Rekurentní rovnice, strukturální indukce

Backusova-Naurova forma

Například syntaxe formulí výrokové logiky

$$\varphi ::= a \mid \mathtt{tt} \mid (\varphi \wedge \varphi) \mid (\neg \varphi)$$

kde $a \in At$.

Poznámky

1. Relaxace BNF.
2. Původ notace: John Warner Backus a Peter Naur pro popis syntaxe jazyka ALGOL 60.
   Pravidla stejné vyjadřovací schopnosti: hindský gramatik Pāṇini (cca 6. století př.n.l.) pro popis sanskrtu.
   První formální popis přirozeného jazyka: 3 959 veršů díla *Astādhyāyī*.

Jiné zápisy BNF

$$\frac{}{a} \;\Big|\; \frac{}{\mathtt{tt}} \;\Big|\; \frac{\varphi_1 \quad \varphi_2}{(\varphi_1 \wedge \varphi_2)} \;\Big|\; \frac{\varphi}{(\neg\varphi)}$$

nebo

$$\frac{}{a}(\text{atom}) \;\Big|\; \frac{}{\mathtt{tt}}(\text{true}) \;\Big|\; \frac{\varphi_1 \quad \varphi_2}{(\varphi_1 \wedge \varphi_2)}(\text{and}) \;\Big|\; \frac{\varphi}{(\neg\varphi)}(\text{not})$$

Axiomy: (atom), (true). Deduktivní pravidla: (and), (not).

Syntaktický strom (Parsing Tree)

$$\cfrac{\cfrac{\cfrac{}{a}(\text{atom}) \qquad \cfrac{}{b}(\text{atom})}{(a \wedge b)}(\text{and})}{(\neg(a \wedge b))}(\text{not})$$

Jazyk generovaný gramatikou

Abeceda: $\Sigma = At \cup \{\mathtt{tt}, \wedge, \neg, (,)\}$.

Množina F všech formulí je $F \subseteq \Sigma^*$.

F je induktivně generovaná "gramatikou" (atom), (true), (and), (not).

Pro každé $\varphi \in \Sigma^*$ platí

$$\varphi \in F \quad \text{iff} \quad \text{existuje parsing tree (dané "gramatiky").}$$

Z toho plyne další indukční princip!

Příklad

Pro každé $\varphi \in F$ platí: φ má stejný počet pravých a levých závorek.

Řešení:

1. Tvrzení platí pro závěr každého z axiomů (atom), (true).
2. Pro pravidlo (and):
   Jestliže tvrzení platí pro každý předpoklad pravidla (and), pak tvrzení platí i pro závěr pravidla (and).
3. Pro pravidlo (not):
   Jestliže tvrzení platí pro každý předpoklad pravidla (not), pak tvrzení platí i pro závěr pravidla (not).

Podle principu strukturální indukce jsme hotovi.

Příklad

Pro každé $\varphi \in F$ platí: φ má stejný počet pravých a levých závorek.

Řešení:

1. Tvrzení platí pro závěr každého z axiomů (atom), (true).
2. Pro pravidlo (and):
   Jestliže tvrzení platí pro každý předpoklad pravidla (and), pak tvrzení platí i pro závěr pravidla (and).
3. Pro pravidlo (not):
   Jestliže tvrzení platí pro každý předpoklad pravidla (not), pak tvrzení platí i pro závěr pravidla (not).

Podle principu strukturální indukce jsme hotovi.

Princip strukturální indukce

Ať Σ je libovolná konečná abeceda. Ať G je konečná sada odvozovacích pravidel, která induktivně zadává množinu slov $L \subseteq \Sigma^*$.

Ať A je množina všech axiomů z G. Ať D je množina všech deduktivních pravidel z G.

Ať V je nějaká vlastnost slov nad abecedou Σ. K tomu, abychom ukázali, že každé slovo v množině L má vlastnost V, stačí ukázat:[a]

1. Základní krok: Závěr každého axiomu z množiny A má vlastnost V.
2. Indukční krok: Pro každou instanci libovolného deduktivního pravidla v množině D platí:

   > Jestliže všechny předpoklady pravidla mají vlastnost V, potom i závěr tohoto pravidla má vlastnost V.

[a] Tomu se říká: Vlastnost V je invariantní na průchod gramatikou G.

Platí:

1. Jestliže platí (silný nebo slabý) princip indukce, platí i princip strukturální indukce.
2. Pro každou neprázdnou abecedu Σ platí: existuje množina $M \subseteq \Sigma^*$, kterou nelze zadat induktivně.

Další poznatky: skripta a sbírka řešených příkladů.

Platí:

1. Jestliže platí (silný nebo slabý) princip indukce, platí i princip strukturální indukce.
2. Pro každou neprázdnou abecedu Σ platí: existuje množina $M \subseteq \Sigma^*$, kterou nelze zadat induktivně.

Další poznatky: skripta a sbírka řešených příkladů.

Příklad (Parketáž triminy z minulé přednášky)

$P(n)$ = počet parket k vyparketování místnosti rozměru n

1. $P(1) = 1$.
2. $P(n+1) = 1 + 4 \cdot P(n)$, $n \geq 1$.

Čili:

1. $P(n+1) - 4 \cdot P(n) = 1$, $n \geq 1$ (rekurentní rovnice).
2. $P(1) = 1$ (počáteční podmínka).

Příklad (složitost algoritmu Bubblesort)

Označte $C(n)$ počet porovnání v segmentu (pseudo)kódu

```
for i:=1 to n
    for j:= i+1 to n
        if A[i] > A[j] then swap(A[i],A[j]) endif
    endfor
endfor
```

Potom platí:

$$C(n) = (n-1) + (n-2) + \cdots + 1 + 0 = \sum_{k=0}^{n-1} k, \quad n \geq 1.$$

Tedy $C(1) = 0$, $C(n+1) = C(n) + n$, $n \geq 1$.

Definice

Lineární rekurentní rovnice k-tého řádu s konstantními koeficienty je zápis

$$a_k X(n+k) + a_{k-1} X(n+k-1) + \cdots + a_0 X(n) = f(n)$$

kde $a_k \neq 0$.

Terminologie:

- Koeficienty: (reálná nebo komplexní) čísla a_k, $a_{k-1}, \ldots, a_0$
- Pravá strana: posloupnost $f(n)$
- Příslušná homogenní rovnice: $a_k X(n+k) + a_{k-1} X(n+k-1) + \cdots + a_0 X(n) = 0$
- Charakteristická rovnice: $a_k \lambda^k + a_{k-1} \lambda^{k-1} + \cdots + a_0 = 0$

Kompletní řešení homogenní rovnice

1. Vyřešíme charakteristickou rovnici
   $a_k\lambda^k + a_{k-1}\lambda^{k-1} + \cdots + a_0 = 0$.
   Kořeny: λ_1 (násobnost k_1), ..., λ_r (násobnost k_r).
2. Kořen λ_1 násobnosti $k_1 \geq 1$ přidá k_1 různých posloupností do fundamentálního systému:
   $$\lambda_1^n, \quad n \cdot \lambda_1^n, \quad n^2 \cdot \lambda_1^n, \quad \ldots, \quad n^{k_1-1} \cdot \lambda_1^n$$
   (analogicky přispějí kořeny λ_2, ..., λ_r).
3. Fundamentální systém má celkově k různých posloupností, protože $k_1 + k_2 + \cdots + k_r = k$.
4. Kompletní řešení homogenní rovnice je lineární kombinace fundamentálního systému.

Kompletní řešení homogenní rovnice

1. Vyřešíme charakteristickou rovnici $a_k\lambda^k + a_{k-1}\lambda^{k-1} + \cdots + a_0 = 0$.
   Kořeny: λ_1 (násobnost k_1), ..., λ_r (násobnost k_r).
2. Kořen λ_1 násobnosti $k_1 \geq 1$ přidá k_1 různých posloupností do fundamentálního systému:

   $$\lambda_1^n, \quad n \cdot \lambda_1^n, \quad n^2 \cdot \lambda_1^n, \quad \ldots, \quad n^{k_1-1} \cdot \lambda_1^n$$

   (analogicky přispějí kořeny $\lambda_2, \ldots, \lambda_r$).
3. Fundamentální systém má celkově k různých posloupností, protože $k_1 + k_2 + \cdots + k_r = k$.
4. Kompletní řešení homogenní rovnice je lineární kombinace fundamentálního systému.

Kompletní řešení homogenní rovnice

1. Vyřešíme charakteristickou rovnici
   $a_k\lambda^k + a_{k-1}\lambda^{k-1} + \cdots + a_0 = 0$.
   Kořeny: λ_1 (násobnost k_1), ..., λ_r (násobnost k_r).
2. Kořen λ_1 násobnosti $k_1 \geq 1$ přidá k_1 různých posloupností do fundamentálního systému:

$$\lambda_1^n, \quad n \cdot \lambda_1^n, \quad n^2 \cdot \lambda_1^n, \quad \ldots, \quad n^{k_1-1} \cdot \lambda_1^n$$

   (analogicky přispějí kořeny λ_2, ..., λ_r).
3. Fundamentální systém má celkově k různých posloupností, protože $k_1 + k_2 + \cdots + k_r = k$.
4. Kompletní řešení homogenní rovnice je lineární kombinace fundamentálního systému.

Kompletní řešení homogenní rovnice

1. Vyřešíme charakteristickou rovnici
   $a_k\lambda^k + a_{k-1}\lambda^{k-1} + \cdots + a_0 = 0$.
   Kořeny: λ_1 (násobnost k_1), ..., λ_r (násobnost k_r).
2. Kořen λ_1 násobnosti $k_1 \geq 1$ přidá k_1 různých posloupností do fundamentálního systému:
   $$\lambda_1^n, \quad n \cdot \lambda_1^n, \quad n^2 \cdot \lambda_1^n, \quad \ldots, \quad n^{k_1-1} \cdot \lambda_1^n$$
   (analogicky přispějí kořeny λ_2, ..., λ_r).
3. Fundamentální systém má celkově k různých posloupností, protože $k_1 + k_2 + \cdots + k_r = k$.
4. Kompletní řešení homogenní rovnice je lineární kombinace fundamentálního systému.

Kompletní řešení homogenní rovnice

1. Vyřešíme charakteristickou rovnici $a_k\lambda^k + a_{k-1}\lambda^{k-1} + \cdots + a_0 = 0$.
   Kořeny: λ_1 (násobnost k_1), ..., λ_r (násobnost k_r).
2. Kořen λ_1 násobnosti $k_1 \geq 1$ přidá k_1 různých posloupností do fundamentálního systému:
   $$\lambda_1^n, \quad n \cdot \lambda_1^n, \quad n^2 \cdot \lambda_1^n, \quad \ldots, \quad n^{k_1-1} \cdot \lambda_1^n$$
   (analogicky přispějí kořeny λ_2, ..., λ_r).
3. Fundamentální systém má celkově k různých posloupností, protože $k_1 + k_2 + \cdots + k_r = k$.
4. Kompletní řešení homogenní rovnice je lineární kombinace fundamentálního systému.

Odhad partikulárního řešení pro pravou stranu $A^n \cdot P(n)$, kde A je číslo a $P(n)$ je polynom

1. d je násobnost A jako kořene charakteristické rovnice. (Násobnost 0 znamená: A není kořen).
2. Odhad partikulárního řešení: $n^d \cdot A^n \cdot p(n)$, kde $p(n)$ je polynom stejného stupně jako $P(n)$.
3. Koeficienty polynomu $p(n)$ získáme z požadavku, že $n^d \cdot A^n \cdot p(n)$ má řešit danou nehomogenní rovnici.

Pro složitější pravou stranu lze použít princip superposice.

Odhad partikulárního řešení pro pravou stranu $A^n \cdot P(n)$, kde A je číslo a $P(n)$ je polynom

1. d je násobnost A jako kořene charakteristické rovnice. (Násobnost 0 znamená: A není kořen).
2. Odhad partikulárního řešení: $n^d \cdot A^n \cdot p(n)$, kde $p(n)$ je polynom stejného stupně jako $P(n)$.
3. Koeficienty polynomu $p(n)$ získáme z požadavku, že $n^d \cdot A^n \cdot p(n)$ má řešit danou nehomogenní rovnici.

Pro složitější pravou stranu lze použít princip superposice.

Odhad partikulárního řešení pro pravou stranu $A^n \cdot P(n)$, kde A je číslo a $P(n)$ je polynom

1. d je násobnost A jako kořene charakteristické rovnice. (Násobnost 0 znamená: A není kořen).
2. Odhad partikulárního řešení: $n^d \cdot A^n \cdot p(n)$, kde $p(n)$ je polynom stejného stupně jako $P(n)$.
3. Koeficienty polynomu $p(n)$ získáme z požadavku, že $n^d \cdot A^n \cdot p(n)$ má řešit danou nehomogenní rovnici.

Pro složitější pravou stranu lze použít princip superposice.

Odhad partikulárního řešení pro pravou stranu $A^n \cdot P(n)$, kde A je číslo a $P(n)$ je polynom

1. d je násobnost A jako kořene charakteristické rovnice. (Násobnost 0 znamená: A není kořen).
2. Odhad partikulárního řešení: $n^d \cdot A^n \cdot p(n)$, kde $p(n)$ je polynom stejného stupně jako $P(n)$.
3. Koeficienty polynomu $p(n)$ získáme z požadavku, že $n^d \cdot A^n \cdot p(n)$ má řešit danou nehomogenní rovnici.

Pro složitější pravou stranu lze použít princip superposice.

Odhad partikulárního řešení pro pravou stranu $A^n \cdot P(n)$, kde A je číslo a $P(n)$ je polynom

1. d je násobnost A jako kořene charakteristické rovnice. (Násobnost 0 znamená: A není kořen).
2. Odhad partikulárního řešení: $n^d \cdot A^n \cdot p(n)$, kde $p(n)$ je polynom stejného stupně jako $P(n)$.
3. Koeficienty polynomu $p(n)$ získáme z požadavku, že $n^d \cdot A^n \cdot p(n)$ má řešit danou nehomogenní rovnici.

Pro složitější pravou stranu lze použít princip superposice.

Kompletní řešení nehomogenní rovnice

1. Sečteme kompletní řešení homogenní rovnice a partikulární řešení.
2. Jsou-li zadány počáteční podmínky: nakonec určíme koeficienty lineární kombinace fundamentálního systému.

Shrnuto:

1. Silná analogie s lineárními diferenciálními rovnicemi.
2. Diferenční a sumační počet, viz např.

J. Kaucký, *Kombinatorické identity*, Veda, Bratislava, 1975

W. G. Kelley a A. C. Peterson, *Difference Equations: An Introduction with Applications*, Academic Press Inc., New York, 1991

Kompletní řešení nehomogenní rovnice

1. Sečteme kompletní řešení homogenní rovnice a partikulární řešení.
2. Jsou-li zadány počáteční podmínky: nakonec určíme koeficienty lineární kombinace fundamentálního systému.

Shrnuto:

1. Silná analogie s lineárními diferenciálními rovnicemi.
2. Diferenční a sumační počet, viz např.

J. Kaucký, *Kombinatorické identity*, Veda, Bratislava, 1975

W. G. Kelley a A. C. Peterson, *Difference Equations: An Introduction with Applications*, Academic Press Inc., New York, 1991

Kompletní řešení nehomogenní rovnice

1. Sečteme kompletní řešení homogenní rovnice a partikulární řešení.
2. Jsou-li zadány počáteční podmínky: nakonec určíme koeficienty lineární kombinace fundamentálního systému.

Shrnuto:

1. Silná analogie s lineárními diferenciálními rovnicemi.
2. Diferenční a sumační počet, viz např.

J. Kaucký, *Kombinatorické identity*, Veda, Bratislava, 1975

W. G. Kelley a A. C. Peterson, *Difference Equations: An Introduction with Applications*, Academic Press Inc., New York, 1991

Kompletní řešení nehomogenní rovnice

1. Sečteme kompletní řešení homogenní rovnice a partikulární řešení.
2. Jsou-li zadány počáteční podmínky: nakonec určíme koeficienty lineární kombinace fundamentálního systému.

Shrnuto:

1. Silná analogie s lineárními diferenciálními rovnicemi.
2. Diferenční a sumační počet, viz např.

J. Kaucký, *Kombinatorické identity*, Veda, Bratislava, 1975

W. G. Kelley a A. C. Peterson, *Difference Equations: An Introduction with Applications*, Academic Press Inc., New York, 1991

Definice (O-notace)

Ať $f, g : \mathbb{N} \to \mathbb{R}$ jsou funkce. Řekneme, že $f \in O(g)$ (někdy i $f(n) \in O(g(n))$) (čteme: f je třídy velké O g),[a] když existují C a n_0 tak, že platí

$$|f(n)| \leq C \cdot |g(n)|, \quad \text{pro všechna } n \geq n_0$$

[a] Zavedl Paul Bachmann v roce 1892.

Příklady

1. $f \in O(f)$ platí vždy.
2. $6n^3 - 127n^2 + \pi n \in O(n^3)$.
3. $3^n \notin O(n^p)$, pro každé $p \in \mathbb{N}$.
4. $n! \in O(n^n)$.

Věta

Ať $f, g : \mathbb{N} \to \mathbb{R}$ jsou funkce a ať limita

$$\lim_{n \to +\infty} \frac{|f(n)|}{|g(n)|}$$

existuje a je konečná. Pak $f \in O(g)$.

Pozor!

Obrácení této věty neplatí.

Pro $f(n) = \sin n$, $g(n) = 1$ platí $f \in O(g)$, ale $\lim_{n \to +\infty} \frac{|f(n)|}{|g(n)|}$ neexistuje.

Hierarchie (asymptotické) složitosti

1. Polynomiální: například $O(n^4)$.
   Slogan: polynomiálně složité algoritmy jsou "rychlé".
   Příklad: Bubblesort je $O(n^2)$.
2. Exponenciální: například $O(2^n)$.
   Slogan: exponenciálně složité algoritmy jsou "pomalé".
   Příklad: test prvočíselnosti postupným dělením (Eratosthenovo síto).
3. A řada dalších tříd složitosti...

Viz například

T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, MIT Press, 2001

Strategie Divide and Conquer

Problém velikosti n je rozdělen na a podproblémů velikosti $\frac{n}{b}$ a při dělení je "spotřebován čas" $f(n)$. Celkový čas $T(n)$ je pak dán vztahem

$$T(n) = a \cdot T(\frac{n}{b}) + f(n), \quad n \geq 1.$$

Příklad (Merge-Sort)

Vstup: pole čísel $\vec{x} = x[1], \ldots, x[n]$.
Výstup: setříděné pole čísel $\vec{x}$.

Složitost: $T(n) = \begin{cases} c & \text{pro } n = 1 \\ 2T(n/2) + cn & \text{pro } n > 1. \end{cases}$

Rovnice Divide and Conquer

$$T(n) = a \cdot T(\frac{n}{b}) + f(n), \quad n \geq 1,$$

kde $a \geq 0$, $b > 0$ jsou přirozená čísla.
Pokud $n = b^k$, pak platí

$$\begin{aligned}
T(n) &= a \cdot T(\frac{n}{b}) + f(n) \\
&= a^2 \cdot T(\frac{n}{b^2}) + a \cdot f(\frac{n}{b}) + f(n) \\
&= a^3 \cdot T(\frac{n}{b^3}) + a^2 \cdot f(\frac{n}{b^2}) + a \cdot f(\frac{n}{b}) + f(n) \\
&\vdots \\
&= a^k \cdot T(\frac{n}{b^k}) + \sum_{j=0}^{k-1} a^j \cdot f(\frac{n}{b^j})
\end{aligned}$$

Divide and Conquer, pokrač.

Pokud $n = b^k$, platí

$$T(n) \;=\; a^k \cdot T(1) + \sum_{j=0}^{k-1} a^j \cdot f(\frac{n}{b^j})$$

Věta

Ať $T : \mathbb{N} \to \mathbb{R}$ je rostoucí funkce, která pro všechna n dělitelná přirozeným číslem $b \geq 2$ splňuje rekurentní rovnici

$$T(n) = \begin{cases} c & \textit{pro } n = 1 \\ a \cdot T(n/b) + cn & \textit{pro } n > 1. \end{cases},$$

kde $a \geq 1$, $c > 0$ jsou reálná čísla. Pak platí:

$$T(n) \in O(n^{\log_b a}),\ \textit{když } a > b, \quad T(n) \in O(n \log n),\ \textit{když } a = b.$$