Zásobníkové automaty

Zásobníkový automat

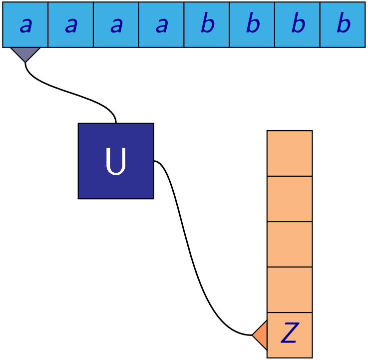
- Chtěli bychom rozpoznávat jazyk $L = \{a^i b^i \mid i \geq 1\}$
- Snažíme se navrhnout zařízení (podobné konečným automatům), které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.
- Při čtení a-ček si musíme pamatovat jejich počet, ať víme, kolik musí následovat b-ček

Zásobníkový automat

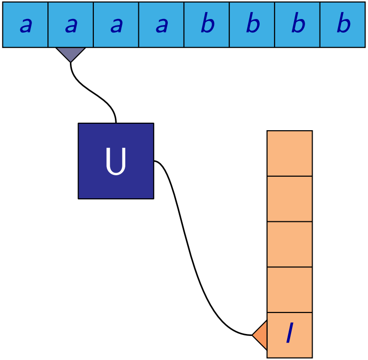
- Chtěli bychom rozpoznávat jazyk $L = \{a^i b^i \mid i \geq 1\}$
- Snažíme se navrhnout zařízení (podobné konečným automatům), které přečte slovo, a sdělí nám, zda toto slovo patří do jazyka L či ne.
- Při čtení a-ček si musíme pamatovat jejich počet, ať víme, kolik musí následovat b-ček
- Můžeme využít paměť typu zásobník
- Každé přečtené a si na zásobník zapíšeme, za každé přečtené b jeden symbol ze zásobníku odstraníme
- Pokud bude zásobník prázdný a podaří se přečíst celé slovo, tak patří do jazyka

Zásobníkový automat

- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$

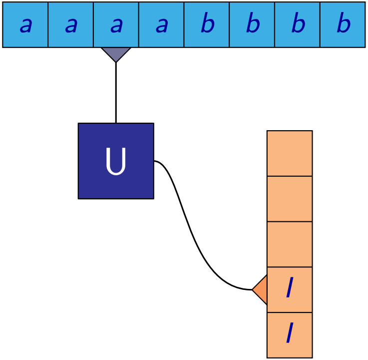


- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



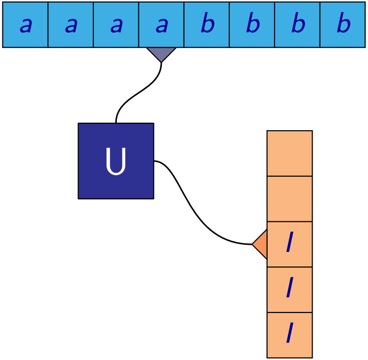
Zásobníkový automat

- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



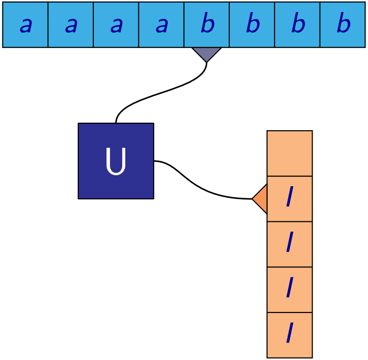
Zásobníkový automat

- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$

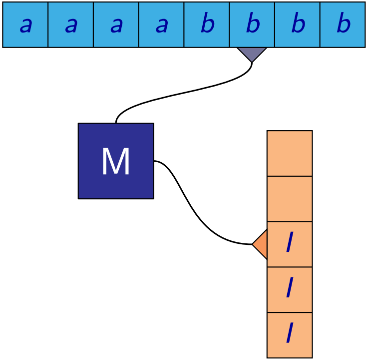


Zásobníkový automat

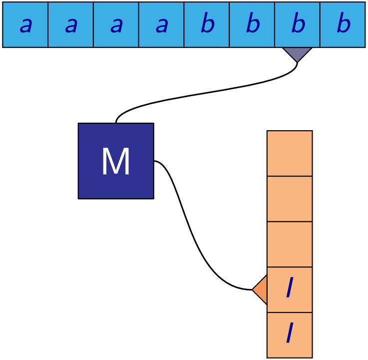
- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$

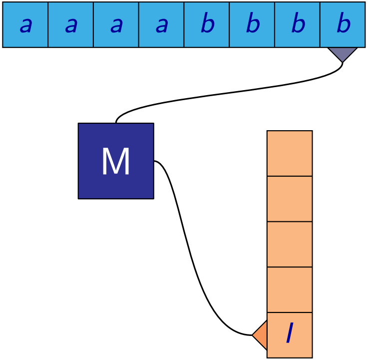


- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



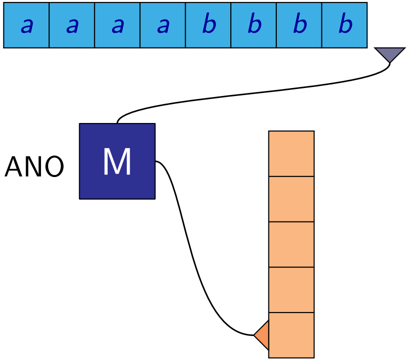
Zásobníkový automat

- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



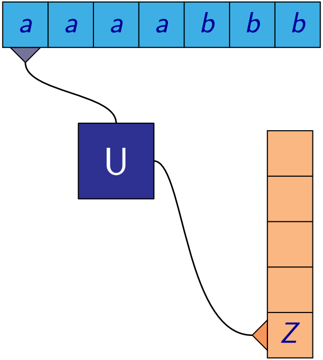
Zásobníkový automat

- Slovo *aaaabbbb* patří do jazyka $L = \{a^i b^i \mid i \geq 1\}$
- Automat přečetl celé slovo a skončil s prázdným zásobníkem, takže slovo přijal



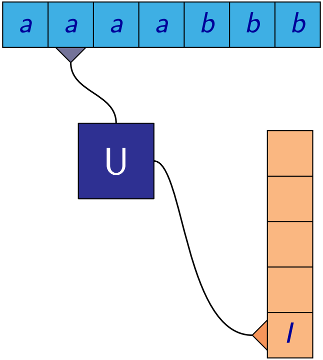
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



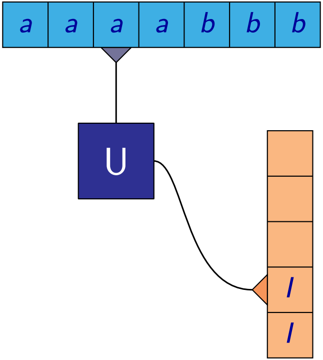
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



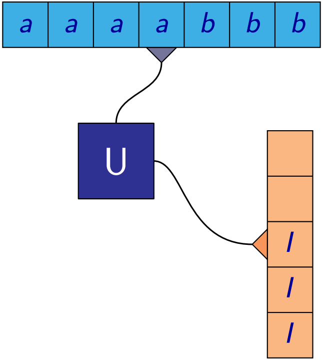
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



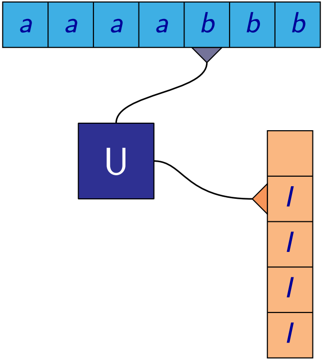
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



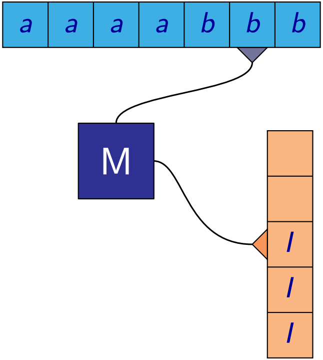
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



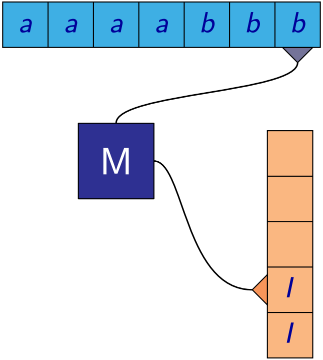
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



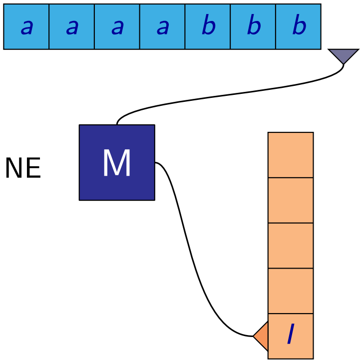
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



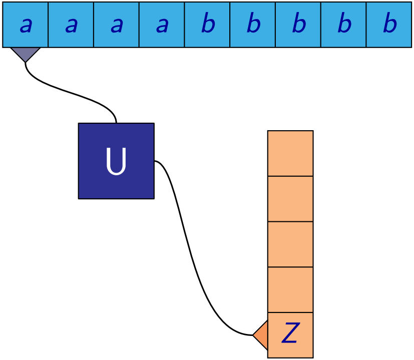
Zásobníkový automat

- Slovo *aaaabbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$
- Automat přečetl celé slovo, ale nevyprázdnil zásobník, takže slovo nepřijal



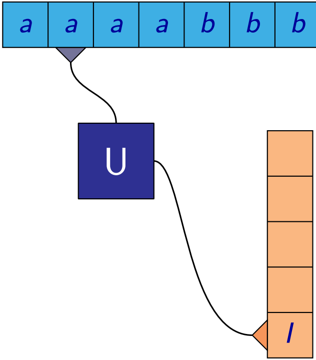
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



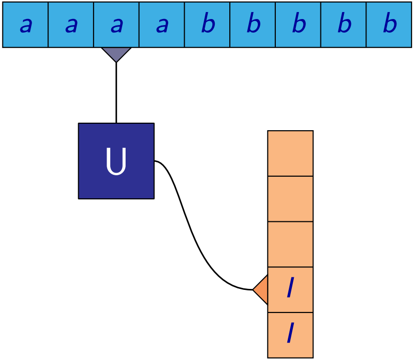
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



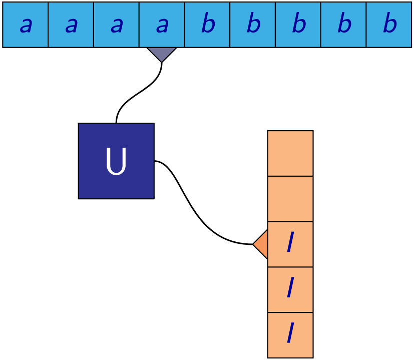
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



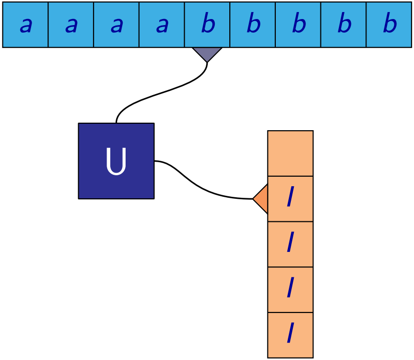
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



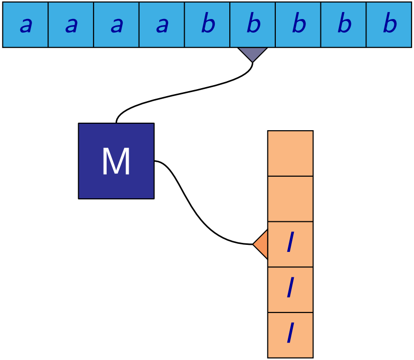
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



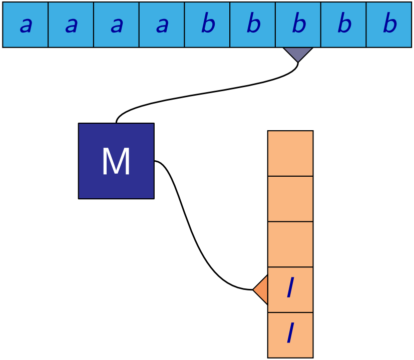
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



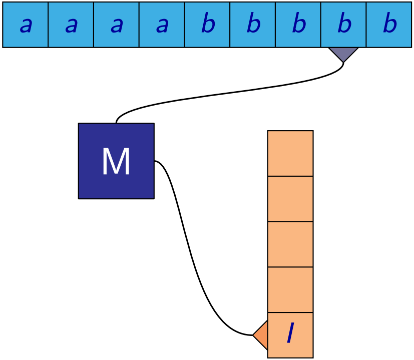
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



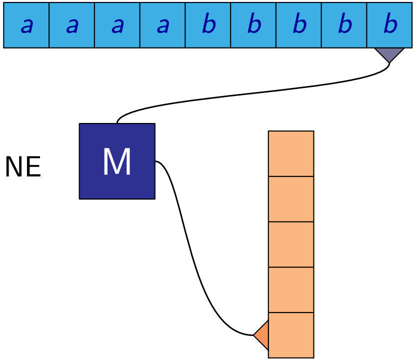
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



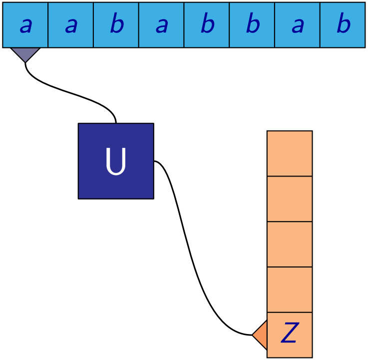
Zásobníkový automat

- Slovo *aaaabbbbb* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$
- Automat čte b, má smazat symbol na zásobníku a tam žádný není, takže slovo nepřijal



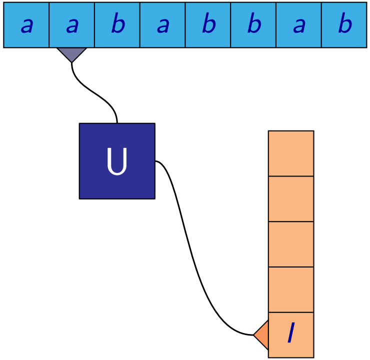
Zásobníkový automat

- Slovo *aababbab* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$

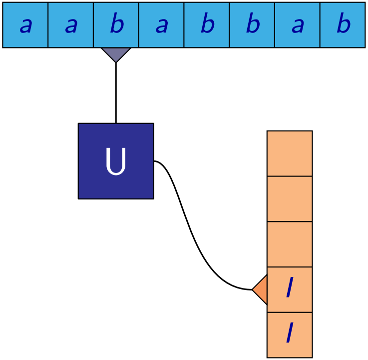


Zásobníkový automat

- Slovo *aababbab* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$

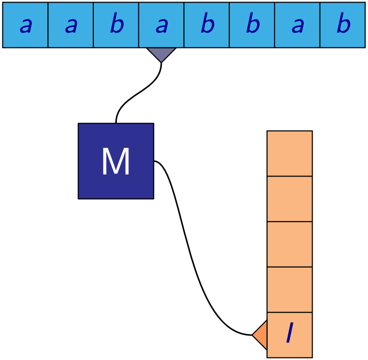


- Slovo *aababbab* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



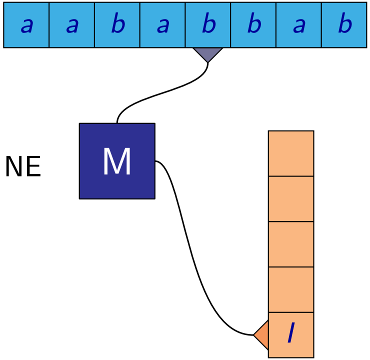
Zásobníkový automat

- Slovo *aababbab* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$



Zásobníkový automat

- Slovo *aababbab* nepatří do jazyka $L = \{a^i b^i \mid i \geq 1\}$
- Automat přečetl a, ale již byl ve stavu, kdy maže, takže slovo nepřijal



Zásobníkový automat

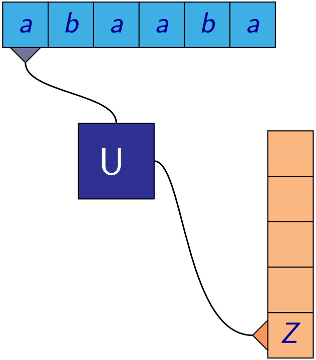
- Uvedený zásobníkový automat měl vždy jasně určeno pokračování - byl deterministický
- Je možné každý bezkontextový jazyk poznat deterministickým zásobníkovým automatem?

Zásobníkový automat

- Uvedený zásobníkový automat měl vždy jasně určeno pokračování - byl deterministický
- Je možné každý bezkontextový jazyk poznat deterministickým zásobníkovým automatem?
- Uvažujme jazyk $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- První půlku slova můžeme uložit na zásobník
- Při čtení druhé půlky mažeme symboly ze zásobníku, pokud jsou stejné jako na vstupu
- Pokud bude zásobník prázdný po přečtení celého slova, byla druhá půlka stejná jako první

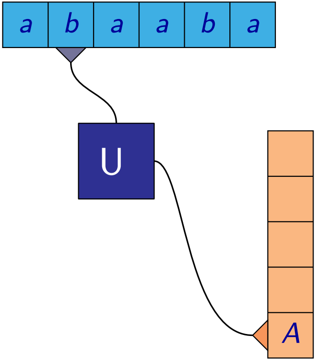
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$



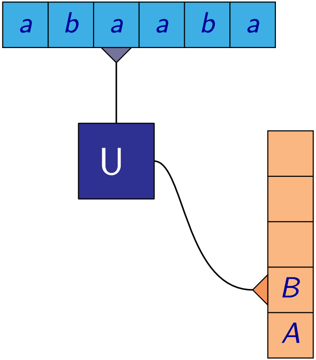
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$



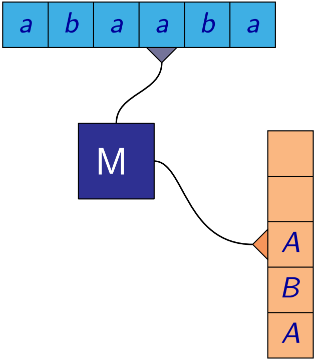
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, musí změnit stav na M a uložit A na zásobník



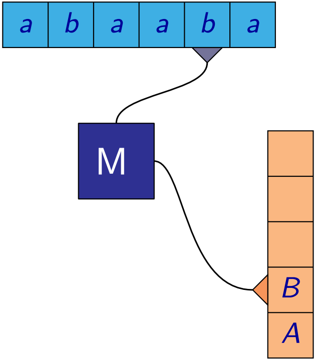
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, musí změnit stav na M a uložit A na zásobník



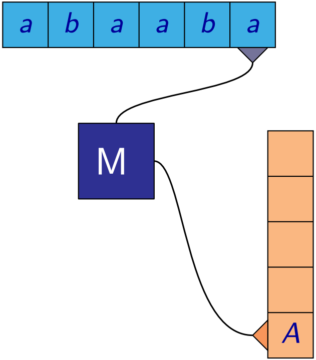
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, musí změnit stav na M a uložit A na zásobník



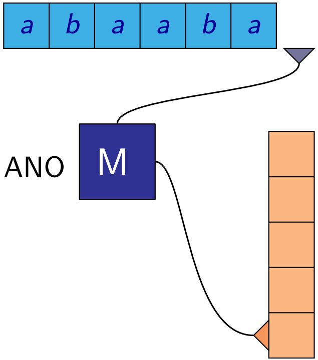
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, musí změnit stav na M a uložit A na zásobník



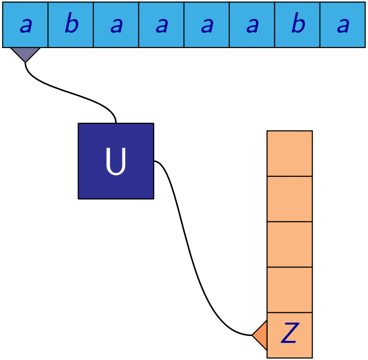
Zásobníkový automat

- Slovo *abaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, musí změnit stav na M a uložit A na zásobník



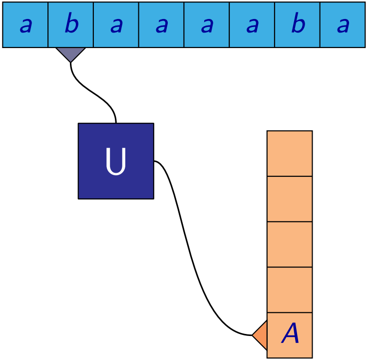
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$



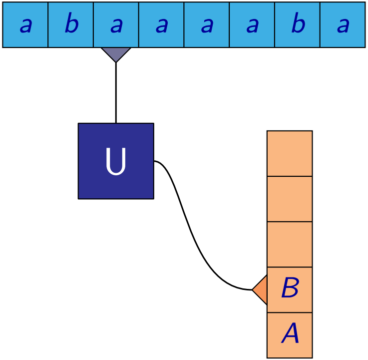
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$



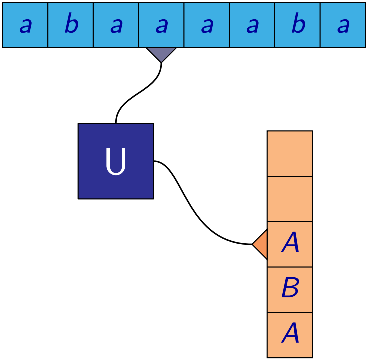
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



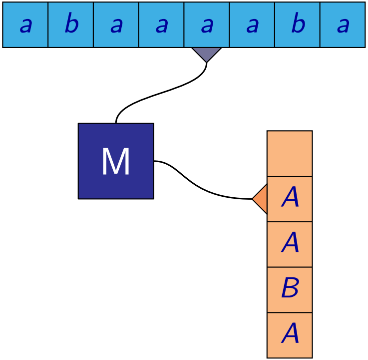
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



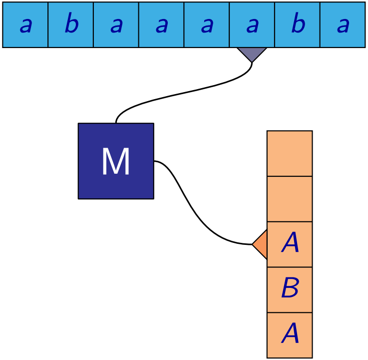
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



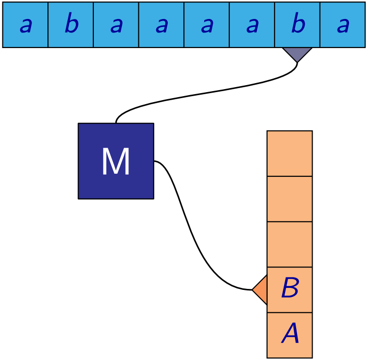
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



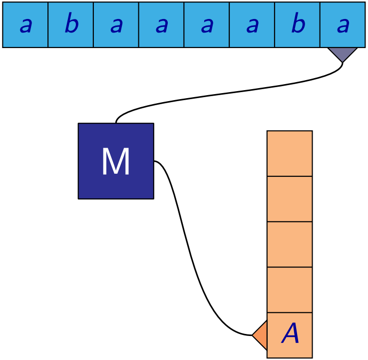
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



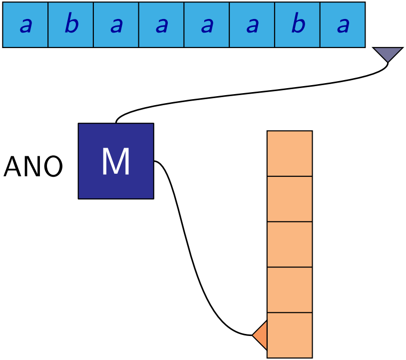
Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



Zásobníkový automat

- Slovo *abaaaaba* patří do jazyka $L = \{w(w)^R \mid w \in \{a, b\}^*\}$
- Při čtení a, stavu U a vrcholu zásobníku B, nemění stav a uloží A na zásobník



Zásobníkový automat

- Uvedený zásobníkový automat se nemůže jednoznačně rozhodnout, jak má pokračovat. Musí „uhádnout“, kde je půlka slova.
- Na rozdíl od konečných automatů je deterministická verze zásobníkových slabší a proto definujeme přímo nedeterministické

Zásobníkový automat

Definice

Zásobníkový automat je uspořádaná šestice $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$, kde

- Q je konečná neprázdná množina stavů
- Σ je konečná neprázdná množina zvaná vstupní abeceda
- Γ je konečná neprázdná množina zvaná zásobníková abeceda
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}$ je (nedeterministická) přechodová funkce
- $q_0 \in Q$ je počáteční stav
- $Z_0 \in \Gamma$ je počáteční zásobníkový symbol

Zásobníkový automat

- **Konfigurace ZA** je trojice (q, w, α), kde $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$.
- Konfiguraci (q_0, w, Z_0), kde $w \in \Sigma^*$, nazýváme **počáteční**
- Konfiguraci $(q, \varepsilon, \varepsilon)$, kde $q \in Q$, nazýváme **koncová**

Zásobníkový automat

- **Konfigurace ZA** je trojice (q, w, α), kde $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$.
- Konfiguraci (q_0, w, Z_0), kde $w \in \Sigma^*$, nazýváme **počáteční**
- Konfiguraci $(q, \varepsilon, \varepsilon)$, kde $q \in Q$, nazýváme **koncová**
- Relaci $\vdash$ mezi konfiguracemi definujeme tak, že $(q, aw, X\beta) \vdash (q', w, \alpha\beta)$ právě když $(q', \alpha) \in \delta(q, a, X)$ pro nějaké $a \in (\Sigma \cup \{\varepsilon\})$, $w \in \Sigma^*$, $\beta \in \Gamma^*$.
- Relace dosažitelnosti mezi konfiguracemi $\vdash^*$ je reflexivní tranzitivní uzávěr relace $\vdash$

Zásobníkový automat

- **Konfigurace ZA** je trojice (q, w, α), kde $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$.
- Konfiguraci (q_0, w, Z_0), kde $w \in \Sigma^*$, nazýváme **počáteční**
- Konfiguraci $(q, \varepsilon, \varepsilon)$, kde $q \in Q$, nazýváme **koncová**
- Relaci $\vdash$ mezi konfiguracemi definujeme tak, že $(q, aw, X\beta) \vdash (q', w, \alpha\beta)$ právě když $(q', \alpha) \in \delta(q, a, X)$ pro nějaké $a \in (\Sigma \cup \{\varepsilon\})$, $w \in \Sigma^*$, $\beta \in \Gamma^*$.
- Relace dosažitelnosti mezi konfiguracemi $\vdash^*$ je reflexivní tranzitivní uzávěr relace $\vdash$
- Slovo $w \in \Sigma^*$ je **přijímáno** ZA $\mathcal{M}$ právě tehdy, když $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ pro nějaké $q \in Q$
- **Jazyk rozpoznávaný ZA** $\mathcal{M}$ je jazyk $L(\mathcal{M}) = \{w \in \Sigma^* \mid w \text{ je přijímáno ZA } \mathcal{M}\}$.

Zásobníkový automat

- Definovali jsme přijímání prázdným zásobníkem
- K ZA $\mathcal{M}$ můžeme přidat množinu **přijímajících** (koncových) stavů $F \subseteq Q$
- Potom můžeme definovat jazyk přijímaný koncovým stavem $L_{KS}(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \alpha) \text{ pro něj. } q \in F, \alpha \in \Gamma^*\}$
- Obě zmíněné definice přijímání ZA jsou ekvivalentní - jazyk je přijímán nějakým ZA $\mathcal{M}$ koncovým stavem právě tehdy, když je rozpoznáván nějakým ZA $\mathcal{M}'$ prázdným zásobníkem

Zásobníkový automat

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, U, Z)$, kde

- $Q = \{U, M\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{Z, I\}$
- $$\begin{array}{ll} \delta(U, a, Z) = \{(U, I)\} & \delta(U, b, Z) = \emptyset \\ \delta(U, a, I) = \{(U, II)\} & \delta(U, b, I) = \{(M, \varepsilon)\} \\ \delta(M, a, I) = \emptyset & \delta(M, b, I) = \{(M, \varepsilon)\} \\ \delta(M, a, Z) = \emptyset & \delta(M, b, Z) = \emptyset \end{array}$$

Poznámka: Často se uvádí jen ty přechody přechodové funkce, které nejsou do prázdné množiny, tedy kdy je skutečně nějaký přechod definován

Příklad: $L = \{w(w)^R \mid w \in \{a,b\}^*\}$
$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, U, Z)$, kde

- $Q = \{U, M\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{Z, A, B\}$
- $$\begin{array}{ll}
\delta(U,a,Z) = \{(U,A)\} & \delta(U,b,Z) = \{(U,B)\} \\
\delta(U,a,A) = \{(U,AA),(M,AA)\} & \delta(U,b,A) = \{(U,BA),(M,BA)\} \\
\delta(U,a,B) = \{(U,AB),(M,AB)\} & \delta(U,b,B) = \{(U,BB),(M,BB)\} \\
\delta(M,a,A) = \{(M,\varepsilon)\} & \delta(M,b,A) = \emptyset \\
\delta(M,a,B) = \emptyset & \delta(M,b,B) = \{(M,\varepsilon)\} \\
\delta(M,a,Z) = \emptyset & \delta(M,b,Z) = \emptyset \\
\delta(U,\varepsilon,Z) = \{(U,\varepsilon)\} & \delta(M,\varepsilon,Z) = \emptyset \\
\delta(U,\varepsilon,A) = \emptyset & \delta(M,\varepsilon,A) = \emptyset \\
\delta(U,\varepsilon,B) = \emptyset & \delta(M,\varepsilon,B) = \emptyset
\end{array}$$

Lemma

Ke každé bezkontextové gramatice G lze sestrojit zásobníkový automat $\mathcal{M}$ (s jedním stavem) tž. $L(\mathcal{M}) = L(G)$.

Důkaz: Pro BG $G = (\Pi, \Sigma, S, P)$ vytvoříme $\mathcal{M} = (\{q_0\}, \Sigma, \Pi \cup \Sigma, \delta, q_0, S)$, kde

- pro $X \in \Pi$: $\delta(q_0, \varepsilon, X) = \{(q_0, \alpha) \mid (X \to \alpha) \in P\}$,
- pro $a \in \Sigma$: $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
- v ostatních případech přiřadíme $\emptyset$

Indukcí můžeme dokázat

$$S \Rightarrow_G^* u\alpha \Leftrightarrow_{df} (q_0, u, S) \vdash_{\mathcal{M}}^* (q_0, \varepsilon, \alpha)$$

kde $u \in \Sigma^*, \alpha \in \{\varepsilon\} \cup \Pi(\Pi \cup \Sigma)^*$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro *aaaabbbb*
$S \Rightarrow aSb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb)$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro *aaaabbbb*
$S \Rightarrow aSb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro $aaaabbbb$
$S \Rightarrow aSb \Rightarrow aaSbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb)$

Ekvivalence bezkontextových gramatik a zás. automatů

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro *aaaabbbb*
$S \Rightarrow aSb \Rightarrow aaSbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb) \vdash (q_0, aabbbb, Sbb)$

Ekvivalence bezkontextových gramatik a zás. automatů

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro *aaaabbbb*
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb) \vdash (q_0, aabbbb, Sbb) \vdash (q_0, aabbbb, aSbbb)$

Ekvivalence bezkontextových gramatik a zás. automatů

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro $aaaabbbb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb) \vdash (q_0, aabbbb, Sbb) \vdash (q_0, aabbbb, aSbbb)$
$\vdash (q_0, abbbb, Sbbb)$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \to aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro *aaaabbbb*
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaabbbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb) \vdash (q_0, aabbbb, Sbb) \vdash (q_0, aabbbb, aSbbb)$
$\vdash (q_0, abbbb, Sbbb) \vdash (q_0, abbbb, abbbb)$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro $aaaabbbb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaabbbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb) \vdash (q_0, aabbbb, Sbb) \vdash (q_0, aabbbb, aSbbb)$
$\vdash (q_0, abbbb, Sbbb) \vdash (q_0, abbbb, abbbb) \vdash (q_0, bbbb, bbbb)$

Příklad: $L = \{a^i b^i \mid i \geq 1\}$
$G = (\{S\}, \{a, b\}, S, P)$, kde P je definována jako: $S \rightarrow aSb|ab$

Sestrojíme $\mathcal{M} = (\{q_0\}, \{a, b\}, \{S, a, b\}, \delta, q_0, S)$, kde
$\delta(q_0, \varepsilon, S) = \{(q_0, aSb), (q_0, ab)\}$, $\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$,
$\delta(q_0, b, b) = \{(q_0, \varepsilon)\}$

Ukážeme si situaci pro $aaaabbbb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaabbbb$
$(q_0, aaaabbbb, S) \vdash (q_0, aaaabbbb, aSb) \vdash (q_0, aaabbbb, Sb)$
$\vdash (q_0, aaabbbb, aSbb) \vdash (q_0, aabbbb, Sbb) \vdash (q_0, aabbbb, aSbbb)$
$\vdash (q_0, abbbb, Sbbb) \vdash (q_0, abbbb, abbbb) \vdash (q_0, bbbb, bbbb)$
$\vdash (q_0, bbb, bbb) \vdash (q_0, bb, bb) \vdash (q_0, b, b) \vdash (q_0, \varepsilon, \varepsilon)$

Lemma

Ke každému zásobníkovému automatu $\mathcal{M}$ s jedním stavem lze sestrojit bezkontextovou gramatiku G tž. $L(G) = L(\mathcal{M})$.

Důkaz: Pro ZA $\mathcal{M} = (\{q_0\}, \Sigma, \Gamma, \delta, q_0, Z_0)$, kde $\Sigma \cap \Gamma = \emptyset$ vytvoříme $G = (\Gamma, \Sigma, Z_0, P)$, kde $(A \longrightarrow a\alpha) \in P \Leftrightarrow \delta(q_0, a, A) \ni (q_0, \alpha)$, kde $u \in \Sigma^*, \alpha \in \Gamma^*$

Indukcí můžeme dokázat

$$Z_0 \Rightarrow_G^* u\alpha \Leftrightarrow (q_0, u, Z_0) \vdash_{\mathcal{M}}^* (q_0, \varepsilon, \alpha)$$

kde $u \in \Sigma^*, \alpha \in \Gamma^*$

Lemma

Ke každému zásobníkovému automatu $\mathcal{M}$ lze sestrojit zásobníkový automat $\mathcal{M}'$ s jedním stavem tž. $L(\mathcal{M}') = L(\mathcal{M})$.

Myšlenka důkazu:

- Stav automatu $\mathcal{M}$ si budeme pamatovat na zásobníku.
- Pro $\delta(q, a, X) = \{(q', \varepsilon)\}$ musíme kontrolovat nejen, že jsme ve stavu q, ale také, že se dostaneme do stavu q'
- Každý zásobníkový symbol automatu $\mathcal{M}'$ je tedy trojice, kde si pamatujeme zásobníkový symbol, aktuální stav a aktuální stav ze symbolu o jedna níže na zásobníku
- Převod na ZA s jedním stavem tedy způsobí nárust počtu zásobníkových symbolů

Omezení bezkontextových jazyků

- Ani bezkontextové gramatiky a zásobníkové automaty nejsou dostatečně silné, aby rozpoznávaly všechny jazyky
- Existuje verze pumping lemmatu pro bezkontextové jazyky, kterou je možno využít k důkazu nebezkontextovosti jazyka
- Např. $L = \{a^i b^i c^i \mid i \geq 0\}$ nebo $L = \{ww \mid w \in \{a, b\}^*\}$ nejsou bezkontextové
- Potřebujeme nějaký silnější mechanismus pro rozpoznávaní takových jazyků
  - Turingův stroj
  - Lineárně omezený automat

- Rozšíříme deterministické konečné automaty o možnost zápisu na vstupní pásku a pohyb čtecí hlavy oběma směry

Turingův stroj

- Rozšíříme deterministické konečné automaty o možnost zápisu na vstupní pásku a pohyb čtecí hlavy oběma směry

Definice

Formálně je **Turingův stroj** definován jako šestice $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ kde:

- Q je konečná množina **stavů**
- Γ je konečná množina **páskových symbolů**
- $\Sigma \subseteq \Gamma, \Sigma \neq \emptyset$ je konečná množina **vstupních symbolů**
- $\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$ je **přechodová funkce**
- $q_0 \in Q$ je **počáteční stav**
- $F \subseteq Q$ je množina **koncových stavů**

Turingův stroj

- Předpokládáme, že v $\Gamma - \Sigma$ je vždy speciální prvek $\square$ označující prázdný znak
- Konfigurace je dána slovem na pásce, stavem a pozicí čtecí hlavy
- Konfigurace je **počáteční**, pokud je hlava na prvním symbolu, stav q_0 a na pásce jsou symboly jen z množiny Σ
- Konfigurace je **koncová**, je-li stav z množiny F

Máme-li stav q a b na vstupu, tak pro:

- $\delta(q, b) = (q', b', 0)$ změníme stav na q', změníme na aktuální pozici b na b'
- $\delta(q, b) = (q', b', +1)$ změníme stav na q', změníme na aktuální pozici b na b' a posuneme aktuální pozici o 1 doprava
- $\delta(q, b) = (q', b', -1)$ změníme stav na q', změníme na aktuální pozici b na b' a posuneme aktuální pozici o 1 doleva

Turingův stroj

- Výpočet v koncovém stavu **končí**
- Slovo je přijato, pokud na něm stroj někdy dojde do přijímajícího stavu
- Slovo není přijato, pokud běh stroje nikdy neskončí
- Jazyk $L(\mathcal{M})$ Turingova stroje $\mathcal{M}$ je množina všech slov, která stroj $\mathcal{M}$ přijímá

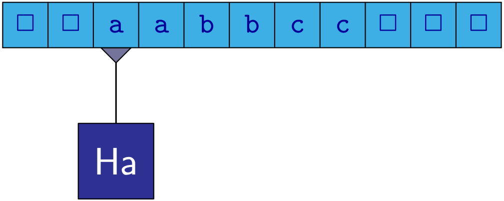
Turingův stroj

- Výpočet v koncovém stavu **končí**
- Slovo je přijato, pokud na něm stroj někdy dojde do přijímajícího stavu
- Slovo není přijato, pokud běh stroje nikdy neskončí
- Jazyk $L(\mathcal{M})$ Turingova stroje $\mathcal{M}$ je množina všech slov, která stroj $\mathcal{M}$ přijímá
- U Turingova stroje nás někdy zajímá, co zůstane na pásce po skončení běhu.

Turingův stroj

- Výpočet v koncovém stavu **končí**
- Slovo je přijato, pokud na něm stroj někdy dojde do přijímajícího stavu
- Slovo není přijato, pokud běh stroje nikdy neskončí
- Jazyk $L(\mathcal{M})$ Turingova stroje $\mathcal{M}$ je množina všech slov, která stroj $\mathcal{M}$ přijímá
- U Turingova stroje nás někdy zajímá, co zůstane na pásce po skončení běhu.
- Turingův stroj můžeme reprezentovat grafem obdobně jako KA. Jen popis šipek rozšíříme o informaci, co se zapíše na pásku (pokud se obsah pásky mění) a jak se posune hlava.

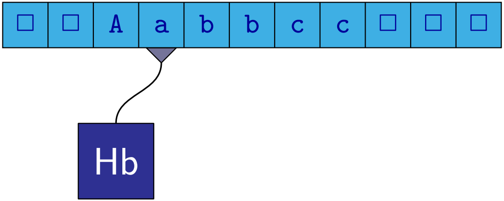
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.

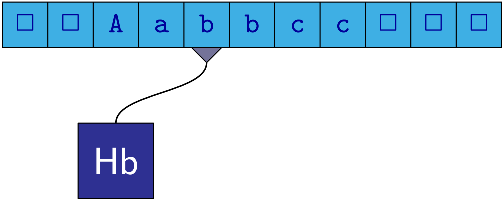
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.

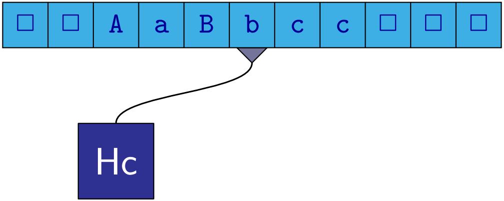
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

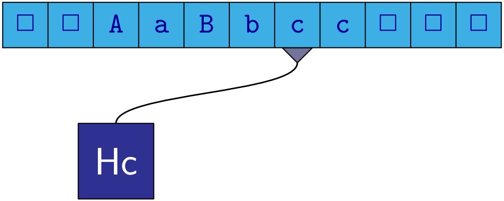
1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.

Turingův stroj



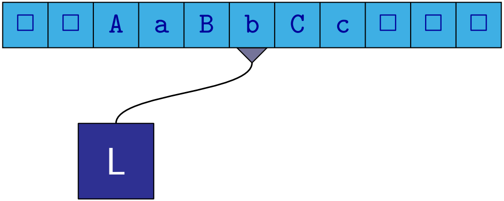
Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.



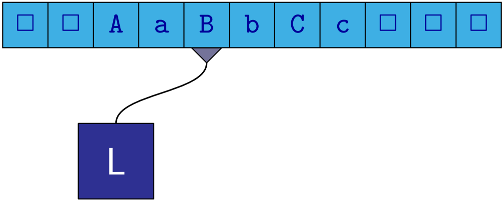
Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

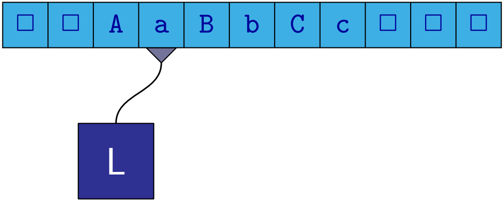
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.

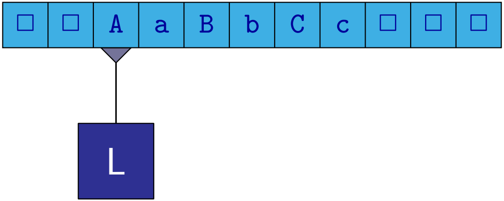
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

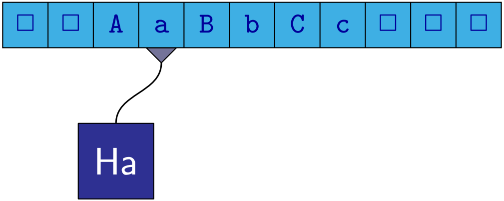
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.

Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

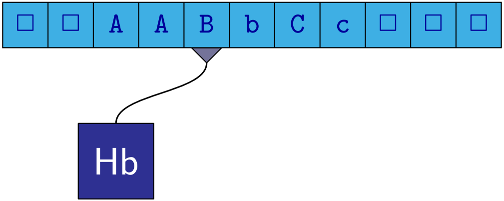
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

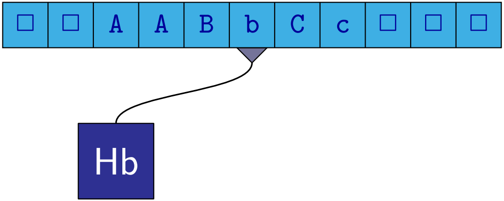
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

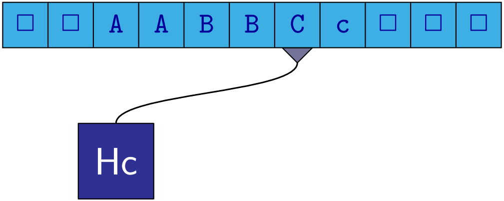
1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.
3. Čte do prvního `c`, nahradí jej `C`. Na □ se zacyklí.
4. Vrací se doleva na nejbližší `A`.
5. Pokračuje bodem 1



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

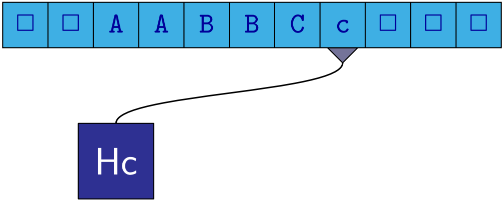
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

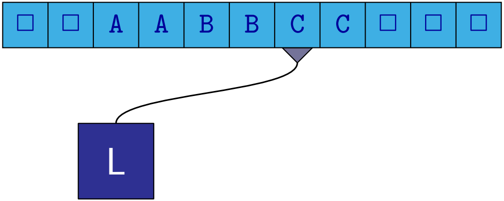
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

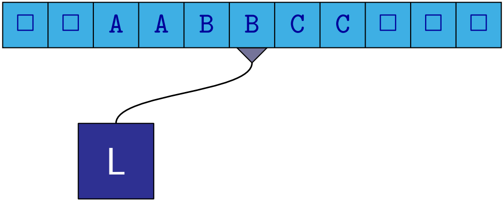
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

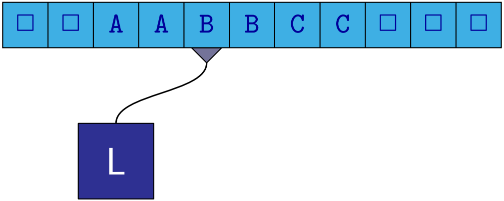
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

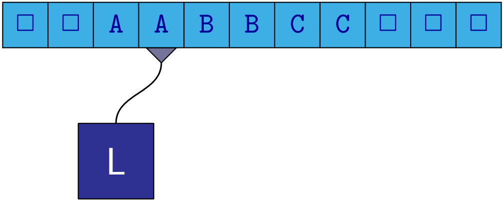
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

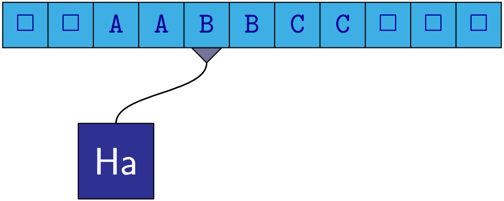
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

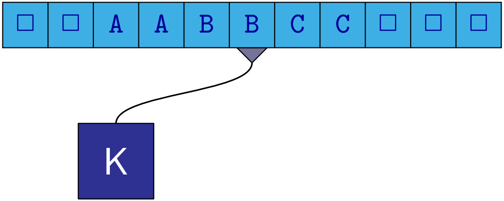
1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.
3. Čte do prvního `c`, nahradí jej `C`. Na □ se zacyklí.
4. Vrací se doleva na nejbližší `A`.
5. Pokračuje bodem 1

Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

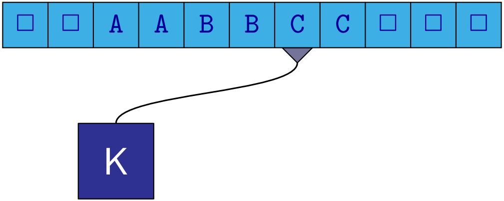
1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se. Pokud už `a` není, zkontroluje, zda už jsou jen velká písmena.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.
3. Čte do prvního `c`, nahradí jej `C`. Na □ se zacyklí.
4. Vrací se doleva na nejbližší `A`.
5. Pokračuje bodem 1



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

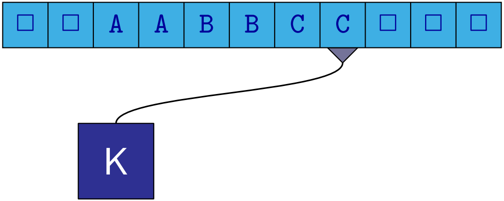
1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se. Pokud už a není, zkontroluje, zda už jsou jen velká písmena.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

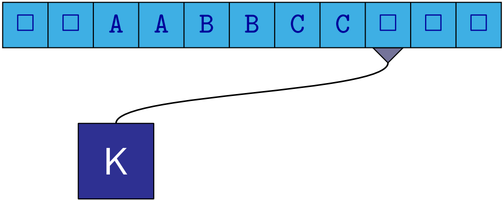
1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se. Pokud už `a` není, zkontroluje, zda už jsou jen velká písmena.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.
3. Čte do prvního `c`, nahradí jej `C`. Na □ se zacyklí.
4. Vrací se doleva na nejbližší `A`.
5. Pokračuje bodem 1



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se. Pokud už `a` není, zkontroluje, zda už jsou jen velká písmena.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.
3. Čte do prvního `c`, nahradí jej `C`. Na □ se zacyklí.
4. Vrací se doleva na nejbližší `A`.
5. Pokračuje bodem 1

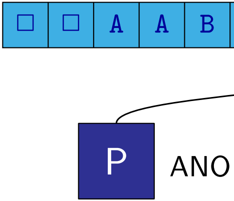
Turingův stroj



Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

1. Čte do prvního a, nahradí jej A. Pokud najde dříve b nebo c, zacyklí se. Pokud už a není, zkontroluje, zda už jsou jen velká písmena.
2. Čte do prvního b, nahradí jej B. Na □ nebo c se zacyklí.
3. Čte do prvního c, nahradí jej C. Na □ se zacyklí.
4. Vrací se doleva na nejbližší A.
5. Pokračuje bodem 1

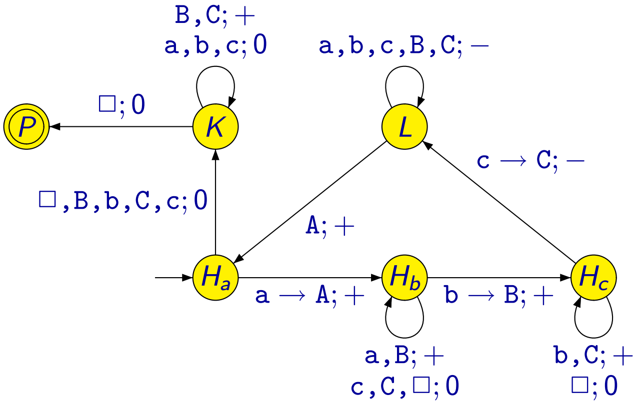
Turingův stroj



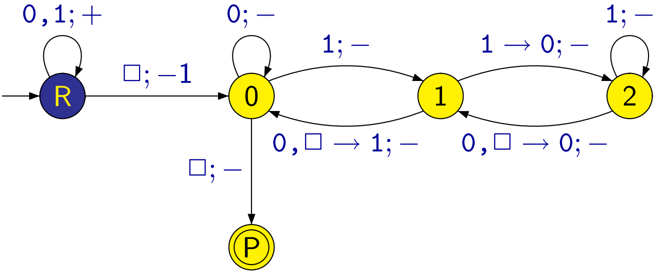
Jazyk $L = \{a^i b^i c^i \mid i \geq 0\}$

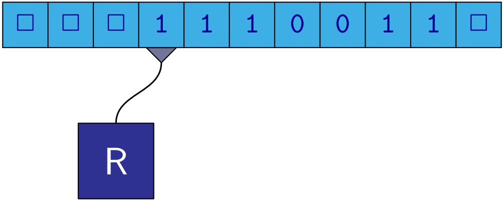
1. Čte do prvního `a`, nahradí jej `A`. Pokud najde dříve `b` nebo `c`, zacyklí se. Pokud už `a` není, zkontroluje, zda už jsou jen velká písmena.
2. Čte do prvního `b`, nahradí jej `B`. Na □ nebo `c` se zacyklí.
3. Čte do prvního `c`, nahradí jej `C`. Na □ se zacyklí.
4. Vrací se doleva na nejbližší `A`.
5. Pokračuje bodem 1

Turingův stroj

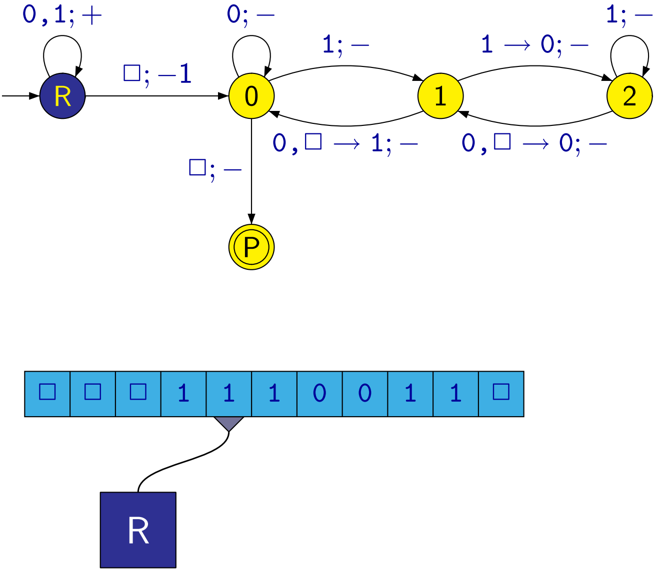


Turingův stroj - násobení třemi

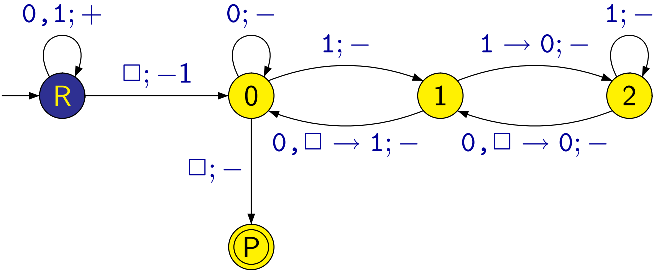


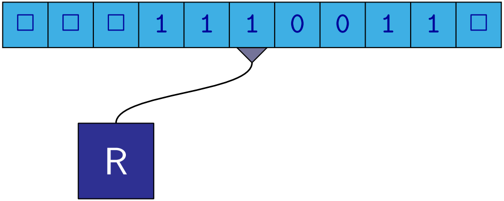


Turingův stroj - násobení třemi

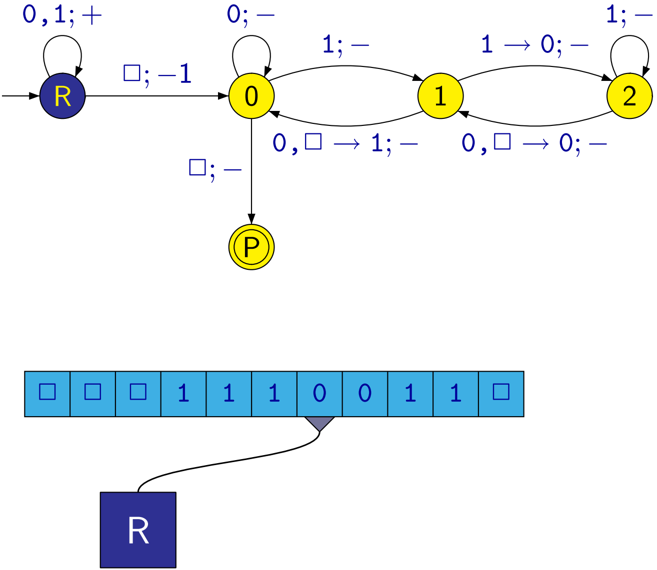


Turingův stroj - násobení třemi

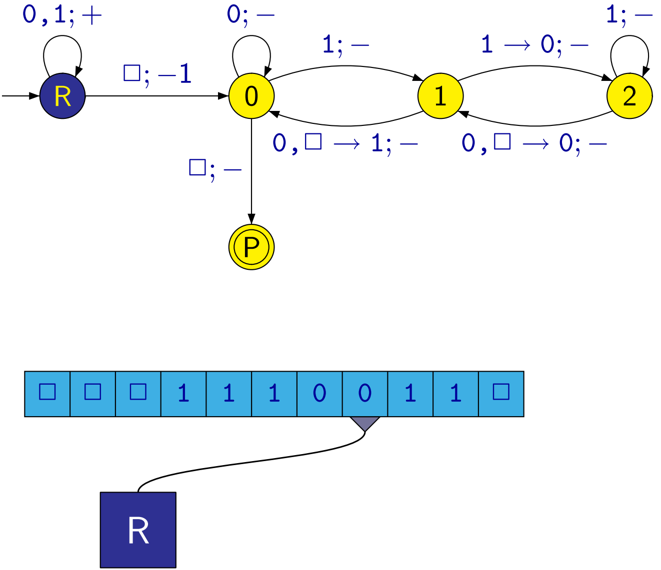




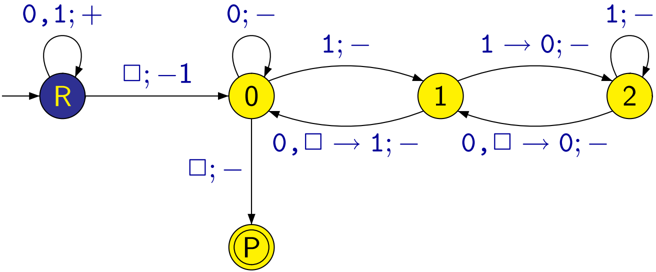
Turingův stroj - násobení třemi

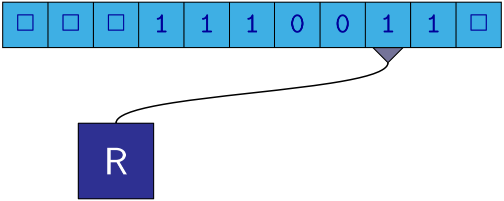


Turingův stroj - násobení třemi

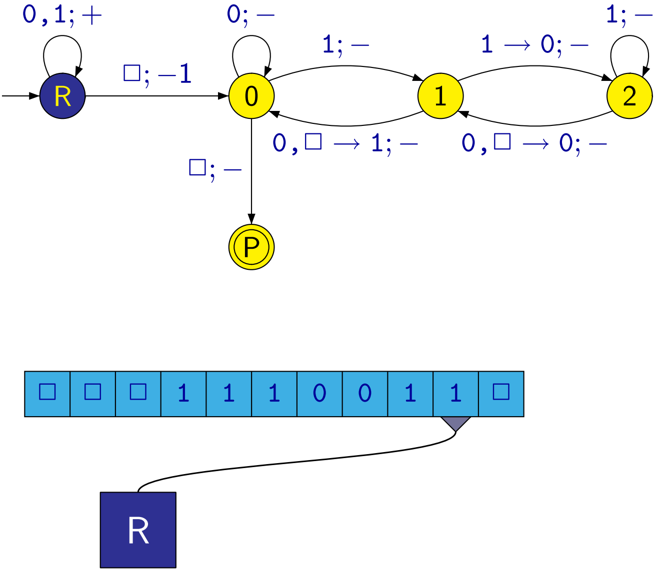


Turingův stroj - násobení třemi

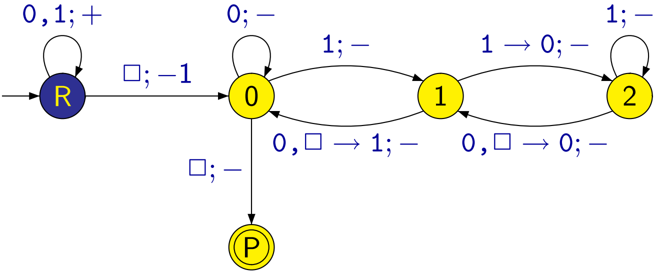


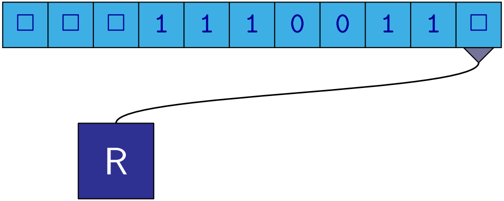


Turingův stroj - násobení třemi

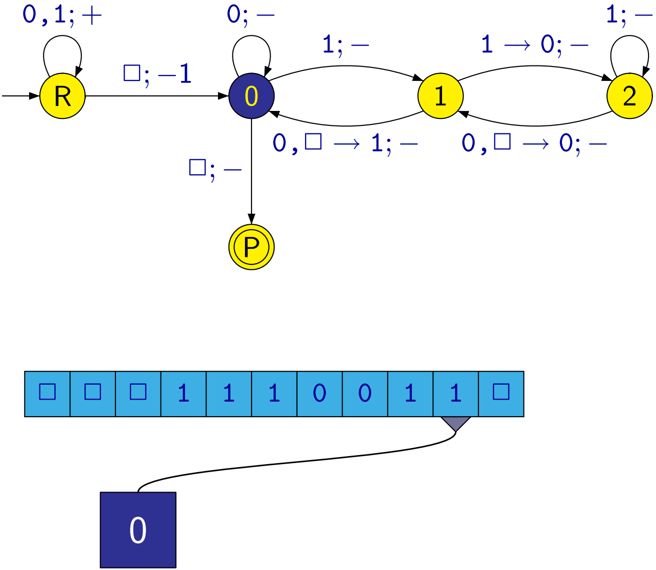


Turingův stroj - násobení třemi

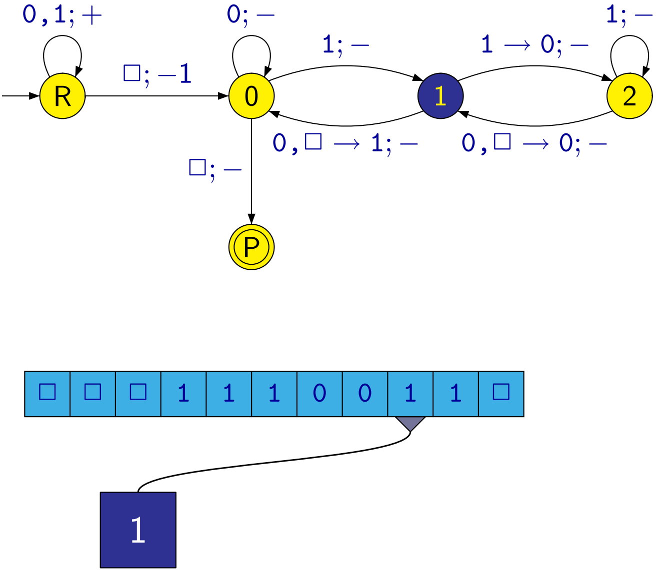




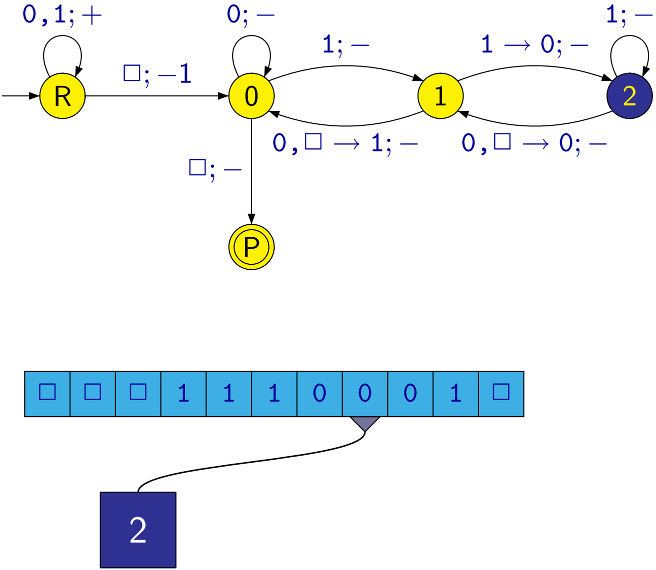
Turingův stroj - násobení třemi



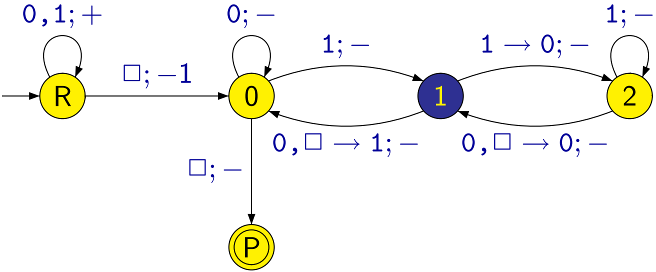
Turingův stroj - násobení třemi

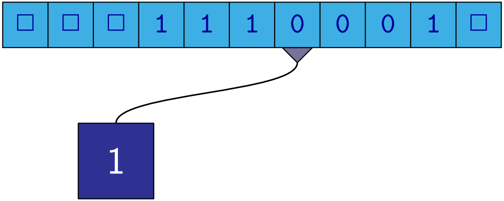


Turingův stroj - násobení třemi

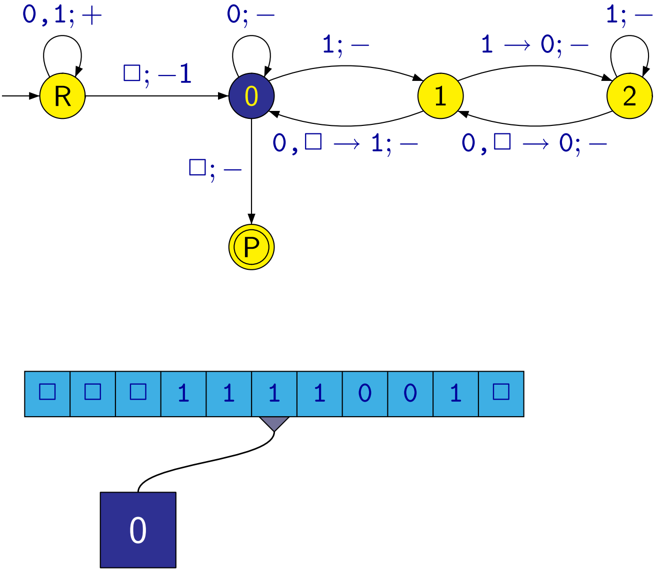


Turingův stroj - násobení třemi

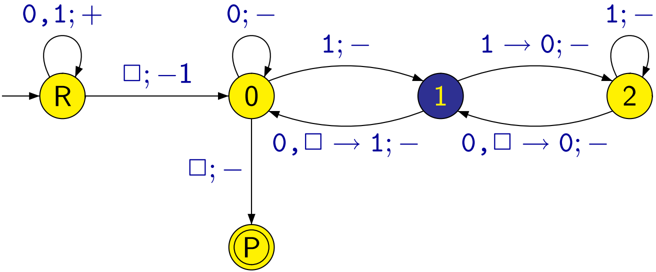


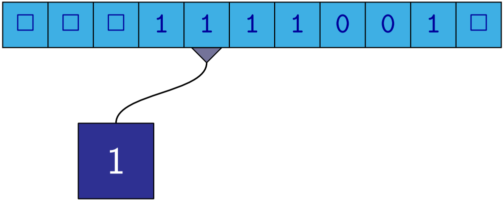


Turingův stroj - násobení třemi

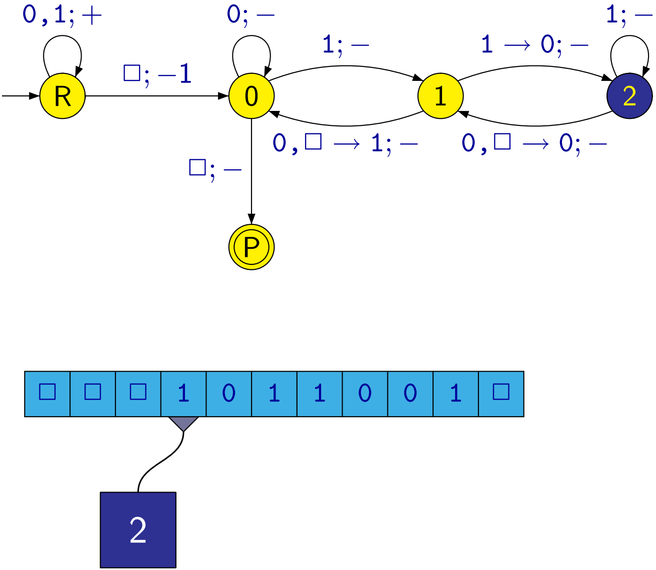


Turingův stroj - násobení třemi

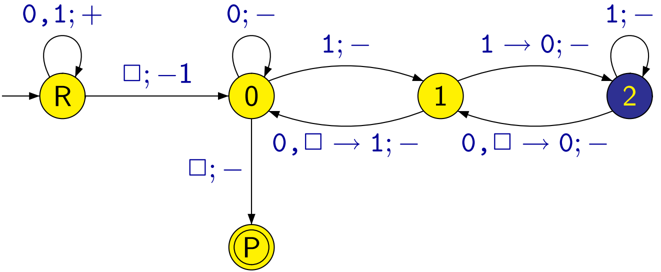


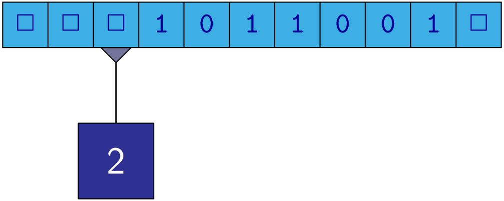


Turingův stroj - násobení třemi

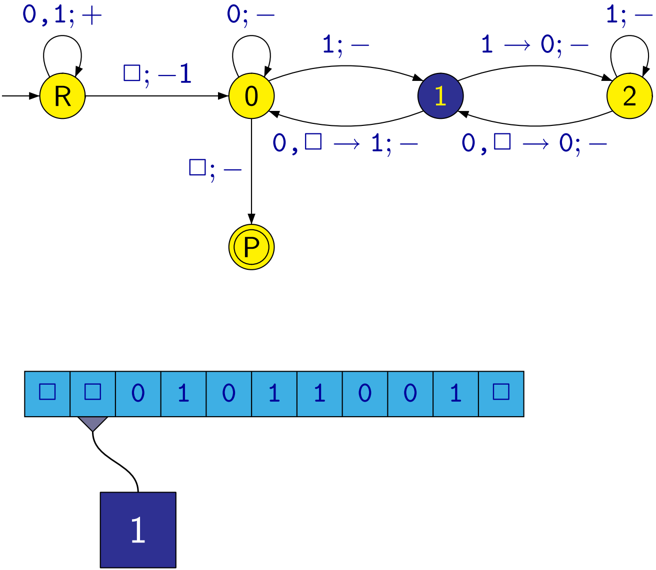


Turingův stroj - násobení třemi

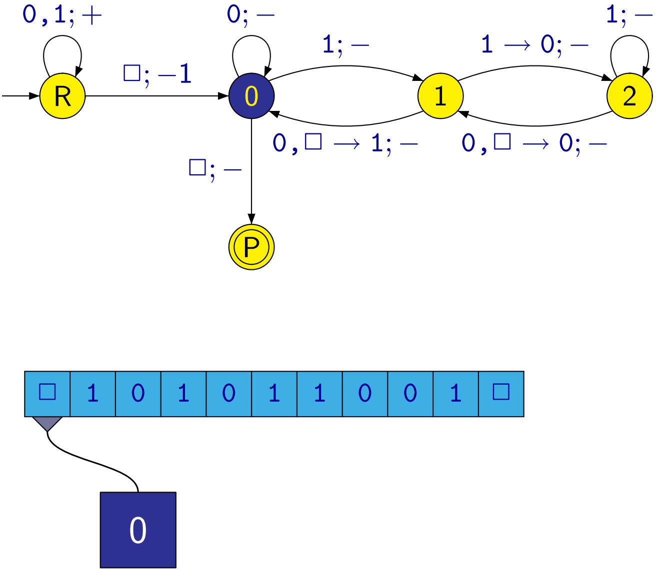




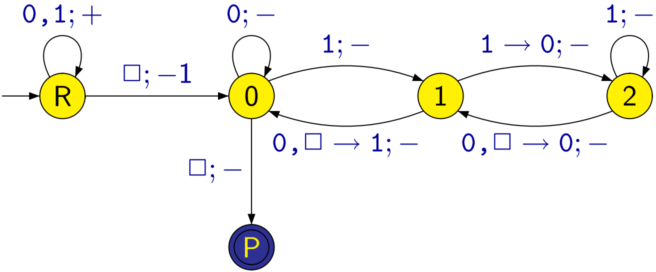
Turingův stroj - násobení třemi

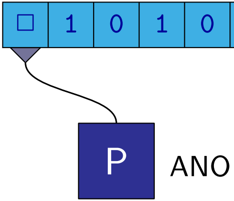


Turingův stroj - násobení třemi



Turingův stroj - násobení třemi





Lineárně omezený automat

- Turingův stroj, který využívá jen úsek pásky, kde je zapsáno vstupní slovo
- Představa levé a pravé zarážky kolem slova, které nemohou být přepsány
- Z levé zarážky je možný pohyb jen vpravo, z pravé zarážky jen vlevo
- LBA (linear bounded automata) se uvažují v nedeterministické verzi
- Je otevřená otázka, jestli jsou deterministické LBA stejně „silné“ jako nedeterministické

Generativní gramatika

Definice

Generativní gramatika je dána čtveřicí parametrů $G = (\Pi, \Sigma, S, P)$, kde

- Π je konečná množina neterminálů
- Σ je konečná množina terminálů, $\Pi \cap \Sigma = \emptyset$
- $S \in \Pi$ je počáteční neterminál
- P je konečná množina pravidel typu $\alpha \rightarrow \beta$, kde $\alpha \in (\Pi \cup \Sigma)^* \Pi (\Pi \cup \Sigma)^*$ a $\beta \in (\Pi \cup \Sigma)^*$.

- $\mu_1 \alpha \mu_2 \Rightarrow_G \mu_1 \beta \mu_2$ pokud $\alpha \rightarrow \beta$ je pravidlo v P
- $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$

Chomského hierarchie

Podle tvaru pravidel, která v gramatice povolíme, dělíme gramatiky na gramatiky typu:

0 - Obecné **generativní gramatiky** - pravidla bez omezení

1 - **Kontextové gramatiky** - pravidla tvaru $\alpha X \beta \rightarrow \alpha\gamma\beta$, kde $|\gamma| \geq 1$ (Výjimka $S \rightarrow \varepsilon$, ale S pak není na pravé straně žádného pravidla)

2 - **Bezkontextové gramatiky** - pravidla tvaru $X \rightarrow \gamma$

3 - **Regulární gramatiky** - pravidla tvaru $X \rightarrow wY$ nebo $X \rightarrow w$

kde $\alpha, \beta, \gamma \in (\Pi \cup \Sigma)^*, X \in \Pi, w \in \Sigma^*$

Chomského hierarchie

Podle tvaru pravidel, která v gramatice povolíme, dělíme gramatiky na gramatiky typu:

0 - Obecné **generativní gramatiky** - pravidla bez omezení

1 - **Kontextové gramatiky** - pravidla tvaru $\alpha X\beta \rightarrow \alpha\gamma\beta$, kde $|\gamma| \geq 1$ (Výjimka $S \rightarrow \varepsilon$, ale S pak není na pravé straně žádného pravidla)

2 - **Bezkontextové gramatiky** - pravidla tvaru $X \rightarrow \gamma$

3 - **Regulární gramatiky** - pravidla tvaru $X \rightarrow wY$ nebo $X \rightarrow w$

kde $\alpha, \beta, \gamma \in (\Pi \cup \Sigma)^*, X \in \Pi, w \in \Sigma^*$

- Jazyk je typu i, jestliže jej generuje nějaká gramatika typu i
- Označíme-li $\mathcal{L}_i$ třídu jazyků typu i, pak $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$

Chomského hierarchie

Pojmenování jazyků jednotlivých typů:

0 - Rekurzivně spočetné

1 - Kontextové

2 - Bezkontextové

3 - Regulární

Pojmenování jazyků jednotlivých typů:

0 - Rekurzivně spočetné

1 - Kontextové

2 - Bezkontextové

3 - Regulární

Každé třídě jazyků odpovídá typ stroje nebo automatu, který dokáže rozpoznat všechny jazyky z dané třídy

0 - Turingův stroj (deterministický nebo nedeterministický)

1 - Lineárně omezený automat (nedeterministický, pro deterministické to není známo)

2 - Zásobníkový automat (jen nedeterministický)

3 - Konečný automat (deterministický nebo nedeterministický)