Složitost

Od Turingových strojů k P=NP

Zbyněk Konečný

Zimnění 2011

12.–16.2.2011

O čem to dnes bude?

1. Co to je složitost
2. Výpočetní modely
3. Vyčíslitelnost
4. Složitost na TS
5. Složitostní třídy

Problém, jazyk

- rozhodovací problém: máme rozhodnout, jestli má nějaký objekt nějakou vlastnost (Je výroková formule splnitelná? Lze graf obarvit čtyřmi barvami?)
- výpočetní problém: máme na základě jednoho objektu najít jiný objekt (Co dosadit do formule, aby byla splněna? Jak obarvit graf?)
- na této přednášce budeme mluvit pouze o rozhodovacích
- jazykem nazýváme množinu všech kódů (binárních, nebo nad jinou konečnou abecedou) objektů, které mají nějakou vlastnost
- rozpoznávat jazyk = řešit rozhodovací problém
- instance problému = konkrétní objekt, o němž se má rozhodnout

Složitost

- složitost algoritmu = množství zdrojů spotřebovanéné algoritmem k vyřešení problému v závislosti na velikosti instance
- složitost problému = množství zdrojů potřebné k vyřešení problému v závislosti na velikosti jeho instance
- typicky čas (počet operací) a prostor (množství pamětí)
- závisí na výpočetním modelu

Turingův stroj

- chceme počítat provedené instrukce
- běžný počítač: 32b / 64b, RISC/CISC ...
- pro formální popis je lepší Turingův stroj (dále TS)
- má jednu "proměnnou" ve které je uložen stav, a nekonečně dlouhou pásku, na které je ze začátku vstup
- na pásku může zapisovat pouze symboly dané abecedy
- na začátku pásky je nepřepsatelné δ
- "program"je funkce, která symbolu na pásce a stavu přiřadí nový symbol, nový stav a posun na pásce
- dva speciální stavy: přijato a zamítnuto

TS a jazyky

- TS částečně rozpoznává jazyk L pokud nad jeho slovy skončí ve stavu přijato, pro ostatní slova se zacyklí nebo skončí ve stavu zamítnuto
- TS rozpoznává jazyk L pokud nad jeho slovy skončí ve stavu přijato a nad ostatními ve stavu zamítnuto

Příklady

- najděte TS pro binární zápisy čísel dělitelných 7
- najděte TS pro jazyk všech slov, která obsahují stejně znaků A, B a C

Alternativní definice

- více hlav
- více pásek
- oboustranná páska
- pevná abeceda $\delta, 0, 1, -$
- pouze jeden stav (pak je potřeba velká abeceda)

Slabší modely

- stroj má dvě pásky, na výstupní pásce musí hlava vždy ukazovat na poslední symbol různý od - (zásobníkový automat)
- stroj smí pouze číst a hlavou pohybvat pouze doprava (konečný automat)
- omezení na počet operací / délku popsané pásky

Další modely

- nedeterministické Turingovy stroje
- běžné počítače + programovací jazyky (C, Pascal, Java)
- while-programy
- Random Access Machine
- paralelní počítače
- kvantové počítače
- všechny tyto modely umí řešit stejné problémy, ale různě rychle

Trocha teorie množin

- Turingovy stroje lze očíslovat (je jich spočetně)
- jazyky očíslovat nelze (je jich stejně jako reálných čísel)
- existují jazyky, které nelze částečně rozpoznat
- ostatní nazýváme "rekurzivně spočetné" (Re)

Halting problem

- existuje program, který by dle zdrojáku určil, zda program pro daný vstup skončí?
- jistě lze rozpoznat částečně (simulace)
- předpokládejme, že halt vstup (i,x) akceptuje, pokud TS číslo i nad vstupem x skončí; jinak zamítá
- vyrobme TS confuse, který vstup i zamítá, pokud halt akceptuje (i,i); jinak confuse cyklí
- nechť e je číslo programu *confuse*. Pokud $\mathit{confuse}(e) = 0$, pak $\mathit{halt}(e, e) = 0$, tedy $\mathit{confuse}(e)$ cyklí.
- pokud $\mathit{confuse}(e)$ cyklí, pak $\mathit{halt}(e, e) \neq 0$. To znamená, že $\mathit{halt}(e, e) = 0$, spor.

Rekurzivní jazyky

- některé jazyky (např. jazyk čísel všech TS, které necyklí) jsou jen částečně rozpoznatelné
- ostatní nazýváme "rekurzivní" (Rec)
- doplněk rekurzivního jazyka je jistě rekurzivní
- doplněk rekurzivně spočetného L nemusí je rekurzivně spočetný, právě když je L rekurzivní

Složitost na TS

- časová složitost TS = maximální možný počet přechodů TS (v závislosti na délce vstupu)
- pro nedeterministický stroj délka nejkratšího možného výpočtu
- časová složitost problému = časová složitost nejlepšího TS, který problém řeší
- uvažujeme TS se dvěma páskami, první je read only
- prostorová složitost TS = maximální počet polí druhé pásky, který je třeba popsat (v závislosti na délce vstupu)
- prostorová složitost problému = opět optimum
- při nedeterminismu opět minimum přes výpočty

Landauova notace

- $f \in O(g)$ – funkce f roste nejvýše tak rychle, jako g (jejich poměr neroste do nekonečna)
- $f \in o(g)$ – funkce f roste výrazně pomaleji, než g (jejich poměr klesá do nuly)
- $f \in \theta(g)$ – platí oboje výše uvedené
- více nž

Složitostní třídy dané konkrétní fcí

- f je funkce
- $DTIME(f), DSPACE(f), NTIME(f), NSPACE(f)$ značí množiny problémů s časovou / prostorovou složitostí v $O(f)$ a to na deterministickém / nedeterministickém stroji
- $DTIME(f) \subseteq DSPACE(f)$, $NTIME(f) \subseteq NSPACE(f)$
- $DSPACE(f) \subseteq DTIME(2^f)$, $NSPACE(f) \subseteq NTIME(2^f)$

Savitchova věta

- $NSPACE(f) \subseteq DSPACE(f^2)$
- myšlenka: máme orientovaný graf konfigurací TS (až 2^f vrcholů), hledáme cestu
- rekurze, půlení intervalů

Obecné složitostní třídy

- P – polynomiální časová složitost
- NP – nedeterministická obdoba P
- $PSPACE$ – polynomiální časová složitost
- $NPSPACE$ – nepoužívá se (dle Savithovy věty splývá s $PSPACE$
- E, NE – časová složitost v $2^{O}(n)$
- $EXPTIME$, $NEXPTIME$ – časová složitost v $2^{O}(n^k)$ pro nějaké k
- $EXPSPACE$

P=NP

- jeden z Problémů milénia (odměna milion USD)
- "In a 2002 poll of 100 researchers, 61 believed the answer to be no, 9 believed the answer is yes, and 22 were unsure; 8 believed the question may be independent of the currently accepted axioms and so impossible to prove or disprove."
- spousta pokusů o důkaz (poslední seriózně vypadající v létě 2010)

NP-úplné problémy

- problém A se polynomiálně redukuje na problém B pokud existuje turingův stroj řešící A tak, že udělá polynomiálně mnoho operací a polynomiálně mnohokrát odsimuluje stroj B nad vstupem, který má polynomiální délku vzhledem k původnímu vstupu
- NP-těžký problém je takový, že se na něj redukují všechny NP problémy
- NP-úplný problém je NP těžký NP problém
- problém splnitelnosti (SAT) je NP-úplný

3-SAT

- formule v konjunktivní normální formě, pouze 3 literály (proměnná nebo neproměnná) v každé klauzuli
- SAT se redukuje na 3-SAT
- 2-SAT je v P

Subset sum

- Umíme z dané množiny vybrat podmnožinu s pevně daným součtem?

Hamiltonovský cyklus

- Existuje v grafu cyklus který prochází každým vrcholem právě jednou.

Další možná rozšíření P

- ZPP – algoritmus dostává náhodná data (krom vstupu), nulová pravděpodobnost chyby, průměrná doba běhu je polynomiální (pod NP)
- RP – algoritmus dostává náhodná data (krom vstupu), v případě záporné odpovědi nulová pravděpodobnost chyby, v případě záporné omezená pravděpodobnost chyby. Průměrná doba běhu je polynomiální (pod NP)
- BPP – algoritmus dostává náhodná data (krom vstupu), omezená pravděpodobnost chyby
- BQP – používá kvantové výpočty, jinak totéž, co BPP