Hluboké zásobníkové automaty konečného indexu

Vendula Poncová

xponco00@stud.fit.vutbr.cz

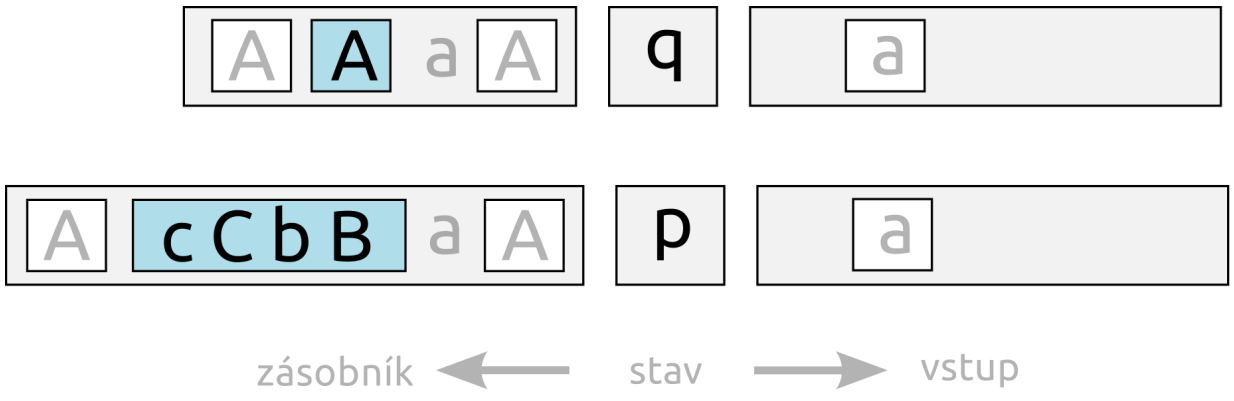
Hluboký zásobníkový automat

Hluboký zásobníkový automat je sedmice $M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

Q je konečná množina stavů,
Σ vstupní abeceda,
Γ zásobníková abeceda, $\Sigma \subseteq \Gamma$, $Q \cap \Gamma = \emptyset$,
R je konečná množina pravidel,
$s \in Q$ je počáteční stav,
$S \in \Gamma$ počáteční zásobníkový symbol a
$F \subseteq Q$ je množina koncových stavů.

Pravidla množiny R jsou tvaru:
$mqA \to pv$, kde $m \in \{1, 2, 3, \ldots, n\}$, q, $p \in Q$, $A \in (\Gamma - \Sigma)$, $v \in \Gamma^+$.

Hluboký zásobníkový automat

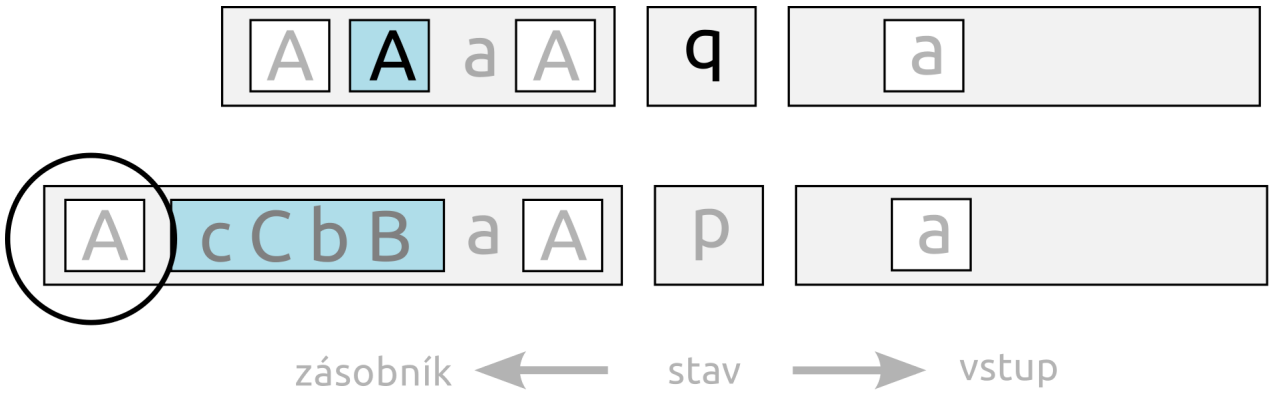


Příklad aplikace pravidla $2qA \to pBbCc$.

Konečný index

Hluboký zásobníkový automat konečného indexu je osmice $M = (Q, \Sigma, \Gamma, R, s, S, F, n)$, kde $n \in \{1, 2, 3, \dots\}$ je maximální počet nevstupních symbolů na zásobníku.

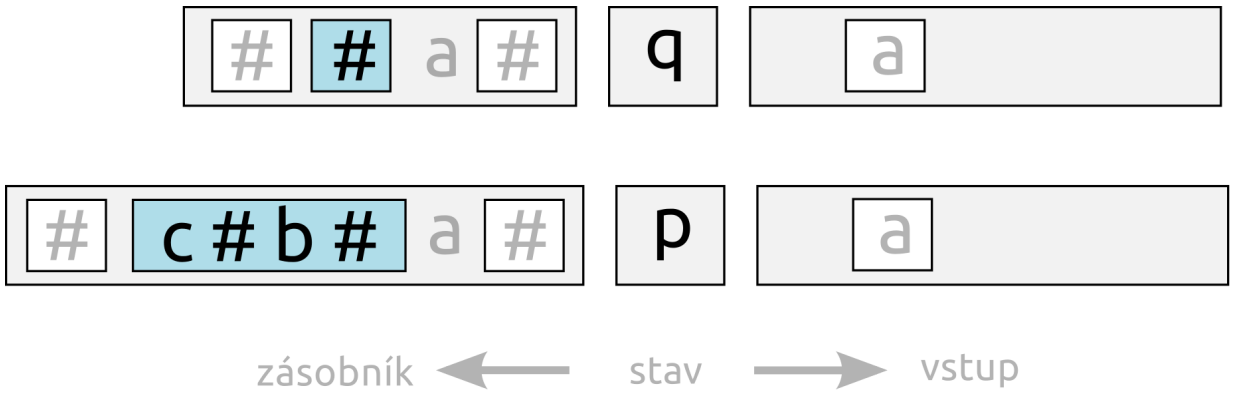
Konečný index



V hlubokém zásobníkovém automatu konečného indexu 3 nelze pravidlo $2qA \to pBbCc$ *v aktuální konfiguraci aplikovat.*

Omezení počtu nevstupních symbolů

Hluboký zásobníkový automat konečného indexu s jedním nevstupním symbolem $M_{\#}$.



Aplikace pravidla $2q\# \to p\#b\#c$ v aktuální konfiguraci.

Ekvivalence s hlubokými PDA

Je ekvivalentní s hlubokým zásobníkovým automatem konečného indexu?

1. Každý hluboký zásobníkový automat konečného indexu s jedním nevstupním symbolem splňuje definici pro obecný hluboký zásobníkový automat konečného indexu.

2. Existuje pro každý hluboký zásobníkový automat konečného indexu ekvivalentní s jedním nevstupním symbolem?

Převod na zredukovaný automat

Algoritmus:

Vstup: $M = (Q, \Sigma, \Gamma, R, s, S, F, n)$
Výstup: $M_{\#} = (Q_{\#}, \Sigma, \{\#\} \cup \Sigma, R_{\#}, < s, \# >, \#, F_{\#}, n)$

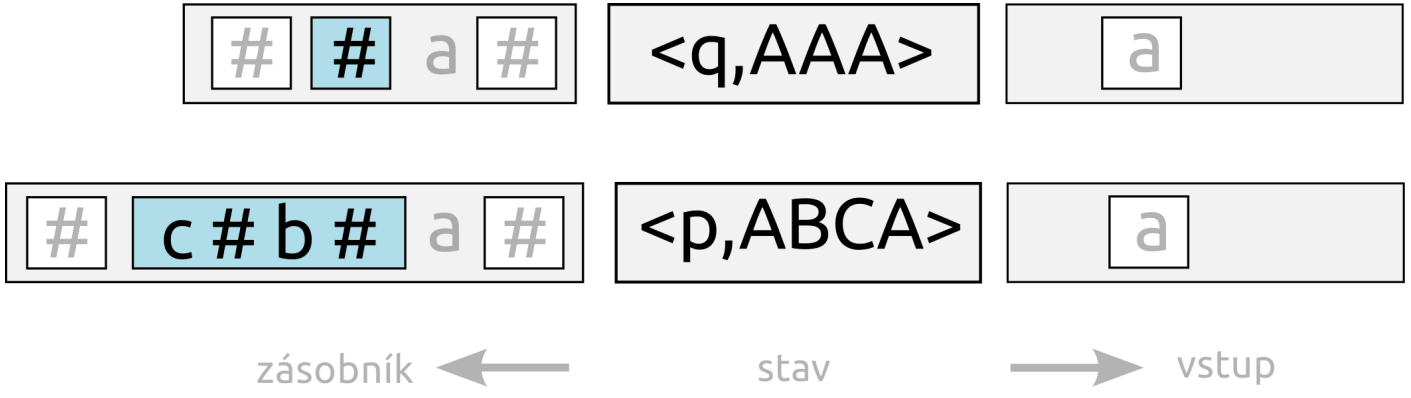
Pro každé pravidlo $mqA \to pb_0B_1b_1B_2b_2 \dots b_{j-1}B_jb_j \in R$, kde $j \in \{0, 1, 2, \dots, n\}$, $b_0, b_1, \dots, b_j \in \Sigma^*$ a $B_1, B_2, \dots, B_j \in (\Gamma - \Sigma)$, a každé $(u, z) \in (\Gamma - \Sigma)^* \times (\Gamma - \Sigma)^*$, kde $|u| = m - 1$, $|z| \leq n - m$:

přidej do $R_{\#}$ pravidlo

$m < q, uAz > \# \to < p, uB_1B_2 \dots B_{j-1}B_jz > b_0\#b_1\#b_2 \dots b_{j-1}\#b_j$,

přidej do $Q_{\#}$ stavy $< q, uAz >$, $< p, uB_1B_2 \dots B_{j-1}B_jz >$,
pokud $p \in F$, přidej do $F_{\#}$ stav $< p, uB_1B_2 \dots B_{j-1}B_jz >$,
pokud $q \in F$, přidej do $F_{\#}$ stav $< q, uAz >$.

Převod na zredukovaný automat



Převod pravidla $2qA \to pBbCc$ *pro obsah zásobníku* $AaAA$.

Programová gramatika

Programová gramatika je čtveřice $G = (V, T, P, S)$, kde

$V = T \cup N$ je úplná abeceda,
T je abeceda terminálů,
N je abeceda neterminálů,
P je konečná množina pravidel,
$S \in N$ je počáteční symbol.

Pravidla množiny P jsou tvaru $r\colon A \to v, g(r)$, kde

r je označení pravidla,
$A \to v$ je pravidlo bezkontextové gramatiky a
$g(r)$ je množina značení těch pravidel, která mohou být provedena v dalším derivačním kroku po aplikaci pravidla r.

Konečný index

Programová gramatika konečného indexu n je programová gramatika $G = (V, T, P, S)$, pro jejíž každou větnou formu $w \in L(G)$ existuje taková posloupnost derivačních kroků, která v žádném kroku neobsahuje více než n neterminálů.

Ekvivalence s PG

Je hluboký zásobníkový automat s jedním nevstupním symbolem ekvivalentní s programovými gramatikami konečného indexu?

Převod gramatiky na automat

Algoritmus:

Vstup: $G = (T \cup N, T, P, S)$ konečného indexu n
Výstup: $M = (Q, T, T \cup \{\#\}, R, < \sigma >, \#, F, n)$

Pro každé $p : S \to v, g(p) \in P$:
přidej do $R < \sigma >_1 \# \to < p, S > \#$ a do $Q < p, S >$.

Pro každé $p : A \to b_0 B_1 b_1 B_2 b_2 \dots b_{j-1} B_j b_j, g(p) \in P$, kde
$j \in \{0, 1, 2, \dots\}$, $b_0, b_1, \dots, b_j \in T^*$ a $B_1, B_2, \dots, B_j \in N$ a pro každé
$(k, u, z) \in \{1, 2, 3, \dots, n - j + 1\} \times N^* \times N^*$, kde $|u| = k - 1$, $|z| \leq n - k$:

Pokud $g(p) \neq \emptyset$, pak pro každé $q \in g(p)$:
přidej do Q stavy $< p, uAz >$, $< q, uB_1 B_2 \dots B_{j-1} B_j z >$ a do R
$< p, uAz >_k \# \to < q, uB_1 B_2 \dots B_{j-1} B_j z > b_0 \# b_1 \# b_2 \dots b_{j-1} \# b_j$.

Jinak přidej do $Q < p, uAz >$, $< \varepsilon, uB_1 B_2 \dots B_{j-1} B_j z >$ a do R
$< p, uAz >_k \# \to < \varepsilon, uB_1 B_2 \dots B_{j-1} B_j z > b_0 \# b_1 \# b_2 \dots b_{j-1} \# b_j$.

Pro každé $q \in (Q \cup \{\varepsilon\})$ přidej do $F < q, \varepsilon >$.

Převod automatu na gramatiku

Programová gramatika simuluje každý krok zásobníkového automatu sekvencí několika derivací.

Neterminály jsou ve tvaru:
<aktuální pozice, aktuální pozice výskytu #, celkový počet # v konfiguraci>.

Vlastní simulace probíhá následovně:

1. **Přečíslení.** Aktualizace pozice a celkového počtu nevstupních symbolů u všech neterminálů.
2. **Expanze.** Expanduje neterminál na příslušné pozici.
3. **Finalizace.** Přepis pomocného tvaru nonterminálů.

Postup jsem převzala z článku Generation of Languages by Rewriting Systems that Resemble Automata od Křivky.

Shrnutí

Vlastnosti hlubokého zásobníkového automatu konečného indexu s jedním nevstupním symbolem:

- Redukce nevstupních symbolů na jeden.
- Zjednodušení zápisu pravidel.
- Ekvivalence s hlubokým zásobníkovým automatem konečného indexu.
- Ekvivalence s programovými gramatikami.
- Nekonečná hierarchie jazyků.