Referáty — zimní semestr 2024/25

- Referáty jsou studentům přiděleny v průběhu semestru. Každý student obdrží e-mail s informací o čísle referátu, který mu byl přidělen. Na webové stránce předmětu bude uveden i nejpozdější termín přidělení. Student, kterému případně nebyl referát příslušným způsobem přidělen k danému termínu, se musí bez zbytečného prodlení přihlásit cvičícímu (například e-mailem).

- Rychlé prověření podkladů k referátu s otestováním skutečného porozumění proběhne v termínech ke konci semestru (jak bude sděleno na webové stránce předmětu). Na referátu je nejdůležitější vaše **prezentace**, kdy prokážete, že jste tématu plně porozuměli a prezentaci si pečlivě připravili tak, abyste podstatu věci stihli kolegům a vyučujícímu rozumně vysvětlit (ilustrovat) v čase **nejvýše 15 minut**. Z technických důvodů nebudete ve skutečnosti 15 minut prezentovat, ale otestujeme, zda jste na to připraveni.

- Ke každému referátu je uveden jako výchozí zdroj jeden konkrétní článek, ve kterém v zásadě naleznete veškeré podstatné informace, které by měly postačovat ke zpracování daného tématu.

  Všechny články k referátům jsou k dispozici v elektronické formě v MS Teams v týmu `Teoretická informatika 2024/25` v kanálu `Obecné` pod kartou `Soubory` ve složce

  `Dokumenty / General / Referáty / Podklady`.

  K tématu můžete ale samozřejmě čerpat informace a podklady z jakýchkoli veřejných zdrojů.

  Dané téma musíte sami za sebe srozumitelně a uceleně písemně zpracovat. Na všechny zdroje, ze kterých čerpáte, musíte řádně odkázat.

- **Písemný podklad** vyučujícím nezasílejte. Přineste si jej na papíře k onomu otestování. Během prezentace jej můžete používat a po prezentaci ho pak odevzdáte. Může být zpracován jak na počítači, tak rukou napsaný, s příslušnými obrázky apod. Musí obsahovat dostatečné informace k posouzení, zda je to vámi promyšleně sestavený podklad k prezentaci (a nikoli jen např. překopírované kusy textu z jiných zdrojů).

  Tento písemný podklad nemá mít podobu slidů, ale uceleného textu.

  Musí na něm být rovněž uvedeno vaše jméno a login.

Referát č. 1 — *Simulace vícepáskového Turingova stroje pomocí dvoupáskového*

Řekněme, že máme dán Turingův stroj s k páskami (kde k je nějaké přirozené číslo) a chtěli bychom činnost tohoto stroje simulovat pomocí Turingova stroje se dvěma páskami. Popište obecný způsob, jak to udělat tak, aby platilo, že pokud původní k-páskový stroj provede t kroků, tak simulující stroj se dvěma páskami provede nejvýše $\mathcal{O}(t \log t)$ kroků.

Hlavním zdrojem pro tento referát je článek

> F. C. Hennie, R. E. Stearns: Two-Tape Simulation of Multitape Turing Machines, *Journal of the Association for Computing Machinery*, Vol. 13, No. 4, pp. 533–546, 1966.
> DOI: `https://doi.org/10.1145/321356.321362`

Referát č. 2 — *Turingovská úplnost 2-tag systémů*

Přepisovací systémy označované jako *tag systémy* jsou jedny z nejjednodušších výpočetních modelů. *Tag systém* je možné popsat jako trojici (k, Σ, R), kde $k \geq 1$ je přirozené číslo, Σ je konečná abeceda a R je množina pravidel tvaru $a \mapsto w$, kde $a \in \Sigma$ a $w \in \Sigma^*$, přičemž pro každé $a \in \Sigma$ existuje vždy nejvýše jedno w takové, že $(a \mapsto w) \in R$.

Konfigurace daného tag systému jsou reprezentovány jako slova nad abecedou Σ (tj. jako prvky množiny Σ^*). Kroky, které může systém provádět, vypadají tak, že pokud má konfigurace u délku alespoň k, začíná symbolem a a v R existuje nějaké pravidlo tvaru $a \mapsto w$, je možné přejít z konfigurace u do konfigurace v, kterou dostaneme z u tak, že nejprve odmažeme prvních k znaků a následně na konec zbylé sekvence znaků přidáme slovo w.

Jako 2-*tag systémy* jsou označovány tag systémy, ve kterých je výše uvedené číslo k rovno 2, tj. kde jsou v každém kroku odmazávány vždy první dva znaky.

Ukažte, že 2-tag systémy jsou Turingovsky úplné, tj. že pro každý Turingův stroj $\mathcal{M}$ je možné sestrojit 2-tag systém, který jeho činnost simuluje.

Hlavním zdrojem pro tento referát je článek

> J. Cocke, M. Minsky: Universality of Tag Systems with $P = 2$, *Journal of the ACM*, Vol. 11, Issue 1, pp. 15–20, 1964.
> DOI: `https://doi.org/10.1145/321203.321206`

Referát č. 3 — *Presburgerova aritmetika*

Presburgerova aritmetika je formální systém tvořený formulemi predikátové logiky prvního řádu, ve kterých se v termech mohou vyskytovat pouze proměnné, celočíselné konstanty a operace sčítání, a kde jediné predikátové symboly jsou $=$ a $<$. Tyto formule jsou vyhodnocovány v interpretaci, kde universem je množina celých čísel, a kde jednotlivé symboly ($+$, $=$, $<$ a číselné konstanty) jsou interpretovány standardním způsobem jako sčítání, rovnost, atd. na oboru celých čísel.

Vzhledem k tomu, že zápis jako třeba $3x$ je možné chápat jako zkratku pro $x+x+x$, často se v zápisu formulí Presburgerovy aritmetiky povoluje používat v termech též násobení proměnné konstantou. Není však možné násobit například dvě proměnné.

Typickým příkladem formule Presburgerovy aritmetiky je třeba

$$\forall x \exists y \forall z (x = 4y + 5z + 2 \vee (\neg(5x < z) \wedge y < 8x + 13)$$

Vezměme si následující problém:

VSTUP: Uzavřená formule Presburgerovy aritmetiky φ.

OTÁZKA: Je formule φ pravdivá (ve standardní interpretaci na celých číslech)?

Ukažte, že tento problém je algoritmicky rozhodnutelný. Detailně vysvětlete činnost algoritmu řešícího tento problém popsaného v níže uvedeném článku, a to nejen z hlediska rozhodnutelnosti, ale i včetně všech různých vylepšení navržených v tomto článku, které je možné využít pro efektivnější implementaci daného algoritmu. (Otázku výpočetní složitosti daného algoritmu v rámci referátu řešit nemusíte.)

Hlavním zdrojem pro tento referát je článek

> D.C. Cooper: Theorem Proving in Arithmetic without Multiplication, *Machine Intelligence* 7, pp. 91–99, 1972.

Referát č. 4 — *NP-úplnost hry Minesweeper*

Minesweeper je známá hra, ve které máme dánu plochu skládající se ze čtvercových políček, z nichž některá mohou obsahovat miny. Na začátku jsou všechna políčka zakrytá. Postupným klikáním se postupně odkrývají, přičemž se na odkrytých políčcích zobrazuje informace o počtu min na sousedních políčcích (jako číslo v intervalu 1 až 8). Cílem je vyhnout se kliknutí na minu, označit pozice, kde se nachází miny, a postupně odkrýt všechna políčka, kde se miny nenachází.

Uvažujme následující problém:

VSTUP: Pozice z průběhu hry Minesweeper, kde je část políček odkrytých, část zakrytých a u části z nich je vyznačeno, že se tam nachází miny.

OTÁZKA: Je tato daná pozice konzistentní v tom smyslu, že existuje nějaká možnost, jak mohou být miny rozmístěny v zakrytých políčcích tak, aby to odpovídalo zobrazené informaci?

Ukažte, že tento problém je NP-úplný.

Hlavním zdrojem pro tento referát je článek

> R. Kaye: Minesweeper is NP-complete, *The Mathematical Intelligencer*, Vol. 22, No. 2, pp. 9–15, 2000.
> DOI: `https://doi.org/10.1007/BF03025367`

Referát č. 5 — *NP-úplnost jigsaw puzzle a dalších podobných skládaček*

Uvažujme různé skládačky toho typu, kdy máme dánu nějakou sadu dílků, a cílem je poskládat dané dílky podle určitých stanovených pravidel k sobě tak, aby zaplnily danou plochu (např. čtvercového nebo obdélníkového tvaru). V některých variantách takových skládaček mohou být všechny dílky stejného tvaru (např. všechny čtvercové), v jiných mohou být dílky různého tvaru. Navíc v některých variantách mohou být jednotlivé strany dílků označeny například různými barvami, symboly, obrázky, apod. Pravidla pro konkrétní druh skládačky pak omezují to, které dílky je možné k sobě přikládat a které ne.

Pro dané typy skládaček můžeme uvažovat algoritmické problémy, kdy vstupem je popis sady dílků takové skládačky a tvar plochy, kterou je třeba jimi zaplnit, a otázka je, zda existuje pro danou sadu dílků řešení, tj. zda je možné je poskládat tak, aby zaplnily danou plochu.

Popište důkazy toho, že pro velkou část variant těchto skládaček je daný problém NP-úplný a problémy pro jednotlivé typy skládaček jsou mezi sebou vzájemně převeditelné, tj. popište, jak vypadají jednotlivé polynomiální redukce mezi těmito problémy.

Hlavním zdrojem pro tento referát je článek

> E. D. Demaine, M. L. Demaine: Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity, *Graphs and Combinatorics*, Vol. 23, supplement issue 1, pp. 195–208, 2007.
> DOI: `https://doi.org/10.1007/s00373-007-0713-4`

Referát č. 6 — *PSPACE-úplnost zobecněné varianty hry Hex*

Uvažujme hru dvou hráčů, kterou hrají na neorientovaném grafu, na jehož vrcholy střídavě pokládají kameny — každý z hráčů má kameny své vlastní barvy. V grafu jsou vyznačeny dva speciální vrcholy s a t, přičemž na tyto dva vrcholy nesmí hráči kameny pokládat. Cílem prvního hráče je vytvořit ze svých kamenů cestu z vrcholu s do vrcholu t. Cílem druhého hráče je mu v tom zabránit, tj. položit své kameny tak, aby na každé cestě z s do t ležel alespoň jeden jeho kámen, takže první hráč nebude schopen cestu z s do t ze svých kamenů vytvořit.

Jedná se o určité zobecnění hry Hex, která se hraje na hrací ploše pravidelného tvaru skládající se z šestiúhelníků, na případ, kdy může mít hrací plocha podobu libovolného grafu (který ani nemusí být takový, aby ho bylo možné nakreslit do roviny).

Uvažujme následující problém:

VSTUP: Neorientovaný graf G s vyznačenými speciálními vrcholy s a t.

OTÁZKA: Má první hráč ve hře hrané na grafu G vyhrávající strategii?

(*Poznámka:* Předpokládá se, že hra začíná s prázdným grafem, kdy na začátku neleží v grafu žádné kameny.)

Ukažte, že tento problém je PSPACE-úplný.

Hlavním zdrojem pro tento referát je článek

> S. Even, R. E. Tarjan: A Combinatorial Problem Which Is Complete in Polynomial Space, *Journal of the Association for Computing Machinery*, Vol. 23, No. 4, pp. 710–719, 1976.
> DOI: `https://doi.org/10.1145/321978.321989`

Referát č. 7 — *PSPACE-obtížnost hry GO*

Hra GO je známá desková hra, kterou hrají dva hráči, kdy jeden z nich hraje černými kameny a druhý bílými, přičemž střídavě pokládají tyto kameny na hrací plochu. Standardně má tato hrací plocha velikost 19×19, můžeme ale uvažovat obecnější variantu, kdy má hrací plocha velikost $n \times n$, kde n je libovolné přirozené číslo.

Vezměme si následující problém:

VSTUP: Pozice ve hře GO na hrací ploše libovolné velikosti $n \times n$, kde některé černé i bílé kameny jsou již na této ploše položeny, a informace, který hráč je na tahu.

OTÁZKA: Má hráč, který je v dané pozici na tahu, vyhrávající strategii?

Ukažte, že tento problém je PSPACE-těžký.

Hlavním zdrojem pro tento referát je článek

> D. Lichtenstein, M. Sipser: GO Is Polynomial-Space Hard, *Journal of the Association for Computing Machinery*, Vol. 27, No. 2, pp. 393–401, 1980.
> DOI: `https://doi.org/10.1145/322186.322201`

Referát č. 8 — *P-úplnost prohledávání grafu do hloubky*

Předpokládejme, že máme dán orientovaný graf G s vybraným počátečním vrcholem s, kde je pro každý vrchol stanoveno nějaké konkrétní pořadí hran, které z něj vedou. Pokud spustíme na daném grafu algoritmus prohledávání do hloubky, kde budeme hrany probírat ve stanoveném pořadí, dostaneme konkrétní pořadí, ve kterém budou jednotlivé vrcholy navštíveny.

Uvažujme následující problém:

VSTUP: Orientovaný graf G s počátečním vrcholem s a určeným pořadím hran vedoucích z jednotlivých vrcholů a dvojice jeho vrcholů u a v.

OTÁZKA: Bude při prohledávání do hloubky v grafu G navštíven vrchol u dříve než vrchol v?

Ukažte, že tento problém je P-úplný.

Hlavním zdrojem pro tento referát je článek

J. H. Reif: Depth-first search is inherently sequential, *Information Processing Letters*, Vol. 20, No. 5, pp. 229–234, 1985.
DOI: `https://doi.org/10.1016/0020-0190(85)90024-9`

Referát č. 9 — *P-úplnost hledání maximálního toku v síti*

Připomeňme si problém hledání maximálního toku v síti. U tohoto problému je vstupem síť, což je speciální případ orientovaného grafu, kde jsou jednotlivým hranám přiřazeny kapacity, a kde dva vrcholy jsou označeny jako speciální — vrchol s představuje zdroj a vrchol t stok. Dané síti je možné přiřadit tok, který specifikuje pro každou hranu, kolik toho protéká danou hranou. Aby se jednalo skutečně o tok, musí být dodrženy následující omezující podmínky. Jednak nesmí být překročena kapacita žádné z hran, jednak pro každý z vrcholů, s výjimkou zdroje s a stoku t, musí platit, že co do daného vrcholu vtéká, to z něj musí také vytékat. Hodnota toku je dána jako součet toho, co vytéká ze zdroje s (resp. co vtéká do stoku t). Cílem je najít takový tok, jehož hodnota je maximální.

Uvažujme následující rozhodovací variantu tohoto problému:

VSTUP: Síť G se zdrojem s a stokem t a přirozené číslo i.
(Kapacity hran v síti G jsou reprezentovány binárně.)

OTÁZKA: Má i-tý bit binárně zapsané maximální hodnoty toku v síti G hodnotu 1?

Ukažte, že tento problém je P-úplný.

(*Poznámka:* Není třeba, abyste detailně popisovali nějaký polynomiální algoritmus řešící tento problém. Soustřeďte se primárně na důkaz jeho P-obtížnosti.)

Hlavním zdrojem pro tento referát je článek

L. M. Goldschlager, R. Shaw, J. Staples: The maximum flow problem is log space complete for P, *Theoretical Computer Science*, Vol. 21, No. 1, pp. 105–111, 1982.
DOI: `https://doi.org/10.1016/0304-3975(82)90092-5`

Referát č. 10 — *P-úplnost detekce deadlocku*

Uvažujme situaci, kdy máme řadu paralelně běžících procesů, které mohou využívat několik sdílených prostředků, přičemž každý z těchto prostředků má určitou kapacitu, která stanovuje maximální počet jednotek daného prostředku, které jsou celkově k dispozici. Procesy mohou žádat o přidělení daných prostředků tím, že spustí příslušné požadavky. Proces je vždy pozastaven do doby, než mu jsou příslušné prostředky, o které žádá, přiděleny. Poté může některé z přidělených prostředků uvolnit a případně vyvolat další požadavky, atd.

Stav s deadlockem je situace, kdy v daném systému existuje nějaká neprázdná podmnožina procesů, které jsou pozastaveny při čekání na přidělení zdrojů, a které zůstanou takto pozastaveny už navždy bez ohledu na to, jak budou dále probíhat výpočty ostatních procesů a přidělování zdrojů.

Toto je možné modelovat jako grafovou úlohu, kde pracujeme s orientovaným bipartitním multigrafem, kde jedna část vrcholů představuje procesy a druhá část přidělované prostředky. Hrany pak reprezentují celkovou aktuální situaci z hlediska přidělení prostředků a dosud nevyřízených požadavků. Hrany z vrcholů reprezentujících procesy do vrcholů reprezentujících prostředky reprezentují aktuální požadavky jednotlivých procesů, a hrany vedoucí od prostředků k procesům reprezentují aktuální přidělení prostředků.

Zaměříme se na následující problém:

VSTUP: Orientovaný multigraf reprezentující aktuální celkový stav systému.

OTÁZKA: Existuje v daném stavu deadlock?

Ukažte, že tento problém je P-úplný.

Hlavním zdrojem pro tento referát je článek

P. Spirakis: The parallel complexity of deadlock detection, *Theoretical Computer Science*, Vol. 52, Issues 1–2, pp. 155–163, 1987.
DOI: `https://doi.org/10.1016/0304-3975(87)90084-3`