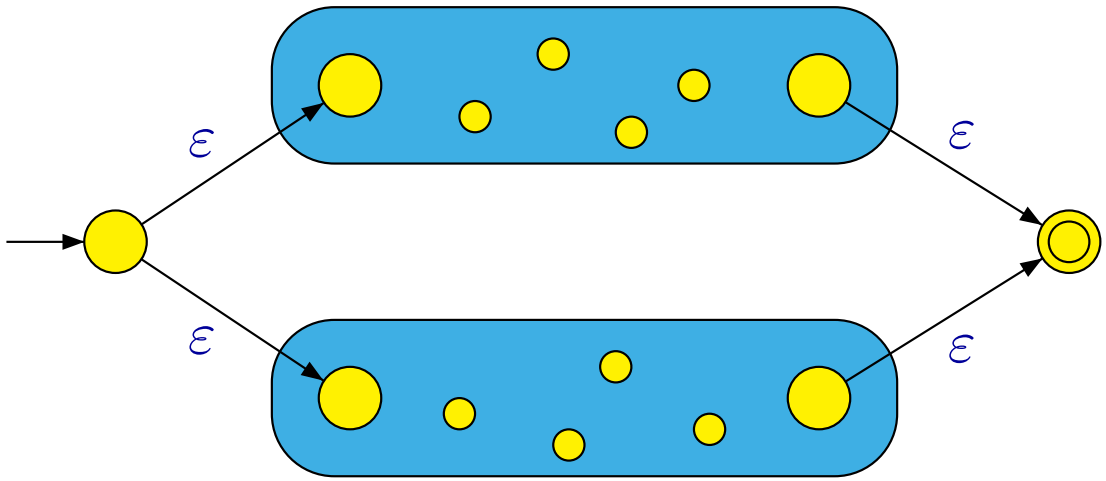


Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:

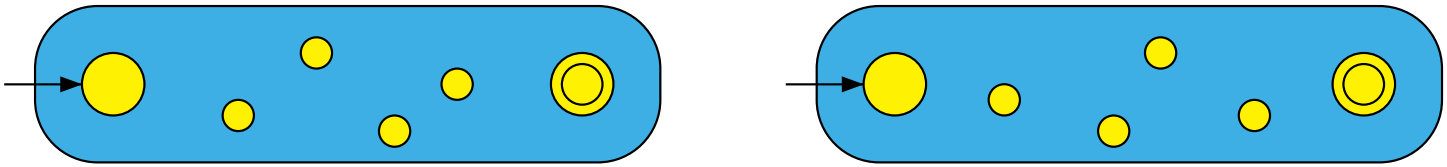


Konstrukce pro sjednocení:



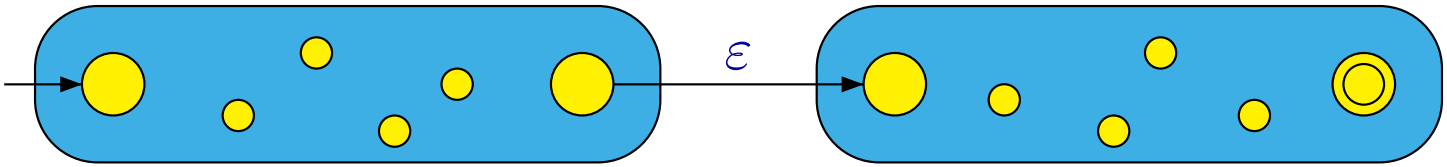
Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:



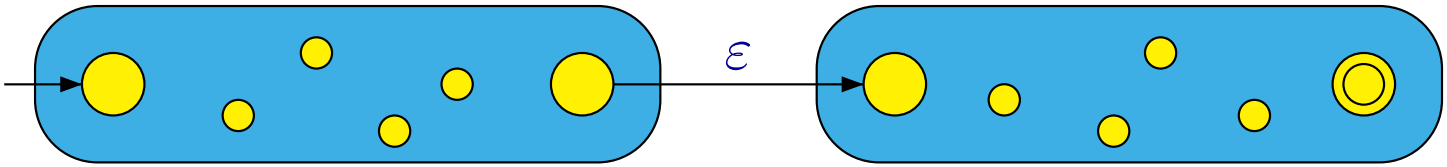
Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:

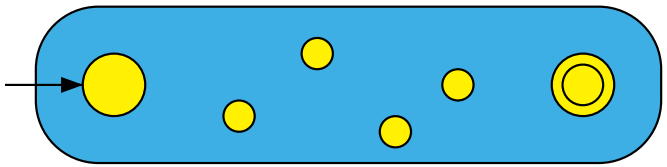


Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:

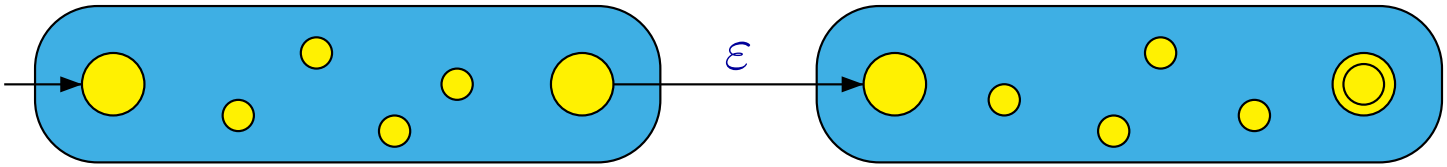


Konstrukce pro iteraci:

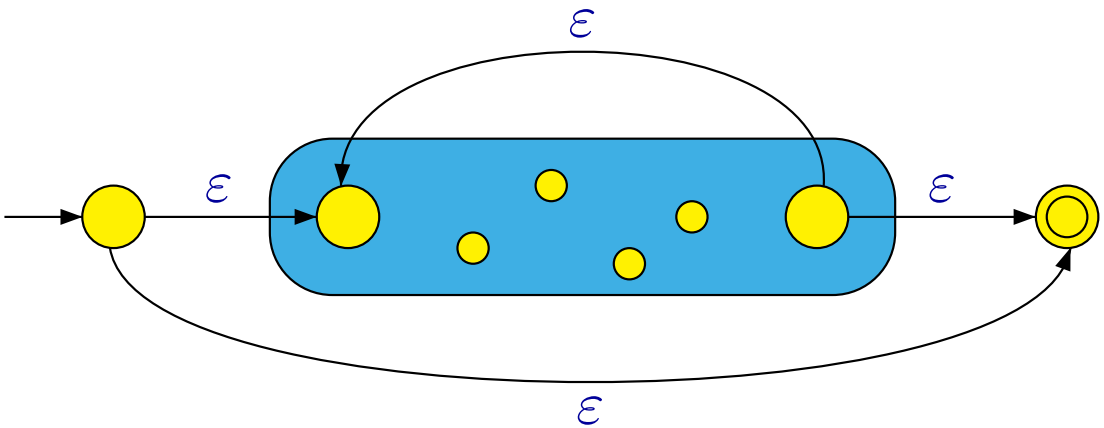


Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:



Konstrukce pro iteraci:

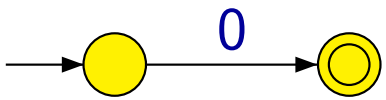


Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:

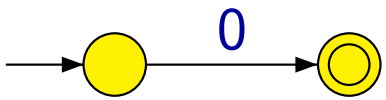
Převod regulárního výrazu na konečný automat

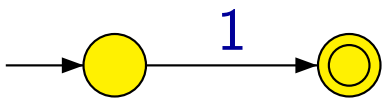
Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



Převod regulárního výrazu na konečný automat

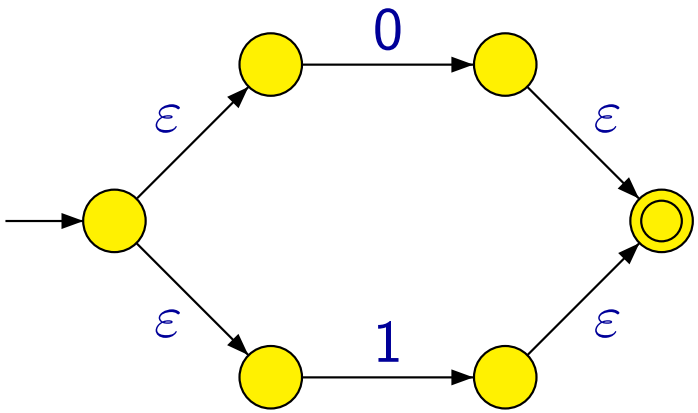
Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:





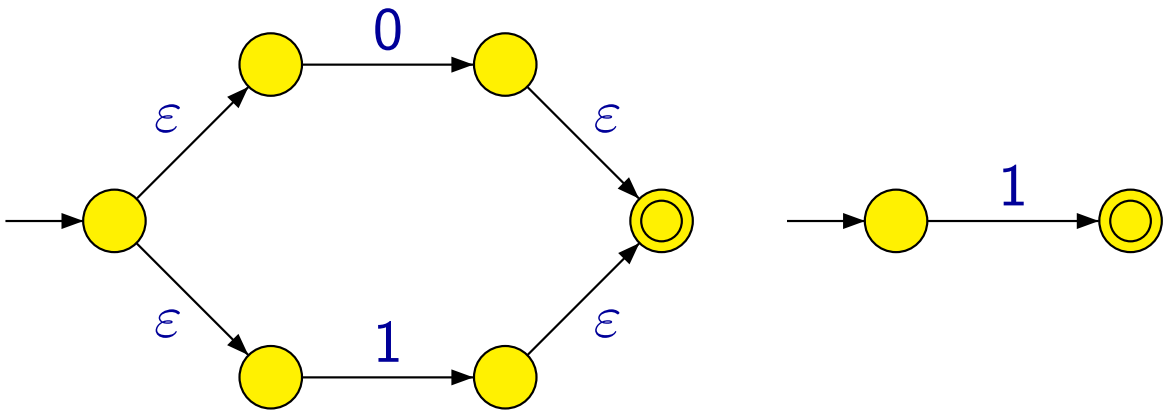
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



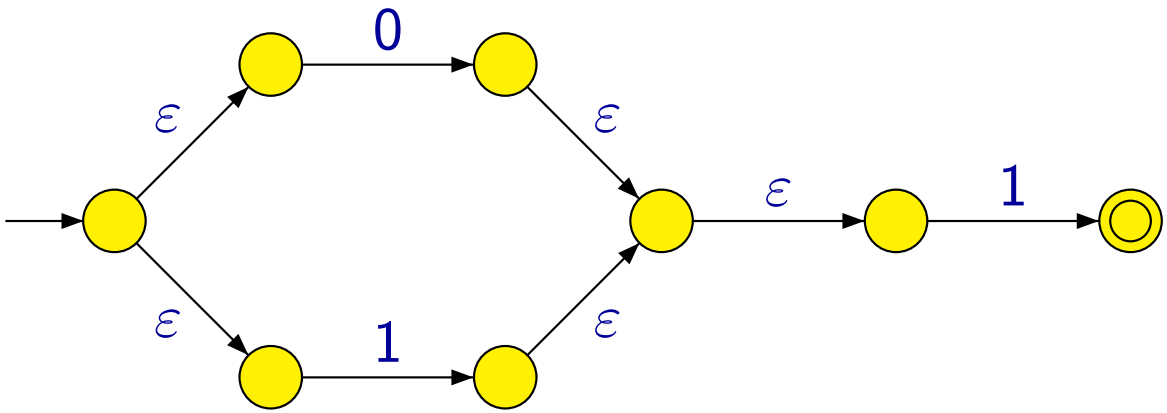
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



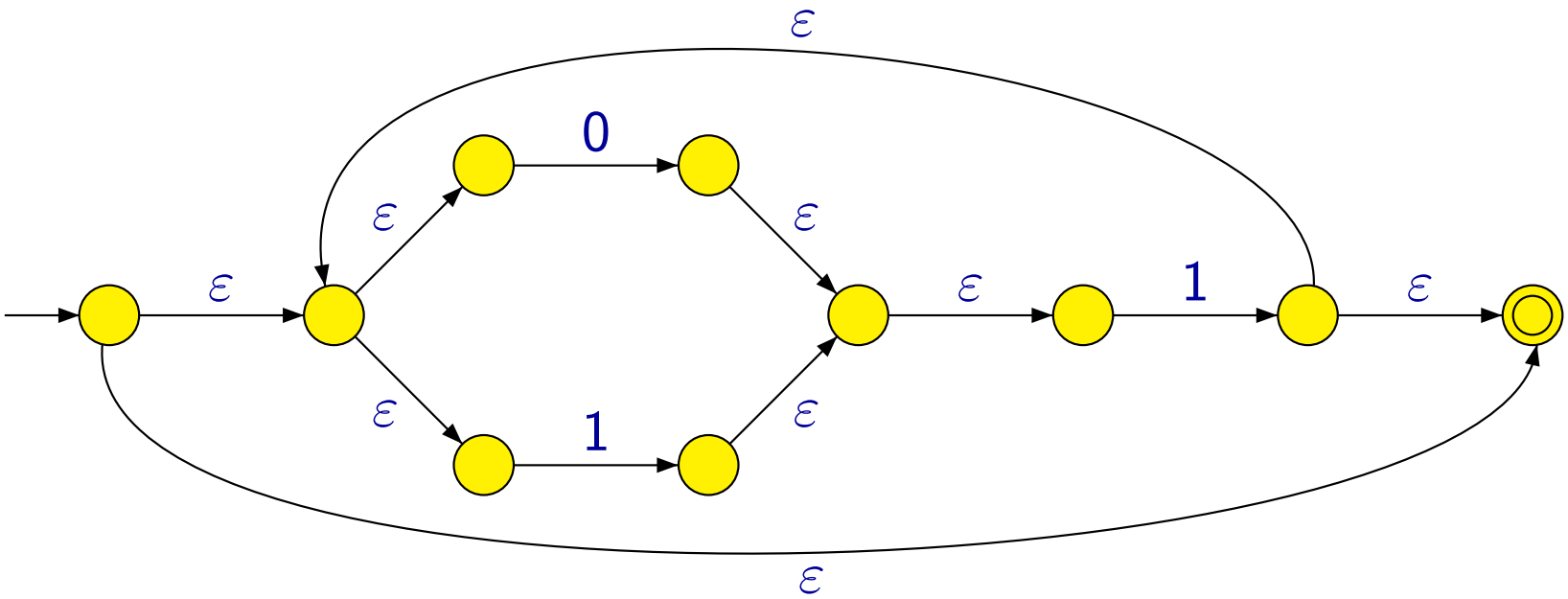
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0+1)\cdot 1)^*$:



Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0+1)\cdot 1)^*$:



Převod regulárního výrazu na konečný automat

Pokud se výraz α skládá z n znaků (nepočítáme-li závorky), má výsledný automat:

- nejvýše $2n$ stavů,
- nejvýše $4n$ přechodů.

Poznámka: Převodem ze zobecněného nedeterministického automatu na deterministický však může počet stavů vzrůst exponenciálně, tj. výsledný automat pak může mít až $2^{2n} = 4^n$ stavů.

Převod konečného automatu na regulární výraz

Tvrzení

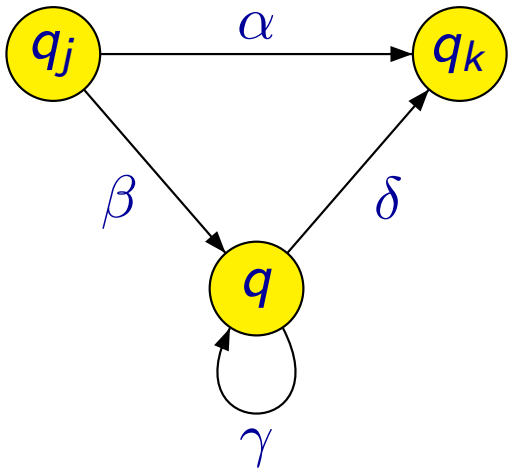
Každý regulární jazyk je možné popsat nějakým regulárním výrazem.

Důkaz: Stačí ukázat, jak pro libovolný konečný automat A zkonstruovat regulární výraz α takový, že $[\alpha] = L(A)$.

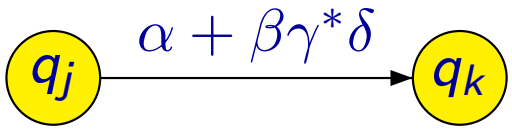
- A upravíme tak, aby měl právě jeden počáteční a právě jeden koncový stav.
- Budeme postupně odebírat jednotlivé stavy.
- Přechody budou označeny regulárními výrazy.
- Zbude automat se dvěma stavy – počátečním a koncovým, a jedním přechodem ohodnoceným výsledným regulárním výrazem.

Převod konečného automatu na regulární výraz

Hlavní myšlenka: Při odstraňování stavu q nahradit pro každou dvojici zbylých stavů q_j, q_k cestu z q_j do q_k vedoucí přes q.

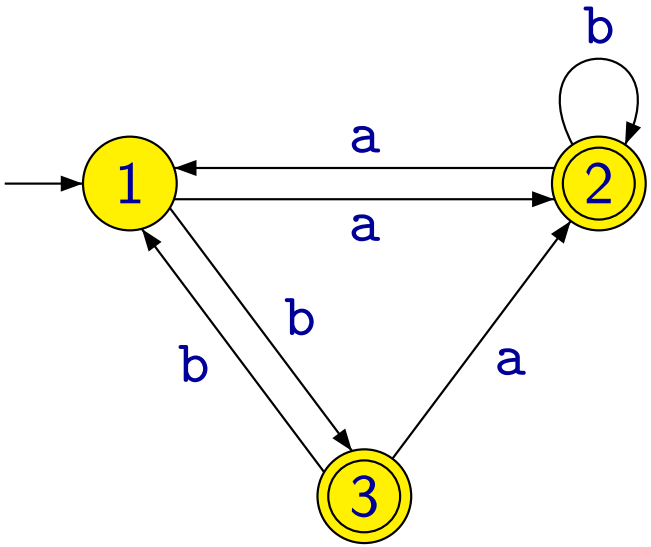


Po odstranění stavu q:



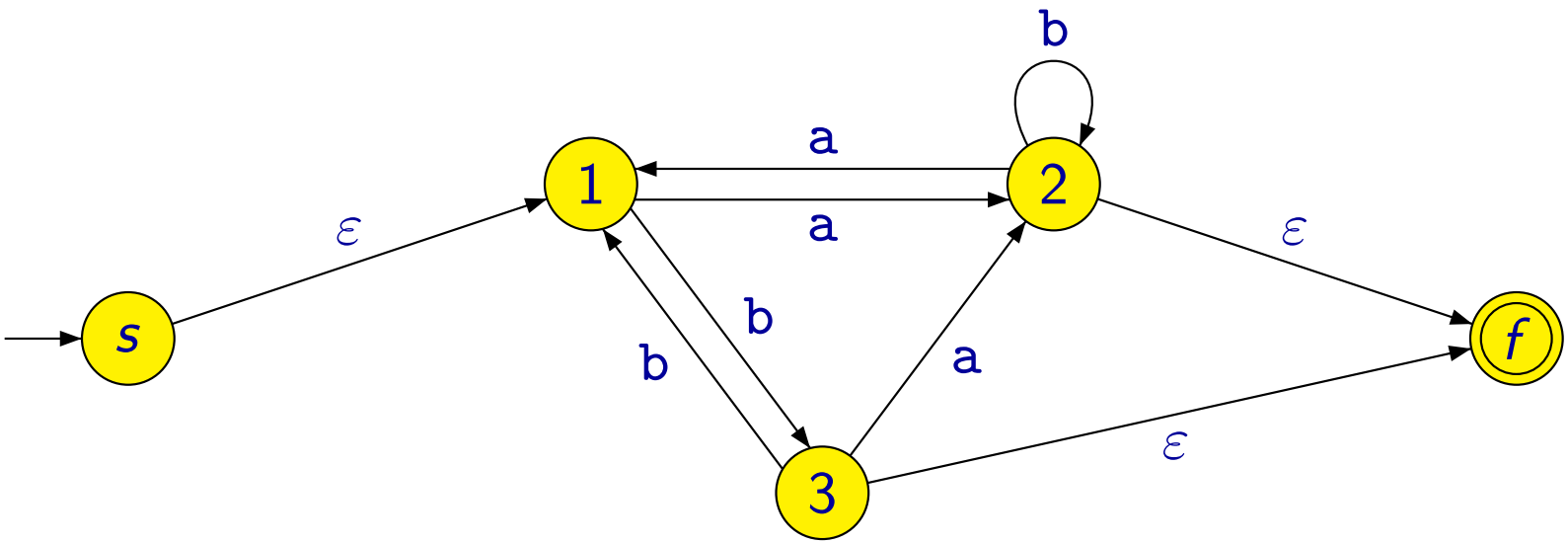
Převod konečného automatu na regulární výraz

Příklad:



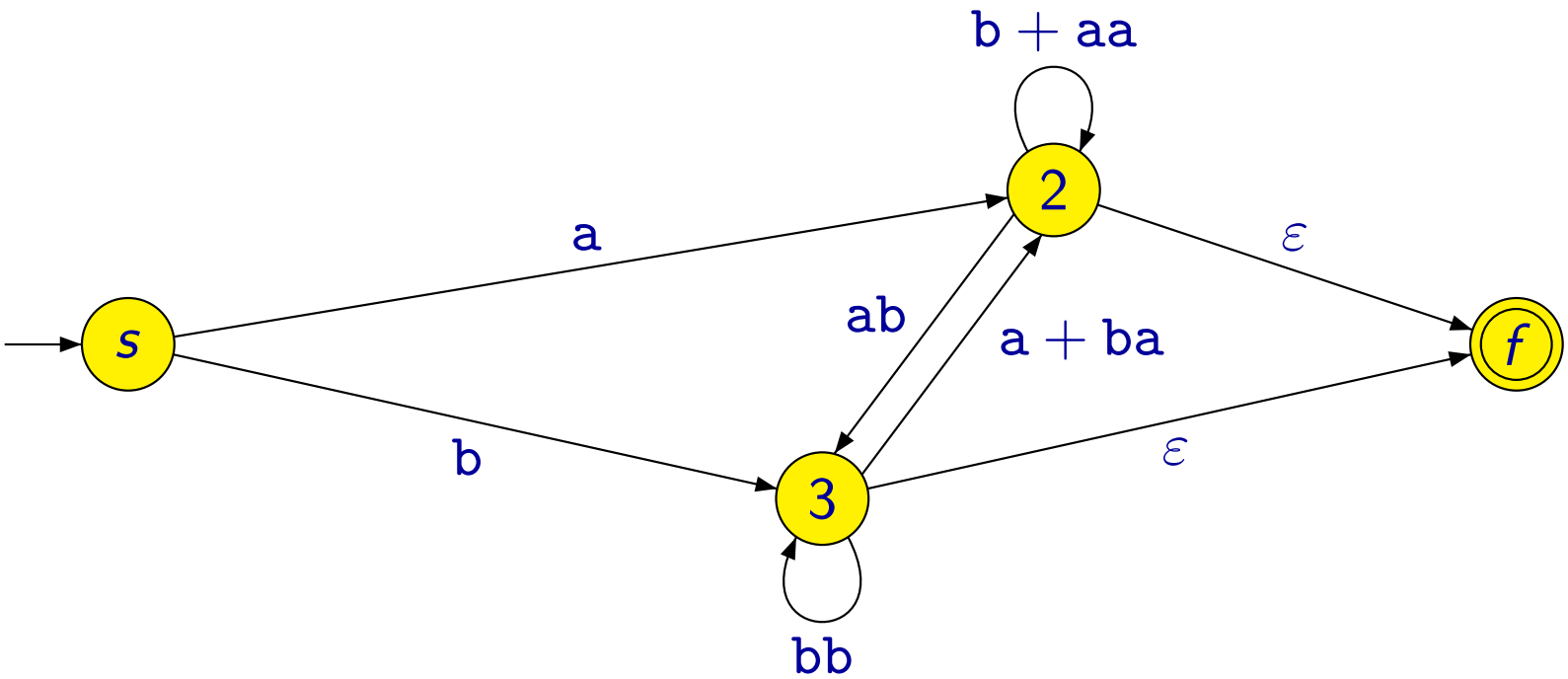
Převod konečného automatu na regulární výraz

Příklad:



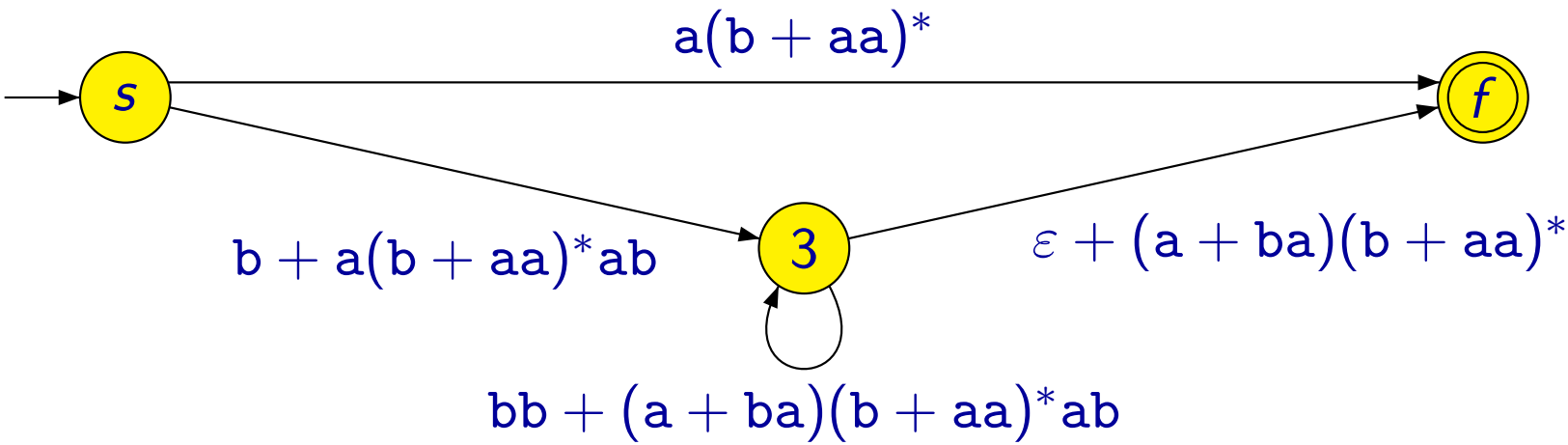
Převod konečného automatu na regulární výraz

Příklad:



Převod konečného automatu na regulární výraz

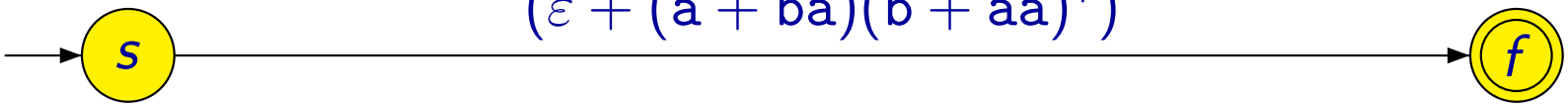
Příklad:



Převod konečného automatu na regulární výraz

Příklad:

$$\mathtt{a}(\mathtt{b}+\mathtt{aa})^*+$$
$$(\mathtt{b}+\mathtt{a}(\mathtt{b}+\mathtt{aa})^*\mathtt{ab})$$
$$(\mathtt{bb}+(\mathtt{a}+\mathtt{ba})(\mathtt{b}+\mathtt{aa})^*\mathtt{ab})^*$$
$$(\varepsilon+(\mathtt{a}+\mathtt{ba})(\mathtt{b}+\mathtt{aa})^*)$$



Ekvivalence konečných automatů a regulárních výrazů

Věta

Jazyk je regulární právě tehdy, když je ho možné popsat regulárním výrazem.

Bezkontextové gramatiky

Bezkontextové gramatiky

Příklad:

$$
\begin{aligned}
\langle \text{STMT} \rangle &\to \langle \text{IF-STMT} \rangle \mid \langle \text{WHILE-STMT} \rangle \mid \langle \text{BLOCK-STMT} \rangle \mid \\
&\quad \langle \text{ASSG-STMT} \rangle \\
\langle \text{IF-STMT} \rangle &\to \textbf{if } \langle \text{BOOL-EXPR} \rangle \textbf{ then } \langle \text{STMT} \rangle \textbf{ else } \langle \text{STMT} \rangle \\
\langle \text{WHILE-STMT} \rangle &\to \textbf{while } \langle \text{BOOL-EXPR} \rangle \textbf{ do } \langle \text{STMT} \rangle \\
\langle \text{BLOCK-STMT} \rangle &\to \textbf{begin } \langle \text{STMT-LIST} \rangle \textbf{ end} \\
\langle \text{STMT-LIST} \rangle &\to \langle \text{STMT} \rangle \mid \langle \text{STMT} \rangle \,; \langle \text{STMT-LIST} \rangle \\
\langle \text{ASSG-STMT} \rangle &\to \langle \text{VAR} \rangle := \langle \text{ARITH-EXPR} \rangle \\
\langle \text{BOOL-EXPR} \rangle &\to \langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle \\
\langle \text{COMPARE-OP} \rangle &\to < \mid > \mid \leq \mid \geq \mid = \mid \neq \\
\langle \text{ARITH-EXPR} \rangle &\to \langle \text{VAR} \rangle \mid \langle \text{CONST} \rangle \mid \\
&\quad (\langle \text{ARITH-EXPR} \rangle \langle \text{ARITH-OP} \rangle \langle \text{ARITH-EXPR} \rangle) \\
\langle \text{ARITH-OP} \rangle &\to + \mid - \mid * \mid / \\
\langle \text{CONST} \rangle &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
\langle \text{VAR} \rangle &\to a \mid b \mid c \mid \cdots \mid x \mid y \mid z
\end{aligned}
$$

Bezkontextové gramatiky

Symbolům, které mají v předchozím příkladě tvar $\langle \text{XXX} \rangle$, se říká **neterminální symboly** (**neterminály**).

Pravidla popisují, jaké řetězce může daný neterminál reprezentovat.

Z neterminálu $\langle \text{STMT} \rangle$ můžeme například dostat text

$$\textbf{while } x \leq y \textbf{ do begin } x := (x + 1)\,;\; y := (y - 1) \textbf{ end}$$

$$\textbf{while } x \leq y \textbf{ do begin } x := (x+1)\,;\; y := (y-1) \textbf{ end}$$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$

Bezkontextové gramatiky

$$\textbf{while } x \leq y \textbf{ do begin } x := (x + 1)\,;\; y := (y - 1) \textbf{ end}$$

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq y$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq y$ **do** $\langle \text{STMT} \rangle$
while $x \leq y$ **do** $\langle \text{BLOCK-STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle$STMT$\rangle$
$\langle$WHILE-STMT$\rangle$
while $\langle$BOOL-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$ARITH-EXPR$\rangle\langle$COMPARE-OP$\rangle\langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle\langle$COMPARE-OP$\rangle\langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle \leq \langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle \leq \langle$VAR$\rangle$ **do** $\langle$STMT$\rangle$
while $x \leq \langle$VAR$\rangle$ **do** $\langle$STMT$\rangle$
while $x \leq y$ **do** $\langle$STMT$\rangle$
while $x \leq y$ **do** $\langle$BLOCK-STMT$\rangle$
while $x \leq y$ **do begin** $\langle$STMT-LIST$\rangle$ **end**

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle$STMT$\rangle$
$\langle$WHILE-STMT$\rangle$
while $\langle$BOOL-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$ARITH-EXPR$\rangle\langle$COMPARE-OP$\rangle\langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle\langle$COMPARE-OP$\rangle\langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle \leq \langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle \leq \langle$VAR$\rangle$ **do** $\langle$STMT$\rangle$
while $x \leq \langle$VAR$\rangle$ **do** $\langle$STMT$\rangle$
while $x \leq y$ **do** $\langle$STMT$\rangle$
while $x \leq y$ **do** $\langle$BLOCK-STMT$\rangle$
while $x \leq y$ **do begin** $\langle$STMT-LIST$\rangle$ **end**
...

Bezkontextové gramatiky

Formálně je **bezkontextová gramatika** definována jako čtveřice

$$G = (\Pi, \Sigma, S, P)$$

kde:

- Π je konečná množina **neterminálních symbolů** (**neterminálů**)
- Σ je konečná množina **terminálních symbolů** (**terminálů**), přičemž $\Pi \cap \Sigma = \emptyset$
- $S \in \Pi$ je **počáteční neterminál**
- $P \subseteq \Pi \times (\Pi \cup \Sigma)^*$ je konečná množina **přepisovacích pravidel**

Bezkontextové gramatiky

Poznámky:

- Pro označení neterminálních symbolů budeme používat velká písmena A, B, C, ...
- Pro označení terminálních symbolů budeme používat malá písmena a, b, c, ... nebo číslice 0, 1, 2, ...
- Pro označení řetězců z $(\Pi \cup \Sigma)^*$ budeme používat malá písmena řecké abecedy α, β, γ, ...
- Místo zápisu (A, α) budeme pro pravidla používat zápis

$$A \to \alpha$$

A – levá strana pravidla
α – pravá strana pravidla

Bezkontextové gramatiky

Příklad: Gramatika $G = (\Pi, \Sigma, S, P)$, kde

- $\Pi = \{A, B, C\}$
- $\Sigma = \{a, b\}$
- $S = A$
- P obsahuje pravidla

$$A \to aBBb$$
$$A \to AaA$$
$$B \to \varepsilon$$
$$B \to bCA$$
$$C \to AB$$
$$C \to a$$
$$C \to b$$

Bezkontextové gramatiky

Poznámka: Pokud máme více pravidel se stejnou levou stranou, jako třeba

$$A \to \alpha_1 \qquad A \to \alpha_2 \qquad A \to \alpha_3$$

můžeme je stručněji zapsat jako

$$A \to \alpha_1 \mid \alpha_2 \mid \alpha_3$$

Například pravidla dříve uvedené gramatiky můžeme zapsat jako

$$\begin{aligned} A &\to aBBb \mid AaA \\ B &\to \varepsilon \mid bCA \\ C &\to AB \mid a \mid b \end{aligned}$$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

A

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$\underline{A}$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \to \underline{aBBb} \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$\underline{A} \Rightarrow \underline{aBBb}$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$\underline{B} \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow a\underline{B}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aB\underline{A}Bb \Rightarrow abC\underline{A}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \to \underline{aBBb} \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb \Rightarrow abC\underline{aBBb}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$\underline{B} \to \underline{\varepsilon} \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb \Rightarrow abCaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$\underline{C} \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$\underline{C} \to AB \mid a \mid \underline{b}$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb \Rightarrow ab\underline{b}aBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$\underline{B} \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$\underline{B} \to \underline{\varepsilon} \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b \Rightarrow abbaBbb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$\underline{B} \to \underline{\varepsilon} \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb \Rightarrow abbabb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Bezkontextové gramatiky

Na řetězcích z $(\Pi \cup \Sigma)^*$ definujeme relaci $\Rightarrow \subseteq (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ takovou, že

$$\alpha \Rightarrow \alpha'$$

právě když $\alpha = \beta_1 A \beta_2$ a $\alpha' = \beta_1 \gamma \beta_2$ pro nějaká $\beta_1, \beta_2, \gamma \in (\Pi \cup \Sigma)^*$ a $A \in \Pi$, kde $(A \to \gamma) \in P$.

Příklad: Jestliže $(B \to bCA) \in P$, pak

$$aCBbA \Rightarrow aCbCAbA$$

Poznámka: Neformálně řečeno zápis $\alpha \Rightarrow \alpha'$ znamená, že z α je možné jedním krokem odvodit α', a to tak, že výskyt nějakého neterminálu A v α nahradíme pravou stranou nějakého pravidla $A \to \gamma$, kde se A vyskytuje na levé straně.

Bezkontextové gramatiky

Na řetězcích z $(\Pi \cup \Sigma)^*$ definujeme relaci $\Rightarrow \subseteq (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ takovou, že

$$\alpha \Rightarrow \alpha'$$

právě když $\alpha = \beta_1 A \beta_2$ a $\alpha' = \beta_1 \gamma \beta_2$ pro nějaká $\beta_1, \beta_2, \gamma \in (\Pi \cup \Sigma)^*$ a $A \in \Pi$, kde $(A \to \gamma) \in P$.

Příklad: Jestliže $(B \to bCA) \in P$, pak

$$aC\underline{B}bA \Rightarrow aC\underline{bCA}bA$$

Poznámka: Neformálně řečeno zápis $\alpha \Rightarrow \alpha'$ znamená, že z α je možné jedním krokem odvodit α', a to tak, že výskyt nějakého neterminálu A v α nahradíme pravou stranou nějakého pravidla $A \to \gamma$, kde se A vyskytuje na levé straně.

Bezkontextové gramatiky

Derivace délky n je posloupnost $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$, kde $\beta_i \in (\Pi \cup \Sigma)^*$ a kde $\beta_{i-1} \Rightarrow \beta_i$ pro všechna $1 \leq i \leq n$, což můžeme stručněji zapsat

$$\beta_0 \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \ldots \Rightarrow \beta_{n-1} \Rightarrow \beta_n$$

Skutečnost, že pro dané $\alpha, \alpha' \in (\Pi \cup \Sigma)^*$ a $n \in \mathbb{N}$ existuje nějaká derivace $\beta_0 \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \ldots \Rightarrow \beta_{n-1} \Rightarrow \beta_n$, kde $\alpha = \beta_0$ a $\alpha' = \beta_n$, zapisujeme

$$\alpha \Rightarrow^n \alpha'$$

Skutečnost, že $\alpha \Rightarrow^n \alpha'$ pro nějaké $n \geq 0$, zapisujeme

$$\alpha \Rightarrow^* \alpha'$$

Poznámka: Relace $\Rightarrow^*$ je reflexivním a tranzitivním uzávěrem relace $\Rightarrow$ (tj. nejmenší reflexivní a tranzitivní relací obsahující relaci $\Rightarrow$).

Bezkontextové gramatiky

Větné formy jsou ty $\alpha \in (\Pi \cup \Sigma)^*$, pro které platí

$$S \Rightarrow^* \alpha$$

kde S je počáteční neterminál.

Bezkontextové gramatiky

Jazyk $L(G)$ generovaný gramatikou $G = (\Pi, \Sigma, S, P)$ je množina všech slov v abecedě Σ, která lze odvodit nějakou derivací z počátečního neterminálu S pomocí pravidel z P, tj.

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk

$$L = \{a^n b^n \mid n \geq 0\}$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk

$$L = \{a^n b^n \mid n \geq 0\}$$

Gramatika $G = (\Pi, \Sigma, S, P)$, kde $\Pi = \{S\}$, $\Sigma = \{a, b\}$ a P obsahuje

$$S \to aSb \mid \varepsilon$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk

$$L = \{a^n b^n \mid n \geq 0\}$$

Gramatika $G = (\Pi, \Sigma, S, P)$, kde $\Pi = \{S\}$, $\Sigma = \{a, b\}$ a P obsahuje

$$S \to aSb \mid \varepsilon$$

$S \Rightarrow \varepsilon$
$S \Rightarrow aSb \Rightarrow ab$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaabbbb$
$\ldots$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk tvořený všemi palindromy nad abecedou $\{a, b\}$, tj.

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

Poznámka: w^R označuje tzv. **zrcadlový obraz** slova w, tj. slovo w zapsané pozpátku.

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk tvořený všemi palindromy nad abecedou $\{a, b\}$, tj.

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

Poznámka: w^R označuje tzv. **zrcadlový obraz** slova w, tj. slovo w zapsané pozpátku.

Řešení:

$$S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk tvořený všemi palindromy nad abecedou $\{a, b\}$, tj.

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

Poznámka: w^R označuje tzv. **zrcadlový obraz** slova w, tj. slovo w zapsané pozpátku.

Řešení:

$$S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$$

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow abaaaba$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře uzávorkovanými sekvencemi symbolů '(' a ')'.

Například $(()())(()) \in L$, ale $)()) \notin L$.

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře uzávorkovanými sekvencemi symbolů '(' a ')'.

Například $(()())(()) \in L$, ale $)()) \notin L$.

Řešení:

$$S \to \varepsilon \mid (S) \mid SS$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře uzávorkovanými sekvencemi symbolů '(' a ')'.

Například $(()())(()) \in L$, ale $)()) \notin L$.

Řešení:

$$S \to \varepsilon \mid (S) \mid SS$$

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow (S)(S) \Rightarrow (SS)(S) \Rightarrow ((S)S)(S) \Rightarrow$
$(()S)(S) \Rightarrow (()(S))(S) \Rightarrow (()())(S) \Rightarrow (()())((S)) \Rightarrow$
$(()())(())$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře vytvořenými aritmetickými výrazy, kde operandy jsou vždy tvaru 'a', a kde jako operátory můžeme používat symboly $+$ a $*$.

Například $(a + a) * a + (a * a) \in L$.

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře vytvořenými aritmetickými výrazy, kde operandy jsou vždy tvaru 'a', a kde jako operátory můžeme používat symboly $+$ a $*$.

Například $(a + a) * a + (a * a) \in L$.

Řešení:

$$E \to a \mid E + E \mid E * E \mid (E)$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře vytvořenými aritmetickými výrazy, kde operandy jsou vždy tvaru 'a', a kde jako operátory můžeme používat symboly $+$ a $*$.

Například $(a + a) * a + (a * a) \in L$.

Řešení:

$$E \to a \mid E + E \mid E * E \mid (E)$$

$$E \Rightarrow E + E \Rightarrow E * E + E \Rightarrow (E) * E + E \Rightarrow (E + E) * E + E \Rightarrow$$
$$(a + E) * E + E \Rightarrow (a + a) * E + E \Rightarrow (a + a) * a + E \Rightarrow (a + a) * a + (E) \Rightarrow$$
$$(a + a) * a + (E * E) \Rightarrow (a + a) * a + (a * E) \Rightarrow (a + a) * a + (a * a)$$

Derivační strom

$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

Derivační strom

A

$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

A

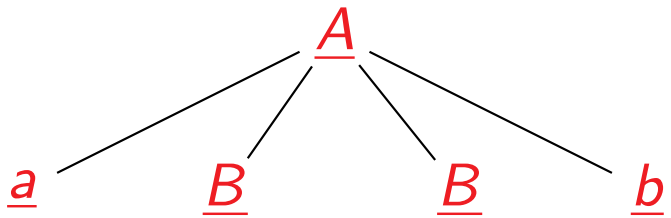
Derivační strom

$\underline{A}$

$\underline{A} \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

$\underline{A}$

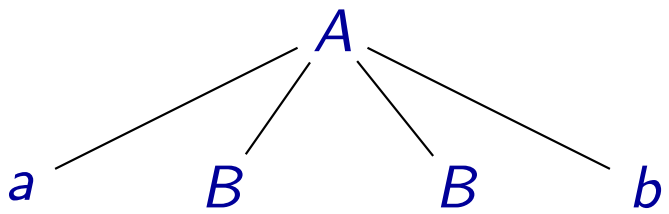
Derivační strom



$\underline{A} \to \underline{aBBb} \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

$\underline{A} \Rightarrow \underline{aBBb}$

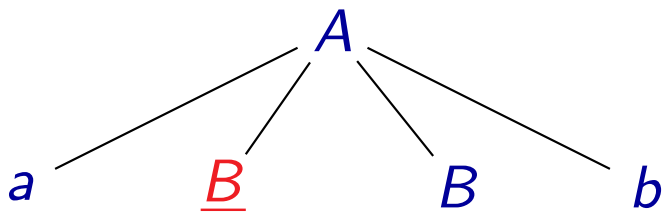
Derivační strom



$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

$A \Rightarrow aBBb$

Derivační strom

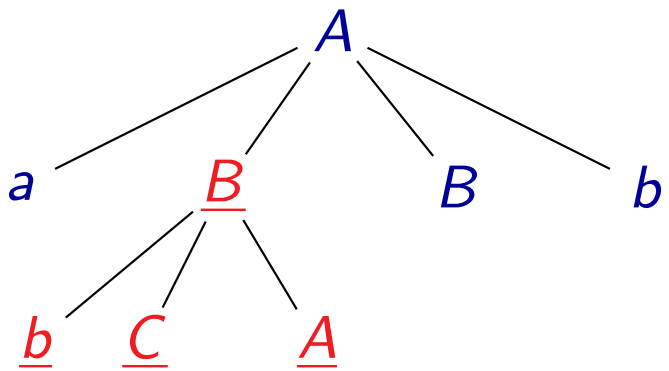


$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

$A \Rightarrow a\underline{B}Bb$

Derivační strom

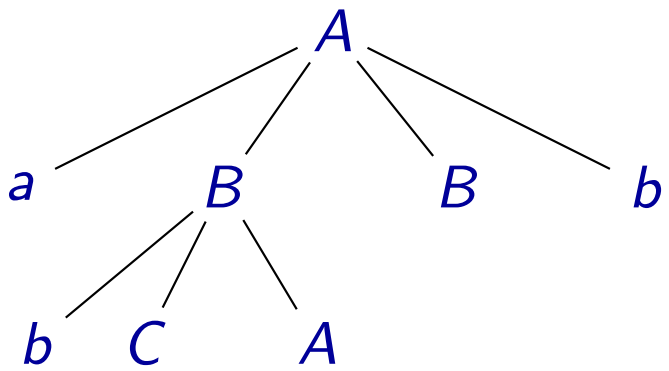
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid \underline{bCA}$
$C \to AB \mid a \mid b$



$A \Rightarrow a\underline{B}Bb \Rightarrow a\underline{bCA}Bb$

Derivační strom

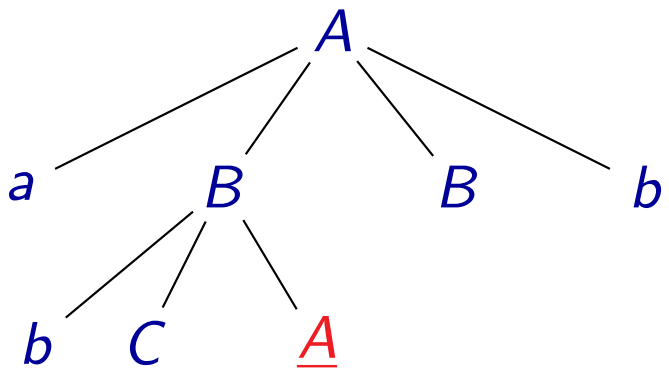
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb$

Derivační strom

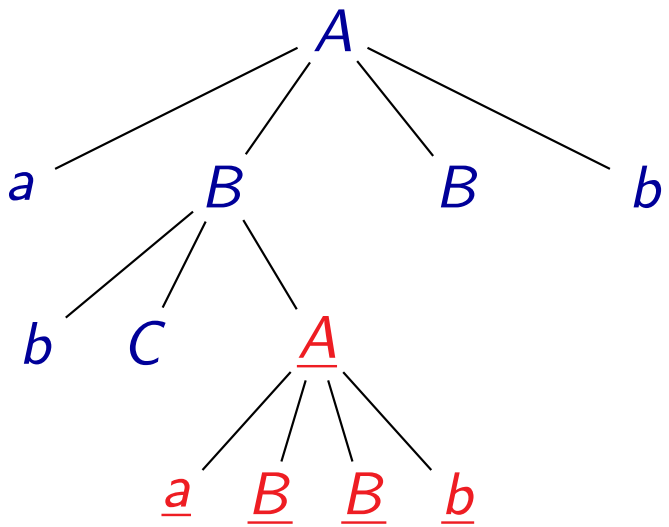
$\underline{A} \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb$

Derivační strom

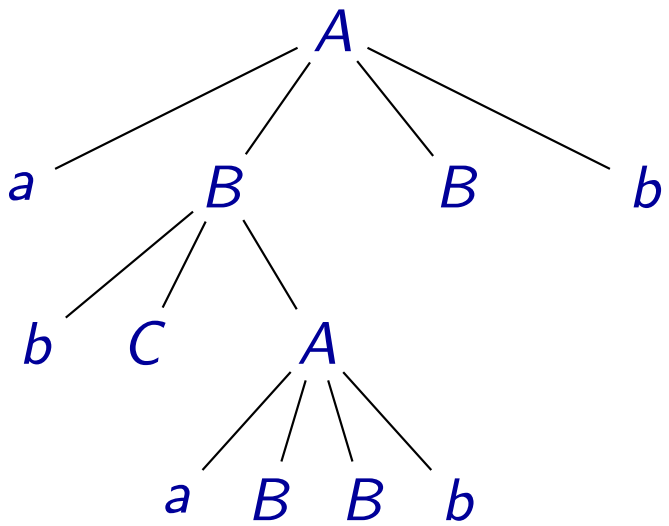
$\underline{A} \to \underline{aBBb} \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb \Rightarrow abC\underline{aBBb}Bb$

Derivační strom

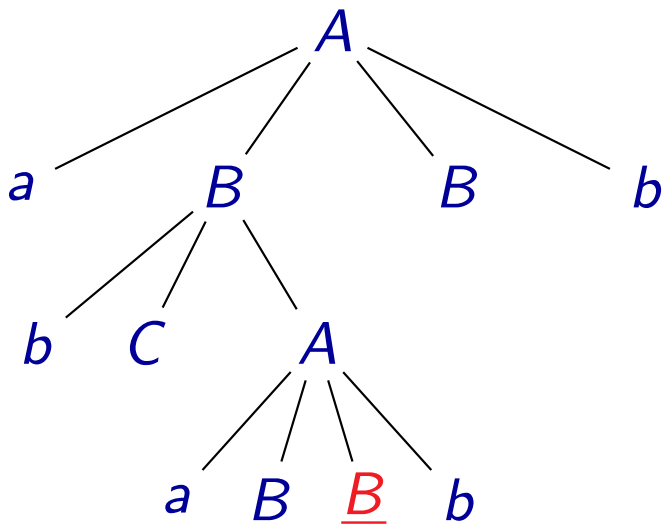
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$

Derivační strom

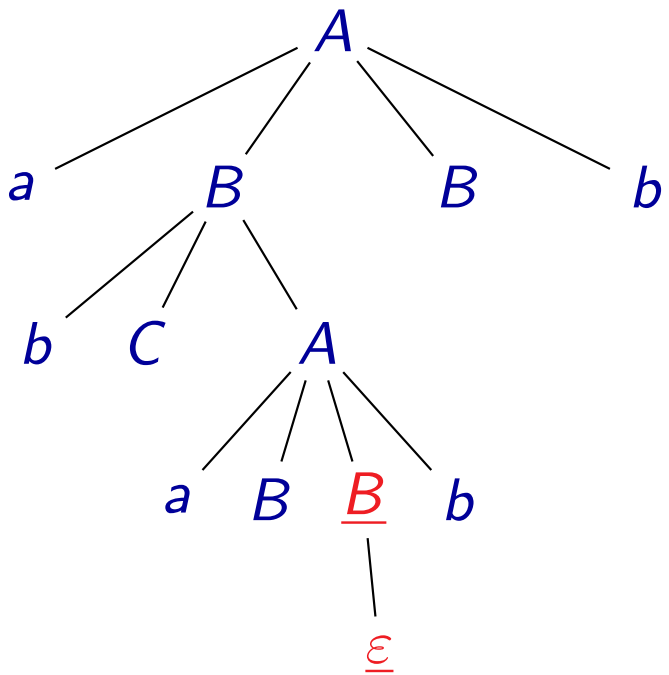
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb$

Derivační strom

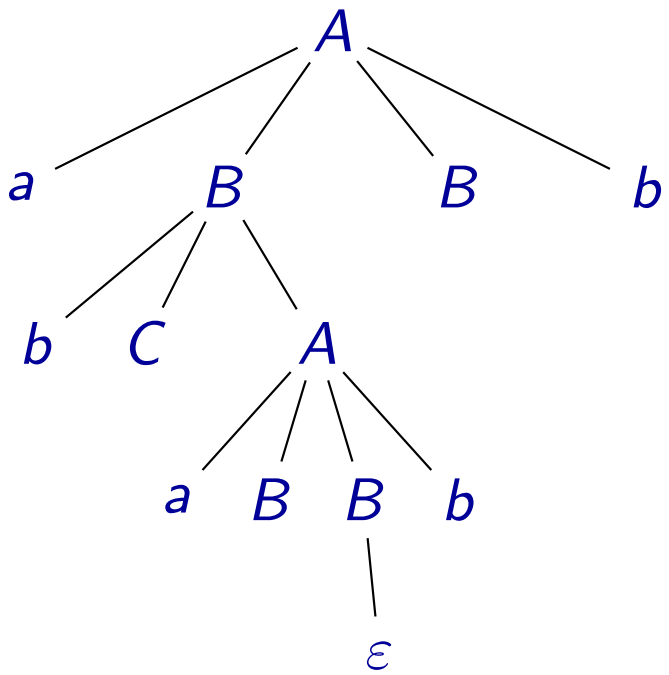
$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb \Rightarrow abCaBbBb$

Derivační strom

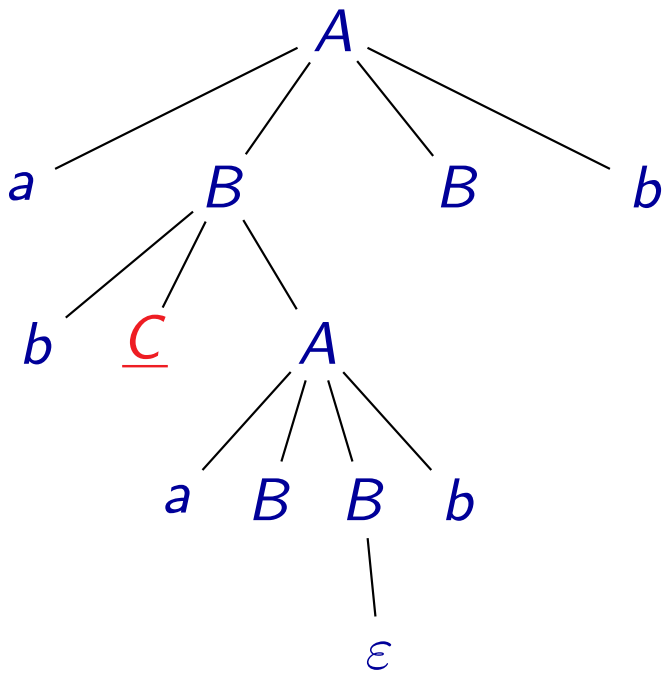
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$

Derivační strom

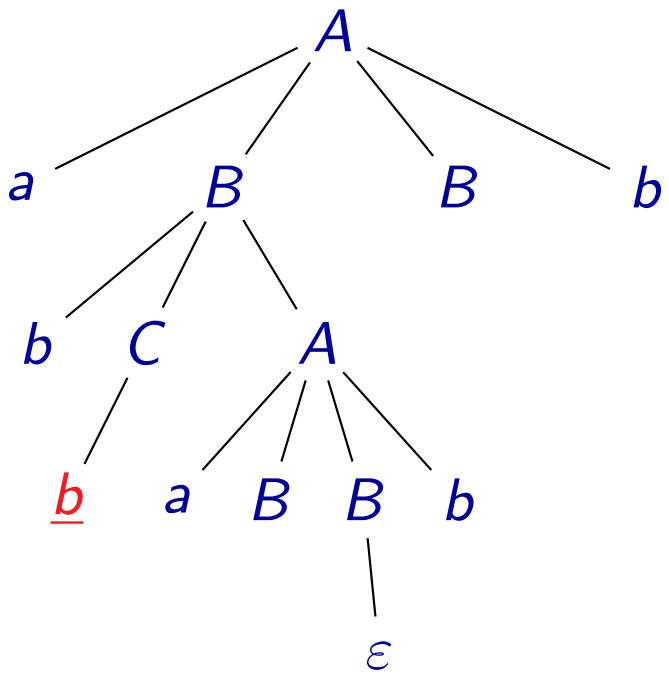
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$\underline{C} \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb$

Derivační strom

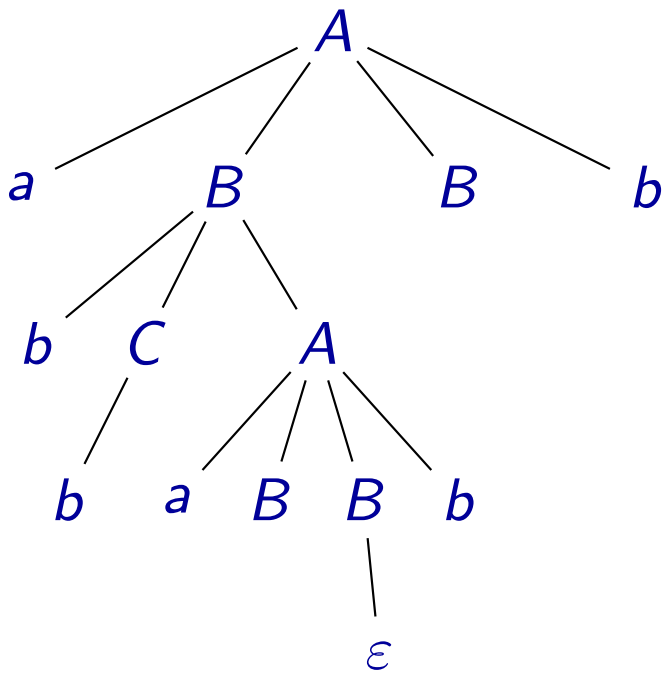
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$\underline{C} \to AB \mid a \mid \underline{b}$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb \Rightarrow ab\underline{b}aBbBb$

Derivační strom

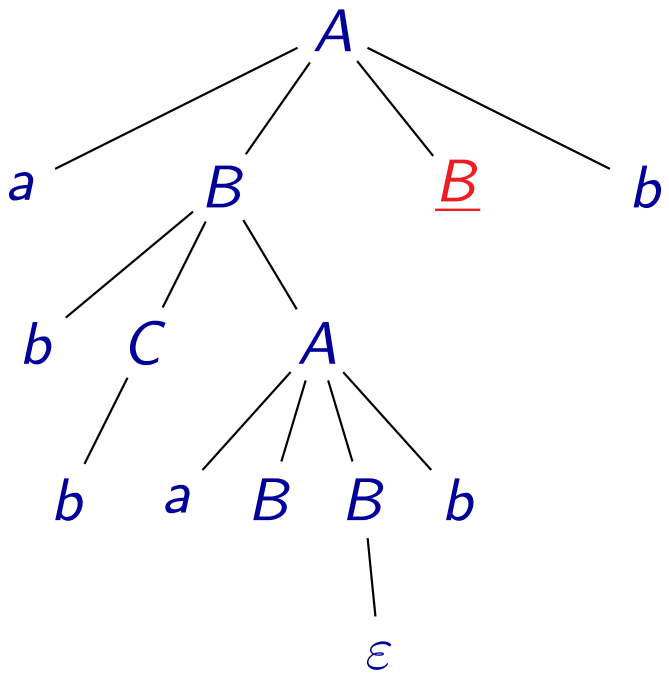
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$

Derivační strom

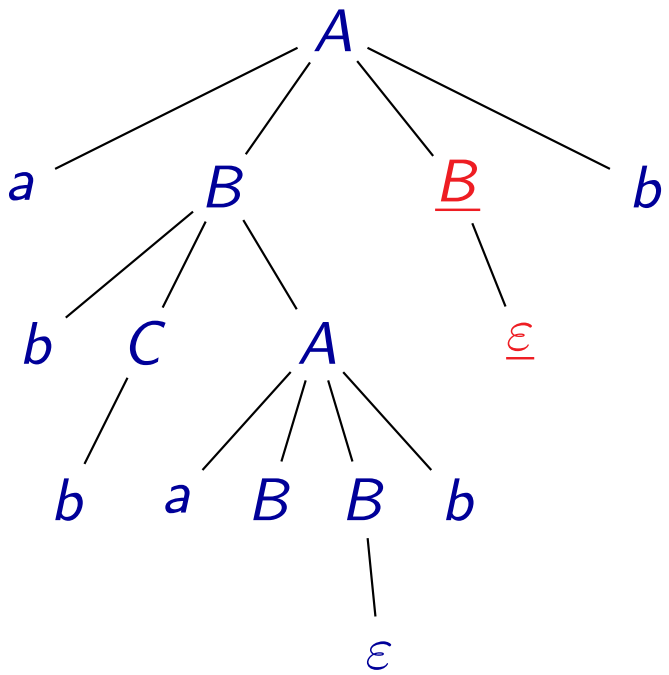
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b$

Derivační strom

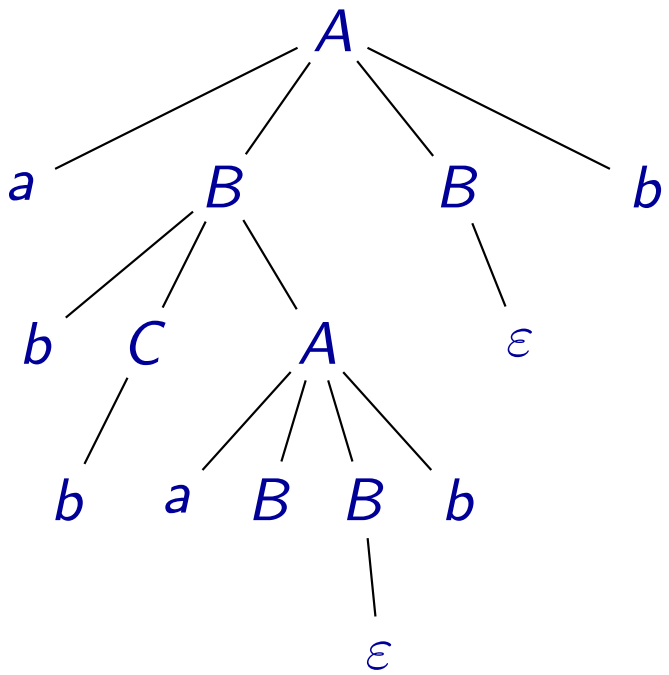
$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b \Rightarrow abbaBbb$

Derivační strom

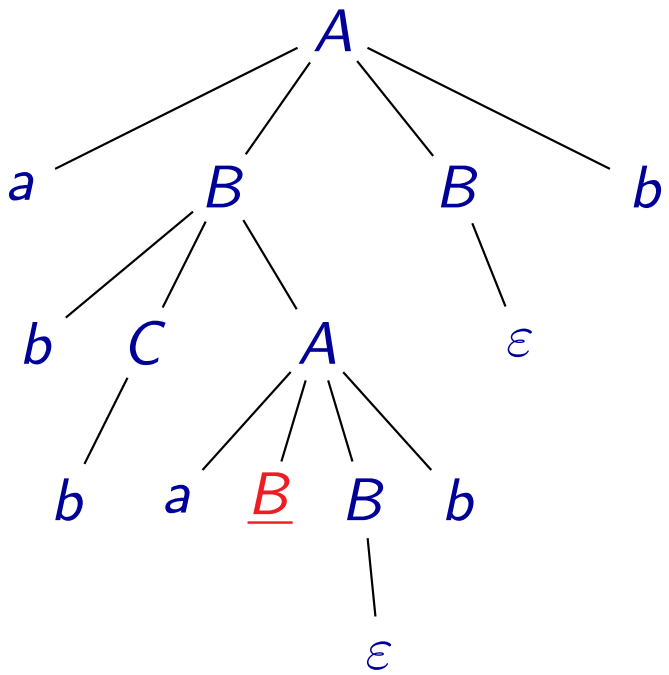
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Derivační strom

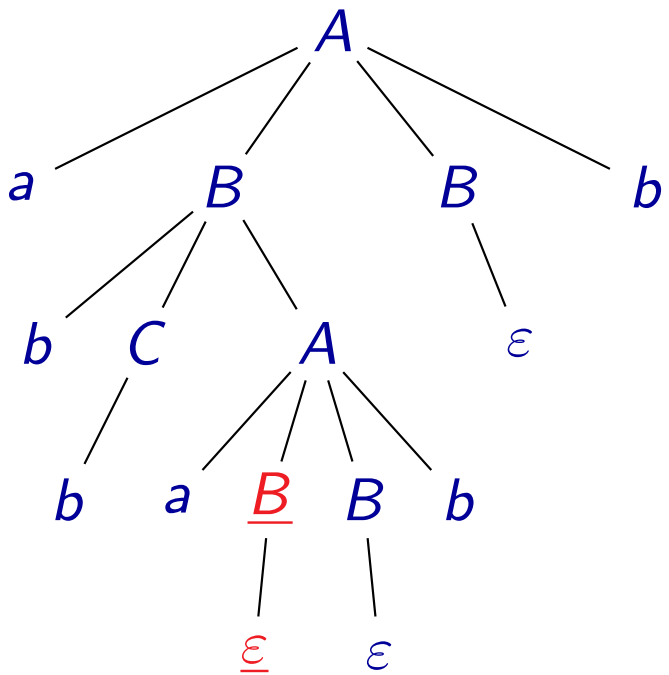
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb$

Derivační strom

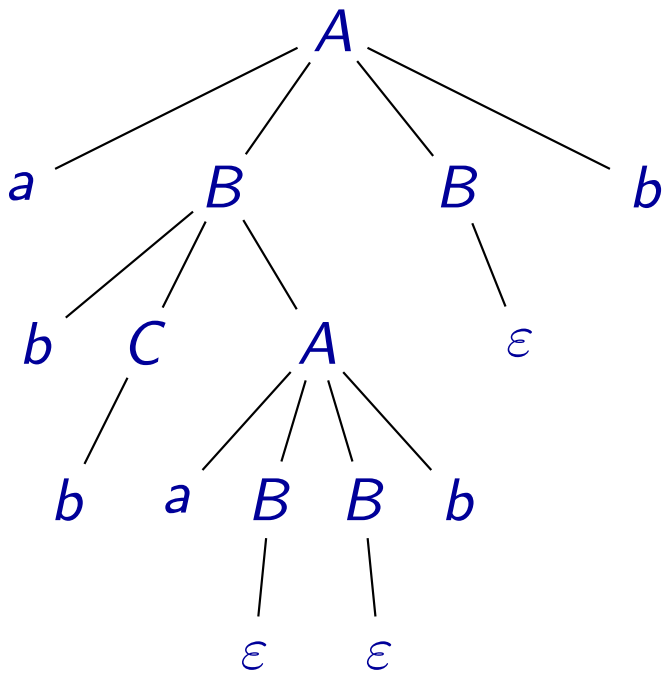
$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb \Rightarrow abbabb$

Derivační strom

$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Derivační strom

Každé derivaci odpovídá nějaký **derivační strom**:

- Vrcholy stromu jsou ohodnoceny terminály a neterminály.
- Kořen stromu je ohodnocen počátečním neterminálem.
- Listy stromu jsou ohodnoceny terminály nebo symboly ε.
- Ostatní vrcholy stromu jsou ohodnoceny neterminály.
- Pokud je vrchol ohodnocen neterminálem A, pak jeho potomci jsou ohodnoceni symboly pravé strany nějakého přepisovacího pravidla $A \to \alpha$.

Levá a pravá derivace

$$E \to E + E \mid E * E \mid (E) \mid a$$

Levá derivace je derivace, ve které v každém kroku nahrazujeme vždy nejlevější neterminál.

$$\underline{E} \Rightarrow \underline{E} + E \Rightarrow \underline{E} * E + E \Rightarrow a * \underline{E} + E \Rightarrow a * a + \underline{E} \Rightarrow a * a + a$$

Pravá derivace je derivace, ve které v každém kroku nahrazujeme vždy nejpravější neterminál.

$$\underline{E} \Rightarrow E + \underline{E} \Rightarrow \underline{E} + a \Rightarrow E * \underline{E} + a \Rightarrow \underline{E} * a + a \Rightarrow a * a + a$$

Derivace však nemusí být ani levá ani pravá:

$$\underline{E} \Rightarrow \underline{E} + E \Rightarrow E * \underline{E} + E \Rightarrow E * a + \underline{E} \Rightarrow \underline{E} * a + a \Rightarrow a * a + a$$

Levá a pravá derivace

- Jednomu derivačnímu stromu může odpovídat více různých derivací.
- Každému derivačnímu stromu odpovídá právě jedna levá a právě jedna pravá derivace.

Ekvivalence gramatik

Gramatiky G_1 a G_2 jsou **ekvivalentní**, jestliže generují tentýž jazyk, tj. jestliže $L(G_1) = L(G_2)$.

Poznámka: Problém ekvivalence bezkontextových gramatik je algoritmicky nerozhodnutelný. Dá se dokázat, že není možné vytvořit algoritmus, který by pro libovolné dvě bezkontextové gramatiky rozhodl, zda jsou ekvivalentní či ne.

Dokonce je algoritmicky nerozhodnutelný i problém, zda gramatika generuje jazyk Σ^*.

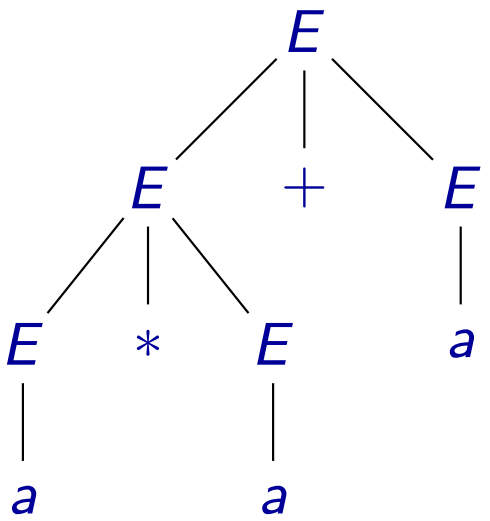
Nejednoznačné gramatiky

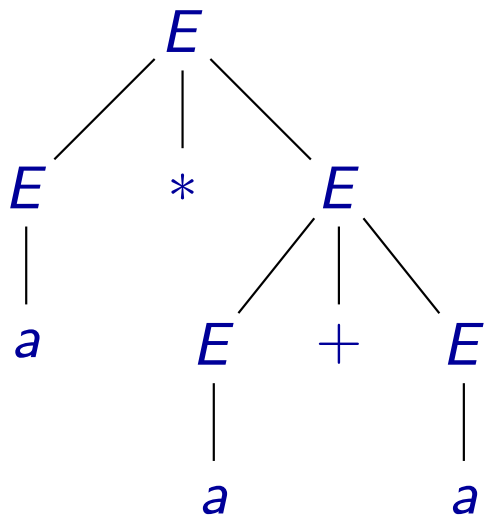
Gramatika G je **nejednoznačná**, jestliže existuje nějaké slovo $w \in L(G)$, kterému přísluší dva různé derivační stromy, resp. dvě různé levé či dvě různé pravé derivace.

Příklad:

$E \Rightarrow E + E \Rightarrow E * E + E \Rightarrow a * E + E \Rightarrow a * a + E \Rightarrow a * a + a$

$E \Rightarrow E * E \Rightarrow E * E + E \Rightarrow a * E + E \Rightarrow a * a + E \Rightarrow a * a + a$





Nejednoznačné gramatiky

Někdy je možné nejednoznačnou gramatiku nahradit gramatikou, která generuje tentýž jazyk, ale není nejednoznačná.

Příklad: Gramatiku

$$E \to E + E \mid E * E \mid (E) \mid a$$

můžeme nahradit ekvivalentní gramatikou

$$\begin{aligned} E &\to T \mid T + E \\ T &\to F \mid F * T \\ F &\to a \mid (E) \end{aligned}$$

Poznámka: Pokud se nejednoznačná gramatika žádnou ekvivalentní jednoznačnou gramatikou nahradit nedá, říkáme, že je **podstatně nejednoznačná**.

Bezkontextové jazyky

Definice

Jazyk L je **bezkontextový**, jestliže existuje bezkontextová gramatika G taková, že $L = L(G)$.

Třída bezkontextových jazyků je uzavřená vůči:

- zřetězení
- sjednocení
- iteraci

Třída bezkontextových jazyků však není uzavřená vůči:

- doplňku
- průniku

Bezkontextové jazyky

Máme dány gramatiky $G_1 = (\Pi_1, \Sigma, S_1, P_1)$ a $G_2 = (\Pi_2, \Sigma, S_2, P_2)$, přičemž můžeme předpokládat, že $\Pi_1 \cap \Pi_2 = \emptyset$ a $S \notin \Pi_1 \cup \Pi_2$.

- Gramatika G taková, že $L(G) = L(G_1)L(G_2)$:

$$G = (\Pi_1 \cup \Pi_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1 S_2\})$$

- Gramatika G taková, že $L(G) = L(G_1) \cup L(G_2)$:

$$G = (\Pi_1 \cup \Pi_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\})$$

- Gramatika G taková, že $L(G) = L(G_1)^*$:

$$G = (\Pi_1 \cup \{S\}, \Sigma, S, P_1 \cup \{S \to \varepsilon, S \to S_1 S\})$$

Vyčíslitelnost a složitost

Co je to algoritmus?

Algoritmus

Algoritmus je mechanický postup skládající se z nějakých jednoduchých elementárních kroků, který pro nějaký zadaný **vstup** vyprodukuje nějaký **výstup**.

Algoritmus může být zadán:

- slovním popisem v přirozeném jazyce
- pseudokódem
- jako počítačový program v nějakém programovacím jazyce
- jako hardwarový obvod
- ...

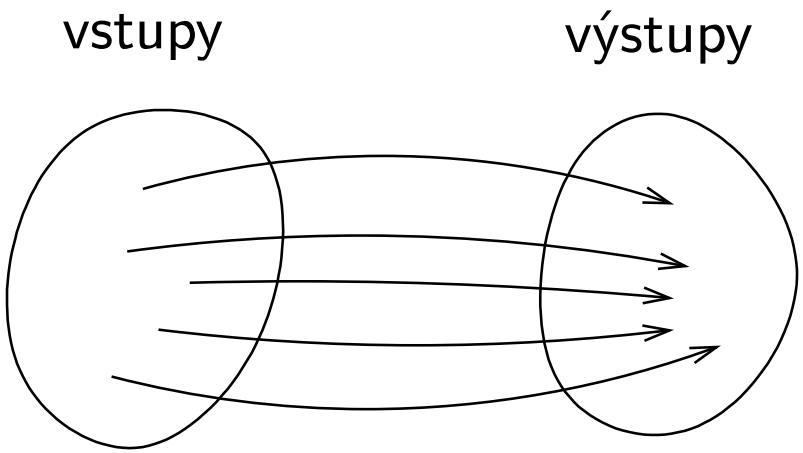
Algoritmy slouží k řešení různých **problémů**.

Problémy

Problém

V zadání **problému** musí být určeno:

- co je množinou možných vstupů
- co je množinou možných výstupů
- jaký je vztah mezi vstupy a výstupy



Příklady problémů

Problém „Třídění“

Vstup: Sekvence prvků $a_1, a_2, \ldots, a_n$.

Výstup: Prvky sekvence $a_1, a_2, \ldots, a_n$ seřazené od nejmenšího po největší.

Příklad:

- Vstup: $8, 13, 3, 10, 1, 4$
- Výstup: $1, 3, 4, 8, 10, 13$

Poznámka: Konkrétní vstup nějakého problému se nazývá **instance** problému.

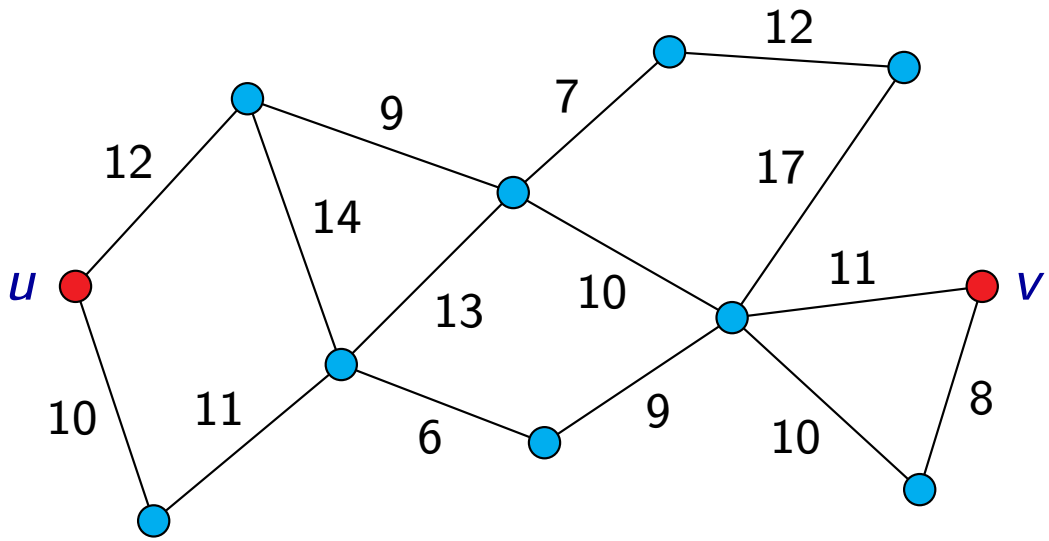
Příklady problémů

Problém „Hledání nejkratší cesty v (neorientovaném) grafu“

Vstup: Neorientovaný graf $G = (V, E)$ s ohodnocením hran, a dvojice vrcholů $u, v \in V$.

Výstup: Nejkratší cesta z vrcholu u do vrcholu v.

Příklad:



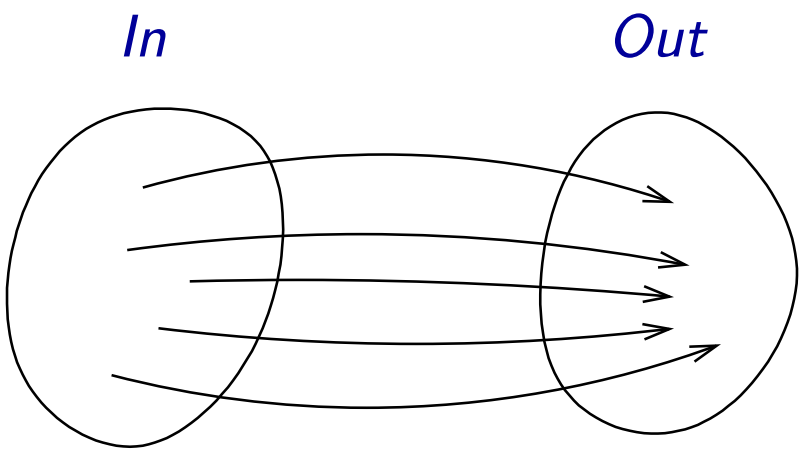
Problémy

Problém

Formálně tedy můžeme **problém** definovat jako trojici (In, Out, R), kde:

- In je množina možných vstupů
- Out je množina možných výstupů
- $R \subseteq In \times Out$ je relace přiřazující každému vstupu možné odpovídající výstupy. Tato relace musí splňovat

$$\forall x \in In : \exists y \in Out : R(x, y).$$



Kódování vstupu a výstupu

Obecně se můžeme omezit na to, že vstupy i výstupy problému jsou slova v nějaké abecedě Σ, tj. $In = Out = \Sigma^*$.

Různé jiné objekty (čísla, posloupnosti čísel, grafy, ...) pak zapisujeme (kódojeme) jako slova v této abecedě.

Příklad: Například u problému „Třídění“ bychom mohli zvolit jako abecedu $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ,\}$.

Vstupem by pak mohlo být například slovo

`826,13,3901,101,128,562`

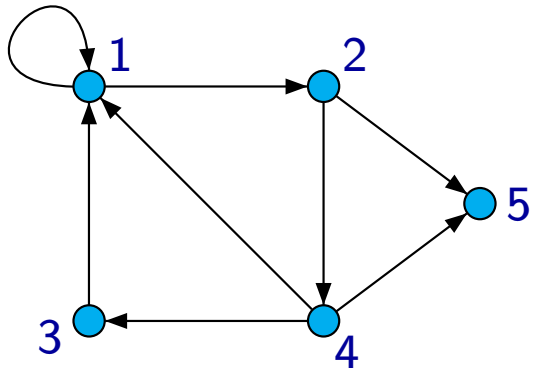
a výstupem slovo

`13,101,128,562,826,3901`

Kódování vstupu a výstupu

Příklad: Pokud je vstupem nějakého problému například graf, můžeme ho reprezentovat jako seznam vrcholů a hran:

Například následující graf



můžeme reprezentovat jako slovo

$$(1,2,3,4,5),((1,2),(2,4),(4,3),(3,1),(1,1),(2,5),(4,5),(4,1))$$

v abecedě $\Sigma = \{0,1,2,3,4,5,6,7,8,9,,,(,)\}$.

Kódování vstupu a výstupu

Poznámka: Ne každé slovo ze Σ^* musí reprezentovat nějaký vstup. Kódování bychom ale měli zvolit tak, abychom byli schopni snadno poznat ta slova, která nějaký vstup reprezentují.

Kódování vstupu a výstupu

Můžeme se omezit na případ, kdy jsou vstupy i výstupy kódovány jako slova v abecedě $\{0, 1\}$ (tj. jako sekvence bitů).

Symboly jakékoli jiné abecedy lze reprezentovat jako sekvence bitů.

Příklad: Abeceda $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$

$$
\begin{aligned}
\mathtt{a} &\leftrightarrow 001 \\
\mathtt{b} &\leftrightarrow 010 \\
\mathtt{c} &\leftrightarrow 011 \\
\mathtt{d} &\leftrightarrow 100 \\
\mathtt{e} &\leftrightarrow 101 \\
\mathtt{f} &\leftrightarrow 110 \\
\mathtt{g} &\leftrightarrow 111
\end{aligned}
$$

Slovo '$\mathtt{defb}$' můžeme reprezentovat jako '100101110010'.

Další příklady problémů

Problém „Prvočíselnost“

Vstup: Přirozené číslo n.

Výstup: ANO pokud je n prvočíslo, NE v opačném případě.

Poznámka: Přirozené číslo n je **prvočíslo**, pokud je větší než 1 a je dělitelné beze zbytku pouze čísly 1 a n.

Prvních několik prvočísel: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ...

Rozhodovací problémy

Situace, kdy množina výstupů *Out* je $\{\text{Ano}, \text{Ne}\}$ je poměrně častá. Takovým problémům se říká **rozhodovací problémy**.

Rozhodovací problémy většinou specifikujeme tak, že místo popisu toho, co je výstupem, uvedeme otázku.

Příklad:

Problém „Prvočíselnost“

Vstup: Přirozené číslo n.

Otázka: Je n prvočíslo?

Rozhodovací problémy

Rozhodovací problém

Jednou z možností, jak formálně definovat **rozhodovací problém**, je definovat ho jako dvojici (In, T), kde:

- In je množina možných vstupů,
- $T \subseteq In$ je množina vstupů, pro které je odpověď ANO.

Rozhodovací problémy a jazyky

Pokud se omezíme na to, že vstupy jsou slova z nějaké abecedy Σ, můžeme na rozhodovací problémy pohlížet jako na jazyky.

Jazyk odpovídající danému rozhodovacímu problému je množina těch slov ze Σ^*, která reprezentují ty vstupy, pro něž je odpověď ANO.

Příklad: Jazyk L tvořený těmi slovy ze $\{0,1\}^*$, která jsou binárním zápisem nějakého prvočísla.

Například $101 \in L$, ale $110 \notin L$.

Příklad rozhodovacího problému

Problém SAT (splnitelnost booleovských formulí)

Vstup: Booleovská formule φ.

Otázka: Je φ splnitelná?

Příklad:

Formule $\varphi_1 = x_1 \wedge (\neg x_2 \vee x_3)$ je splnitelná:
např. při ohodnocení ν, kde $\nu(x_1) = 1$, $\nu(x_2) = 0$, $\nu(x_3) = 1$, platí $\nu(\varphi_1) = 1$.

Formule $\varphi_2 = (x_1 \wedge \neg x_1) \vee (\neg x_2 \wedge x_3 \wedge x_2)$ není splnitelná:
pro libovolné ohodnocení ν platí $\nu(\varphi_2) = 0$.

Optimalizační problémy

Dalším speciálním případem jsou tzv. optimalizační problémy.

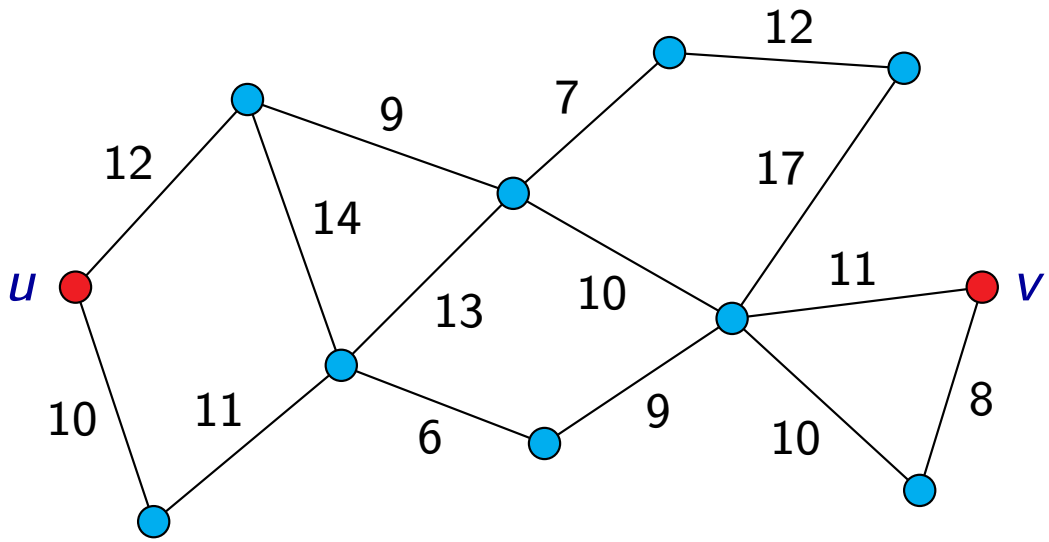
Optimalizační problém je problém, kde je úkolem vybrat z nějaké množiny přípustných řešení takové řešení, které je v nějakém ohledu optimální.

Optimalizační problémy

Dalším speciálním případem jsou tzv. optimalizační problémy.

Optimalizační problém je problém, kde je úkolem vybrat z nějaké množiny přípustných řešení takové řešení, které je v nějakém ohledu optimální.

Příklad: V problému „Hledání nejkratší cesty v grafu“ je množina všech přípustných řešení tvořena všemi cestami z vrcholu u do vrcholu v. Kritériem, podle kterého cesty hodnotíme, je délka cesty.



Optimalizační problémy

Formálně můžeme **optimalizační problém** definovat jako pětici (In, Out, f, m, g), kde:

- In je množina možných vstupů,
- Out je množina **řešení**,
- $f : In \to \mathcal{P}(Out)$ je funkce přiřazující každému vstupu x odpovídající množinu **přípustných řešení** $f(x)$,
- $m : \bigcup_{x \in In}(\{x\} \times f(x)) \to \mathbb{R}$ je **optimalizační funkce** (**kriteriální funkce**),
- g je min nebo max.

Cílem je pro daný vstup $x \in In$ najít nějaké přípustné řešení $y \in f(x)$ takové, že

$$m(x, y) = g\{m(x, y') \mid y' \in f(x)\},$$

nebo zjistit, že pro daný vstup x žádné přípustné řešení neexistuje (tj. $f(x) = \emptyset$).

Optimalizační problémy

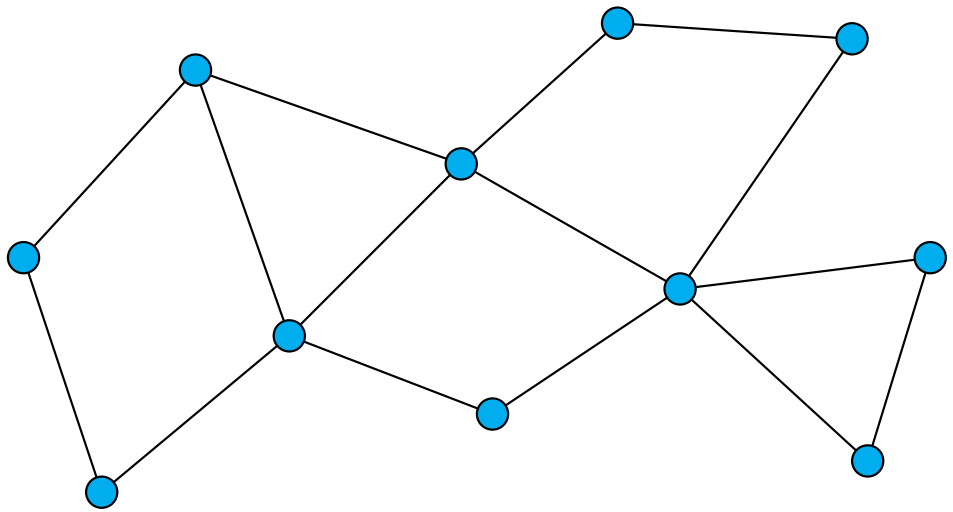
- Optimalizačním problémům, kde g je min, se říká **minimalizační problémy**.
- Optimalizačním problémům, kde g je max, se říká **maximalizační problémy**.

Příklady problémů

Problém „Barvení grafu k barvami“

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Je možné obarvit vrcholy grafu G k barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?



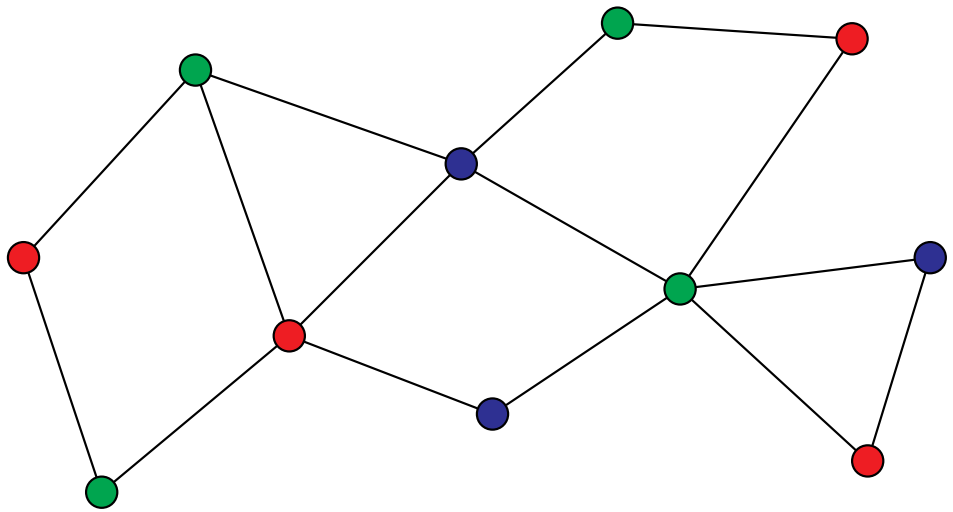
$k = 3$

Příklady problémů

Problém „Barvení grafu k barvami“

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Je možné obarvit vrcholy grafu G k barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?



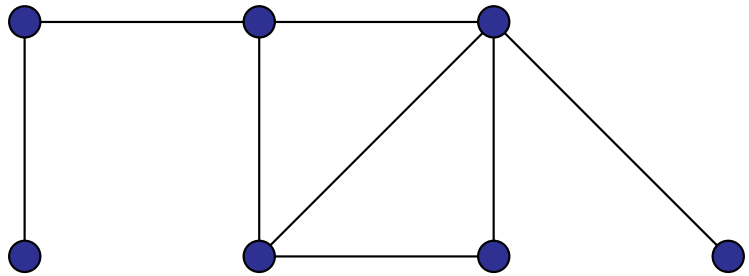
$k = 3$

Příklady problémů

Problém nezávislé množiny (IS)

Vstup: Neorientovaný graf G, číslo k.

Otázka: Existuje v grafu G nezávislá množina velikosti k?



$k = 4$

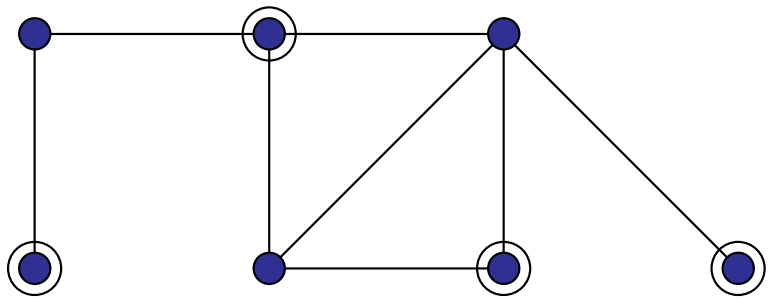
Poznámka: **Nezávislá množina** v grafu je podmnožina vrcholů grafu taková, že žádné dva vrcholy z této podmnožiny nejsou spojeny hranou.

Příklady problémů

Problém nezávislé množiny (IS)

Vstup: Neorientovaný graf G, číslo k.

Otázka: Existuje v grafu G nezávislá množina velikosti k?

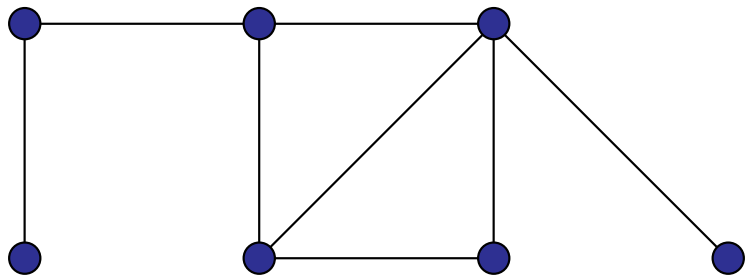


$k = 4$

Poznámka: **Nezávislá množina** v grafu je podmnožina vrcholů grafu taková, že žádné dva vrcholy z této podmnožiny nejsou spojeny hranou.

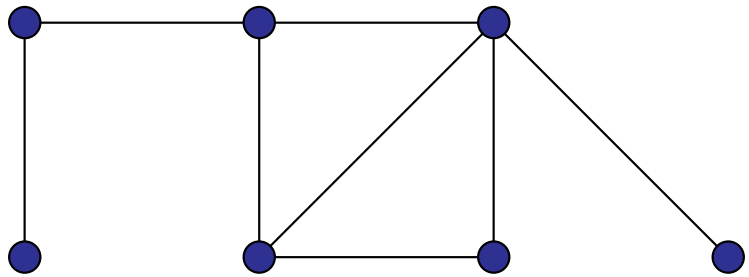
Problém nezávislé množiny (IS)

Příklad instance, kde je odpověď ANO:



$k = 4$

Příklad instance, kde je odpověď NE:



$k = 5$

ILP – celočíselné lineární programování

Problém ILP (celočíselné lineární programování)

Vstup: Celočíselná matice A a celočíselný vektor b.

Otázka: Existuje celočíselný vektor x, takový že $Ax \leq b$?

Příklad instance problému:

$$A = \begin{pmatrix} 3 & -2 & 5 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \qquad b = \begin{pmatrix} 8 \\ -3 \\ 5 \end{pmatrix}$$

Ptáme se tedy, zda existuje celočíselné řešení následující soustavy nerovnic:

$$\begin{aligned} 3x_1 - 2x_2 + 5x_3 &\leq 8 \\ x_1 + x_3 &\leq -3 \\ 2x_1 + x_2 &\leq 5 \end{aligned}$$

ILP – celočíselné lineární programování

Jedním z řešení soustavy

$$\begin{aligned} 3x_1 - 2x_2 + 5x_3 &\leq 8 \\ x_1 + x_3 &\leq -3 \\ 2x_1 + x_2 &\leq 5 \end{aligned}$$

je například $x_1 = -4$, $x_2 = 1$, $x_3 = 1$, tj.

$$x = \begin{pmatrix} -4 \\ 1 \\ 1 \end{pmatrix}$$

neboť

$$\begin{aligned} 3 \cdot (-4) - 2 \cdot 1 + 5 \cdot 1 &= -9 \leq 8 \\ -4 + 1 &= -3 \leq -3 \\ 2 \cdot (-4) + 1 &= -7 \leq 5 \end{aligned}$$

Pro tuto instanci je tedy odpověď Ano.

Příklady problémů

Problém

Vstup: Deterministické konečné automaty A_1 a A_2.

Otázka: Je $L(A_1) = L(A_2)$?

Problém

Vstup: Bezkontextové gramatiky G_1 a G_2.

Otázka: Je $L(G_1) = L(G_2)$?

Řešení problémů

Řešení problému

Algoritmus **řeší** daný problém, když:

1. Se pro libovolný vstup daného problému (libovolnou vstupní instanci) po konečném počtu kroků zastaví.
2. Vyprodukuje výstup z množiny možných výstupů, který vyhovuje podmínkám uvedeným v zadání problému.

Pro jeden problém může existovat celá řada algoritmů, které jej korektně řeší.

Poznámka: **korektnost algoritmu** — algoritmus řeší daný problém

Řešení problémů

Každému algoritmu A můžeme přiřadit funkci

$$f_A : In \to Out$$

kde:

- In je množina vstupů pro algoritmus A,
- Out je množina výstupů generovaných algoritmem A,
- $f_A(x)$ je výstup, který algoritmus A vygeneruje pro vstup $x \in In$.

Funkce f_A nemusí být **totální** (tj. hodnota $f_A(x)$ nemusí být definovaná pro každé $x \in In$), ale může být **častečná** (**parciální**):

- hodnota $f_A(x)$ není definována, pokud se výpočet algoritmu A pro vstup x nikdy nezastaví, pokud během výpočtu dojde k chybě apod.

Řešení problémů

Pokud tedy máme dán nějaký problém $P = (In, Out, R)$ a nějaký algoritmus A realizující funkci $f_A : In \to Out$, pak řekneme, že

$$\textit{algoritmus } A \textit{ řeší problém } P$$

jestliže:

- hodnota $f_A(x)$ je definovaná pro každé $x \in In$,
- pro každé $x \in In$ platí $(x, f_A(x)) \in R$

Algoritmicky řešitelné problémy

Předpokládejme, že máme dán nějaký problém P.

Jestliže existuje nějaký algoritmus, který řeší problém P, pak říkáme, že problém P je **algoritmicky řešitelný**.

Jestliže P je rozhodovací problém a jestliže existuje nějaký algoritmus, který problém P řeší, pak říkáme, že problém P je **(algoritmicky) rozhodnutelný**.

Když chceme ukázat, že problém P je algoritmicky řešitelný, stačí ukázat nějaký algoritmus, který ho řeší (a případně ukázat, že daný algoritmus problém P skutečně řeší).

Algoritmicky neřešitelné problémy

Problém, který není algoritmicky řešitelný, je **algoritmicky neřešitelný**.

Rozhodovací problém, který není rozhodnutelný, je **nerozhodnutelný**.

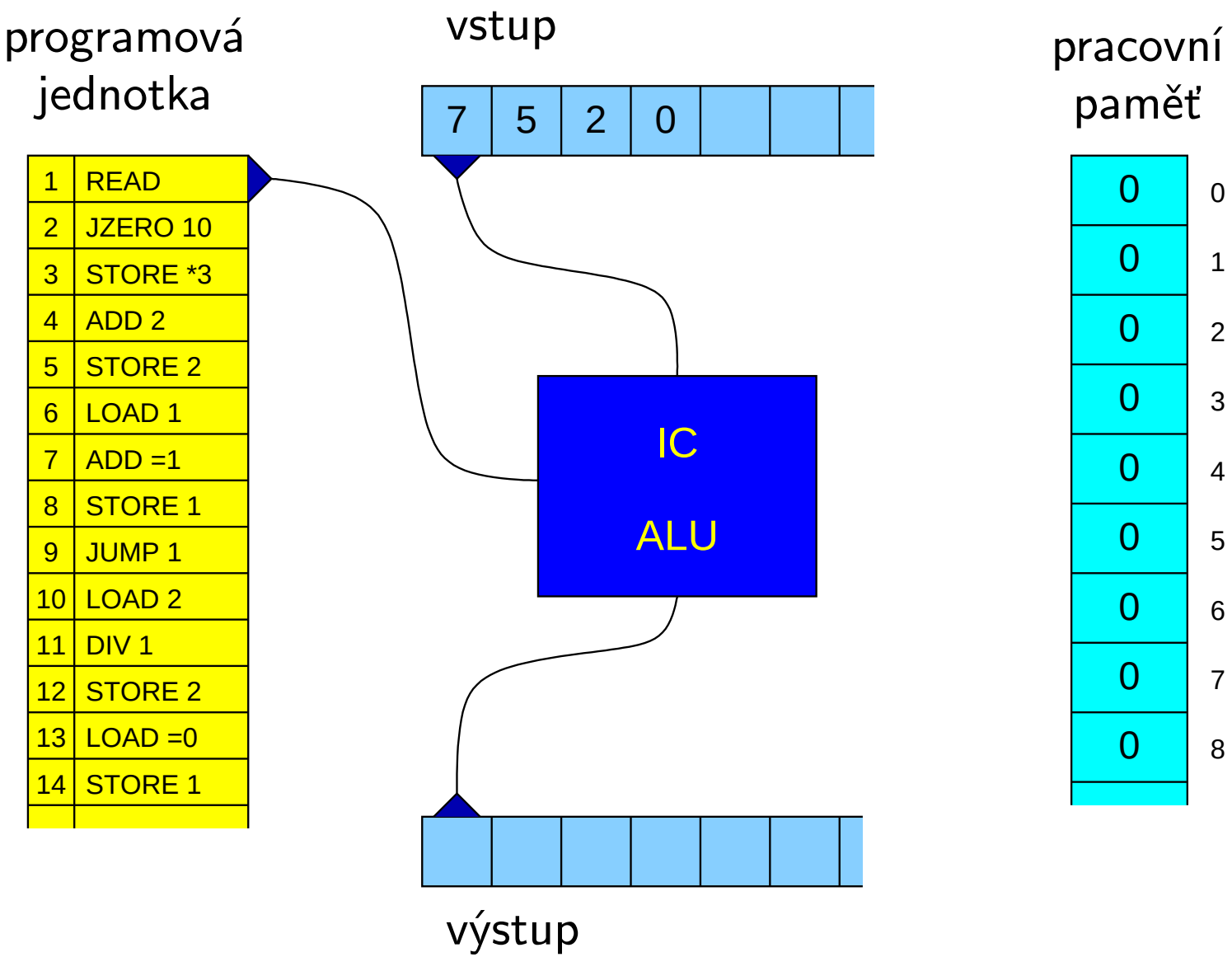
Stroj RAM

Stroj RAM (Random Access Machine) je idealizovaný model počítače.

Skládá se z těchto částí:

- **Programová jednotka** – obsahuje program stroje RAM a ukazatel na právě prováděnou instrukci
- **Pracovní paměť** tvořená buňkami očíslovanými $0, 1, 2, \ldots$; obsah buněk je možno číst i do nich zapisovat
- **Vstupní páska** – je z ní možné pouze číst
- **Výstupní páska** – je na ni možno pouze zapisovat

Stroj RAM


programová jednotka
vstup
pracovní paměť
1 READ
2 JZERO 10
3 STORE *3
4 ADD 2
5 STORE 2
6 LOAD 1
7 ADD =1
8 STORE 1
9 JUMP 1
10 LOAD 2
11 DIV 1
12 STORE 2
13 LOAD =0
14 STORE 1
7 5 2 0
IC
ALU
0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8
výstup

Stroj RAM

Buňky 0 a 1 mají speciální význam a slouží jako „registry“ stroje RAM:

- **Buňka 0** – **pracovní registr** (akumulátor) – registr, který je jedním z operandů většiny instrukcí a do kterého se ukládá výsledek většiny operací.
- **Buňka 1** – **indexový registr** – je použit při nepřímém adresování.

Tvary **operandů** instrukcí ($i \in \mathbb{N}$):

tvar	hodnota operandu
$=i$	přímo číslo udané zápisem i
i	číslo obsažené v buňce s adresou i
$*i$	číslo v buňce s adresou $i + j$, kde j je aktuální obsah indexového registru

Příklad:

`LOAD <op>`

načte obsah operandu `<op>` do pracovního registru (tj. do buňky číslo 0).

`LOAD =5`	–	uloží do pracovního registru hodnotu 5
`LOAD 5`	–	uloží do pracovního registru obsah buňky číslo 5
`LOAD *5`	–	uloží do pracovního registru obsah buňky číslo $5+j$, kde j je aktuální obsah indexového registru

Stroj RAM

Instrukce vstupu a výstupu (jsou bez operandu):

`READ` – do pracovního registru se uloží číslo, které je v políčku snímaném vstupní hlavou, a vstupní hlava se posune o jedno políčko doprava

`WRITE` – výstupní hlava zapíše do snímaného políčka výstupní pásky obsah pracovního registru a posune se o jedno políčko doprava

Instrukce přesunu v paměti:

`LOAD <op>` – do pracovního registru se načte hodnota operandu

`STORE <op>` – hodnota operandu se přepíše obsahem pracovního registru (zde se nepřipouští operand tvaru $=i$)

Stroj RAM

Instrukce aritmetických operací:

- `ADD <op>` – číslo v pracovním registru se zvýší o hodnotu operandu (tedy přičte se k němu hodnota operandu)
- `SUB <op>` – od čísla v pracovním registru se odečte hodnota operandu
- `MUL <op>` – číslo v pracovním registru se vynásobí hodnotou operandu
- `DIV <op>` – číslo v pracovním registru se celočíselně vydělí hodnotou operandu (do pracovního registru se uloží výsledek příslušného celočíselného dělení)

Stroj RAM

Instrukce skoku:

`JUMP <návěští>` – výpočet bude pokračovat instrukcí určenou návěštím

`JZERO <návěští>` – je-li obsahem pracovního registru číslo 0, bude výpočet pokračovat instrukcí určenou návěštím; v opačném případě bude pokračovat následující instrukcí

`JGTZ <návěští>` – je-li číslo v pracovním registru kladné, bude výpočet pokračovat instrukcí určenou návěštím; v opačném případě bude pokračovat následující instrukcí

Instrukce zastavení:

`HALT` – výpočet je ukončen („regulérně“ zastaven)

Stroj RAM

Problém „Vyhledávání“

Vstup: Celé číslo x a sekvence celých čísel $a_1, a_2, \ldots, a_n$ (kde $a_i \neq 0$) ukončená 0.

Výstup: Pokud $a_i = x$, je výstupem i (pokud jich je takových i více, tak nejmenší z nich), jinak je výstupem 0.

```
start:  READ
        STORE  3
        LOAD   =1
cyklus: STORE  2
        READ
        JZERO  vypis
        SUB    3
        JZERO  nasel
        LOAD   2
        ADD    =1
        JUMP   cyklus
nasel:  LOAD   2
vypis:  WRITE
        HALT
```

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 0
Buňka 3: 0
Buňka 4: 0
⋮

Výstup:

Instrukcí: 0

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 0
Buňka 3: 0
Buňka 4: 0
⋮

Výstup:

Instrukcí: 0

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 0
Buňka 3: 0
Buňka 4: 0
⋮

Výstup:

Instrukcí: 1

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 0
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 2

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 1
Buňka 1: 0
Buňka 2: 0
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 3

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 1
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 4

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 13
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 5

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 13
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 6

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 4
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 7

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 4
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 8

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 1
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 9

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 10

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 11

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 12

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 5
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 13

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 5
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 14

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: -4
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 15

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: -4
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 16

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 17

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 18

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 19

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 20

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 21

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 22

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 23

Stroj RAM

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 24

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 25

Stroj RAM

```
 start:   READ
          STORE  3
          LOAD   =1
cyklus:   STORE  2
          READ
          JZERO  vypis
          SUB    3
          JZERO  nasel
          LOAD   2
          ADD    =1
          JUMP   cyklus
 nasel:   LOAD   2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup: 3

Instrukcí: 26

Stroj RAM

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

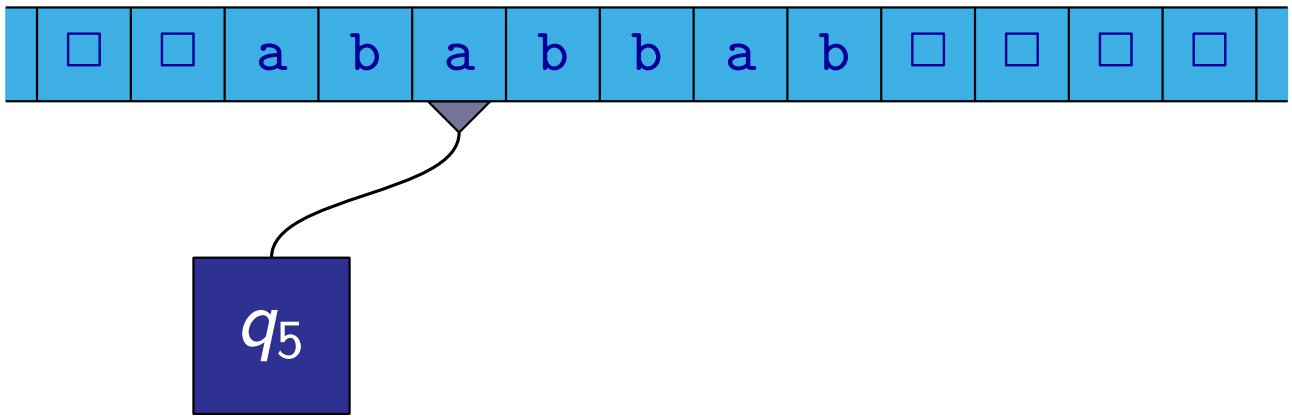
Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup: 3

Instrukcí: 27

Turingův stroj

- Rozšíříme deterministický konečný automat o pohyb čtecí hlavy oběma směry, možnost zápisu na vstupní pásku, a prodloužíme jeho pásku do nekonečna.



Turingův stroj

Definice

Formálně je **Turingův stroj** definován jako šestice $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ kde:

- Q je konečná neprázdná množina **stavů**
- Γ je konečná neprázdná množina **páskových symbolů** (**pásková abeceda**)
- $\Sigma \subseteq \Gamma$ je konečná neprázdná množina **vstupních symbolů** (**vstupní abeceda**)
- $\delta : (Q - F) \times \Gamma \to Q \times \Gamma \times \{-1, 0, +1\}$ je **přechodová funkce**
- $q_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **koncových stavů**

Turingův stroj

- Předpokládáme, že v $\Gamma - \Sigma$ je vždy speciální prvek $\square$ označující prázdný znak.
- **Konfigurace** je dána:
  - stavem řídící jednotky
  - obsahem pásky
  - pozicí hlavy
- Výpočet nad slovem $w \in \Sigma^*$ začíná v **počáteční konfiguraci**, kde:
  - stav řídící jednotky je q_0
  - na pásce je zapsáno slovo w, zbývající políčka pásky jsou vyplněna prázdnými symboly ($\square$)
  - hlava se nachází na prvním symbolu slova w (nebo na symbolu $\square$ pokud je $w = \varepsilon$)

Turingův stroj

Jeden krok Turingova stroje:

Předpokládejme, že:

- stav řídící jednotky je q
- na políčku, kde se právě nachází hlava, je zapsán symbol b

Řekněme, že $\delta(q, b) = (q', b', d)$, kde $d \in \{-1, 0, 1\}$.

Jeden krok Turingova stroje se provede následovně:

- stav řídící jednotky se změní na q'
- na políčko na pozici hlavy se místo symbolu b zapíše symbol b'
- V závislosti na hodnotě d se hlava posune:
  - pro $d = -1$ se posune o jedno políčko doleva
  - pro $d = 1$ se posune o jedno políčko doprava
  - pro $d = 0$ se pozice hlavy nezmění

Turingův stroj

Jestliže stav řídící jednotky patří do množiny F, další krok není definován a výpočet stroje končí.

Často volíme množinu přijímajících stavů $F = \{q_{\text{ano}}, q_{\text{ne}}\}$.

Můžeme pak pro slovo $w \in \Sigma^*$ definovat, zda ho daný Turingův stroj přijímá:

- Pokud je po skončení výpočtu nad slovem w řídící jednotka ve stavu q_{ano}, stroj slovo w přijímá.
- Pokud je po skončení výpočtu nad slovem w řídící jednotka ve stavu q_{ne}, stroj slovo w nepřijímá.
- Výpočet stroje nad slovem w může být nekonečný. V tom případě stroj slovo w nepřijímá.

Jazyk $L(\mathcal{M})$ Turingova stroje $\mathcal{M}$ je množina všech slov, která stroj $\mathcal{M}$ přijímá.

Turingův stroj

- Turingův stroj nemusí dávat jen odpověď ANO nebo NE, ale může realizovat nějakou funkci, která každému slovu ze Σ^* přiřazuje nějaké jiné slovo (z Γ^*).

- Slovo přiřazené slovu w je slovo, které zůstane zapsáno na pásce po výpočtu nad slovem w, když odstraníme všechny znaky $\Box$.

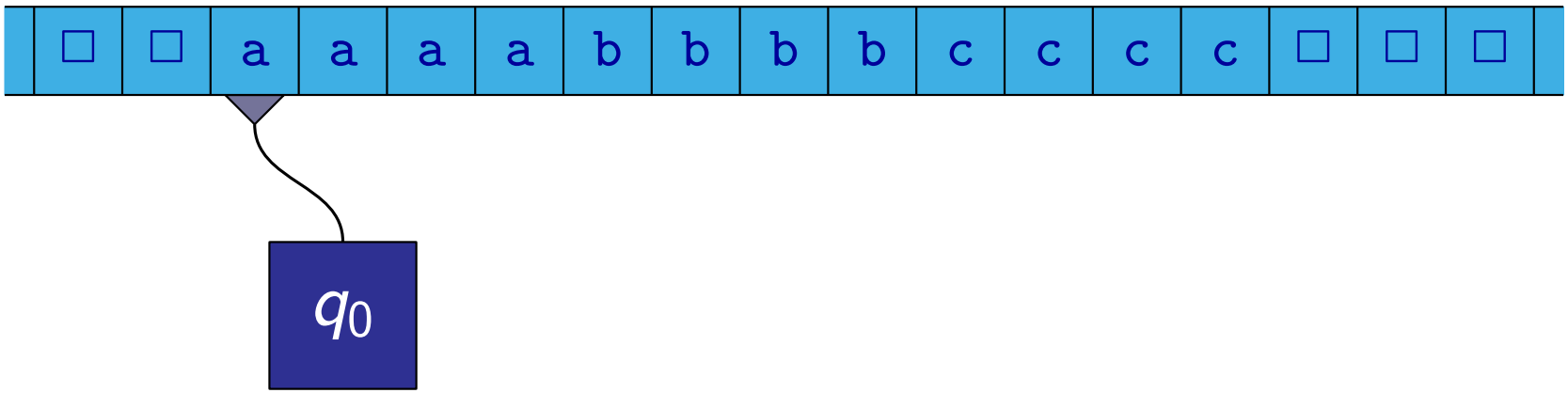
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



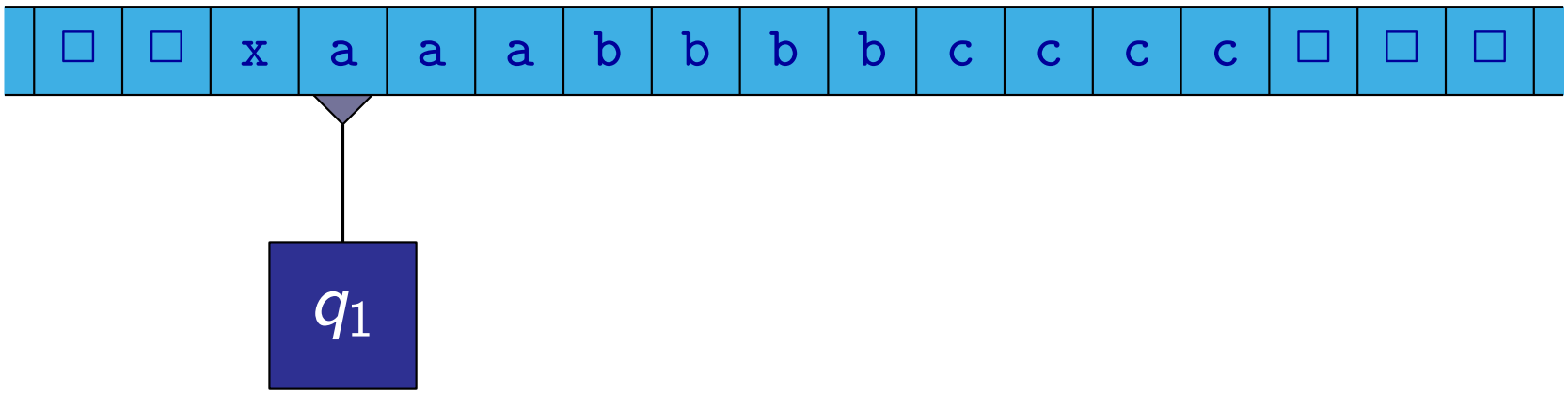
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



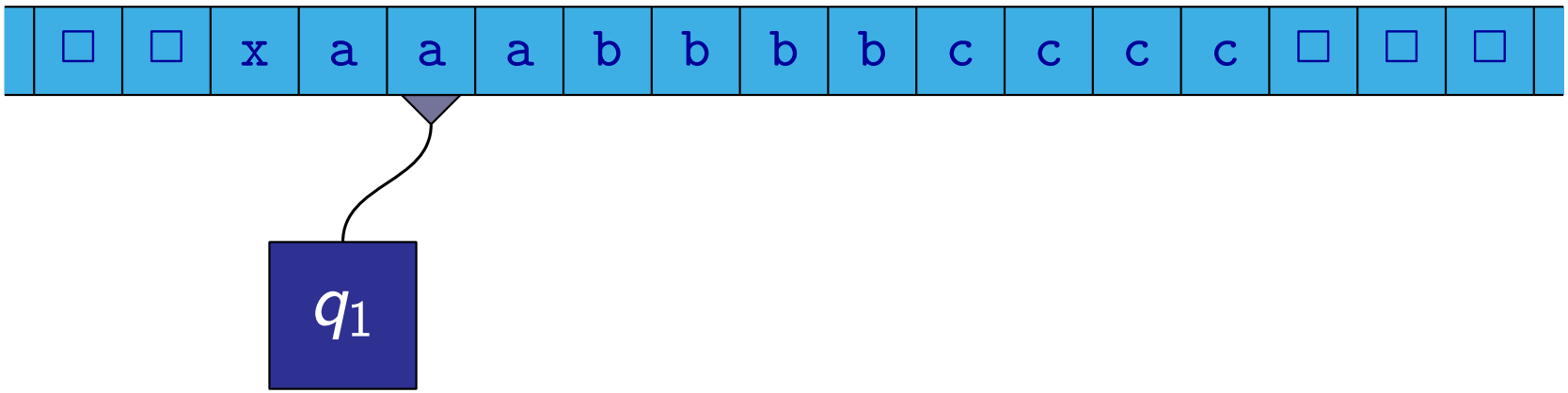
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



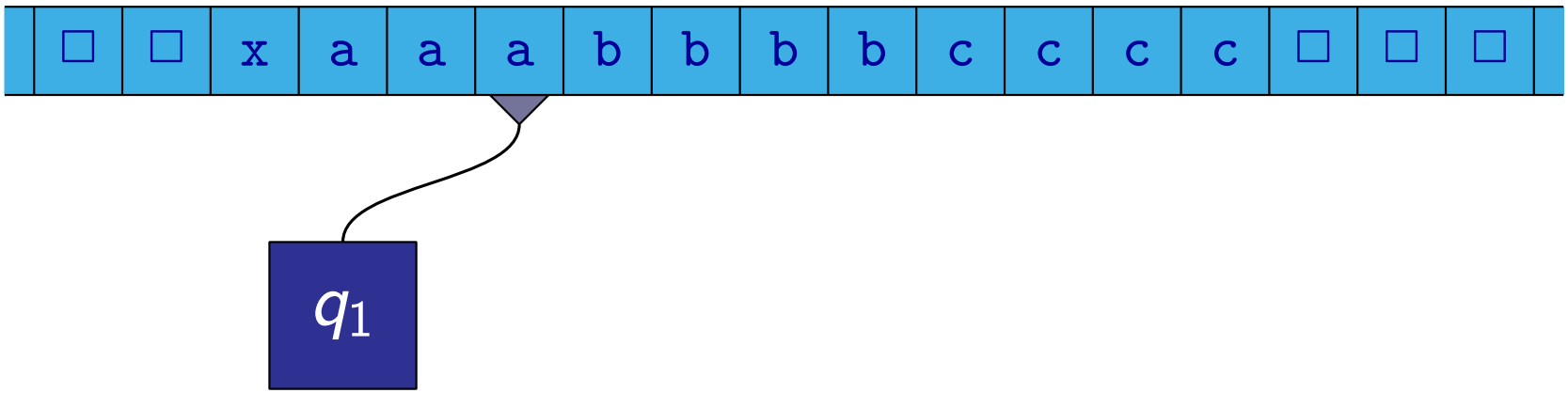
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



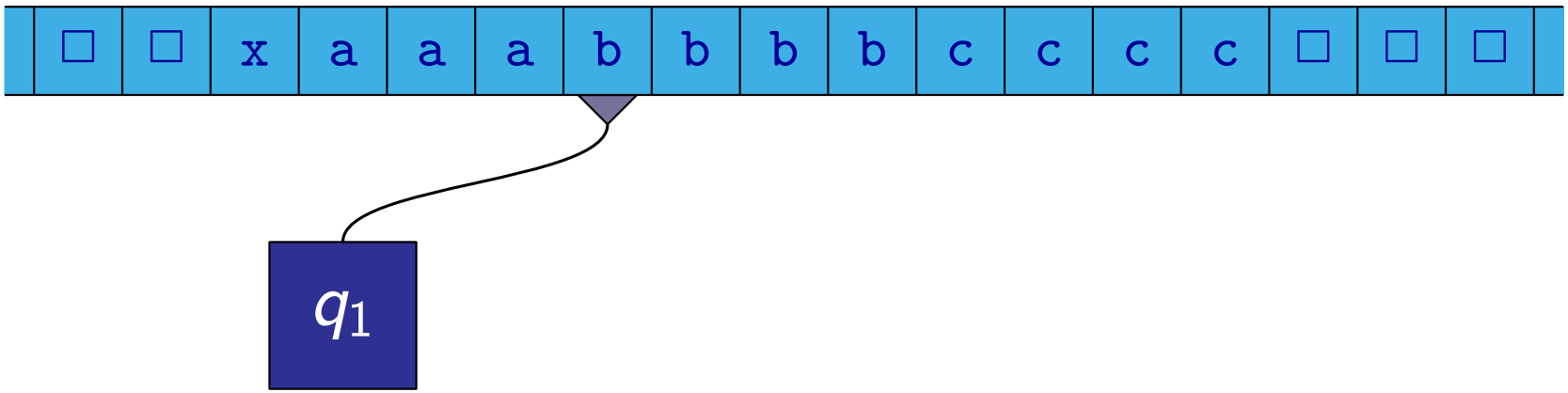
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



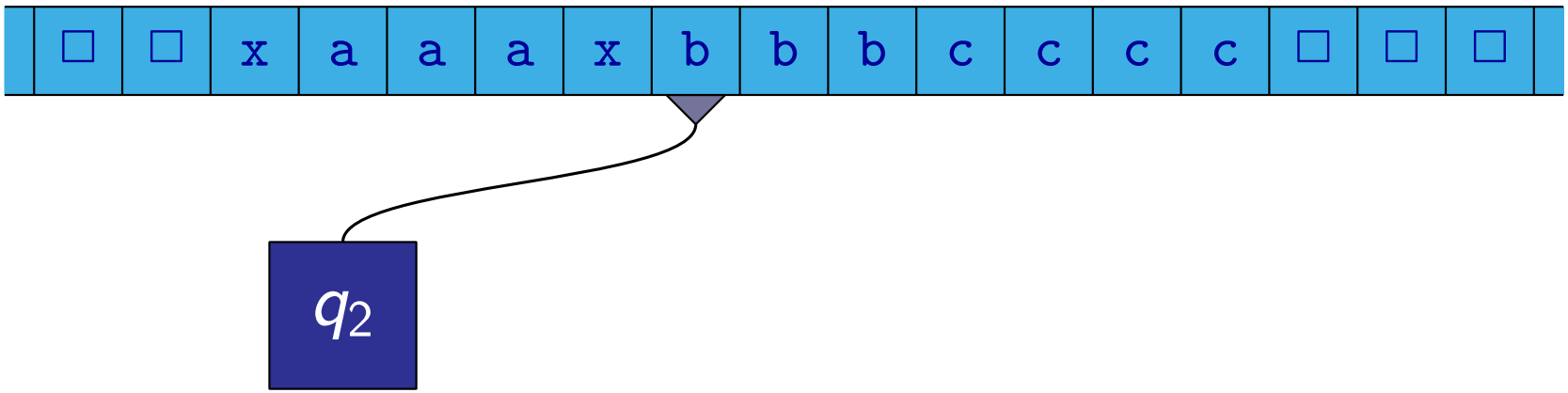
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



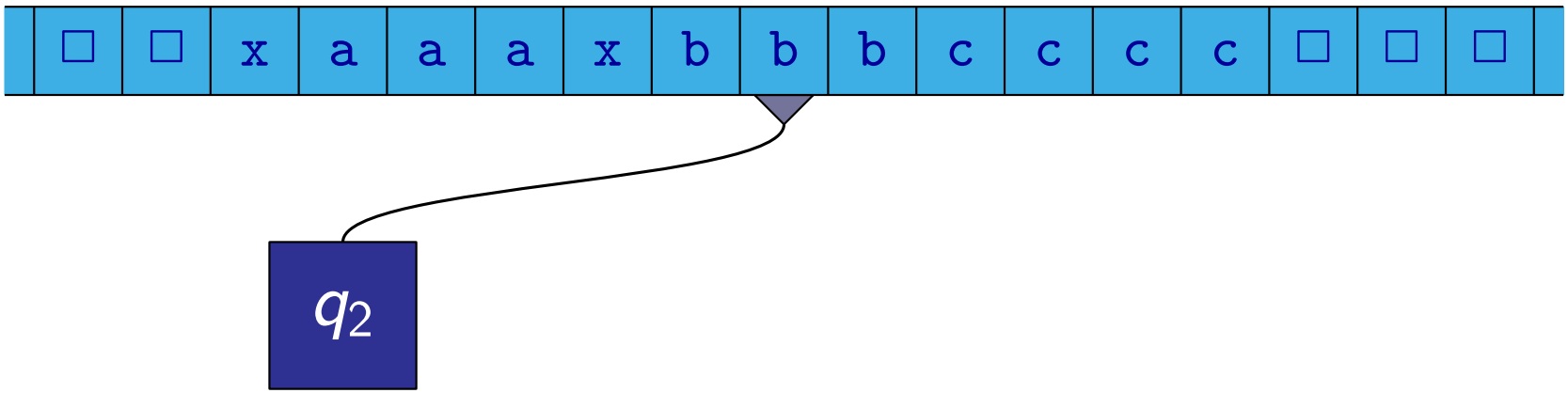
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



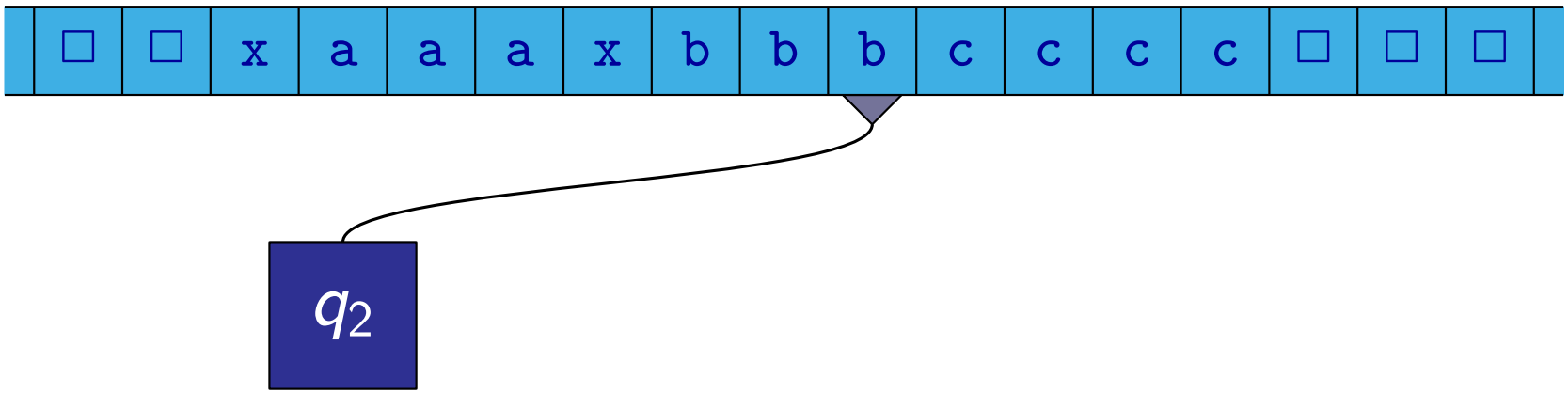
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



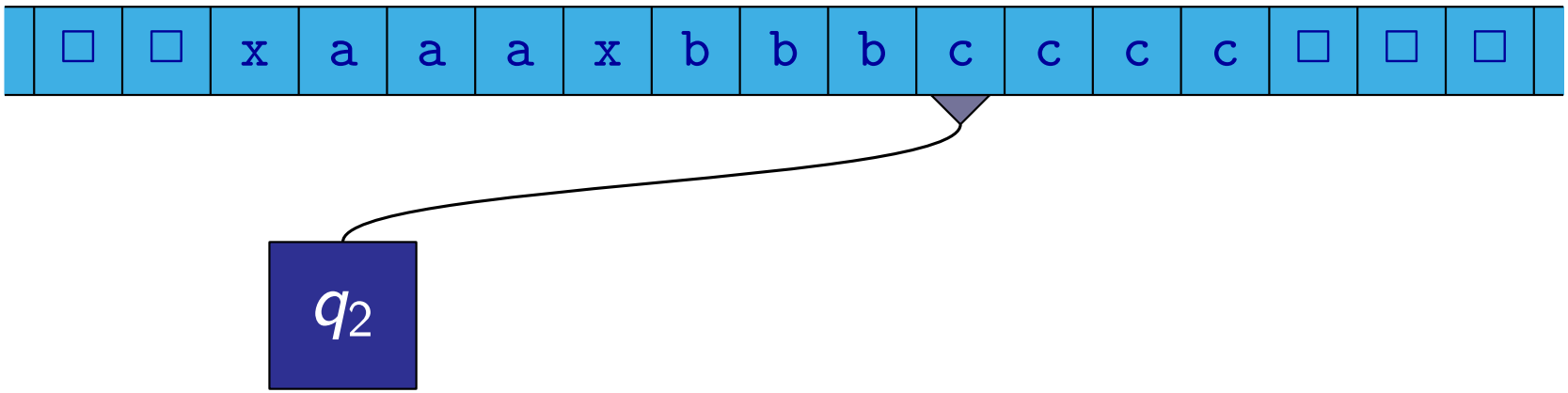
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



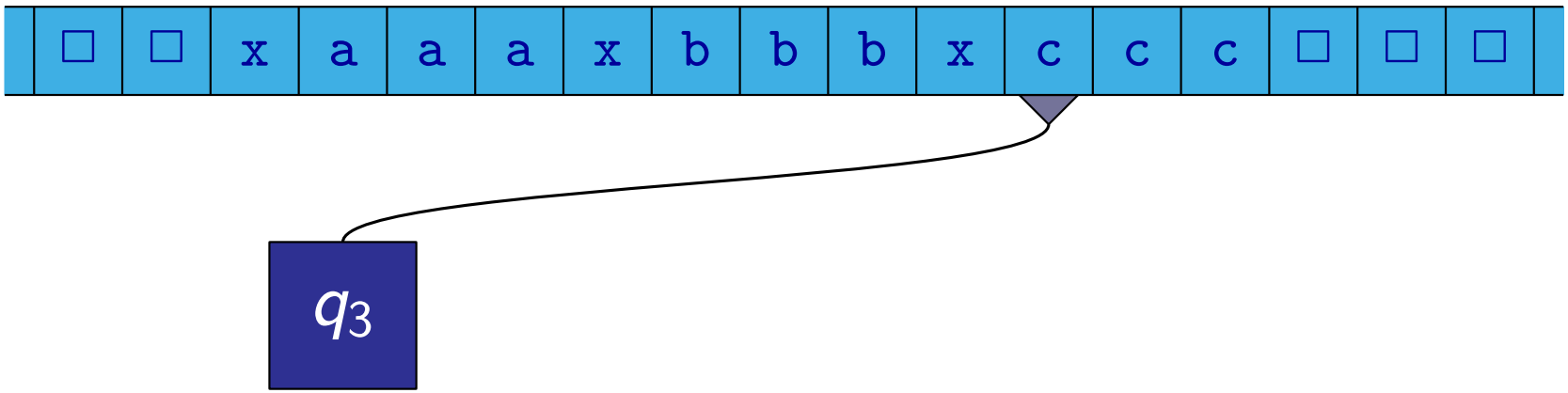
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



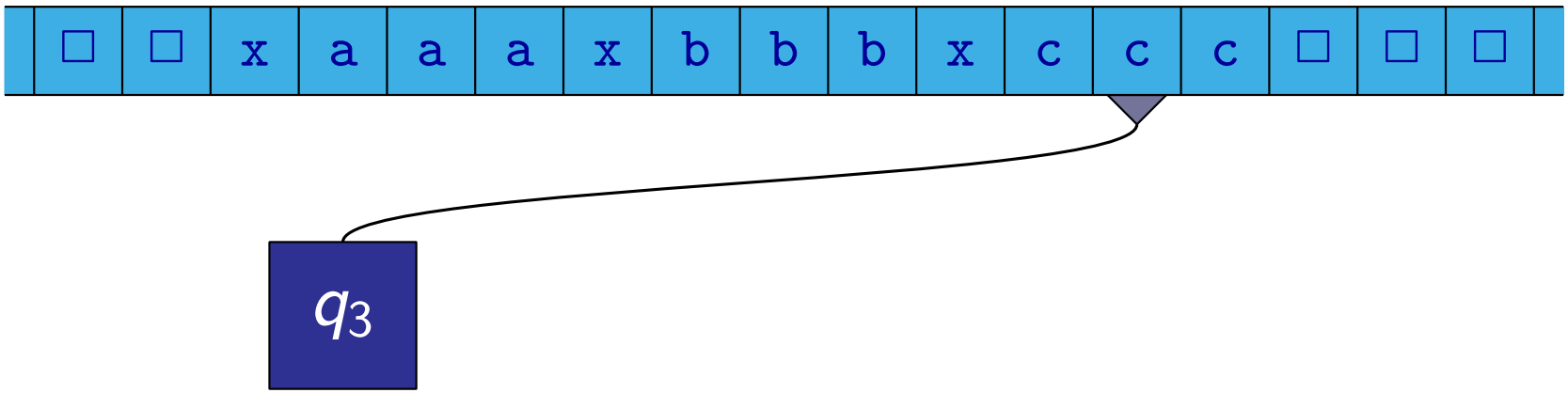
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



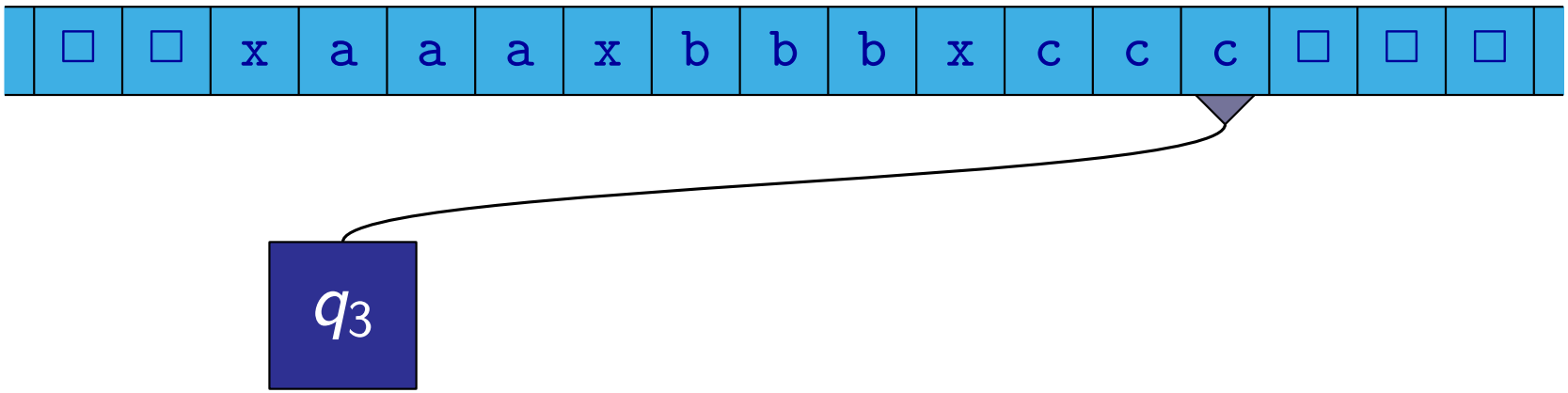
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



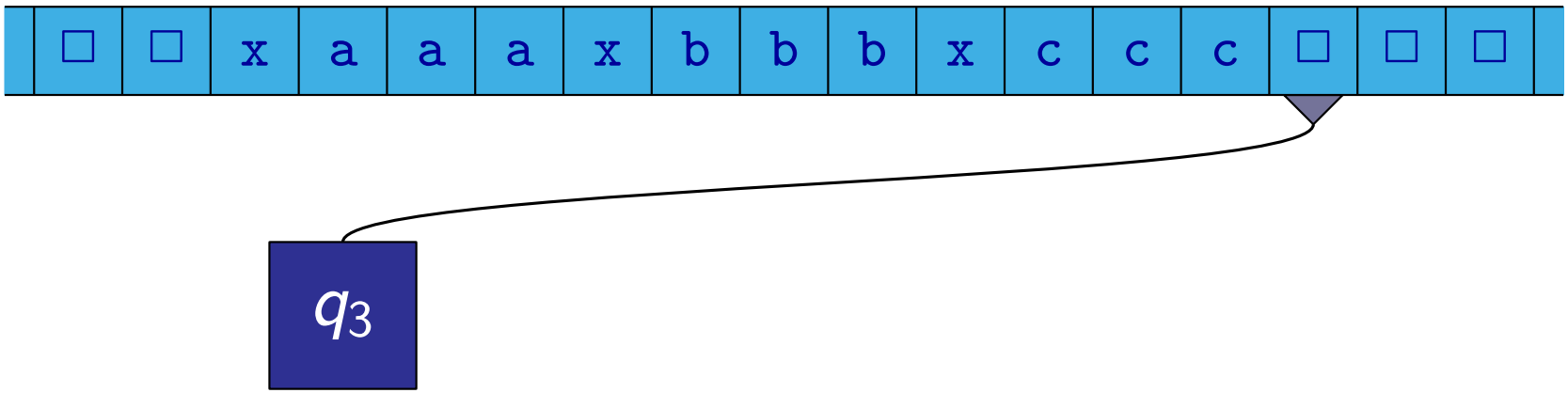
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



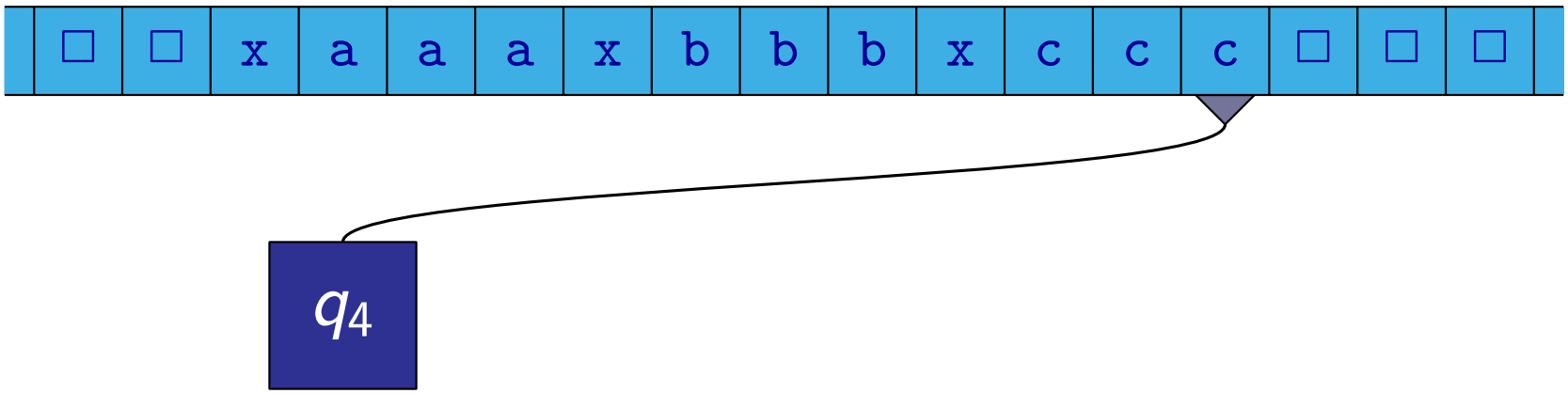
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



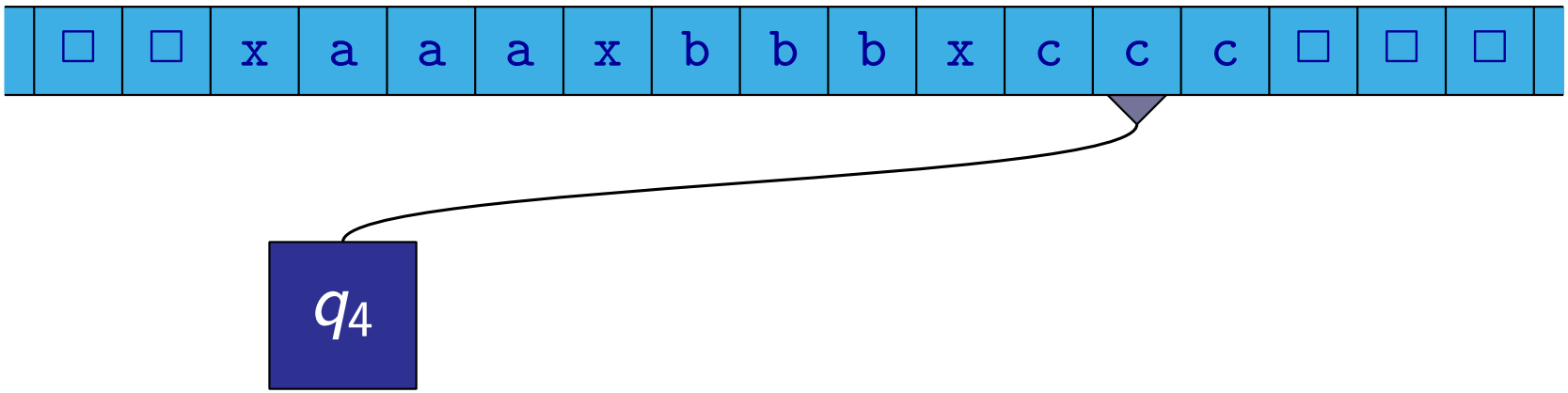
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



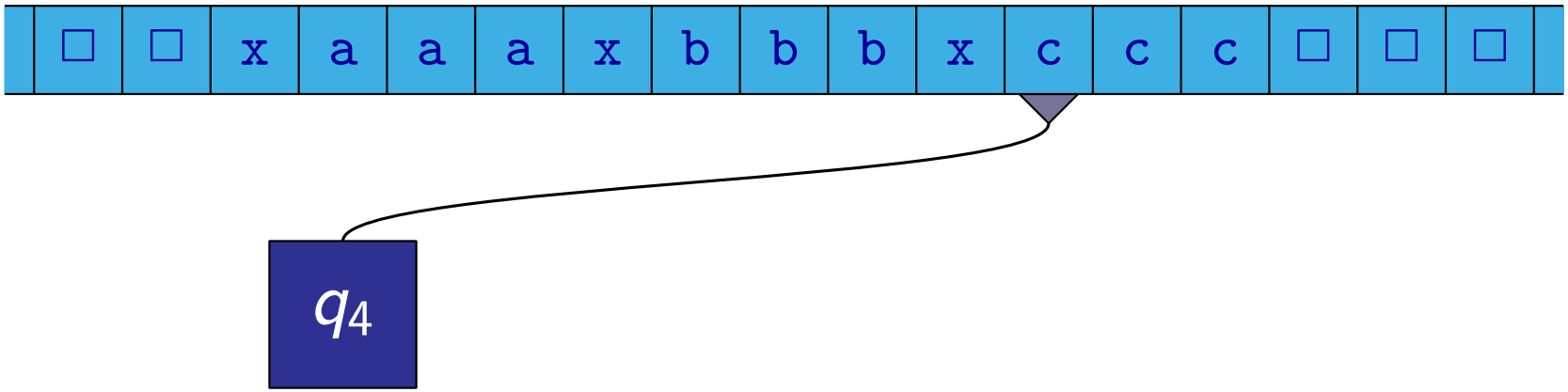
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



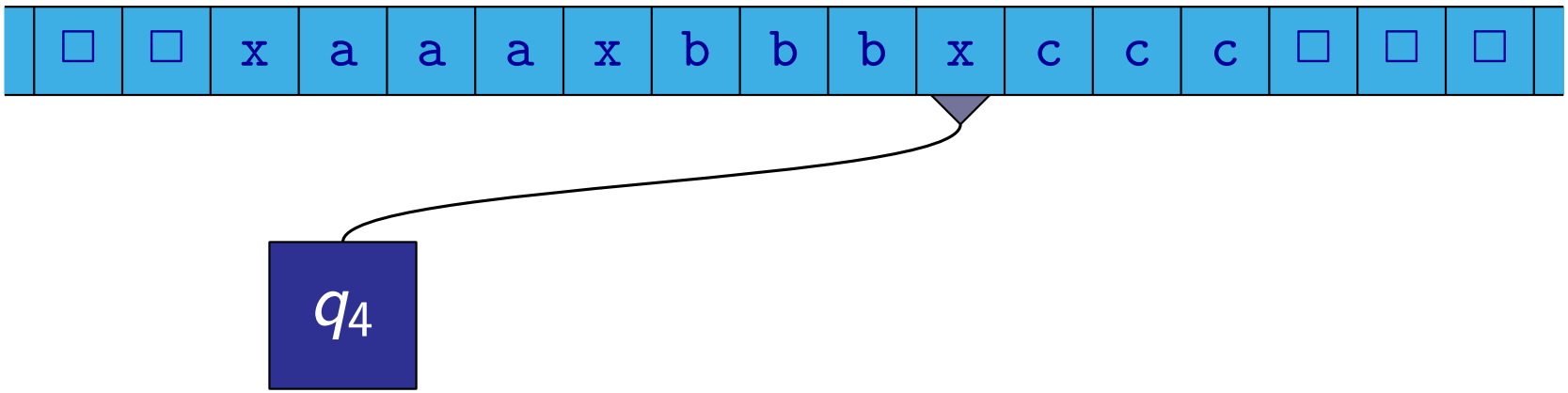
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



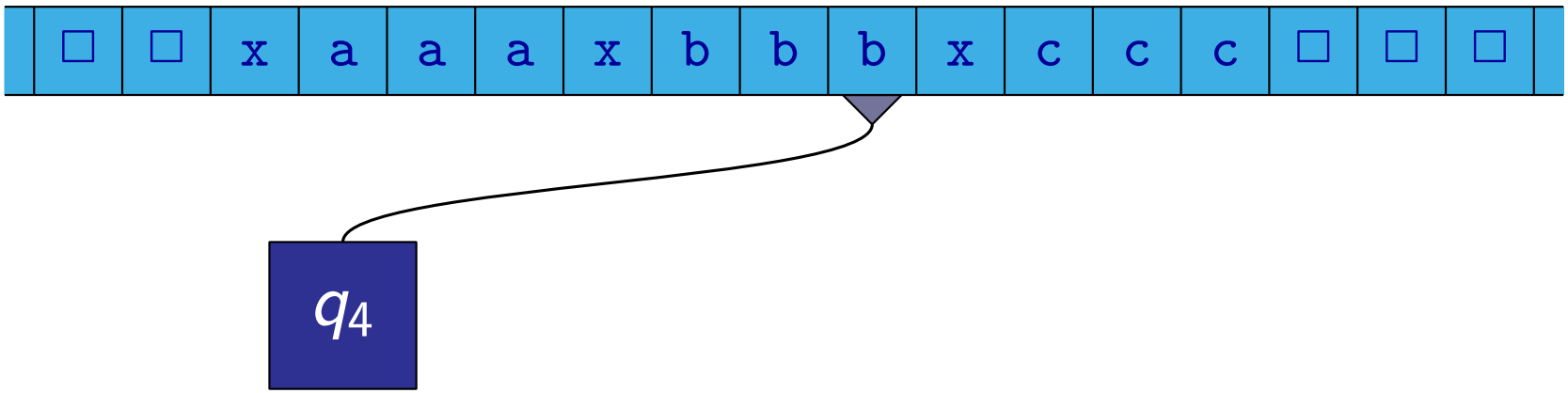
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



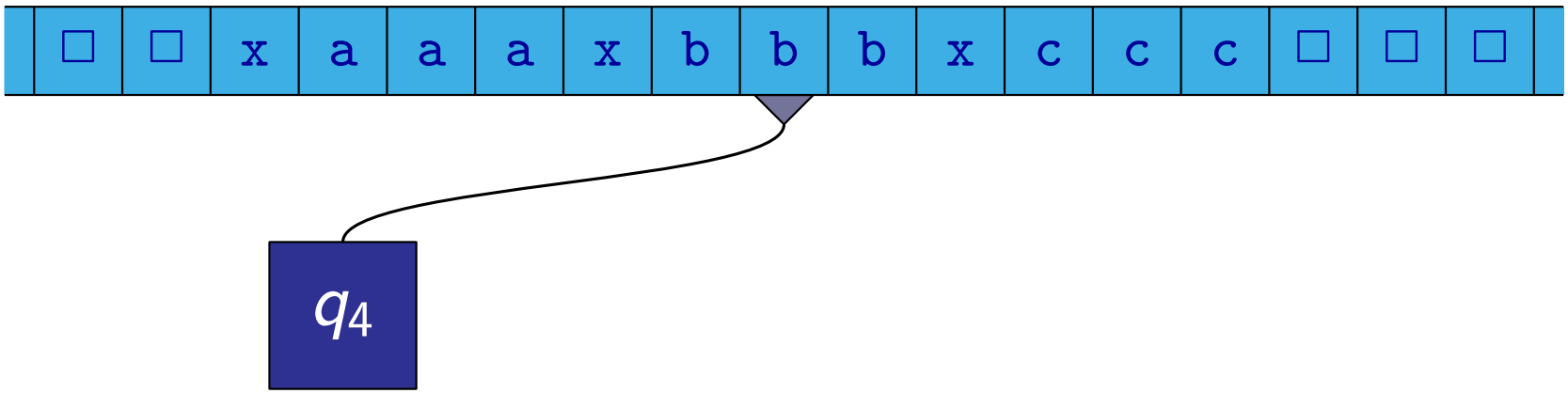
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



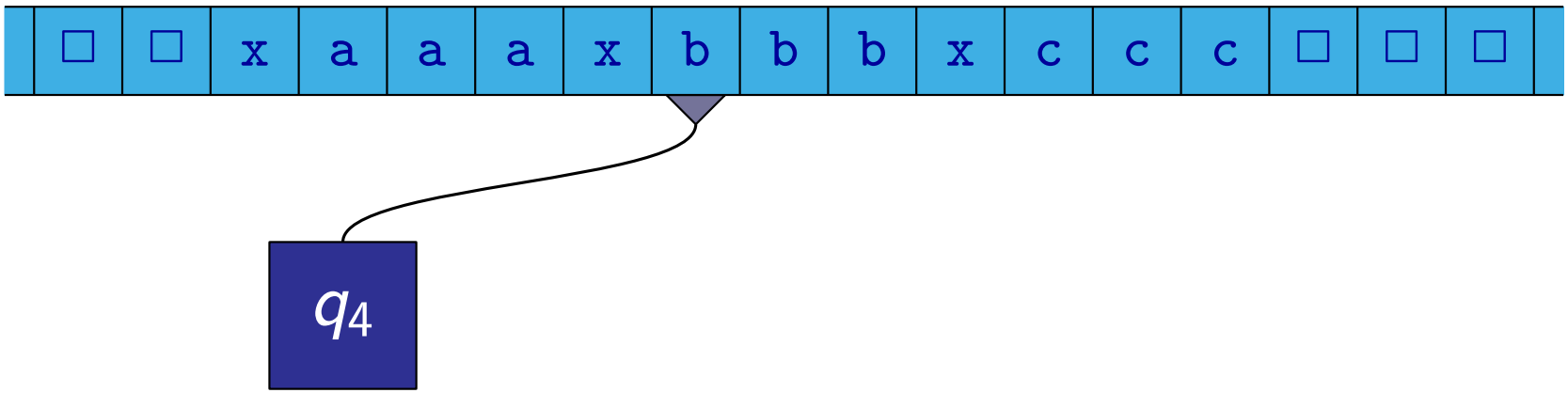
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



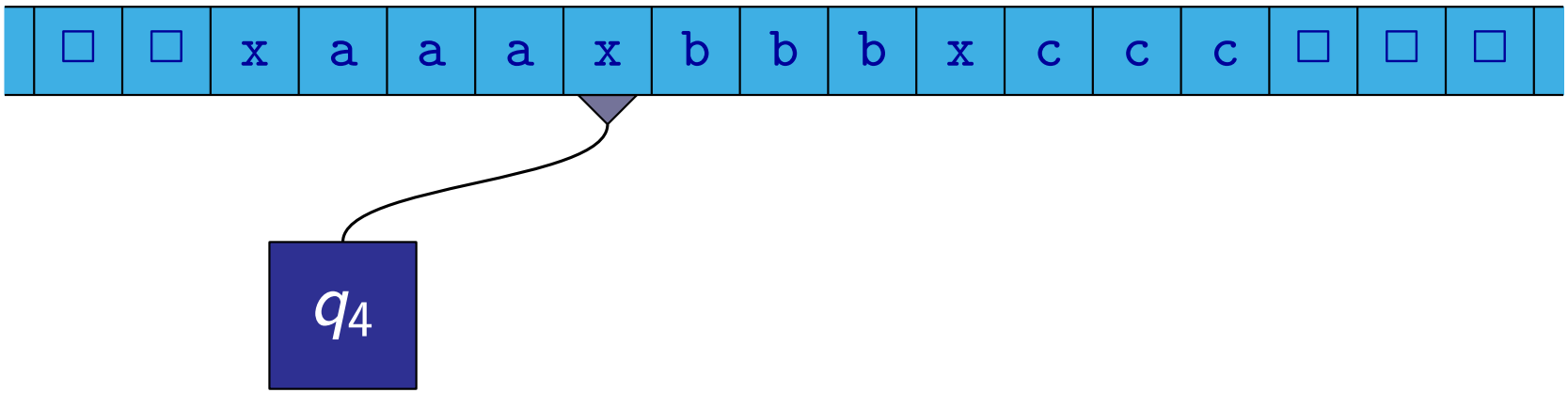
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



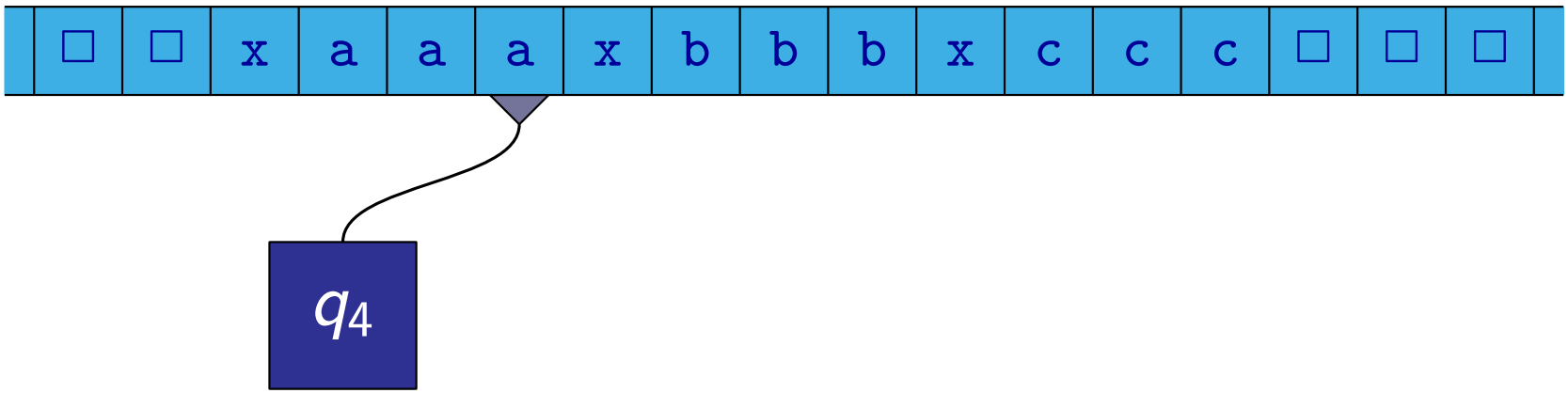
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



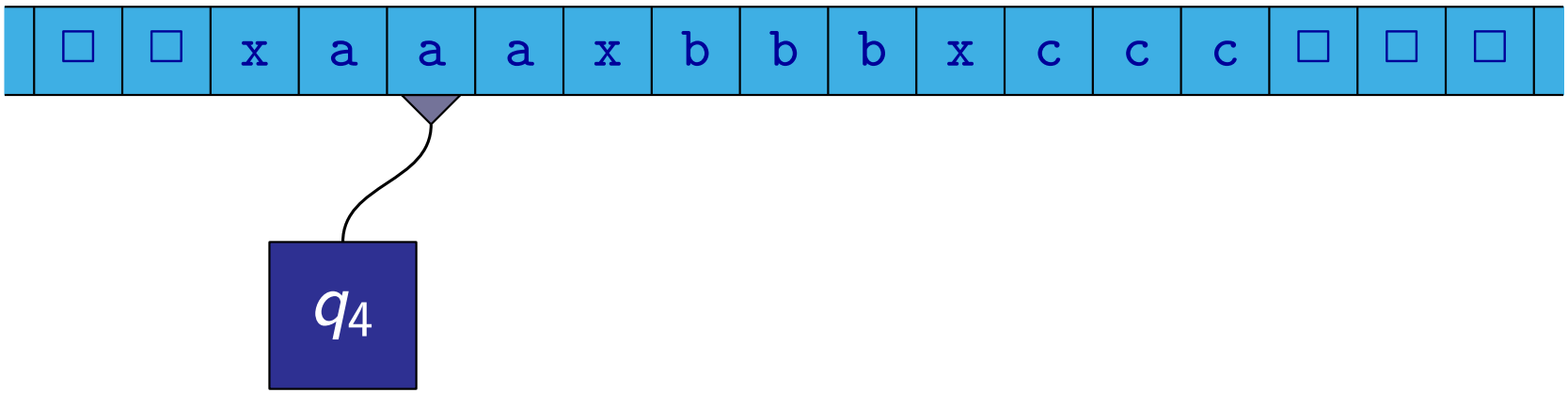
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



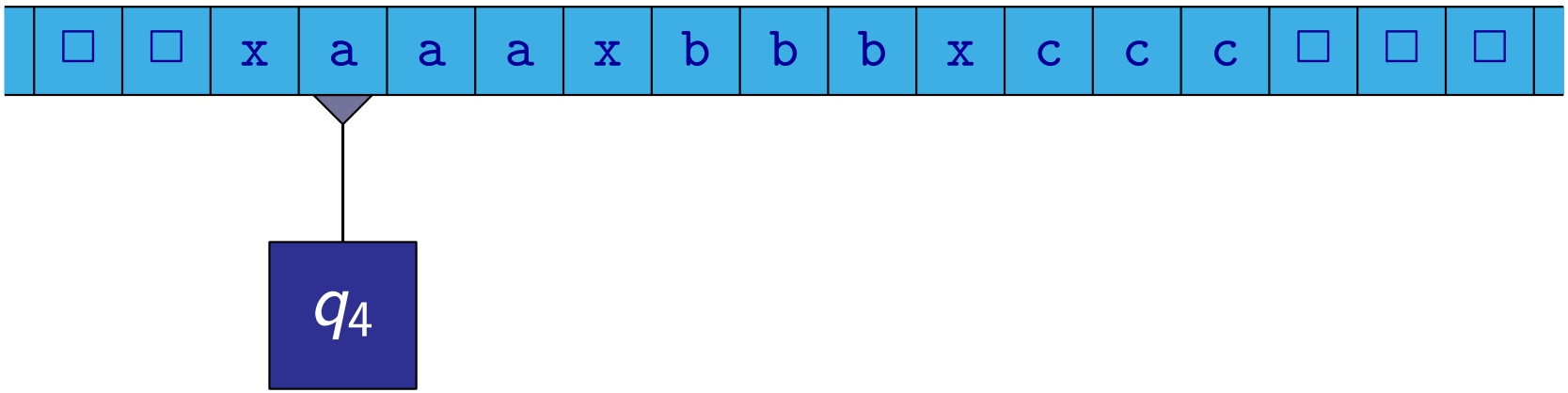
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



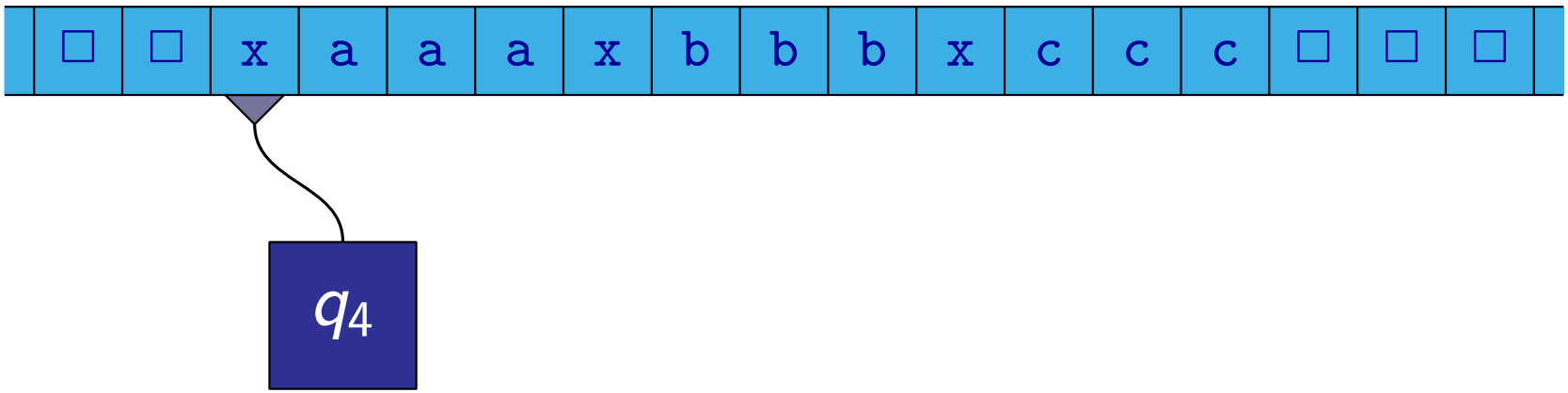
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



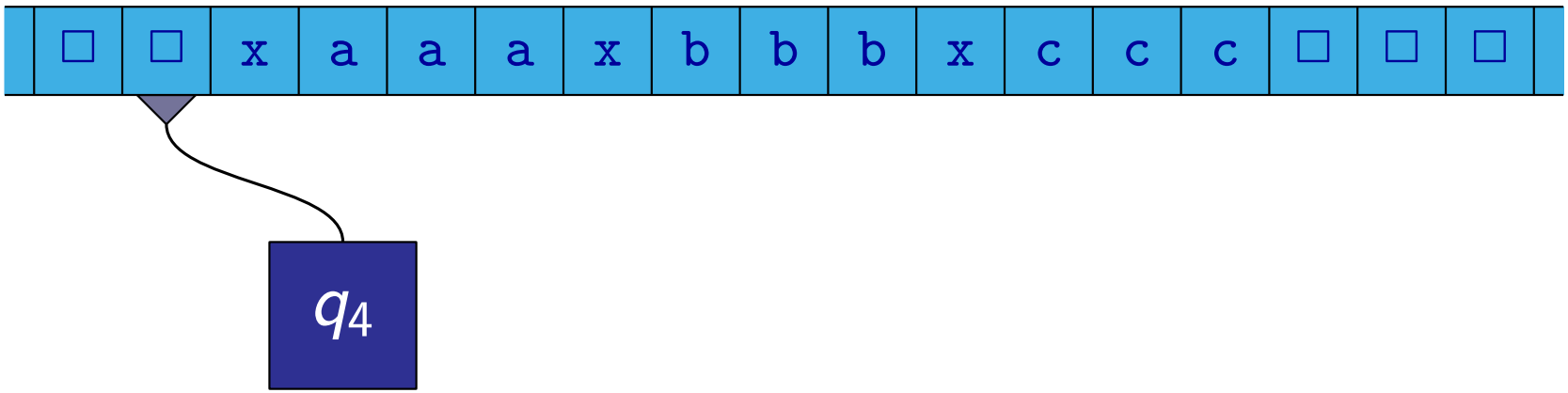
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



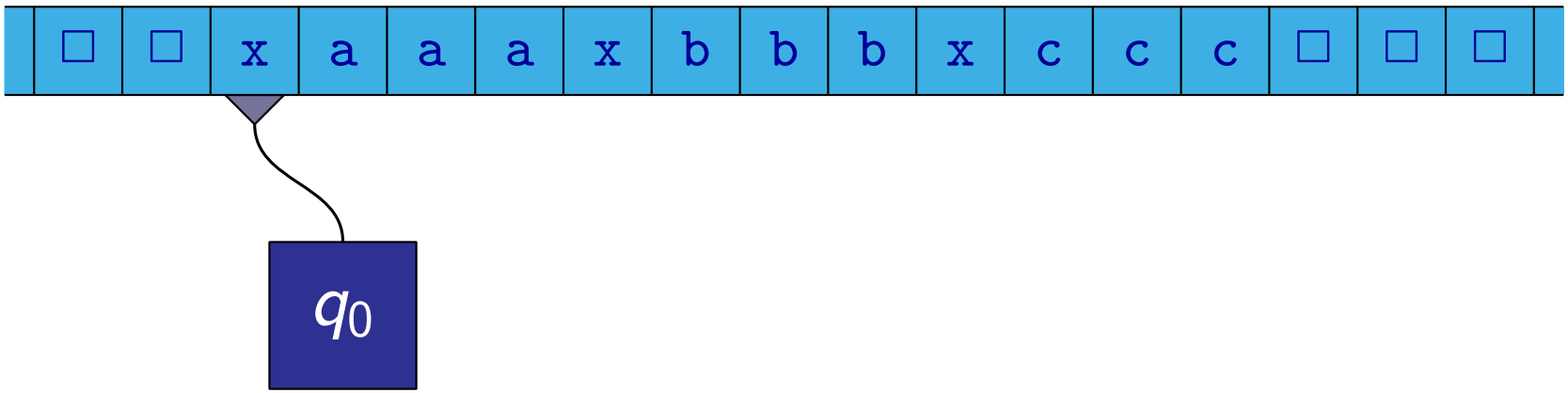
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



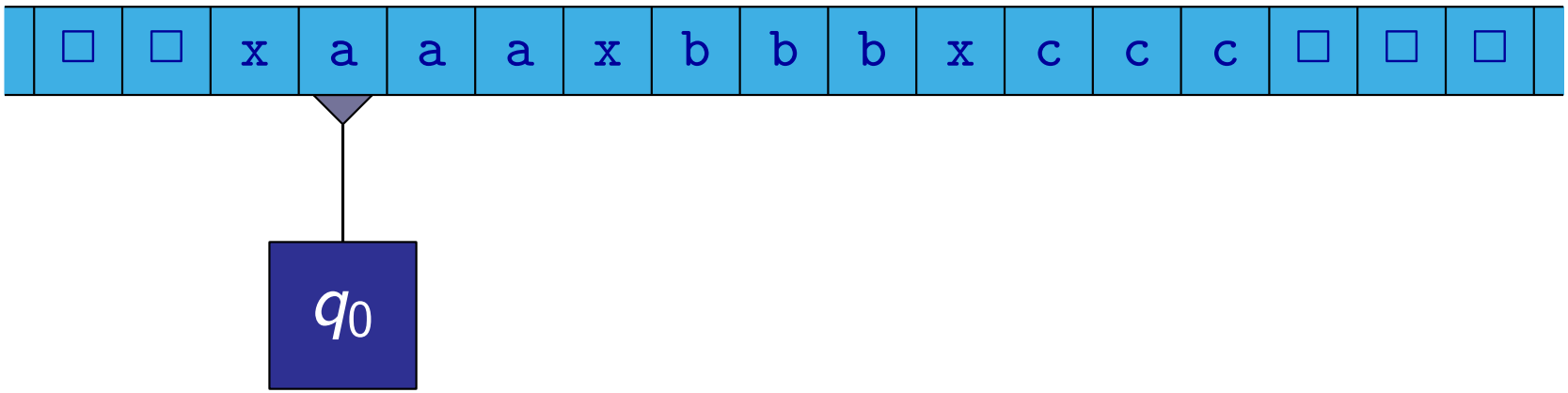
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



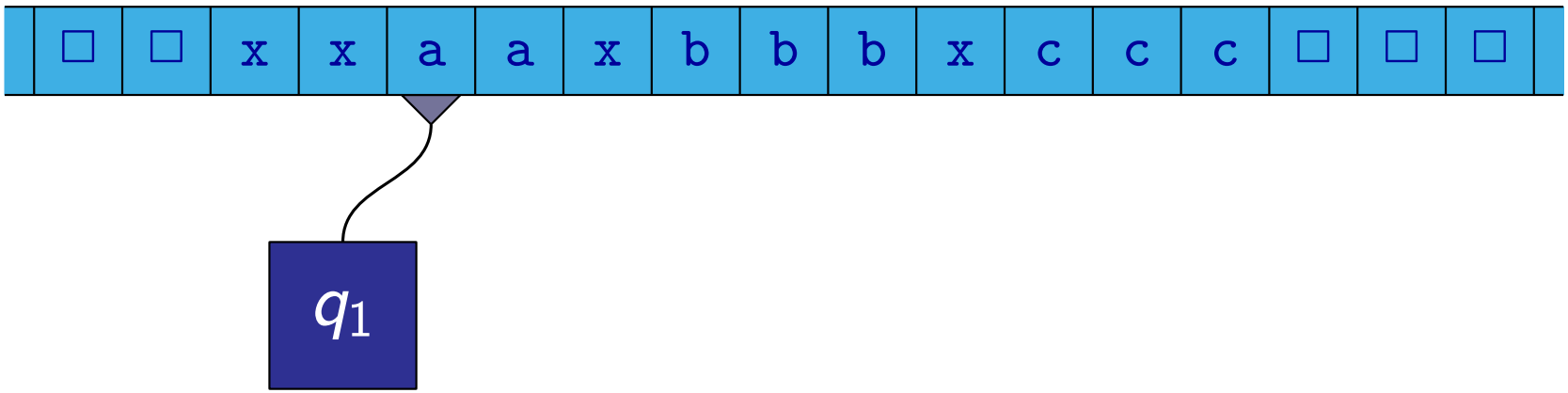
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



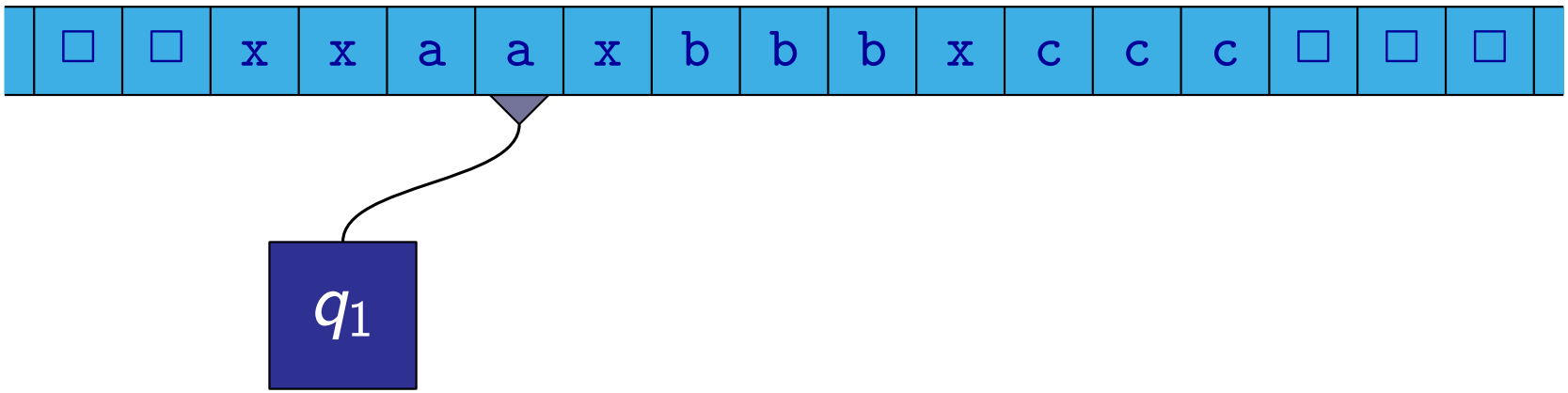
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



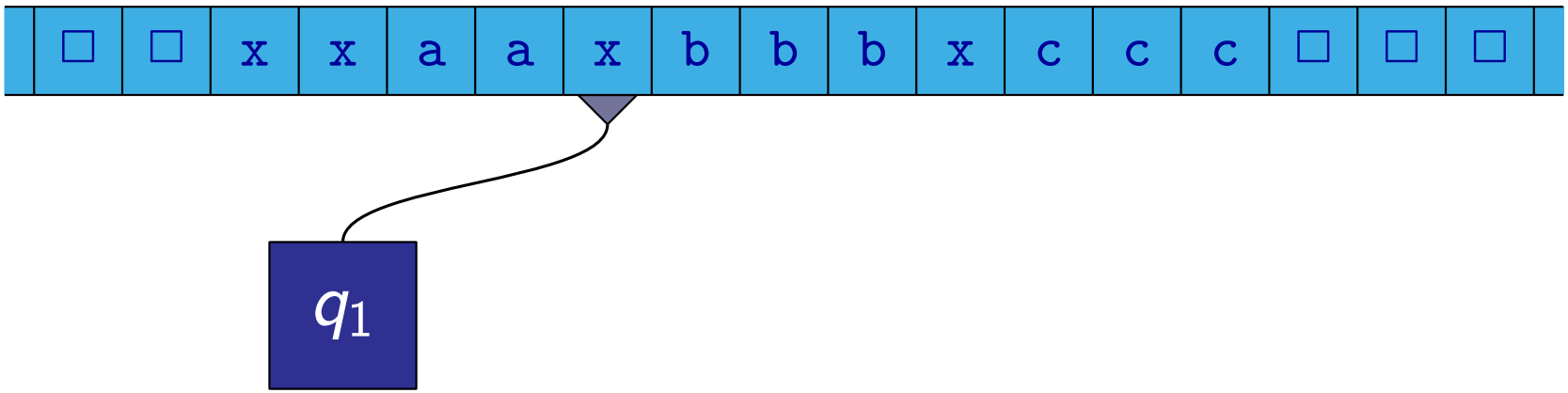
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



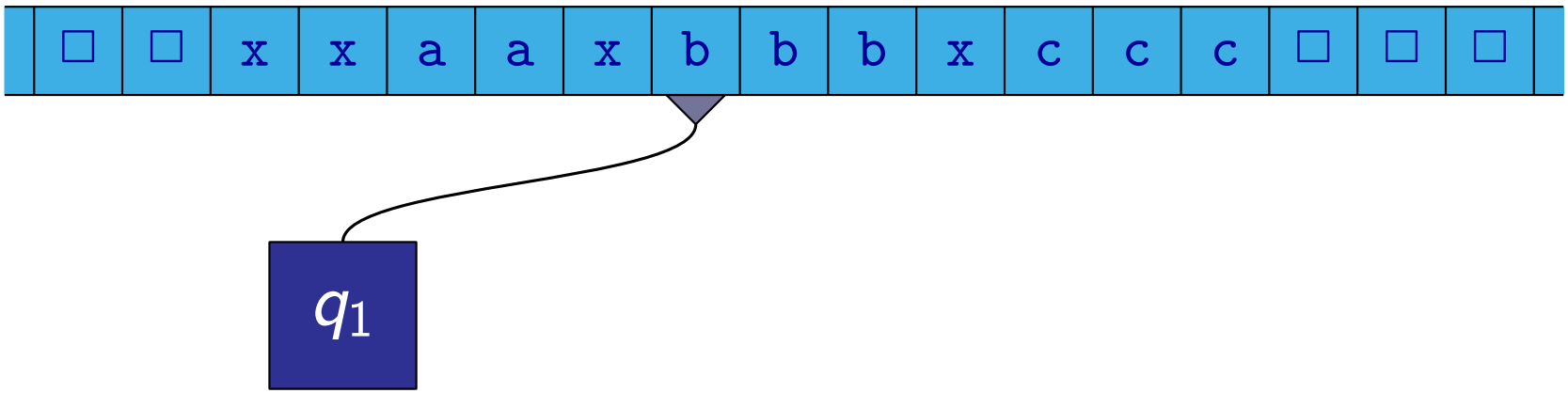
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



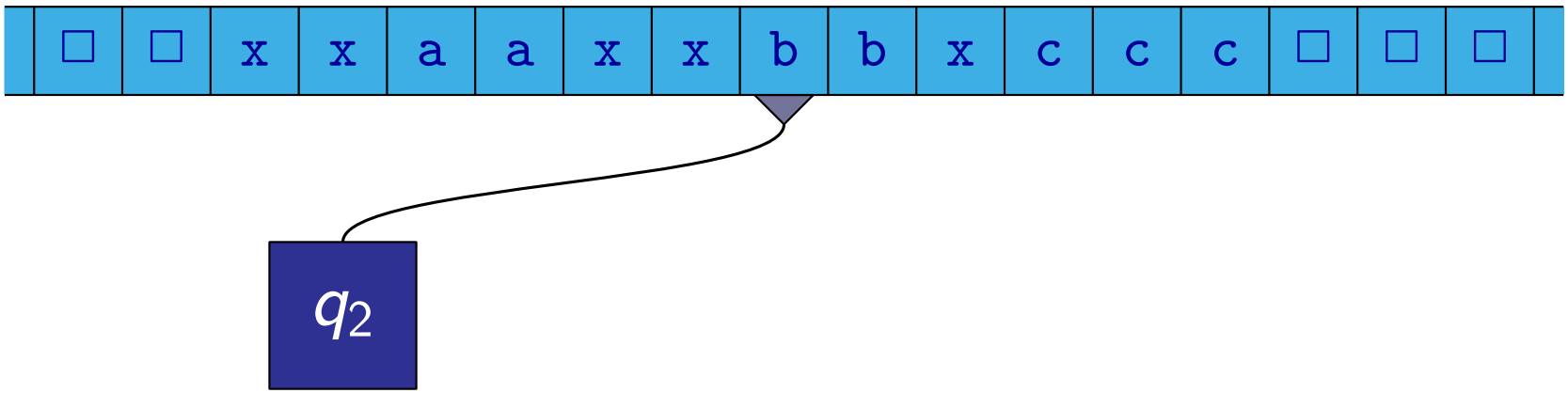
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



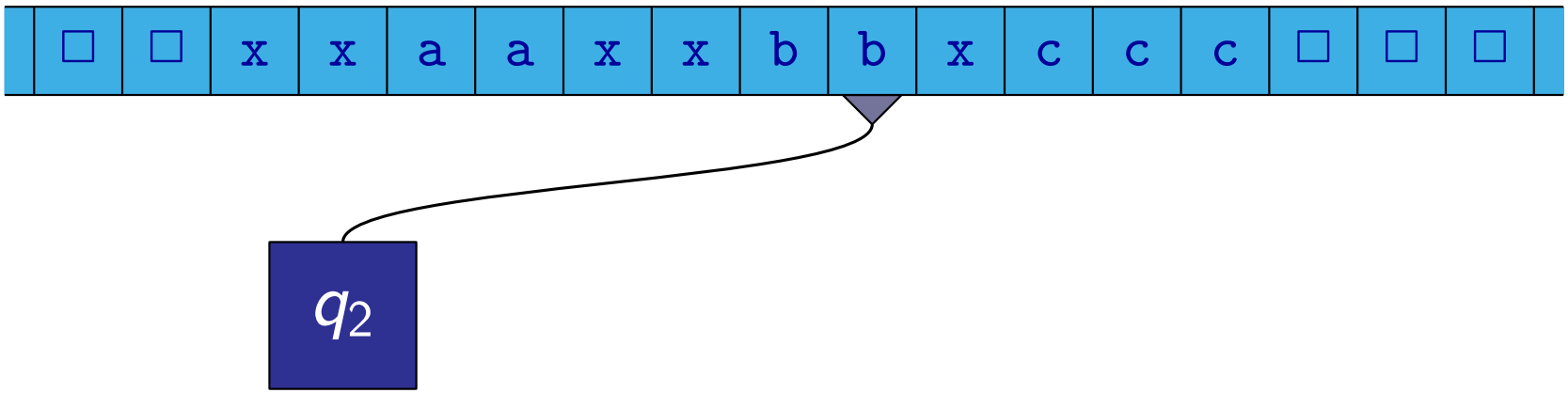
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



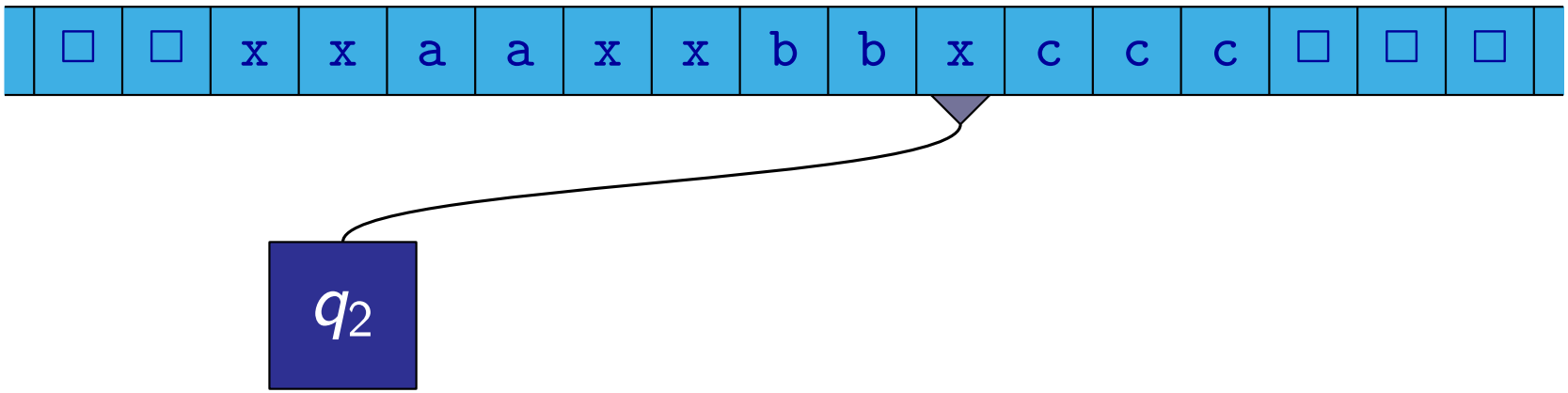
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



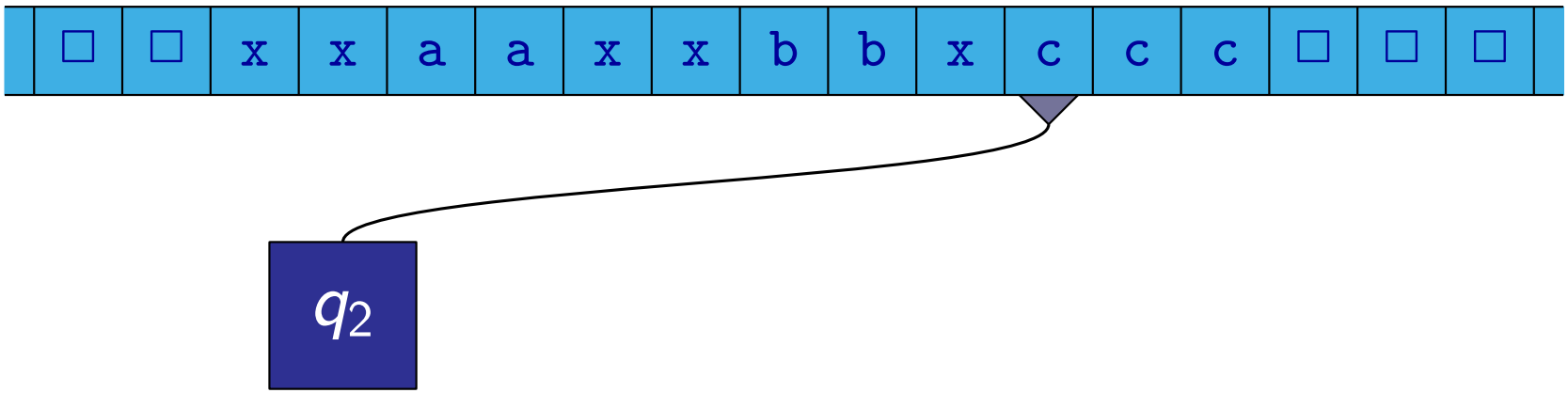
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



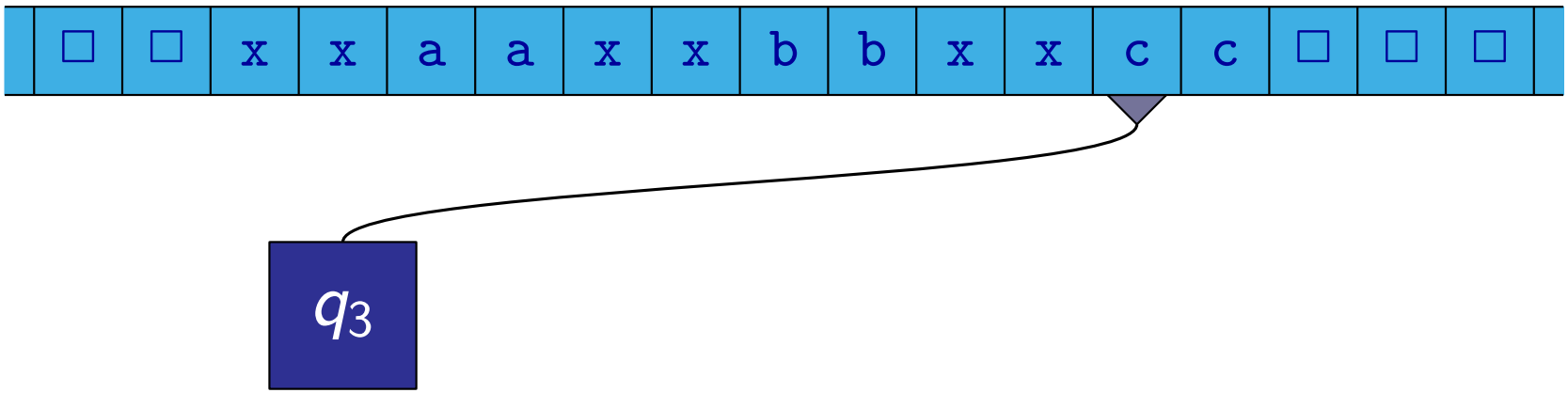
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



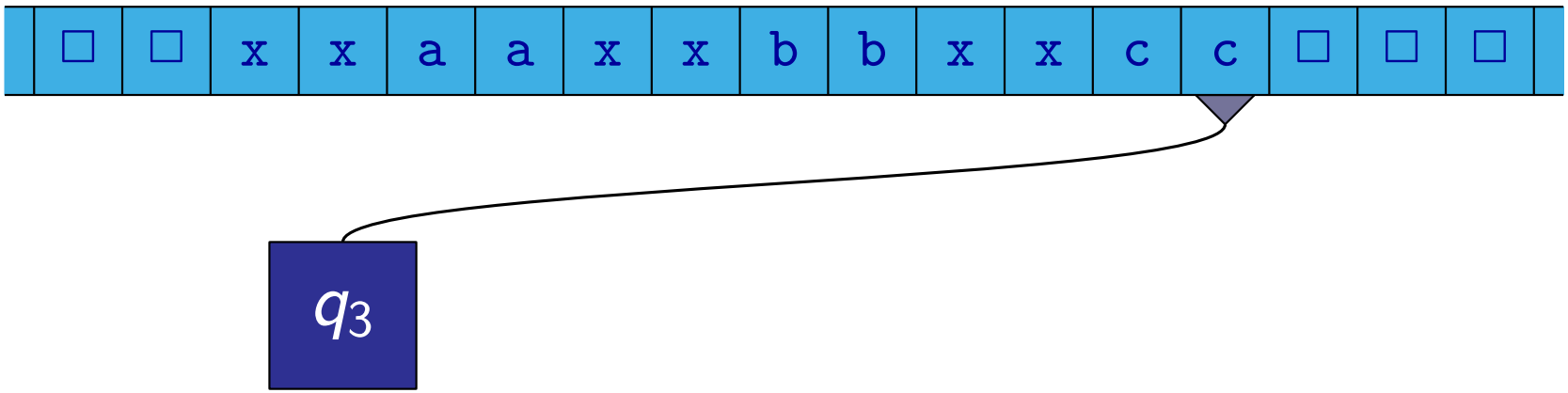
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



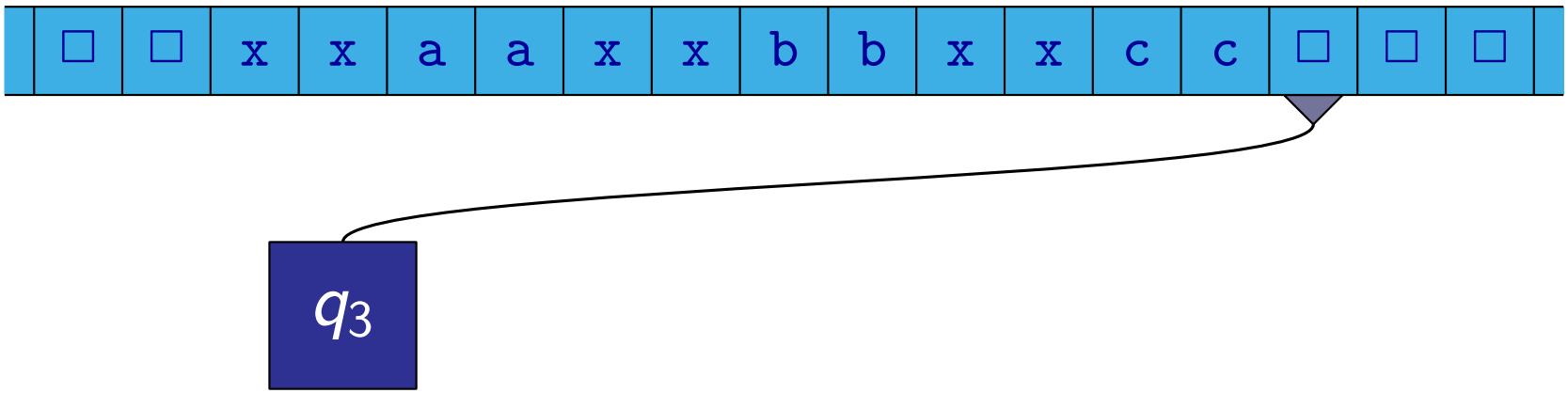
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



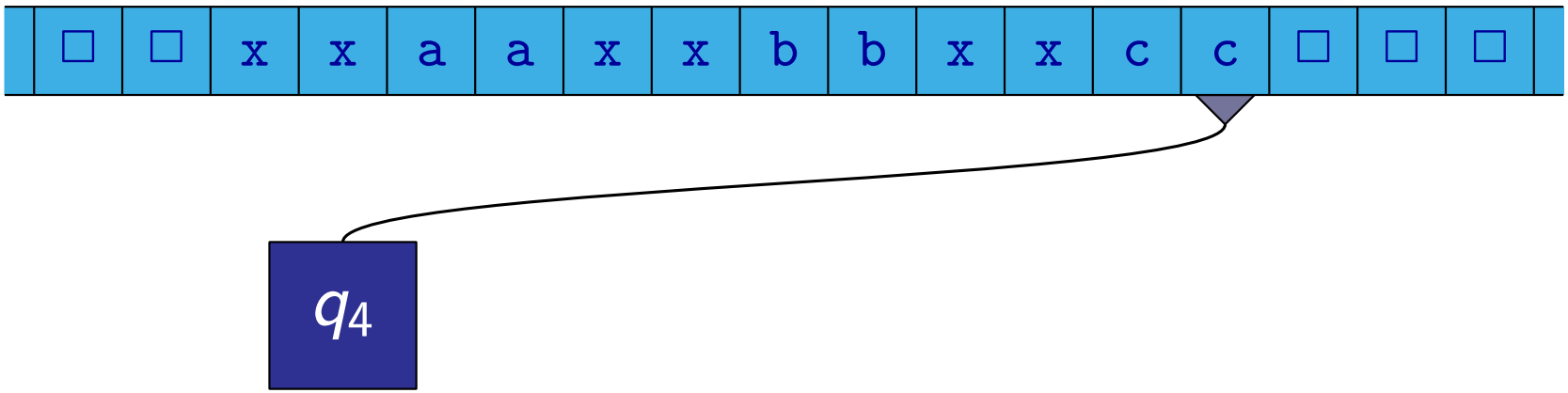
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



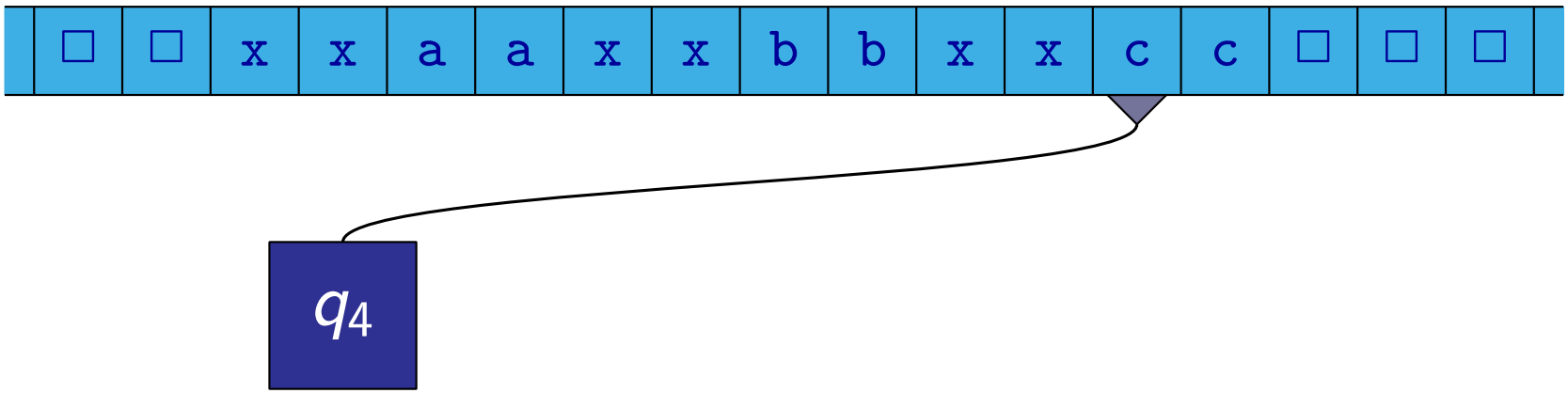
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



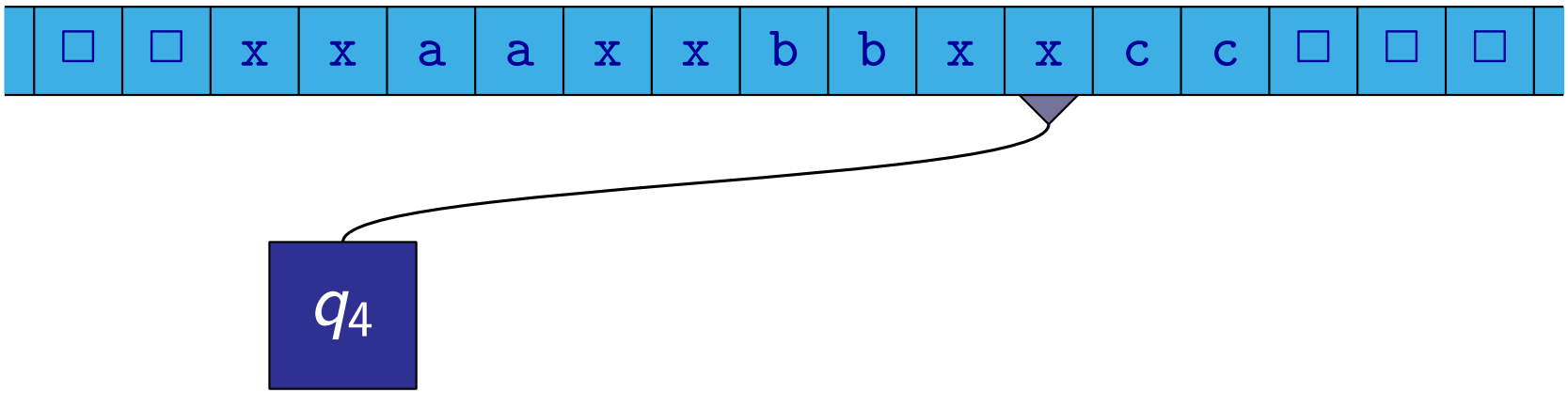
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



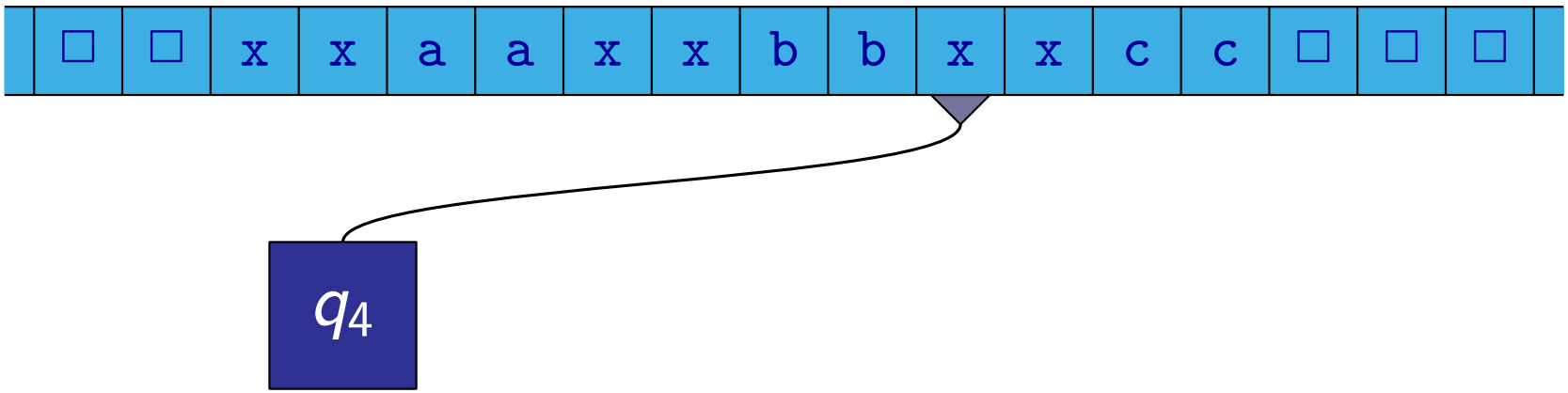
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



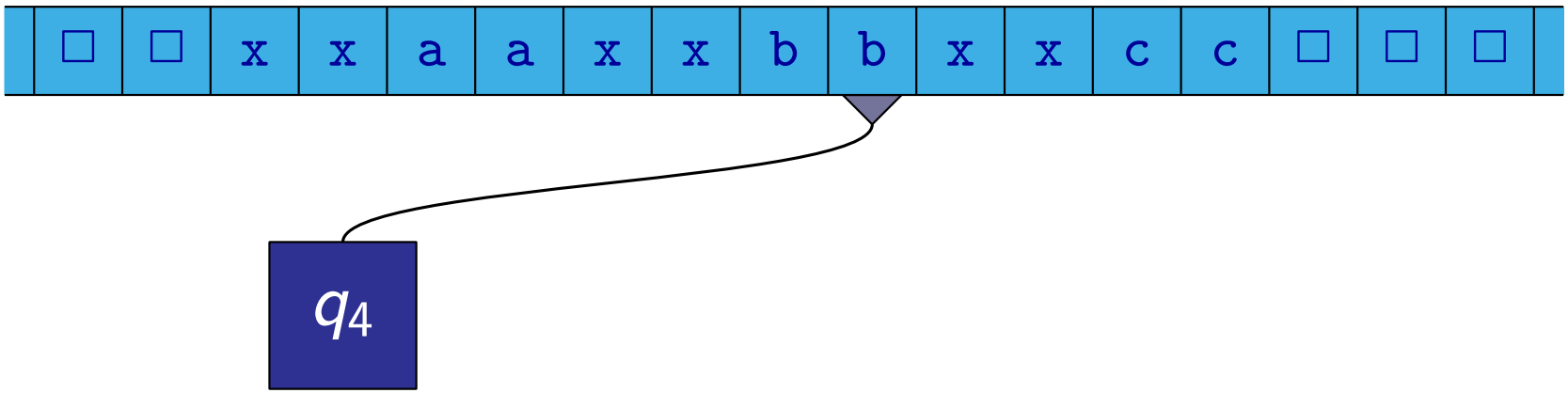
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



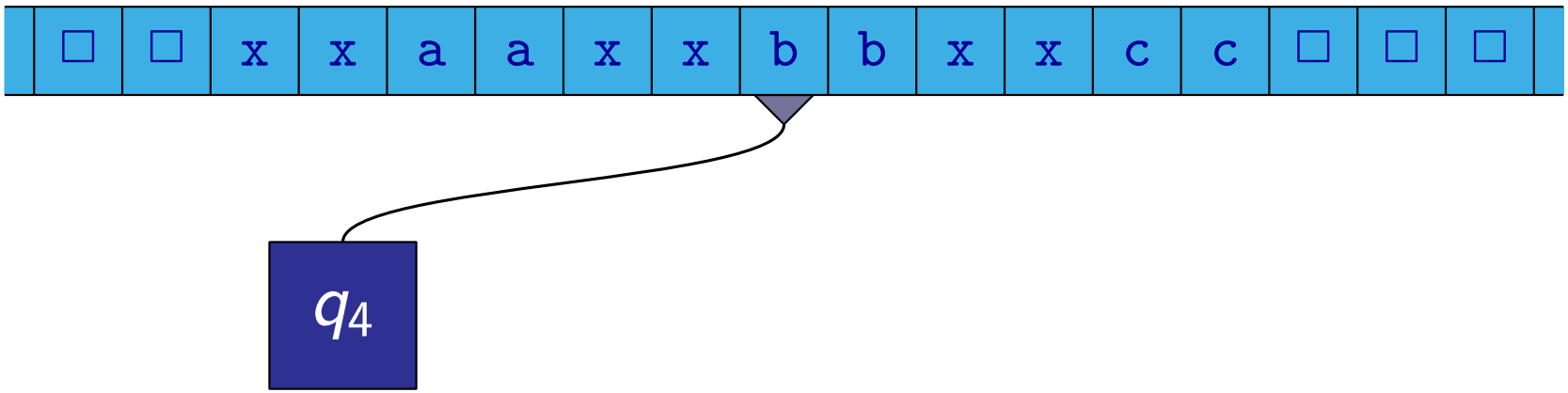
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



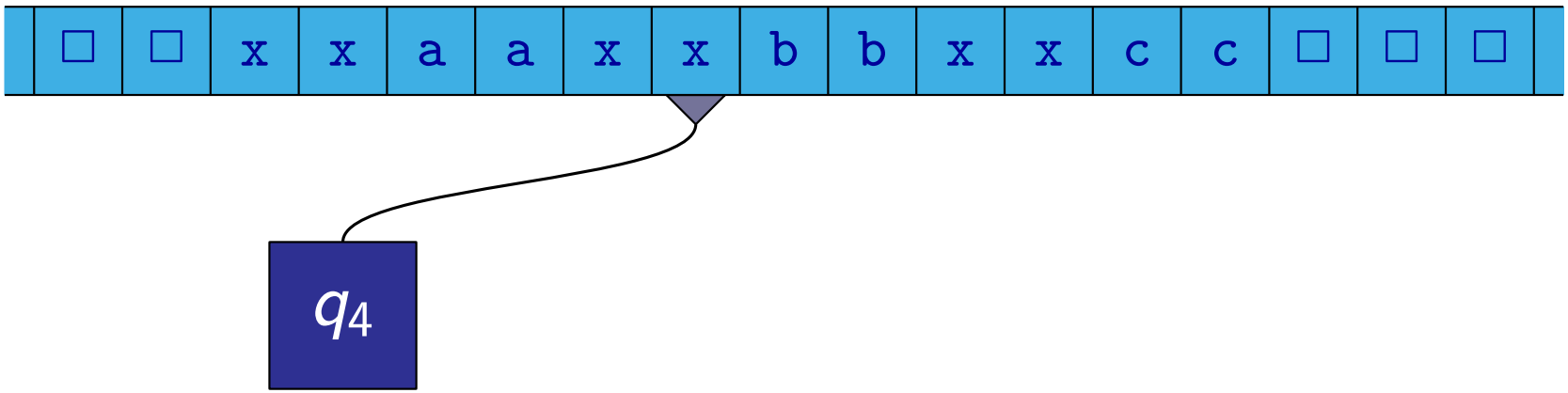
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



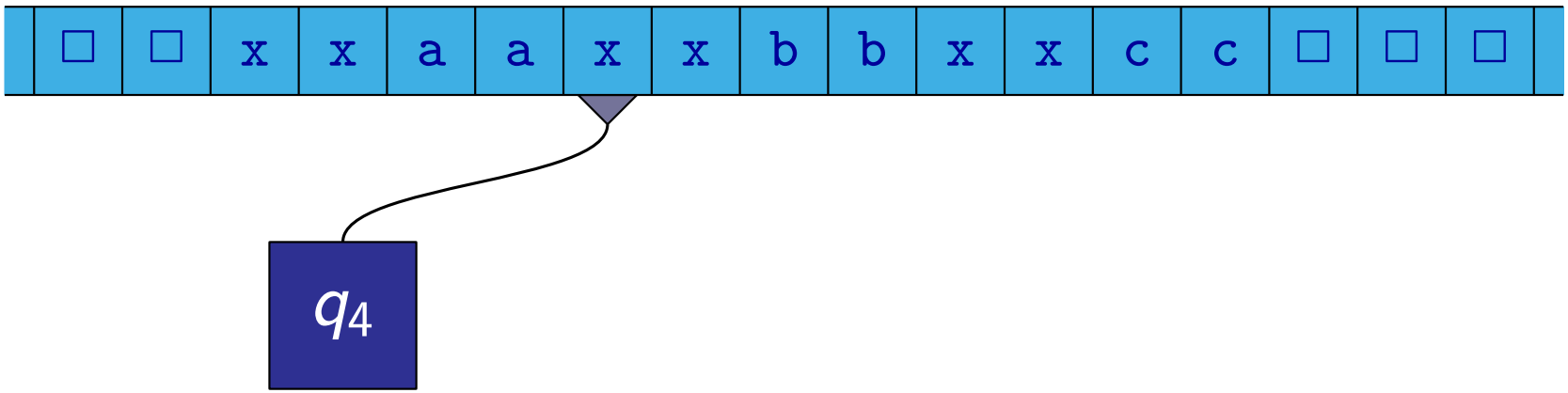
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



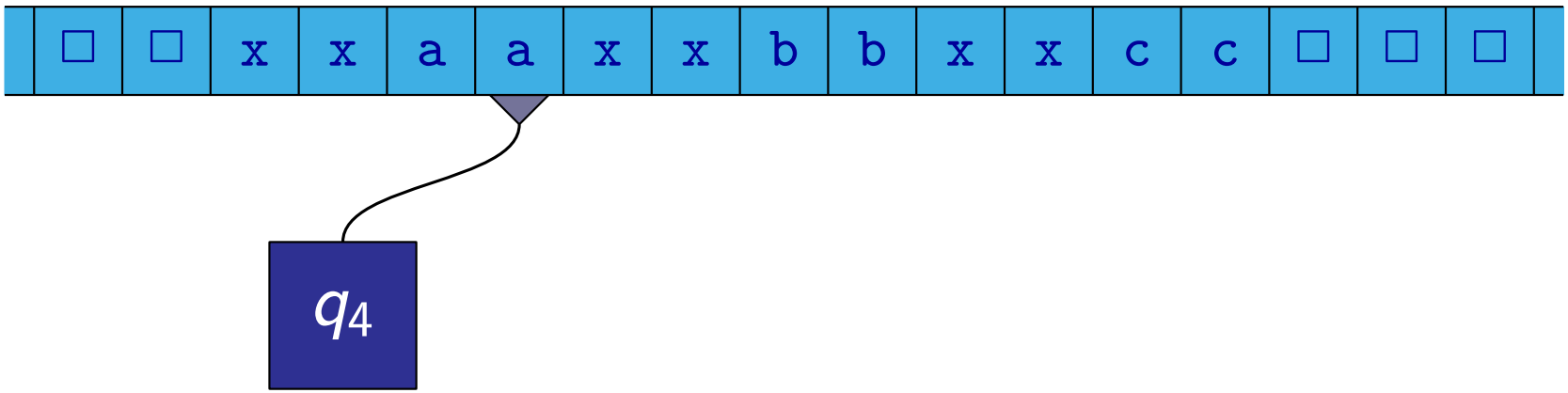
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



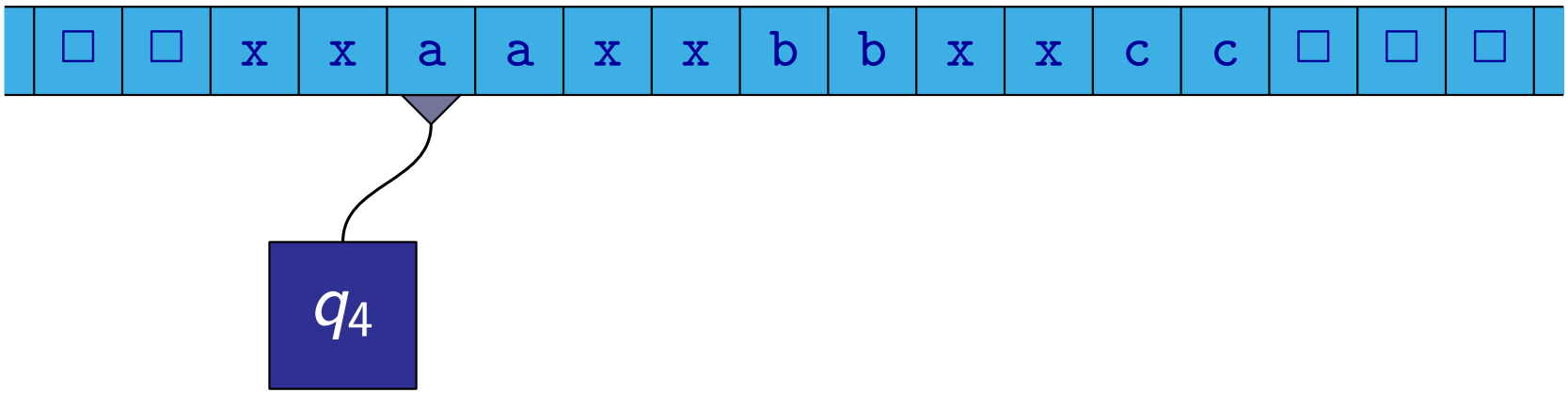
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



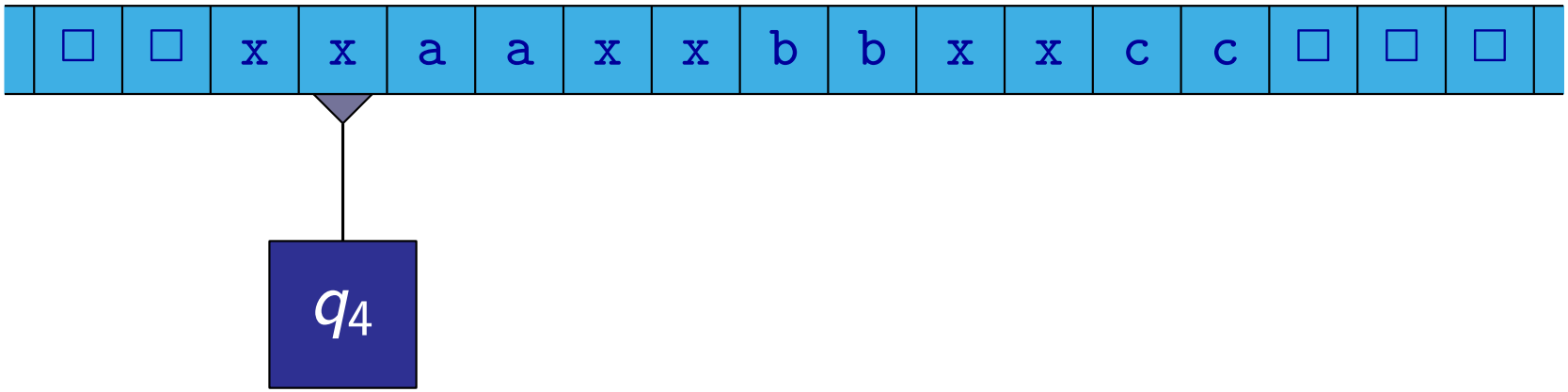
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



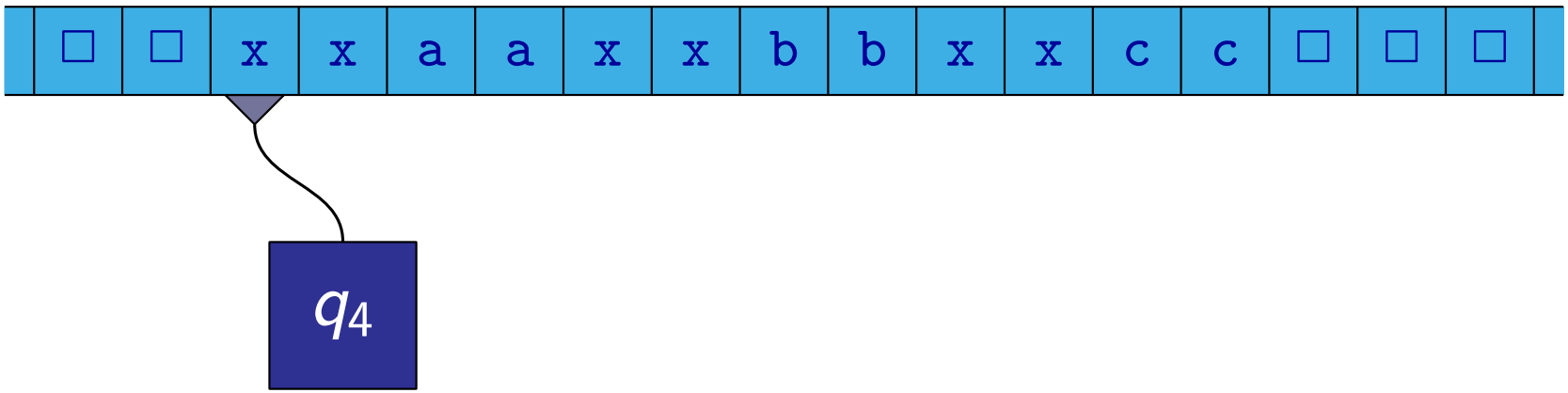
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



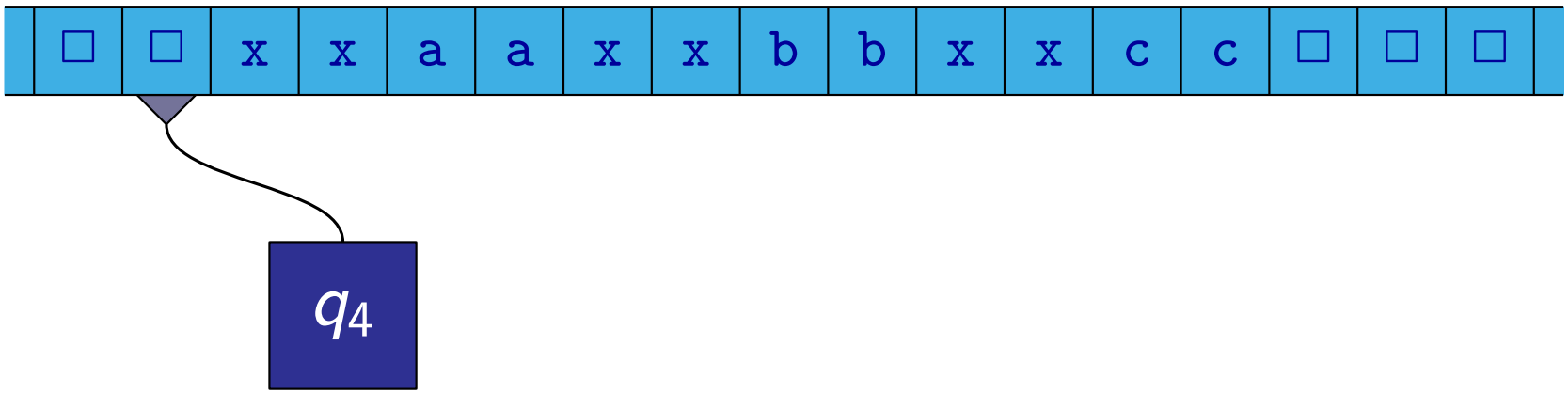
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



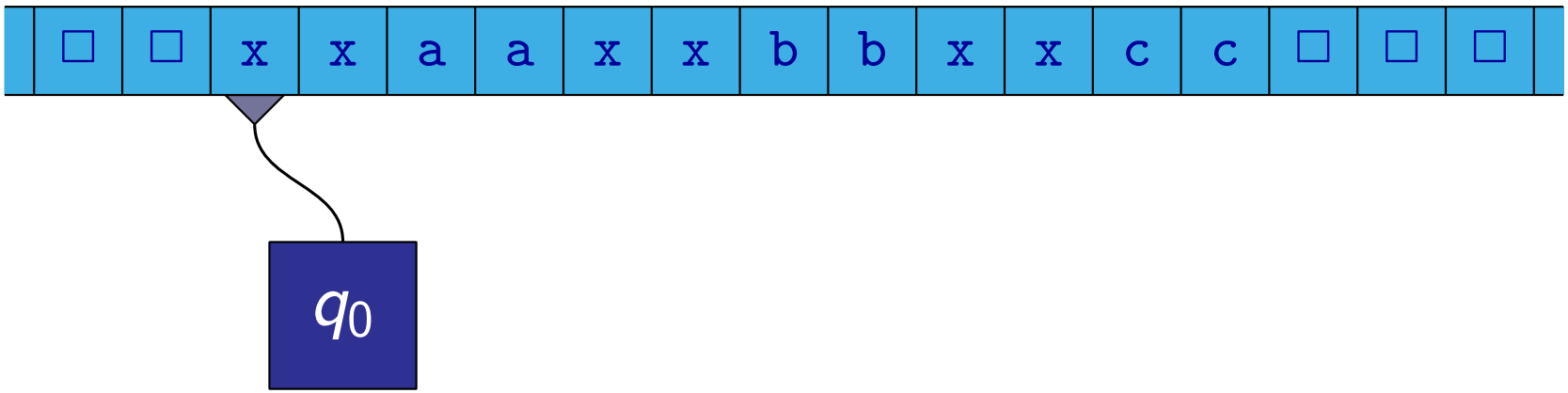
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



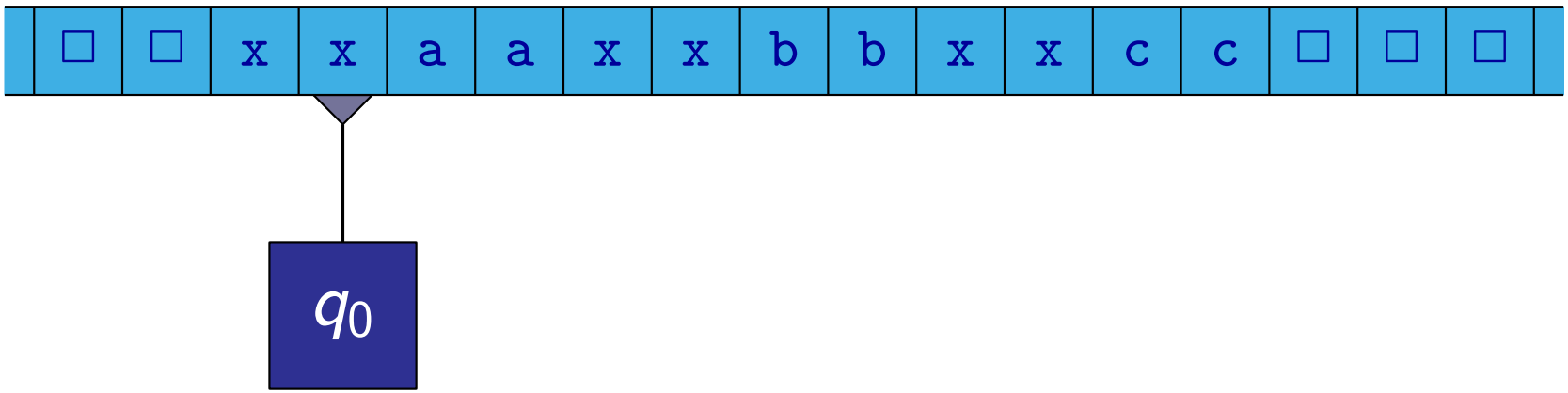
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



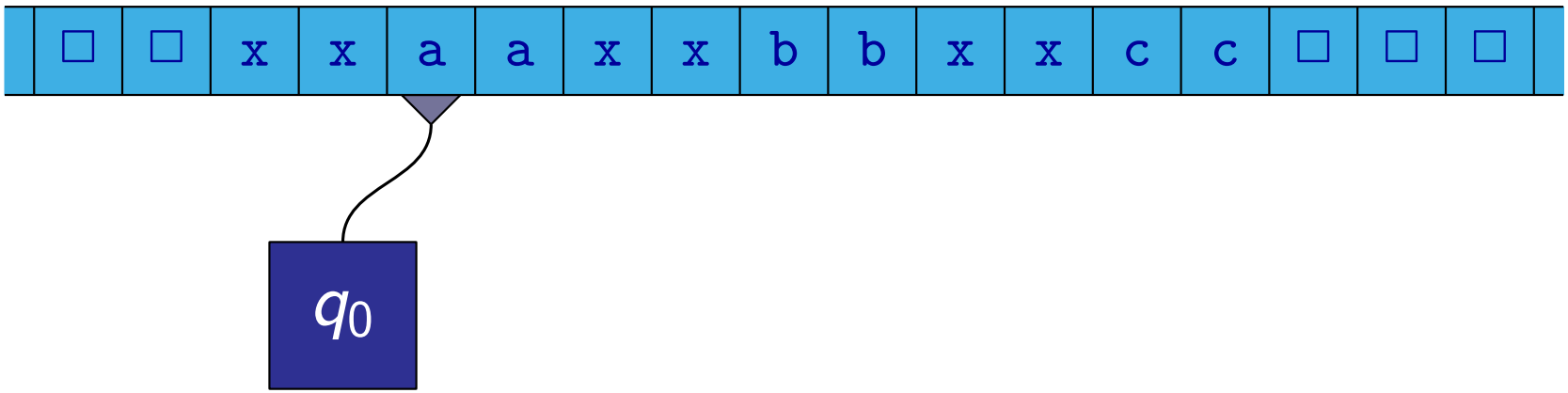
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



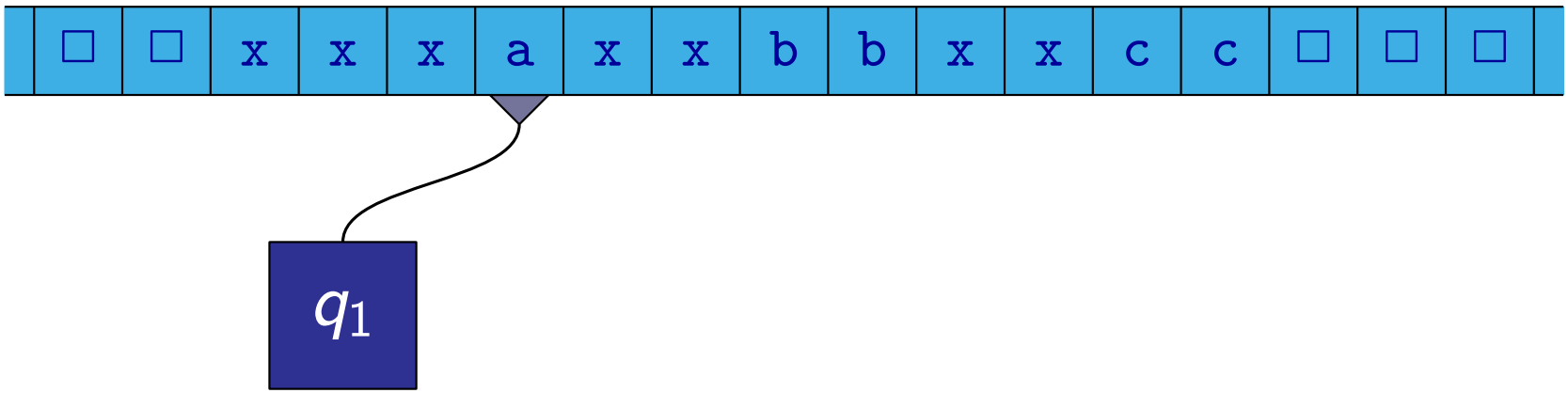
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



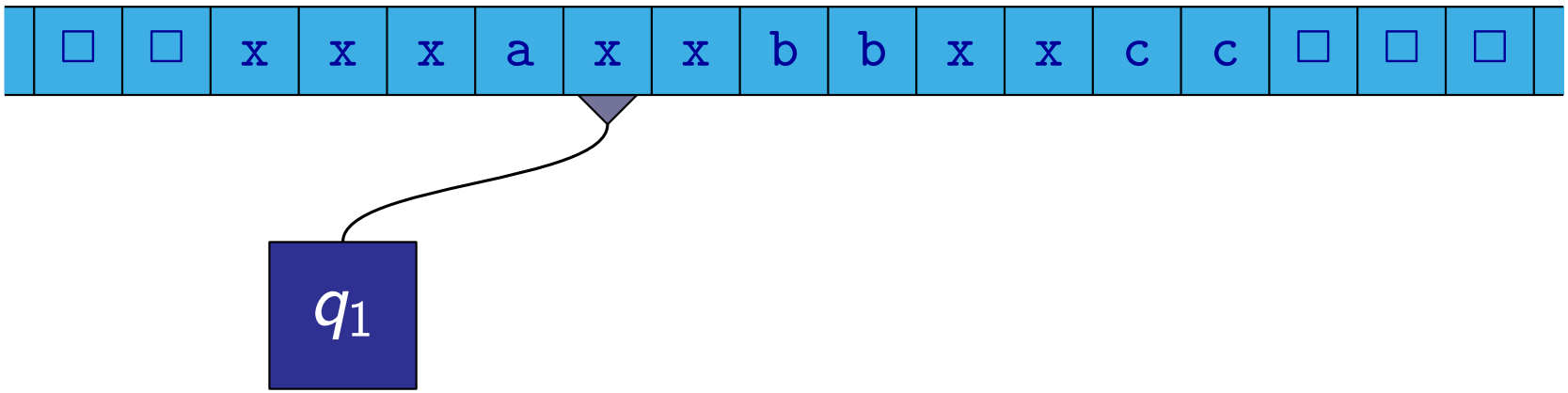
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



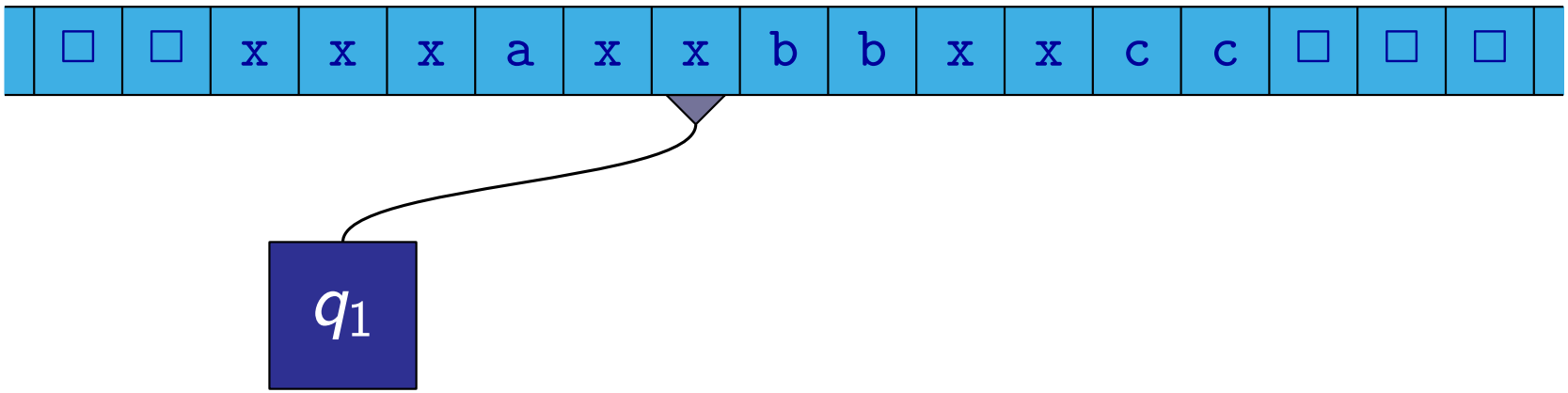
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



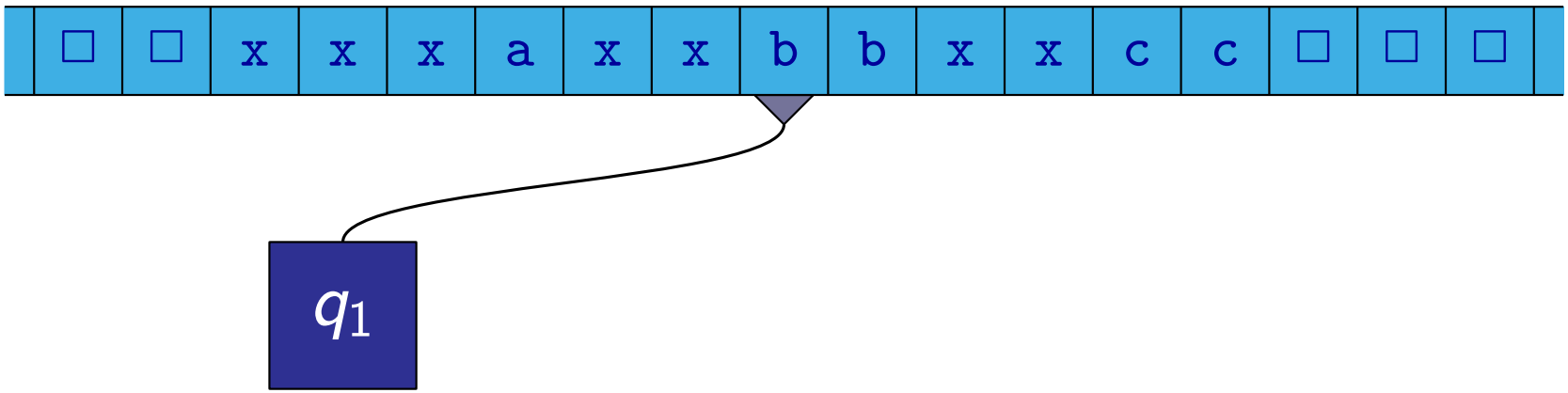
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



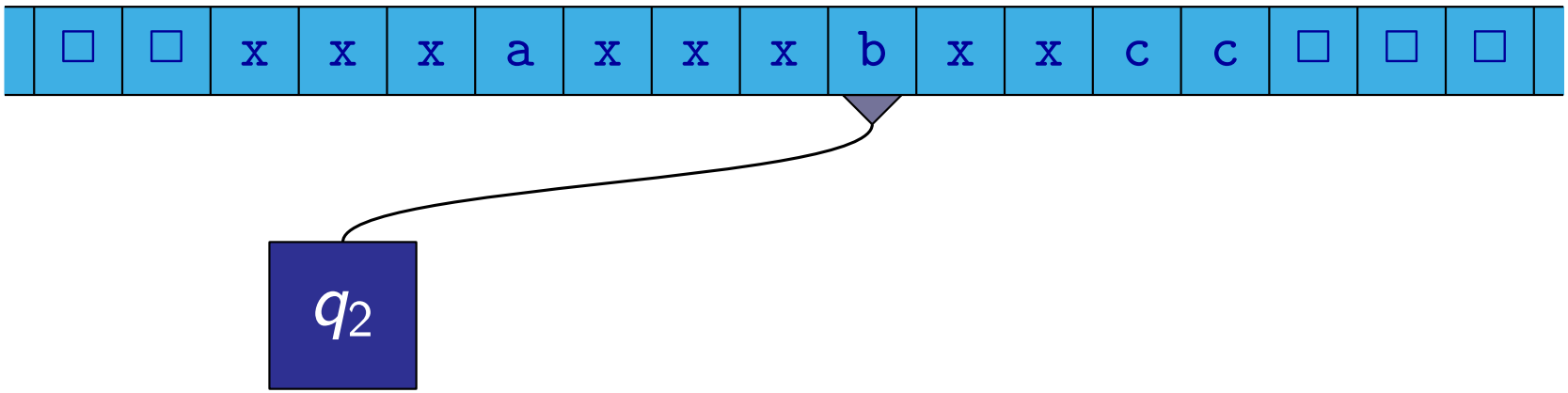
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



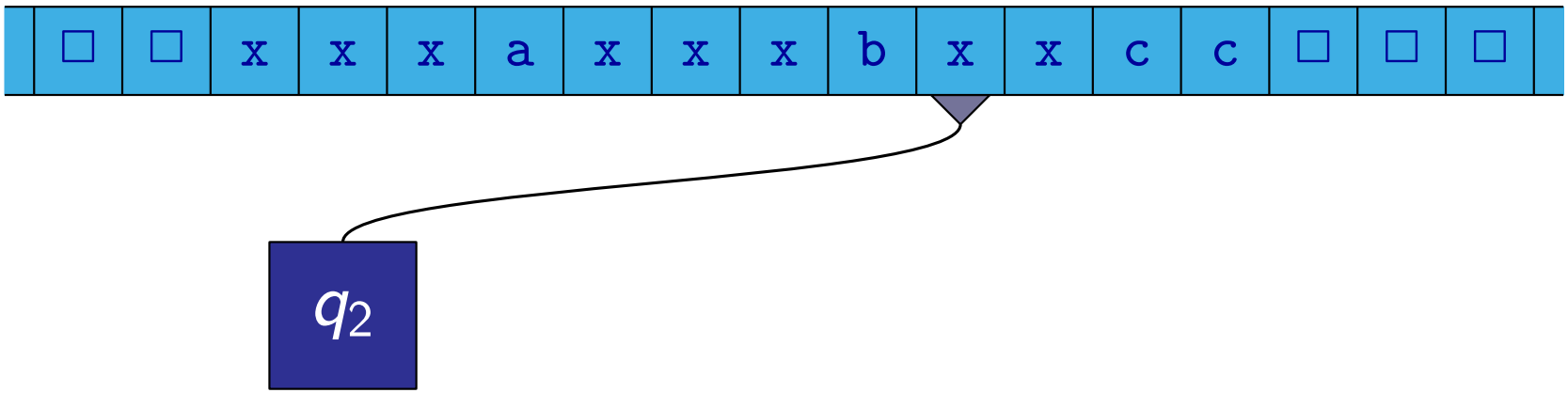
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



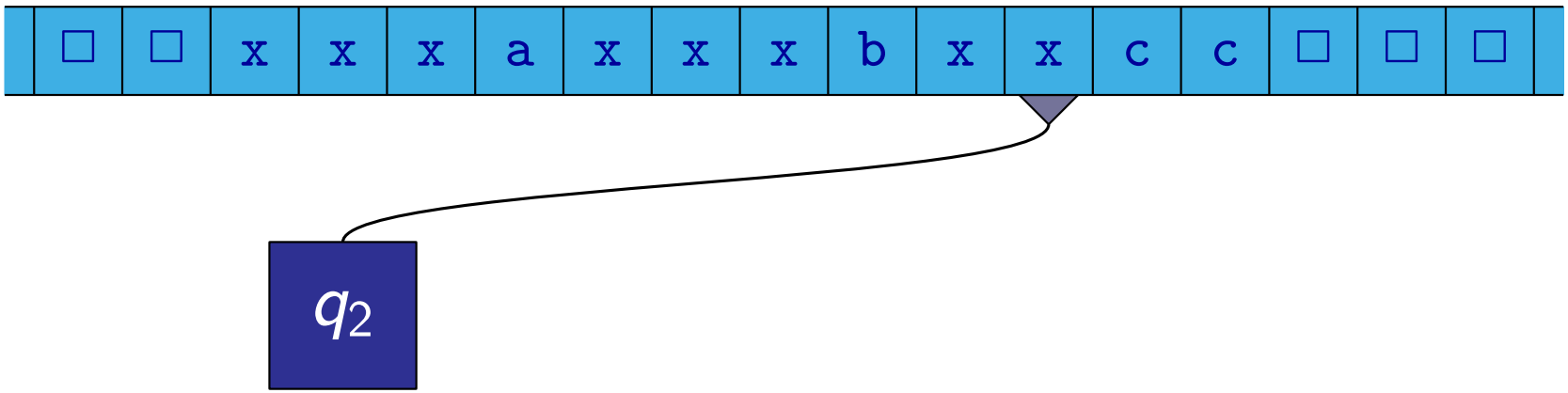
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



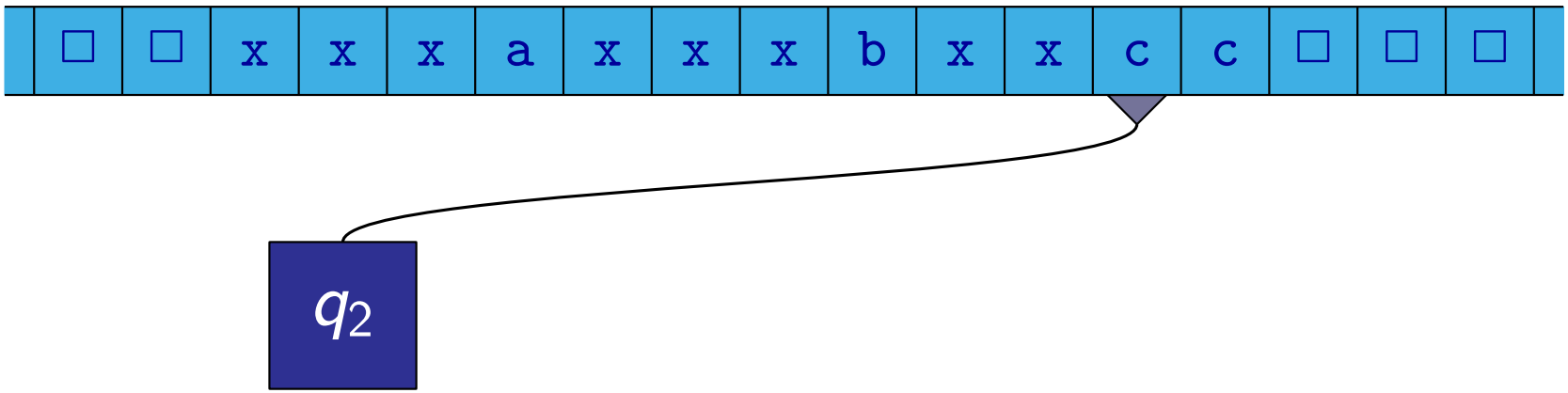
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



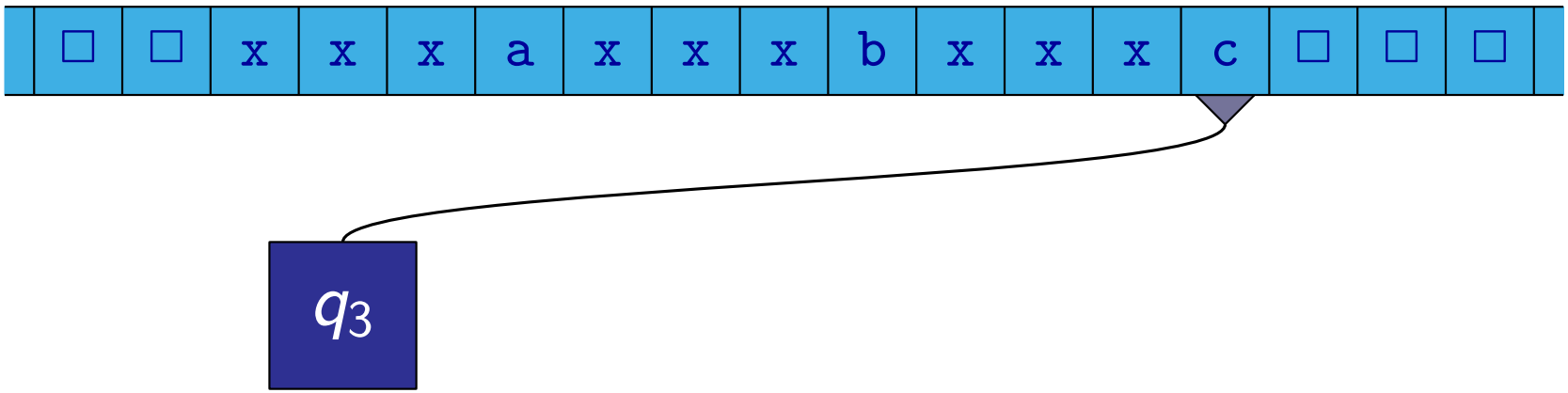
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



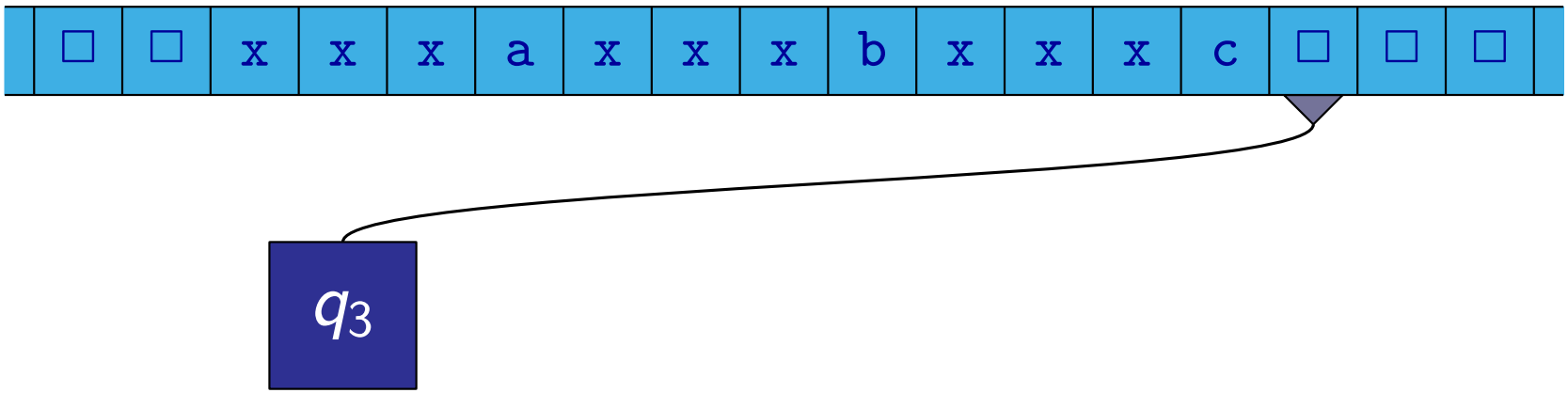
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



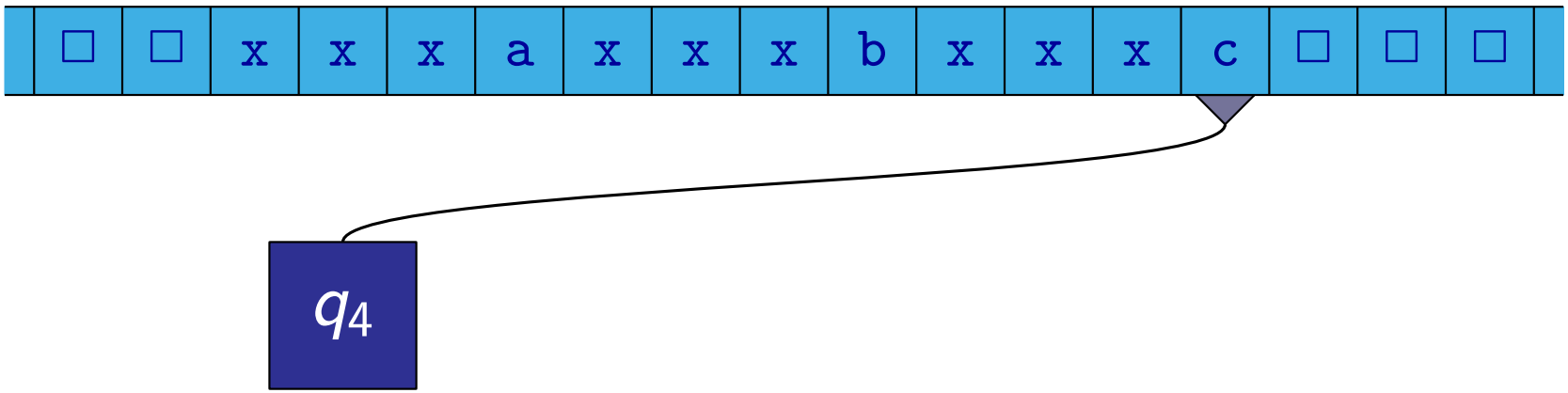
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



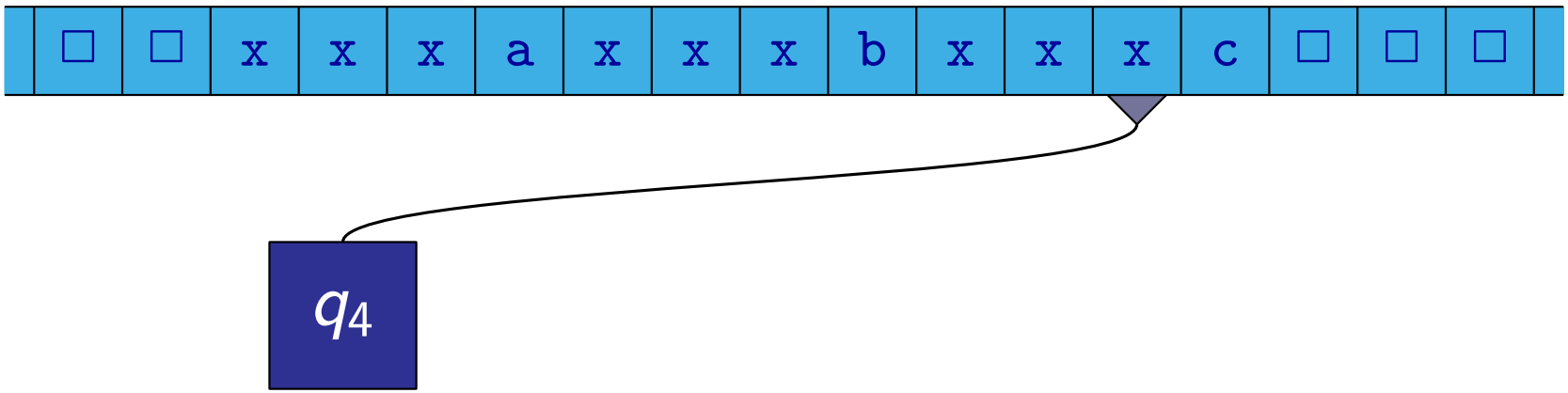
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



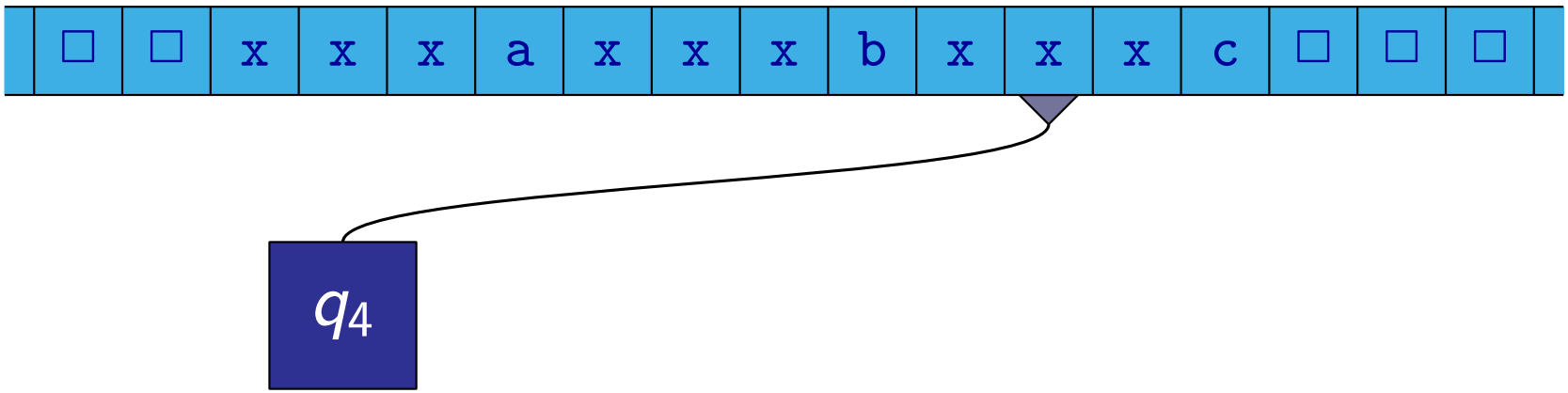
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



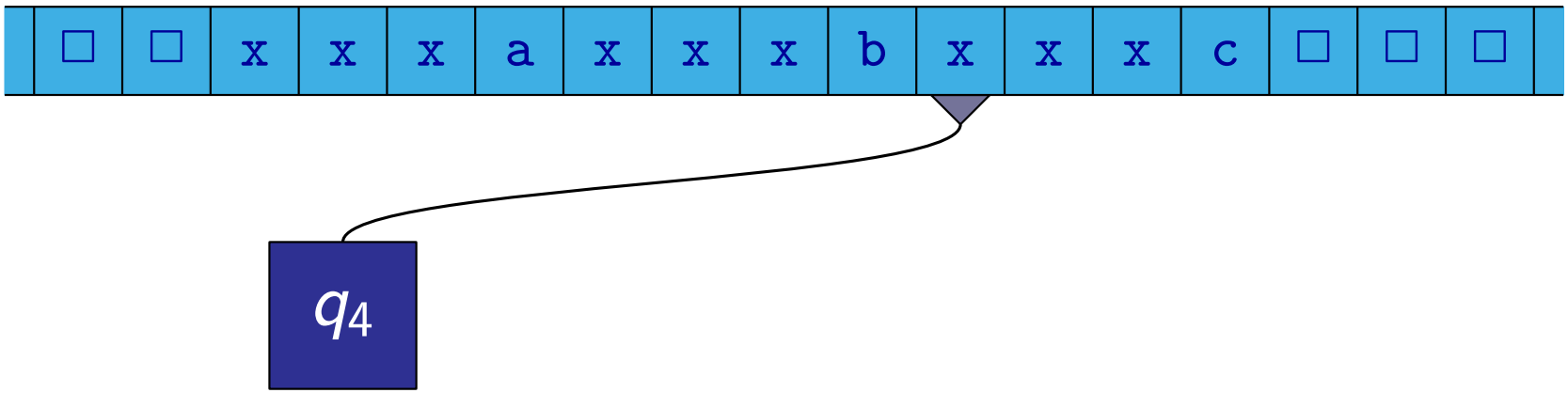
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



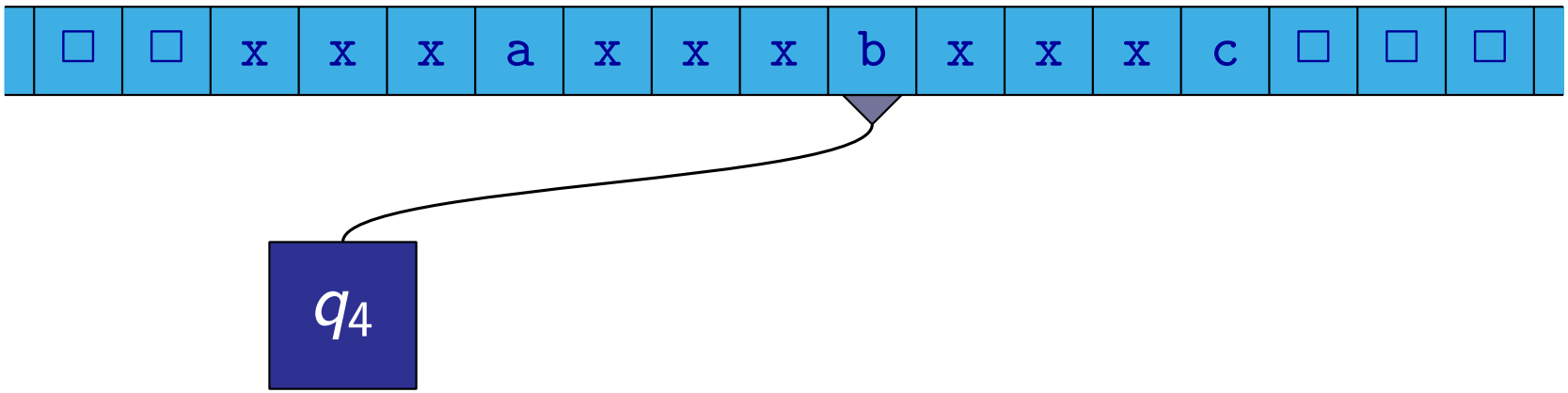
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



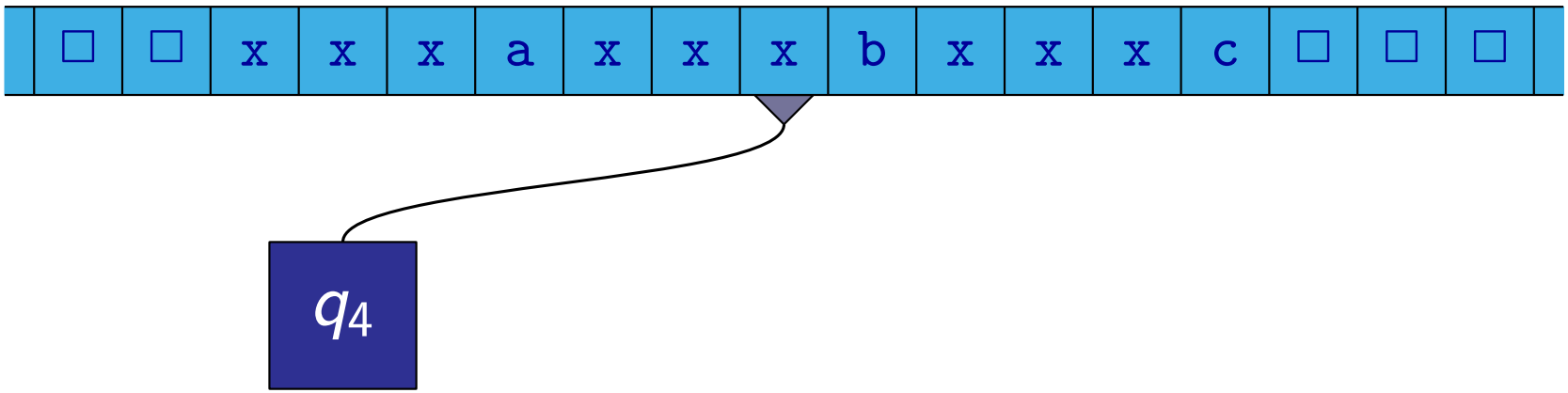
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



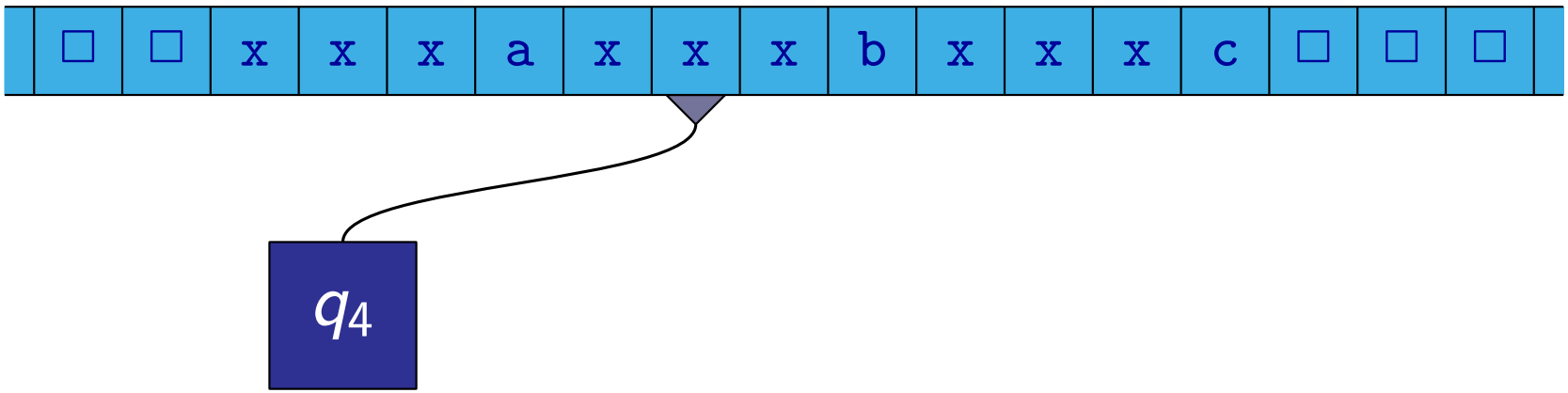
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



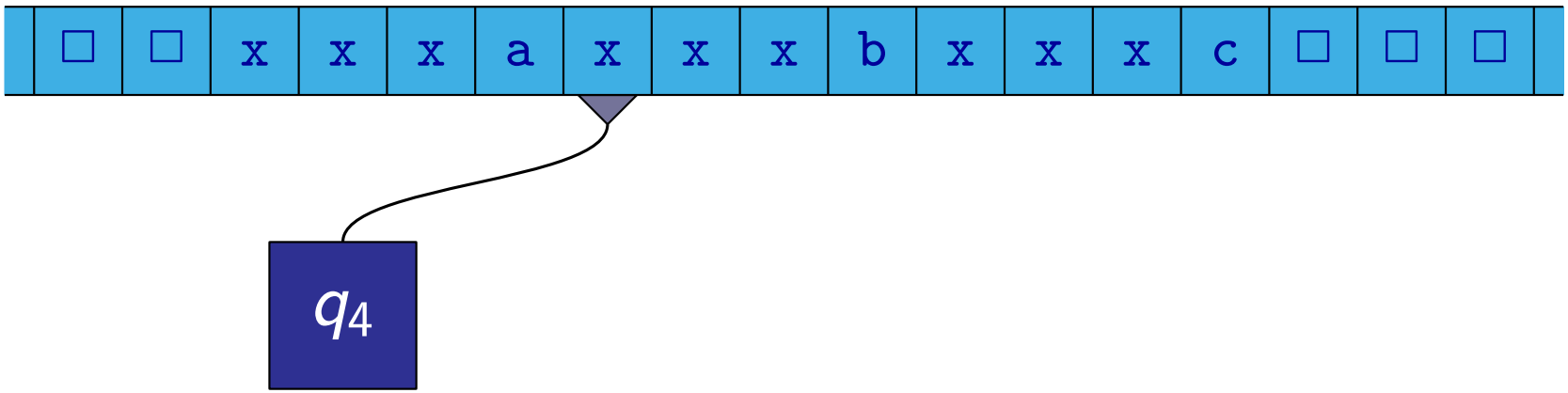
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



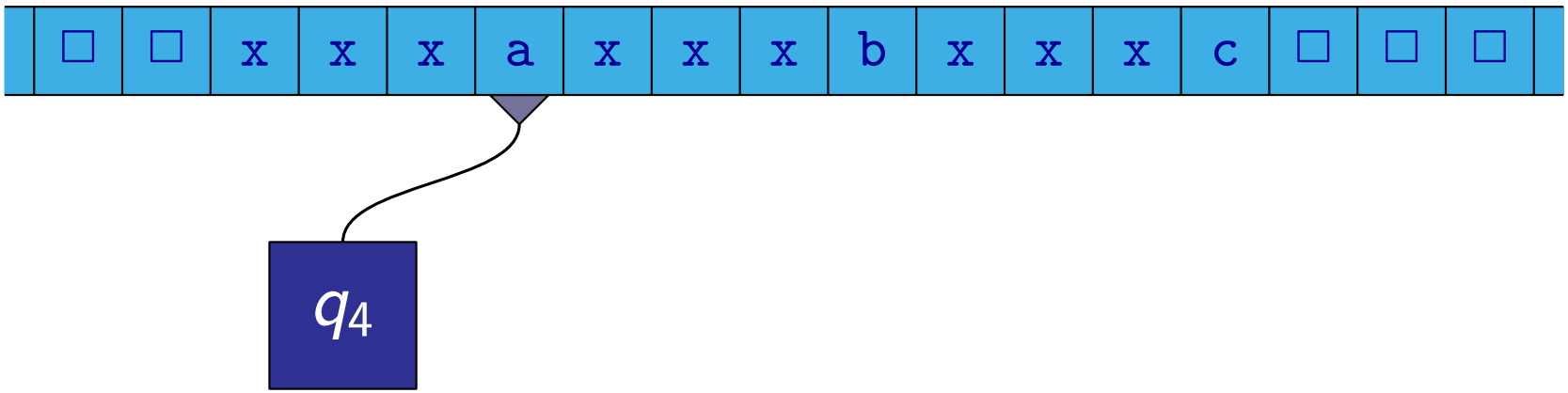
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



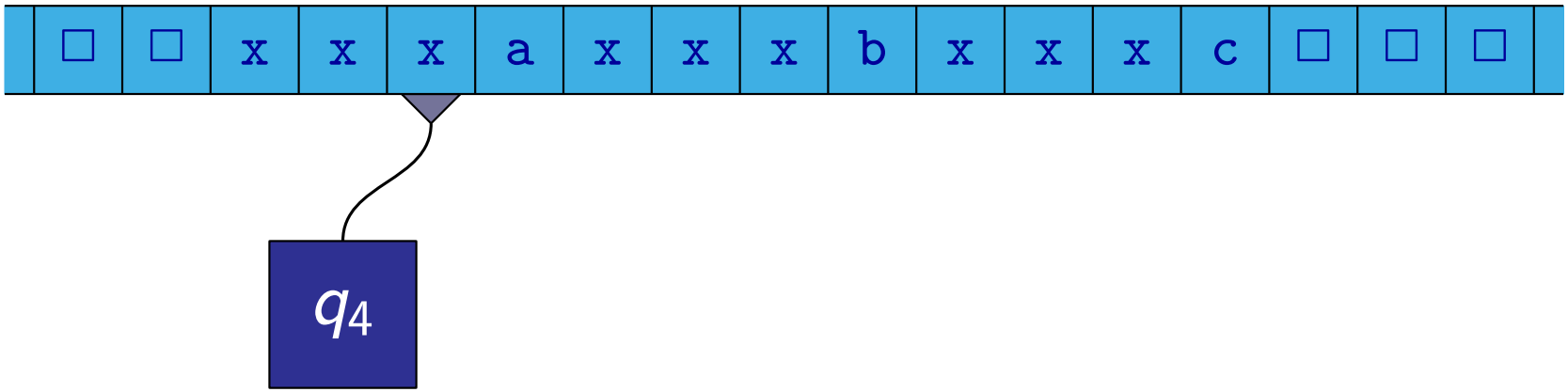
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



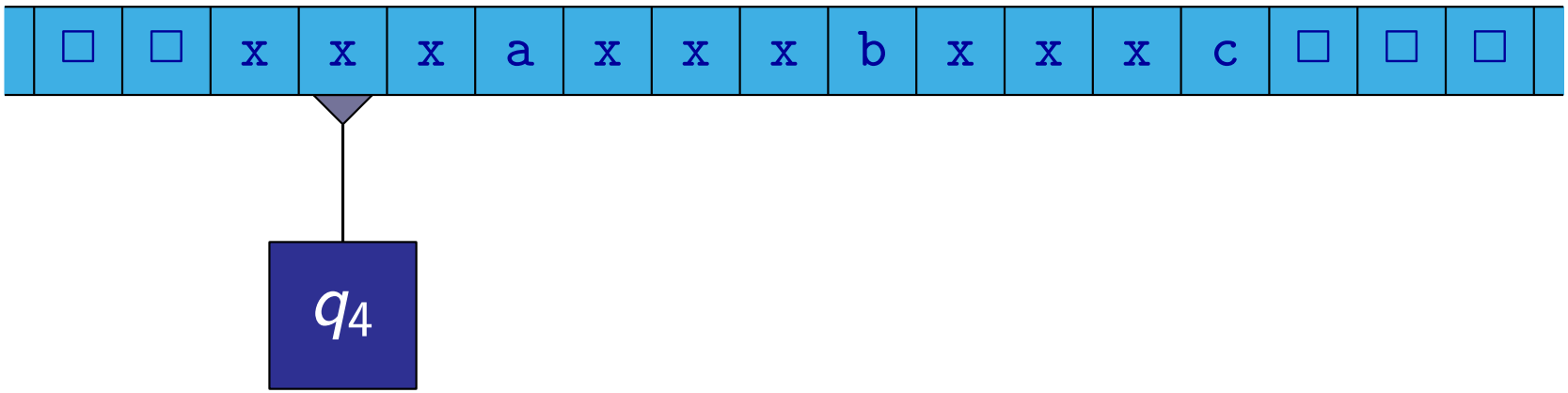
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



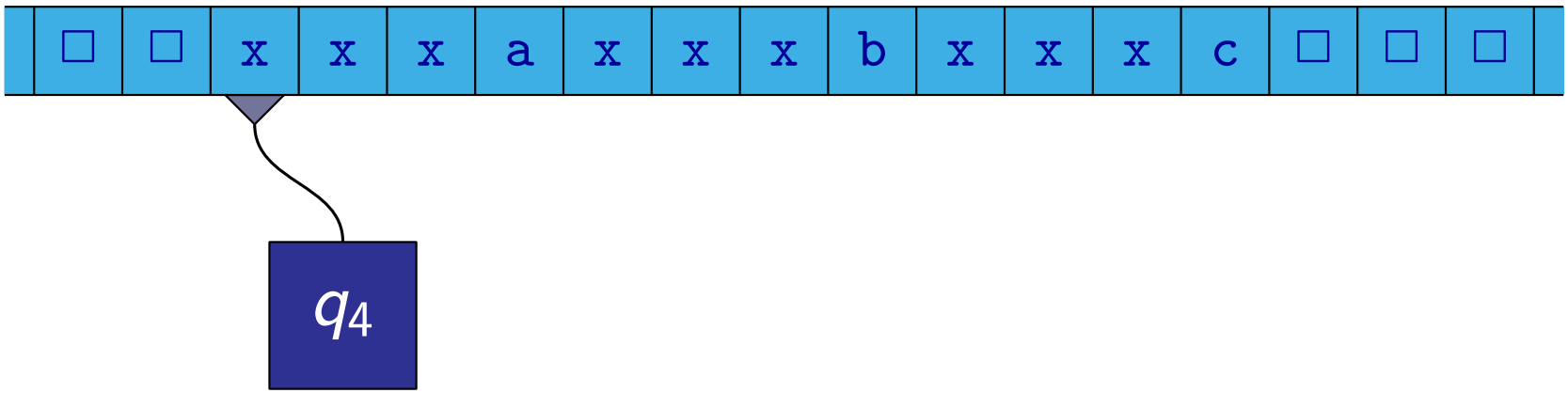
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



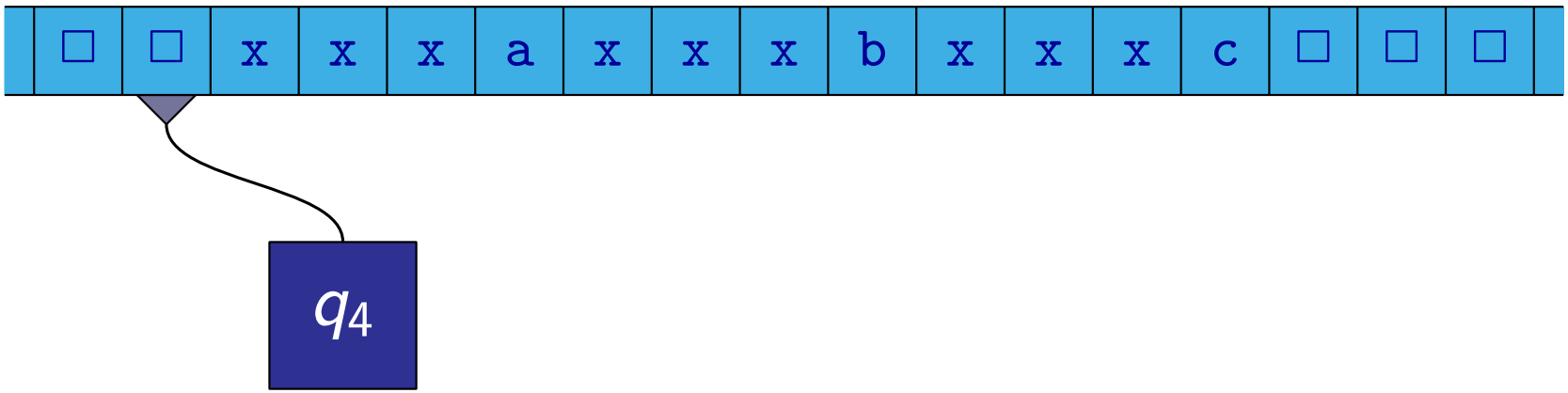
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



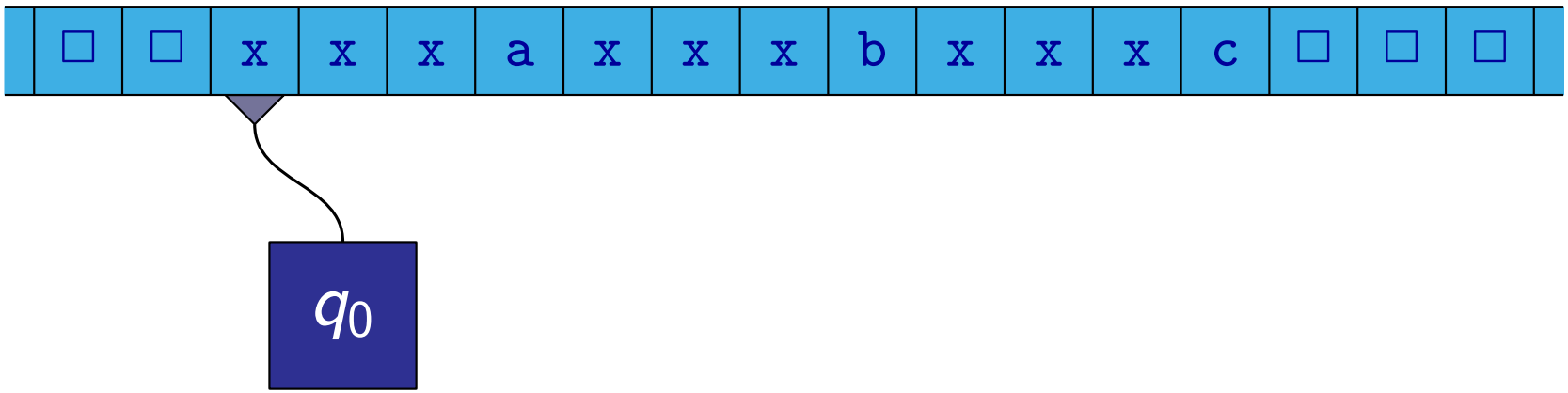
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



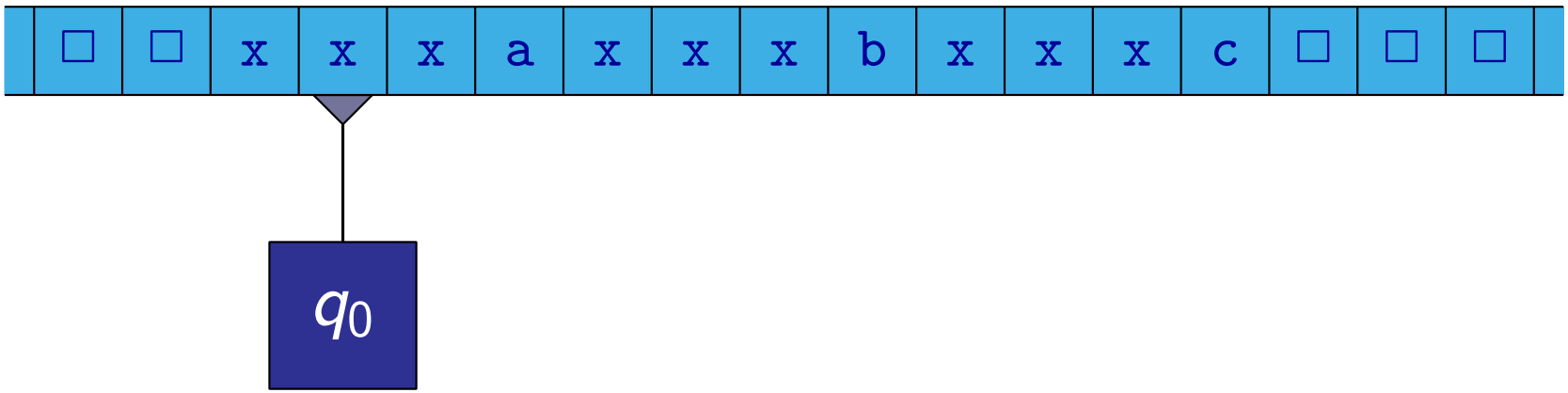
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



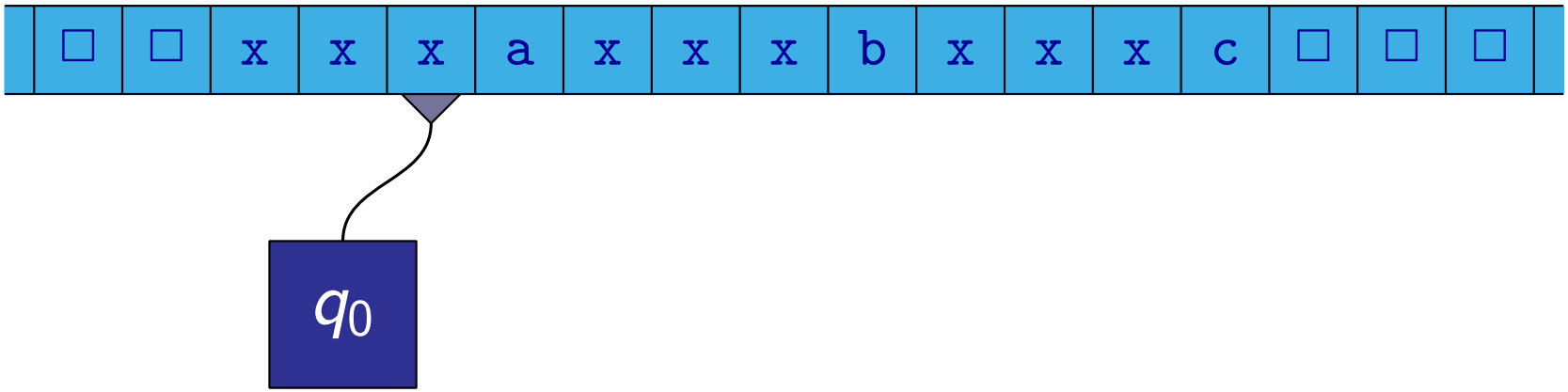
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



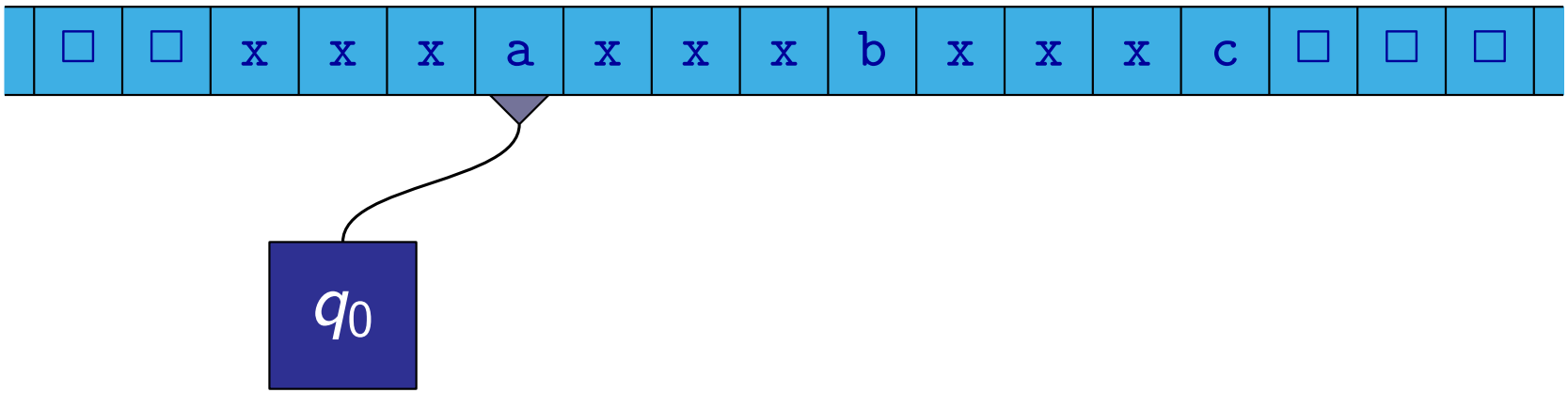
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



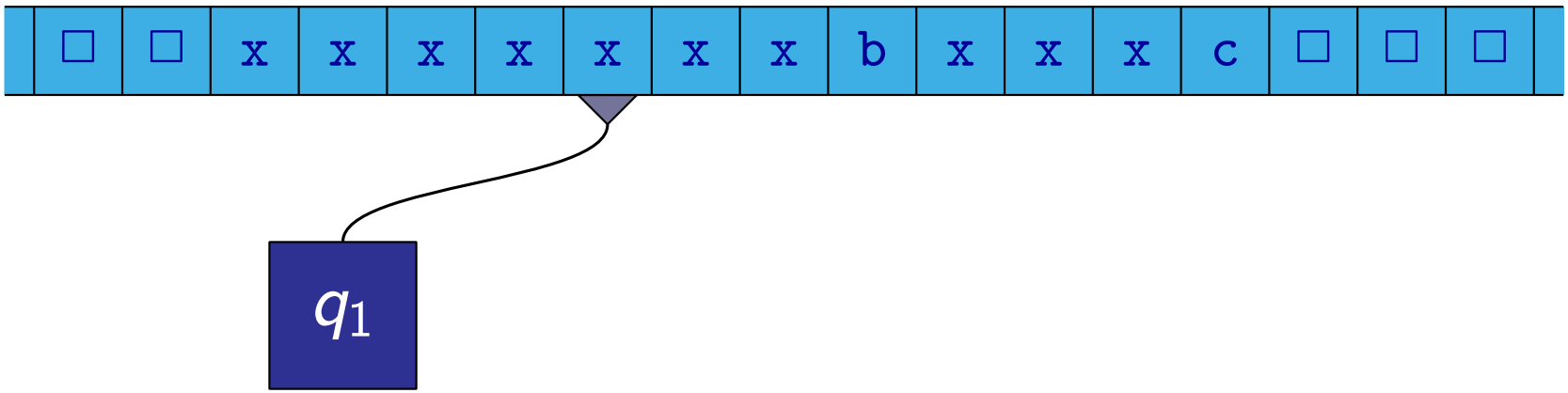
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



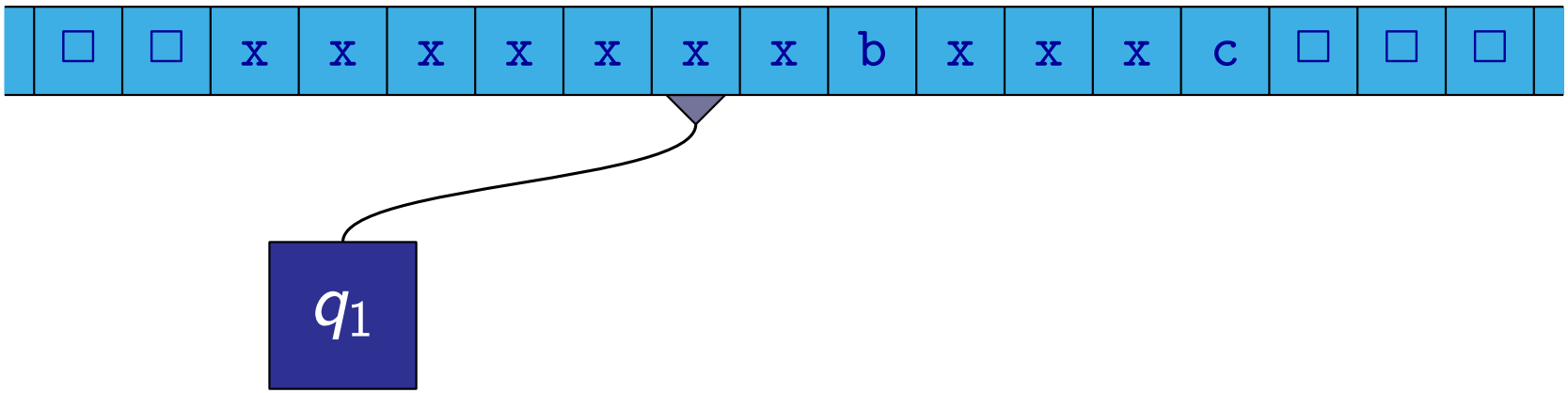
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



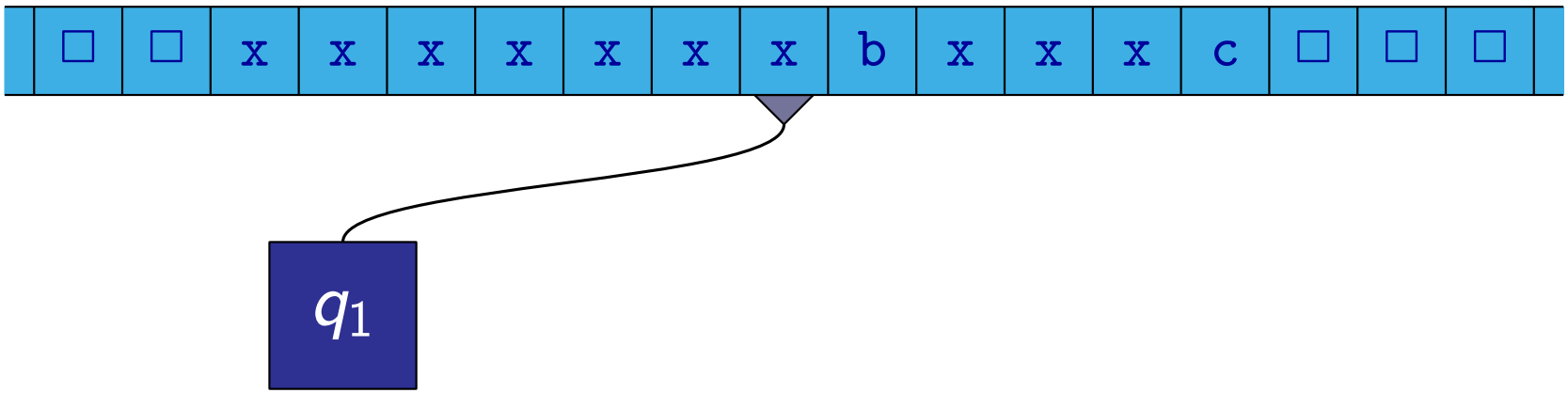
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



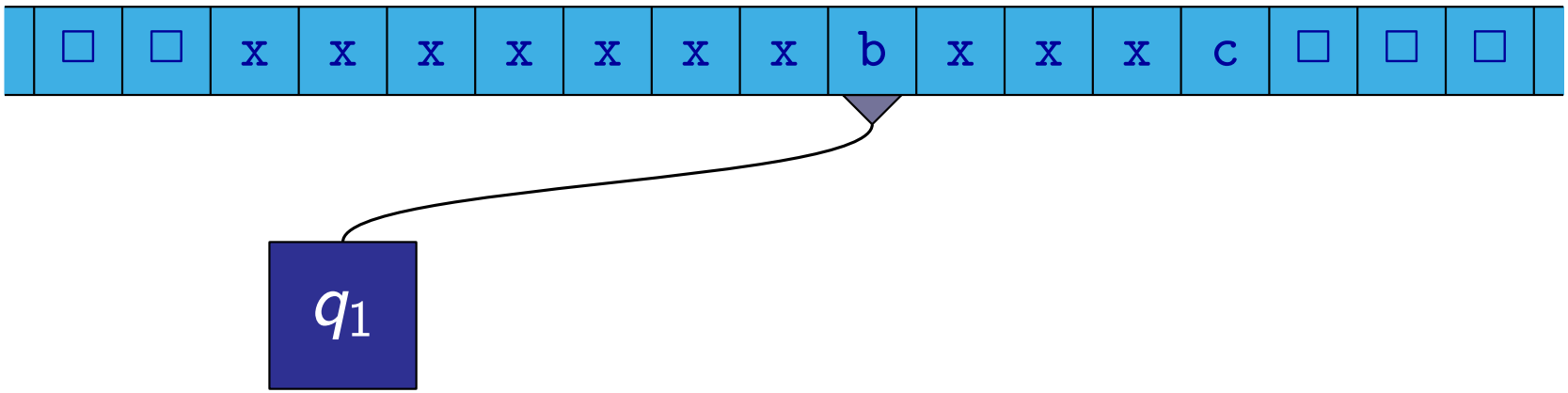
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



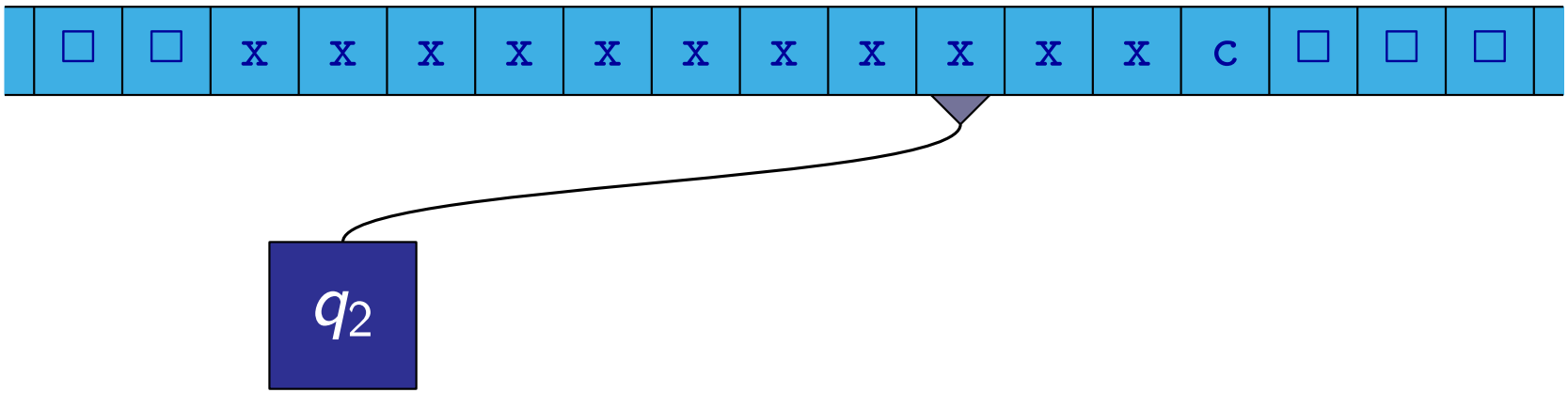
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



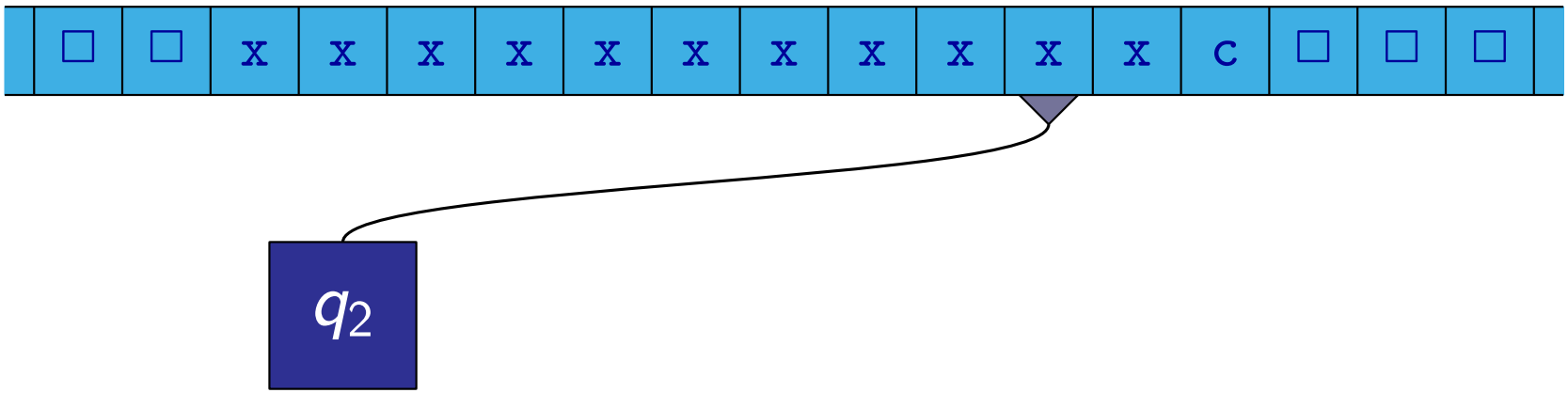
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



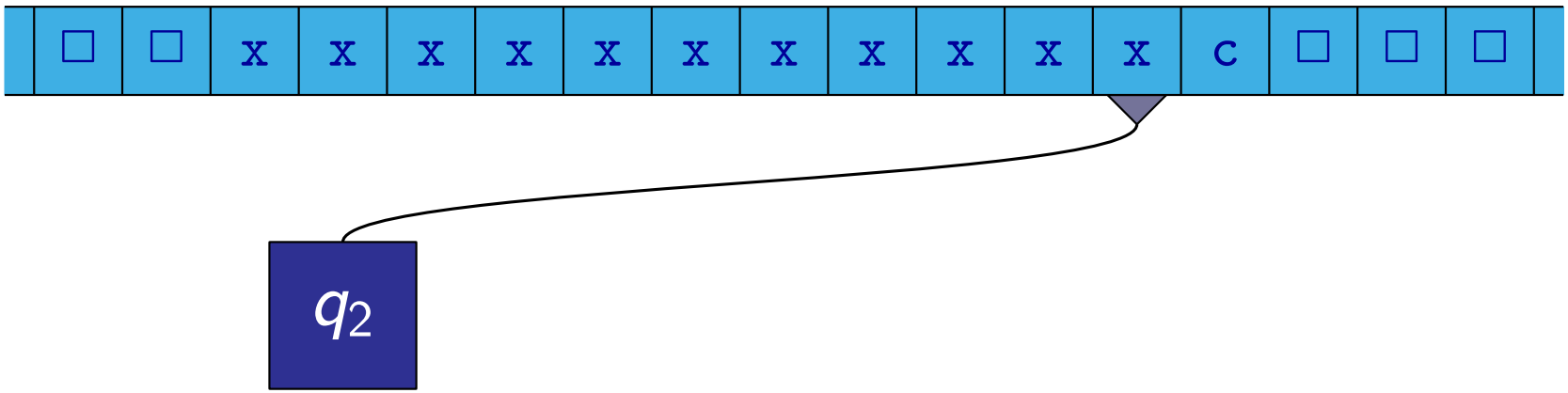
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



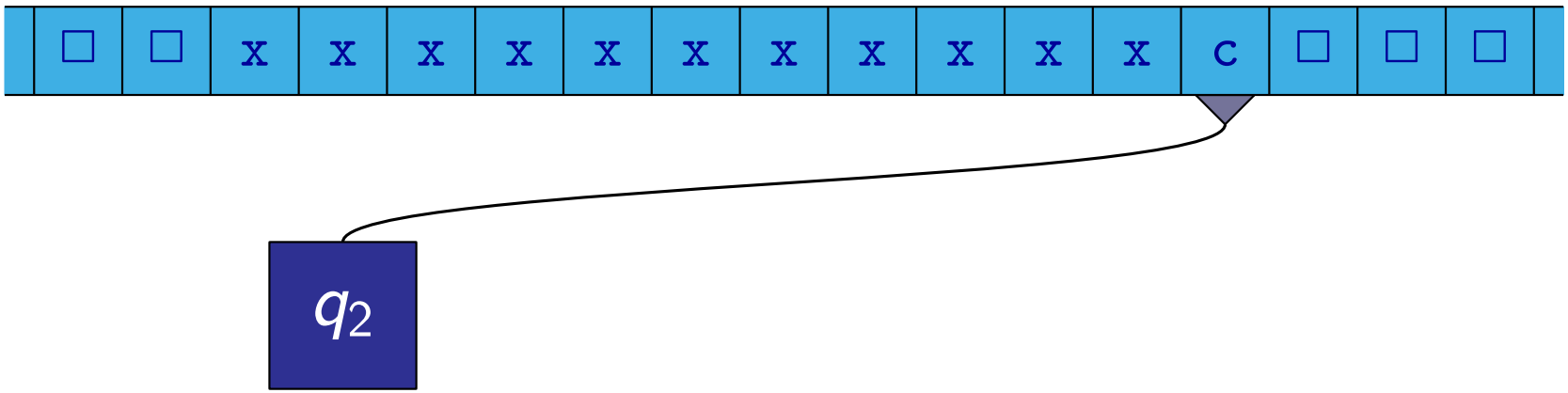
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



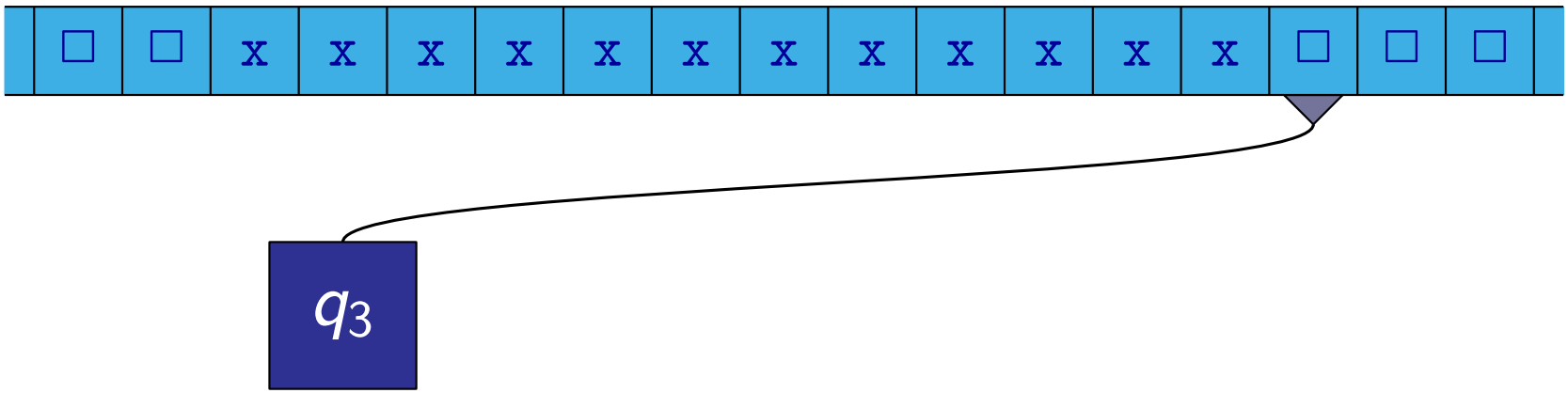
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



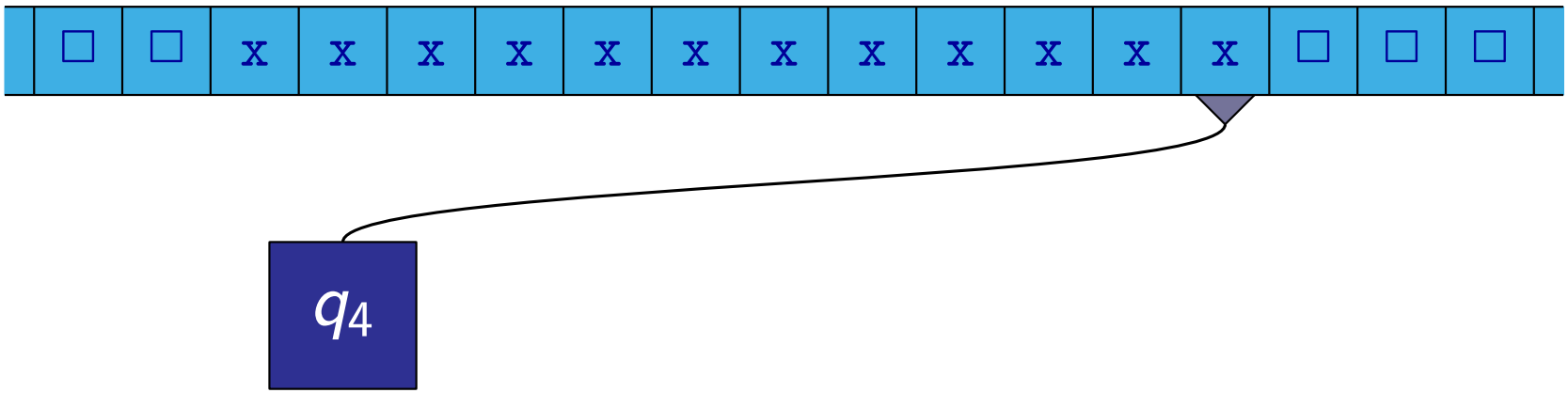
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



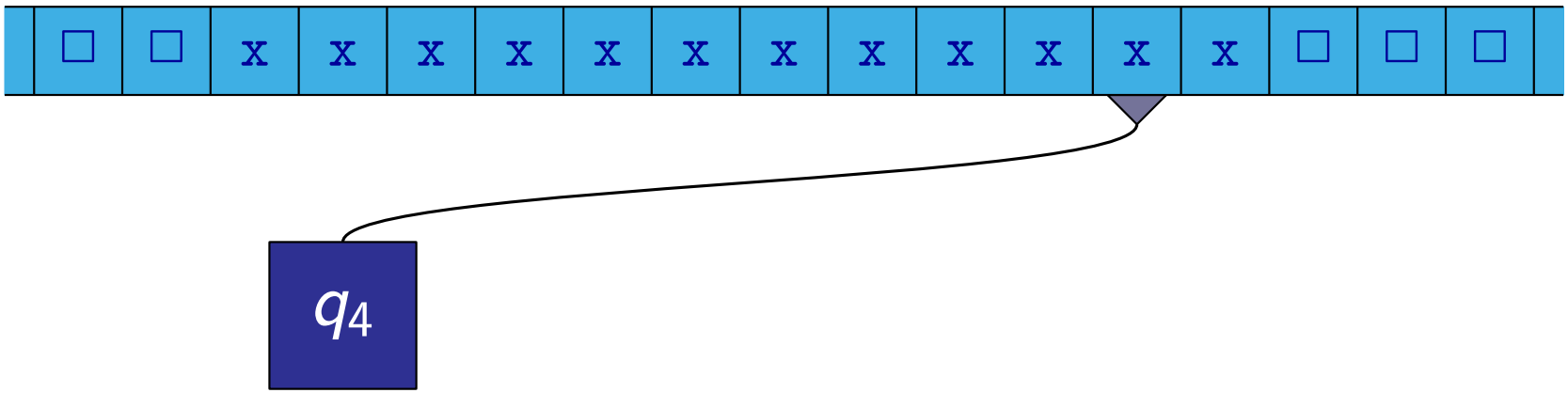
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



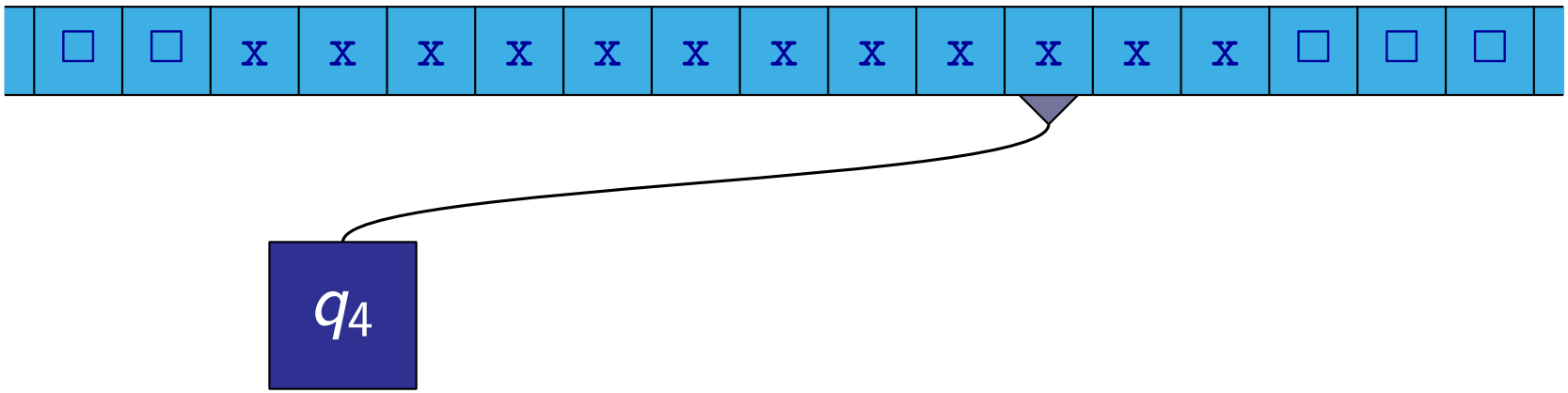
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



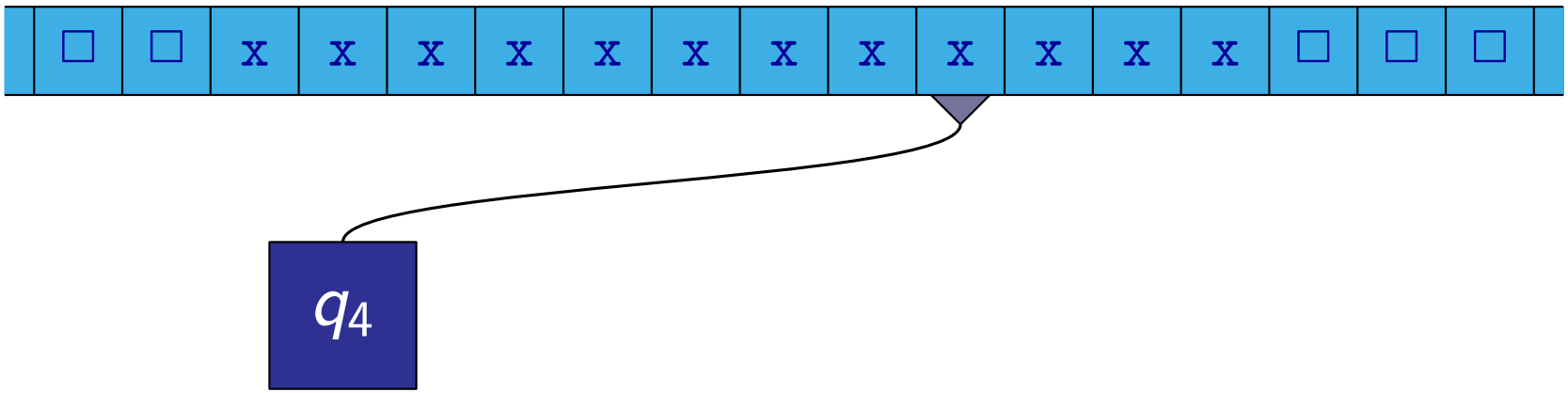
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



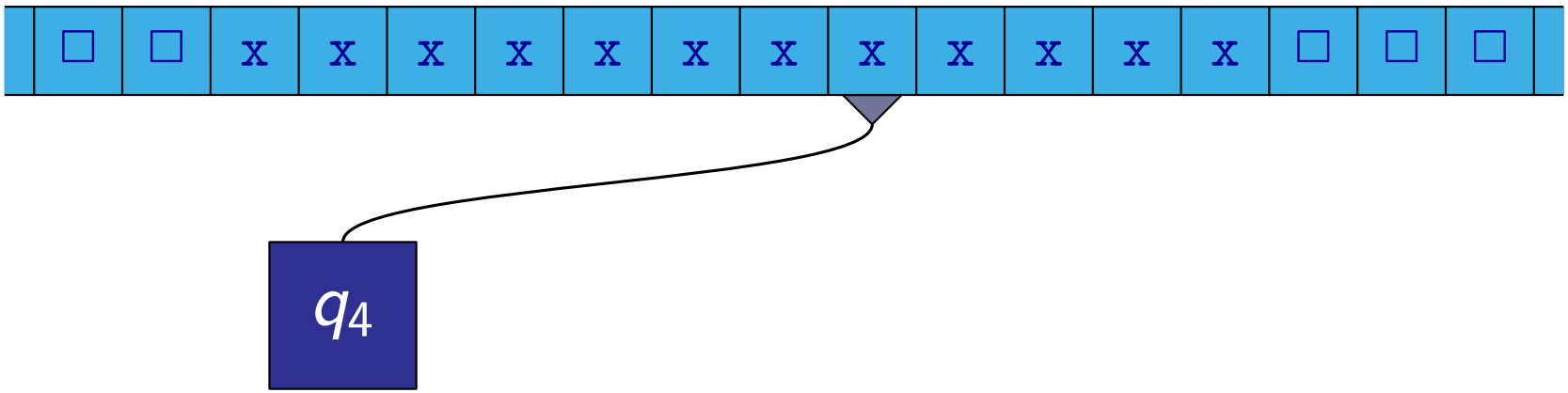
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



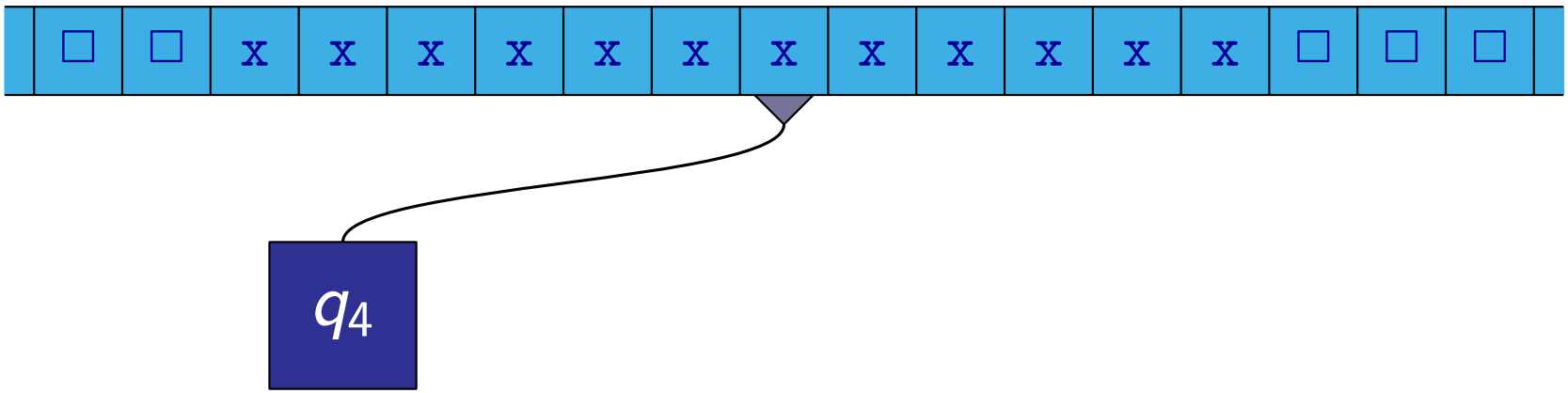
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



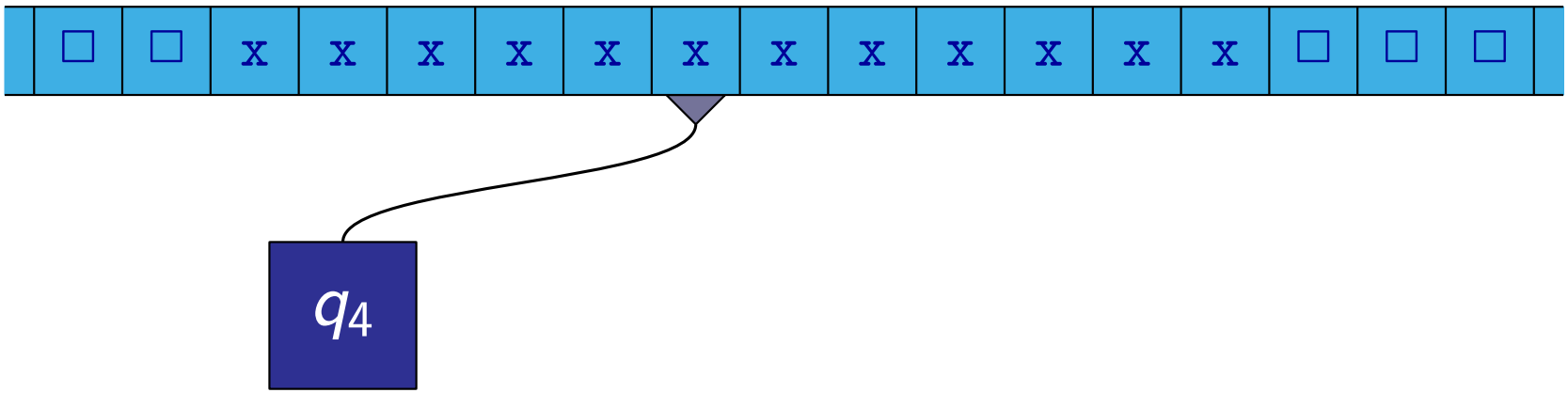
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



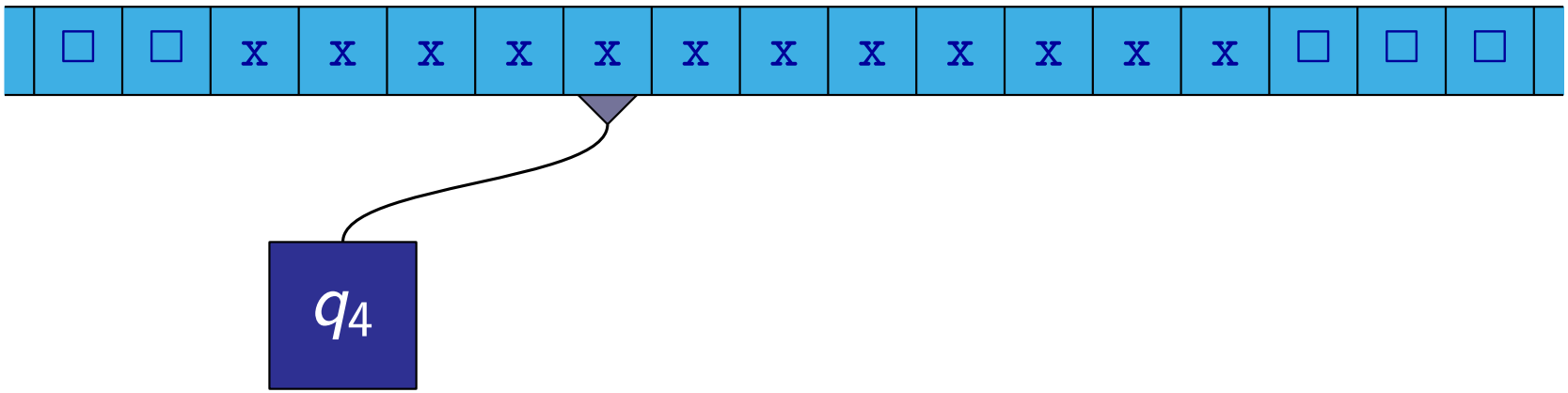
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



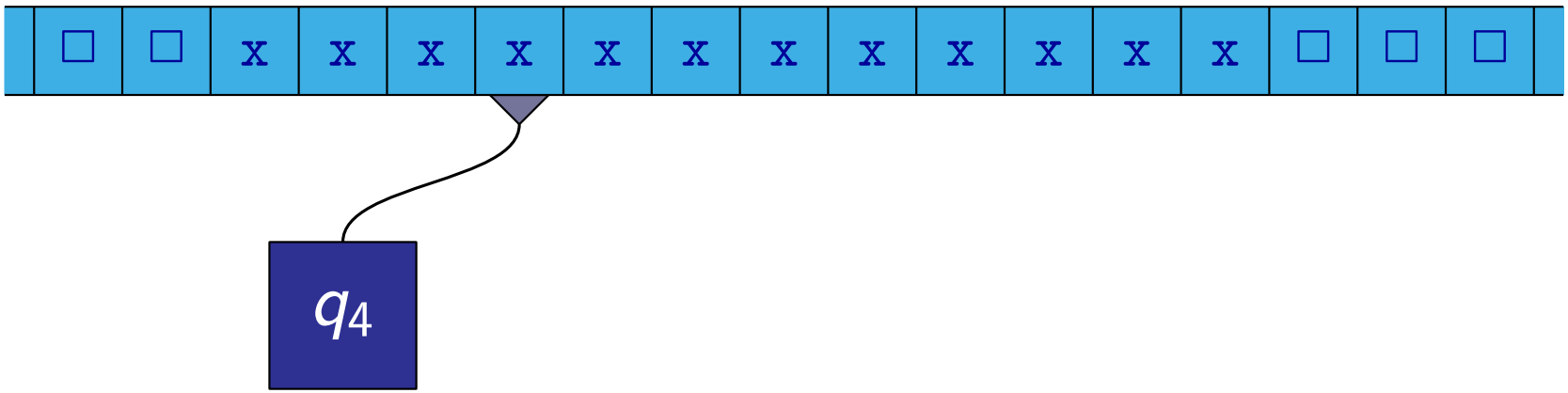
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



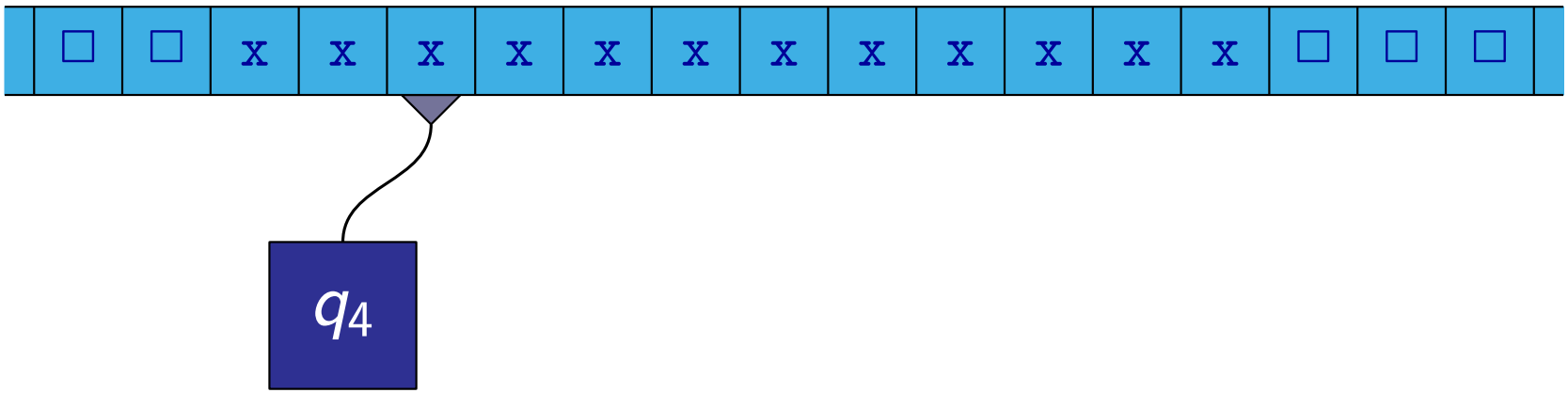
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



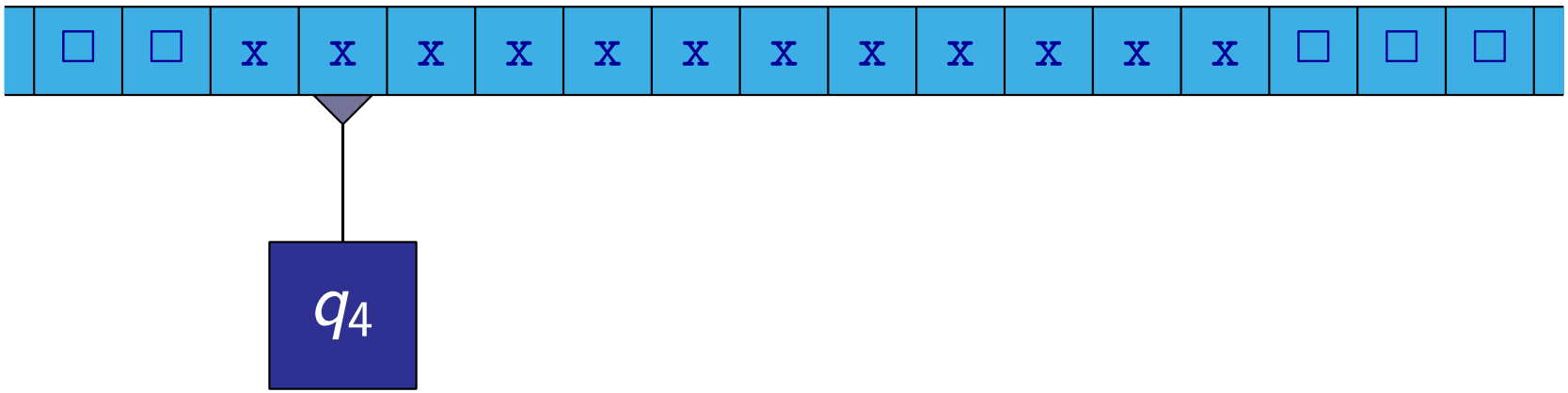
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



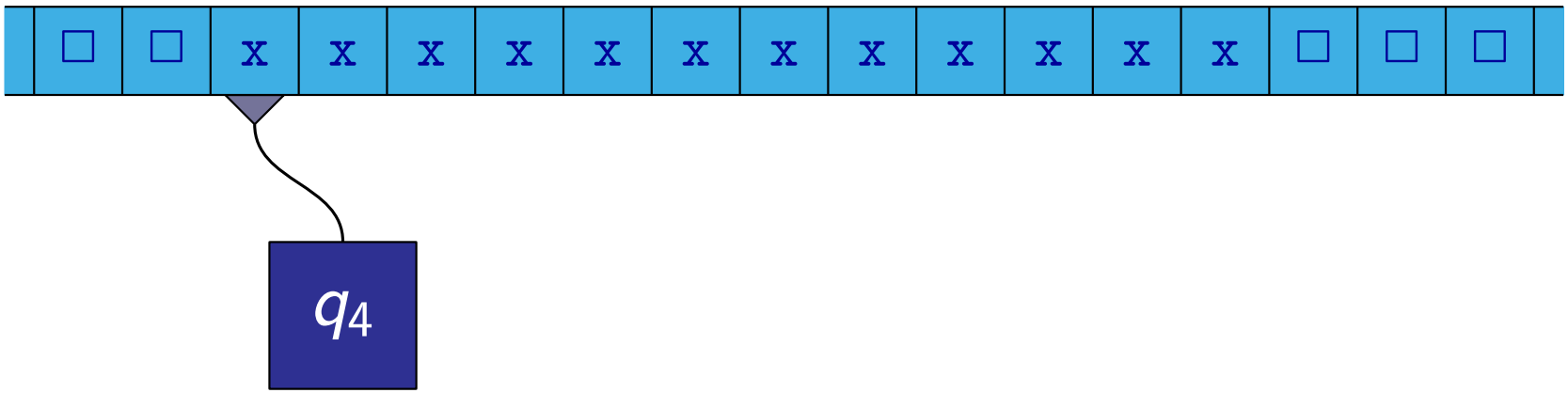
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



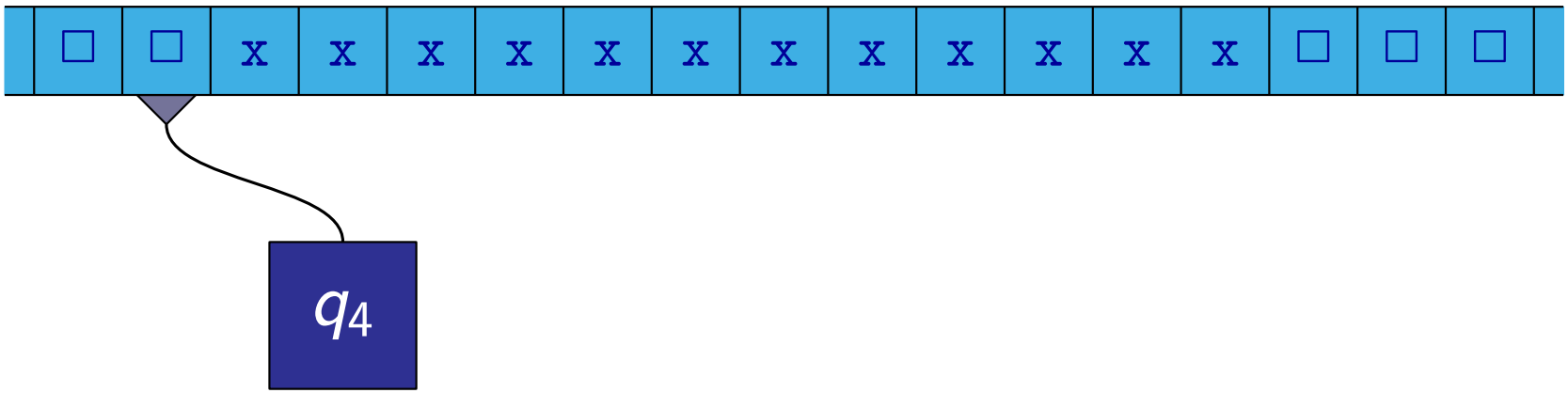
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



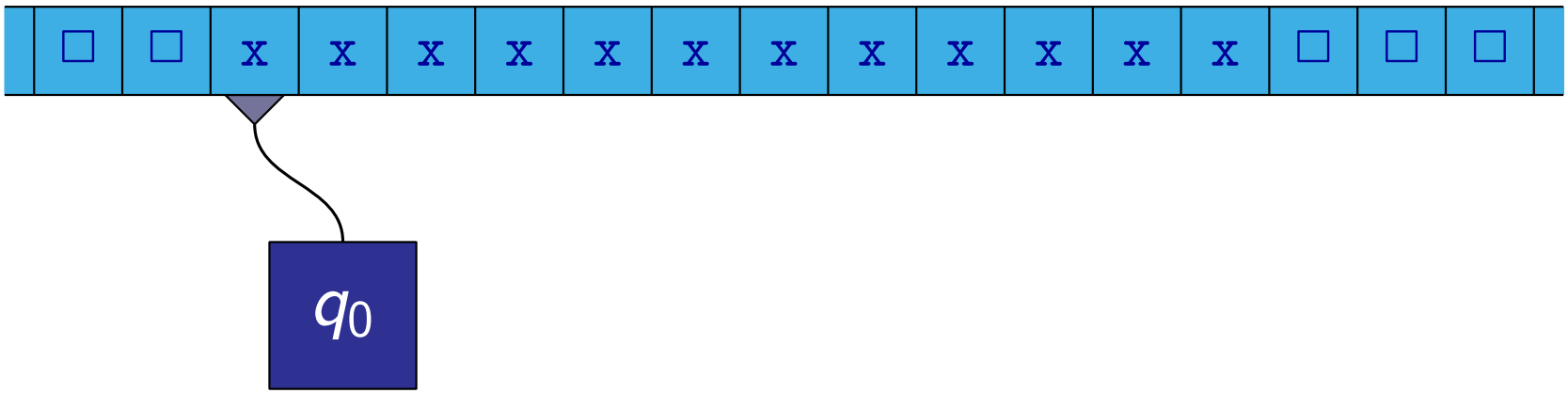
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



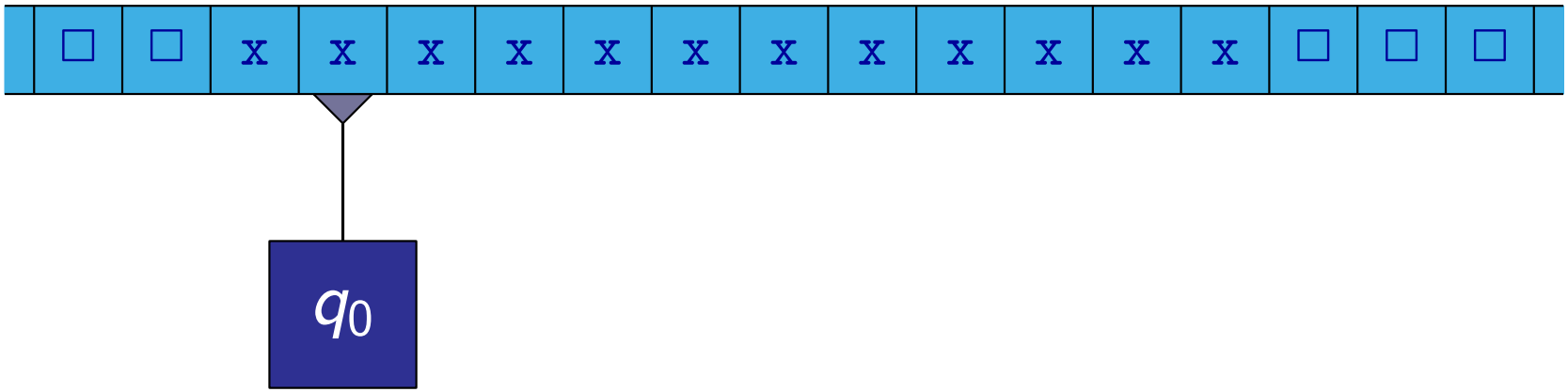
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



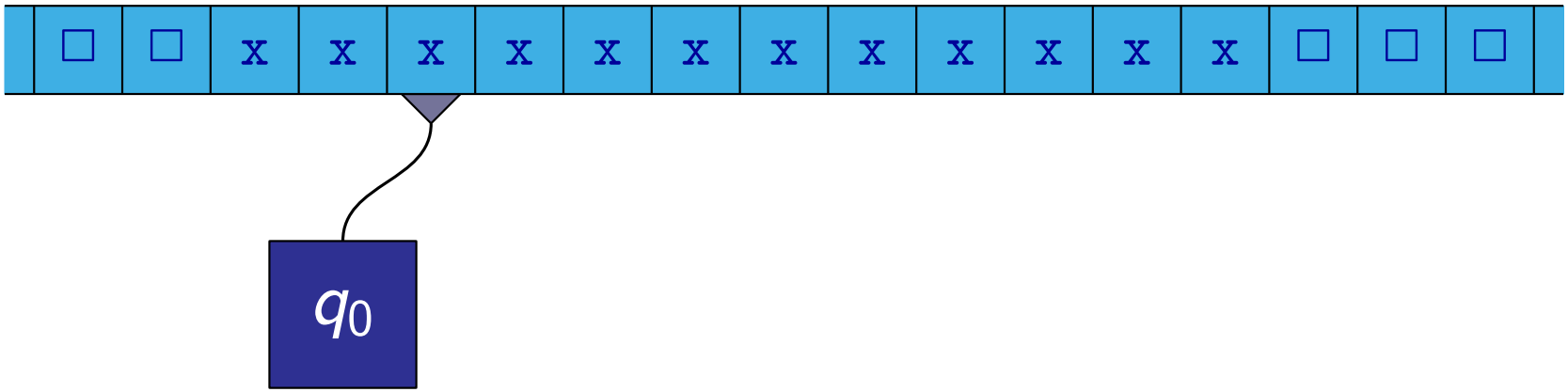
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



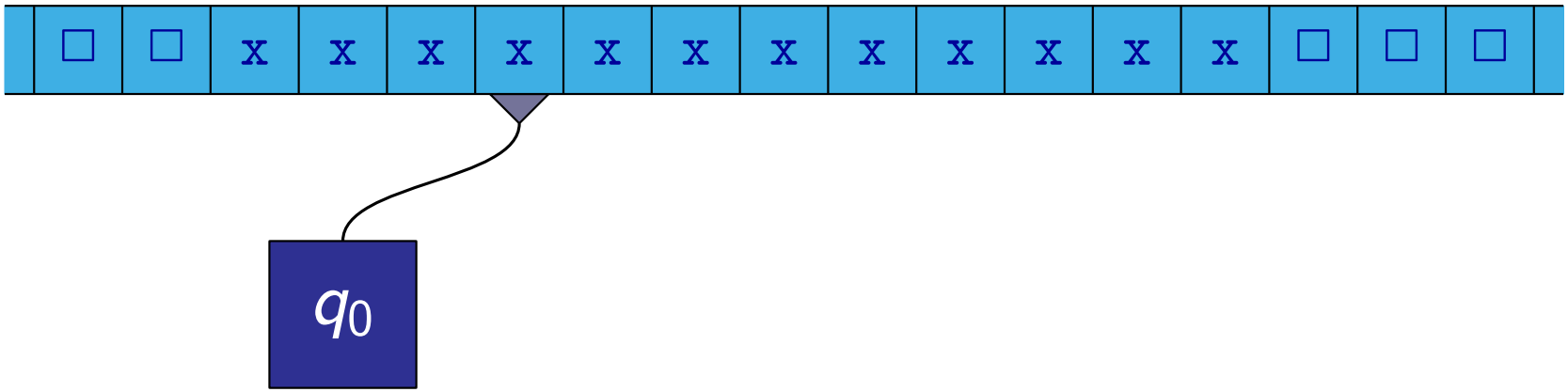
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



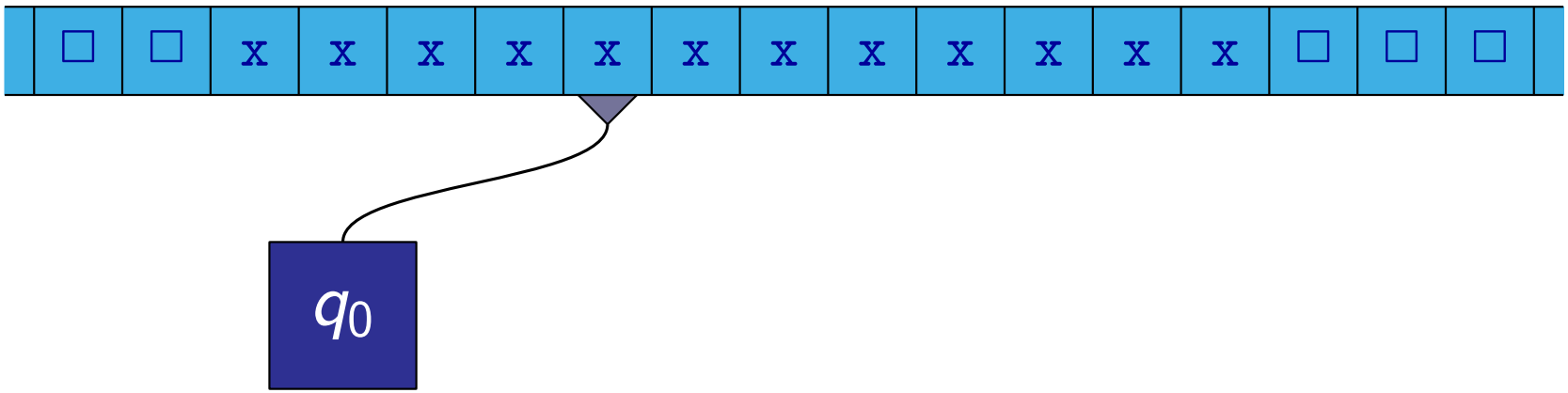
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



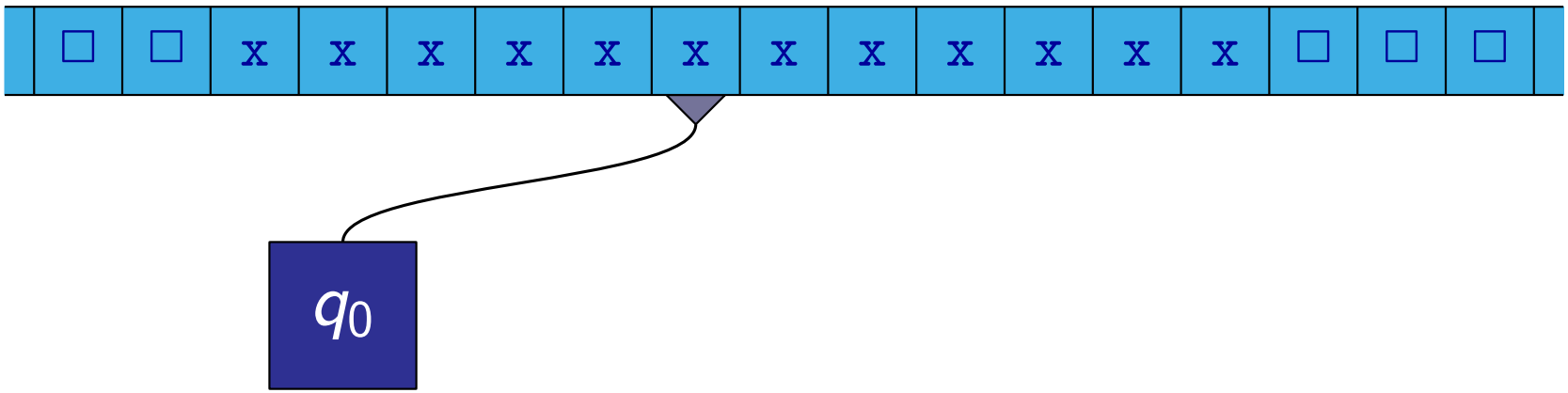
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



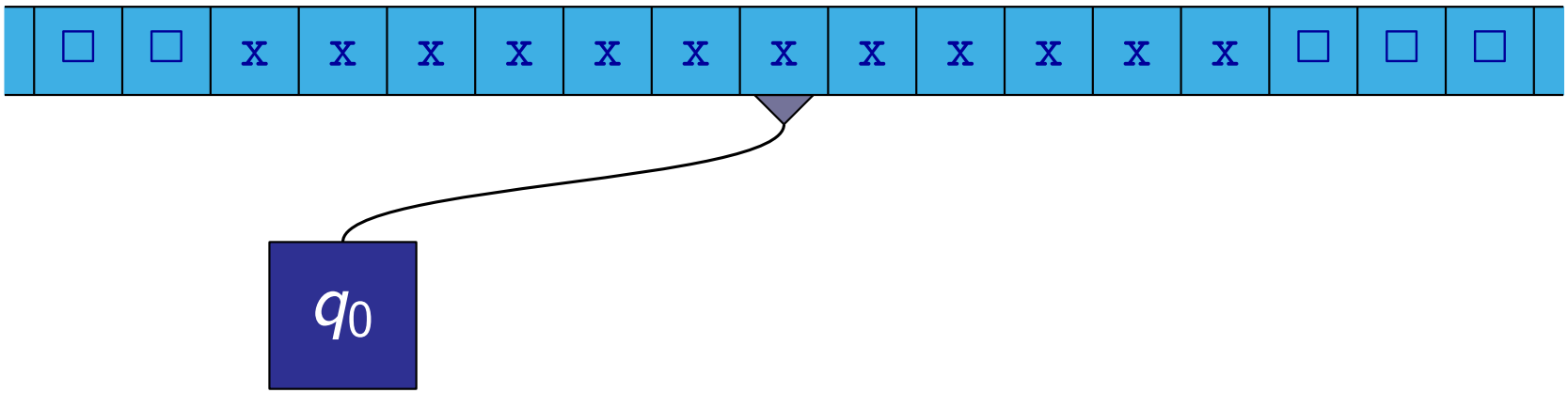
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



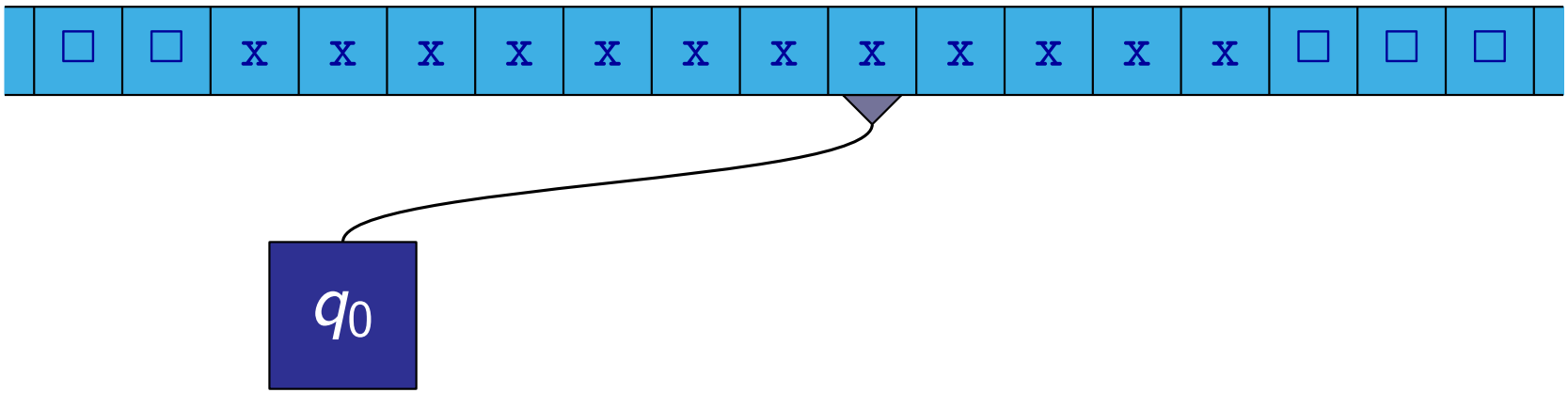
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



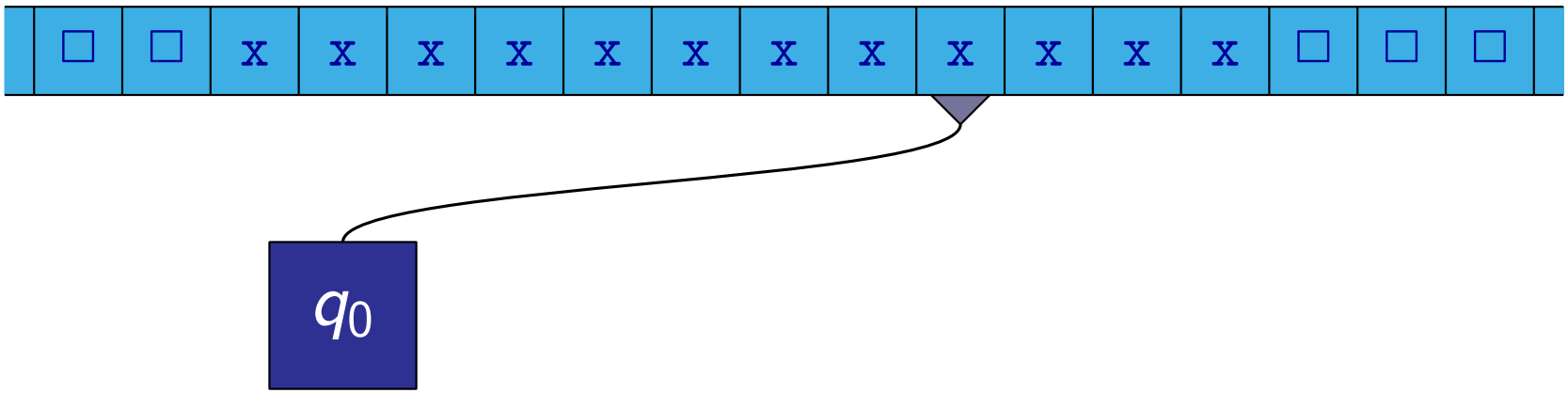
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



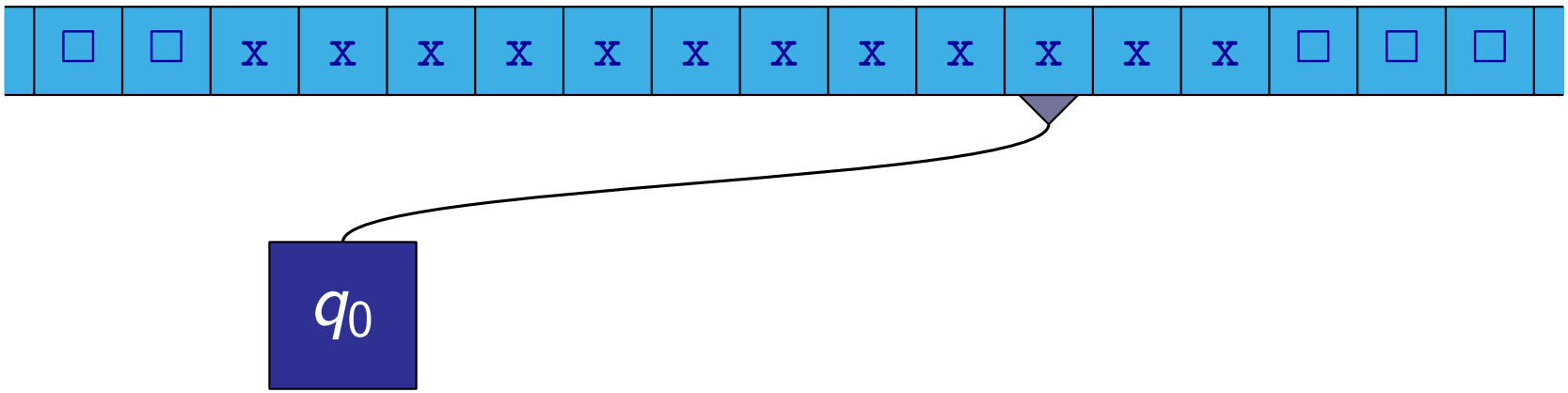
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



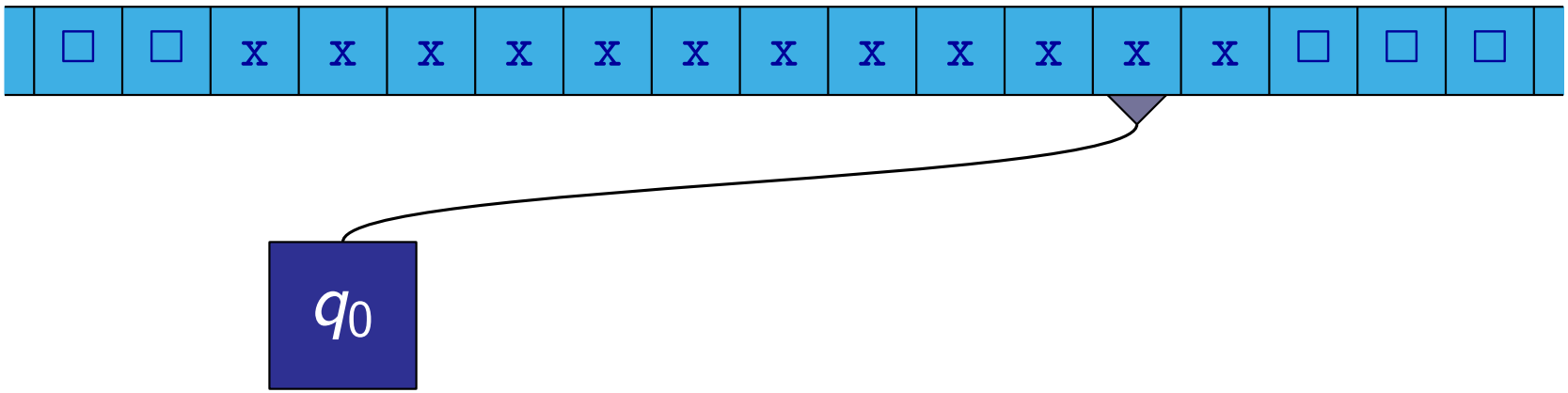
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



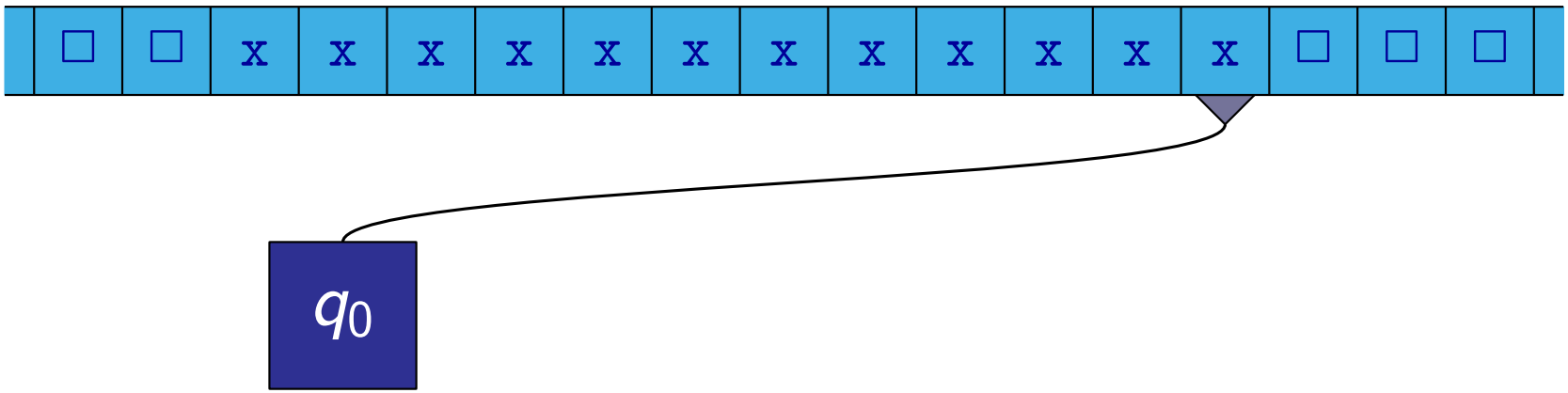
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



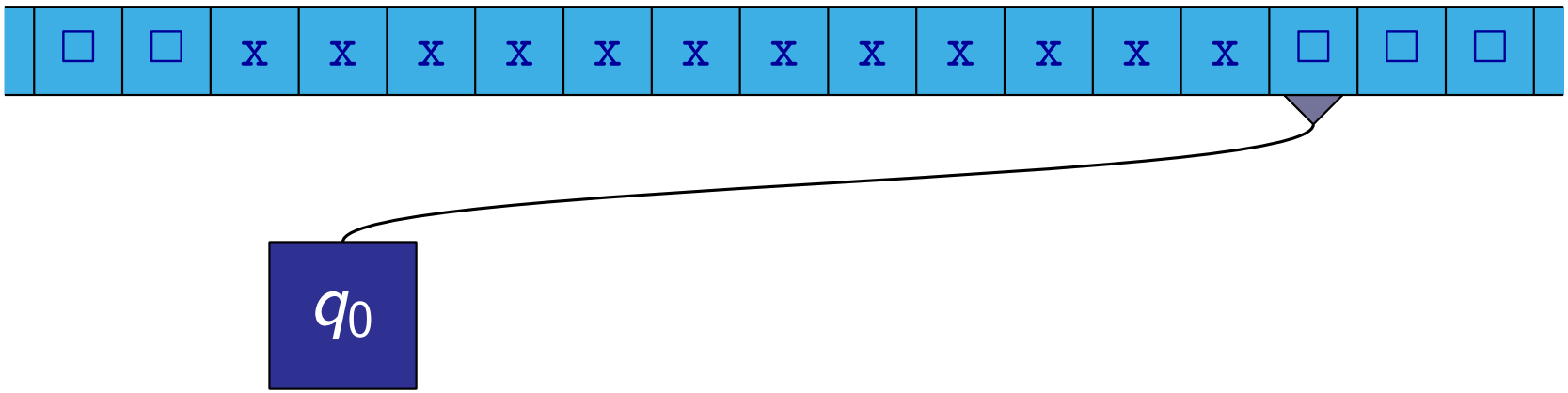
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$



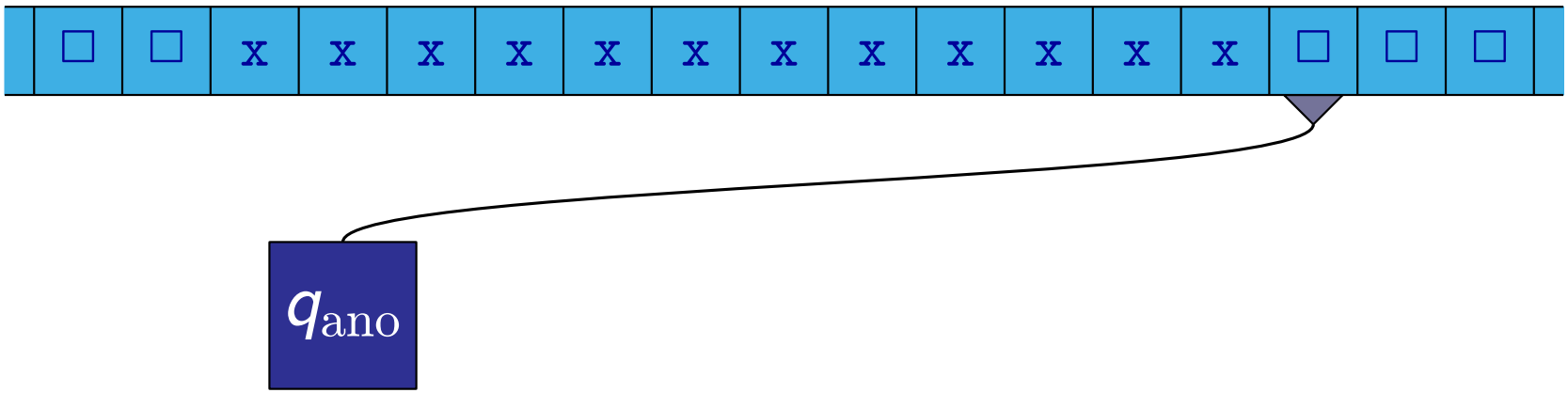
Turingův stroj

Jazyk $L = \{a^n b^n c^n \mid n \geq 0\}$

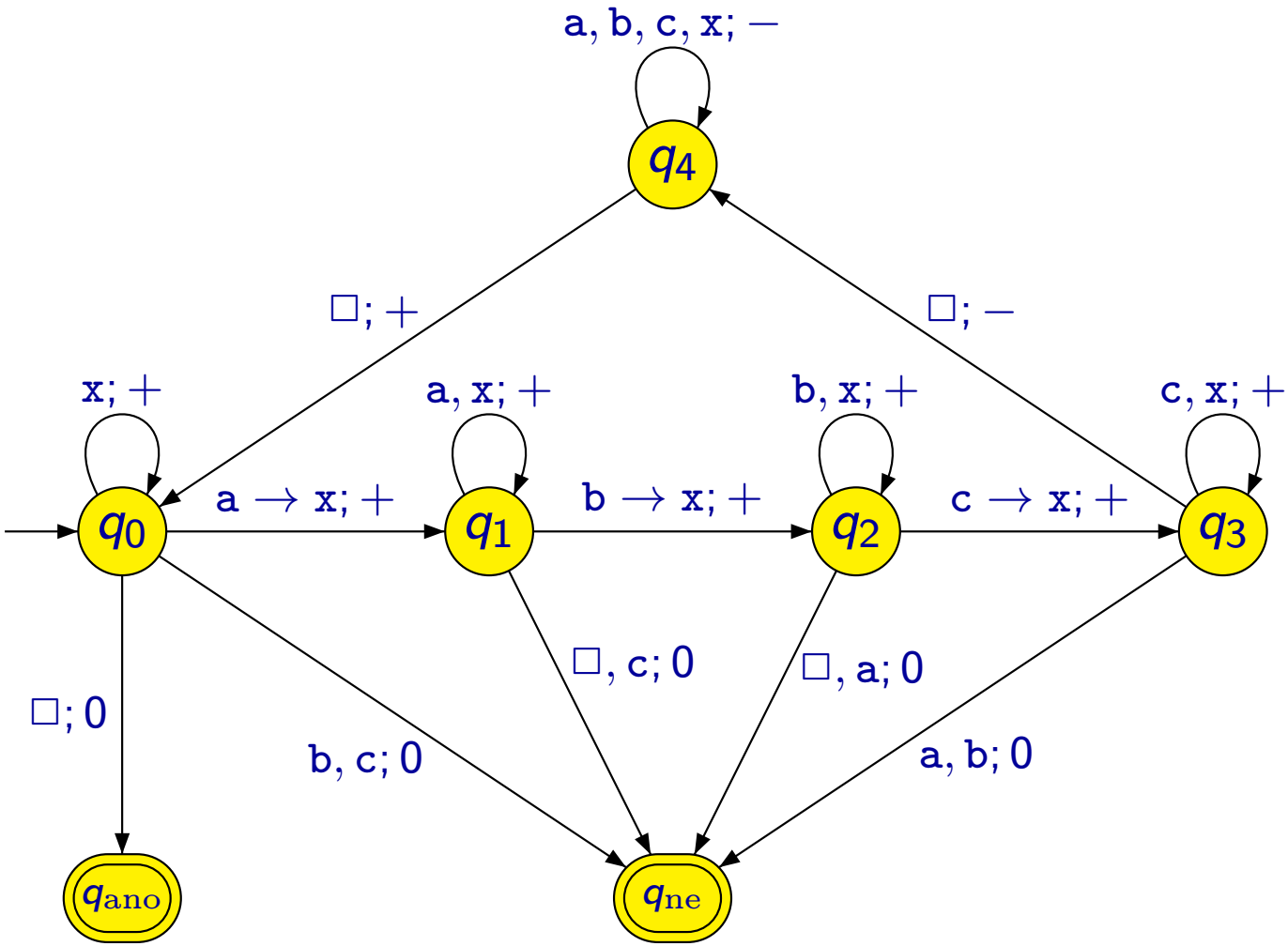
$Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{ano}}, q_{\text{ne}}\}$ $\quad F = \{q_{\text{ano}}, q_{\text{ne}}\}$

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ $\quad \Gamma = \{\square, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{x}\}$

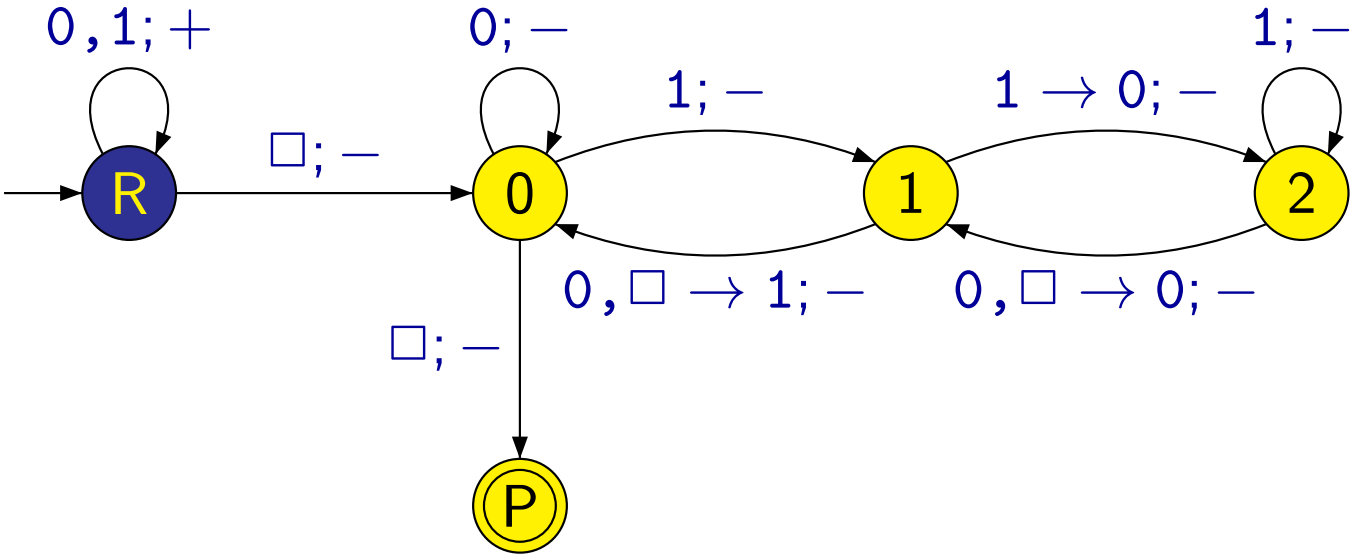
δ	$\square$	a	b	c	x
q_0	$(q_{\text{ano}}, \square, 0)$	$(q_1, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_0, \mathtt{x}, +1)$
q_1	$(q_{\text{ne}}, \square, 0)$	$(q_1, \mathtt{a}, +1)$	$(q_2, \mathtt{x}, +1)$	$(q_{\text{ne}}, \mathtt{c}, 0)$	$(q_1, \mathtt{x}, +1)$
q_2	$(q_{\text{ne}}, \square, 0)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_2, \mathtt{b}, +1)$	$(q_3, \mathtt{x}, +1)$	$(q_2, \mathtt{x}, +1)$
q_3	$(q_4, \square, -1)$	$(q_{\text{ne}}, \mathtt{a}, 0)$	$(q_{\text{ne}}, \mathtt{b}, 0)$	$(q_3, \mathtt{c}, +1)$	$(q_3, \mathtt{x}, +1)$
q_4	$(q_0, \square, +1)$	$(q_4, \mathtt{a}, -1)$	$(q_4, \mathtt{b}, -1)$	$(q_4, \mathtt{c}, -1)$	$(q_4, \mathtt{x}, -1)$

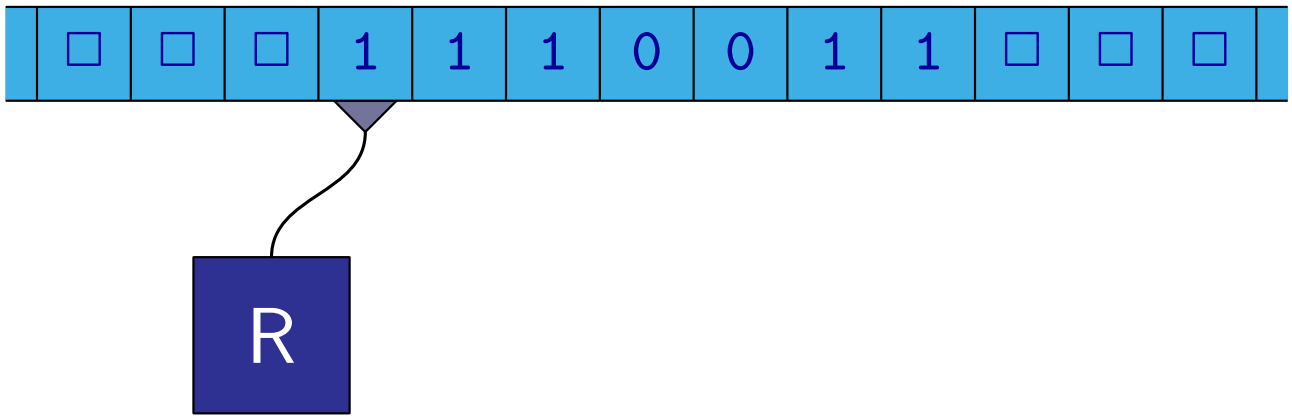


Turingův stroj

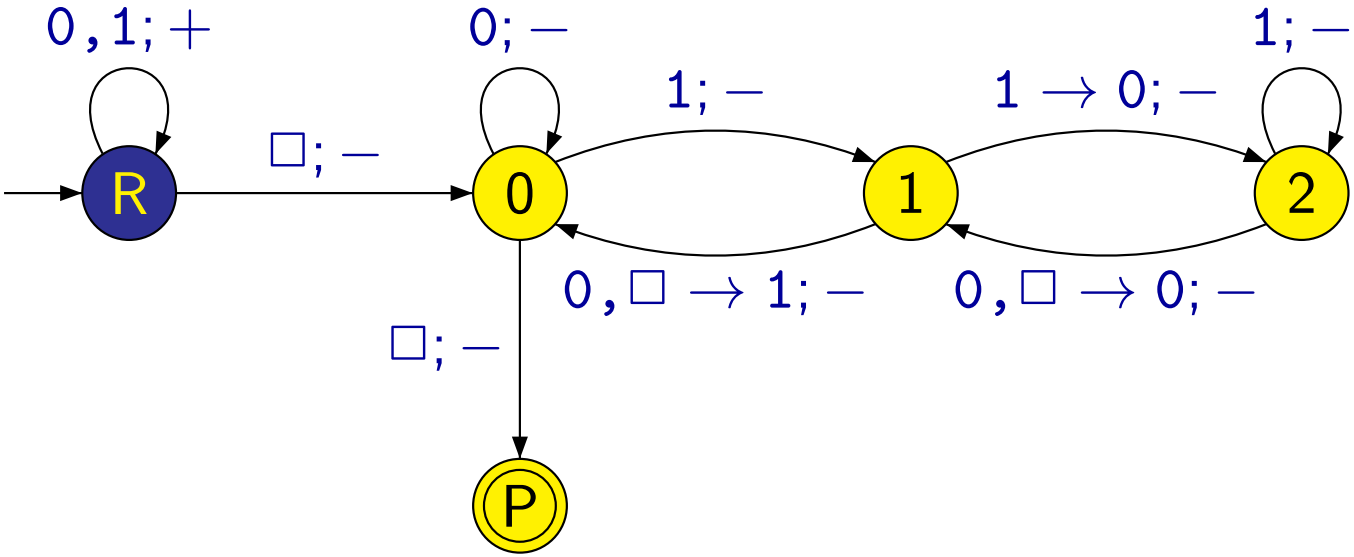


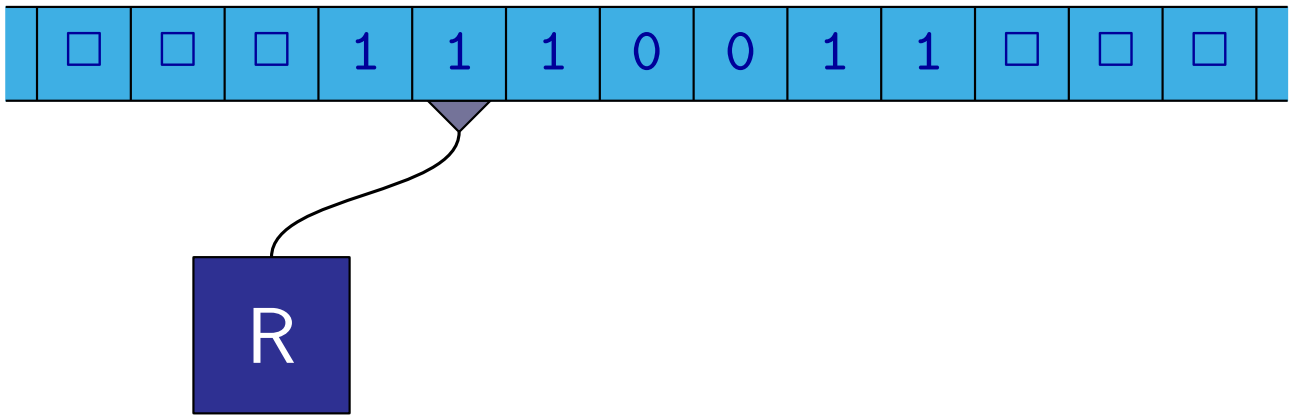
Turingův stroj – násobení třemi



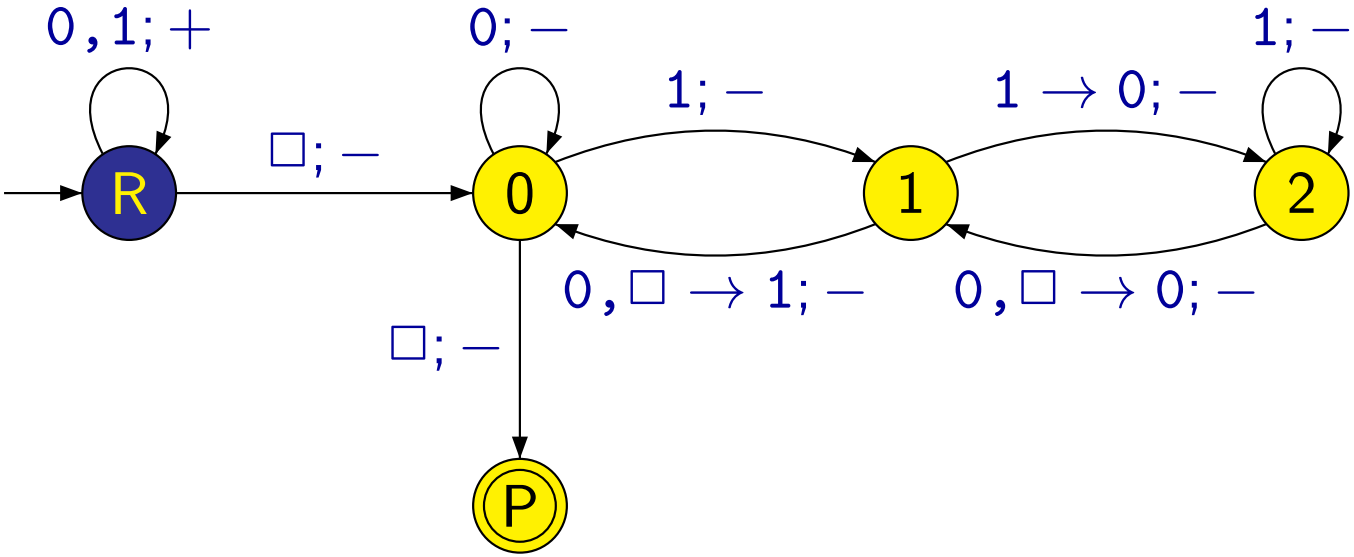


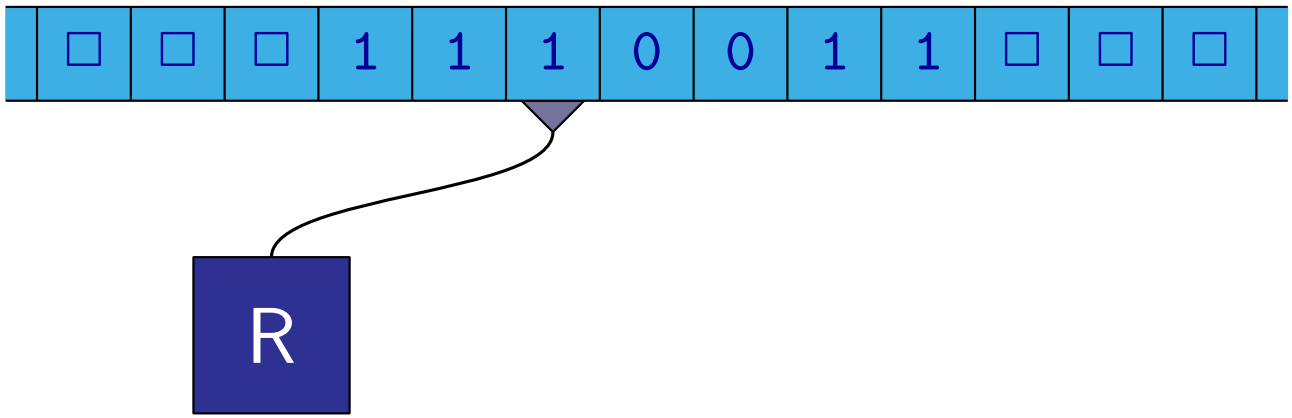
Turingův stroj – násobení třemi



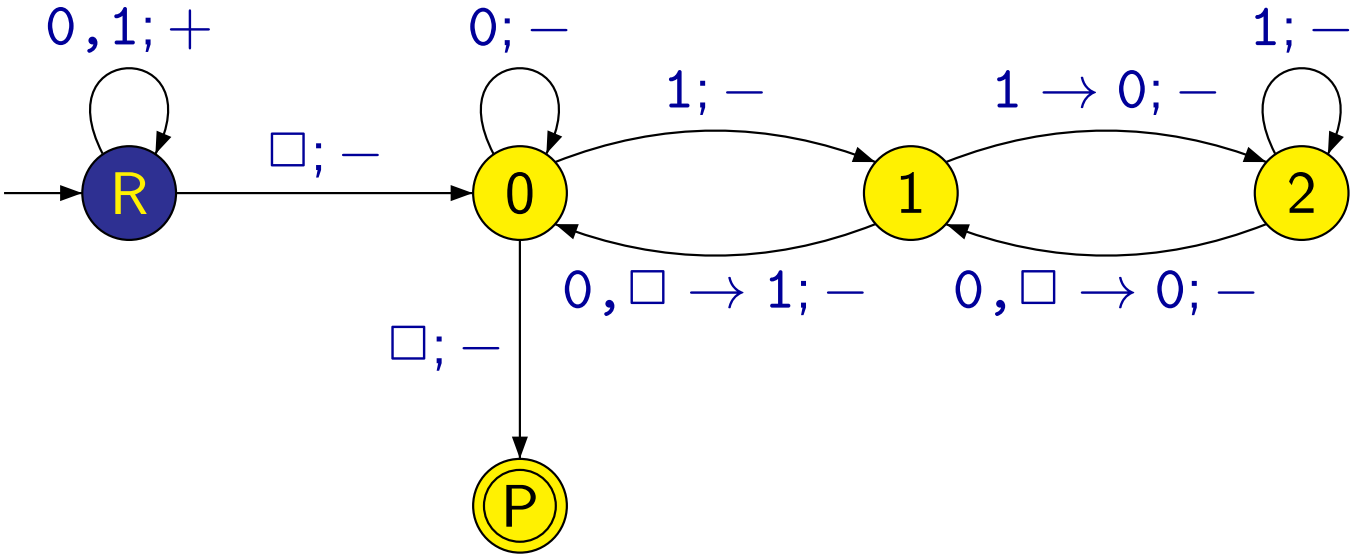


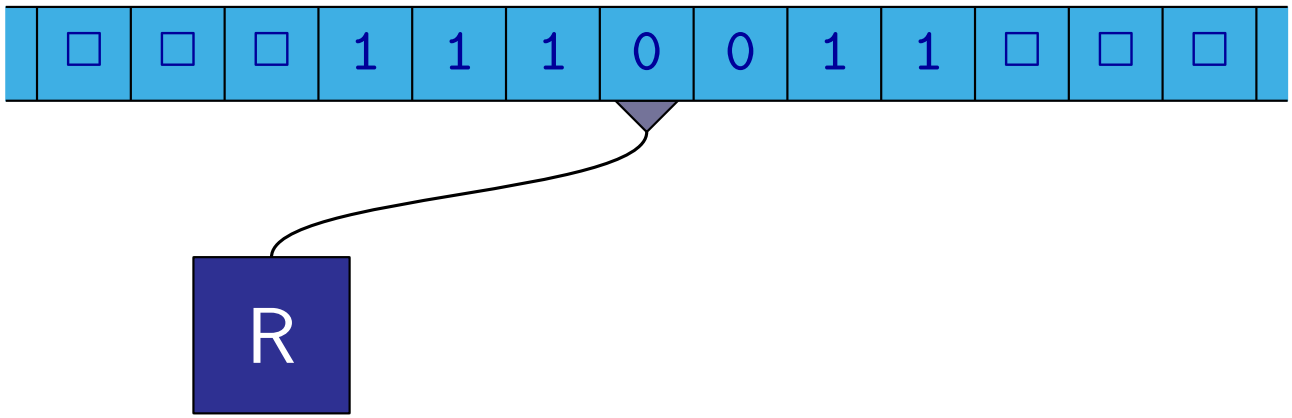
Turingův stroj – násobení třemi



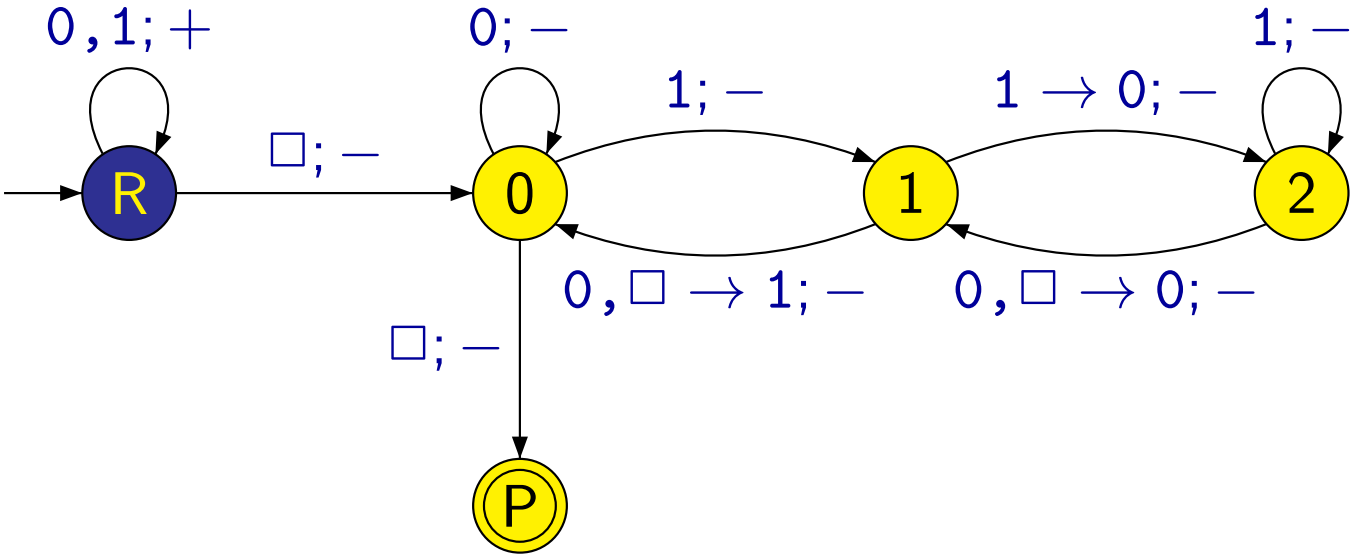


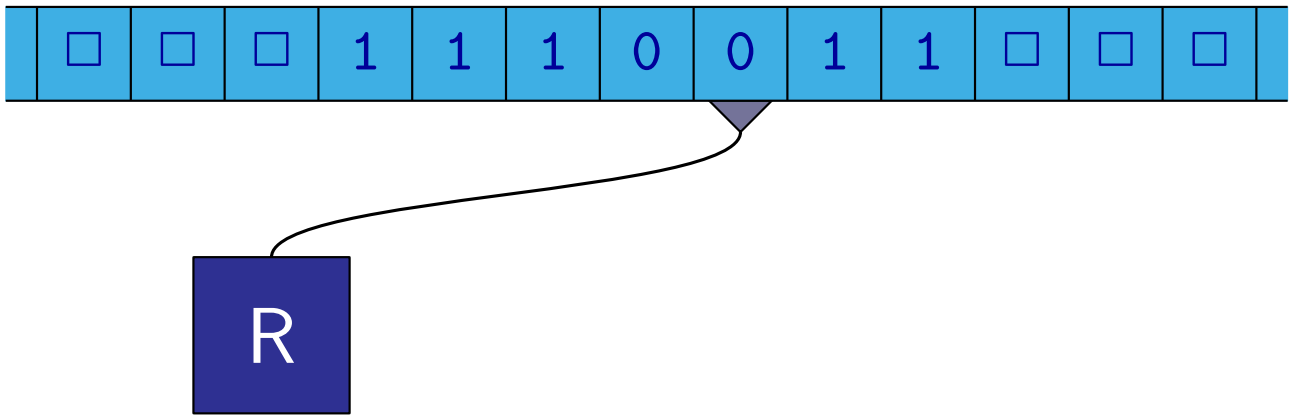
Turingův stroj – násobení třemi



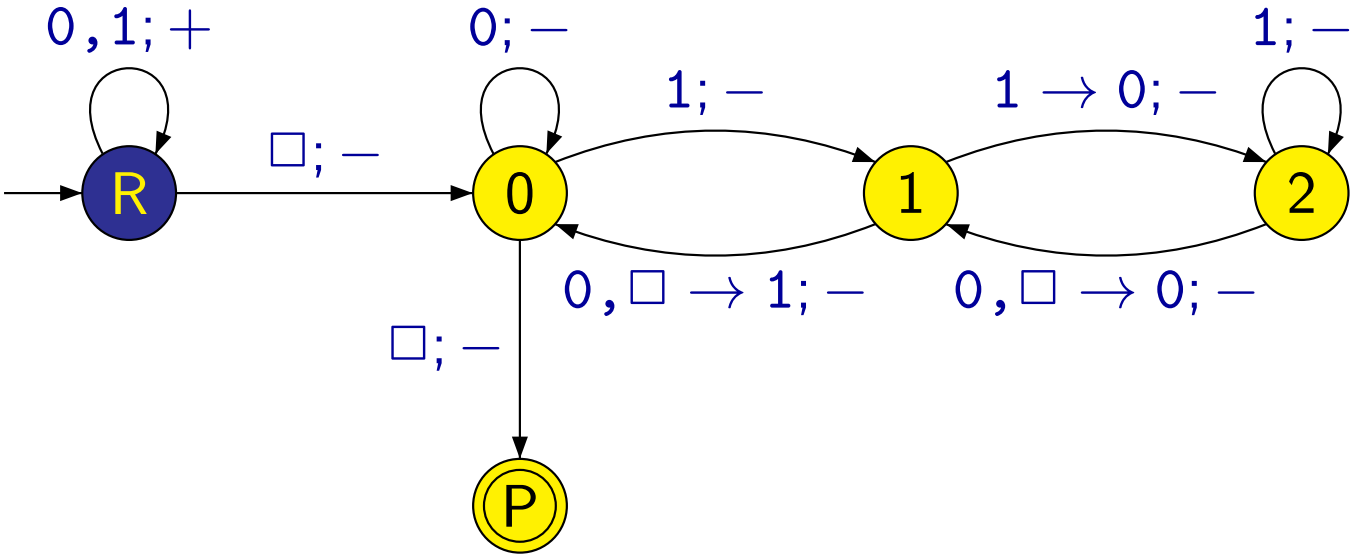


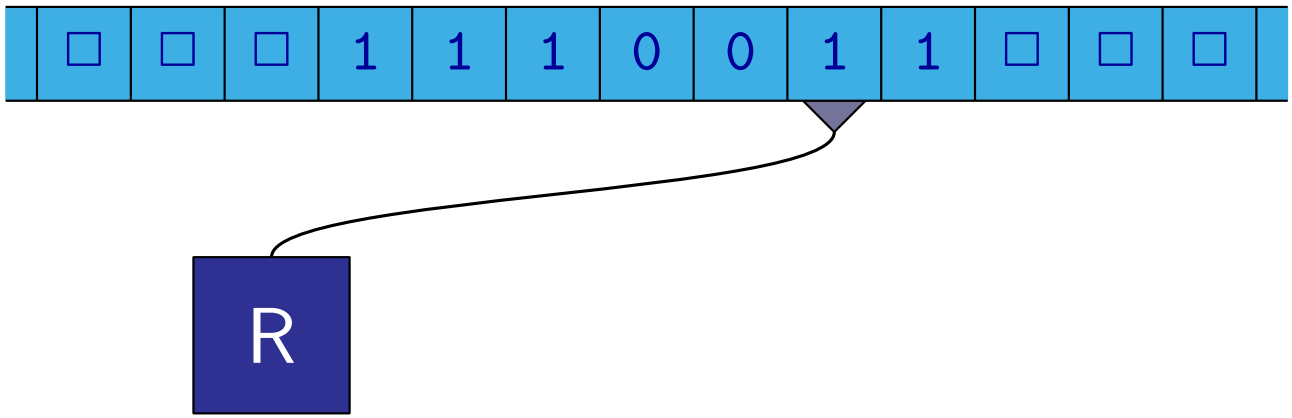
Turingův stroj – násobení třemi



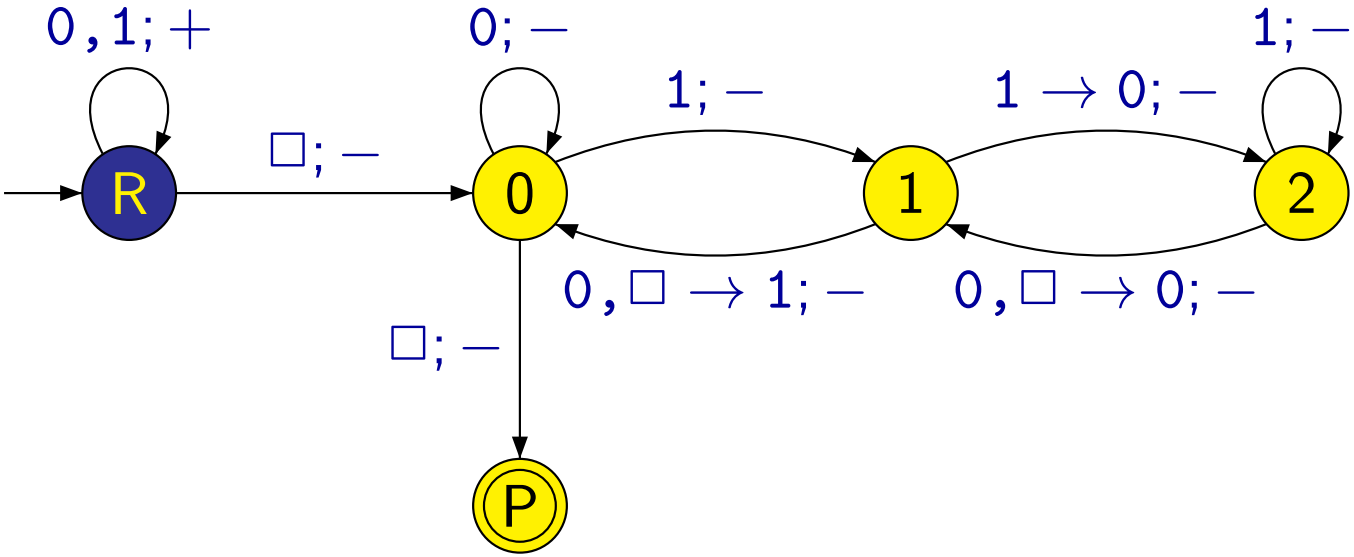


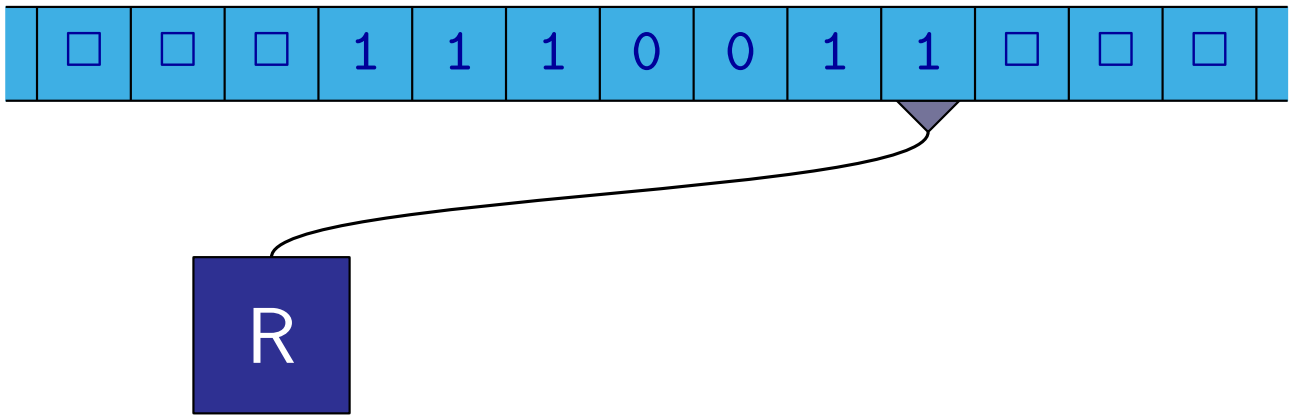
Turingův stroj – násobení třemi



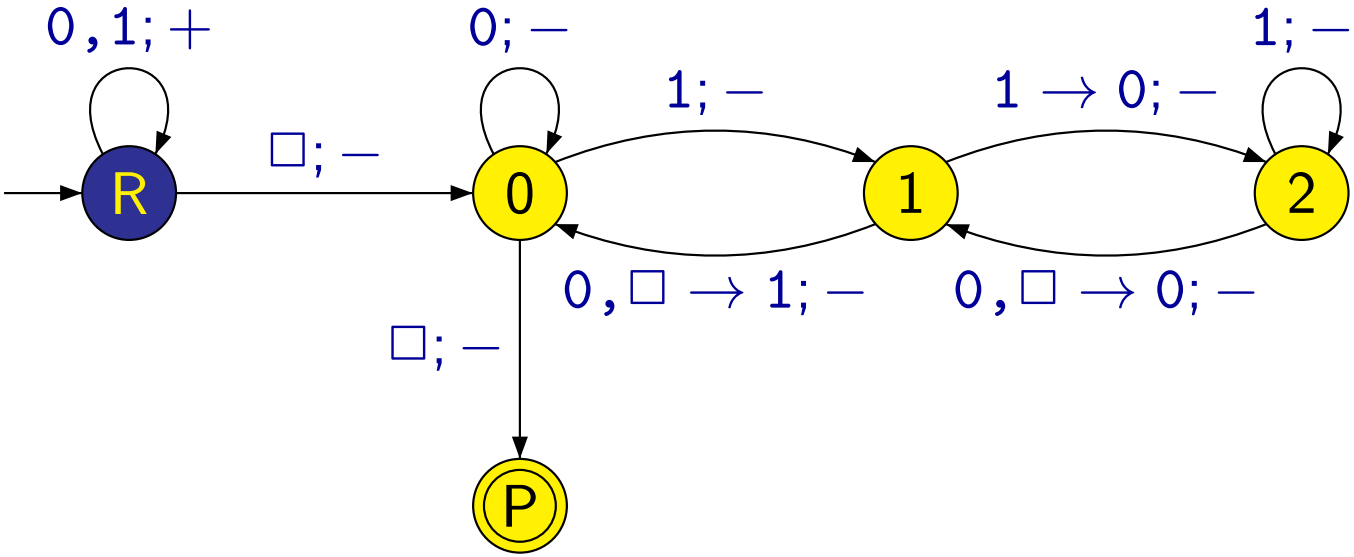


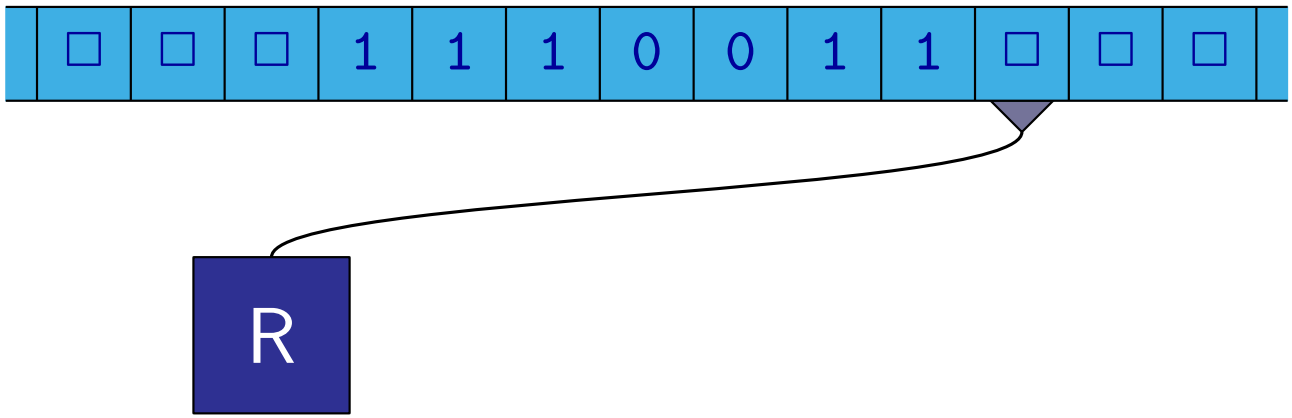
Turingův stroj – násobení třemi



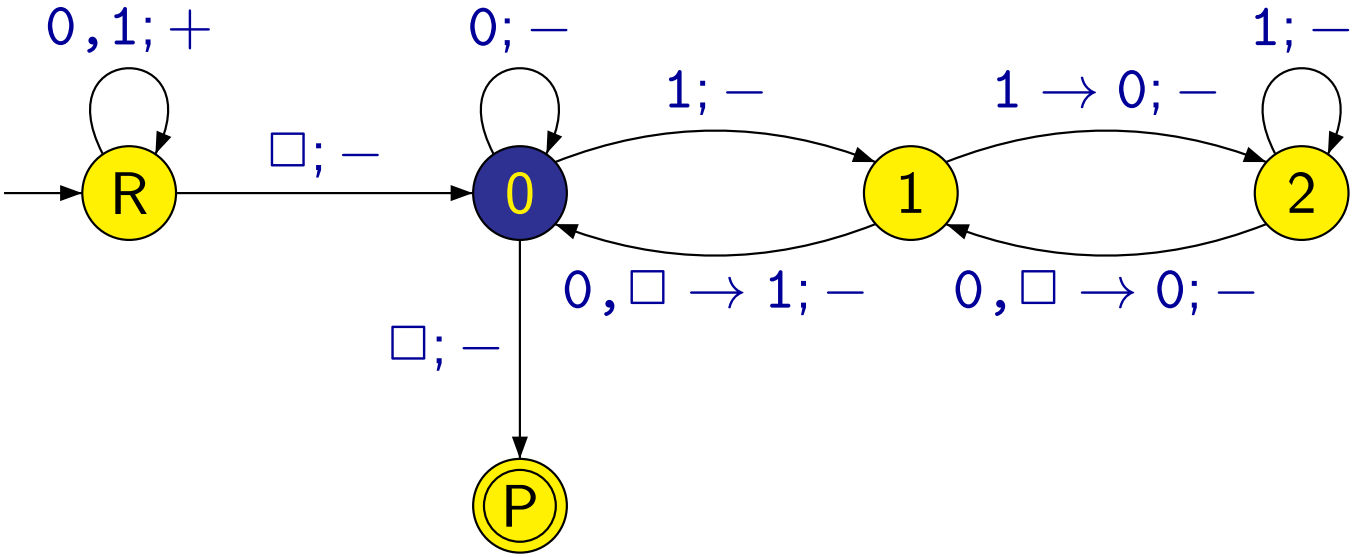


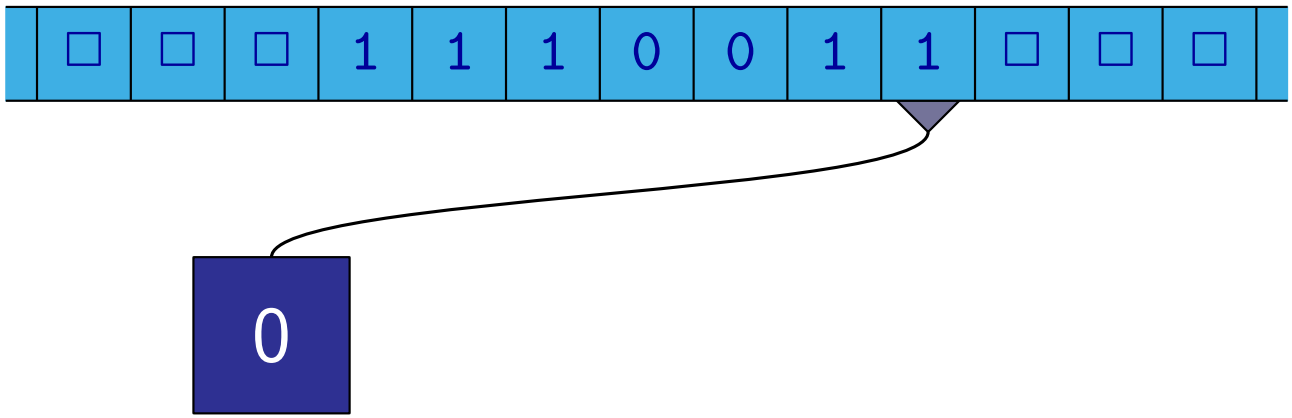
Turingův stroj – násobení třemi



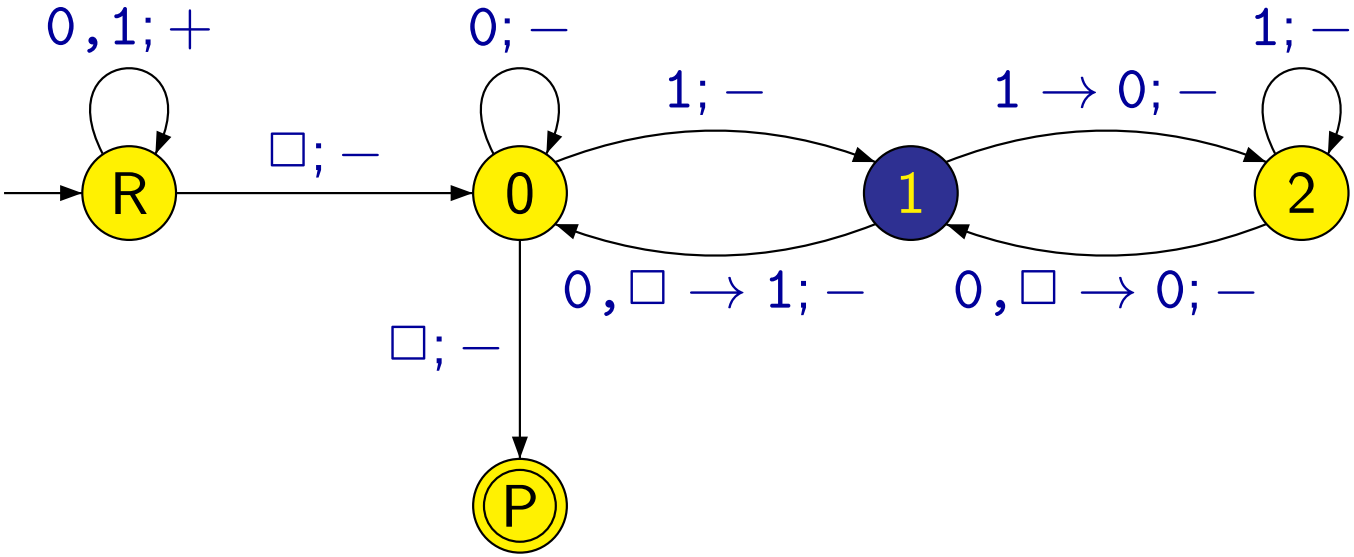


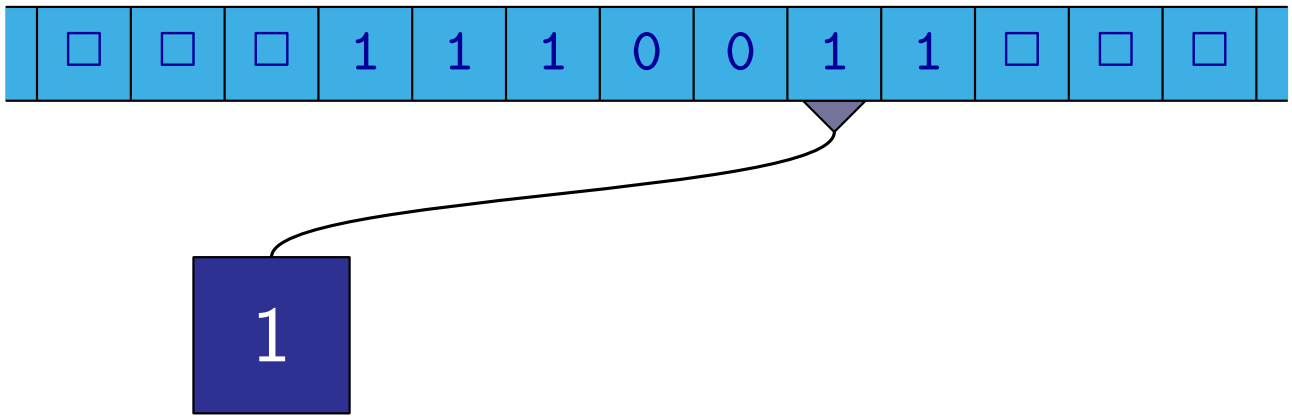
Turingův stroj – násobení třemi



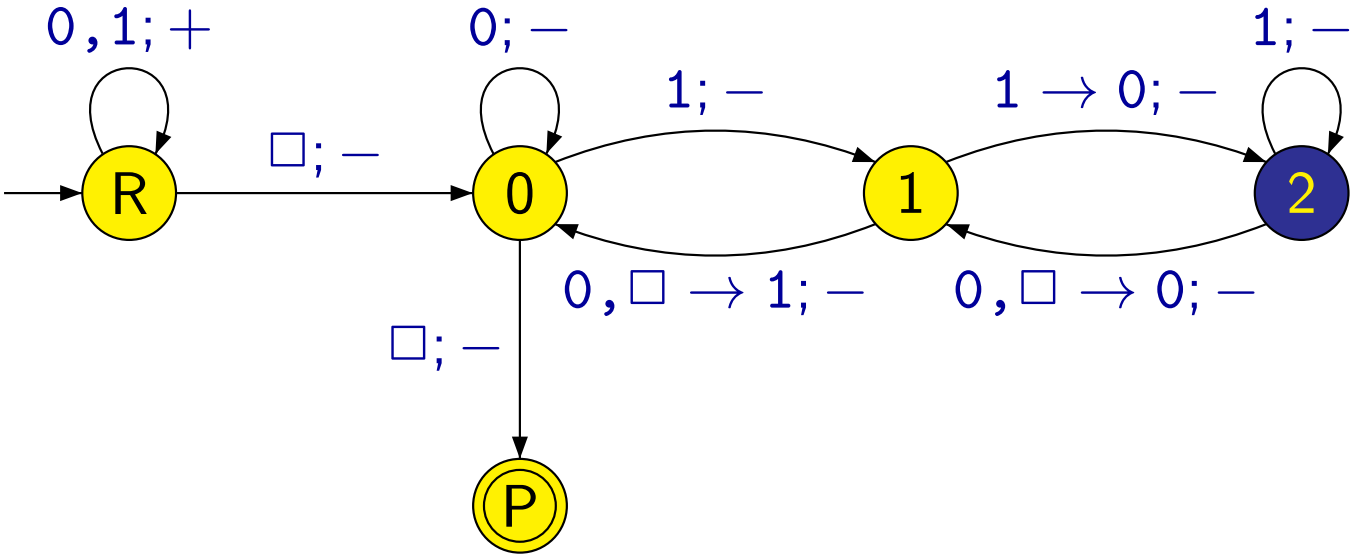


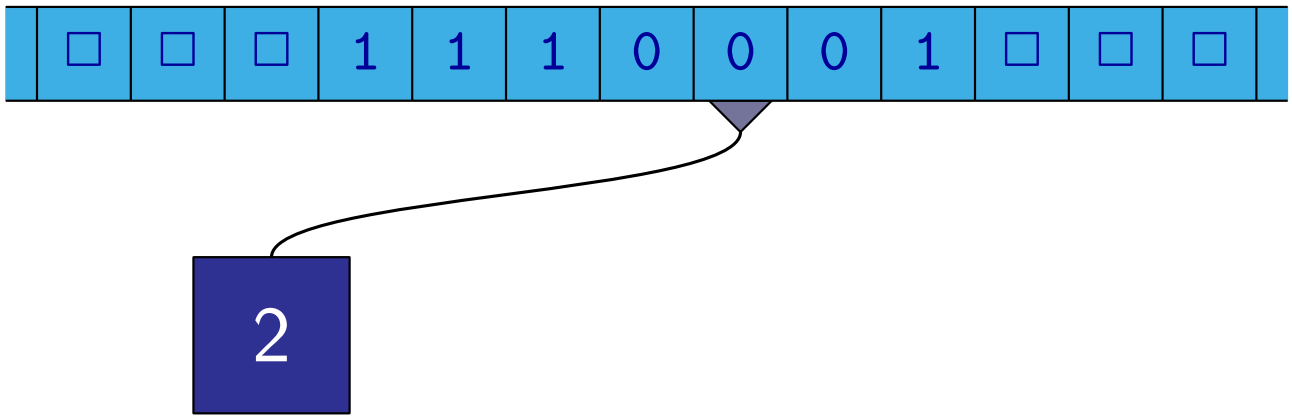
Turingův stroj – násobení třemi



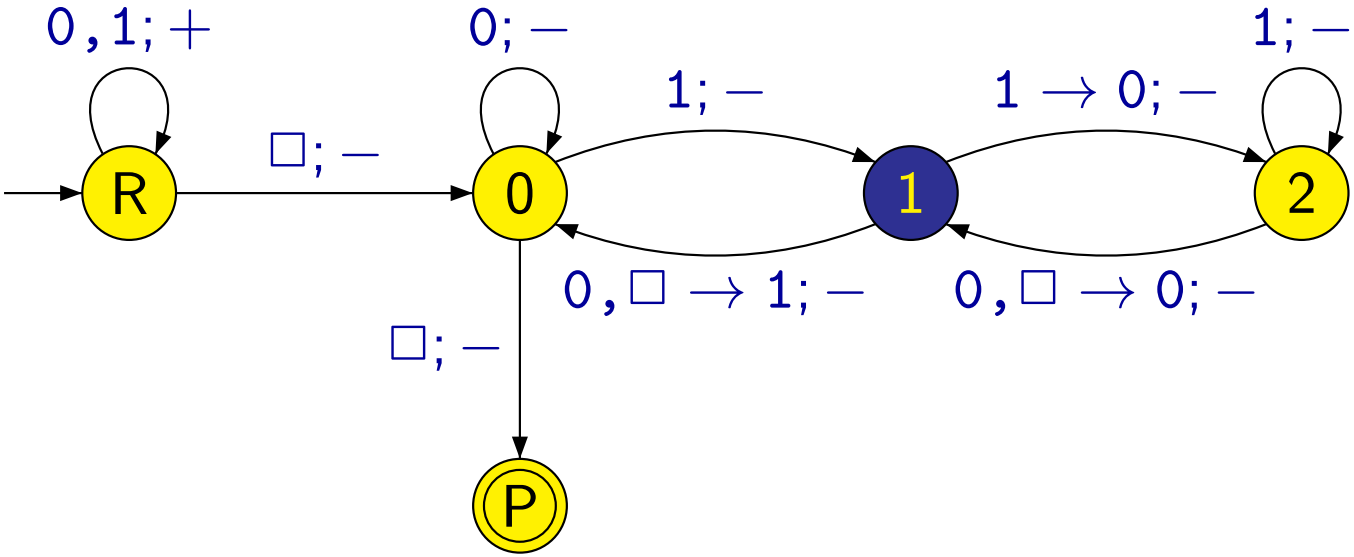


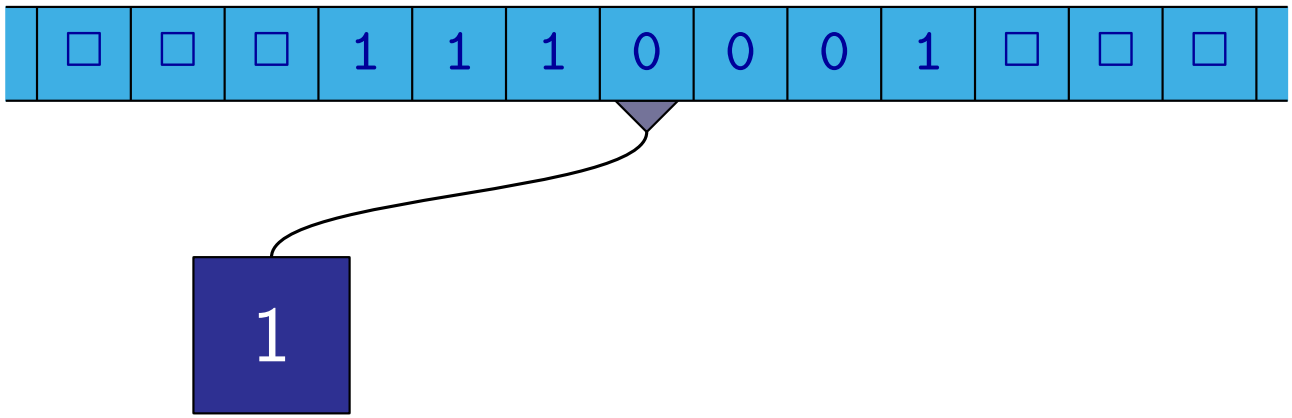
Turingův stroj – násobení třemi



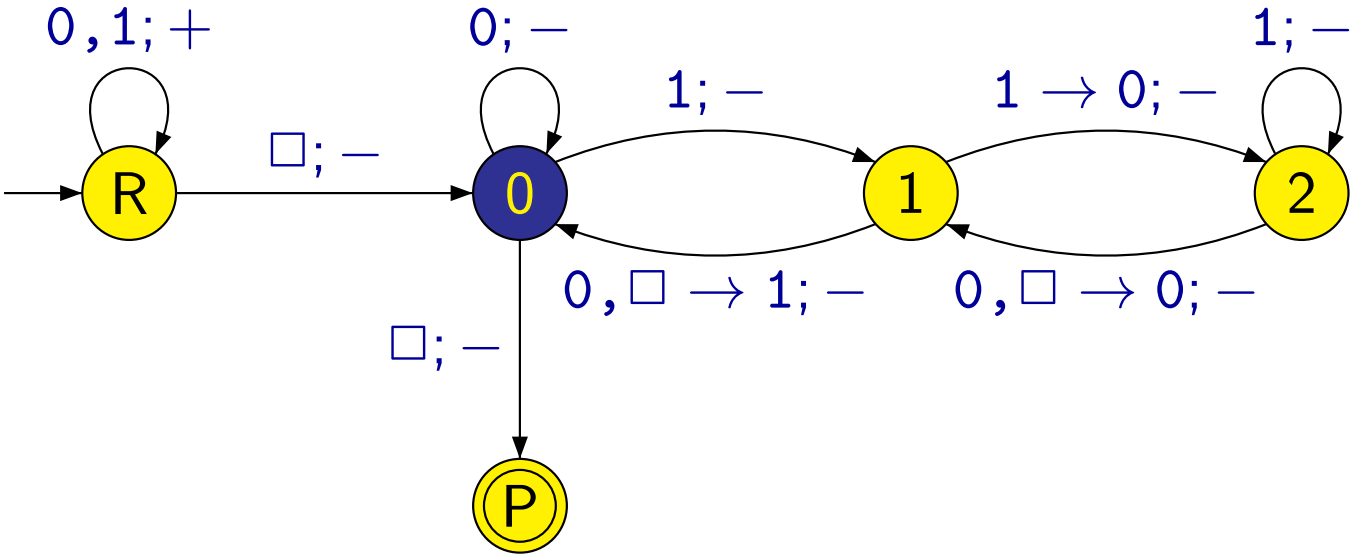


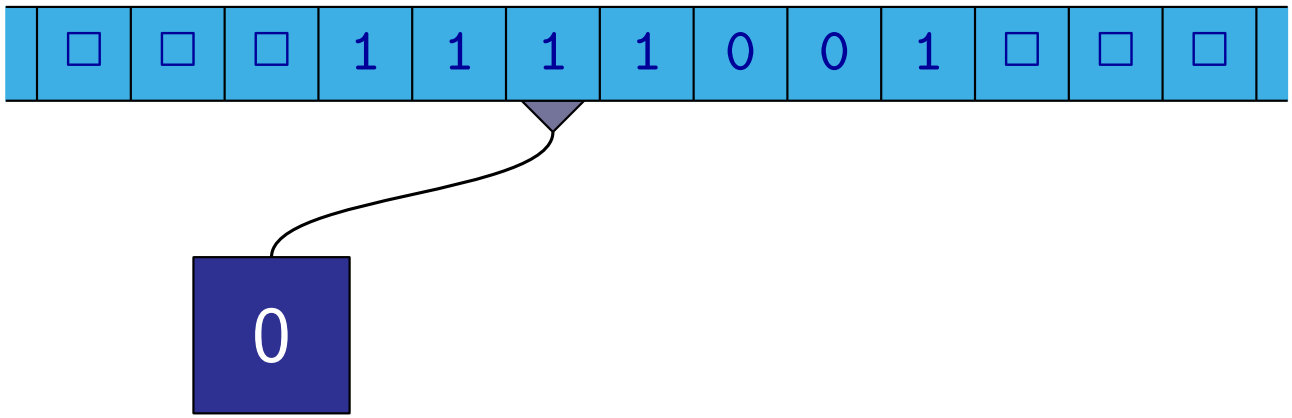
Turingův stroj – násobení třemi



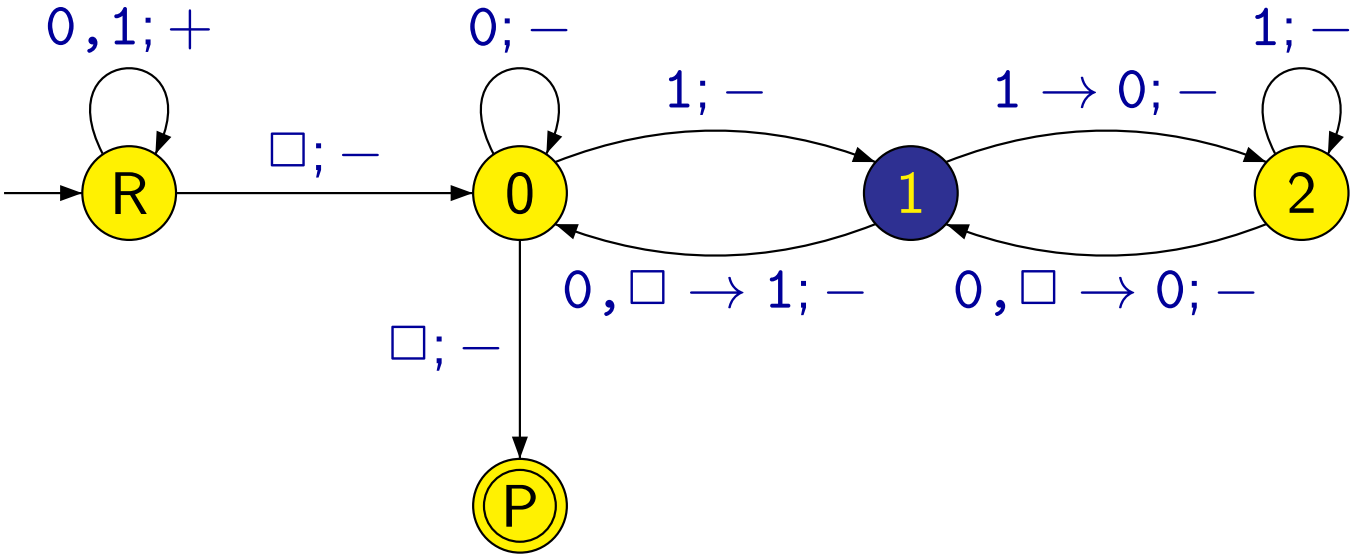


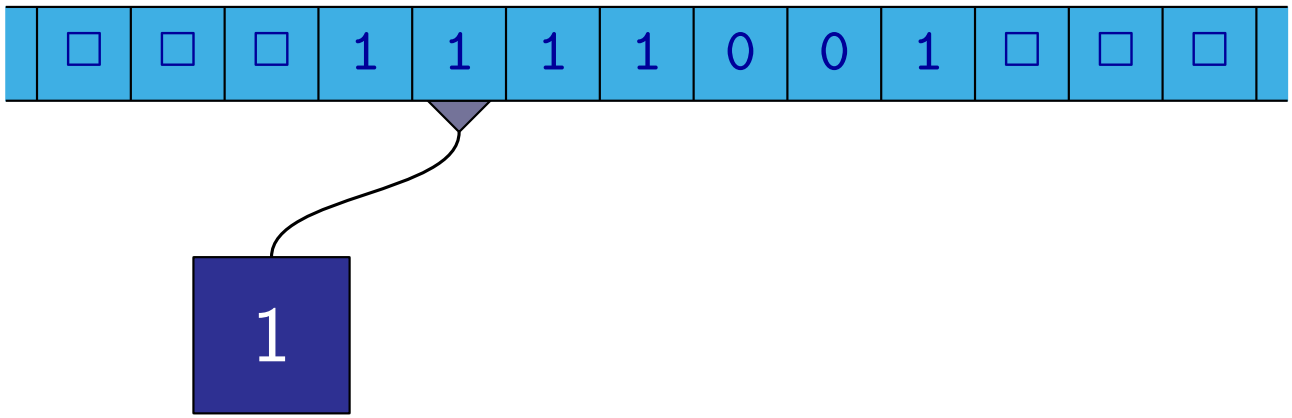
Turingův stroj – násobení třemi



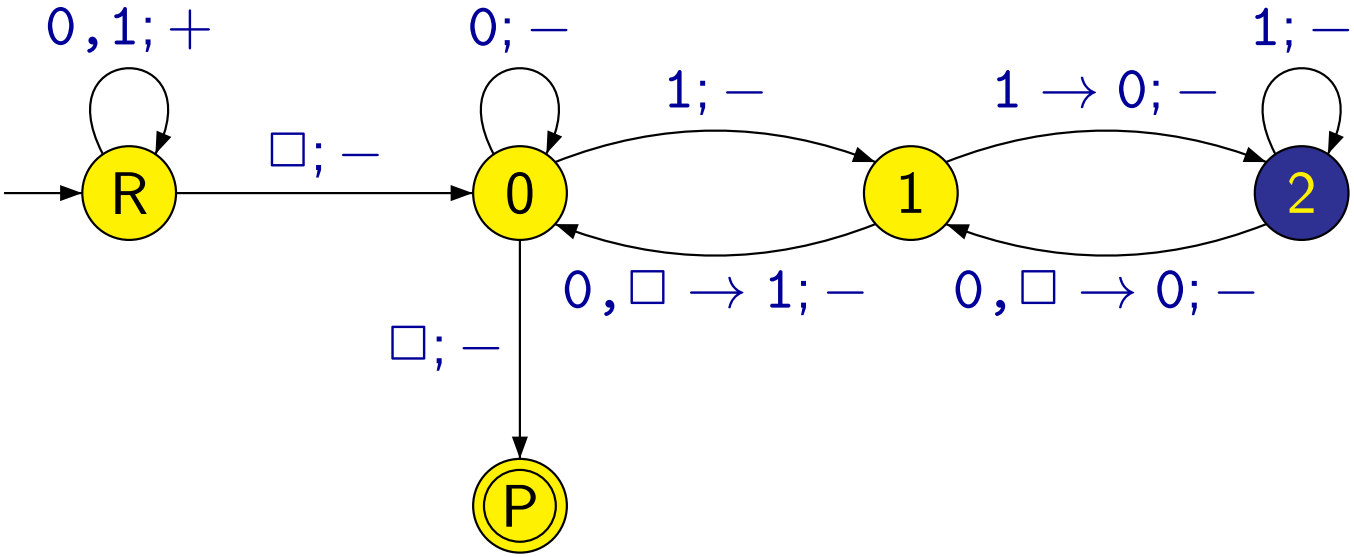


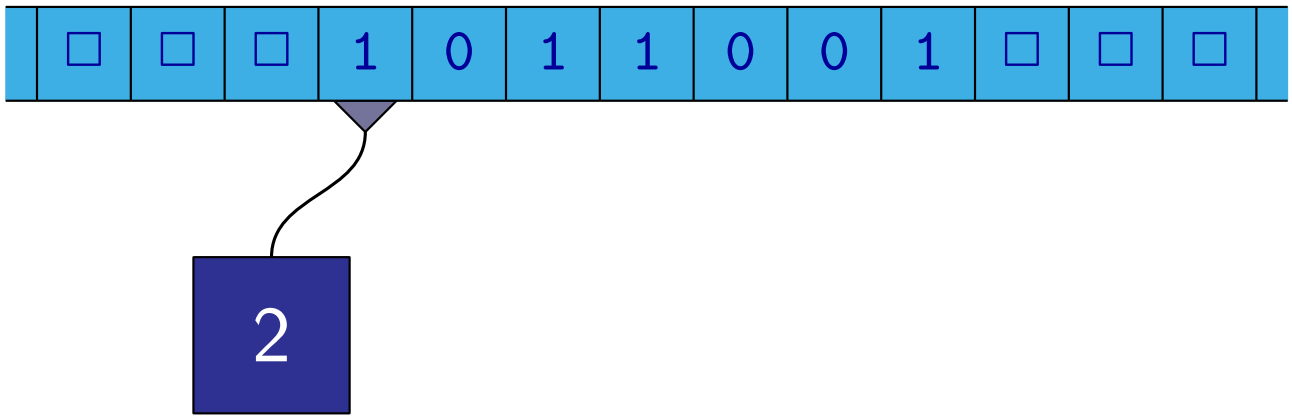
Turingův stroj – násobení třemi



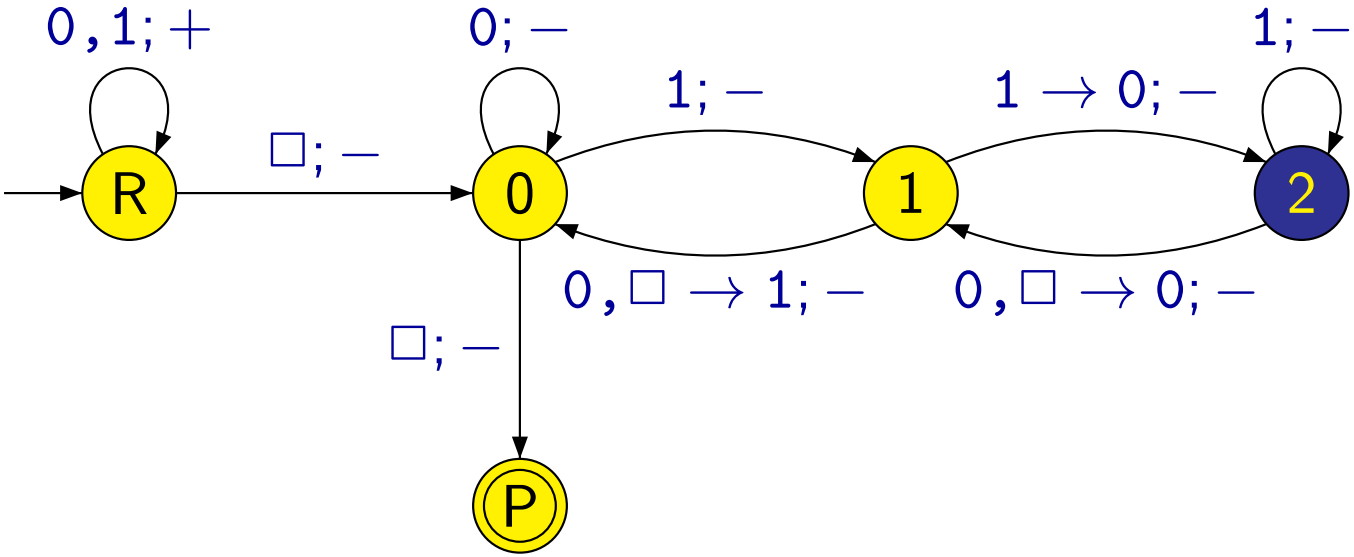


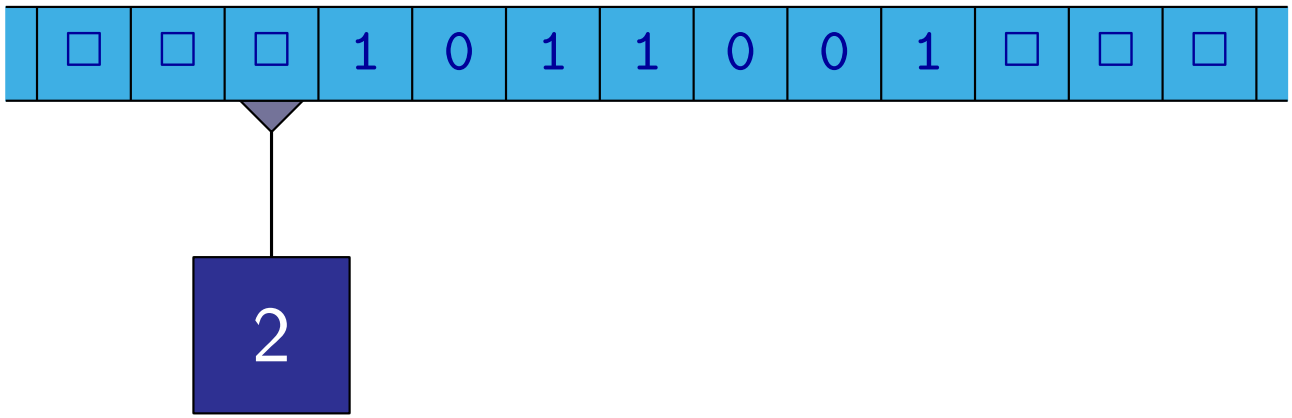
Turingův stroj – násobení třemi



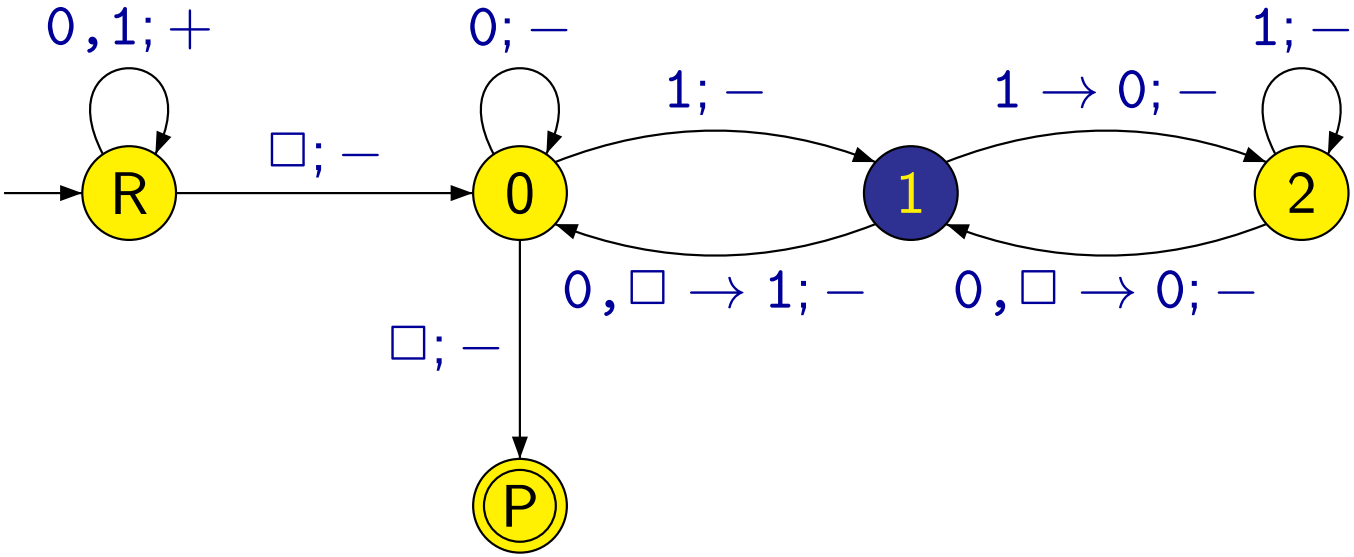


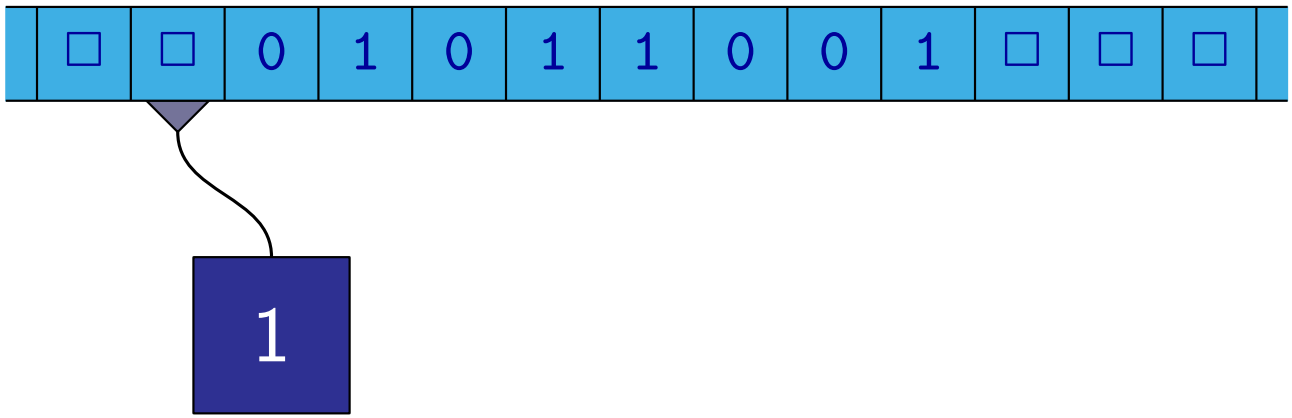
Turingův stroj – násobení třemi



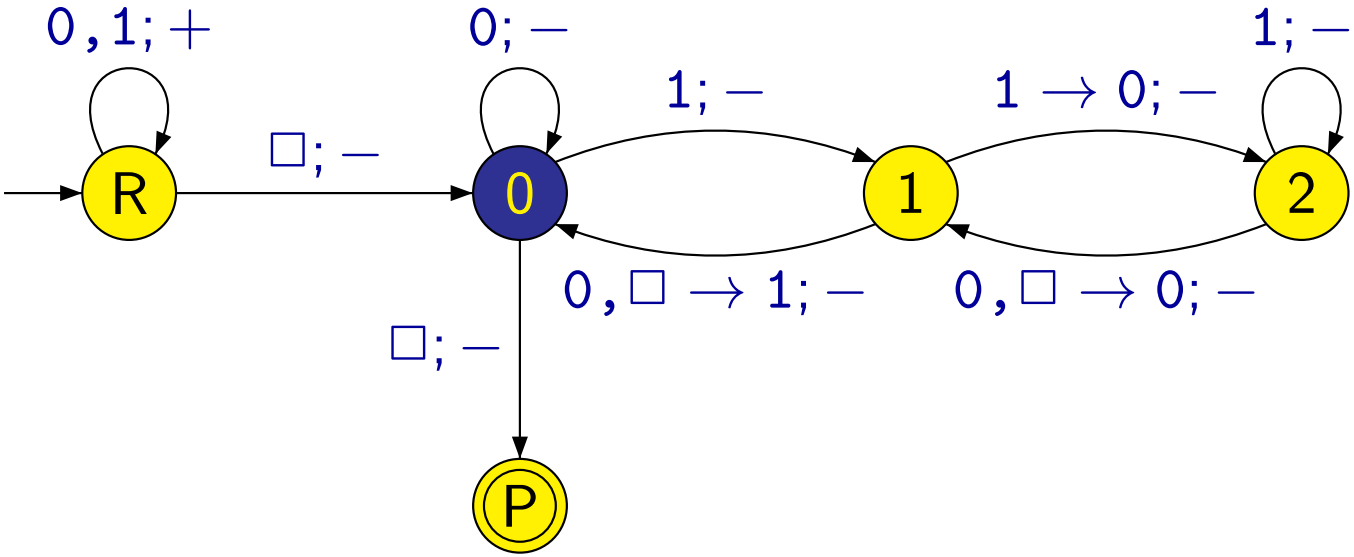


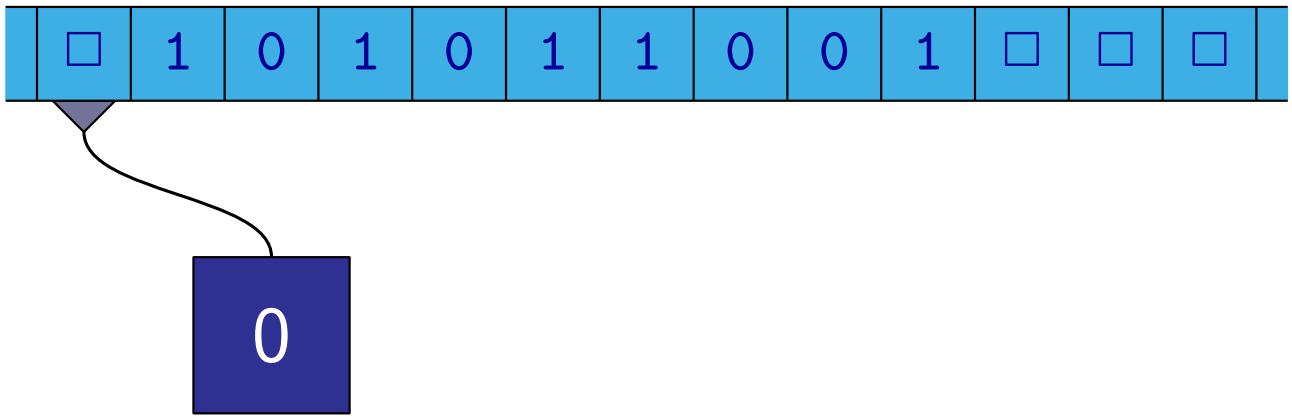
Turingův stroj – násobení třemi



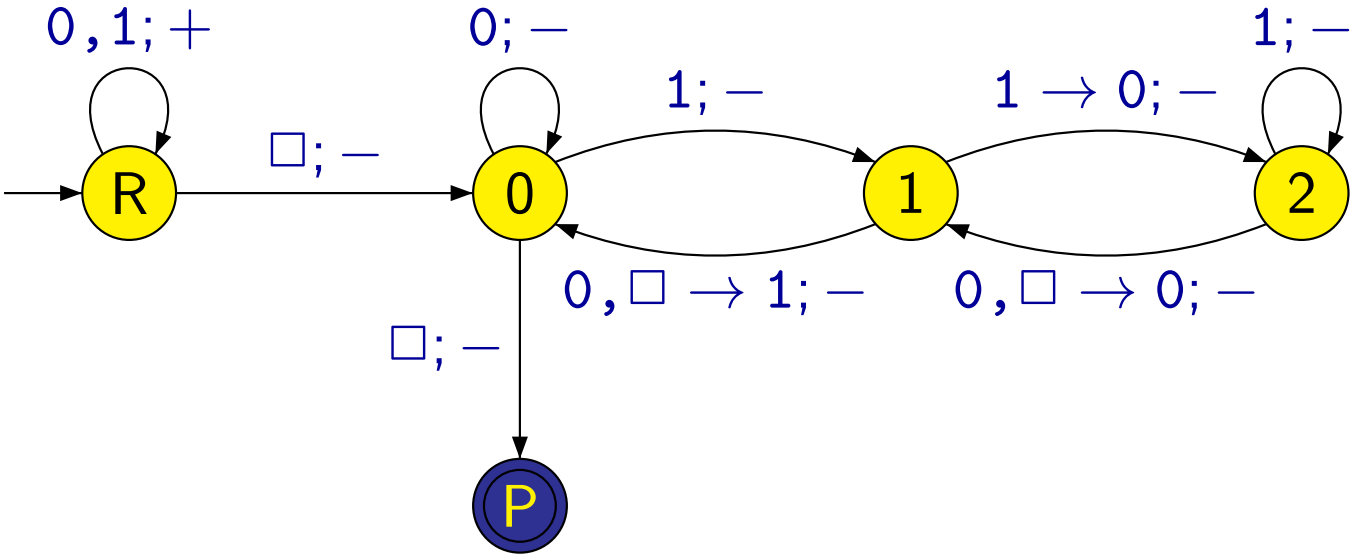


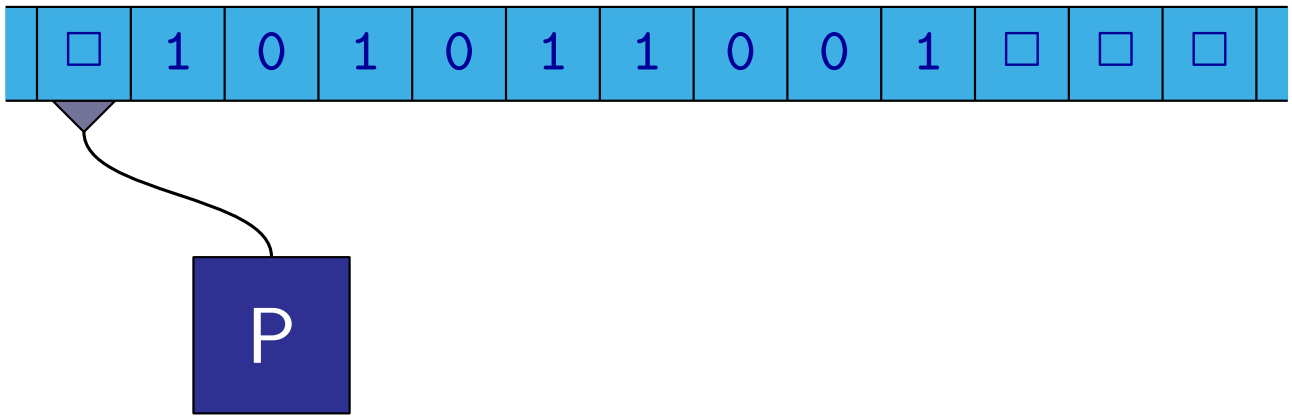
Turingův stroj – násobení třemi





Turingův stroj – násobení třemi





Nedeterministické Turingovy stroje

Můžeme uvažovat i **nedeterministické Turingovy stroje**, kde pro každý stav q a symbol b přechodová funkce $\delta(q, b)$ určuje více různých trojic (q', b', d).

Stroj si může vybrat libovolnou z nich.

Stroj přijímá slovo w, jestliže existuje alespoň jeden jeho výpočet vedoucí k přijetí slova w.

Poznámka: Ke každému nedeterministickému Turingovu stroji je možné sestrojit ekvivalentní deterministický Turingův stroj.

Churchova-Turingova teze

Churchova-Turingova teze

Každý algoritmus je možné realizovat nějakým Turingovým strojem.

Není to věta, kterou by bylo možno dokázat v matematickém smyslu – není formálně definováno, co je to algoritmus.

Tezi formulovali nezávisle na sobě v polovině 30. let 20. století Alan Turing a Alonzo Church.

Churchova-Turingova teze

Příklady matematických formalismů zachycujících pojem algoritmus:

- stroje RAM
- Turingovy stroje
- lambda kalkulus
- rekurzivní funkce
- ...

Dále můžeme uvést:

- Libovolný (obecný) programovací jazyk (jako např. C, Java, Lisp, Haskell, Prolog apod.).

Všechny tyto modely jsou ekvivalentní z hlediska algoritmů, které jsou schopny realizovat.

Složitost algoritmů

Složitost algoritmu

- Počítače pracují rychle, ale ne nekonečně rychle. Provedení každé instrukce trvá nějakou (i když velmi krátkou) dobu.
- Stejný problém může řešit více různých algoritmů a doba výpočtu (daná hlavně počtem provedených instrukcí) může být pro různé algoritmy různá.
- Algoritmy bychom chtěli mezi sebou porovnávat a zvolit si ten lepší.
- Algoritmy můžeme naprogramovat a změřit čas výpočtu. Tím zjistíme jak dlouho trvá výpočet na konkrétních datech, na kterých algoritmus testujeme.
- Chtěli bychom mít i nějakou přesnější představu o tom, jak dlouho bude trvat výpočet na všech možných vstupních datech.

Časová složitost

Problém „Vyhledávání“

Vstup: Celé číslo x a sekvence celých čísel $a_1, a_2, \ldots, a_n$ (kde $a_i \neq 0$) ukončená 0.

Výstup: Pokud $a_i = x$, je výstupem i (pokud jich je takových i více, tak nejmenší z nich), jinak je výstupem 0.

```
start:   READ
         STORE   3
         LOAD    =1
cyklus:  STORE   2
         READ
         JZERO   vypis
         SUB     3
         JZERO   nasel
         LOAD    2
         ADD     =1
         JUMP    cyklus
nasel:   LOAD    2
vypis:   WRITE
         HALT
```

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 0
Buňka 3: 0
Buňka 4: 0
⋮

Výstup:

Instrukcí: 0

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 0
Buňka 3: 0
Buňka 4: 0
⋮

Výstup:

Instrukcí: 0

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 0
Buňka 3: 0
Buňka 4: 0
⋮

Výstup:

Instrukcí: 1

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 0
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 2

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 1
Buňka 1: 0
Buňka 2: 0
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 3

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 1
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 4

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 13
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 5

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 13
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 6

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 4
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 7

Časová složitost

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 4
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 8

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 1
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 9

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 10

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 1
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 11

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 12

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 5
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 13

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 5
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 14

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: -4
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 15

Časová složitost

```
start:   READ
         STORE   3
         LOAD    =1
cyklus:  STORE   2
         READ
         JZERO   vypis
         SUB     3
         JZERO   nasel
         LOAD    2
         ADD     =1
         JUMP    cyklus
nasel:   LOAD    2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: -4
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 16

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 2
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 17

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 18

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 2
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 19

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 20

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 21

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 9
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 22

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 23

Časová složitost

```
 start:   READ
          STORE   3
          LOAD    =1
cyklus:   STORE   2
          READ
          JZERO   vypis
          SUB     3
          JZERO   nasel
          LOAD    2
          ADD     =1
          JUMP    cyklus
 nasel:   LOAD    2
 vypis:   WRITE
          HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 0
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 24

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup:

Instrukcí: 25

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
⋮

Výstup: 3

Instrukcí: 26

Časová složitost

```
start:   READ
         STORE  3
         LOAD   =1
cyklus:  STORE  2
         READ
         JZERO  vypis
         SUB    3
         JZERO  nasel
         LOAD   2
         ADD    =1
         JUMP   cyklus
nasel:   LOAD   2
vypis:   WRITE
         HALT
```

Vstup:
$9, 13, 5, 9, 7, 2, 0$

Buňka 0: 3
Buňka 1: 0
Buňka 2: 3
Buňka 3: 9
Buňka 4: 0
$\vdots$

Výstup: 3

Instrukcí: 27

Časová složitost

- V uvedeném příkladě, kdy vstup byl

$$9, 13, 5, 9, 7, 2, 0$$

bylo provedeno 27 instrukcí.

Časová složitost

- V uvedeném příkladě, kdy vstup byl

  $$9, 13, 5, 9, 7, 2, 0$$

  bylo provedeno 27 instrukcí.

Rozeberme nyní, kolik instrukcí se provede obecně pro vstup

$$x, a_1, a_2, \ldots, a_n, 0$$

Časová složitost

- V uvedeném příkladě, kdy vstup byl

$$9, 13, 5, 9, 7, 2, 0$$

bylo provedeno 27 instrukcí.

Rozeberme nyní, kolik instrukcí se provede obecně pro vstup

$$x, a_1, a_2, \ldots, a_n, 0$$

- Pokud pro všechna i (kde $1 \leq i \leq n$) platí $x \neq a_i$:

$$3 + 8 \cdot n + 3 + 2 = 8n + 8$$

Časová složitost

- V uvedeném příkladě, kdy vstup byl

$$9, 13, 5, 9, 7, 2, 0$$

bylo provedeno 27 instrukcí.

Rozeberme nyní, kolik instrukcí se provede obecně pro vstup

$$x, a_1, a_2, \ldots, a_n, 0$$

- Pokud pro všechna i (kde $1 \leq i \leq n$) platí $x \neq a_i$:

$$3 + 8 \cdot n + 3 + 2 = 8n + 8$$

- Pokud pro nějaké i (kde $1 \leq i \leq n$) platí $x = a_i$:

$$3 + 8 \cdot (i - 1) + 5 + 3 = 8i + 3$$

Velikost vstupu

Pro různé vstupy provede program různý počet instrukcí.

Pokud chceme počet provedených instrukcí nějak analyzovat, je vhodné si zavést pojem **velikost vstupu**.

Typicky je velikost vstupu číslo, které udává, jak je daná instance „velká“ (čím větší číslo, tím větší instance).

Příklad: Pro problém „Vyhledávání“, kde jsou vstupy tvaru

$$x, a_1, a_2, \ldots, a_n, 0$$

můžeme jako velikost vstupu zvolit například hodnotu n.

Velikost vstupu $9, 13, 5, 9, 7, 2, 0$ je tedy potom 5.

Velikost vstupu

Poznámka: Velikost vstupu si v daném konkrétním případě můžeme definovat, jak chceme a jak je to pro další analýzu výhodné.

Co přesně zvolíme jako velikost vstupu není předem dáno, ale z podstaty zadaného problému většinou nějak přirozeně vyplývá, co za velikost vstupu zvolit.

Příklady:

- Pro problém „Třídění“, kde vstupem je sekvence čísel $a_1, a_2, \ldots, a_n$ a výstupem jsou tato čísla setříděná, můžeme vzít jako velikost vstupu hodnotu n.
- Pro problém „Prvočíselnost“, kde vstupem je přirozené číslo x, a kde se ptáme, zda x je prvočíslo, můžeme vzít jako velikost vstupu počet bitů čísla x.

  (Jinou možností by bylo vzít jako velikost vstupu přímo hodnotu x.)

Velikost vstupu

Někdy je vhodné popsat velikost vstupu pomocí více čísel.

Například u problémů, kde vstupem je graf, můžeme definovat velikost vstupu jako dvojici čísel n, m, kde:

- n – počet vrcholů grafu
- m – počet hran grafu

Poznámka: Jinou možností by bylo definovat velikost vstupu jako jediné číslo $n + m$.

Velikost vstupu

Obecně můžeme pro libovolný problém definovat velikost vstupu následovně:

- Pokud je vstupem slovo w z nějaké obecedy Σ:
  délka slova w
- Pokud je vstupem sekvence bitů (tj. slovo z abecedy $\{0,1\}$):
  počet bitů v této sekvenci
- Pokud je vstupem přirozené číslo x:
  počet bitů nutných k zápisu čísla x

Časová složitost

Chceme analyzovat konkrétní algoritmus (jeho konkrétní implementaci).

Zajímá nás, kolik instrukcí se provede, pokud algoritmus dostane vstup velikosti $1, 2, 3, 4, \ldots$.

Je zřejmé, že i pro vstupy, které mají stejnou velikost, může být počet provedených instrukcí různý.

Předpokládejme, že X je množina všech možných vstupů daného problému a $g(x)$ je počet instrukcí provedených algoritmem pro vstup $x \in X$.

Označme si velikost vstupu $x \in X$ jako $|x|$.

Nyní definujme následující funkci $f : \mathbb{N} \to \mathbb{N}$ takovou, že pro $n \in \mathbb{N}$ je

$$f(n) = \max\{g(x) \mid x \in X, |x| = n\}$$

Časová složitost v nejhorším případě

Takto definované funkci $f(n)$ (tj. funkci, která pro daný algoritmus a danou definici velikosti vstupu přiřazuje každému přirozenému číslu n maximální počet instrukcí, které algoritmus provede, pokud dostane vstup velikosti n) se říká **časová složitost algorimu v nejhorším případě**.

Časová složitost v nejhorším případě

Takto definované funkci $f(n)$ (tj. funkci, která pro daný algoritmus a danou definici velikosti vstupu přiřazuje každému přirozenému číslu n maximální počet instrukcí, které algoritmus provede, pokud dostane vstup velikosti n) se říká **časová složitost algorimu v nejhorším případě**.

Příklad: Jak jsme zjistili, dříve popsaný algorimus řešící problém „Vyhledávání“ provede pro vstup $x, a_1, a_2, \ldots, a_n, 0$ následující počty instrukcí:

- Pokud pro všechna i platí $x \neq a_i$, algorimus provede $8n + 8$ instrukcí.
- Pokud pro nějaké i platí $x = a_i$, algorimus provede $8i + 3$ instrukcí.

Vzhledem k tomu, že ve druhém případě je vždy $i \leq n$, je časová složitost tohoto algorimu v nejhorším případě $f(n) = 8n + 8$.

Časová složitost v průměrném případě

Kromě časové složitosti v nejhorším případě má smysl zkoumat i časovou složitost **v průměrném případě**.

V tomto případě $f(n)$ nedefinujeme jako maximum, ale jako aritmetický průměr z hodnot

$$\{g(x) \mid x \in X, |x| = n\}$$

- Určit časovou složitost v průměrném případě je většinou těžší než určit časovou složitost v nejhorším případě.
- Často se tyto dvě funkce příliš neliší, někdy je ale rozdíl významný.

Poznámka: Zkoumat složitost v nejlepším případě většinou moc smysl nemá.

Rychlost růstu funkcí

Program zpracovává vstup velikosti n.
Předpokládejme, že pro vstup velikosti n provede $f(n)$ operací, a že provedení jedné operace trvá $1\,\mu\mathrm{s}$ ($10^{-6}\,\mathrm{s}$).

	n							
$f(n)$	20	40	60	80	100	200	500	1000
n	$20\,\mu$s	$40\,\mu$s	$60\,\mu$s	$80\,\mu$s	0.1 ms	0.2 ms	0.5 ms	1 ms
$n \log n$	$86\,\mu$s	0.213 ms	0.354 ms	0.506 ms	0.664 ms	1.528 ms	4.48 ms	9.96 ms
n^2	0.4 ms	1.6 ms	3.6 ms	6.4 ms	10 ms	40 ms	0.25 s	1 s
n^3	8 ms	64 ms	0.216 s	0.512 s	1 s	8 s	125 s	16.7 min.
n^4	0.16 s	2.56 s	12.96 s	42 s	100 s	26.6 min.	17.36 hod.	11.57 dní
2^n	1.05 s	12.75 dní	36560 let	$38.3 \cdot 10^9$ let	$40.1 \cdot 10^{15}$ let	$50 \cdot 10^{45}$ let	$10.4 \cdot 10^{136}$ let	–
$n!$	77147 let	$2.59 \cdot 10^{34}$ let	$2.64 \cdot 10^{68}$ let	$2.27 \cdot 10^{105}$ let	$2.96 \cdot 10^{144}$ let	–	–	–

Rychlost růstu funkcí

Uvažujme 3 algoritmy se složitostmi $t_1(n) = n, t_2(n) = n^3, t_3(n) = 2^n$. Náš počítač zvládne v reálném čase (kolik jsme ochotni počkat) 10^{12} kroků.

Složitost	Velikost vstupu
$t_1(n) = n$	10^{12}
$t_2(n) = n^3$	10^4
$t_3(n) = 2^n$	40

Rychlost růstu funkcí

Uvažujme 3 algoritmy se složitostmi $t_1(n) = n, t_2(n) = n^3, t_3(n) = 2^n$. Náš počítač zvládne v reálném čase (kolik jsme ochotni počkat) 10^{12} kroků.

Složitost	Velikost vstupu
$t_1(n) = n$	10^{12}
$t_2(n) = n^3$	10^4
$t_3(n) = 2^n$	40

Nyní počítač 1000 násobně zrychlíme. Zvládne tedy 10^{15} kroků.

Složitost	Velikost vstupu	Nárust
$t_1(n) = n$	10^{15}	$1000\times$
$t_2(n) = n^3$	10^5	$10\times$
$t_3(n) = 2^n$	50	$+10$

Asymptotická notace

- Přesnou složitost bývá problém vyjádřit.
- Přesná složitost je silně závislá na konkrétním zvoleném modelu a konkrétní implementaci (na detailech této implementace).
- Složitost nás většinou zajímá hlavně pro velké vstupy. Pro malé vstupy obvykle i neefektivní algoritmus proběhne rychle.
- Ve většině případů nepotřebujeme znát přesný počet provedených instrukcí, ale spokojíme se s odhadem toho, jak rychle tento počet narůstá se zvětšováním velikosti vstupu.
- Proto zavádíme tzv. **asymptotickou notaci**, která nám umožní zanedbat méně důležité detaily a odhadnout přibližně, jak rychle daná funkce roste, a která analýzu podstatným způsobem zjednodušuje.

Asymptotická notace

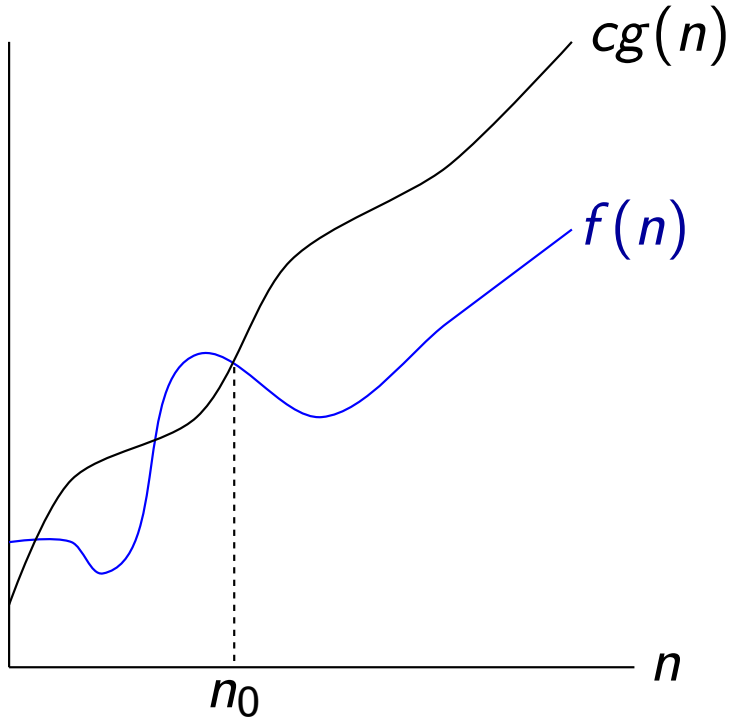
Vezměme si libovolnou funkci $g : \mathbb{N} \to \mathbb{N}$. Zápisy $O(g)$, $\Omega(g)$ a $\Theta(g)$ označují **množiny funkcí** typu $\mathbb{N} \to \mathbb{N}$, kde:

- $O(g)$ – množina všech funkcí, které rostou nejvýše tak rychle jako g
- $\Omega(g)$ – množina všech funkcí, které rostou alespoň tak rychle jako g
- $\Theta(g)$ – množina všech funkcí, které rostou stejně rychle jako g

Poznámka: Toto nejsou definice! Ty následují na následujících slidech.

- O – velké „O“
- Ω – velké řecké písmeno „omega“
- Θ – velké řecké písmeno „theta“

Asymptotická notace – symbol O

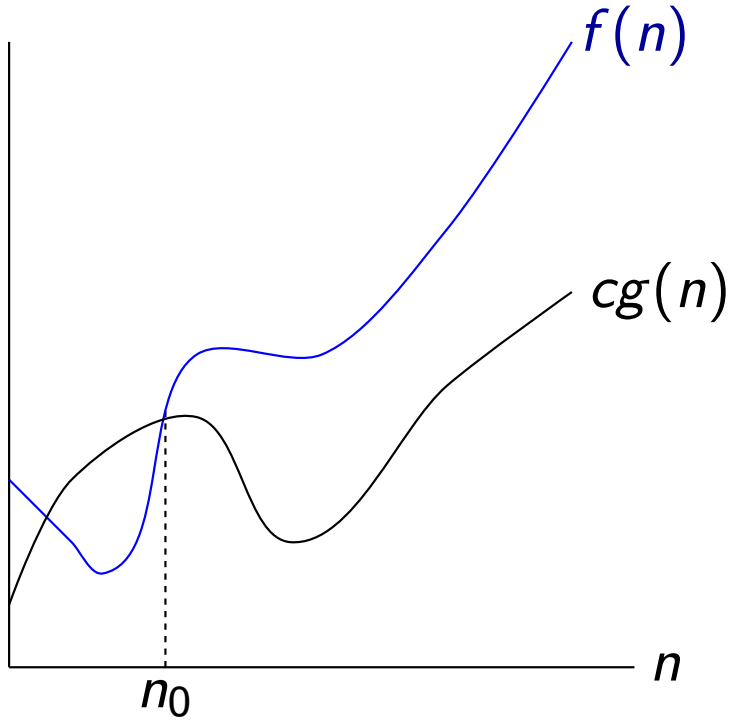


Definice

Vezměme si libovolnou funkci $g : \mathbb{N} \to \mathbb{N}$. Pro funkci $f : \mathbb{N} \to \mathbb{N}$ platí $f \in O(g)$ právě tehdy, když

$$(\exists c > 0)(\exists n_0 \geq 0)(\forall n \geq n_0) : f(n) \leq c\, g(n).$$

Asymptotická notace – symbol Ω

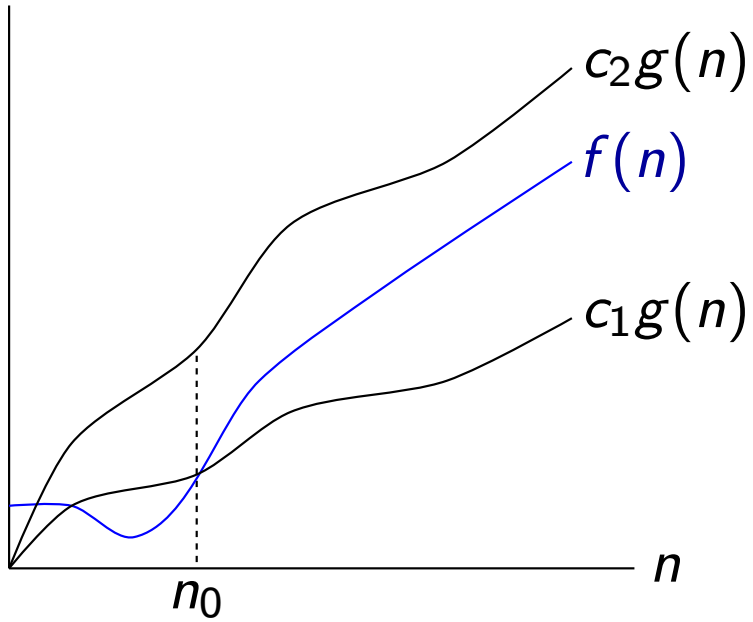


Definice

Vezměme si libovolnou funkci $g : \mathbb{N} \to \mathbb{N}$. Pro funkci $f : \mathbb{N} \to \mathbb{N}$ platí $f \in \Omega(g)$ právě tehdy, když

$$(\exists c > 0)(\exists n_0 \geq 0)(\forall n \geq n_0) : c\, g(n) \leq f(n).$$

Asymptotická notace – symbol Θ



Definice

Vezměme si libovolnou funkci $g : \mathbb{N} \to \mathbb{N}$. Pro funkci $f : \mathbb{N} \to \mathbb{N}$ platí $f \in \Theta(g)$ právě tehdy, když

$$(\exists c_1 > 0)(\exists c_2 > 0)(\exists n_0 \geq 0)(\forall n \geq n_0) : c_1\, g(n) \leq f(n) \leq c_2\, g(n).$$

Asymptotická notace

Pro jednoduchost uvažujeme v předchozích definicích pouze funkce typu $\mathbb{N} \to \mathbb{N}$.

Ve skutečnosti by se tyto definice daly rozšířit na všechny **asymptoticky nezáporné** funkce typu $\mathbb{R}_+ \to \mathbb{R}$, které navíc mohou být na nějakém konečné podintervalu svého definičního nedefinované.

Funkce $f : \mathbb{R}_+ \to \mathbb{R}$ je **asymptoticky nezáporná** pokud pro ni platí:

$$(\exists n_0 \geq 0)(\forall n \geq n_0)(f(n) \geq 0)$$

Poznámka: Pro $n < n_0$, může být hodnota $f(n)$ nedefinovaná.

$\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$

Asymptotická notace

Příklady:

$n \in O(n^2)$
$1000n \in O(n)$
$2^{\log_2 n} \in \Theta(n)$
$n^3 \notin O(n^2)$
$n^2 \notin O(n)$
$n^3 + 2^n \notin O(n^2)$

$n^3 \in O(n^4)$
$0.00001n^2 - 10^{10}n \in \Theta(10^{10}n^2)$
$n^3 - n^2 \log_2^3 n + 1000n - 10^{100} \in \Theta(n^3)$
$n^3 + 1000n - 10^{100} \in O(n^3)$
$n^3 + n^2 \notin \Theta(n^2)$
$n! \notin O(2^n)$

Asymptotická notace

- Pro libovolnou funkci $g : \mathbb{N} \to \mathbb{N}$ platí:

$$g \in O(g) \qquad g \in \Omega(g) \qquad g \in \Theta(g)$$

- Pro libovolné dvě funkce $f, g : \mathbb{N} \to \mathbb{N}$ platí:
  - $f \in O(g)$ právě tehdy, když $g \in \Omega(f)$
  - $f \in \Theta(g)$ právě tehdy, když $g \in \Theta(f)$
  - $f \in \Theta(g)$ právě tehdy, když $f \in O(g)$ a $f \in \Omega(g)$

- Pro libovolné tři funkce $f, g, h : \mathbb{N} \to \mathbb{N}$ platí:
  - jestliže $f \in O(g)$ a $g \in O(h)$, pak $f \in O(h)$
  - jestliže $f \in \Omega(g)$ a $g \in \Omega(h)$, pak $f \in \Omega(h)$
  - jestliže $f \in \Theta(g)$ a $g \in \Theta(h)$, pak $f \in \Theta(h)$

Asymptotická notace

- Existují dvojice funkcí $f, g : \mathbb{N} \to \mathbb{N}$ takové, že

$$f \notin O(g) \qquad \text{a} \qquad g \notin O(f),$$

  například

$$f(n) = n \qquad\qquad g(n) = n^{1+\sin(n)}.$$

- $O(1)$ označuje množinu všech **omezených** funkcí, tj. funkcí jejichž funkční hodnoty jsou shora omezeny nějakou konstantou.

Asymptotická notace

- Pro libovolné dvě funkce $f, g : \mathbb{N} \to \mathbb{N}$ platí:
  - $\max(f, g) \in \Theta(f + g)$
  - pokud $f \in O(g)$, pak $f + g \in \Theta(g)$

- Pro libovolné čtyři funkce $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{N}$ platí:
  - pokud $f_1 \in O(f_2)$ a $g_1 \in O(g_2)$, pak $f_1 + g_1 \in O(f_2 + g_2)$ a $f_1 \cdot g_1 \in O(f_2 \cdot g_2)$
  - pokud $f_1 \in \Theta(f_2)$ a $g_1 \in \Theta(g_2)$, pak $f_1 + g_1 \in \Theta(f_2 + g_2)$ a $f_1 \cdot g_1 \in \Theta(f_2 \cdot g_2)$

Asymptotická notace

- O funkci f řekneme, že je:

  logaritmická, pokud $f(n) \in \Theta(\log n)$
  lineární, pokud $f(n) \in \Theta(n)$
  kvadratická, pokud $f(n) \in \Theta(n^2)$
  kubická, pokud $f(n) \in \Theta(n^3)$
  polynomiální, pokud $f(n) \in O(n^k)$ pro nějaké $k > 0$
  exponenciální, pokud $f(n) \in O(c^{n^k})$ pro nějaké $c > 1$ a $k > 0$

- Exponenciální funkce se v asymptotické notaci často uvádí ve tvaru $2^{O(n^k)}$, protože potom již nemusíme uvažovat různé základy mocniny.

Asymptotická notace

- Pro libovolná $k, \ell > 0$ taková, že $k < \ell$, a libovolné $c > 1$ platí:

$$n^k \in O(n^\ell) \qquad n^\ell \notin O(n^k)$$
$$\log_c^k n \in O(\log_c^\ell n) \qquad \log_c^\ell n \notin O(\log_c^k n)$$
$$c^{n^k} \in O(c^{n^\ell}) \qquad c^{n^\ell} \notin O(c^{n^k})$$

- Pro libovolná $k, \ell > 0$ a libovolné $c > 1$ platí:

$$\log_c^k n \in O(n^\ell) \qquad n^\ell \notin O(\log_c^k n)$$
$$n^k \in O(c^{n^\ell}) \qquad c^{n^\ell} \notin O(n^k)$$

Poznámka: $\log_c^k n$ je stručnější zápis pro $(\log_c n)^k$.

Asymptotická notace

Tvrzení

Pro libovolná $a, b > 1$ a libovolné $n > 0$ platí

$$\log_a n = \frac{\log_b n}{\log_b a}$$

Důkaz: Z $n = a^{\log_a n}$ plyne $\log_b n = \log_b(a^{\log_a n})$.
Protože $\log_b(a^{\log_a n}) = \log_a n \cdot \log_b a$, dostáváme $\log_b n = \log_a n \cdot \log_b n$,
z čehož plyne výše uvedený závěr. □

Pro libovolné konstanty $a, b > 1$ tedy platí $\log_a n \in \Theta(\log_b n)$.

Z toho důvodu se při použití asymptotické notace základ logaritmu obvykle vynechává: například místo $\Theta(n \log_2 n)$ můžeme napsat $\Theta(n \log n)$.

Asymptotická notace

Jak bylo uvedeno, výrazy $O(g)$, $\Omega(g)$ a $\Theta(g)$ označují určité množiny funkcí.

V odborných textech se však někdy používají tyto výrazy i v poněkud odlišném významu:

- zápis $O(g)$, $\Omega(g)$ nebo $\Theta(g)$ nereprezentuje danou množinu funkcí, ale **nějakou** funkci z dané množiny.

Tato konvence se používá zejména v zápisu rovnic nebo nerovnic.

Příklad: $3n^3 + 5n^2 - 11n + 2 = 3n^3 + O(n^2)$

Při použití této konvence je tedy možné například psát $f = O(g)$ místo $f \in O(g)$.

Asymptotická notace

- Asymptotická notace se dá přímočaře zobecnit pro funkce více argumentů:

  Uvažujme například funkci $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Pro funkci $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ platí $f(n, m) \in O(g(n, m))$ právě tehdy, když

  $$(\exists c > 0)(\exists n_0 \geq 0)(\exists m_0 \geq 0)(\forall n \geq n_0)(\forall m \geq m_0)(f(n, m) \leq c\, g(n, m))$$

- Kromě výrazů $O(g)$, $\Omega(g)$ a $\Theta(g)$ se používají ještě výrazy $o(g)$ a $\omega(g)$.
  - $o(g)$ – množina všech funkcí, které rostou pomaleji než funkce g
  - $\omega(g)$ – množina všech funkcí, které rostou rychleji než funkce g

  Jejich přesné definice zde nebudeme uvádět a nebudeme se jimi dále zabývat.

Složitost algoritmů

Řekněme, že bychom chtěli analyzovat časovou složitost $t(n)$ nějakého algoritmu, který se skládá z instrukcí $I_1, I_2, \ldots, I_k$:

- Pokud $m_1, m_2, \ldots, m_k$ jsou počty provedení jednotlivých instrukcí pro nějaký vstup x (tj. pro vstup x se instrukce I_i provede m_i krát), tak celkový počet instrukcí provedených pro vstup x je
$$m_1 + m_2 + \cdots + m_k.$$
- Vezměme si funkce $f_1, f_2, \ldots, f_k$, kde $f_i : \mathbb{N} \to \mathbb{N}$, přičemž $f_i(n)$ je maximum z počtu provedení instrukce I_i pro všechny vstupy velikosti n.
- Zjevně platí, že pro libovolnou funkci f_i je $t \in \Omega(f_i)$.
- Zjevně také platí $t \in O(f_1 + f_2 + \cdots + f_k)$.

Složitost algoritmů

- Připomeňme si, že pokud $f \in O(g)$, pak $f + g \in O(g)$.
- Pokud tedy pro některou fukci f_i platí, že pro všechny f_j, kde $j \neq i$, je $f_j \in O(f_i)$, pak

$$t \in O(f_i).$$

- Často se tedy při analýze celkové časové složitosti $t(n)$ můžeme omezit pouze na analýzu počtu provedení nejčastěji prováděné instrukce (pro vstup velikosti n je provedena maximálně $f_i(n)$ krát), protože platí

$$t \in \Theta(f_i).$$

Složitost algoritmů

Příklad: Při analýze složitosti vyhledávání čísla v posloupnosti jsme zjistili, že časová složitost daného algoritmu v nejhorším případě je

$$f(n) = 8n + 8\,.$$

Kdybychom to nechtěli takto podrobně zjišťovat a spokojili se s hrubším odhadem, mohli jsme určit, že časová složitost tohoto algoritmu je $\Theta(n)$, protože:

- Algoritmus obsahuje jediný cyklus, který se provede nejvýše n krát, přičemž pro některé vstupy se skutečně může až n krát provést.
- V rámci jednoho průchodu cyklem se provede několik instrukcí, jejichž počet je shora i zdola omezen nějakými konstantami nezávislými na velikosti vstupu.
- Ostatní instrukce se provedou maximálně jednou. K celkové složitosti tak přispívají přičtením nějaké konstanty.

Složitost algoritmů

Pokusme se analyzovat časovou složitost následujícího algoritmu:

INSERTION-SORT(A, n):

for $j := 2$ **to** n **do**
  $x := A[j]$
  $i := j - 1$
  while $i > 0$ and $A[i] > x$ **do**
    $A[i+1] := A[i]$
    $i := i - 1$
  end while
  $A[i+1] := x$
end for

Tj. chceme najít funkci $t(n)$ takovou, že časová složitost algoritmu INSERTION-SORT v nejhorším případě je v $\Theta(t(n))$.

Složitost algoritmů

Uvažujme vstupy velikosti n:

- Vnější cyklus **for** se provede $n-1$ krát.
- Vnitřní cyklus **while** se pro danou hodnotu j provede maximálně $(j-1)$ krát.
- Existují vstupy, pro které platí že pro každou hodnotu j od 2 do n se vnitřní cyklus **while** provede právě $(j-1)$ krát.
- V nejhorším případě se tedy cyklus **while** provede celkem m krát, kde

$$m = 1 + 2 + \cdots + (n-1) = (1 + (n-1)) \cdot \tfrac{n-1}{2} = \tfrac{1}{2}n^2 - \tfrac{1}{2}n$$

- Celková časová složitost algoritmu Insertion-Sort v nejhorším případě je tedy $\Theta(n^2)$.

Složitost algoritmů

V předchozím případě jsme přesně spočítali celkový počet průchodů cyklem **while**.

Obecně to není vždy možné spočítat takto přesně nebo to může být hodně komplikované. Pokud nás zajímá jen asymptotický odhad, tak to často ani není nutné.

Složitost algoritmů

Pokud bychom například neuměli spočítat součet aritmetické posloupnosti, mohli bychom provést analýzu následovně:

- Vnější cyklus **for** se neprovede více než n krát, vnitřní cyklus **while** se při každé iteraci vnějšího cyklu provede maximálně n krát. Celkově se tedy vnitřní cyklus provede maximálně n^2 krát.

  Platí tedy $t \in O(n^2)$.

- Pro některé vstupy se při posledních $\lfloor n/2 \rfloor$ průchodech cyklem **for** provede cyklus **while** alespoň $\lceil n/2 \rceil$ krát.

  Pro některé vstupy se tedy cyklus **while** provede alespoň $\lfloor n/2 \rfloor \cdot \lceil n/2 \rceil$ krát.

  $\lfloor n/2 \rfloor \cdot \lceil n/2 \rceil \geq (n/2 - 1) \cdot (n/2) = \frac{1}{4}n^2 - \frac{1}{2}n$

  Platí tedy $t \in \Omega(n^2)$.

Složitost algoritmů

Při používání asymptotických odhadů časové složitosti algoritmů bychom si měli být vědomi některých úskalí:

- Asyptotické odhady se týkají pouze toho, jak roste čas s rostoucí velikostí vstupu.
- Neříkají nic o konkrétní době výpočtu. V asymptotické notaci mohou být skryty velké konstanty.
- Algoritmus, který má lepší asymptotickou časovou složitost než nějaký jiný algoritmus, může být ve skutečnosti rychlejší až pro nějaké hodně velké vstupy.
- Většinou analyzujeme složitost v nejhorším případě. Pro některé algoritmy může být doba výpočtu v nejhorším případě mnohem větší než doba výpočtu na „typických“ instancích.

Složitost algoritmů

- Můžeme si to ilustrovat na algoritmech pro třídění.

Algoritmus	Nejhorší	Průměrný
Bubblesort	$\Theta(n^2)$	$\Theta(n^2)$
Heapsort	$\Theta(n \log n)$	$\Theta(n \log n)$
Quicksort	$\Theta(n^2)$	$\Theta(n \log n)$

- Quicksort má horší asymptotickou složitost v nejhorším případě než Heapsort, stejnou asymptotickou složitost v průměrném případě a přesto je v praxi nejrychlejší.

Složitost algoritmů

- Zatím jsme uvažovali, že provedení všech instrukcí trvá stejně dlouho.
- V praxi mohou být některé instrukce časově náročnější.
- Pokud známe časy provedení jednotlivých instrukcí a můžeme si spočítat počty jejich provedení, čas běhu potom dostaneme jako

$$\sum_i m_i t_i$$

kde m_i je počet provedení instrukce i a t_i je čas potřebný k vykonání této instrukce.

Složitost algoritmů

Předpokládejme, že analyzujeme časovou složitost nějakého algoritmu realizovaného strojem RAM.

- Zatím jsme při analýze počítali jen počet provedených instrukcí. Tato se označuje jako použití tzv. **jednotkové míry**.

  Odhady časové složitosti v jednotkové míře odpovídají době běhu na skutečných počítačích za předpokladu, že operace, které provádí stroj RAM, může skutečný počítač provést v konstantním čase.

  To platí, pokud čísla, se kterými algoritmus pracuje, jsou malá (vejdou se např. do 32 nebo 64 bitů).

- Pokud by stroj RAM pracoval s „velkými“ čísly (např. 1000 bitovými), bude odhad časové složitosti v jednotkové míře nerealistický v tom smyslu, že výpočet na skutečném počítači bude trvat mnohem déle.

Složitost algoritmů

- Proto se při analýze časové složitosti algoritmů, u kterých se předpokládá práce s velkými čísly, používá tzv. **logaritmická míra**, kdy doba trvání jedné instrukce stroje RAM není 1, ale je úměrná počtu **bitových operací**, které je třeba pro provedení dané instrukce provést.
- Doba trvání instrukce je tedy závislá na aktuálních hodnotách jejích operandů.
- Například doba provádění instrukcí `ADD` a `SUB` je rovna součtu počtů bitů jejich operandů.
- Doba provádění instrukcí `MUL` a `DIV` je rovna součinu počtů bitů jejich operandů.

Prostorová (paměťová) složitost algoritmů

- Zatím jsme se zajímali o čas, který potřebujeme k výpočtu
- Někdy bývá kritickou velikost paměti potřebné k provedení výpočtu.

Množstvím paměti stroje RAM $\mathcal{M}$ použitým pro vstup x rozumíme počet buněk paměti, které stroj $\mathcal{M}$ během svého výpočtu nad vstupem x použije.

Definice

Prostorová složitost stroje RAM $\mathcal{M}$ (v nejhorším případě) je funkce $s : \mathbb{N} \to \mathbb{N}$, kde $s(n)$ udává maximální množství paměti použité strojem $\mathcal{M}$ pro vstupy délky n.

Prostorová (paměťová) složitost algoritmů

- Pro konkrétní problém můžeme mít dva algoritmy takové, že jeden má menší prostorovou složitost a druhý zase časovou složitost.
- Je-li časová složitost algoritmu v $O(f(n))$ je i prostorová v $O(f(n))$ (počet buněk navštívených RAMem nemůže být větší než počet kroků, protože v každém kroku použije kromě akumulátoru maximálně jednu další buňku).

Složitost problémů

Složitost problémů

- Ukazuje se, že různé (algoritmické) problémy jsou různě těžké.
- Obtížnější jsou ty problémy, k jejichž řešení potřebujeme více času a paměti.
- Obtížnost problémů chceme nějak posuzovat, a to jak
  - absolutně – kolik času a kolik paměti potřebujeme k jejich řešení, tak
  - relativně – o kolik je jejich řešení obtížnější nebo naopak jednodušší oproti jiným problémům.
- Proč se u některých problémů nedaří nalézt efektivní algoritmy? Může vůbec nějaký efektivní algoritmus pro daný problém existovat?
- Kde přesně jsou limity toho, co je možné prakticky zvládnout?

Složitost problémů

Je potřeba odlišovat **složitost algoritmu** a **složitost problému**.

Pokud například zkoumáme časovou složitost v nejhorším případě, mohli bychom neformálně říct:

- **složitost algoritmu** – funkce, která vyjadřuje, kolik kroků maximálně udělá daný algoritmus pro vstup velikosti n
- **složitost problému** – jaká je časová složitost „nejefektivnějšího“ algoritmu, který řeší daný problém

Zavedení pojmu „složitost problému“ ve výše uvedeném smyslu naráží na značné technické obtíže. Pojem „složitost problému“ se tedy jako takový nedefinuje, ale obchází se zavedením tzv. **tříd složitosti**.

Třídy složitosti

Třídy složitosti jsou podmnožiny množiny všech (algoritmických) **problémů**.

Daná konkrétní třída složitosti je vždy charakterizována nějakou vlastností, kterou mají problémy do ní patřící.

Typickým příkladem takové vlastnosti je vlastnost, že pro daný problém existuje nějaký algoritmus s určitým omezením (např. časové nebo prostorové složitosti):

- Do dané třídy pak patří všechny problémy, pro které takovýto algoritmus existuje.
- Naopak do ní nepatří problémy, pro které žádný takový algoritmus neexistuje.

Třídy složitosti

Definice

Pro libovolnou funkci $f : \mathbb{N} \to \mathbb{N}$ definujeme třídu $\mathcal{T}(f(n))$ jako třídu obsahující právě ty problémy, pro něž existuje algoritmus s časovou složitostí $O(f(n))$.

Příklad:

- $\mathcal{T}(n)$ – třída všech problémů pro něž existuje algoritmus s časovou složitostí $O(n)$
- $\mathcal{T}(n^2)$ – třída všech problémů pro než existuje algoritmus s časovou složitostí $O(n^2)$
- $\mathcal{T}(n \log n)$ – třída všech problémů pro než existuje algoritmus s časovou složitostí $O(n \log n)$

Třídy složitosti

Definice

Pro libovolnou funkci $f : \mathbb{N} \to \mathbb{N}$ definujeme třídu $\mathcal{S}(f(n))$ jako třídu obsahující právě ty problémy, pro něž existuje algoritmus s prostorovou složitostí $O(f(n))$.

Příklad:

- $\mathcal{S}(n)$ – třída všech problémů pro něž existuje algoritmus s prostorovou složitostí $O(n)$
- $\mathcal{S}(n^2)$ – třída všech problémů pro než existuje algoritmus s prostorovou složitostí $O(n^2)$
- $\mathcal{S}(n \log n)$ – třída všech problémů pro než existuje algoritmus s prostorovou složitostí $O(n \log n)$

Třídy složitosti

Poznámka:

Všimněte si, že u tříd $\mathcal{T}(f)$ a $\mathcal{S}(f)$ může to, které problémy do dané třídy patří, záviset na použitém výpočetním modelu (zda je to stroj RAM, jednopáskový Turingův stroj, vícepáskový Turingův stroj, ...).

Třídy složitosti

Pomocí tříd $\mathcal{T}(f(n))$ a $\mathcal{S}(f(n))$ můžeme definovat třídy PTIME a PSPACE jako

$$\text{PTIME} = \bigcup_{k \geq 0} \mathcal{T}(n^k) \qquad \text{PSPACE} = \bigcup_{k \geq 0} \mathcal{S}(n^k)$$

- PTIME je třída všech problémů, pro které existuje algoritmus s polynomiální časovou složitostí, tj. s časovou složitostí $O(n^k)$, kde k je nějaká konstanta.
- PSPACE je třída všech problémů, pro které existuje algoritmus s polynomiální prostorovou složitostí, tj. s prostorovou složitostí $O(n^k)$, kde k je nějaká konstanta.

Třídy složitosti

Poznámka: Vzhledem k tomu, že všechny (rozumné) výpočetní modely jsou schopné se navzájem simulovat tak, že při dané simulaci nevzroste počet kroků ani množství použité paměti víc než polynomiálně, není definice tříd PTIME a PSPACE závislá na použitém výpočetním modelu. Pro jejich zadefinování můžeme použít kterýkoliv výpočetní model.

Říkáme, že tyto třídy jsou **robustní** – jejich definice nezávisí na použitém výpočetním modelu.

Třídy složitosti

Analogicky můžeme zavést další třídy:

EXPTIME – množina všech problémů, pro které existuje algoritmus s časovou složitostí $2^{O(n^k)}$, kde k je nějaká konstanta

EXPSPACE – množina všech problémů, pro které existuje algoritmus s prostorovou složitostí $2^{O(n^k)}$, kde k je nějaká konstanta

LOGSPACE – množina všech problémů, pro které existuje algoritmus s prostorovou složitostí $O(\log n)$

Poznámka: Místo $2^{O(n^k)}$ bychom mohli psát také $O(c^{n^k})$, kde c a k jsou nějaké konstanty.

Vztahy mezi třídami složitosti

Pokud Turingův stroj provede m kroků, tak použije maximálně m políček na pásce.

Pokud tedy existuje pro nějaký problém algoritmus s časovou složitostí $O(f(n))$, má tento algoritmus paměťovou složitost (nejvýše) $O(f(n))$.

Je tedy zřejmé, že platí následující vztah.

Pozorování

Pro libovolnou funkci $f : \mathbb{N} \to \mathbb{N}$ platí $\mathcal{T}(f(n)) \subseteq \mathcal{S}(f(n))$.

Poznámka: Analogicky bychom mohli argumentovat například pro stroj RAM.

Vztahy mezi třídami složitosti

Z předchozího okamžitě plyne:

$$\text{PTIME} \subseteq \text{PSPACE}$$
$$\text{EXPTIME} \subseteq \text{EXPSPACE}$$

Vzhledem k tomu, že polynomiální funkce rostou pomaleji než exponenciální, zjevně platí:

$$\text{PTIME} \subseteq \text{EXPTIME}$$

$$\text{LOGSPACE} \subseteq \text{PSPACE} \subseteq \text{EXPSPACE}$$

Vztahy mezi třídami složitosti

- Pro libovolná dvě reálná čísla $0 \leq \epsilon_1 < \epsilon_2$ platí

$$\mathcal{S}(n^{\epsilon_1}) \subsetneq \mathcal{S}(n^{\epsilon_2})$$

- LOGSPACE $\subsetneq$ PSPACE
- PSPACE $\subsetneq$ EXPSPACE

- Pro libovolná dvě reálná čísla $0 \leq \epsilon_1 < \epsilon_2$ platí

$$\mathcal{T}(n^{\epsilon_1}) \subsetneq \mathcal{T}(n^{\epsilon_2})$$

- PTIME $\subsetneq$ EXPTIME

Vztahy mezi třídami složitosti

Při zkoumání vztahů mezi třídami složitosti se ukazuje jako užitečný pojem **konfigurace**.

Konfigurací budeme rozumět celkový stav, ve kterém se během jednoho kroku nachází stroj, provádějící nějaký daný algoritmus.

- U Turingova stroje je konfigurace dána stavem jeho řídící jednotky, obsahem pásky (resp. pásek) a pozicí hlavy (resp. hlav).
- U stroje RAM je konfigurace dána obsahem paměti, obsahem všech registrů (včetně IP), obsahem vstupní a výstupní pásky a pozicemi čtecí a zapisovací hlavy.

Vztahy mezi třídami složitosti

Mělo by být jasné, že konfigurace (resp. jejich popisy) můžeme zapisovat jako slova v nějaké abecedě.

Navíc můžeme konfigurace zapisovat tak, že délka těchto slov bude zhruba stejná jako množství paměti použité algoritmem (tj. počet políček na pásce použitých Turingovým stojem, počet bitů paměti použitých strojem RAM apod.).

Poznámka: Pokud máme abecedu Σ, kde $|\Sigma| = c$, tak:

- Počet slov délky n je c^n, tj. $2^{\Theta(n)}$.
- Počet slov délky nejvýše n je

$$\sum_{i=0}^{n} c^n = \frac{c^{n+1} - 1}{c - 1}$$

tj. také $2^{\Theta(n)}$.

Vztahy mezi třídami složitosti

Je jasné, že během výpočtu korektního algoritmu se žádná konfigurace nemůže zopakovat, protože jinak by se algoritmus zacyklil a běžel by donekonečna.

Pokud tedy víme, že paměťová složitost nějakého algoritmu je $O(f(n))$, znamená to, že počet různých konfigurací dosažitelných během výpočtu je $2^{O(f(n))}$.

Protože se konfigurace během žádného výpočtu neopakují, je i časová složitost daného algoritmu maximálně $2^{O(f(n))}$.

Pozorování

Pro libovolnou funkci $f : \mathbb{N} \to \mathbb{N}$ platí, že $\mathcal{S}(f(n)) \subseteq \mathcal{T}(2^{f(n)})$.

Vztahy mezi třídami složitosti

Z předchozího plynou následující důsledky:

$$\mathrm{LOGSPACE} \subseteq \mathrm{PTIME}$$
$$\mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$$

Vztahy mezi třídami složitosti

Shrnutí:

$$\mathrm{LOGSPACE} \subseteq \mathrm{PTIME} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME} \subseteq \mathrm{EXPSPACE}$$

- $\mathrm{PTIME} \subsetneq \mathrm{EXPTIME}$
- $\mathrm{LOGSPACE} \subsetneq \mathrm{PSPACE}$

Horní a dolní odhady složitosti problémů

Horním odhadem složitosti problému rozumíme to, že složitost problému není vyšší než nějaká uvedená.

Většinou je to formulováno tak, že daný problém patří do nějaké určité třídy složitosti.

Příklady tvrzení, které se týkají horních odhadů složitosti:

- Problém dosažitenosti v grafu je v PTIME.
- Problém ekvivalence dvou regulárních výrazů je v EXPSPACE.

Pokud chceme zjistit nějaký horní odhad složitosti problému, stačí ukázat, že existuje algoritmus s danou složitostí.

Horní a dolní odhady složitosti problémů

Dolním odhadem složitosti problému rozumíme to, že složitost problému je alespoň taková jako nějaká uvedená.

Obecně je zjišťování (netriviálních) dolních odhadů složitosti problémů mnohem obtížnější než zjišťování horních odhadů.

Pro odvození dolního odhadu musíme totiž ukázat, že **každý** algoritmus řešící daný problém má danou složitost.

Horní a dolní odhady složitosti problémů

Problém „Třídění“

Vstup: Posloupnost prvků $a_1, a_2, \ldots, a_n$.

Výstup: Prvky $a_1, a_2, \ldots, a_n$ setříděné od nejmenšího po největší.

Dá se dokázat, že každý algoritmus, který řeší problém “Třídění” a na prvcích tříděné posloupnosti používá pouze operaci porovnávání (tj. nezkoumá obsah těchto prvků), má časovou složist v nejhorším případě v $\Omega(n \log n)$ (tj. pro každý takový algoritmus existují konstanty $c > 0$ a $n_0 \geq 0$ takové, že pro každé $n \geq n_0$ existuje vstup velikosti n, pro který provede algoritmus nejmémě $cn \log n$ operací).

Nedeterminismus

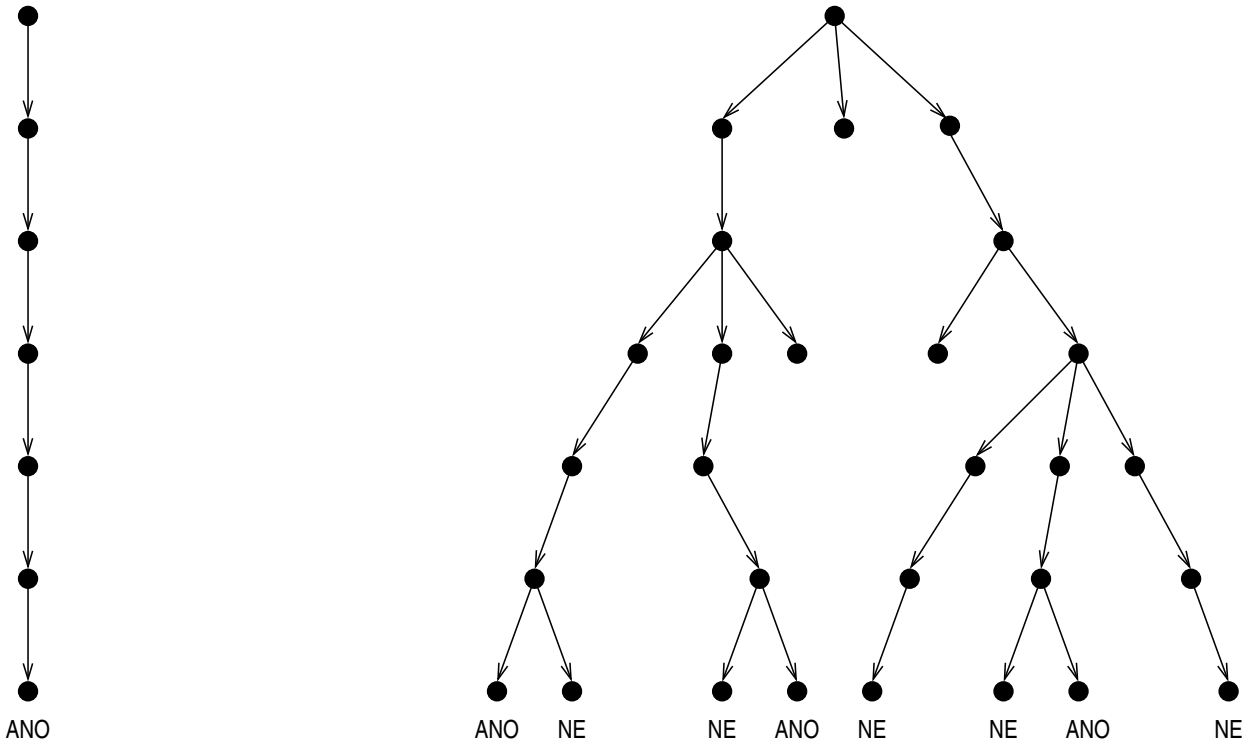
Nedeterminismus

Nedeterministický stroj RAM:

- Je definován velice podobně jako deterministický RAM.
- Navíc má instrukci NDJUMP x OR y, která umožňuje stroji vybrat si jedno z možných pokračování.
- Pokud ze všech možných výpočtů takového stroje nad zadaným vstupem alespoň jeden skončí s odpovědí ANO, je odpověď ANO.
- Pokud všechny výpočty skončí s odpovědí NE, je odpověď NE.

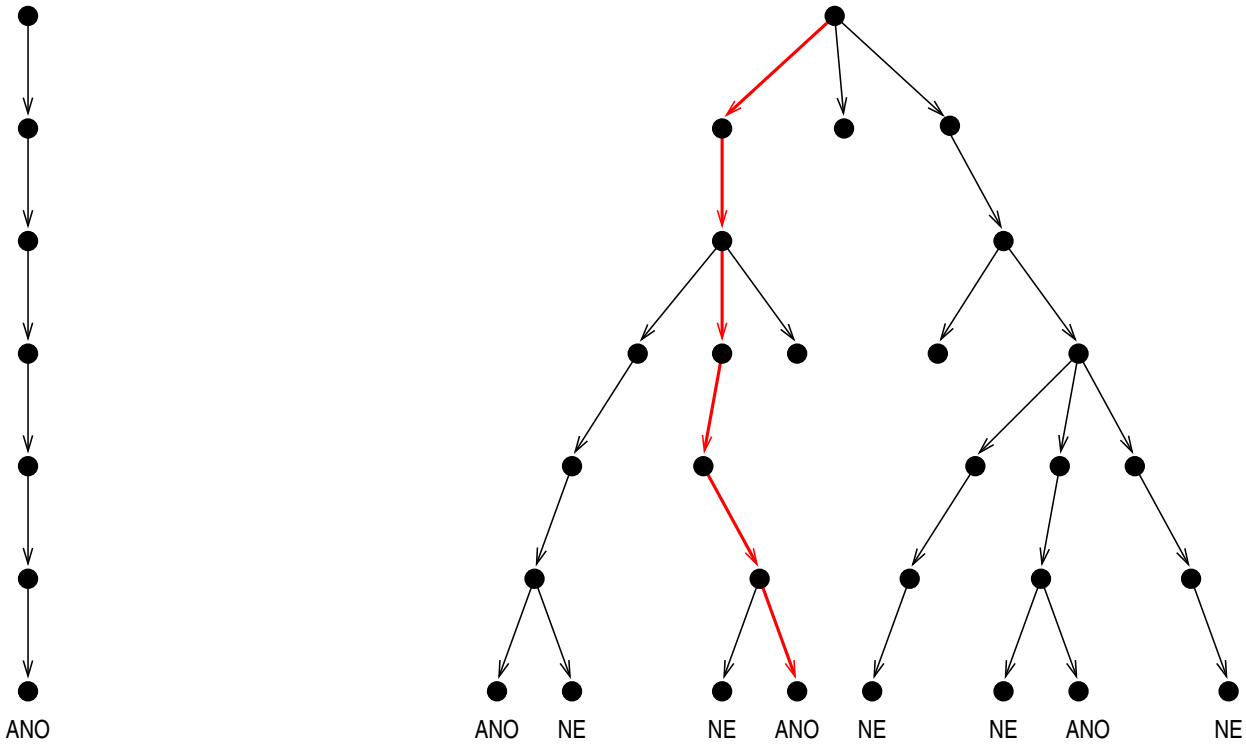
Podobně můžeme definovat nedeterministické verze jiných výpočetních modelů, např. nedeterministické Turingovy stroje.

Nedeterminismus



- Doba výpočtu nedeterministického stroje RAM (nebo jiného nedeterministického stroje) nad zadaným vstupem je definována jako délka nejdelšího možného výpočtu nad tímto vstupem.

Nedeterminismus



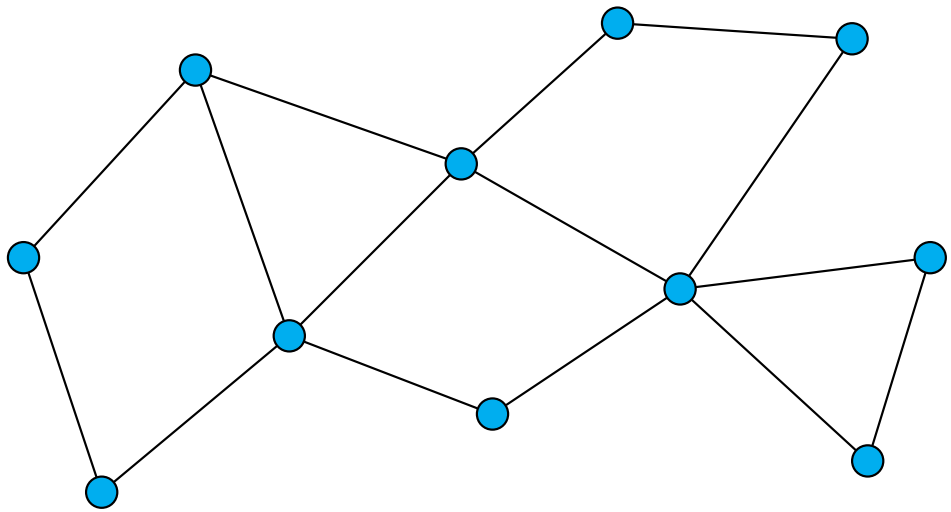
- Doba výpočtu nedeterministického stroje RAM (nebo jiného nedeterministického stroje) nad zadaným vstupem je definována jako délka nejdelšího možného výpočtu nad tímto vstupem.

Nedeterminismus

Problém „Barvení grafu k barvami“

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Je možné obarvit vrcholy grafu G k barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?



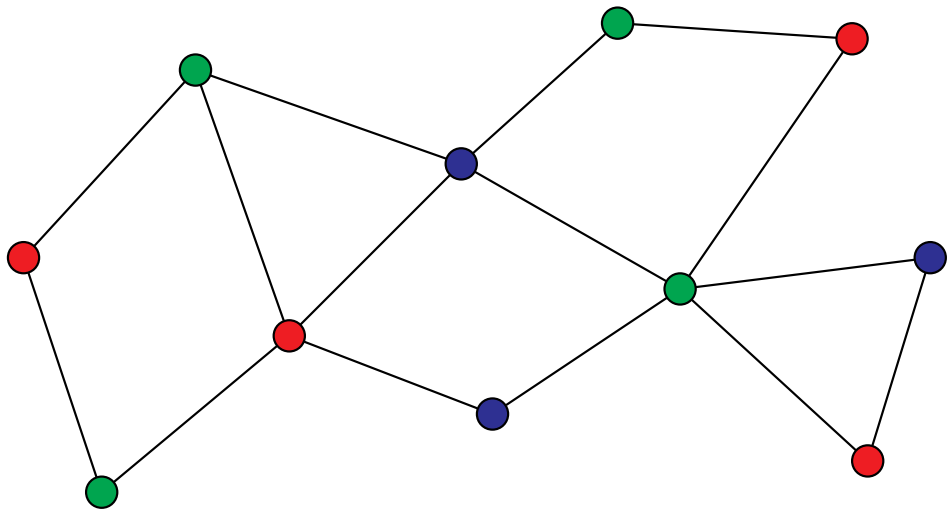
$k = 3$

Nedeterminismus

Problém „Barvení grafu k barvami“

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Je možné obarvit vrcholy grafu G k barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?



$k = 3$

Nedeterminismus

Problém „Barvení grafu k barvami“

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Je možné obarvit vrcholy grafu G k barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?

Nedeterministický algoritmus pracuje následovně:

1. Každému vrcholu grafu G nedeterministicky přiřadí jednu z k barev.
2. Projde všechny hrany grafu G a u každé z nich zkotroluje, že oba její koncové vrcholy jsou obarveny různými barvami. Pokud ne, skončí s odpovědí NE.
3. Pokud prošel všechny hrany a u všech byly koncové vrcholy obarveny různými barvami, skončí s odpovědí ANO.

Nedeterminismus

Problém „Isomorfismus grafů“

Vstup: Neorientované grafy $G_1 = (V_1, E_1)$ a $G_2 = (V_2, E_2)$.

Otázka: Jsou grafy G_1 a G_2 isomorfní?

Poznámka: Grafy G_1 a G_2 jsou isomorfní, jestliže existuje nějaká bijekce $f : V_1 \to V_2$ taková, že pro libovolné dva vrcholy $u, v \in V_1$ platí $(u, v) \in E_1$ právě když $(f(u), f(v)) \in E_2$.

Nedeterministický algoritmus pracuje následovně:

1. Nedeterministicky zvolí hodnoty funkce f pro všechny $v \in V_1$.
2. Deterministicky ověří, že f je bijekce a že pro všechny dvojice vrcholů je splněna výše uvedená podmínka.
3. Pokud je některá z podmínek porušena, skončí s odpovědí NE, v opačném případě s odpovědí ANO.

Nedeterminismus

Na nedeterminismus můžeme nahlížet dvěma způsoby:

1. Ve chvíli, kdy má stroj nedeterministicky zvolit mezi několika možnostmi, tak „uhodne“, která z těchto možností povede k odpovědi ANO (pokud taková možnost existuje).

2. Ve chvíli, kdy má stroj nedeterministicky zvolit mezi několika možnostmi, rozdělí se do tolika kopií, kolik je těchto možností, a každá z těchto kopií pokračuje ve výpočtu odpovídající jedné z možností, přičemž pracují všechny paralelně.

   Odpověď je ANO právě tehdy, když alespoň jedna z kopií stroje odpoví ANO.

Nedeterminismus

- Z hlediska rozhodnutelnosti nepřináší nedeterministické algoritmy oproti deterministickým nic dalšího navíc:
  Pokud je nějaký problém možné řešit nedeterministickým strojem RAM nebo TS, tak je ho možné řešit i deterministickým, který postupně vyzkouší všechny možné výpočty nedeterministického stroje nad daným vstupem.
- Nedeterminismus má význam především při zkoumání složitosti problémů.

Nedeterministické třídy složitosti

Definice

Pro funkci $f : \mathbb{N} \to \mathbb{N}$ rozumíme **třídou časové složitosti** $\mathcal{NT}(f)$ množinu těch problémů, které jsou řešeny nedeterministickými RAMy s časovou složitostí v $O(f(n))$.

Definice

Pro funkci $f : \mathbb{N} \to \mathbb{N}$ rozumíme **třídou prostorové složitosti** $\mathcal{NS}(f)$ množinu těch problémů, které jsou řešeny nedeterminictickými RAMy s prostorovou složitostí v $O(f(n))$.

Třída NPTIME

Definice

$$\text{NPTIME} = \bigcup_{k=0}^{\infty} \mathcal{NT}(n^k)$$

- NPTIME (někdy se píše jen NP) je třída všech problémů, pro které existuje nedeterministický algoritmus s polynomiání časovou složitostí.
- Do NPTIME tedy patří problémy, u kterých je možné pro daný vstup rychle ověřit, že odpověď je ANO, pokud nám ten, kdo nás o tom chce přesvědčit, dodá nějakou dodatečnou informaci.
- Je zřejmé, že PTIME $\subseteq$ NPTIME, neboť na deterministické algoritmy se můžeme dívat jako na speciální případ nedeterministických.

Třídy složitosti

- Podobně můžeme definovat třídu NPSPACE.
- Obdobně jako v deterministickém případě zřejmě platí $\mathrm{NPTIME} \subseteq \mathrm{NPSPACE}$.
- Nedeterministický RAM můžeme simulovat deterministickým, který zkusí všechny možné výpočty. Každý z možných výpočtů použije maximálně polynomiálně mnoho paměti a jednotlivé výpočty si nic nepředávájí, takže můžeme používat pro všechny stejnou oblast paměti. Platí tedy $\mathrm{NPTIME} \subseteq \mathrm{PSPACE}$.
- Je rovněž známo, že pokud je problém rozhodován nedeterministickým Turingovým strojem s prostorovou složitostí $f(n)$, tak je rozhodován i nějakým deterministickým Turingovým strojem s prostorovou složitostí $O((f(n))^2)$. Z toho plyne, že $\mathrm{NPSPACE} \subseteq \mathrm{PSPACE}$.

$$\mathrm{PTIME} \subseteq \mathrm{NPTIME} \subseteq \mathrm{PSPACE} = \mathrm{NPSPACE}$$

NP-úplné problémy

Polynomiální převod mezi problémy

Mějme nějaké dva rozhodovací problémy A a B.

Problém A je **převeditelný** na problém B, jestliže existuje algoritmus P takový, že:

- Jako vstup může dostat libovolnou instanci problému A.
- K instanci problému A, kterou dostane jako vstup (označme ji x), vyprodukuje jako svůj výstup instanci problému B (označme ji $P(x)$).
- Platí

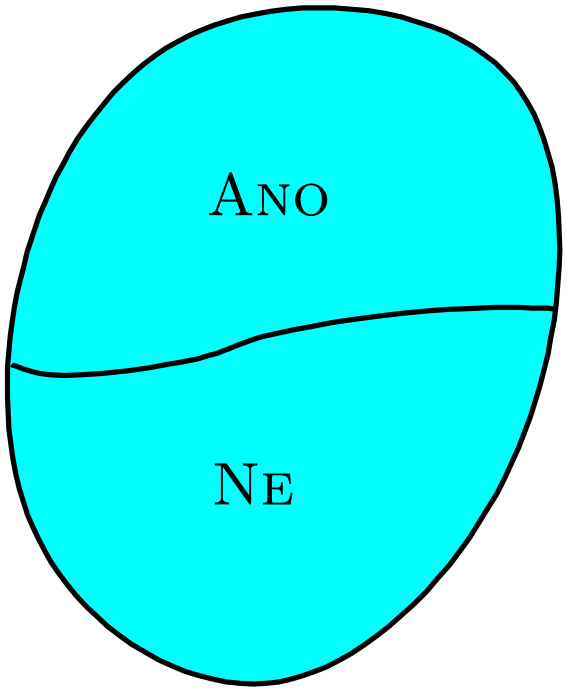
$$x \in A \quad \Leftrightarrow \quad P(x) \in B$$

tj. pro vstup x je v problému A odpověď ANO právě tehdy, když pro vstup $P(x)$ je v problému B odpověď ANO.

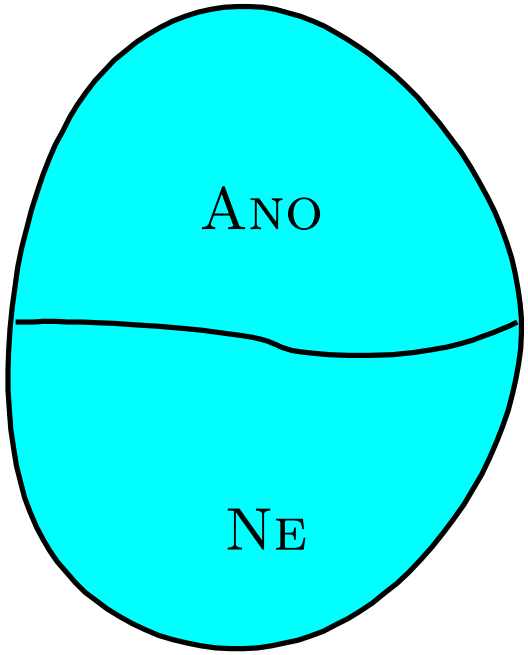
Pokud je navíc algoritmus P polynomiální, pak říkáme, že problém A je **polynomiálně převeditelný** na problém B.

Polynomiální převod mezi problémy

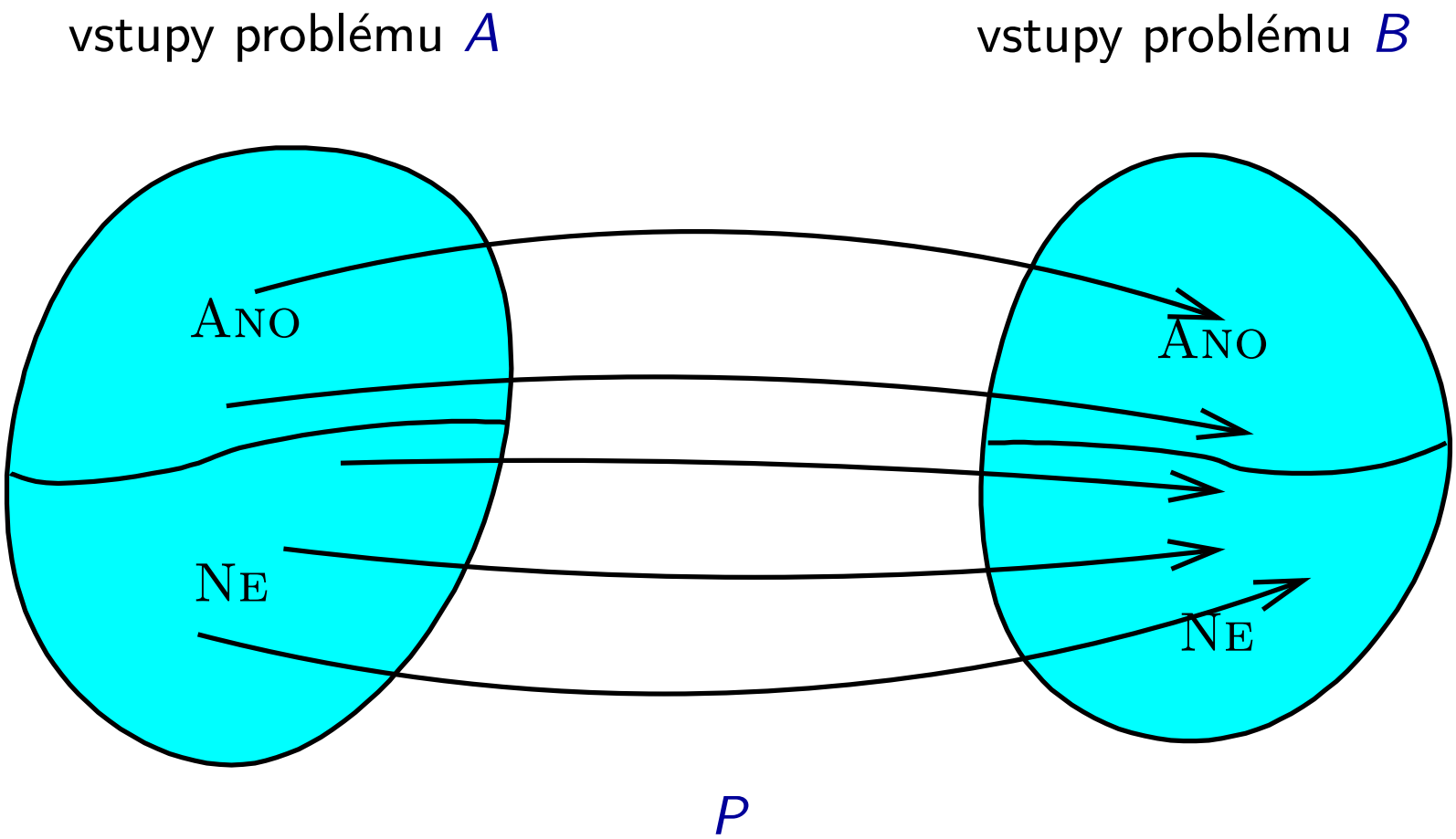
vstupy problému A



vstupy problému B



Polynomiální převod mezi problémy



Polynomiální převod mezi problémy

Řekněme, že problém A je polynomiálně převeditelný na problém B, tj. existuje (polynomiální) algoritmus P realizující tento převod.

Pokud je problém B ve třídě PTIME, pak i problém A je ve třídě PTIME.

Řešení problému A pro vstup x:

- Zavoláme P se vstupem x, vrátí nám hodnotu $P(x)$.
- Zavoláme algoritmus řešící problém B se vstupem $P(x)$. Hodnotu, kterou nám vrátí, vypíšeme jako výsledek.

Z toho plyne:

Pokud A není v PTIME, tak ani B nemůže být v PTIME.

Problém SAT

Definice problému SAT:

SAT (splnitelnost booleovských formulí)

Vstup: Booleovská formule φ.

Otázka: Je φ splnitelná?

Příklad:

Formule $\varphi_1 = x_1 \wedge (\neg x_2 \vee x_3)$ je splnitelná:
např. při ohodnocení ν, kde $[x_1]_\nu = 1$, $[x_2]_\nu = 0$, $[x_3]_\nu = 1$, platí $[\varphi_1]_\nu = 1$.

Formule $\varphi_2 = (x_1 \wedge \neg x_1) \vee (\neg x_2 \wedge x_3 \wedge x_2)$ není splnitelná:
pro libovolné ohodnocení ν platí $[\varphi_2]_\nu = 0$.

Problém 3-SAT

3-SAT je varianta problému SAT, ve které se omezujeme na formule určitého speciálního typu:

3-SAT

Vstup: Formule φ v konjunktivní normální formě, kde každá klauzule obsahuje právě 3 literály.

Otázka: Je φ splnitelná?

Problém 3-SAT

Některé pojmy:

- **Literál** je formule tvaru x nebo $\neg x$, kde x je booleovská proměnná.
- **Klauzule** je disjunkce literálů.

  Příklady: $x_1 \vee \neg x_2$ $\neg x_5 \vee x_8 \vee \neg x_{15} \vee \neg x_{23}$ x_6
- Formule je v **konjunktivní normální formě (KNF)**, jestliže je konjunkcí klauzulí.

  Příklad: $(x_1 \vee \neg x_2) \wedge (\neg x_5 \vee x_8 \vee \neg x_{15} \vee \neg x_{23}) \wedge x_6$

V případě problému 3-SAT tedy vyžadujeme, aby formule φ byla v KNF a navíc, aby každá klauzule obsahovala právě tři literály.

Příklad:

$(x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee x_4)$

Problém 3-SAT

Následující formule je splnitelná:

$$(x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee x_3) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee x_4)$$

Např. při ohodnocení ν, kde

$$[x_1]_\nu = 0$$
$$[x_2]_\nu = 1$$
$$[x_3]_\nu = 0$$
$$[x_4]_\nu = 1$$

je $[\varphi_1]_\nu = 1$.

Naproti tomu následující formule není splnitelná:

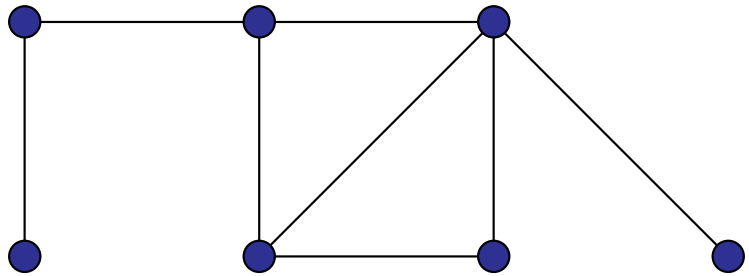
$$(x_1 \vee x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_1 \vee \neg x_1)$$

Problém nezávislé množiny (IS)

Problém nezávislé množiny (IS)

Vstup: Neorientovaný graf G, číslo k.

Otázka: Existuje v grafu G nezávislá množina velikosti k?



$k = 4$

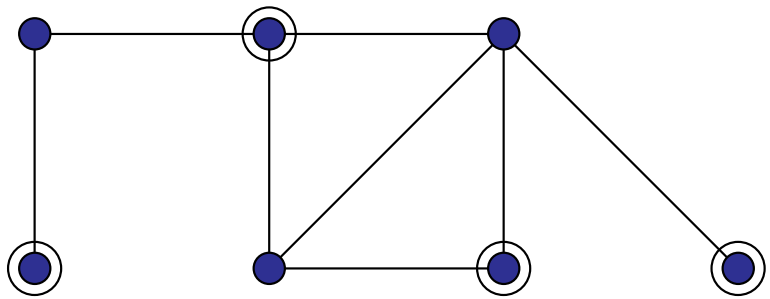
Poznámka: **Nezávislá množina** v grafu je podmnožina vrcholů grafu taková, že žádné dva vrcholy z této podmnožiny nejsou spojeny hranou.

Problém nezávislé množiny (IS)

Problém nezávislé množiny (IS)

Vstup: Neorientovaný graf G, číslo k.

Otázka: Existuje v grafu G nezávislá množina velikosti k?

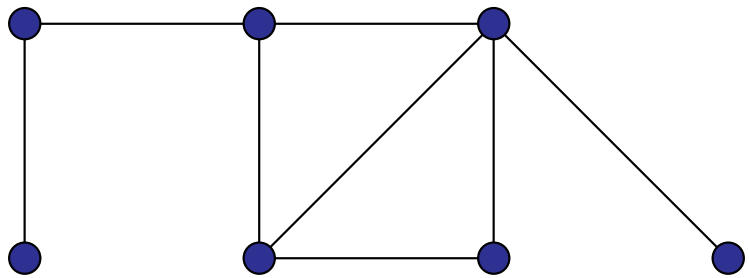


$k = 4$

Poznámka: **Nezávislá množina** v grafu je podmnožina vrcholů grafu taková, že žádné dva vrcholy z této podmnožiny nejsou spojeny hranou.

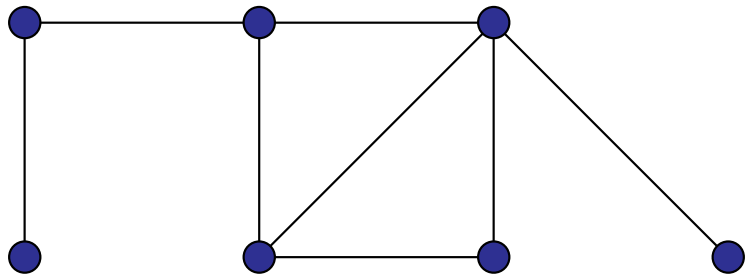
Problém nezávislé množiny (IS)

Příklad instance, kde je odpověď Ano:



$k = 4$

Příklad instance, kde je odpověď Ne:



$k = 5$

Převod 3-SAT na IS

Popíšeme (polynomiální) algoritmus, který bude mít následující vlastnosti:

- **Vstup:** Libovolná instance problému 3-SAT, tj. formule φ v konjunktivní normální formě, kde každá klauzule obsahuje právě tři literály.
- **Výstup:** Instance problému IS, tj. neorientovaný graf G a číslo k.
- Navíc bude pro libovolný vstup (tj. pro libovolnou formuli φ ve výše uvedeném tvaru) zaručeno následující:

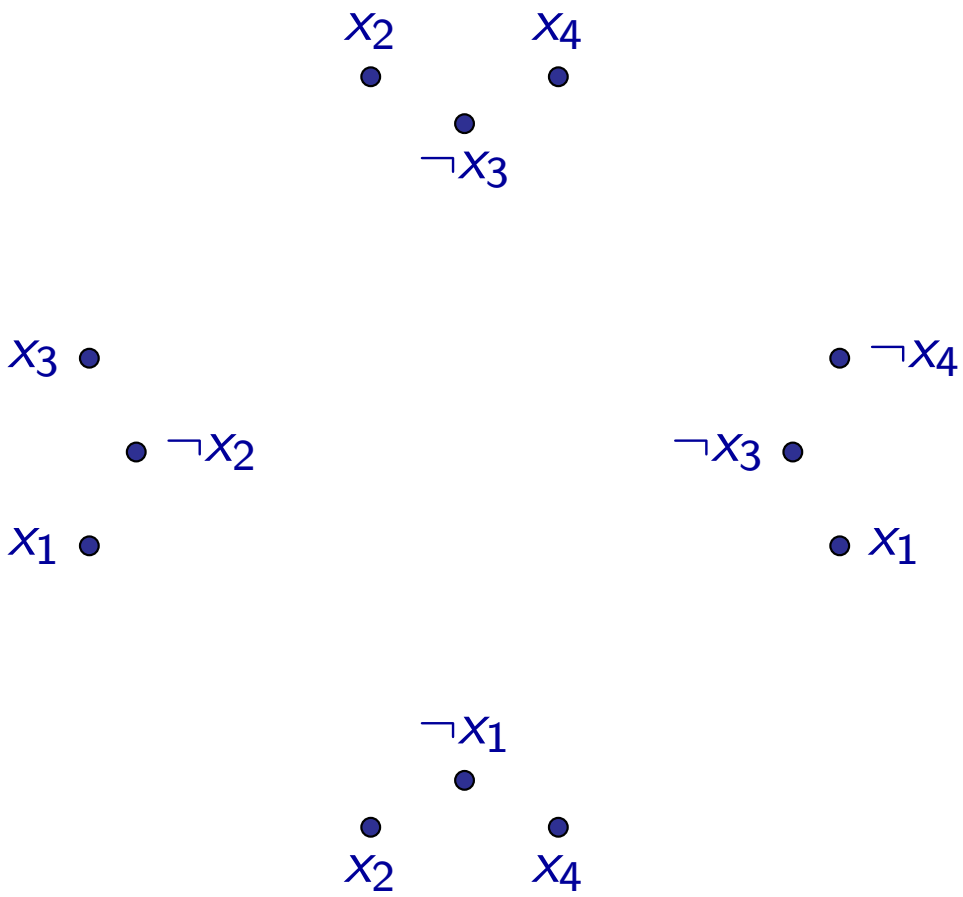
  V grafu G bude existovat nezávislá množina velikosti k právě tehdy, když formule φ bude splnitelná.

Převod 3-SAT na IS

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Převod 3-SAT na IS

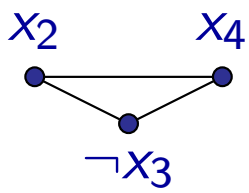
$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

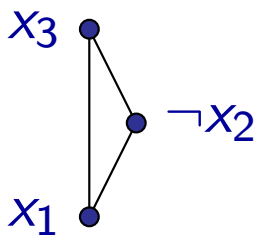


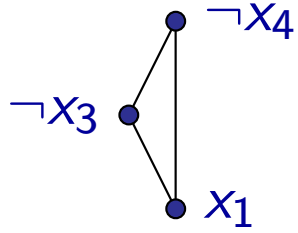
Pro každý literál přidáme do grafu jeden vrchol.

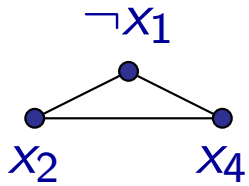
Převod 3-SAT na IS

$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$





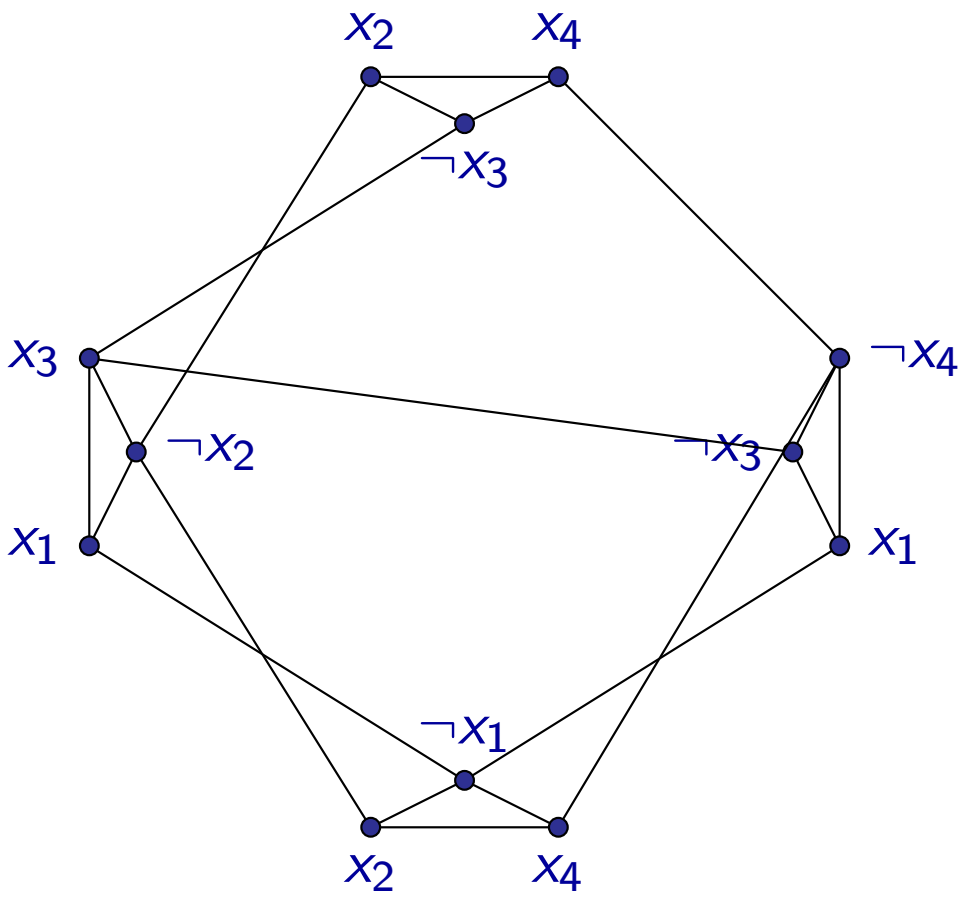




Vrcholy odpovídající literálům patřícím do stejné klauzule spojíme hranami.

Převod 3-SAT na IS

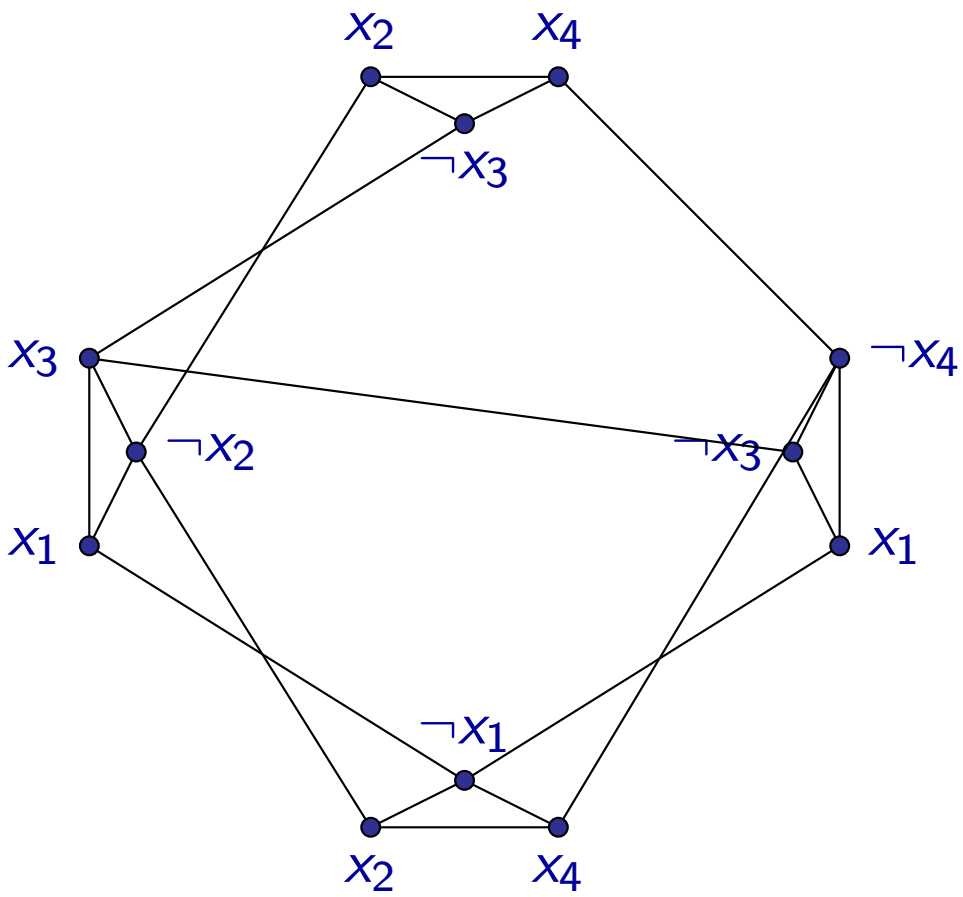
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$



Dvojice vrcholů odpovídající literálům x_i a $\neg x_i$ spojíme hranami.

Převod 3-SAT na IS

$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$

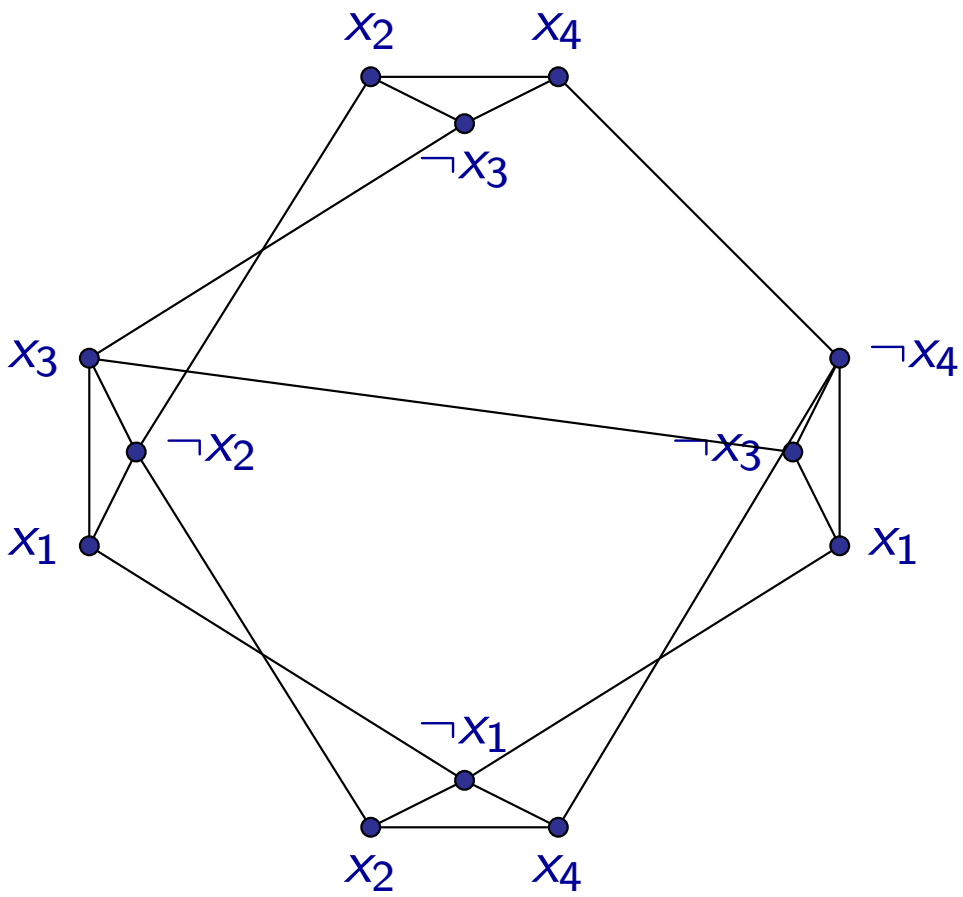


$k = 4$

Číslo k položíme rovno počtu klauzulí.

Převod 3-SAT na IS

$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$



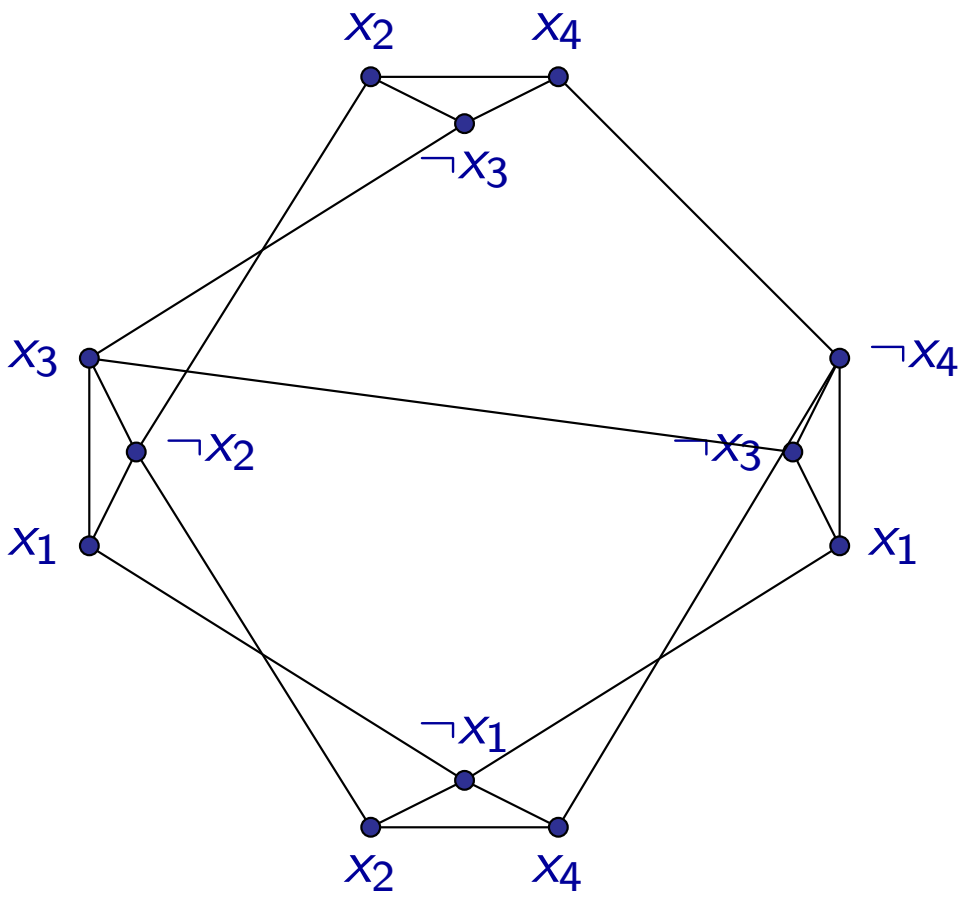
$k = 4$

Vytvořený graf a číslo k vydá algoritmus jako výstup.

Převod 3-SAT na IS

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

$\nu(x_1) = 1$
$\nu(x_2) = 1$
$\nu(x_3) = 0$
$\nu(x_4) = 1$



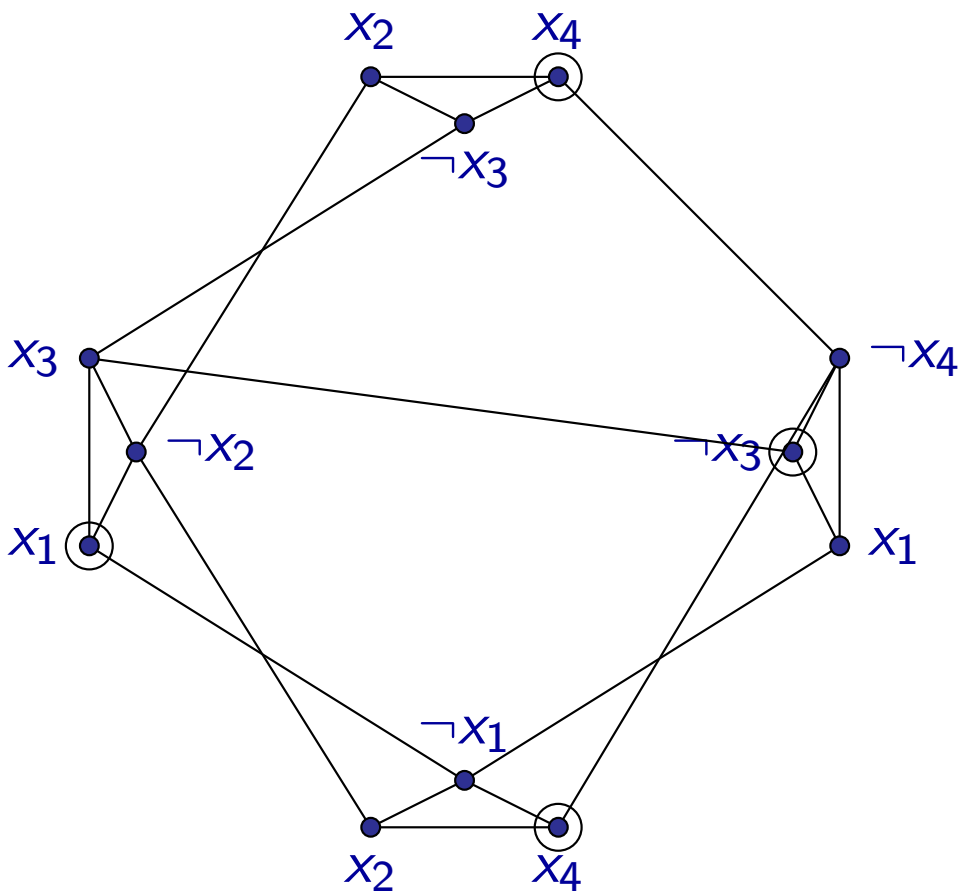
$k = 4$

Jestliže je formule φ splnitelná, existuje ohodnocení ν, při kterém má v každé klauzuli alespoň jeden literál hodnotu 1.

Převod 3-SAT na IS

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

$\nu(x_1) = 1$
$\nu(x_2) = 1$
$\nu(x_3) = 0$
$\nu(x_4) = 1$



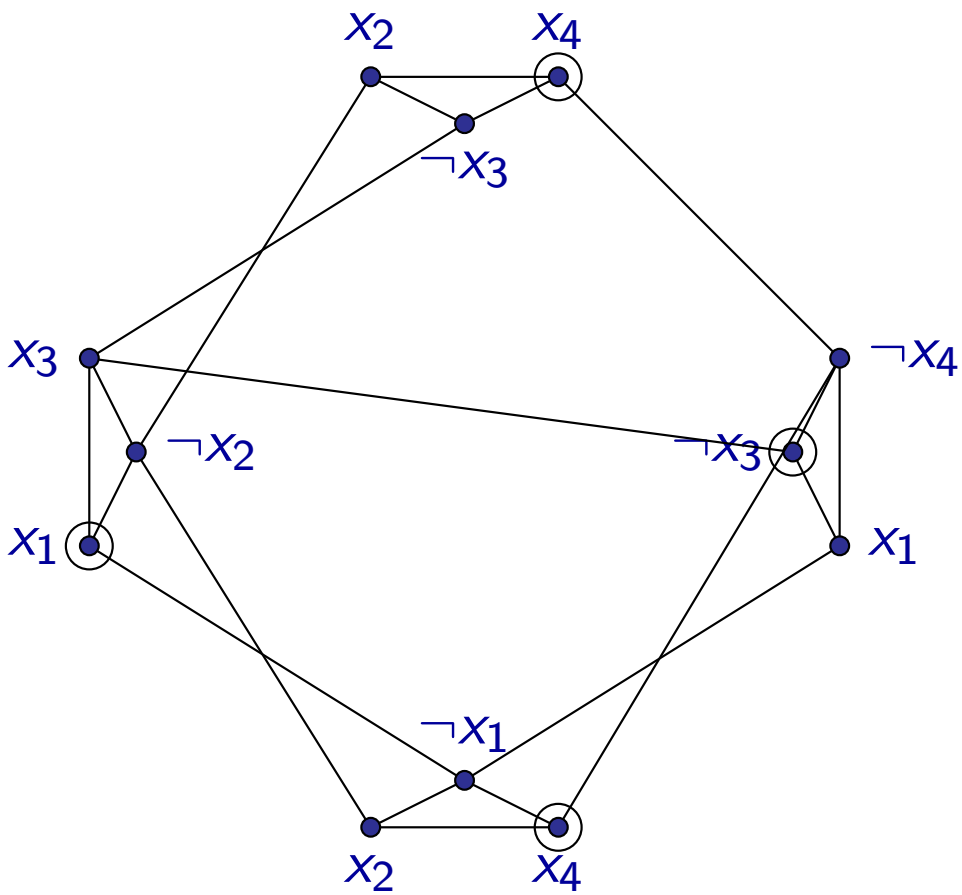
$k = 4$

Z každé klauzule vybereme jeden literál, který má při ohodnocení ν hodnotu 1, a do nezávislé množiny přidáme odpovídající vrchol.

Převod 3-SAT na IS

$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$

$\nu(x_1) = 1$
$\nu(x_2) = 1$
$\nu(x_3) = 0$
$\nu(x_4) = 1$



$k = 4$

Lehce ověříme, že vybrané vrcholy tvoří nezávislou množinu.

Převod 3-SAT na IS

Vybrané vrcholy tvoří nezávislou množinu, protože:

- Z každé trojice vrcholů odpovídající jedné klauzuli byl vybrán jen jeden vrchol.
- Nemohly být současně vybrány vrcholy označené x_i a $\neg x_i$.
  (Při daném ohodnocení ν má hodnotu 1 jen jeden z nich.)

Převod 3-SAT na IS

Na druhou stranu, pokud v grafu G existuje nezávislá množina velikosti k, musí určitě splňovat následující vlastnosti:

- Z každé trojice vrcholů odpovídající jedné klauzuli musí být vybrán nejvýše jeden vrchol.

  Protože je ale klauzulí k a je vybráno k vrcholů, musí být z každé takové trojice vybrán právě jeden.

- Nemohly být současně vybrány vrcholy označené x_i a $\neg x_i$.

Ohodnocení tedy zvolíme podle vybraných vrcholů, protože z předchozího vyplývá, že nehrozí, že by neexistovalo.
(Zbylým proměnným přiřadíme libovolné hodnoty.)

Při daném ohodnocení má formule φ určitě hodnotu 1, neboť v každé klauzuli má hodnotu 1 alespoň jeden literál.

Převod 3-SAT na IS

Popsaný algoritmus je určitě polynomiální:

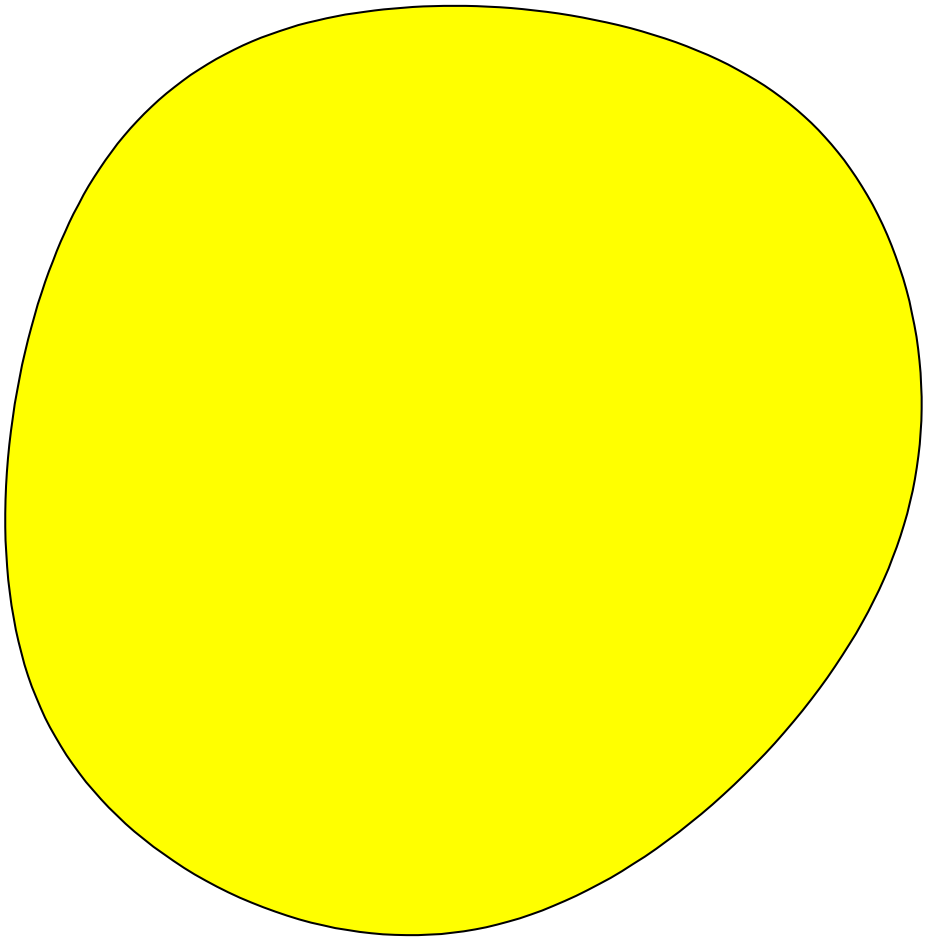
Graf G a číslo k je možné zkonstruovat k formuli φ v čase $O(n^2)$, kde n je velikost formule φ.

Navíc jsme viděli, že ve zkonstruovaném grafu G existuje nezávislá množina velikosti k právě tehdy, když formule φ je splnitelná.

Popsaný algoritmus tedy ukazuje, že problém 3-SAT je polynomiálně převeditelný na problém IS.

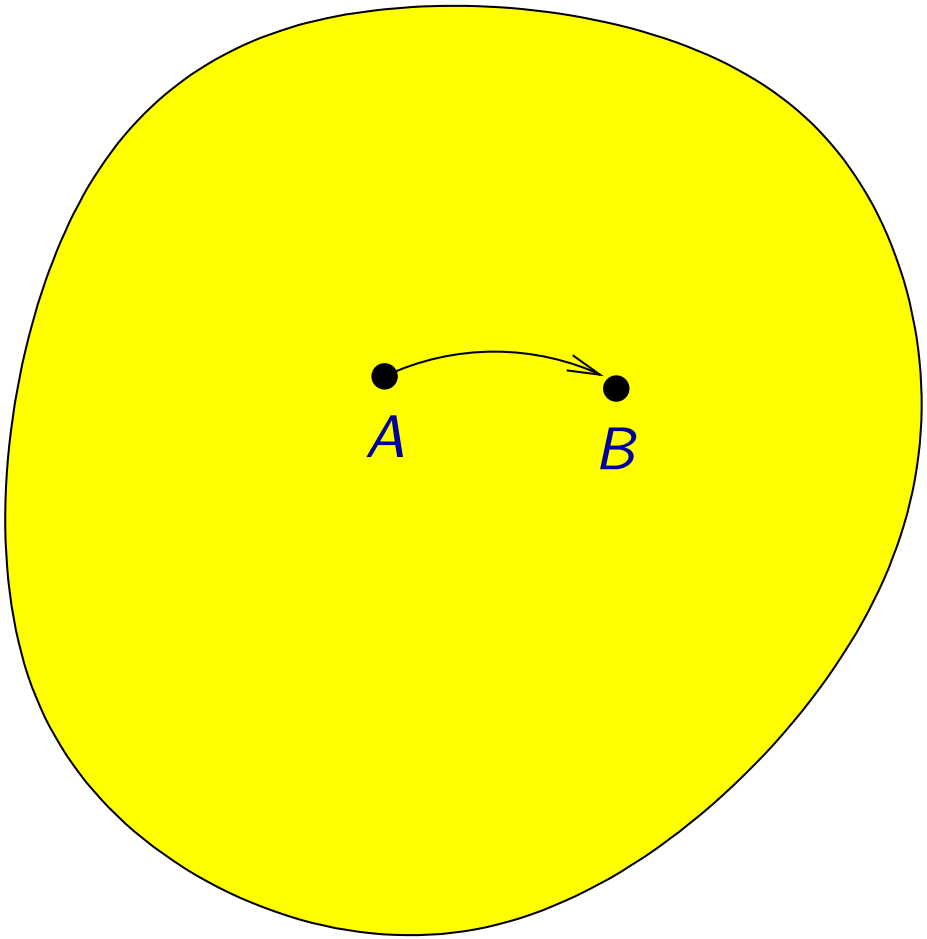
NP-úplné problémy

Vezměme si množinu všech možných rozhodovacích problémů.



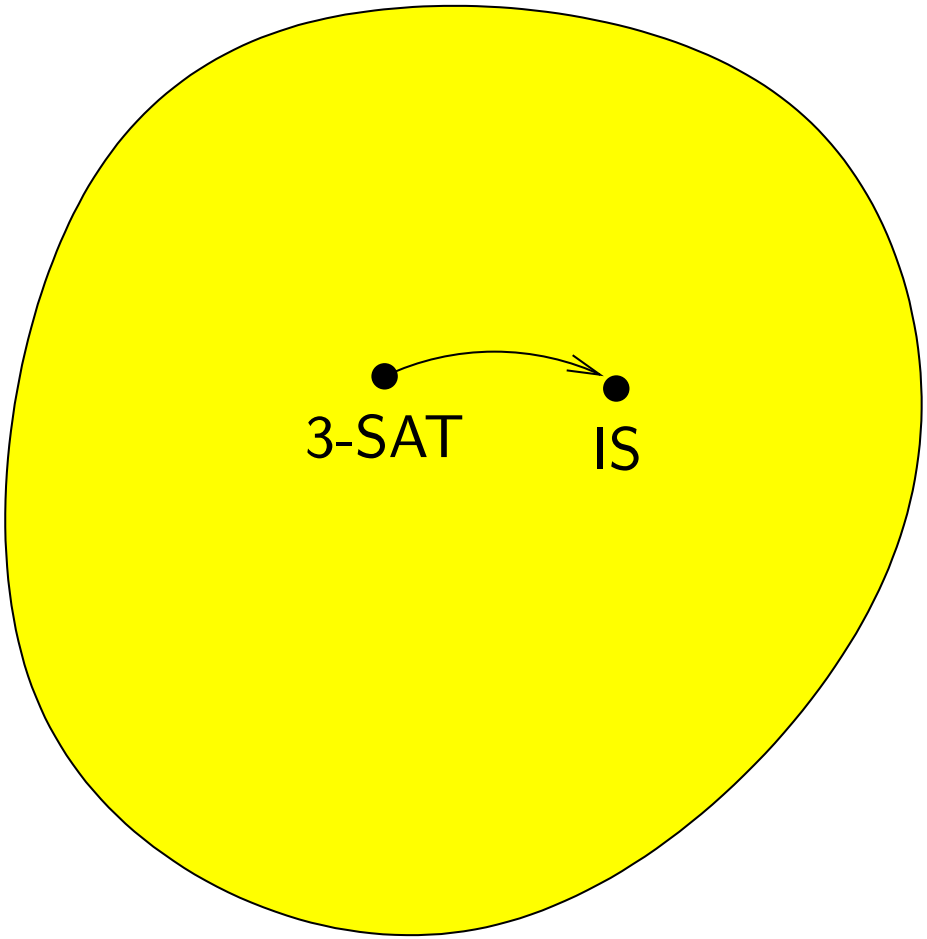
NP-úplné problémy

Šipkou si znázorníme, že problém A je polynomiálně převeditelný na problém B.



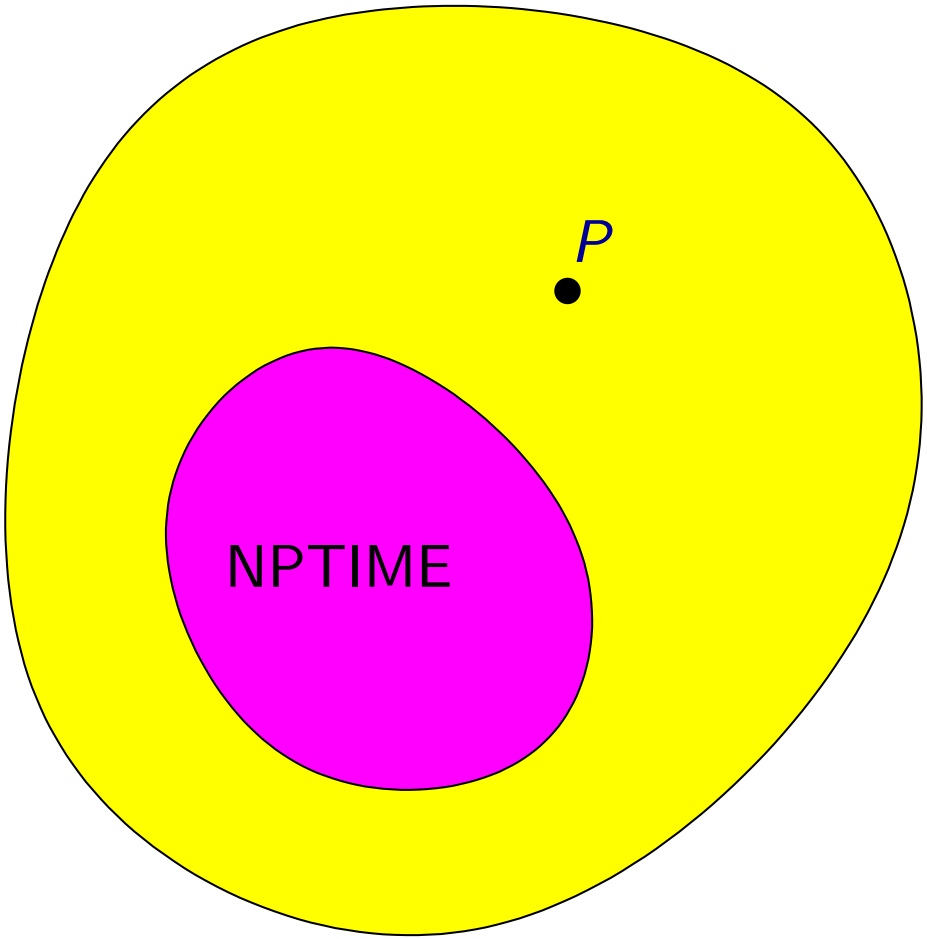
NP-úplné problémy

Například problém 3-SAT je polynomiálně převeditelný na problém IS.



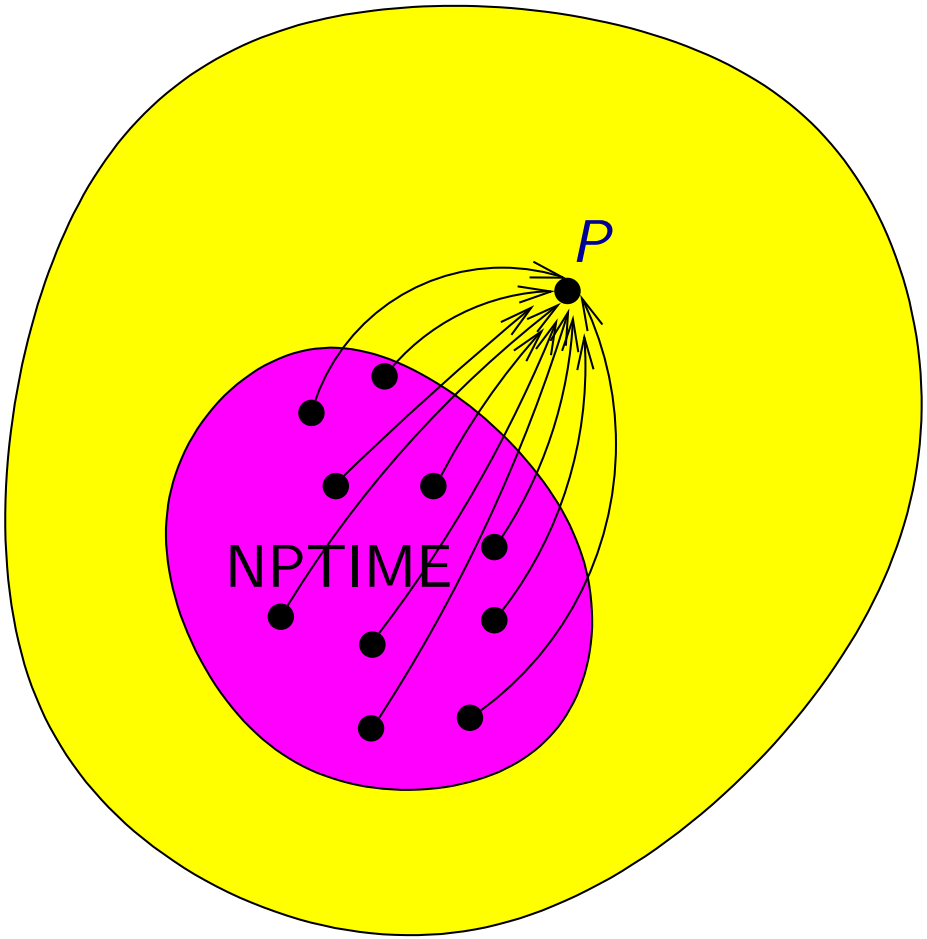
NP-úplné problémy

Vezměme si nyní třídu NPTIME a nějaký problém P.



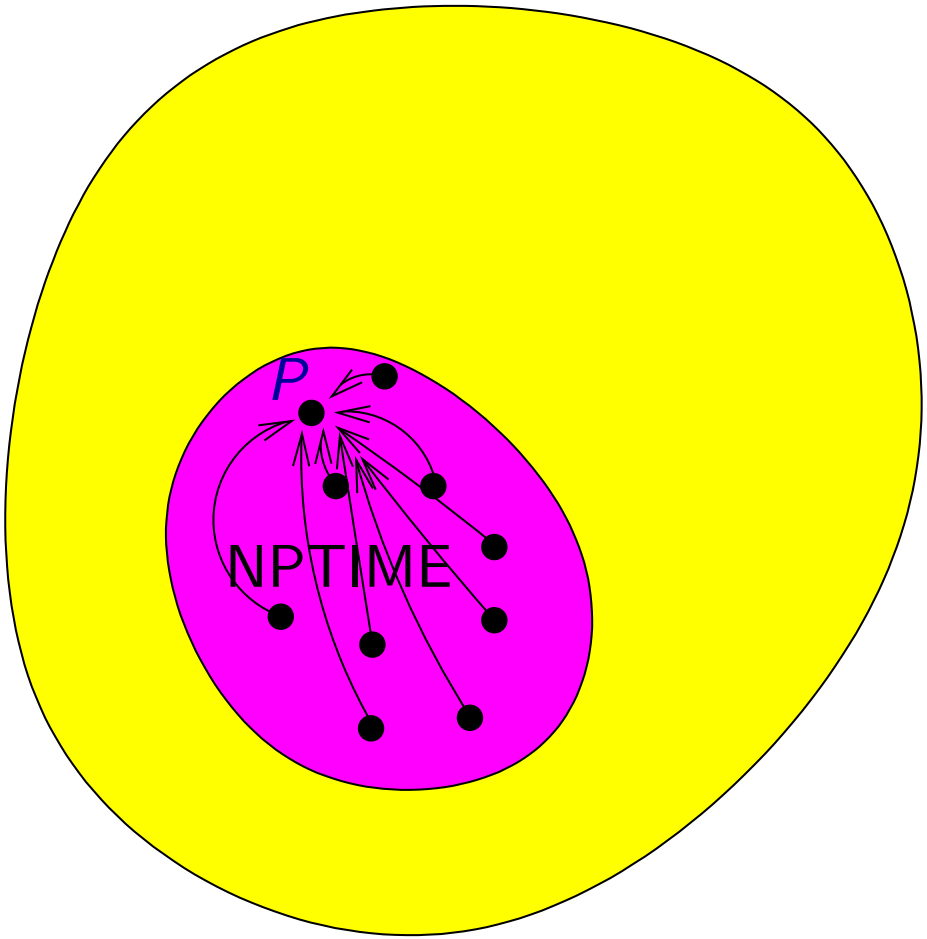
NP-úplné problémy

Problém P je NP-**těžký**, jestliže každý problém z NPTIME je polynomiálně převeditelný na P.



NP-úplné problémy

Problém P je NP-**úplný**, jestliže je NP-těžký a navíc sám patří do třídy NPTIME.



NP-úplné problémy

Pokud bychom pro nějaký NP-těžký problém P nalezli polynomiální algoritmus, získali bychom tím polynomiální algoritmus pro každý problém P' z NPTIME:

- Na vstup problému P' bychom nejprve aplikovali algoritmus realizující polynomiální převod z P' na P.
- Na vytvořenou instanci problému P bychom aplikovali polynomiální algoritmus řešící problém P a výsledek bychom vrátili jako odpověď pro danou instanci problému P'.

V takovém případě by tedy platilo PTIME $=$ NPTIME, neboť pro každý problém z NPTIME by existoval polynomiální (deterministický) algoritmus.

NP-úplné problémy

Na druhou stranu, pokud existuje alespoň jeden problém z NPTIME, pro který neexistuje polynomiální algoritmus, tak z předchozího plyne, že pro žádný NP-těžký problém nemůže existovat polynomiální algoritmus.

Zda platí první nebo druhá možnost, je otevřený problém.

NP-úplné problémy

Není těžké si rozmyslet následující:

Pokud je problém A polynomiálně převeditelný na problém B a problém B je polynomiálně převeditelný na problém C, pak problém A je polynomiálně převeditelný na problém C.

Pokud tedy o nějakém problému P víme, že je NP-těžký a že P je polynomiálně převeditelný na problém P', pak víme, že i problém P' je NP-těžký.

NP-úplné problémy

Věta

Problém SAT je NP-úplný.

Dá se ukázat, že SAT je polynomiálně převeditelný na 3-SAT a viděli jsme, že 3-SAT je polynomiálně převeditelný na IS.

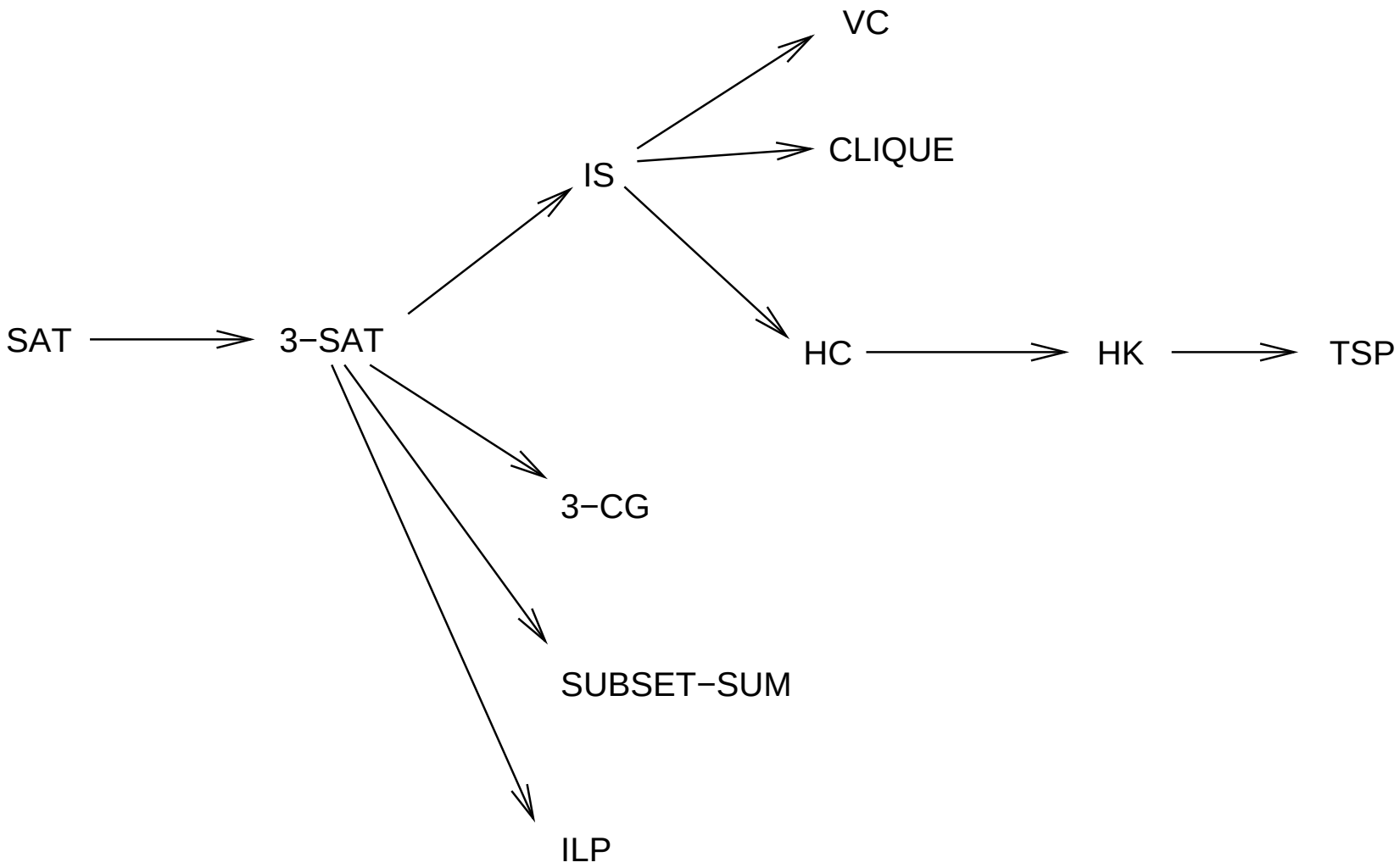
Z toho plyne, že problémy 3-SAT a IS jsou NP-těžké.

To, že 3-SAT i IS jsou v NPTIME je očividné.

Problémy 3-SAT i IS jsou NP-úplné.

NP-úplné problémy

Polynomiálními převody z již známých NP-úplných problémů se dá ukázat NP-obtížnost celé řady různých dalších problémů:



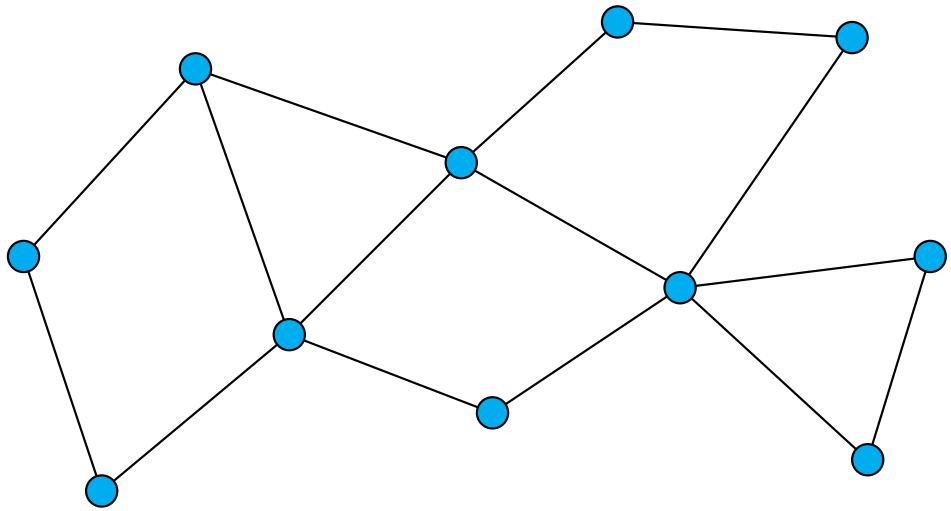
3-CG – Barvení grafu 3 barvami

3-CG - Problém „Barvení grafu 3 barvami“

Vstup: Neorientovaný graf G

Otázka: Lze vrcholy grafu G obarvit 3 barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?

Příklad:



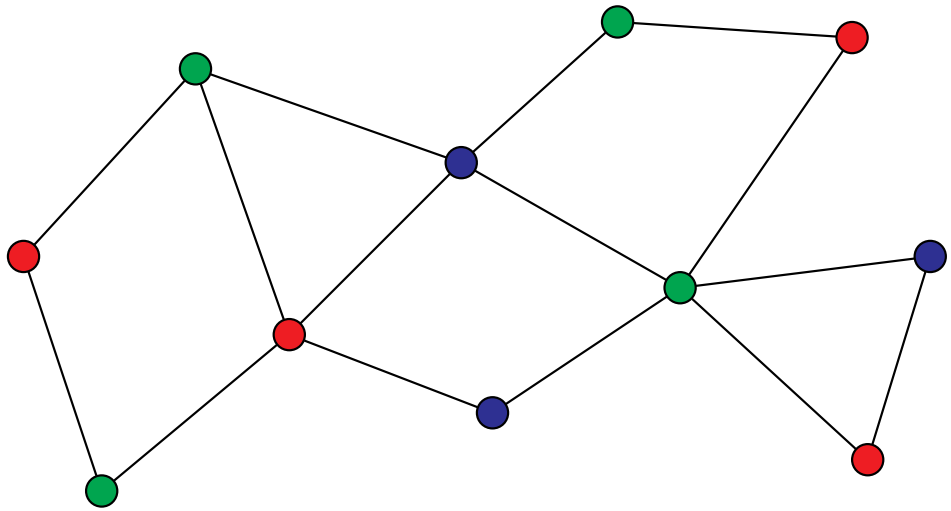
3-CG – Barvení grafu 3 barvami

3-CG - Problém „Barvení grafu 3 barvami“

Vstup: Neorientovaný graf G

Otázka: Lze vrcholy grafu G obarvit 3 barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?

Příklad:



Odpověď: Ano

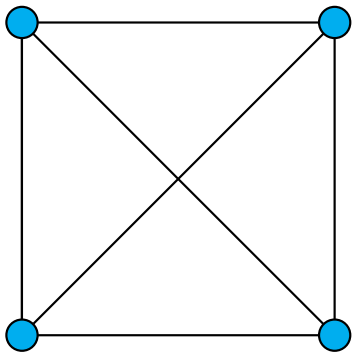
3-CG – Barvení grafu 3 barvami

3-CG - Problém „Barvení grafu 3 barvami“

Vstup: Neorientovaný graf G

Otázka: Lze vrcholy grafu G obarvit 3 barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?

Příklad:



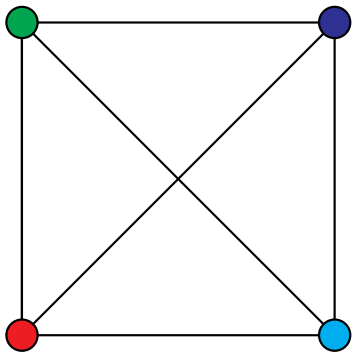
3-CG – Barvení grafu 3 barvami

3-CG - Problém „Barvení grafu 3 barvami“

Vstup: Neorientovaný graf G

Otázka: Lze vrcholy grafu G obarvit 3 barvami tak, aby žádné dva vrcholy spojené hranou neměly stejnou barvu?

Příklad:



Odpověď: NE

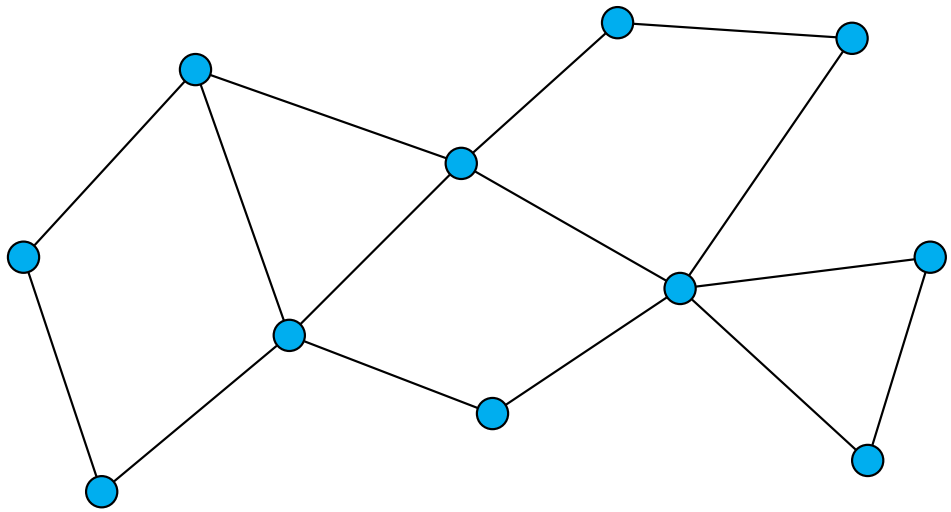
IS – Nezávislá množina

IS - Problém „Nezávislá množina“ (independent set)

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Existuje v grafu G nezávislá množina velikosti k (množina k vrcholů, které vzájemně nejsou propojeny hranou)?

Příklad: $k = 5$



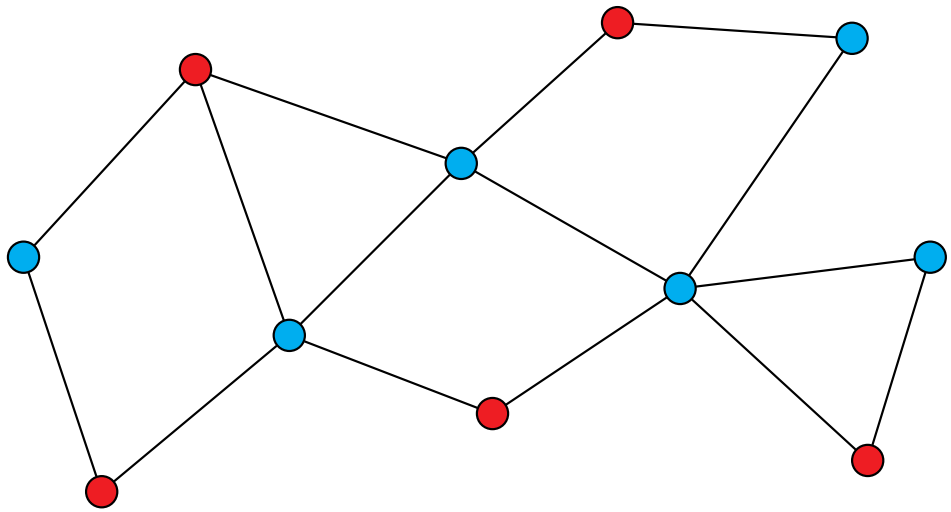
IS – Nezávislá množina

IS - Problém „Nezávislá množina“ (independent set)

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Existuje v grafu G nezávislá množina velikosti k (množina k vrcholů, které vzájemně nejsou propojeny hranou)?

Příklad: $k = 5$



Odpověď: ANO

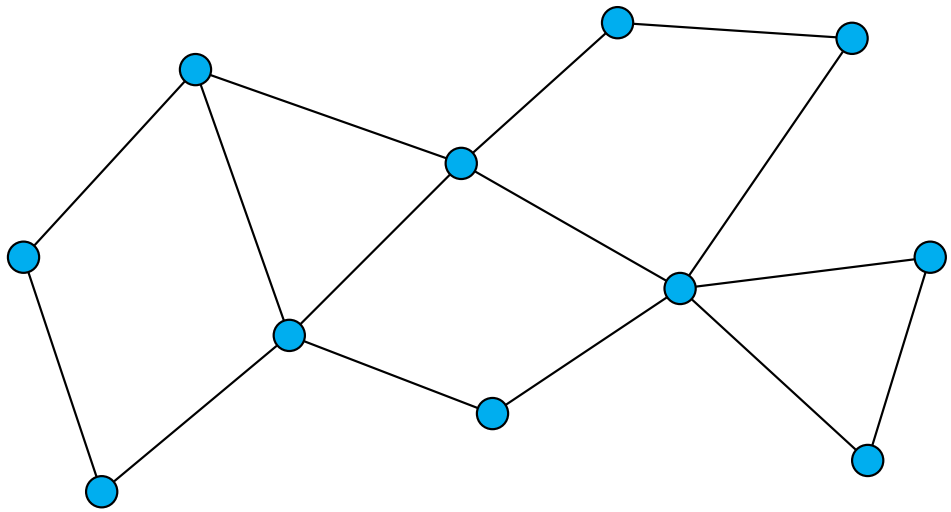
VC – Vrcholové pokrytí

VC – vrcholové pokrytí (vertex cover)

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Existuje v grafu G množina vrcholů velikosti k taková, že každá hrana má alespoň jeden svůj vrchol v této množině?

Příklad: $k = 6$



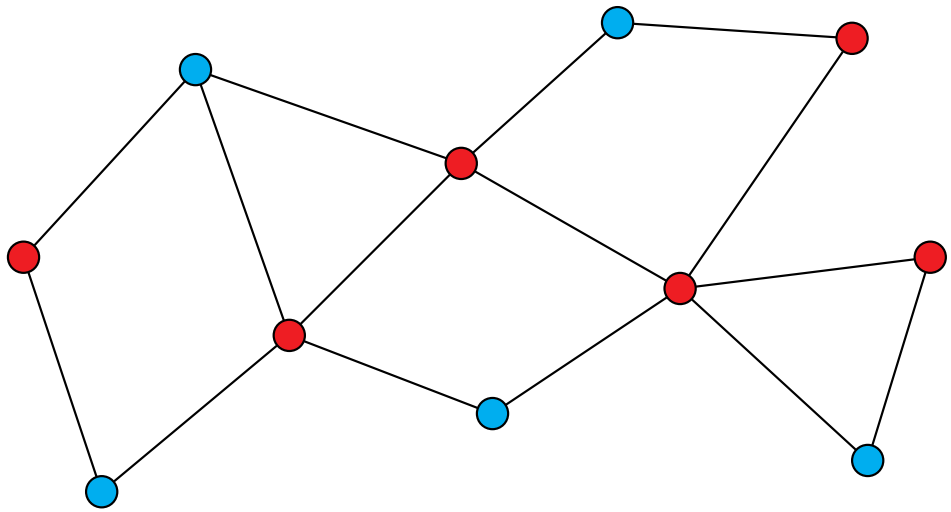
VC – Vrcholové pokrytí

VC – vrcholové pokrytí (vertex cover)

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Existuje v grafu G množina vrcholů velikosti k taková, že každá hrana má alespoň jeden svůj vrchol v této množině?

Příklad: $k = 6$



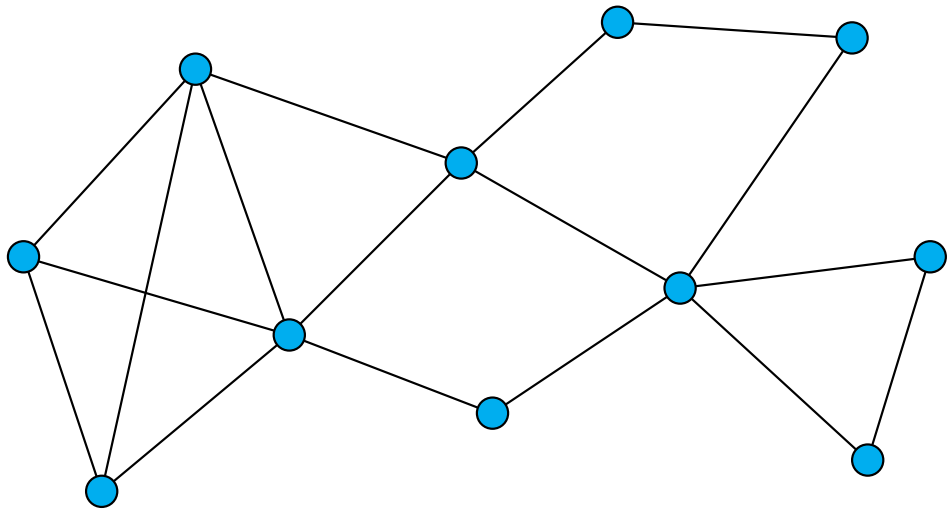
Odpověď: Ano

CLIQUE – problém kliky

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Existuje v grafu G množina vrcholů velikosti k taková, že každé dva vrcholy této množiny jsou spojeny hranou?

Příklad: $k = 4$

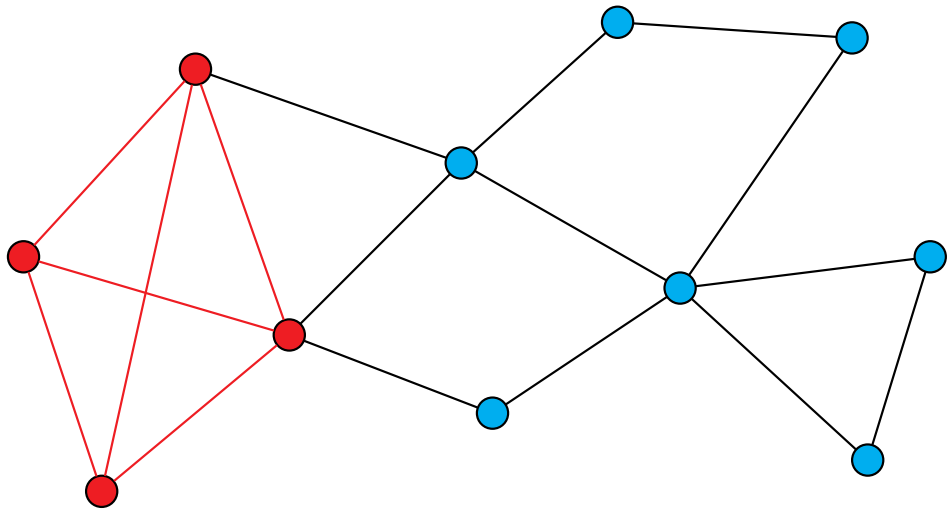


CLIQUE – problém kliky

Vstup: Neorientovaný graf G a přirozené číslo k.

Otázka: Existuje v grafu G množina vrcholů velikosti k taková, že každé dva vrcholy této množiny jsou spojeny hranou?

Příklad: $k = 4$



Odpověď: Ano

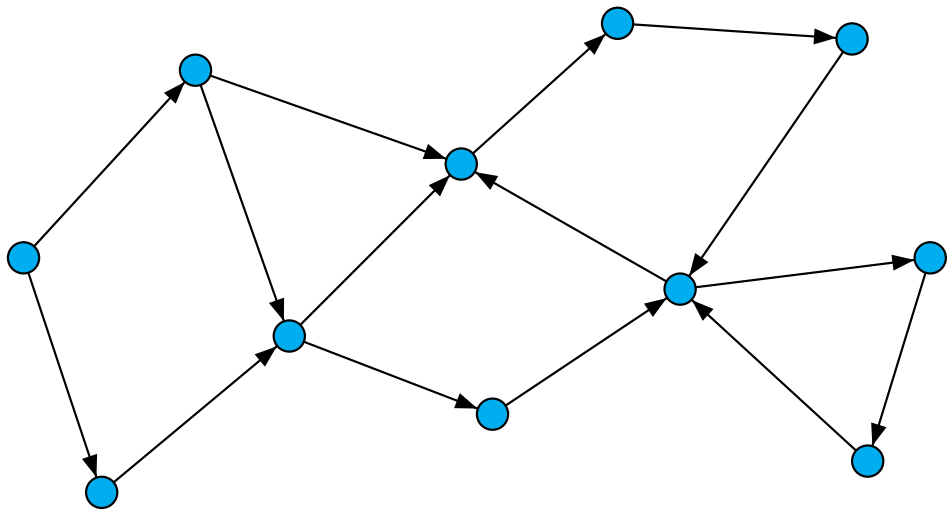
Hamiltonovský cyklus

HC – Problém „Hamiltonovský cyklus“

Vstup: Orientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovský cyklus (orientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



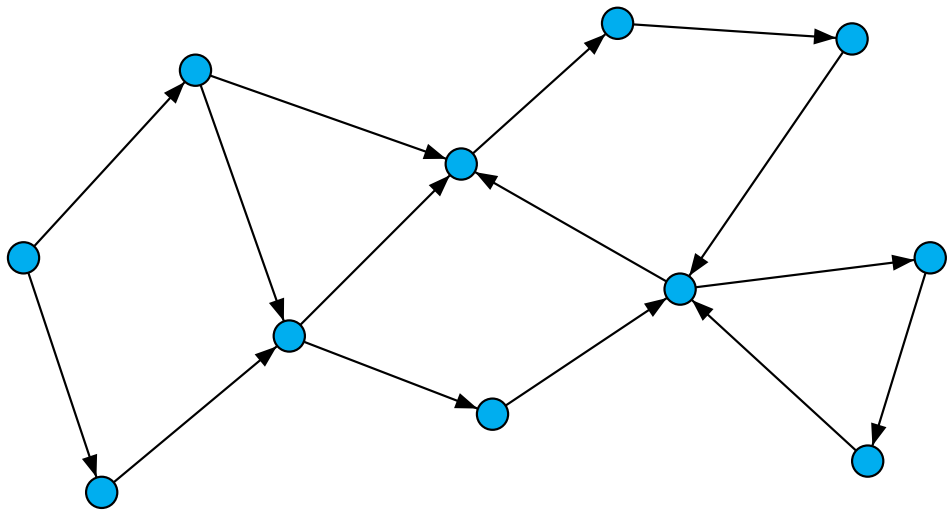
Hamiltonovský cyklus

HC – Problém „Hamiltonovský cyklus“

Vstup: Orientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovský cyklus (orientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



Odpověď: NE

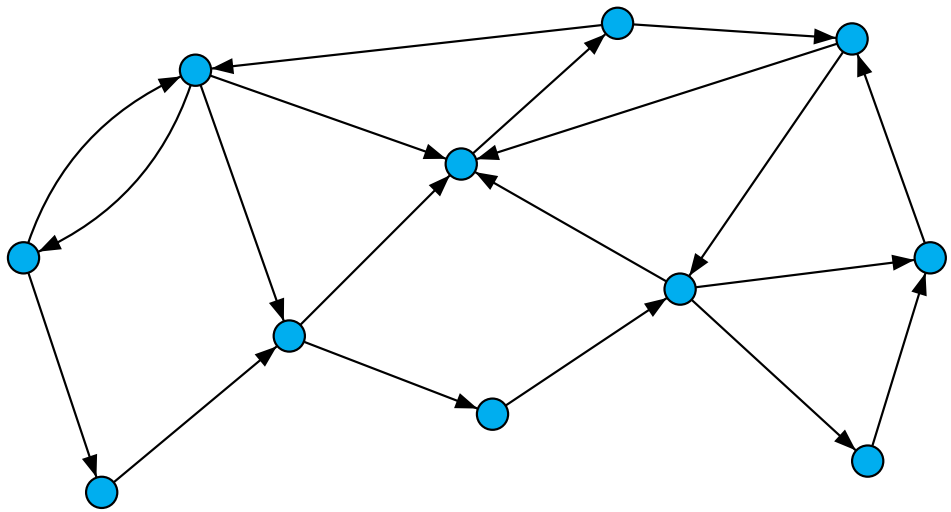
Hamiltonovský cyklus

HC – Problém „Hamiltonovský cyklus“

Vstup: Orientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovský cyklus (orientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



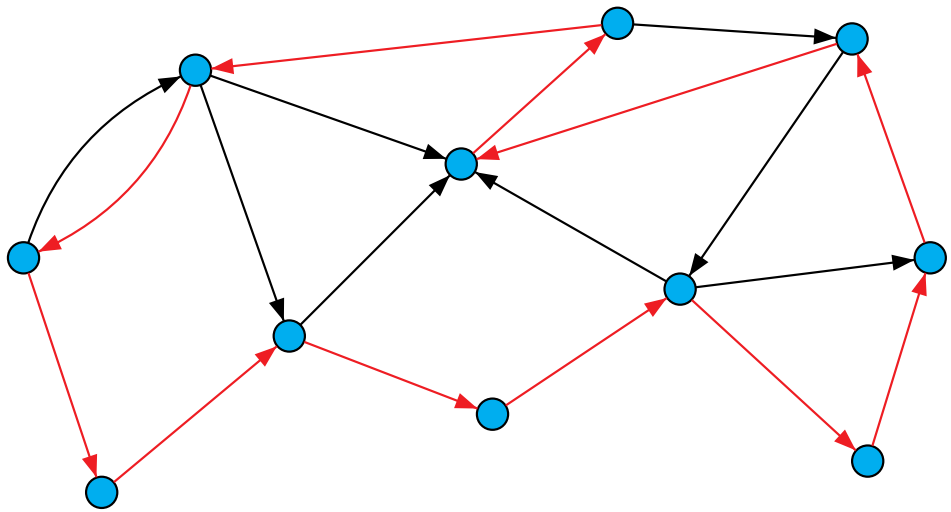
Hamiltonovský cyklus

HC – Problém „Hamiltonovský cyklus“

Vstup: Orientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovský cyklus (orientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



Odpověď: Ano

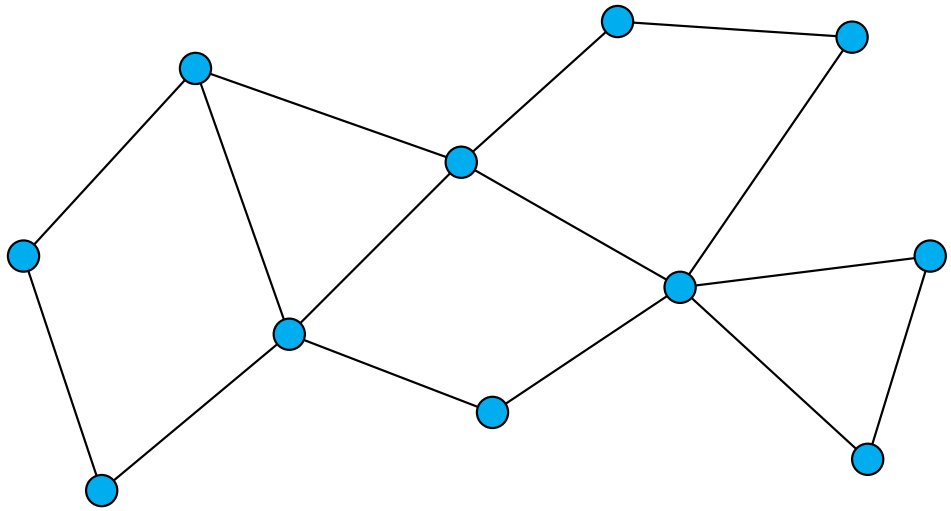
Hamiltonovská kružnice

HK – Problém „Hamiltonovská kružnice“

Vstup: Neorientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovská kružnice (neorientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



Odpověď: NE

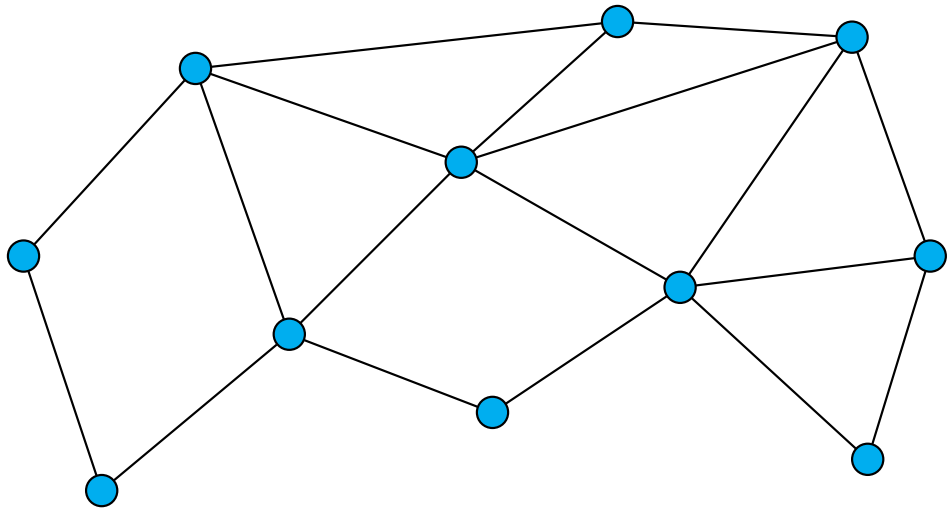
Hamiltonovská kružnice

HK – Problém „Hamiltonovská kružnice“

Vstup: Neorientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovská kružnice (neorientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



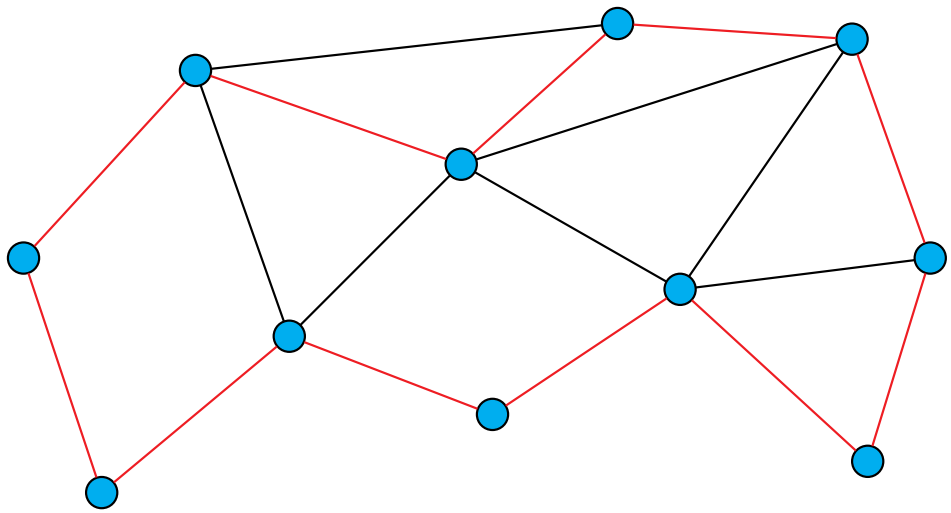
Hamiltonovská kružnice

HK – Problém „Hamiltonovská kružnice“

Vstup: Neorientovaný graf G.

Otázka: Existuje v grafu G Hamiltonovská kružnice (neorientovaný cyklus procházející každým vrcholem právě jednou)?

Příklad:



Odpověď: ANO

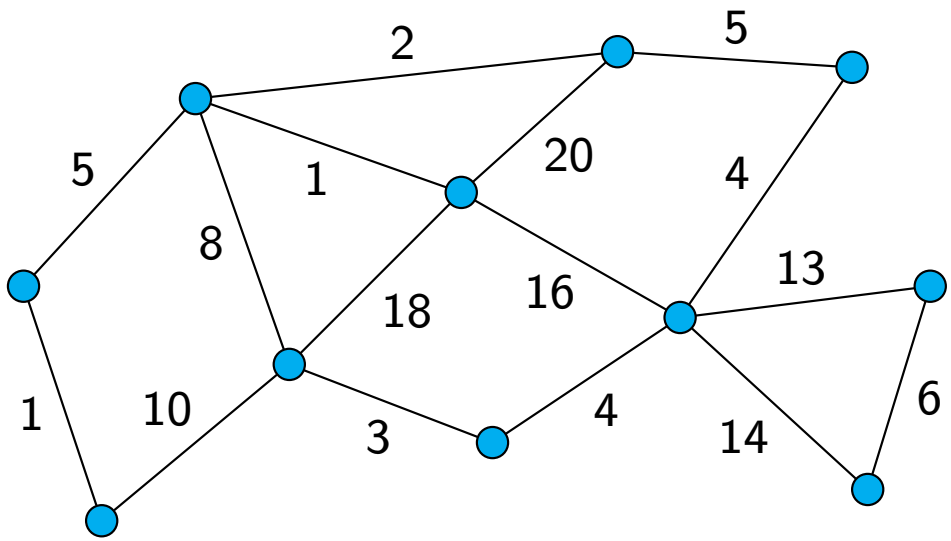
Problém obchodního cestujícího

TSP - Problém „obchodního cestujícího“

Vstup: Neorientovaný graf G s hranami ohodnocenými přirozenými čísly a číslo k.

Otázka: Existuje v grafu G uzavřený sled procházející všemi vrcholy takový, že součet délek hran na tomto sledu (včetně opakovaných) je maximálně k?

Příklad: $k = 70$



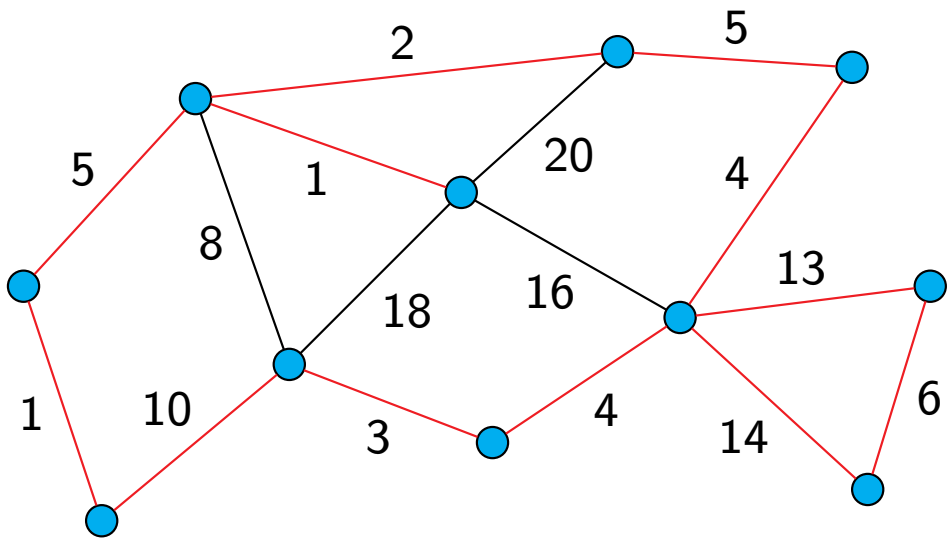
Problém obchodního cestujícího

TSP - Problém „obchodního cestujícího“

Vstup: Neorientovaný graf G s hranami ohodnocenými přirozenými čísly a číslo k.

Otázka: Existuje v grafu G uzavřený sled procházející všemi vrcholy takový, že součet délek hran na tomto sledu (včetně opakovaných) je maximálně k?

Příklad: $k = 70$



Odpověď: Ano, protože byl nalezen sled se součtem 69.

SUBSET-SUM

Problém SUBSET-SUM

Vstup: Sekvence přirozených čísel $a_1, a_2, \ldots, a_n$ a přirozené číslo s.

Otázka: Existuje množina $I \subseteq \{1, 2, \ldots, n\}$ taková, že $\sum_{i \in I} a_i = s$?

Jinak řečeno, ptáme se zda z dané (multi)množiny čísel je možné vybrat podmnožinu, jejíž součet je s.

Příklad: Pro vstup tvořený čísly $3, 5, 2, 3, 7$ a číslem $s = 15$ je odpověď Ano, neboť $3 + 5 + 7 = 15$.

Pro vstup tvořený čísly $3, 5, 2, 3, 7$ a číslem $s = 16$ je odpověď Ne, neboť žádná podmnožina těchto čísel nedává součet 16.

SUBSET-SUM

Poznámka:
Pořadí čísel $a_1, a_2, \ldots, a_n$ na vstupu není důležité.

Všimněte si však určitého rozdílu oproti tomu, kdybychom problém formulovali tak, že vstupem je množina $\{a_1, a_2, \ldots, a_n\}$ a číslo s:

V množině se čísla neopakují, zatímco v sekvenci se může totéž číslo vyskytnout vícekrát.

SUBSET-SUM

Problém SUBSET-SUM je speciálním případem **problému batohu** (knapsack problem):

Knapsack problem

Vstup: Sekvence dvojic přirozených čísel $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$ a dvě přirozená čísla s a t.

Otázka: Existuje množina $I \subseteq \{1, 2, \ldots, n\}$ taková, že $\sum_{i \in I} a_i \leq s$ a $\sum_{i \in I} b_i \geq t$?

SUBSET-SUM

Neformálně můžeme problém batohu formulovat takto:

Máme n předmětů, kde i-tý předmět váží a_i gramů a má cenu b_i Kč. Do batohu se vejdou předměty o maximální celkové váze s gramů.

Otázka zní, zda můžeme z předmětů vybrat podmnožinu, která by vážila maximálně s gramů a měla celkovou cenu alespoň t Kč.

Poznámka:
Zde jsme problém batohu formulovali jako rozhodovací problém.

Běžnější je formulovat tento problém jako optimalizační problém, kde je cílem najít takovou množinu $I \subseteq \{1, 2, \ldots, n\}$, kde hodnota $\sum_{i \in I} b_i$ je maximální, přičemž ovšem musí být dodržena podmínka $\sum_{i \in I} a_i \leq s$, tj. vybrat předměty s maximální celkovou cenou tak, aby nebyla překročena kapacita batohu.

SUBSET-SUM

To, že SUBSET-SUM je speciálním případem problému batohu, vidíme z následující (téměř triviální) redukce:

Řekněme, že $a_1, a_2, \ldots, a_n$, s_1 je instance problému SUBSET-SUM. Je očividné, že pro instanci problému batohu, kde máme sekvenci $(a_1, a_1), (a_2, a_2), \ldots, (a_n, a_n)$, $s = s_1$ a $t = s_1$, je odpověď stejná jako pro původní instanci SUBSET-SUM.

SUBSET-SUM

Pokud chceme studovat složitost problémů jako jsou SUBSET-SUM nebo problém batohu, je dobré si nejprve ujasnit, co považujeme za velikost vstupu.

Asi nejpřirozenější je definovat velikost vstupu jako celkový počet bitů, který potřebujeme k zápisu instance.

Musíme však určit, jakým způsobem jsou na vstupu zadána přirozená čísla – zda binárně (případně v jiné číselné soustavě o základu alespoň 2, např. desítkové nebo šestnáctkové) nebo unárně.

Pokud uvažujeme, že čísla jsou na vstupu zadána **binárně**, tj. že velikost vstupu je úměrná součtu délek binárních zápisů jednotlivých čísel na vstupu, tak problém SUBSET-SUM je NP-těžký.
(Dá se ukázat polynomiální převod z 3-SAT.)

SUBSET-SUM

Není těžké se přesvědčit, že SUBSET-SUM i problém batohu (jeho rozhodovací varianta) patří do třídy NPTIME:

- Nedeterministický algoritmus nejprve nedeterministicky zvolí podmnožinu prvků sekvence na vstupu a poté (deterministicky) ověří, zda splňuje splňuje danou podmínku (resp. podmínky).

  Je zřejmé, že toto ověření je možné provést v čase polynomiálním vzhledem k velikosti instance.

Problémy SUBSET-SUM i problém batohu jsou tedy NP-úplné.

ILP – celočíselné lineární programování

Problém ILP (celočíselné lineární programování)

Vstup: Celočíselná matice A a celočíselný vektor b.

Otázka: Existuje celočíselný vektor x, takový že $Ax \leq b$?

Příklad instance problému:

$$A = \begin{pmatrix} 3 & -2 & 5 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \qquad b = \begin{pmatrix} 8 \\ -3 \\ 5 \end{pmatrix}$$

Ptáme se tedy, zda existuje celočíselné řešení následující soustavy nerovnic:

$$\begin{aligned} 3x_1 - 2x_2 + 5x_3 &\leq 8 \\ x_1 + x_3 &\leq -3 \\ 2x_1 + x_2 &\leq 5 \end{aligned}$$

ILP – celočíselné lineární programování

Jedním z řešení soustavy

$$\begin{aligned} 3x_1 - 2x_2 + 5x_3 &\leq 8 \\ x_1 + x_3 &\leq -3 \\ 2x_1 + x_2 &\leq 5 \end{aligned}$$

je například $x_1 = -4$, $x_2 = 1$, $x_3 = 1$, tj.

$$x = \begin{pmatrix} -4 \\ 1 \\ 1 \end{pmatrix}$$

neboť

$$\begin{aligned} 3 \cdot (-4) - 2 \cdot 1 + 5 \cdot 1 &= -9 \leq 8 \\ -4 + 1 &= -3 \leq -3 \\ 2 \cdot (-4) + 1 &= -7 \leq 5 \end{aligned}$$

Pro tuto instanci je tedy odpověď Ano.

ILP – celočíselné lineární programování

Poznámka: Analogický problém, kdy se pro danou soustavu lineálních nerovnic ptáme, zda existuje její řešení v oboru **reálných** čísel, je možné řešit v polynomiálním čase.

P vs. NP

Jak už bylo zmíněno, otázka, zda $\mathrm{PTIME} = \mathrm{NPTIME}$, je dlouhodobě otevřený problém.

Obecně se má za to, že pravděpodobně $\mathrm{PTIME} \neq \mathrm{NPTIME}$, dosud to však nikdo nedokázal.

Poznámka: Například se dá ukázat, že $\mathrm{PTIME} \neq \mathrm{NPTIME}$ je nutnou (nikoli však postačující) podmínkou pro to, aby vůbec mohly existovat šifrovací algoritmy, které by nebyly snadno prolomitelné.

Pokud by se podařilo někomu najít polynomiální algoritmus pro alespoň jeden NP-úplný problém, okamžitě bychom tím získali algoritmy pro rychlé prolomení všech v současné době používaných šifer.

Nerozhodnutelné problémy

Algoritmicky nerozhodnutelné problémy

Problém, který není algoritmicky řešitelný je **algoritmicky neřešitelný**.

Rozhodovací problém, který není rozhodnutelný je **nerozhodnutelný**.

Příklad: Následující problém zvaný **Problém zastavení (Halting problem)** je nerozhodnutelný:

Halting problem

Vstup: Popis Turingova stroje M a slovo w.

Otázka: Zastaví se stroj M po nějakém konečném počtu kroků, pokud dostane jako svůj vstup slovo w?

Halting Problem

Alternativně bychom ho mohli formulovat třeba takto:

Halting problem

Vstup: Zdrojový kód programu P v jazyce C, vstupní data x.

Otázka: Zastaví se program P po nějakém konečném počtu kroků, pokud dostane jako vstup data x?

Halting Problem

Předpokládejme, že by existoval nějaký program, který by rozhodoval Halting problem.

Mohli bychom tedy vytvořit podprogram H, deklarovaný jako

boolean H(String kod, String vstup)

kde $H(P, x)$ vrátí:

- true pokud se program P zastaví pro vstup x,
- false pokud se program P nezastaví pro vstup x.

Poznámka: Řekněme, že podprogram $H(P, x)$ by vracel false v případě, že P není syntakticky správný kód programu.

Halting Problem

S použitím podprogramu H bychom vytvořili program D, který bude provádět následující kroky:

- Načte svůj vstup do proměnné x typu String.
- Zavolá podprogram $H(x, x)$.
- Pokud podprogram H vrátil true, skočí do nekonečné smyčky

loop: goto loop

V případě, že H vrátil false, program D se ukončí.

Co udělá program D, pokud mu předložíme jako vstup jeho vlastní kód?

Halting Problem

Pokud D dostane jako vstup svůj vlastní kód, tak se buď zastaví nebo nezastaví.

- Pokud se D zastaví, tak $H(D, D)$ vrátí true a D skočí do nekonečné smyčky. Spor!
- Pokud se D nezastaví, tak $H(D, D)$ vrátí false a D se zastaví. Spor!

V obou případech dospějeme ke sporu a další možnost není. Nemůže tedy platit předpoklad, že H řeší Halting problem.

Další nerozhodnutelné problémy

S jedním příkladem nerozhodnutelného problému už jsme se setkali:

Problém

Vstup: Bezkontextové gramatiky G_1 a G_2.

Otázka: Je $L(G_1) = L(G_2)$?

případně

Problém

Vstup: Bezkontextová gramatika G generující jazyk nad abecedou Σ.

Otázka: Je $L(G) = \Sigma^*$?

Redukce mezi problémy

Pokud máme o nějakém (rozhodovacím) problému dokázáno, že je nerozhodnutelný, můžeme ukázat nerozhodnutelnost dalších problémů pomocí redukcí.

Řekněme, že A a B jsou rozhodovací problémy.
Redukce problému A na problém B je algoritmus P, který:

- Dostane jako vstup instanci problému A (označme ji x).
- Jako svůj výstup (označme jej $P(x)$) vyprodukuje instanci problému B.
- Pro každou instanci x problému A platí, že pro vstup x je v problému A odpověď ANO právě tehdy, když pro vstup $P(x)$ je v problému B odpověď ANO.

Redukce mezi problémy

Řekněme, že existuje redukce P problému A na problém B.

Pokud by problém B byl rozhodnutelný, pak i problém A je rozhodnutelný.

Řešení problému A pro vstup x:

- Zavoláme P se vstupem x, vrátí nám hodnotu $P(x)$.
- Zavoláme algoritmus řešící problém B se vstupem $P(x)$.
- Hodnotu, kterou nám vrátí vypíšeme jako výsledek.

Je zřejmé, že pokud A je nerozhodnutelný, tak B nemůže být rozhodnutelný.

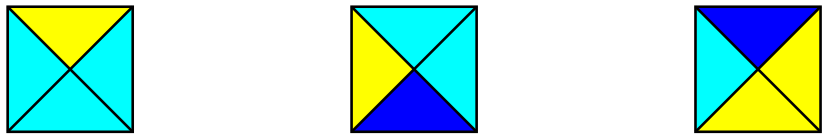
Další nerozhodnutelné problémy

Redukcí z Halting problému se dá ukázat nerozhodnutelnost celé řady problémů, které se týkají ověřování chování programů:

- Vydá daný program pro nějaký vstup odpověď ANO?
- Zastaví se daný program pro libovolný vstup?
- Dávají dva dané programy pro stejné vstupy stejný výstup?
- ...

Další nerozhodnutelné problémy

Vstupem je množina typů kachliček, jako třeba:

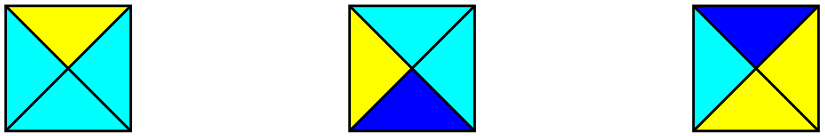


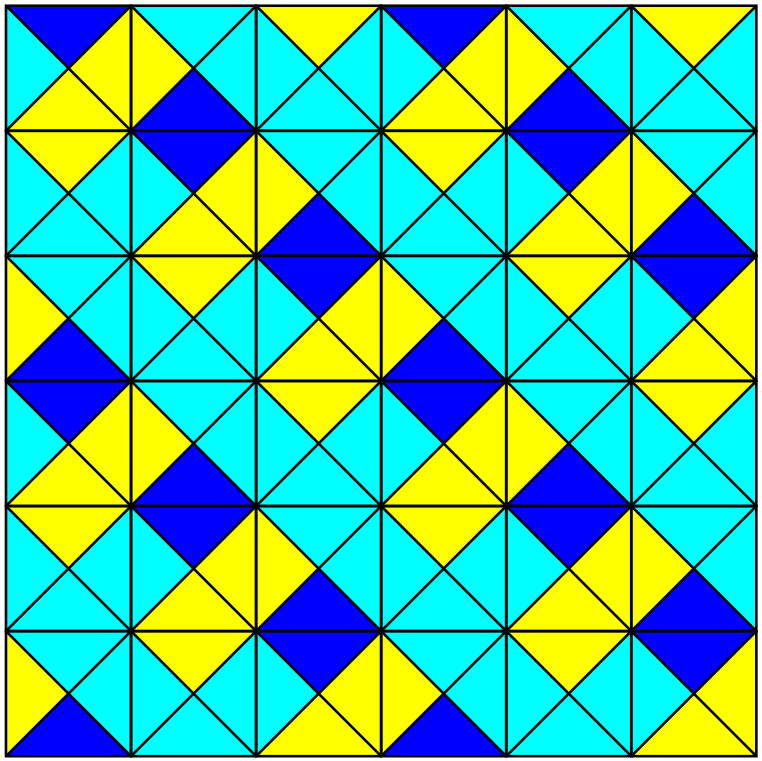
Otázka je, zda je možné použitím daných typů kachliček pokrýt každou libovolně velkou konečnou plochu tak, aby všechny kachličky spolu sousedily stejnými barvami.

Poznámka: Můžeme předpokládat, že máme v zásobě neomezené množství kachliček všech typů.

Kachličky není dovoleno otáčet.

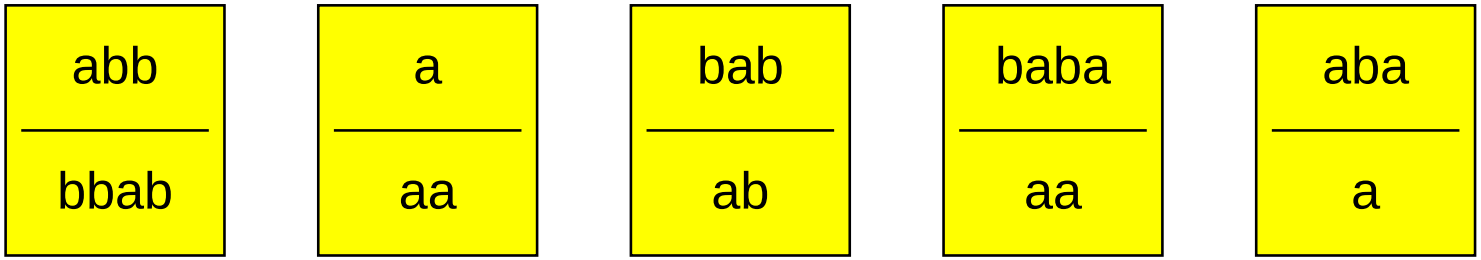
Další nerozhodnutelné problémy



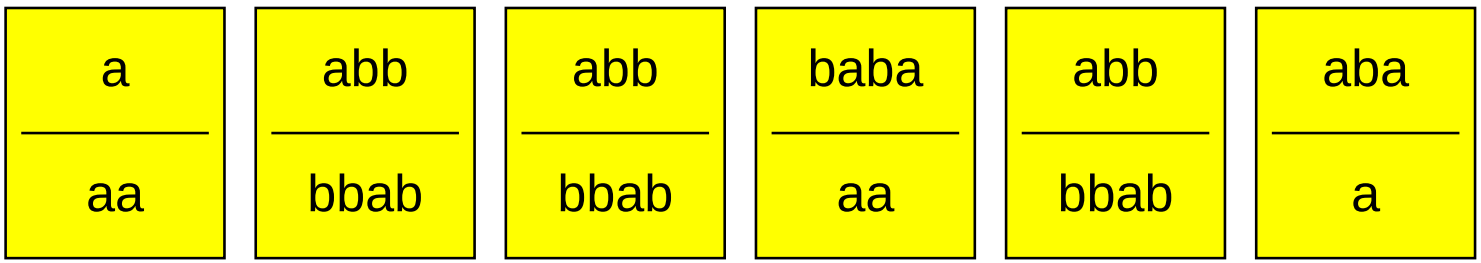


Další nerozhodnutelné problémy

Vstupem je množina typů kartiček, jako třeba:



Otázka je, zda je možné z těchto typů kartiček vytvořit neprázdnou konečnou posloupnost, kde zřetězením slov nahoře i dole vznikne totéž slovo. Každý typ kartičky je možné používat opakovaně.



Nahoře i dole vznikne slovo aabbabbbabaabbaba.

Další nerozhodnutelné problémy

Redukcí z předchozího problému se dá snadno ukázat nerozhodnutelnost některých dalších problémů z oblasti bezkontextových gramatik:

Problém

Vstup: Bezkontextové gramatiky G_1 a G_2.

Otázka: Je $L(G_1) \cap L(G_2) = \emptyset$?

Problém

Vstup: Bezkontextová gramatika G.

Otázka: Je G nejednoznačná?

Další nerozhodnutelné problémy

Problém

Vstup: Uzavřená formule PL1, ve které mohou být použity jako funkční symboly $+$ a $*$ a celočíselné konstanty a jako predikátové symboly $=$ a $<$.

Otázka: Je daná formule pravdivá v oboru přirozených čísel (při přirozené interpretaci všech funkčních a predikátových symbolů)?

Příklad vstupu:

$$\forall x \exists y \forall z((x * y = z) \wedge (y + 5 = x))$$

Poznámka: Úzce souvisí s Gödelovou větou o neúplnosti.

Další nerozhodnutelné problémy

Je zajímavé, že analogický problém, kde ale místo přirozených čísel uvažujeme čísla reálná, je algoritmicky rozhodnutelný (i když popis daného algoritmu a důkaz jeho korektnosti jsou značně netriviální).

Rovněž pokud uvažujeme přirozená nebo celá čísla a stejné formule jako v předchozím případě, ale s tím, že v nich nesmí být použit funkční symbol $*$ (násobení), tak je problém algoritmicky rozhodnutelný.

Další nerozhodnutelné problémy

Pokud můžeme používat $*$, je ve skutečnosti je nerozhodnutelný už velmi omezený případ:

Desátý Hilbertův problém

Vstup: Polynom $f(x_1, x_2, \ldots, x_n)$ vytvořený z proměnných $x_1, x_2, \ldots, x_n$ a celočíselných konstant.

Otázka: Existují přirozená čísla $x_1, x_2, \ldots, x_n$ taková, že $f(x_1, x_2, \ldots, x_n) = 0$?

Příklad vstupu: $5x^2y - 8yz + 3z^2 - 15$

Tj. ptáme se, zda

$$\exists x \exists y \exists z (5 * x * x * y + (-8) * y * z + 3 * z * z + (-15) = 0)$$

platí v oboru přirozených čísel.

Další nerozhodnutelné problémy

Také následující problém je algoritmicky nerozhodnutelný:

Problém

Vstup: Uzavřená formule φ PL1.

Otázka: Platí $\models \varphi$?

Poznámka: Zápis $\models \varphi$ znamená, že formule φ je logicky platná, tj. pravdivá v každé interpretaci.

Částečně rozhodnutelné problémy

Problém je **částečně rozhodnutelný**, jestliže existuje algoritmus, který:

- Pokud dostane jako vstup instanci, pro kterou je odpověď ANO, tak se po konečném počtu kroků zastaví a vypíše `"ANO"`.
- Pokud dostane jako vstup instanci, pro kterou je odpověď NE, tak se buď zastaví a vypíše `"NE"` nebo se nikdy nezastaví.

Je očividné, že například HP (Halting problem) je částečně rozhodnutelný.

Některé problémy však nejsou ani částečně rozhodnutelné.