Rezoluční metoda

Rezoluční metoda je jedním z algoritmů pro zjištění toho, zda daný závěr vyplývá z daných předpokladů.

Řeší následující problém:

Vstup: Formule $\varphi_1, \varphi_2, \ldots, \varphi_n, \psi$.

Otázka: Platí $\varphi_1, \varphi_2, \ldots, \varphi_n \models \psi$?

Poznámka: Dá se použít také pro zjištění, zda je daná formule tautologie, kontradikce nebo splnitelná.

Na použití různých variant rezoluční metody jsou založeny některé systémy pro automatické dokazování a také logický programovací jazyk Prolog.

Rezoluční metoda

- Pracuje s formulemi v KNF.
- Vytváří důkaz toho, že daný závěr plyne z předpokladů.
- Jedná se o důkaz sporem — postupně generuje formule, které vyplývají z předpokladů

$$\varphi_1, \varphi_2, \dots, \varphi_n, \neg\psi$$

- Výpočet může skončit dvěma různými způsoby:
  - Podaří se najít spor, tj. podaří se odvodit formuli $\bot$ — pak platí, že závěr ψ logicky vyplývá z předpokladů $\varphi_1, \varphi_2, \dots, \varphi_n$.
  - Nepodaří se odvodit $\bot$ a žádné další nové formule se nedají přidat — pak závěr ψ z daných předpokladů nevyplývá.

Rezoluční metoda

- Rezoluční metoda pracuje s formulemi, které mají podobu elementárních disjukcí, např.

$$(\neg p \vee q \vee \neg s \vee \neg t)$$

Těmto formulím se říká **klauzule**.

- Speciálním případem klauzule je **prázdná klauzule** $\bot$, která představuje nalezený spor.

- Algorimus začne svou činnost tím, že formule

$$\varphi_1, \varphi_2, \ldots, \varphi_n, \neg\psi$$

převede do KNF a vezme všechny klazule z takto vytvořených formulí jako počáteční množinu předpokladů

$$\chi_1, \chi_2, \ldots, \chi_m.$$

Rezoluční metoda

Při generování dalších klauzulí ke dříve přidaným klauzulím se používá tzv. **rezoluční pravidlo** (či **rezoluční princip**):

Pro libovolné formule φ, ψ a χ platí

$$\varphi \vee \psi, \neg\varphi \vee \chi \models \psi \vee \chi$$

V rezoluční metodě se tento princip používá jen pro klauzule.
V případě rezoluční metody jsou $\varphi \vee \psi$, $\neg\varphi \vee \chi$ a $\psi \vee \chi$ vždy klauzule a φ je navíc atomický výrok.

Příklad: Z klauzulí

$$p \vee \neg q \vee r \vee s \qquad \text{a} \qquad \neg r \vee t \vee \neg u$$

je možno vyvodit pomocí rezolučního pravidla klauzuli $p \vee \neg q \vee s \vee t \vee \neg u$.

Poznámky:

- Pořadí literálů v klauzulích není podstatné.
- Vícenásobné výskyty stejných literálů v téže klauzuli je možno odstranit.
- Pokud je nově vygenerovaná klauzule stejná, jako nějaká dříve přidaná klauzule (a liší se nanejvýš pořadím literálů), nemá smysl ji přidávat.
- Klauzule, které obsahují zároveň literály p a $\neg p$ jsou ekvivalentní $\top$ a je možné je odstranit.
- Klauzule je možno používat pro aplikaci rezolučního pravidla opakovaně (s jinými klauzulemi).

Speciální případy použití rezolučního pravidla:

- Jedna z klauzulí obsahuje jen jeden literál a druhá více než jeden literál:

  Z klauzulí

  $$\neg q \qquad\qquad p \vee q \vee \neg t$$

  je možno vyvodit klauzuli $p \vee \neg t$.

- Obě klauzule obsahují jeden literál:

  Z klauzulí

  $$p \qquad\qquad \neg p$$

  je možno vyvodit prázdnou klauzuli $\perp$, tj. spor.

Chceme ověřit platnost následujícího úsudku:

- *Není pravda, že Jana je ve škole a Petr není doma.*
- *Jana není ve škole nebo je všední den nebo prší.*
- *Jestliže je všední den, pak Petr není doma.*

- *Jestliže je Jana ve škole, pak prší.*

Jednotlivá tvrzení nejprve zformalizujeme pomocí formulí výrokové logiky:

$$\begin{array}{l} \neg(j \wedge \neg p) \\ \neg j \vee d \vee r \\ d \rightarrow \neg p \\ \hline j \rightarrow r \end{array}$$

j – Jana je škole
p – Petr je doma
d – je všední den
r – prší

Rezoluční metoda

$$\begin{array}{l} \neg(j \wedge \neg p) \\ \neg j \vee d \vee r \\ d \rightarrow \neg p \\ \hline j \rightarrow r \end{array}$$

Jednotlivé předpoklady převedeme do KNF:

- $\neg(j \wedge \neg p) \;\Leftrightarrow\; \neg j \vee p$
- $\neg j \vee d \vee r$
- $d \rightarrow \neg p \;\Leftrightarrow\; \neg d \vee \neg p$

Závěr znegujeme a převedeme do KNF:

- $\neg(j \rightarrow r) \;\Leftrightarrow\; j \wedge \neg r$

Rezoluční metoda

Sepíšeme si jednotlivé klauzule:

1.	$\neg j \vee p$	– předpoklad 1
2.	$\neg j \vee d \vee r$	– předpoklad 2
3.	$\neg d \vee \neg p$	– přespoklad 3
4.	j	– 1. klauzule z negovaného závěru
5.	$\neg r$	– 2. klauzule z negovaného závěru

Rezoluční metoda

Sepíšeme si jednotlivé klauzule:

1.	$\neg j \vee p$	– předpoklad 1
2.	$\neg j \vee d \vee r$	– předpoklad 2
3.	$\neg d \vee \neg p$	– přespoklad 3
4.	j	– 1. klauzule z negovaného závěru
5.	$\neg r$	– 2. klauzule z negovaného závěru
6.	p	– rezoluce: 1,4

Rezoluční metoda

Sepíšeme si jednotlivé klauzule:

1.	$\neg j \vee p$	– předpoklad 1
2.	$\neg j \vee d \vee r$	– předpoklad 2
3.	$\neg d \vee \neg p$	– přespoklad 3
4.	j	– 1. klauzule z negovaného závěru
5.	$\neg r$	– 2. klauzule z negovaného závěru
6.	p	– rezoluce: 1,4
7.	$d \vee r$	– rezoluce: 2,4

Sepíšeme si jednotlivé klauzule:

1.	$\neg j \vee p$	– předpoklad 1
2.	$\neg j \vee d \vee r$	– předpoklad 2
3.	$\neg d \vee \neg p$	– přespoklad 3
4.	j	– 1. klauzule z negovaného závěru
5.	$\neg r$	– 2. klauzule z negovaného závěru
6.	p	– rezoluce: 1,4
7.	$d \vee r$	– rezoluce: 2,4
8.	$\neg d$	– rezoluce: 3,6

Rezoluční metoda

Sepíšeme si jednotlivé klauzule:

1.	$\neg j \vee p$	– předpoklad 1
2.	$\neg j \vee d \vee r$	– předpoklad 2
3.	$\neg d \vee \neg p$	– přespoklad 3
4.	j	– 1. klauzule z negovaného závěru
5.	$\neg r$	– 2. klauzule z negovaného závěru
6.	p	– rezoluce: 1,4
7.	$d \vee r$	– rezoluce: 2,4
8.	$\neg d$	– rezoluce: 3,6
9.	r	– rezoluce: 7,8

Rezoluční metoda

Sepíšeme si jednotlivé klauzule:

1.	$\neg j \vee p$	– předpoklad 1
2.	$\neg j \vee d \vee r$	– předpoklad 2
3.	$\neg d \vee \neg p$	– přespoklad 3
4.	j	– 1. klauzule z negovaného závěru
5.	$\neg r$	– 2. klauzule z negovaného závěru
6.	p	– rezoluce: 1,4
7.	$d \vee r$	– rezoluce: 2,4
8.	$\neg d$	– rezoluce: 3,6
9.	r	– rezoluce: 7,8
10.	$\perp$	– rezoluce: 5,9

Byl odvozen spor, takže závěr skutečně z daných předpokladů vyplývá.

Rezoluční metoda

Poznámky:

- Na rezoluční metodu se lze dívat jako na vytváření jediné „obří“ formule v KNF, která je ekvivalentní formuli
$$\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n \wedge \neg\psi,$$
a která vzniká postupným přidáváním jednotlivých klauzulí.

- Pokud se nepodaří vygenerovat spor, lze odvozené klauzule použít k nalezení příkladu pravdivostního ohodnocení v, při kterém platí předpoklady $\varphi_1, \varphi_2, \ldots, \varphi_n$ a neplatí závěr ψ.

Rezoluční metoda

- Lze postupovat i přímou metodou, kdy se začne jen z předpokladů

$$\varphi_1, \varphi_2, \ldots, \varphi_n$$

a snažíme se postupně vygenerovat všechny klauzule závěru ψ.

Tento postup však nezaručuje, že se tyto klauzule podaří vygenerovat ve všech případech, kdy závěr ψ logicky vyplývá z předpokladů $\varphi_1, \varphi_2, \ldots, \varphi_n$.

Příklad: Klauzuli $p \vee q$ tímto způsobem nelze vygenerovat z předpokladu p, i když platí

$$p \models p \vee q.$$

Predikátová logika

Predikátová logika

- *Ryby jsou obratlovci žijící ve vodě.*
- *Kapři jsou ryby.*
- *Existuje alespoň jeden kapr.*

- *Existuje alespoň jeden obratlovec žijící ve vodě.*

- *Trojúhelníky jsou konvexní mnohoúhelníky.*
- *Rovnostranné trojúhelníky jsou trojúhelníky.*
- *Existuje alespoň jeden rovnostranný trojúhelník.*

- *Existuje alespoň jeden konvexní mnohoúhelník.*

Predikátová logika

- *Ryby jsou obratlovci žijící ve vodě.*
- *Kapři jsou ryby.*
- *Existuje alespoň jeden kapr.*

- *Existuje alespoň jeden obratlovec žijící ve vodě.*

Použití **proměnných**:

- *Pro každé x platí, že pokud x je ryba, tak x je obratlovec a x žije ve vodě.*
- *Pro každé x platí, že pokud x je kapr, tak x je ryba.*
- *Existuje alespoň jedno x takové, že x je kapr.*

- *Existuje alespoň jedno x takové, že x je obratlovec a x žije ve vodě.*

Predikátová logika

- *Trojúhelníky jsou konvexní mnohoúhelníky.*
- *Rovnostranné trojúhelníky jsou trojúhelníky.*
- *Existuje alespoň jeden rovnostranný trojúhelník.*

- *Existuje alespoň jeden konvexní mnohoúhelník.*

Použití **proměnných**:

- *Pro každé x platí, že pokud x je trojúhelník, tak x je mnohoúhelník a x je konvexní.*
- *Pro každé x platí, že pokud x je rovnostranný trojúhelník, tak x je trojúhelník.*
- *Existuje alespoň jedno x takové, že x je rovnostranný trojúhelník.*

- *Existuje alespoň jedno x takové, že x je mnohoúhelník a x je konvexní.*

Predikátová logika

- *Pro každé x platí, že pokud x má vlastnost P, tak x má vlastnost Q a x má vlastnost R.*
- *Pro každé x platí, že pokud x má vlastnost S, tak x má vlastnost P.*
- *Existuje alespoň jedno x takové, že x má vlastnost S.*

- *Existuje alespoň jedno x takové, že x má vlastnost Q a x má vlastnost R.*

P	je ryba	je trojúhelník
Q	je obratlovec	je mnohoúhelník
R	žije ve vodě	je konvexní
S	je kapr	je rovnostranný trojúhelník

Predikátová logika

- *Pro každé* x *platí, že pokud* $P(x)$, *tak* $Q(x)$ *a* $R(x)$.
- *Pro každé* x *platí, že pokud* $S(x)$, *tak* $P(x)$.
- *Existuje* x *takové, že* $S(x)$.

- *Existuje* x *takové, že* $Q(x)$ *a* $R(x)$.

$P(x)$	x je ryba	x je trojúhelník
$Q(x)$	x je obratlovec	x je mnohoúhelník
$R(x)$	x žije ve vodě	x je konvexní
$S(x)$	x je kapr	x je rovnostranný trojúhelník

Predikátová logika

- *Pro každé* x *platí* $(P(x) \rightarrow (Q(x) \wedge R(x)))$.
- *Pro každé* x *platí* $(S(x) \rightarrow P(x))$.
- *Existuje* x *takové, že* $S(x)$.

- *Existuje* x *takové, že* $(Q(x) \wedge R(x))$.

$P(x)$	x je ryba	x je trojúhelník
$Q(x)$	x je obratlovec	x je mnohoúhelník
$R(x)$	x žije ve vodě	x je konvexní
$S(x)$	x je kapr	x je rovnostranný trojúhelník

Predikátová logika

- $\forall x(P(x) \rightarrow (Q(x) \wedge R(x)))$
- $\forall x(S(x) \rightarrow P(x))$
- $\exists x\, S(x)$

- $\exists x(Q(x) \wedge R(x))$

$P(x)$	x je ryba	x je trojúhelník
$Q(x)$	x je obratlovec	x je mnohoúhelník
$R(x)$	x žije ve vodě	x je konvexní
$S(x)$	x je kapr	x je rovnostranný trojúhelník

- $\forall$ — univerzální kvantifikátor („*pro každé*“)
- $\exists$ — existenční kvantifikátor („*existuje*“)

Predikátová logika

Formule predikátové logiky vyjadřují tvrzení o objektech, které mají nějaké vlastnosti a které jsou v určitých vzájemných vztazích.

Interpretace či **interpretační struktura** — konkrétní soubor těchto objektů, jejich vlastností a vztahů.

Universum — soubor všech objektů v dané interpretaci

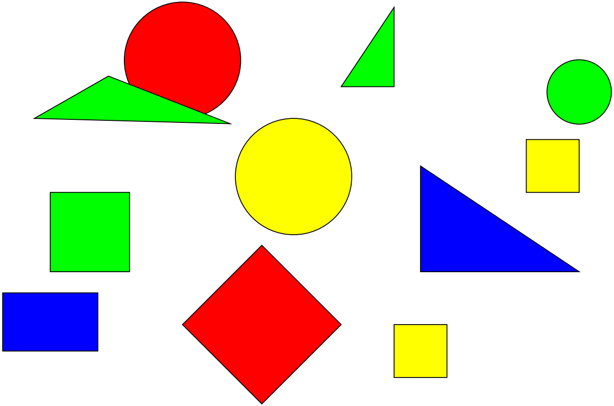
- Universem může být libovolná **neprázdná** množina.
- Objektům z tohoto universa se říká **prvky** universa.

Valuace — přiřazení prvků universa proměnným

Pravdivost formulí závisí na dané interpretaci a valuaci.

Predikátová logika

Příklad universa:



Další příklady univers:

- Nějaká přesně vymezená množina lidí, například množina všech obyvatel daného domu („*Jan Novák*“, „*František Vomáčka*“, ...)
- Množina všech knih v dané knihovně.
- Množina přirozených čísel $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$.
- Množina všech bodů v rovině.
- Množina $\{a, b, c, d, e\}$.
- Množina $\{a\}$.

Proměnné

Proměnné — x, y, z, ..., případně s indexy — x_0, x_1, x_2, ...

Předpokládáme, že k dispozici je nekonečný počet proměnných.

Valuace — přiřazení prvků universa proměnným

Příklad:

- Universum — nějaká množina lidí; valuace v, kde:

  $v(x) =$ *„František Vomáčka"*
  $v(y) =$ *„Pavla Nováková"*
  ...

- Universum — množina přirozených čísel $\mathbb{N} = \{0, 1, 2, \ldots\}$; valuace v, kde

$$v(x) = 57 \qquad v(y) = 3 \qquad v(z) = 57 \qquad \ldots$$

Predikáty — P, Q, R, ...

- **Unární predikáty** — reprezentují **vlastnosti** prvků universa

  Příklad: Predikát P reprezentující vlastnost *„být modrý"*:

  $$P(x) \quad — \quad „x \textit{ je modrý}“$$

  Unární predikát přiřazuje prvkům universa pravdivostní hodnoty.

  Např. hodnota $P(x)$ může být:

  - 1 — prvek přiřazený proměnné x má danou vlastnost P (tj. je modrý)
  - 0 — prvek přiřazený proměnné x nemá danou vlastnost P (tj. není modrý)

Predikáty

- **Binární predikáty** — reprezentují **vztahy** mezi dvojicemi prvků universa

  Příklad: Predikát R reprezentující vztah *„být rodičem"*:

  $$R(x, y) \quad — \quad \text{„}x \text{ je rodičem } y\text{"}$$

  Binární predikát přiřazuje pravdivostní hodnoty dvojicím prvků universa.

  Např. hodnota $R(x, y)$ může být:

  - 1 — když x a y jsou v daném vztahu (tj. x je rodičem y)
  - 0 — když x a y v daném vztahu nejsou (tj. x není rodičem y)

Predikáty

Můžeme uvažovat i predikáty libovolné jiné arity.

Například:

- **Ternární** predikát T (tj. predikát s aritou 3) reprezentující vztah mezi rodiči a dítětem:

$$T(x, y, z)$$

— x a y jsou rodiči dítěte z, přičemž x je jeho matka a y je jeho otec

- Na **nulární** predikáty (tj. predikáty s aritou 0) se můžeme dívat jako na atomické výroky, které se nevztahují k prvkům universa.

Formule predikátové logiky

Atomická formule — predikát aplikovaný na nějaké proměnné

Příklad:

- P — unární predikát reprezentující vlastnost *„být modrý"*
- Q — unární predikát reprezentující vlastnost *„být čtverec"*
- R — binární predikát reprezentující vztah *„překrývá"*

$$
\begin{array}{ccl}
P(x) & — & \text{„}x \text{ je modrý"} \\
P(y) & — & \text{„}y \text{ je modrý"} \\
Q(y) & — & \text{„}y \text{ je čtverec"} \\
R(z,x) & — & \text{„}z \text{ překrývá } x\text{"} \\
R(y,y) & — & \text{„}y \text{ překrývá sám sebe"}
\end{array}
$$

Poznámka: Později pojem atomické formule poněkud rozšíříme.

Formule predikátové logiky

Z jednodušších formulí je možno vytvářet složitější formule pomocí **logických spojek** ("$\neg$", "$\wedge$", "$\vee$", "$\rightarrow$", "$\leftrightarrow$") stejně jako ve výrokové logice.

Příklad:

- P — unární predikát reprezentující vlastnost *„být modrý"*
- Q — unární predikát reprezentující vlastnost *„být čtverec"*
- R — binární predikát reprezentující vztah *„překrývá"*

„Jestliže x je modrý čtverec nebo y nepřekrývá x, tak z není čtverec."

$$\big((P(x) \wedge Q(x)) \vee \neg R(y,x)\big) \rightarrow \neg Q(z)$$

Formule predikátové logiky

Z jednodušších formulí je možno vytvářet složitější formule pomocí **logických spojek** (“$\neg$”, “$\wedge$”, “$\vee$”, “$\rightarrow$”, “$\leftrightarrow$”) stejně jako ve výrokové logice.

Příklad:

- P — unární predikát reprezentující vlastnost *„být žena“*
- Q — unární predikát reprezentující vlastnost *„mít tmavé vlasy“*
- R — binární predikát reprezentující vztah *„být rodičem“*

„Jestliže x je žena s tmavými vlasy nebo y není rodičem x, tak z nemá tmavé vlasy.“

$$\big((P(x) \wedge Q(x)) \vee \neg R(y,x)\big) \rightarrow \neg Q(z)$$

Formule predikátové logiky

Z jednodušších formulí je možno vytvářet složitější formule pomocí **logických spojek** (“$\neg$”, “$\wedge$”, “$\vee$”, “$\rightarrow$”, “$\leftrightarrow$”) stejně jako ve výrokové logice.

Příklad:

- P — unární predikát reprezentující vlastnost *„být sudý“*
- Q — unární predikát reprezentující vlastnost *„být prvočíslo“*
- R — binární predikát reprezentující vztah *„být větší“*

„Jestliže x je sudé prvočíslo nebo y není větší než x, tak z není prvočíslo.“

$$\big((P(x) \wedge Q(x)) \vee \neg R(y,x)\big) \rightarrow \neg Q(z)$$

Kvantifikátory

Univerzální kvantifikátor — symbol "$\forall$"

Jestliže φ je formule reprezentující určité tvrzení, tak

$$\forall x \varphi$$

je formule reprezentující tvrzení

„pro každé x platí φ“.

Příklad: P — *„být čtverec“*

$$\forall x \, P(x)$$

- *„Pro každé x platí, že x je čtverec.“*
- *„Každé x je čtverec.“*
- *„Všechny prvky jsou čtverce.“*

Kvantifikátory

Příklad:

- *„Pro každé* x *platí, že pokud* x *je čtverec, tak* x *je zelený."*
- *„Všechny čtverce jsou zelené."*

$$\forall x(P(x) \rightarrow Q(x))$$

- P — *„být čtverec"* (arita 1)
- Q — *„být zelený"* (arita 1)

Kvantifikátory

Příklad:

- *„Jestliže pro každé x platí, že x je čtverec nebo x je zelený, tak pro každé y platí, že y je trojúhelník.“*
- *„Jestliže je každý objekt čtverec nebo je zelený, tak jsou všechny prvky trojúhelníky.“*

$$\forall x(P(x) \vee Q(x)) \rightarrow \forall y T(y)$$

- P — *„být čtverec“* (arita 1)
- Q — *„být zelený“* (arita 1)
- T — *„být trojúhelník“* (arita 1)

Kvantifikátory

Zásadní rozdíl mezi následujícími formulemi:

- $P(x)$ — *„x je čtverec“*

  Mluví o **jednom** konkrétním prvku přiřazeném proměnné x.

  Pravdivost tohoto tvrzení závisí na konkrétním prvku přiřazeném proměnné x, tj. na konkrétní valuaci.

- $\forall x P(x)$ — *„každé x je čtverec“* (tj. *„všechny prvky jsou čtverce“*)

  Mluví o **všech** prvcích universa.

  Pravdivost tohoto tvrzení nezávisí na konkrétní valuaci.

Kvantifikátory

Příklad:

- *„Jestliže je x prvočíslo, pak je liché."*

$$P(x) \rightarrow L(x)$$

- *„Pro každé x platí, že pokud je x prvočíslo, pak je liché"*.
  (Tj. *„všechna prvočísla jsou lichá"*.)

$$\forall x(P(x) \rightarrow L(x))$$

Predikáty:

- P — *„být prvočíslo"* (arita 1)
- L — *„být lichý"* (arita 1)

Kvantifikátory

Příklad:

- *„Pro každé y platí, že pokud y je zelený, tak x překrývá y."*
- *„Objekt x překrývá všechny zelené objekty."*

$$\forall y (G(y) \rightarrow R(x, y))$$

Predikáty:

- R — *„překrývá"* (arita 2)
- G — *„je zelený"* (arita 1)

Kvantifikátory

Příklad:

- *„Pro každé x platí, že pro každé y platí, že pokud x je rodičem y, tak x má rád y."*
- *„Pro každé x a y platí, že pokud x je rodičem y, tak x má rád y."*
- *„Pro každé dva prvky x, y platí, že pokud x je rodičem y, tak x má rád y."*

$$\forall x \forall y (R(x, y) \rightarrow S(x, y))$$

Predikáty:

- R — *„je rodičem"* (arita 2)
- S — *„má rád"* (arita 2)

Kvantifikátory

Existenční kvantifikátor — symbol "$\exists$"

Jestliže φ je formule reprezentující určité tvrzení, tak

$$\exists x \varphi$$

je formule reprezentující tvrzení

„*existuje* x*, pro které platí* φ“.

Příklad: P — „*být čtverec*“

$$\exists x\, P(x)$$

- „*Existuje* x*, pro které platí, že* x *je čtverec.*“
- „*Existuje* x *takové, že* x *je čtverec.*“
- „*Existuje alespoň jeden čtverec.*“

Kvantifikátory

Příklad:

- *„Existuje* x, *pro které platí, že* x *je čtverec a* x *je zelený."*
- *„Existuje* x *takové, že* x *je čtverec a* x *je zelený."*
- *„Pro nějaké* x *platí, že* x *je čtverec a* x *je zelený."*
- *„Existuje zelený čtverec."*
- *„Některé čtverce jsou zelené."*
- *„Alespoň jedno* x *je zelený čtverec."*

$$\exists x(P(x) \wedge Q(x))$$

Predikáty:

- P — *„být čtverec"* (arita 1)
- Q — *„být zelený"* (arita 1)

Kvantifikátory

Příklad:

- *„Existuje* x *takové, že pro každé* y *je* x *větší než* y*.“*

$$\exists x \forall y\, P(x, y)$$

- *„Pro každé* y *existuje* x *takové, že* x *je větší než* y*.“*

$$\forall y \exists x\, P(x, y)$$

P — *„být větší než“* (arita 2)