Časová a paměťová složitost

ZPRACUJE: Mystik

Blumův teorém? Ostrost hierarchie tříd složitosti? časově/prostorově konstruovatelné funkce?

Obsah

Složitost

Složitost
: vyjádření použitých zdrojů jako funkce závisející na velikosti vstupu

Určujeme:

- složitost nejhoršího případu (nejčastější)
- nejlepšího případu
- průměrného případu (vážený průměr dle pravděpodobnosti)

Časová složitost
: počet kroků výpočtu (cena kroků může být shodná nebo i rozdílná)

Paměťová složitost
: potřebný počet buněk pásky (nepočítáme vstup)

Asymptotické omezení složitosti
: pro dostatečně velké vstupy se složitost stále více blíží k asymptotické složitosti (s tolerancí danou násobením nějakými konstantami)

- **Asymptotické horní omezení** funkce $f(n)$ je množina funkcí:

$$O(f(n)) = \{g(n) \mid \exists c, n_0 \in \mathbb{N}\ \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq g(n) \leq c \cdot f(n)\}$$

- **Asymptotické dolní omezení** funkce $f(n)$ je množina funkcí:

$$\Omega(f(n)) = \{g(n) \mid \exists c, n_0 \in \mathbb{N}\ \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq c \cdot f(n) \leq g(n)\}$$

- **Asymptotické oboustranné omezení** funkce $f(n)$ je množina funkcí:

$$\Theta(f(n)) = \{g(n) \mid \exists c_1, c_2, n_0 \in \mathbb{N}\ \forall n \in \mathbb{N} : n \geq n_0 \Rightarrow 0 \leq c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n)\}$$

- Složitost výpočtu TS není příliš odlišná od současných počítačů
- Je-li prostorová složitost n pak časová složitost nemůže být nižší než n+1 (jednoduše by nebylo kdy toho tolik zapsat)
- Je-li jazyk přijímán vícepáskovým TS v čase t(n) je také přijímán jednopáskovým TS v čase $t(n)^2$

Nedeterminismus a složitost

Nedeterministický TS lze simulovat deterministickým TS , ale jen za cenu exponenciálního nárůstu času.

Nedeterminismus nepřináší nic z pohledu vyčíslitelnosti, ale výrazně ovlivňuje složitost

P = NP ?
: nejslavnější problém současné teoretické informatiky
: pokud by se ukázalo, že libovolný NP-úplný jazyk je ve třídě P pak by muselo platit P = NP, naopak důkaz, že některý z NP-plných jazyků leží mimo P by znamenal $P \subset NP$

Třídy složitosti

Hierarchie

(od nejsložitějších k nejméně složitým)

Vrchol hierarchie

- **RE**
- **R**
- **PR**

Třídy nad exponenciální složitostí

- **ELEMENTARY**
- **k-NEXP**
- **k-EXP**

Exponenciální třídy

- **EXPSPACE = NEXPSPACE**
- **NEXP**
- **EXP**

Polynomiální třídy

(jsou v ní všechny prakticky dobře řešitelné problémy)

- **PSPACE = NPSPACE**
- **NP**
- **P**

Logaritmické třídy

- **NLOGSPACE**
- **LOGSPACE**
- **LOG**

- **Konstantní složitost**

(nezávisí na délce vstupu)

Vysvětlení značení:

- **...** - deterministický
- **N...** - nedeterministický

- **...SPACE** - prostorová složitost
- jinak - časová složitost

- **RE** - rekurzivně vyčíslitelné
- **R** - rekurzivní
- **PR** - primitivně rekurzivní
- **ELEMENTARY** - nekonečná věž exponenciál
- **k-EXP** - věž exponenciál o výšce k ($2^{2^{\cdots^{2^{2^n}}}}$)
- **EXP** - exponenciální závislost (2^{n^k})
- **P** - polynomická závislost (n^k)
- **LOG** - logaritmická závislost ($klog_x(n)$)

  (pozn.: n - délka vstupu, k - konstanta)

Redukce a úplnost

$\mathcal{R}$-redukce

jazyk L_1 je $\mathcal{R}$-redukovatelný na L_2 jestliže existuje funkce f třídy $\mathcal{R}$ taková, že $w \in L_1 \Leftrightarrow f(w) \in L_2$ - značíme $L_1 \leq^m_{\mathcal{R}} L_2$

(tj. jde o převod jednoho problému na problém jiný převodní funkcí, která má složitost $\mathcal{R}$)

Pokud je L_1 P-redukovatelný na L_2, který je ve třídě P, pak i L_1 je ve třídě P. Pokud je L_1 P-redukovatelný na L_2, který není ve třídě P, pak i L_1 není ve třídě P.

Jazyk C-těžký (angl. C-hard)

Jazyk L_0 je C-těžký vzhledem k $\mathcal{R}$-redukci jestliže $\forall L \in C : L \leq^m_{\mathcal{R}} L_0$

(tj. všechny jazyky třídy C lze na tento jazyk redukovat)

Jazyk C-úplný (angl. C-full)

Jazyk L_0 je C-úplný pokud je C-těžký a patří do třídy C

Nejběžnější typy redukce a úplnosti:

- jazyky NP, PSPACE, EXP jsou P-úplné
- jazyky P, NLOGSPACE jsou DLOGSPACE-úplné
- jazyky NEXP jsou EXP-úplné

P-redukce odpovídá realizovatelné převoditelnosti problémů

SAT - problém

(problém splnitelnosti booleovských formulí)

SAT - problém

je dána množina proměnných V a množina klauzulí v konjunktivní normální formě nad V. Je tato množina klauzulí splnitelná?

- **SAT-problém je NP-úplný vzhledem k P-redukci**
- první problém, jehož NP-úplnost vzhledem k P-redukci byla dokázána
- lze zkonstruovat NTS, který přijímá SAT v P čase

Cookův teorém

libovolný NP problém lze P-redukovat na SAT-problém

Kategorie: Státnice 2011 | Teoretická informatika