Přednáška #13: **Teorie paralelní složitosti - NC algoritmy a P-úplnost**

Úvod

Třída složitosti: soubor problémů řešitelných algoritmy s asymptoticky stejnými omezeními prostředků v nejhorším případě.

Přiměřenost výpočetního prostředku: rychlost jeho růstu je omezena polynomiální funkcí velikosti vstupu n.

Sekvenčně přiměřený (schůdný) problém: řešitelný v polynomiálním čase

$P =$ třída problémů řešitelných v polynomiálním sekvenčním čase

Sekvenčně neschůdný problém: horní mez řešení je striktně vyšší než polynomiální.

Přiměřený počet procesorů: polynomiálně omezen

$\Longrightarrow$ paralelní přiměřenost $\equiv$ sekvenční přiměřenost

$\Longrightarrow$ teorie paralelní složitosti je založena na teorii sekvenční složitosti třídy P.

Optimalizační a rozhodovací abstraktní problémy

Abstraktní problém: relace mezi množinou instancí problému a množinou řešení problému.

PŘÍKLAD: Problém nejkratší cesty (SPP): Je-li dán graf G a dva vrcholy u, v, nalezněte v G nejkratší cestu mezi u a v. Instance: trojice $\langle G, u, v\rangle$. Řešení pro $\langle G, u, v\rangle$: žádná, jedna, nebo více nejkratších cest z u do v.

Velikost instance: počet symbolů (číslic, písmen, omezovačů, apod.) v její specifikaci.

Rozhodovací problém: problém s řešením `Ano`/`Ne`.

- Optimalizační problémy mohou být formulovány jako rozhodovací problémy, typicky uvalením mezí na hodnoty, které se mají optimalizovat.

  PŘÍKLAD. CESTA = rozhodovací problém pro SPP:
  Je dán G, u, v a celé číslo k. $\exists$ v G mezi u a v cesta, jejíž délka je nejvýše k?

- Jestliže rozhodovací problém je těžký, je těžký i odpovídající optimalizační problém (optimalizační problém je snadný $\implies$ rozhodovací problém je snadný). Tudíž,

  teorie složitosti se zabývá rozhodovacími problémy.

Kódování a konkrétní problémy

Kompaktní kódování: kódování nad m-znakovou abecedou $\mathcal{Z}$ je kompaktní, jestliže jakákoli abstraktní instance velikosti n je zakódovaná pomocí $O(\log_m n)$ symbolů ze $\mathcal{Z}$.

Konkrétní problém: abstraktní problém zakódovaný pomocí nějakého kompaktního kódování, typicky binárního (např. ASCII).

Velikost vstupu $|x|$: délka binárního zakódování instance x.

Příklad. Neohodnocený jednoduchý graf G s n vrcholy lze zakódovat pomocí matice sousednosti použitím $O(n^2)$ bitů.

Sekvenční čas: sekvenční algoritmus řeší konkrétní problém v čase $T(n)$, jestliže poskytne řešení v čase $O(T(|x|))$ pro jakýkoli vstup x.

Třída složitosti P:

- množina všech problémů s $T(n) = O(n^{O(1)})$.
- robustní vzhledem ke kompaktnímu kódování:
  jakékoli takové kódování lze převést na binární kódování v polynomiálním čase
  $\Longrightarrow$ teorie složitosti se zabývá pouze binárním kódováním.

Jazyky, přijatelnost, a rozhodnutelnost

Binární jazyk L: množina binárních řetězců.

Rozhodovací problém Q: jazyk `Ano` instancí $L_Q = \{x \in \{0,1\}^*; Q(x) = 1\}$.

Příklad. L_{CESTA}: $\{\langle G, u, v, k\rangle$; $\exists$ cesta $u \to v$ v G délky $\leq k\}$.

Vstup přijatý/odmítnutý algoritmem: alg.A přijme (resp.odmítne) binární řetězec x, jestliže pro vstup x vyprodukuje výstup 1 (resp.0).

Přijatelnost: alg. A přijme L v čase $T(n)$, jestliže přijme $\forall\ x \in L$ v čase $O(T(|x|))$.

- Jestliže $x \notin L$, pak A buď x odmítne nebo se pro x zacyklí.

Charakteristická funkce f_L jazyka L: $f_L(x) = 1$, jestliže $x \in L$, a $f_L(x) = 0$ jinak.

Rozhodnutelnost: L je rozhodnutelný v čase $T(n)$ $\iff$ $f_L(x)$ lze spočíst v čase $O(T(|x|))$ $\iff$ $\exists$ alg., který přijme $\forall\ x \in L$ a odmítne $\forall\ x \notin L$ v čase $O(T(|x|))$.

- $\exists$ problémy (např. Turing Halting Problem), pro které $\exists$ přijímací alg., ale ne$\exists$ rozhodovací alg.

Polynomiální třída P: množina všech jazyků rozhodnutelných v polynomiálním čase.

Třída NP

Verifikační algoritmus: Alg. A verifikuje vstup x, jestliže $\exists$ certifikát y; $A(x,y)=1$.

Jazyk verifikovaný verifikačním algoritmem A:

$$L_A = \{x \in \{0,1\}^*; \exists y \in \{0,1\}^* \text{ takový, že } A(x,y)=1\}.$$

Jestliže $x \notin L_A$, pak nemůže existovat žádný certifikát y takový, že $A(x,y)=1$.

Třída složitosti NP: množina všech jazyků L verifikovatelných v polynomiálním čase.

Příklad. Problém hamiltonovské kružnice (HC): Má graf G hamiltonovskou kružnici?

1. HC je v NP, protože pro daný graf G a permutaci jeho vrcholů, můžeme verifikovat v polynomiálním čase, zda tato permutace je hamiltonovskou kružnicí G.
2. Není znám žádný algoritmus s polynomiálním časem, který by dokázal rozhodnout HC.

Poznámka: NP vzniklo historicky jako zkratka z nedeterministicky polynomiální.

Polynomiální redukce a NP-úplnost

Je zřejmé, že $P \subset NP$, ale neví se, zda $P = NP$ nebo $P \neq NP$. Existuje silná domněnka, že

$$P \neq NP$$

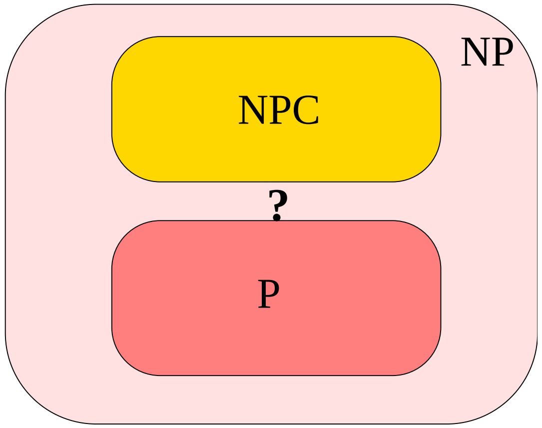
a že $NP - P$ obsahuje NP-úplné problémy.

Polynomiální redukce: L_1 je redukovatelný v polynomiálním čase na L_2, $L_1 \leq_p L_2$, jestliže $\exists$ funkce $r : \{0,1\}^* \to \{0,1\}^*$ taková, že

1. $\forall x \in \{0,1\}^* \ (x \in L_1 \iff r(x) \in L_2)$ a
2. r lze spočítat v polynomiálním čase, t.j. $T(r(x)) = O(|x|^{O(1)})$.

NP-úplný problém: L je NP-úplný, jestliže

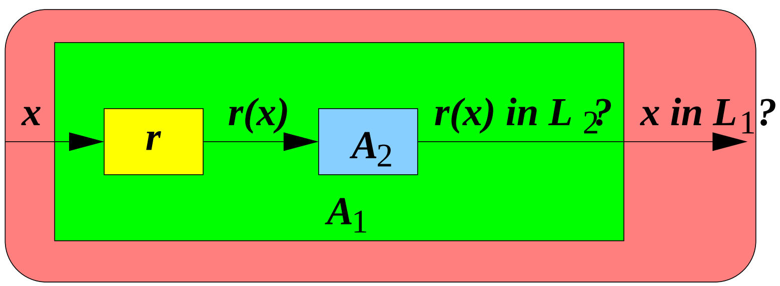
1. $L \in NP$ a
2. $L' \leq_p L$ pro každý $L' \in NP$.

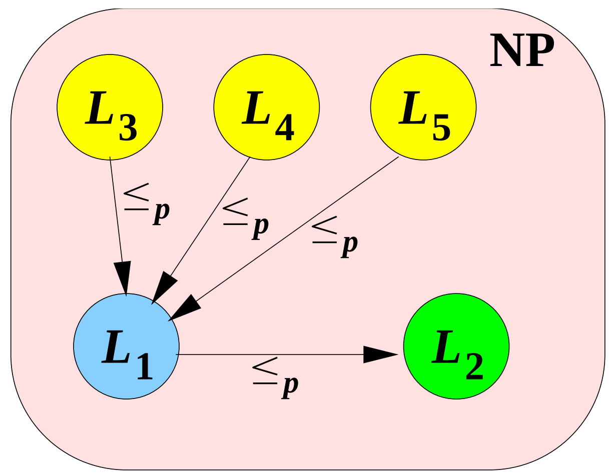


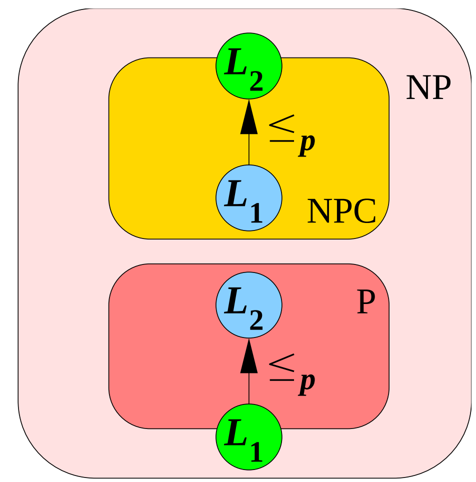
$P \stackrel{?}{=} NP$?

Lemma 1. *Třída P je uzavřena shora vzhledem k relaci "je P-redukovatelný na".*
(Jestliže $L_1 \in NP$ a $\exists\, L_2 \in P$ takový, že $L_1 \leq_p L_2$, pak $L_1 \in P$.)

Lemma 2. *Třída NP-úplných problémů je uzavřena zdola vzhledem k relaci "je P-redukovatelný na".*
(Jestliže $L_2 \in NP$ a $\exists$ NP-úplný L_1 takový, že $L_1 \leq_p L_2$, pak L_2 je NP-úplný.)







Důsledek 3.

Jestliže $\exists$ NP-úplný problém, který je v P, pak $P = NP$.
Jestliže $\exists$ NP-úplný problém, u kterého lze dokázat, že není v P, pak žádný NP-úplný problém nemůže být v P.

Důkazy NP-úplnosti

Abychom dokázali, že jazyk L je NP-úplný, musíme dokázat, že

1) $L \in NP$

(což je zpravidla snadné) a pak jsou na výběr dvě možnosti:

2a) Ukážeme, že jakýkoli jazyk v NP lze redukovat na L v polynomiálním čase.

> POZNÁMKA: Toto bylo nutné pro první NP-úplné problémy, jako např.
> `CIRCUIT-SAT`: Je-li dán boolský kombinatorický obvod složený z hradel AND, OR, a NOT, se vstupy $x_1, x_2, \ldots, x_n$ a vyznačeným výstupem y, existuje vstupní množina dat taková, že y je `true`?

2b) Jestliže existují již nějaké NP-úplné problémy, pak

1. vybereme nějaký vhodný NP-úplný L',
2. definujeme zobrazení r instancí jazyka L' na instance L,
3. dokážeme, že $x \in L' \iff r(x) \in L$ pro všechny $x \in \{0,1\}^*$,
4. dokážeme, že r lze spočítat v polynomiálním čase.

Vysoká paralelizovatelnost

Ve světě paralelních výpočtů existuje navíc důležitý parametr, a to je počet procesorů p. S vyjímkou tohoto rozdílu lze všechny pojmy klasické teorie složitosti použít.

Paralelní rozhodnutelnost: Jazyk L je rozhodnutelný v paralelním čase $T(n,p)$ $\iff$ $\forall x\ f_L(x)$ je spočitatelná v čase $T(|x|,p)$ pomocí p procesorů (typicky PRAM).

Paralelní přiměřenost: rozhodnutelnost v polynomiálním $T(n,p)$ s polynomiálním p

$\Longrightarrow$ paralelní přiměřenost $\equiv$ sekvenční přiměřenost

$\Longrightarrow$ paralelní teorie složitosti pracuje pouze s třídou složitosti P.

Vysoká paralelizovatelnost: subpolynomiální $T(n,p)$. V takovém případě obvykle musíme použít polynomiální počet procesorů, protože

$$p.T(n,p) \geq SU(n).$$

- Jaký polynomiální počet procesorů $p = p(n)$ je přijatelný (si můžeme dovolit)?

  Lineární nebo blízký lineárnímu počtu.

- Jaký subpolynomiální čas je dosažitelný?

  Typicky polylogaritmický, vyjímečně menší (např. plně paralelní algoritmy).

NC algoritmy

Tradiční třída vysoce paralelních algoritmů:

Definice 4. *Třída NC = množina jazyků rozhodnutelných v čase $T(n, p(n)) = O(log^{O(1)} n)$ s $p(n) = O(n^{O(1)})$ procesory (standardně uvažujeme PRAM model).*

Třída NC zahrnuje mnoho paralelních algoritmů, které jsme poznali během semestru:

- paralelní prefixový součet (scan),
- technika Eulerovy cesty a odvozené algoritmy,
- paralelní třídění a výběr k-tého prvku nesetříděné posloupnosti,
- maticové operace (násobení matic, řešení řídkých soustav lineárních rovnic),

---

- paralelní kontrakce stromu a vyhodnocování výrazů,
- paralelní algoritmy pro souvislé komponenty grafu,
- paralelní algoritmy pro 2-souvislost a 3-souvislost,
- paralelní algoritmy pro mnoho problémů ve výpočetní geometrii,
- paralelní algoritmy pro vyhledávaní textových řetězců.

Diskuze: Klady a nedostatky teorie NC-třídy

- Třída NC je robustní vzhledem k použitému PRAM submodelu a vzhledem k jakémukoli přijatému modelu paralelních počítačů (APRAM, Boolean circuit model, BSP, LogP, atd).

- $\exists$ mnoho cenově-optimálních NC algoritmů:

  PŘÍKLAD. Paralelní prefixový součet: $T(n, p(n)) = O(\log n)$ pro $p(n) = n/\log n$.

- Třída NC může zahrnovat některé algoritmy, které nejsou efektivně paralelizovatelné, protože

  jakýkoli sekvenční algoritmus s logaritmickým časem je automaticky v NC bez ohledu na to, jak jej lze paralelizovat!!

  PŘÍKLAD. Binární hledání v seřazeném poli.

$$T(n,p) = \log_{p+1} n, \qquad C(n,p) = (p \log n)/\log(p+1), \qquad S(n,p) = \log(p+1),$$

což je na hony vzdáleno čemukoli, co lze považovat za vysoce paralelní řešení.

☹ $\exists$ mnoho problémů v P, pro které nejsou známy žádné NC algoritmy.

- Proto je teorie považuje za inherentně sekvenční $\equiv$ špatně paralelizovatelné.
- Nicméně, mnoho z nich je řešitelných v $T(n,p) = O(n^\epsilon)$, $0 < \epsilon < 1$.
- Ale funkce n^ϵ může být pomaleji rostoucí pro praktické hodnoty n než polylogaritmické funkce a asymptotické chování se začne prosazovat až pro neprakticky velká n. Např.,

$$\sqrt{n} < \log^3 n \quad \text{pro} \quad n \leq 0.5 \times 10^9.$$

- Takže,

$$\boxed{\text{algoritmy s } T(n,p) = O(n^\epsilon) \text{ mohou být v praxi rychlejší než mnohé } NC \text{ alg.}}$$

☹ NC-teorie předpokládá situace, kdy obrovský počítač (polynomiální počet procesorů, např., milióny) řeší velmi rychle (polylogaritmický čas, např., vteřiny) problémy rozumného rozsahu (např. stovky nebo tisíce vstupních položek).

- Na rozdíl od toho, v praxi počítače s nejvýše stovkami či tisíci procesory řeší rozsáhlé problémy (miliony vstupních položek).
- Čili, počet procesorů má spíše tendenci být subpolynomiální, typicky dokonce sublineární.

Vysoce paralelní redukce a P-úplnost

- Předpokládejme, že mnoho výzkumníků pracovalo velmi usilovně na nalezení vysoce paralelního algoritmu pro řešení nějakého polynomiálně složitého problému K a předpokládejme, že všichni tito výzkumníci byli neúspěšní.
- To povede k hypotéze, i když důkazem nepodpořené, že K je inherentně sekvenční.
- Předpokládejme, že existuje jiný problém K' v P, který také chceme paralelizovat, ale po jistém úsilí to vzdáme, protože žádné vysoce paralelní řešení není na dohled. Spíše opět pojmeme podezření, že i K' je inherentně sekvenční. Ale malé množství práce investované do paralelizování K' je slabým důkazem toho, že K' se opravdu nedá paralelizovat.
- Nicméně předpokládejme, že místo toho se nám podaří dokázat, že K lze zredukovat na K' (čili řešení K lze popsat jako algoritmus, ve kterém je jako podprogram volán algoritmus pro K') a že tento redukční alg. je sám o sobě vysoce paralelní, t.j., NC.
- To podpoří naší hypotézu, že K' je s vysokou pravděpodobností inherentně sekvenční, protože kdyby tomu tak nebylo, K by se stal problémem s vysokou paralelizovatelností, což by bylo v rozporu se současnou převládající zkušeností s K. Přesto však zde stále existuje malá šance, že K (a následně i K') má vysoce paralelní řešení. Je pouze smůla, že dosud nebylo nalezeno.
- Mnohem přesvědčivějším argumentem pro to, že K' je inherentně sekvenční, by byl důkaz, že libovolný známý obtížně paralelizovatelný polynomiální problém lze redukovat na K' vysoce paralelním způsobem. A přesně toto je myšlenka P-úplnosti.

Příklad

Problém Hodnota maximálního toku (MFV): (optimalizační problém)
Vstup: Graf G, jehož každá hrana e je ohodnocena kapacitou $c(e) \geq 0$, a dva vrcholy s, t.
Problém: Vypočtěte maximální tok $f(s, t)$ ze zdroje s do cíle t v G.

Problém Bit maximálního toku (MFB(i)): (rozhodovací problém)
Vstup: Graf G s hranami ohodnocenými kapacitami, vrcholy s, t, a kladné celé číslo i.
Problém: Je i-tý bit maximálního toku $f(s, t)$ roven 1?

Problém Práh maximálního toku (MFT(k)): (rozhodovací problém)
Vstup: Graf G s hranami ohodnocenými kapacitami, vrcholy s, t, a celé kladné číslo k.
Problém: Je maximální tok $f(s, t)$ menší než k?

Problém Struktura maximálního toku (MFP): (optimalizační problém)
Vstup: Graf G s hranami ohodnocenými kapacitami a vrcholy s, t.
Problém: Vypočtěte strukturu maximálního toku $f(s, t)$ v G.

MFB$(i) \leftrightarrow$ MFV

MFB(i) $\to$ MFV: triviální a vysoce paralelní: vyřešíme MFV a podíváme se na i-tý bit.

MFV $\to$ MFB(i): vysoce paralelní:

1. Vypočteme horní mez M hodnoty $f(s,t)$, např., sečtením kapacit všech hran incidentních s vrcholy s a t (což je vysoce paralelní výpočet). Nechť $m = \log M$.
2. Vyřešíme m instancí MFB(i), $i = 1, \ldots, m$, všechny paralelně.
3. Zkombinujeme výstupy MFB(i) do celočíselné hodnoty (vysoce paralelní výpočet).

Závěr: MFB(i) je vysoce paralelní $\iff$ MFV je vysoce paralelní.

Lemma 5. *Výpočet jakékoli funkce* $g : \{0,1\}^* \to \{0,1\}^*$ *indukuje přidružený*

bitový rozhodovací problém g_i*: "Je* i*-tý bit hodnoty* $g(x)$ *roven 1?"*

Jestliže hodnota $g(x)$ *je omezena, pak oba problémy jsou téže paralelní složitosti.*

MFT(k) $\leftrightarrow$ MFV

MFT(k) $\rightarrow$ **MFV:** triviální a vysoce paralelní: vyřešíme MFV a srovnáme výsledek s k.

MFV $\rightarrow$ **MFT**(k)**:** Předpokládejme, že $f(s,t)$ je opět omezen hodnotou M. Nechť $m = \log M$. Potřebujeme provést sekvenční binární prohledání, realizované jako posloupnost rekurzivních volání.
Zavolej MFT(2^{m-1}). Jestliže MFT(2^{m-1}) dá odpověď **NE**,
pak m-tý bit je 1 a voláme rekurzivně MFT($2^{m-1} + 2^{m-2}$)
jinak m-tý bit je 0 a voláme rekurzivně MFT(2^{m-2}).

- Těchto m volání nelze provést paralelně.
- Bez ohledu na to, jak je MFT(k) snadno paralelizovatelné a nezávisle na tom, kolik procesorů máme, musíme provést všech m volání sekvenčně, protože každé volání závisí na výsledku předchozího volání.
- **Závěr:** Dokonce i kdyby MFT(k) mělo vysoce paralelní řešení, redukce MFV na MFT(k) je inherentně sekvenční.

MFP

- MFP je relační problém: opakovaná provedení MFP mohou vrátit různé výsledky.
- Obvyklý přístup: požadovat splnění nějaké dodatečné jedinečné podmínky či vlastnosti, jako např. aby řešení bylo lexikograficky nejnižší.

NC-redukce

Definice 6. *Jazyk $L_1 \in P$ je NC-redukovatelný na jazyk $L_2 \in P$, $L_1 \leq_{nc} L_2$, jestliže*

1. *existuje funkce $f : \{0,1\}^* \to \{0,1\}^*$ taková, že $\forall\, x \in \{0,1\}^*$ $(x \in L_1 \iff f(x) \in L_2)$*
2. *f lze spočítat pomocí NC algoritmu.*

Lemma 7. *Jestliže $L_1 \leq_{nc} L_2$, pak*

1. *buď jestliže $L_2 \in NC$, pak $L_1 \in NC$,*
2. *nebo jestliže $L_1 \notin NC$, pak $L_2 \notin NC$.*

Důsledek 8. *Tudíž, můžeme*

1. *buď dokázat NC horní mez pro L_2, což nám dá NC horní mez pro L_1,*
2. *nebo dokázat polynomiální spodní mez pro L_1, což nám dá polynomiální spodní mez pro L_2.*

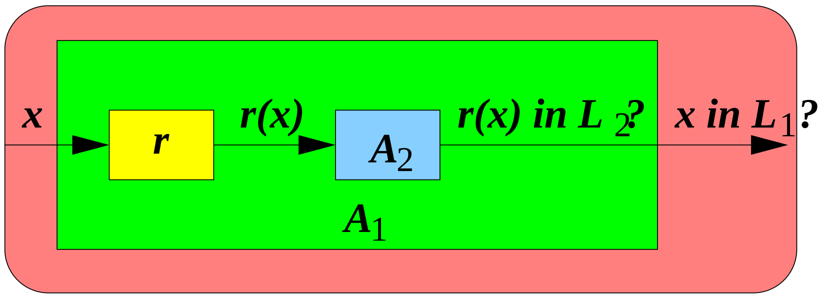
Lemma 9. *Jestliže $L_1 \leq_{nc} L_2$ a $L_2 \leq_{nc} L_1$, pak L_1 a L_2 mají tutéž možnost paralelizovatelnosti.*

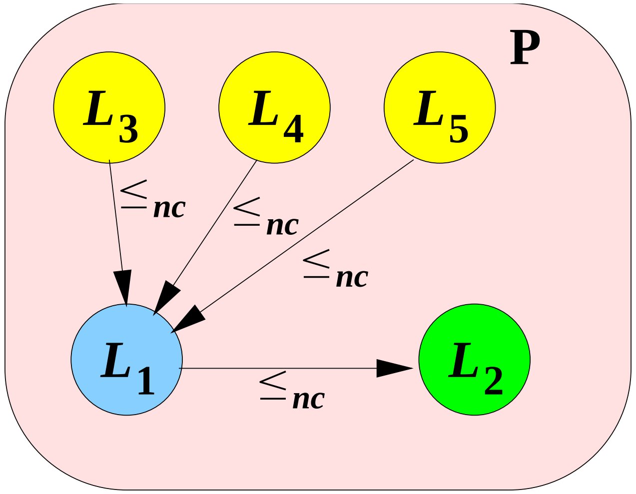
P-úplnost

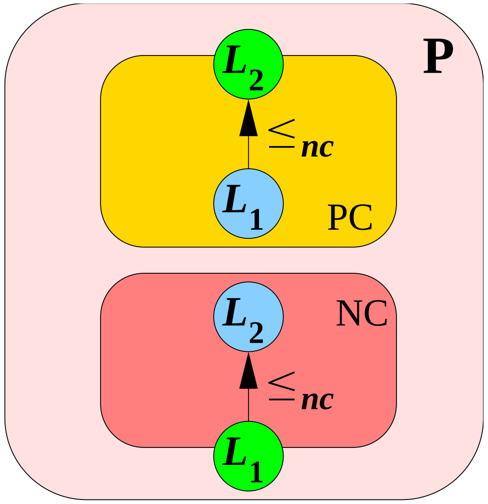
Definice 10. *Jazyk L je P-úplný, jestliže $L \in P$ a současně $\forall L' \in P; \quad L' \leq_{nc} L$.*

Lemma 11. *Třída NC je uzavřena shora vzhledem k relaci "je NC-redukovatelný na". (Jestliže $L_1 \in P$ a $\exists\, L_2 \in NC$ takový, že $L_1 \leq_{nc} L_2$, pak $L_1 \in NC$.)*

Lemma 12. *Třída P-úplných problémů je uzavřena zdola vzhledem k relaci "je NC-redukovatelný na". (Jestliže $L_2 \in P$ a $\exists\, P$-úplný L_1 takový, že $L_1 \leq_{nc} L_2$, pak L_2 je P-úplný.)*





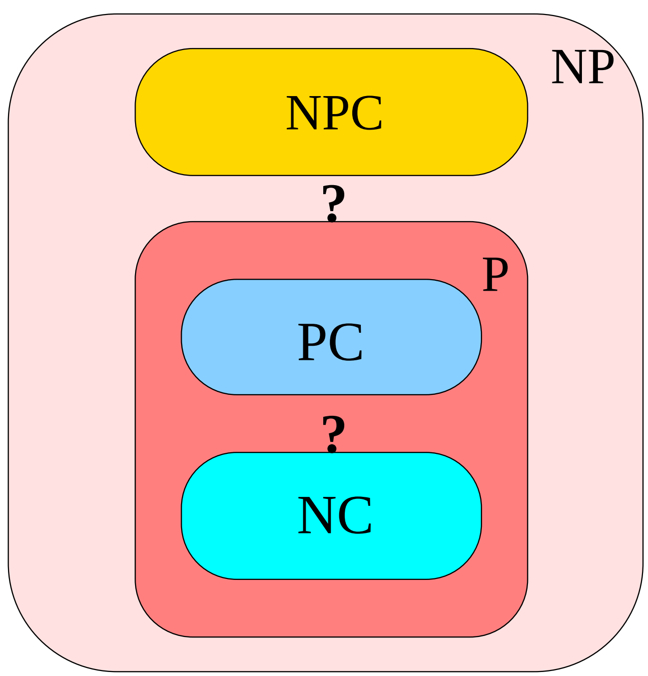


Důsledek 13.
$PC \cap NC \neq \emptyset \quad \Longrightarrow \quad NC = P.$
$PC - NC \neq \emptyset \quad \Longrightarrow \quad PC \cap NC = \emptyset.$

Převažující přesvědčení je, že platí druhý případ, i když zatím nebyl do současnosti podán žádný důkaz.

Existuje nápadná podoba mezi

- strukturou NP z hlediska sekvenční řešitelnosti a
- strukturou P z hlediska vysoce paralelní řešitelnosti.



Důkazy P-úplnosti

Abychom dokázali, že nějaký L je P-úplný, musíme ukázat, že

1) $L \in P$

a pak

2a) buď dokázat, že libovolný $L' \in P$ může být NC-redukován na L. (Tato práce musela být provedena pro první P-úplné problémy.)

2b) nebo najít vhodný známý P-úplný L' a

1. definovat zobrazení f instancí L' na instance L,
2. dokázat, že $x \in L' \iff f(x) \in L$ pro všechny $x \in \{0,1\}^*$,
3. dokázat, že f lze spočítat pomocí NC algoritmu.

P-úplný problém $\equiv$ inherentně sekvenční

CVP = základní P-úplný problém

CVP je jeden z historicky prvních P-úplných problémů, který hraje tutéž roli v paralelní teorie složitosti jako CIRCUIT-SAT v teorii NP-úplnosti.

Problém Hodnota Obvodu (CVP):

Vstup: Binární zakódování boolského kombinatorického obvodu C složeného z AND, OR, a NOT hradel s neomezeným počtem vstupů, vstupní hodnoty $x_1, x_2, \ldots, x_n$, a určený výstup y.

Problém: Je $y =$`true` pro vstupy $x_1, x_2, \ldots, x_n$?

Věta 14. [Ladner75] *CVP je P-úplný.*

Důkaz P-úplnosti: Velmi technický.
Pro jakýkoli binární jazyk L rozhodnutelný standardním Turingovým strojem v polynomiálním čase, zkonstruujeme kombinatorický obvod C_L takový, že

$$L \text{ je přijatý} \iff C_L \text{ dá na výstup hodnotu } \texttt{true}.$$

♣

P-úplné varianty CVP

Problém Hodnoty monotonního obvodu (MCVP): C se skládá pouze z AND a OR hradel.

Problém Hodnoty Alternujícího Monotonního Obvodu s 2-vstup. Hradly: C je monotonní a jakákoli cesta v C střídá AND a OR hradla. Vstupy musí být připojeny k OR hradlům a výstup obvodu musí vycházet z OR hradla. Hradla jsou 2-vstupová a 1-výstupová.

Problém Hodnoty Obvodu z NAND Hradel (NANDCVP): C se skládá pouze z NAND hradel s 2 vstupy.

Problém Hodnoty Obvodu z NOR Hradel (NORCVP): podobné.

Problém Hodnoty Planárního Obvodu (PCVP): C je planární boolský obvod.

Poznámka

Monotonní a současně planární CVP $\in NC$.

Příklady P-úplných problémů

Algebra

Problém Uzávěr (GEN):

Vstup: Konečná množina W, binární operace $\bullet$ na W (definovaná např. tabulkou), podmnožina $V \subset W$ a $w \in W$.

Problém: Je w prvkem množiny $V^* =$ nejmenšího transitivního uzávěru V vzhledem k operaci $\bullet$?

Důkaz P-úplnosti:

1. V^* lze zkonstruovat pomocí iterativního algoritmu v lineárním čase, tudíž GEN $\in P$.
2. MCVP lze NC-redukovat na GEN. ♣

Poznámka

Jestliže $\bullet$ je asociativní, pak GEN $\in NC$.

Formální jazyky

Problém Příslušnost do bezkontextového jazyka (CFLM):

Vstup: Bezkontextová gramatika $G = (N, T, P, S)$ a řetězec $x \in T^*$.
Problém: Je $x \in L(G)$?
Důkaz P-úplnosti: NC-redukcí GEN na CFLM.

1. Nechť $(W, \bullet, V, w)$ je instance problému GEN.
2. Zkonstruujme gramatiku $G = (W, \{a\}, P, w)$, kde
   $P = \{x \to yz \,|\, y \bullet z = x\} \cup \{x \to e \,|\, x \in V\}$.
3. Pak $e \in L(G) \iff w$ je generován uzavíráním V vůči $\bullet$.

CFLM $\in P$: standardní parsing algoritmus (např. Cocke-Kasami-Younger). ♣

Poznámka

Jestliže G je gramatika pevné velikosti (není zahrnuta do vstupu), pak CFLM $\in NC$.

Kombinatorická optimalizace

Problém Lineárních nerovností (LI):

Vstup: Celočíselná $n \times d$ matice A a celočíselný $n \times 1$ vektor b.

Problém: Určete, zda existuje racionální $d \times 1$ vektor $x > 0$ (všechny složky vektoru jsou nezáporné a aspoň jedna je nenulová) takový, že $Ax \leq b$. Není požadováno takový vektor nalézt.

Důkaz P-úplnosti: NC-redukcí CVP na LI. Předpokládejme, že je dán boolský obvod C a definujme transformaci obvodu C na množinu nerovností splňujících omezující podmínky problému LI takovou, že výstup obvodu C je `true` $\iff$ LI dá na odpověď `Ano`.

1. Jestliže vstup $x_i =$ `true` (`false`), přidej rovnici $x_i = 1$ ($x_i = 0$).
2. $\forall$ hradla $v = NOT(u)$, přidej nerovnosti $0 \leq v \leq 1$ a $v = 1 - u$.
3. $\forall$ hradla $w = AND(u, v)$, přidej nerovnosti $0 \leq w \leq 1$, $w \leq u$, $w \leq v$, a $u + v - 1 \leq w$.
4. $\forall$ hradla $w = OR(u, v)$, přidej nerovnosti $0 \leq w \leq 1$, $u \leq w$, $v \leq w$, a $w \leq u + v$.
5. Jestliže z je výstup C, přidej rovnici $z = 1$.

Je zřejmé, že jestliže LI dá odpověď `Ano`, pak $z =$`true`, jinak $z =$`false`. ♣

Problém Lineárních rovností (LE):

Vstup: Celočíselná $n \times d$ matice A a celočíselný $n \times 1$ vektor b.
Problém: Určete, zda existuje racionální $d \times 1$ vektor $x > 0$ takový, že $Ax = b$.
Důkaz P-úplnosti:
1) LE $\leq_{nc}$ LI, protože $Ax = b \iff Ax \leq b$ a $-Ax \leq -b$. Tudíž, LE $\in P$.
2) LI $\leq_{nc}$ LE. Přidej $(d+1)$-tou "výplňovou" proměnnou a převeď nerovnosti na rovnice. ♣

Problém Lineárního programování (LP):

Vstup: Celočíselná $n \times d$ matice A, celočíselný $n \times 1$ vektor b a celočíselný $1 \times d$ vektor c.
Problém: Nalezněte racionální $d \times 1$ vektor $x > 0$ takový, že $Ax \leq b$ a skalární součin $c \cdot x$ je maximalizován.
Důkaz P-úplnosti: Triviální, víme-li, že LE je P-úplný, neboť LE $\equiv$ LP s $c = 0$. ♣

Poznámky

1. **0-1 Celočíselné Programování (IP)**, kde $x =$ binární vektor, je NP-úplný problém.
2. LP $\in P$? byl otevřený problém mnoho let až do roku 1979 (Khachian).

Kombinatorická optimalizace (pokr.)

Problém Práh maximálního toku (MFT(k)):

Vstup: Graf G hranově ohodnocený nezápornými kapacitami, vrcholy s, t a přirozené číslo k.
Problém: Je maximální tok $f(s,t)$ větší nebo roven k?
Důkaz P-úplnosti: NC-redukcí NORCVP na MFT(k). ♣

Tudíž:

MFV, MFB(i) a MFP jsou P-úplné.

Lineární Algebra

Problém Gaussova eliminace s částečnou pivotizací (GEPP):

Vstup: $n \times n$ racionální matice A a celé číslo k.
Problém: Je-li na A použita Gaussova eliminace s částečnou pivotizací, je hodnota pivotu pro k-tý sloupec positivní?
Definice: Částečná pivotizace spočívá ve vyměňovaní řádek A tak, že největší hodnota v daném sloupci se dostane na diagonálu (abychom dosáhli numericky stabilní řešení systému rovnic).
Důkaz P-úplnosti: NC-redukcí NANDCVP na GEPP. ♣

Výpočetní geometrie

Problém Bod na konvexní obálce (PCH):

Vstup: Přirozené číslo d, množina S n bodů, a vybraný bod p v d-rozměrném prostoru.
Problém: Je p na konvexní obálce množiny S?
Důkaz P-úplnosti: NC-redukcí MCVP na PCH. ♣

Problém Lexikograficky první maximální nezávislá množina (LFMIS):

Vstup: Graf $G = (V, E)$ s očíslovanými vrcholy a $v \in V$.
Problém: Je v v lexikograficky první maximální nezávislé množině G?
Poznámky a definice:

- Nezávislá množina (IS) je množina vrcholů, ve které žádné dva vrcholy nejsou sousední v G.
- IS je největší, jestliže v G globálně neexistuje IS s větší kardinalitou.
- Nalezení největší IS je NP-úplný problém.
- IS je maximální, jestliže k ní nelze přidat žádný další vrchol G, aniž by se porušila nezávislost.
- Problém nalezení maximální IS je v P. Jestliže vrcholy jsou uspořádány podle ohodnocení $1, \ldots, n$, stačí procházet vrcholy a pokoušet se přidávat je k existující IS v tomto pořadí.
- Je jasné, že graf může mít maximálních IS více, různých velikostí. Abychom je mohli lexikograficky porovnat, stačí uvést vrcholy maximálních IS v rostoucím pořadí $v_1, \ldots, v_k$, $v_1 < v_2 < \ldots v_k$, a zacházet s takovou posloupností jako s číslem.

Důkaz P-úplnosti: NC-redukcí NORCVP na LFMIS. ♣

Logické programování

Problém Unifikace (UP):

Vstup: Dvě logické formule s a t s termy složenými z proměnných a funkčních symbolů.
Definice: Substituce za proměnnou x v termu u termem w je nahrazení všech výskytů x v u termem w.
Problém: Existuje posloupnost substitucí z taková, že s a t jsou unifikovány, t.j., $z(s) = z(t)$?
Důkaz P-úplnosti: NC-redukcí NORCVP na UP.

Poznámka

Matching problém: "Existuje substituce z taková, že $z(s) = t$?" je v NC.

Odkaz na literaturu

R. Greenlaw, et al. *Limits to Parallel Computation*, Oxford University Press, 1995, ISBN 0-19-508591-4.