Algoritmizace

složitost rekurzivních algoritmů

Jiří Vyskočil, Marko Genyg-Berezovskyj

2010

Vyjádření složitosti rekurzivního algoritmu rekurentním tvarem

- Příklad vyjádření složitosti rekurzivního algoritmu rekurencí:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \,, \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{if } n > 1 \,. \end{cases}$$

  Kde $T(n)$ je celková složitost algoritmu.

  Na pravé straně jsou jednotlivé případy složitostí pro různá n.

- Okrajové případy (pro n < konstanta) můžeme opomenout, protože mají konstantní asymptotickou složitost. Zaokrouhlení rovněž většinou neovlivní celkový výsledek (Pozor existují i výjimky!).

  Z toho dostáváme rekurentní vztah:

$$T(n) = 2T(n/2) + \Theta(n) \,,$$

Převod rekurence na přímé vyjádření

- Přímým vyjádřením složitosti myslíme vyjádření složitosti bez rekurence.
  - Např: $T(n) = \Theta(\log(n))$
- Jaké jsou možnosti řešení?
  - **Substituční metoda**
    - „Uhádneme“ řešení a potom dokážeme, že je správné indukcí.
  - **Metoda rekurzívního stromu**
    - Spočítáme složitost celého rekurzivního stromu.
  - **Použití „kuchařky“** (Master theorem – mistrovská věta)
    - Pro některé speciální tvary rekurentních vztahů známe předem vypočítané řešení dle mistrovské věty.

Substituční metoda

- Řešíme ve dvou krocích
  1. **Odhadneme přesný tvar řešení.**
     - Odhad lze stanovit například pomocí zjišťováním složitosti pro různá vstupní n.
  2. **Matematicky dokážeme, že je náš odhad správný.**
     - Obvykle se dokazuje pomocí matematické indukce.
- Metoda bývá zpravidla velmi účinná.
- Její nevýhodou je určování přesného tvaru řešení v kroku 1 pro které neexistuje obecný postup.

Substituční metoda - příklad

- Příklad:

  $T(n) = 2T(n/2) + n$

- Předpokládejme, že jsme odhadli přímé vyjádření vztahem:

  $T(n) = \mathrm{O}(n \log(n))$

- Z definice horního odhadu O, chceme tedy dokázat, že

  $T(n) \leq cn \log(n)$

  pro nějaké vhodné $c > 0$.

- Nyní stanovíme vhodný indukční předpoklad (tj. nechť odhad platí pro $n/2$):

  $T(n/2) \leq c(n/2) \log(n/2)$

Substituční metoda - příklad

- Nyní dosadíme indukční předpoklad do rekurentního vztahu a pokusíme se dokázat jeho platnost vyjádřením přímého (nerekurentního) původně odhadnutého vztahu pro n.

$$
\begin{aligned}
T(n) &\le 2(c\,(n/2)\log(n/2)) + n \\
&\le cn\log(n/2) + n \\
&= cn\log(n) - cn\log(2) + n \\
&= cn\log(n) - cn + n \\
&\le cn\log(n)
\end{aligned}
$$

  kde poslední krok platí pro $c \ge 1$.

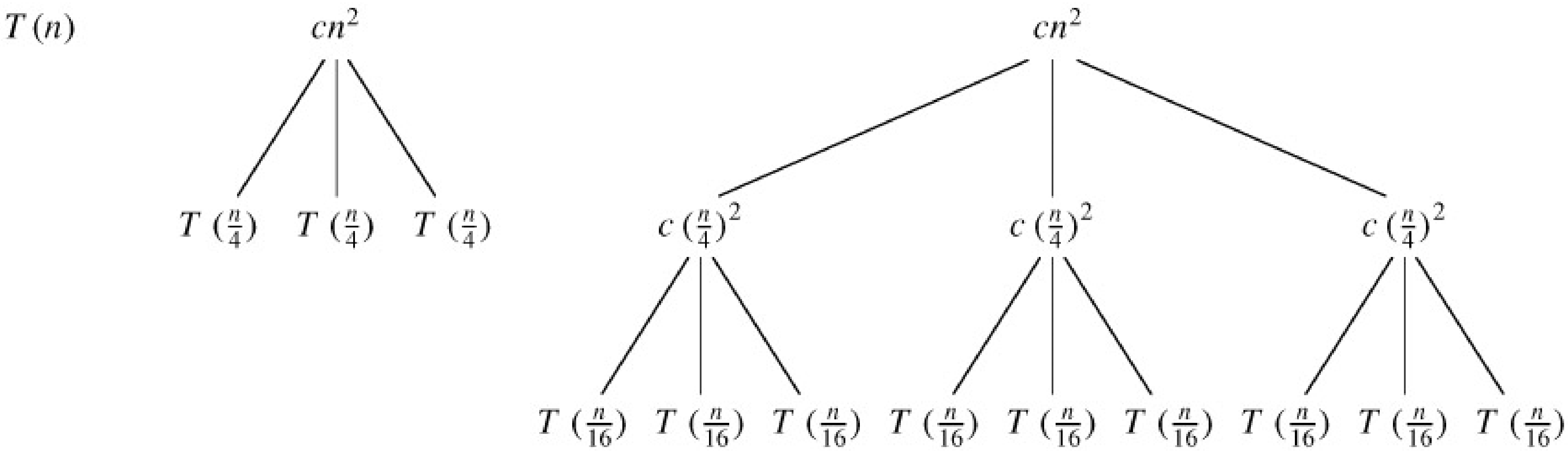
- Počáteční krok indukce platí triviálně. Díky asymptotické notaci stačí ukázat, že odhad platí pro nějaké n_0 a $c > 0$. V našem příkladě tedy platí pro $n_0=3$ a $c \ge 2$).
- Tím je důkaz hotov.

Metoda rekurzívního stromu - příklad

- Příklad:

  $T(n) = 3T(n/4) + cn^2$

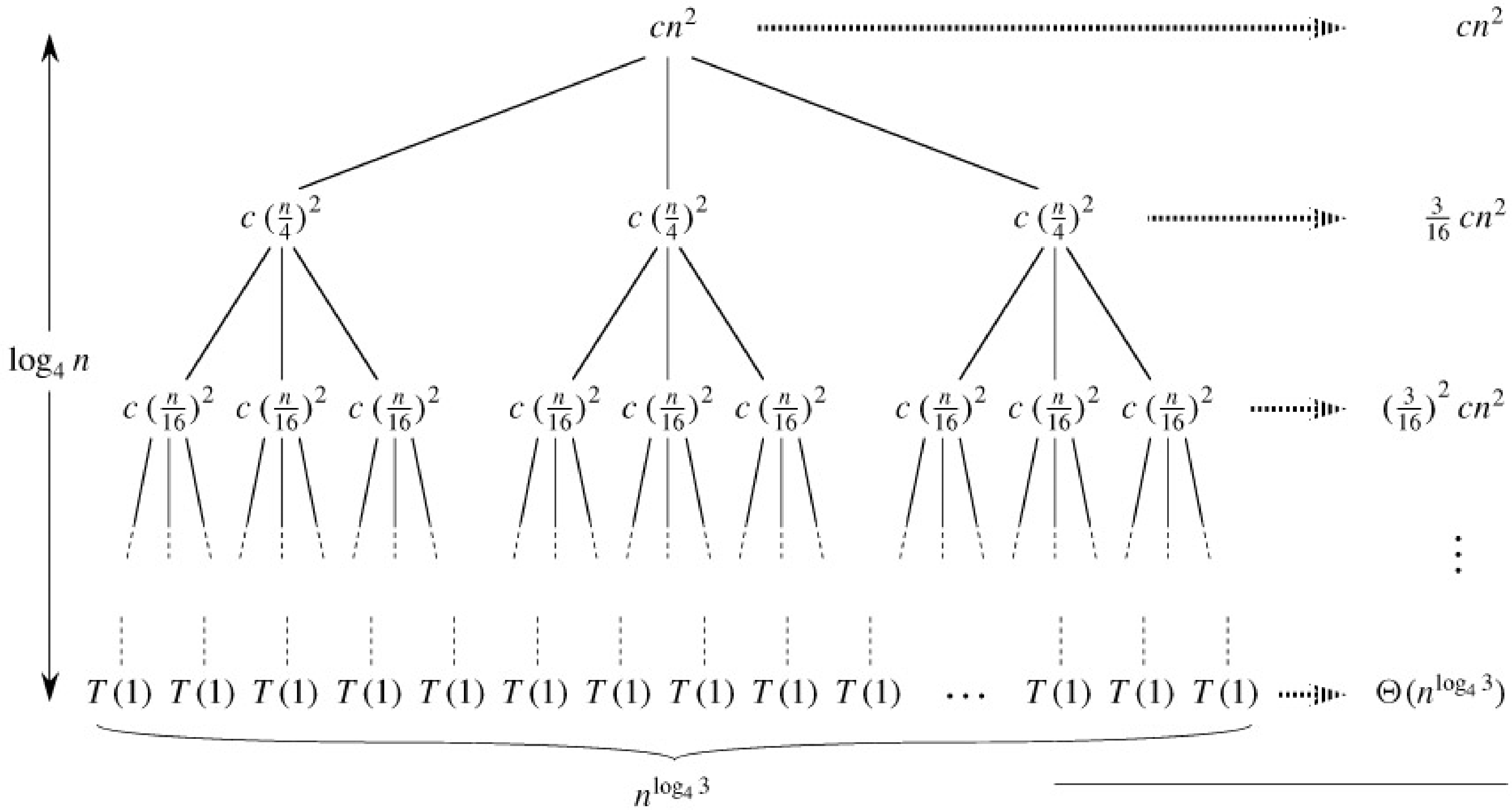
- Iterativně rozkládáme do rekurzivních stromů:



- Pro každý strom platí, že součet všech uzlů dá složitost $T(n)$ podle původního rekurentního vztahu.
- Rekurzivní stromy jsou pouze grafická vizualizace rozvoje rekurentního vztahu.

Metoda rekurzívního stromu - příklad

- Výsledný strom má následující tvar:



- Vyjádříme součty jednotlivých pater stromu.
- Všechna patra sečteme a dostaneme výslednou složitost:

$O(n^2)$

Metoda rekurzívního stromu - příklad

- Součet pater lze spočítat následovně:

$$
\begin{aligned}
T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n-1} cn^2 + \Theta(n^{\log_4 3}) \\
&= \sum_{i=0}^{\log_4 n-1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&< \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{1}{1-(3/16)} cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{16}{13} cn^2 + \Theta(n^{\log_4 3}) \\
&= O(n^2) .
\end{aligned}
$$

podle vzorce $\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$

pro $|x| < 1$

Použití „kuchařky“

- Použití „kuchařky“ nebo tzv. mistrovské věty (master theorem) řeší rekurentní složitost, která má následující tvar:

$$T(n) = a\,T(n/b) + f(n)$$

Kde $a \geq 1$ a $b > 1$ jsou konstanty

a $f(n)$ je asymptoticky kladná funkce.

- Zaokrouhlení u členu $T(n/b)$ na $T(\lfloor n/b \rfloor)$ nebo $T(\lceil n/b \rceil)$ neovlivní v tomto případě výslednou složitost.

Použití „kuchařky“

- **Master theorem** (mistrovská nebo také kuchařková věta)
  - Nechť jsou $a \geq 1$ a $b > 1$ konstanty, nechť je $f(n)$ funkce a nechť $T(n)$ je definováno pro nezáporná celá čísla rekurencí

$$T(n) = a\,T(n/b) + f(n)$$

kde n/b má význam buď $\lceil n/b \rceil$ nebo $\lfloor n/b \rfloor$. Potom lze asymptoticky vyjádřit následovně:

1. Pokud $f(n) \in O(n^{\log_b(a)-\varepsilon})$ pro nějakou konstantu $\varepsilon > 0$, potom

$$\boldsymbol{T(n) \in \Theta(n^{\log_b(a)})}.$$

2. Pokud $f(n) \in \Theta(n^{\log_b(a)})$, potom

$$\boldsymbol{T(n) \in \Theta(n^{\log_b(a)} \log(n))}.$$

3. Pokud $f(n) \in \Omega(n^{\log_b(a)+\varepsilon})$ pro nějakou konstantu $\varepsilon > 0$ a pokud

$a\,f(n/b) \leq c\,f(n)$ pro nějakou konstantu $c < 1$ a všechna dostatečně velké n, potom

$$\boldsymbol{T(n) \in \Theta(f(n))}.$$

Použití „kuchařky" – příklad 1

- Příklad 1:

  $T(n) = 9T(n/3) + n$

- Z toho dostáváme, že $a = 9$, $b = 3$, $f(n) = n \in O(n^{\log_3(9)-1})$.

  Jedná se tedy o případ číslo 1.

- Dostáváme tedy složitost:

  $T(n) \in \Theta(n^{\log_3(9)}) = \boldsymbol{\Theta(n^2)}$

Použití „kuchařky“ – příklad 2

- Příklad 2:

  $T(n) = T(2n/3) + 1$

- Z toho dostáváme, že $a = 1$, $b = 3/2$,

  $f(n) = 1 = n^{\log_{3/2}(1)} \in \Theta(n^{\log_{3/2}(1)})$.

  Jedná se tedy o případ číslo 2.

- Dostáváme tedy složitost:

  $T(n) \in \Theta(n^{\log_{3/2}(1)} \log(n)) = \mathbf{\Theta(\log(n))}$

Použití „kuchařky“ – příklad 3

- Příklad 3:

  $T(n) = 3T(n/4) + n\log(n)$

- Z toho dostáváme, že $a = 3$, $b = 4$,

  $f(n) = n\log(n)$ a víme, že $n^{\log_4(3)} = O(n^{0.793})$.
  Platí tedy, že $f(n) \in \Omega(n^{\log_4(3)+0.2})$.

  Pokud by se mělo jednat o případ 3 musí ještě platit pro $c < 1$ a všechna dostatečně velká n, že $a f(n/b) \le c f(n)$ tedy $a f(n/b) = 3(n/4)\log(n/4) \le (3/4)n\log(n) = c f(n)$ pro $c = 3/4$.

- Dostáváme tedy složitost:

  $T(n) \in \mathbf{\Theta(n\log(n))}$