

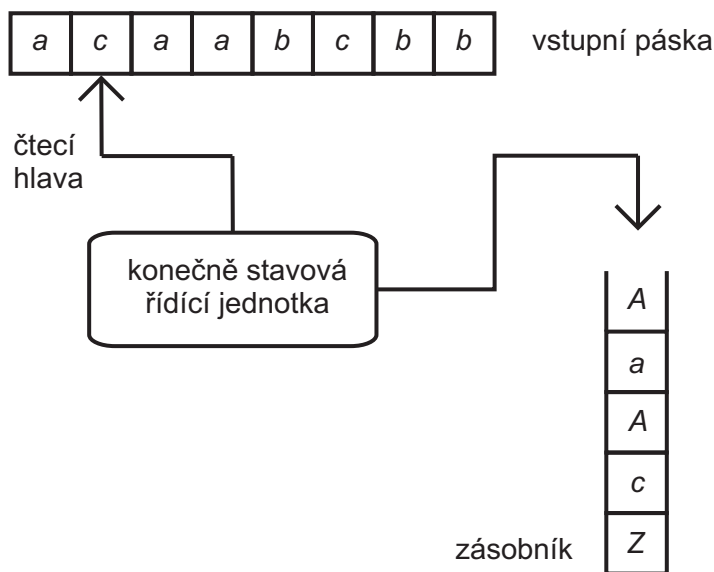
Zásobníkové automaty – úvod

Již víme, že výpočetním strojem pro regulární jazyky jsou konečné automaty. V kapitolách 8, 9 jsme podrobně rozebrali bezkontextové gramatiky, které generují bezkontextové jazyky. Dosud však nevíme, jaký výpočetní stroj této třídě jazyků přísluší. V následujících dvou kapitolách tento „rest“ doženeme a povíme si o zásobníkových automatech, zkráceně PDA (z anglického jazyka Push-Down Automata).

Princip fungování PDA

Zásobníkový automat si představte jako vylepšený konečný automat. Opět se zde setkáváme se **vstupní páskou**, na kterou jsou před začátkem výpočtu zkopírovány všechny symboly slova, u kterého chceme zjistit, zda jej automat akceptuje či ne. I **konečně stavová řídicí jednotka** je zachována, pamatuje si aktuální stav a její čtecí hlava ukazuje na právě čtený symbol na vstupní pásce.

Navíc však PDA obsahuje tzv. **zásobník**, neboli pomocnou paměť, do které můžeme ukládat symboly. Konečně stavová řídicí jednotka se tedy rozhoduje o dalším kroku nejen na základě symbolu na vstupní pásce a aktuálního stavu, ale také podle toho, co má na zásobníku. Prohlédněte si obrázek 1 znázorňující strukturu zásobníkového automatu.



1. Na vstupní pásku se zapíše slovo w u kterého zjišťujeme, zda jej PDA akceptuje. Slovo w se na vstupní pásku zapíše zleva doprava. Na začátku výpočtu se čtecí hlava posune na první symbol slova w .
2. Zásobník funguje na principu *LIFO* (Last in, first out) - poslední symbol, který přišel do zásobníku, je odejde jako první.
3. Konečně stavová řídicí jednotka „vidí“ pouze na vrchol zásobníku, tj. na symbol, který byl do zásobníku naposledy zapsán.
4. Čtení symbolu z vrcholu zásobníku je destruktivní, tj. pokud přečteme symbol, tak jej zároveň mažeme.

Definice – nedeterministický zásobníkový automat

Nedeterministický zásobníkový automat (PDA) je sedmice $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ kde:

- Q : konečná množina stavů
- Σ : konečná množina vstupních symbolů
- Γ : konečná množina zásobníkových symbolů,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma^*)$ přechodová funkce
- $q_0 \in Q$: počáteční stav
- $Z_0 \in \Gamma$: počáteční symbol zásobníku
- $F \subseteq Q$: podmnožina konečných stavů

Poznámka.

1. Všimněte si, že není dán vztah množin Σ, Γ . To znamená, vstupní symbol může být i zásobníkovým symbolem, což se v praxi často stává.
2. Definičním oborem přechodové funkce δ je trojice (stav, vstupní symbol, zásobníkový symbol), což odpovídá myšlence v úvodní části tohoto textu: konečně stavová řídicí jednotka se rozhoduje na základě aktuálního stavu, symbolu na vstupní pásce a symbolu na vrcholu zásobníku.
3. Oborem hodnot funkce δ je potenční množina $\mathcal{P}(Q \times \Gamma^*)$, tj. libovolný, ale konečný počet uspořádaných dvojic (stav, řetězec symbolů). Konečně stavová řídicí jednotka tedy v každém kroku mění svůj stav a na vrchol zásobníku zapíše libovolnou posloupnost zásobníkových symbolů.

Nyní si uvedeme formální definici toho, jak zásobníkový automat pracuje, tj. zavedeme pojmy konfigurace, vnitřní konfigurace a krok výpočtu.

Definice – konfigurace, krok výpočtu

Buď $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ nedeterministický zásobníkový automat.

1. **Konfigurací** rozumíme trojici (q, w, α) , kde $q \in Q$ je aktuální stav, w je dosud nepřečtená část vstupního slova a $\alpha \in \Gamma^*$ jsou všechny symboly zásobníku čtené zleva doprava.
2. **Vnitřní konfigurací** rozumíme libovolnou dvojici (q, α) , kde význam obou složek je stejný jako u konfigurace.
3. **Krok výpočtu** – tj. binární relaci \mapsto na množině konfigurací definujeme tak, že

$$(q, aw, A\alpha) \mapsto (p, w, \beta\alpha),$$

právě když $\delta(q, a, A)$ obsahuje dvojici (p, β) .

Popíšeme si jednotlivé symboly: $q \in \Sigma \cup \{\varepsilon\}$, $p, q \in Q$, $\alpha, \beta \in \Gamma^*$, $A \in \Gamma$.

Poznámka.

Je zřejmé, že u binární relace \mapsto můžeme specifikovat její druh v závislosti na počtu kroků, které při výpočtu udělá. Zápisem

- \mapsto^k ($k \in \mathbb{N}_0$) rozumíme výpočet přesně v k krocích.

- \mapsto^* rozumíme výpočet v libovolně mnoha krocích (reflexivní a tranzitivní uzávěr \mapsto)
- \mapsto^+ rozumíme výpočet v libovolném nenulovém počtu kroků (tranzitivní uzávěr \mapsto).

Nyní nám ještě zbývá uvést, za jakých podmínek PDA akceptuje vstupní slovo. Ještě než si definici uvedeme, prozradíme, že přijímání slov je možné dvěma způsoby: koncovým stavem či prázdným zásobníkem.

Definice – způsob akceptování PDA

Nedeterministický zásobníkový automat $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ akceptuje

- **koncovým stavem**, právě když pro libovolné slovo $w \in L(M)$ platí

$$(q_0, w, Z_0) \mapsto^* (q_{accept}, \varepsilon, \alpha),$$

kde $q_{accept} \in F$, $\alpha \in \Gamma^*$.

- **prázdným zásobníkem**, právě když pro libovolné slovo $w \in L(M)$ platí

$$(q_0, w, Z_0) \mapsto^* (q, \varepsilon, \varepsilon),$$

kde $q \in Q$.

Poznámka.

Způsob akceptování nejlépe rozpoznáte tak, že v případě PDA přijímajícího koncovým stavem je množina F (tj. poslední položka sedmice) neprázdná. Naopak u PDA přijímajícího prázdným zásobníkem je na posledním místě sedmice uvedena prázdná množina \emptyset .

Příklad 1.

Buď $M = (\{p, q, q_{accept}\}, \{a, b\}, \{B, Z\}, \delta, p, Z, q_{accept})$ zásobníkový automat s funkcí δ :

$$\delta(p, a, Z) = (p, BZ)$$

$$\delta(p, a, B) = (p, BB)$$

$$\delta(p, b, B) = (q, \varepsilon)$$

$$\delta(q, b, B) = (q, \varepsilon)$$

$$\delta(q, \varepsilon, Z) = (q_{accept}, Z)$$

Proveďte výpočet automatu M pro slova $aabb, aab, baa, ab$ a rozhodněte, která z nich automat M akceptuje a která ne.

Řešení.

1. $(p, aabb, Z) \mapsto (p, abb, BZ) \mapsto (p, bb, BBZ) \mapsto (q, b, BZ) \mapsto (q, \varepsilon, Z) \mapsto (q_{accept}, \varepsilon, Z)$. Slovo $aabb \in L(M)$, protože jsme skončili v koncovém stavu a přečetli všechny symboly ze vstupní pásky.
2. $(p, aab, Z) \mapsto (p, ab, BZ) \mapsto (p, b, BBZ) \mapsto (q, \varepsilon, BZ)$. Slovo $aab \notin L(M)$, protože jsme výpočet skončili v nekoncovém stavu q .
3. Pro počáteční konfiguraci (p, baa, Z) neexistuje přechod pomocí funkce δ , tedy $baa \notin L(M)$ protože jsme jednak nepřčetli celé slovo a také neskončili v koncovém stavu.
4. $(p, ab, Z) \mapsto (p, b, BZ) \mapsto (q, \varepsilon, Z) \mapsto (q_{accept}, \varepsilon, Z)$. Slovo $ab \in L(M)$ – PDA přečetl celé slovo a skončil výpočet ve stavu $q_{accept} \in F$.

Příklad 2.

Mějme PDA $M = (\{q_0\}, \{a, b\}, \{A, B, Z\}, \delta, q_0, Z, \emptyset)$ přijímající prázdným zásobníkem, kde přechodová funkce δ je definována následovně:

$$\begin{aligned}\delta(q_0, a, Z) &= (q_0, AZ) \\ \delta(q_0, b, Z) &= (q_0, BZ) \\ \delta(q_0, a, A) &= (q_0, AA) \\ \delta(q_0, b, B) &= (q_0, BB) \\ \delta(q_0, a, B) &= (q_0, \varepsilon) \\ \delta(q_0, b, A) &= (q_0, \varepsilon) \\ \delta(q_0, \varepsilon, Z) &= (q_0, \varepsilon)\end{aligned}$$

1. Proveďte výpočet na slovech $abba, abbab, aabb, a, b, abab, baa$.
2. Nalezněte jazyk $L = L(M)$

Řešení.

- $(q_0, abba, Z) \mapsto (q_0, bba, AZ) \mapsto (q_0, ba, Z) \mapsto (q_0, a, BZ) \mapsto (q_0, \varepsilon, Z) \mapsto (q_0, \varepsilon, \varepsilon)$. Slovo $abba \in L(M)$, jelikož PDA přečetl celé slovo ze vstupní pásky a má prázdný zásobník.
- $(q_0, abbab, Z) \mapsto (q_0, bbab, AZ) \mapsto (q_0, bab, Z) \mapsto (q_0, ab, BZ) \mapsto (q_0, b, Z) \mapsto (q_0, \varepsilon, BZ)$. Slovo $abbab$ bylo sice přečteno celé, zásobník však obsahuje řetězec BZ , tedy není prázdný. Z toho důvodu $abbab \notin L(M)$.
- $(q_0, aabb, Z) \mapsto (q_0, abb, AZ) \mapsto (q_0, bb, AAZ) \mapsto (q_0, b, AZ) \mapsto (q_0, \varepsilon, Z) \mapsto (q_0, \varepsilon, \varepsilon)$. Slovo $aabb \in L(M)$.
- $(q_0, a, Z) \mapsto (q_0, \varepsilon, AZ) \Rightarrow a \notin L(M)$.
- $(q_0, b, Z) \mapsto (q_0, \varepsilon, BZ) \Rightarrow b \notin L(M)$.
- $(q_0, abab, Z) \mapsto (q_0, bab, AZ) \mapsto (q_0, ab, Z) \mapsto (q_0, b, AZ) \mapsto (q_0, \varepsilon, Z) \mapsto (q_0, \varepsilon, \varepsilon) \Rightarrow abab \in L(M)$
- $(q_0, baa, Z) \mapsto (q_0, aa, BZ) \mapsto (q_0, a, Z) \mapsto (q_0, \varepsilon, AZ) \Rightarrow baa \notin L(M)$.

Pokud si vše dáme dohromady, zjistíme, že slova $abba, aabb, abab \in L(M)$, kdežto $abbab, a, b, baa \notin L(M)$. PDA M funguje tak, že

- načítá-li symbol a ze vstupní pásky, pak v případě, že na vrcholu zásobníku jsou Z nebo A , přidává na zásobník další symbol A . Pokud je však na vrcholu zásobníku B , tak jej maže.
- V případě symbolu b je to opačně: je-li na vrcholu zásobníku B nebo Z , přidává další B , pokud však na zásobníku „vidí“ A , maže jej.

Shrneme-li předchozí dva body, můžeme usoudit, že pomocí symbolů A si PDA pamatuje počet a , pomocí B si zaznamenává počet b . Jakmile přečte celé slovo a na zásobníku má Z , znamená to, že počet a byl stejný jako počet b . V tom případě vyprazdňuje zásobník a akceptuje dané slovo. V opačném případě nedělá nic, čili nepřijímá slovo (nemá zásobník prázdný). PDA M tedy přijímá jazyk

$$L(M) = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$$