

65. Zásobníkové automaty (jazyky přijímané ZA, varianty ZA).

Zásobníkový automat

(Nedeterministický) zásobníkový automat je sedmice $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$, kde:

- Q je konečná, neprázdná množina stavů
- Σ je abeceda (konečná, neprázdná množina symbolů)
- Γ je zásobníková abeceda (konečná, neprázdná množina zásobníkových symbolů)
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ (v horním indexu je $Q \times \Gamma^*$) je přechodová funkce
- $s_0 \in Q$ je počáteční stav
- $Z_0 \in \Gamma$ je startující symbol zásobníku
- $F \subseteq Q$ je množina koncových stavů

Zápisem $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ rozumíme zobrazení z množiny $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ do množiny $2^{Q \times \Gamma^*}$. Množina $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ obsahuje trojice (q, a, z) , kde q je nějaký stav, a je symbol abecedy, nebo epsilon, z je zásobníkový symbol. $2^{Q \times \Gamma^*}$ je množina množin obsahujících prvky (r, γ) , kde r je nějaký stav a γ je řetězec zásobníkových symbolů. Zobrazení δ nám tedy říká, co je možné provést za předpokladu, že $(r, \gamma) \in \delta(q, a, z)$, tedy pokud se nacházíme v nějakém stavu q , pokud přečteme a ze vstupu a pokud současně odebereme z ze zásobníku, je možné přejít do stavu r a na zásobník vložit řetězec γ . V následujícím textu budeme uvažovat o tom, že při vložení řetězce γ na zásobník odpovídá novému vrcholu zásobníku nejlevější symbol v γ .

Pozn.: Všimněme si, že v rámci provedení každého přechodu je nutné přečíst právě jeden symbol ze zásobníku. Pokud žádný symbol přečíst nechceme (líbilo by se nám, kdyby po provedení přechodu zůstal zásobník nezměněn), přesto musíme současný vrchol zásobníku odstranit. Je však možné jej v rámci přechodu ihned na zásobník vrátit, takže se se zásobníkem efektivně nestane nic.

Pozn.: Přechodovou funkci je možné chápat i tak, že je vždy totální (je definovaná pro každý možný stav, symbol abecedy (nebo ϵ) a zásobníkový symbol), ovšem v některých případech může vstup zobrazovat na prázdnou množinu $\emptyset$, což znamená, že z daného stavu s daným symbolem a s daným zásobníkovým symbolem nevede přechod do žádného stavu.

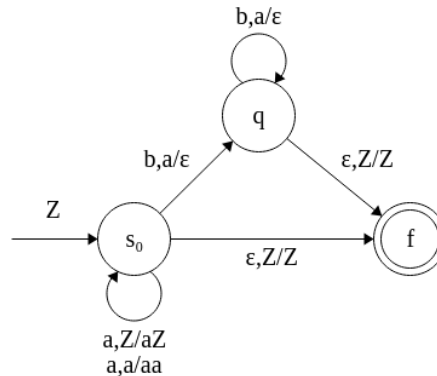
Zásobníkový automat znázorňujeme jako orientovaný graf s ohodnocením hran řetězcí ve tvaru $a, z/\gamma$, kde $a \in \Sigma \cup \{\epsilon\}$ označuje symbol čtený ze vstupu, $z \in \Gamma$ označuje symbol odebíraný ze zásobníku

a $\gamma \in \Gamma^*$ označuje řetězec zásobníkových symbolů vkládaných na zásobník. Zdvojený obvod uzlu označuje koncový stav, šipka vedoucí do stavu odnikud označuje počáteční stav.

Startujícím symbolem zásobníku typicky popisujeme vstupní hranu počátečního stavu.

V níže uvedeném případě $A = (\{s_0, q, f\}, \{a, b\}, \{a, b, Z\}, \delta, s_0, Z_0, \{f\})$, neboli

- $Q = \{s_0, q, f\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, Z\}$
- δ :
 - $\delta(s_0, \varepsilon, Z) = \{(f, Z)\}$
 - $\delta(s_0, a, Z) = \{(s_0, aZ)\}$
 - $\delta(s_0, a, a) = \{(s_0, aa)\}$
 - $\delta(s_0, b, a) = \{(q, \varepsilon)\}$
 - $\delta(q, b, a) = \{(q, \varepsilon)\}$
 - $\delta(q, \varepsilon, Z) = \{(f, \varepsilon)\}$
- $Z_0 = Z$
- $F = \{f\}$



At' $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ je zásobníkový automat. **Konfigurace** zásobníkového automatu A je trojice

$(q, w, \gamma) \in Q \times \Sigma^* \times \Gamma^*$. Konfigurací rozumíme aktuální stav výpočtu zásobníkového automatu.

Prvek

$q \in Q$ značí stav, v němž se aktuálně nacházíme, $w \in \Sigma^*$ je dosud nepřečtená část vstupu a $\gamma \in \Gamma^*$ je obsah zásobníku.

At' $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ je zásobníkový automat. **Přechodová relace** mezi konfiguracemi je relace na množině konfigurací $\vdash \subseteq (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Sigma^* \times \Gamma^*)$ definovaná takto:

$$(q, aw, zy) \vdash (r, w, \lambda y) \Leftrightarrow q, r \in Q \wedge a \in \Sigma \cup \{\varepsilon\} \wedge w \in \Sigma^* \wedge z \in \Gamma \wedge \gamma, \lambda \in \Gamma^* \wedge (r, \lambda) \in \delta(q, a, z),$$

tedy ze stavu q přejdeme do stavu r , ze vstupu přečteme symbol a a na zásobníku nahradíme jediný zásobníkový symbol z za řetězec zásobníkových symbolů λ . Části w, γ zůstaly nedotčeny. Ve výše uvedeném automatu například platí $(s_0, bb, aaZ) \vdash (q, b, aZ)$.

K přechodové relaci $\vdash$ lze vytvořit její **reflexivní a tranzitivní uzávěr** $\vdash^*$, což je $\subseteq$ -minimální nadmnožina relace $\vdash$ taková, že je navíc reflexivní a tranzitivní. Jinak řečeno, do relace $\vdash$

přidáme minimální nutný počet dvojic (minimální vzhledem k relaci $\subseteq$) tak, aby po této změně byla reflexivní i tranzitivní.

Ať $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ je zásobníkový automat. **Jazyk přijímaný automatem** je množina

$$L(A) = \{w \in \Sigma^* \mid \exists f \in F : (s_0, w, Z_0) \vdash^* (f, \varepsilon, \gamma)\},$$

tedy slovo je automatem přijato tehdy, pokud je jeho čtení započato v počátečním stavu a je dočteno v nějakém koncovém stavu. Obsah zásobníku po dočtení slova w je nám lhostejný. Ilustrujme přijetí slova $aabb$ pomocí výše uvedeného automatu:

$$(s_0, aabb, Z) \vdash (s_0, abb, aZ) \vdash (s_0, bb, aaZ) \vdash (q, b, aZ) \vdash (q, \varepsilon, Z) \vdash (f, \varepsilon, Z),$$

tedy

$$(s_0, aabb, Z) \vdash^* (f, \varepsilon, Z),$$

kde f je koncový stav, tedy slovo $aabb$ je přijato automatem A .

Ať Σ je abeceda. Množina všech jazyků nad abecedou Σ je nespočetná množina 2^{Σ^*} . Množina všech jazyků, které mohou být přijaty nedeterministickým zásobníkovým automatem pracujícím s abecedou Σ , se nazývá množina **bezkontextových jazyků** $\mathcal{L}_2 \subset 2^{\Sigma^*}$. Platí $\mathcal{L}_3 \subset \mathcal{L}_2$, neboť každý konečný automat je možné realizovat pomocí zásobníkového automatu tak, že nebudeme se zásobníkem nijak pracovat (v rámci každého přechodu z něj odstraníme startující symbol a ihned jej vrátíme na zásobník). Současně existují jazyky, které jsou bezkontextové, ale nejsou regulární (např. $\{a^n b^n \mid n \geq 0\}$, což lze ukázat kombinací důkazu neregularity pomocí Myhill-Nerodovy věty a důkazu bezkontextovosti pomocí konstrukce zásobníkového automatu).

Příklady bezkontextových jazyků nad abecedou $\Sigma = \{a, b\}$:

- **libovolný regulární jazyk**
- $\{a^n b^n \mid n \geq 0\}$
- $\{ww^R \mid w \in \{a, b\}^*\}$
- $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}, \{w \in \{a, b\}^* \mid \#_a(w) \neq \#_b(w)\}, \{w \in \{a, b\}^* \mid \#_a(w) > \#_b(w)\}$
- $\text{co-}\{a^n b^n a^n \mid n \geq 0\}$
- $\text{co-}\{ww \mid w \in \{a, b\}^*\}$
- Dyckovy jazyky (jazyky správného uzávorkování)

Příklady jazyků nad abecedou $\Sigma = \{a, b\}$, které nejsou bezkontextové:

- $\{a^n b^n a^n \mid n \geq 0\}$
- $\{ww \mid w \in \{a, b\}^*\}$
- $\{ \langle \varphi \rangle \mid \langle \varphi \rangle \text{ je kód splnitelné výrokové formule nad kódovou abecedou } \{a, b\} \}$
- $\{ \langle S \rangle \mid \langle S \rangle \text{ je kód Postova systému nad kódovou abecedou } \{a, b\}, \text{ který nemá řešení} \}$

Varianty zásobníkových automatů

Klasická definice zásobníkového automatu se může lišit **na základě akceptující podmínky**. Ať $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ je zásobníkový automat. Definujeme:

- **zásobníkový automat přijímající dosažením koncového stavu**
 - $L(A) = \{w \in \Sigma^* \mid \exists f \in F : (s_0, w, Z_0) \vdash^* (f, \varepsilon, \gamma)\}$
 - řetězec je přijatý tehdy, pokud je dočtený v koncovém stavu
- **zásobníkový automat přijímající vyprázdněním zásobníku**
 - $L(A) = \{w \in \Sigma^* \mid (s_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)\}$
 - $F = \emptyset$
 - řetězec je přijatý tehdy, pokud běh automatu na tomto řetězci vede k vyprázdnění zásobníku
- **zásobníkový automat přijímající dosažením koncového stavu a vyprázdněním zásobníku**
 - $L(A) = \{w \in \Sigma^* \mid \exists f \in F : (s_0, w, Z_0) \vdash^* (f, \varepsilon, \varepsilon)\}$
 - řetězec je přijatý tehdy, pokud běh automatu na tomto řetězci vede k vyprázdnění zásobníku a pokud je po dočtení řetězce dosažen koncový stav

Tyto tři výpočetní modely jsou **ekvivalentní**, tedy pro každý bezkontextový jazyk najdeme model odpovídající každému z těchto tří typů automatů. Současně žádný z těchto modelů nemůže přijímat jazyk, který není bezkontextový. Tyto modely jsou mezi sebou také vzájemně převoditelné.

Pozn.: Samozřejmě, pokud se na jeden konkrétní zásobníkový automat podíváme postupně z těchto tří úhlů pohledu, může pokaždé přijímat jiný jazyk.

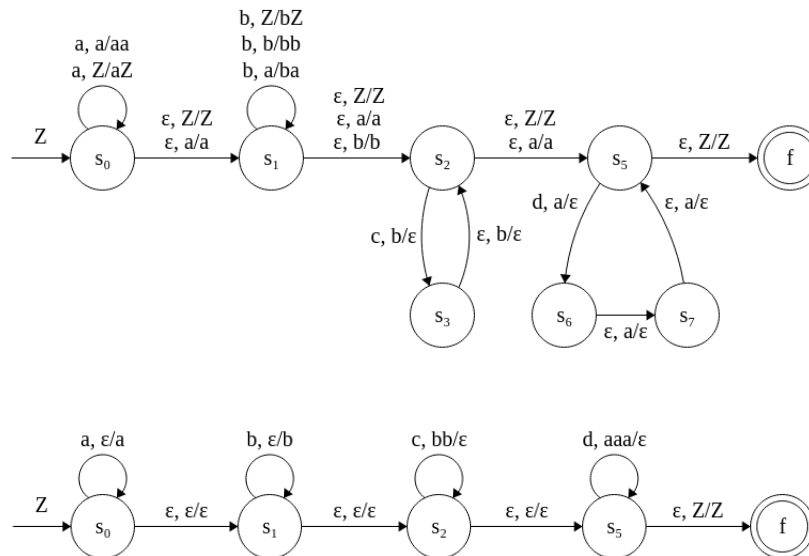
Rozšířený zásobníkový automat je sedmice $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$, kde:

- Q je konečná, neprázdná množina stavů
- Σ je abeceda (konečná, neprázdná množina symbolů)
- Γ je zásobníková abeceda (konečná, neprázdná množina zásobníkových symbolů)
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$ (v horním indexu je $Q \times \Gamma^*$) je přechodová funkce
- $s_0 \in Q$ je počáteční stav
- $Z_0 \in \Gamma$ je startující symbol zásobníku
- $F \subseteq Q$ je množina koncových stavů

Jedinou změnou oproti klasickému nedeterministickému zásobníkovému automatu je definice přechodové funkce, v níž povolujeme odstranění libovolného počtu zásobníkových symbolů z vrcholu zásobníku při provedení přechodu. Je tedy rovněž možné při přechodu neodstranit ze zásobníku žádný symbol.

Modely nedeterministického zásobníkového automatu a rozšířeného zásobníkového automatu jsou **ekvivalentní**, tedy jsou na sebe vzájemně převoditelné. Možnost odstraňovat ze zásobníku více symbolů nepřidá modelu výpočetní sílu, ale umožňuje kompaktnější a přehlednější vyjádření.

Uvažujme o bezkontextovém jazyku $L = \{a^{3n}b^{2m}c^md^n \mid m, n \in \mathbb{N}\}$. Ukažme, jak by bylo možné jej modelovat pomocí zásobníkového automatu a pomocí rozšířeného zásobníkového automatu. Na prvním obrázku je uveden obyčejný nedeterministický zásobníkový automat, na druhém ekvivalentní rozšířený zásobníkový automat.



Všimněme si následujícího:

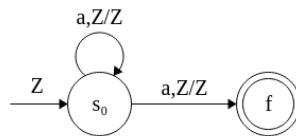
- **Ušetřili jsme stavový prostor**, neboť pro odstranění vyššího množství symbolů ze zásobníku nejsou potřeba žádné speciální stavy
- **Použili jsme méně přechodů**, neboť nemusíme uvažovat o všech možných symbolech, které mohou v daném okamžiku být na vrcholu zásobníku, stačí odebrat ϵ
- **Zvýšili jsme čitelnost**

At' $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ je zásobníkový automat. A je **deterministický zásobníkový automat**, pokud $\forall q \in Q : \forall z \in \Gamma$ platí alespoň jedna z následujících podmínek:

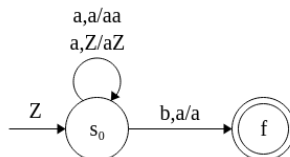
- $(\forall a \in \Sigma : |\delta(q, a, z)| \leq 1) \wedge (|\delta(q, \epsilon, z)| = 0)$
- $(\forall a \in \Sigma : |\delta(q, a, z)| = 0) \wedge (|\delta(q, \epsilon, z)| \leq 1)$

Neformálně řečeno, pokud je zásobníkový automat deterministický, pak platí všechny následující body

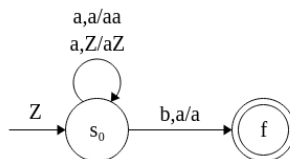
- Ze stavu q **nevede** více než jedna hrana, na níž je čten symbol $a \in \Sigma$ ze vstupu a současně symbol z ze zásobníku
 - V následujícím automatu je tato podmínka porušena



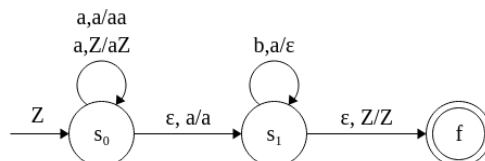
- Ze stavu q **mohou vést** dvě hrany, na nichž je čten stejný symbol $a \in \Sigma$ ze vstupu, pokud je na nich čten jiný symbol ze zásobníku
 - V následujícím automatu je tato vlastnost demonstrována



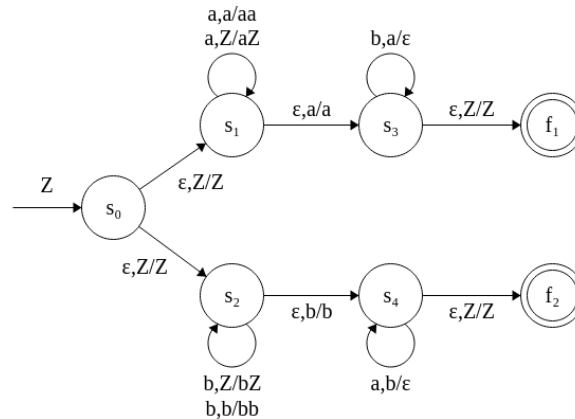
- Ze stavu q **mohou vést** dvě hrany, na nichž je čten stejný symbol ze zásobníku, pokud je na nich čten jiný symbol ze $a \in \Sigma$ ze vstupu
 - V následujícím automatu je tato vlastnost demonstrována



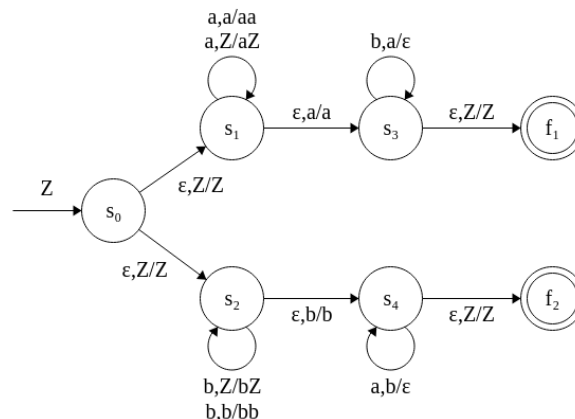
- Ze stavu q **nevede** současně hrana, na níž je ze vstupu čteno ϵ a ze zásobníku je čteno z , a hrana, na níž je rovněž čteno z ze zásobníku, $a \in \Sigma$ ze vstupu
 - V následujícím automatu je tato vlastnost porušena



- Ze stavu q **nevede** současně hrana, na níž je ze vstupu čteno ϵ a ze zásobníku je čteno z , a jiná hrana, na níž je rovněž čteno z ze zásobníku, ϵ ze vstupu
 - V následujícím automatu je tato vlastnost porušena



- Pokud ze stavu q vede hrana, na níž je čteno ϵ ze vstupu a z ze zásobníku, pak z tohoto stavu již **nesmí** vést žádná další hrana, která čte z ze zásobníku
 - V následujícím automatu je tato vlastnost porušena



Pozor. Na rozdíl od konečných automatů, deterministické zásobníkové automaty jsou **slabší** vyjadřovací síly než nedeterministické zásobníkové automaty. Každý deterministický zásobníkový automat je triviálně nedeterministický, ale existují zásobníkové automaty, které není možné determinizovat. Existují bezkontextové jazyky, které není možné přijmout deterministickým zásobníkovým automatem.

Jazyky přijímané zásobníkovými automaty

Rozlišujeme dvě třídy jazyků, které přijímají různé varianty zásobníkových automatů.

- Třída $\mathcal{L}_2$ (třída bezkontextových jazyků), přijímána:
 - nedeterministickými zásobníkovými automaty akceptujícími
 - dosažením koncového stavu
 - vyprázdněním zásobníku
 - současným dosažením koncového stavu a vyprázdněním zásobníku
 - rozšířenými zásobníkovými automaty
- Třída $\mathcal{L}_{DZA}$ (třída deterministických bezkontextových jazyků), přijímána:

- deterministickými zásobníkovými automaty

Platí

$$\mathcal{L}_3 \subset \mathcal{L}_{\text{DZA}} \subset \mathcal{L}_2$$

Příklady deterministických bezkontextových jazyků nad abecedou $\Sigma = \{a, b\}$:

- **libovolný regulární jazyk**
- $\{a^n b^n \mid n \geq 0\}$
- $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$

Příklady bezkontextových jazyků nad abecedou $\Sigma = \{a, b\}$, které nejsou deterministické:

- $\{ww^R \mid w \in \{a, b\}^*\}$
- $\text{co-}\{a^n b^n a^n \mid n \geq 0\}$
- $\text{co-}\{ww \mid w \in \{a, b\}^*\}$

Převod bezkontextových gramatik na zásobníkové automaty

Převod bezkontextových gramatik na zásobníkové automaty realizujeme pomocí **analýzy shora dolů** a pomocí **analýzy zdola nahoru**.

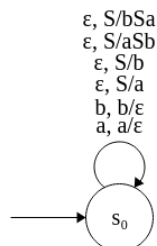
Analýza shora dolů. Ať $G = (N, \Sigma, P, S)$ je bezkontextová gramatika. Pomocí algoritmu analýzy shora dolů vytvoříme ekvivalentní zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ s jedním stavem, který bude přijímat vyprázdněním zásobníku.

- $Q = \{s_0\}$
- $\Gamma = \Sigma \cup N$
- δ :
 - $a \in \Sigma \Rightarrow \delta(s_0, a, a) = \{(s_0, \varepsilon)\}$
 - $(A, \alpha) \in P \Rightarrow (s_0, \alpha) \in \delta(s_0, \varepsilon, A)$
 - žádné jiné přechody neexistují
- $Z_0 = S$
- $F = \emptyset$

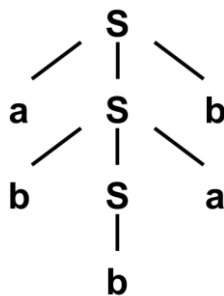
Na začátku máme na zásobníku startující neterminál bezkontextové gramatiky. Pokud je na vrcholu zásobníku neterminál A , je možné ho odebrat a nahradit pravou stranou nějakého pravidla, jehož levá strana je právě A . Pokud je na vrcholu zásobníku terminál, je možné ho odstranit, pokud současně přečteme tentýž terminál ze vstupu. Tento postup odpovídá posloupnosti nejlevějších derivací v dané bezkontextové gramatice. Jakmile vyprázdníme zásobník, analyzovali jsme celé slovo, které je přijato.

Příklad:

- $G = (\{S\}, \{a, b\}, \{(S, aSb), (S, bSa), (S, a), (S, b)\}, S)$
- $A = (\{s_0\}, \{a, b\}, \{a, b, S\}, \delta, s_0, S, \emptyset)$
 - $\delta(s_0, a, a) = \{(s_0, \epsilon)\}$
 - $\delta(s_0, b, b) = \{(s_0, \epsilon)\}$
 - $\delta(s_0, \epsilon, S) = \{(s_0, aSb), (s_0, bSa), (s_0, a), (s_0, b)\}$



- generování slova $w = abbab$ gramatikou G
 - $S \Rightarrow aSb \Rightarrow abSab \Rightarrow abbab$
 - derivační strom je zobrazený níže



- analýza slova $w = abbab$ automatem A
 - $(s_0, abbab, S) \vdash (s_0, abbab, aSb) \vdash (s_0, bbab, Sb) \vdash (s_0, bbab, bSab) \vdash (s_0, bab, Sab) \vdash (s_0, bab, bab) \vdash (s_0, ab, ab) \vdash (s_0, b, b) \vdash (s_0, \epsilon, \epsilon)$
 - průběh analýzy je možné snadno vyčíst z derivačního stromu
 - pokud je na vrcholu zásobníku neterminál, použijeme příslušné pravidlo dle derivačního stromu
 - pokud je na vrcholu zásobníku terminál, přečteme odpovídající symbol ze vstupu

Analýza zdola nahoru. Ať $G = (N, \Sigma, P, S)$, kde $\# \notin N, \Sigma$, je bezkontextová gramatika. Pomocí algoritmu analýzy zdola nahoru vytvoříme ekvivalentní rozšířený zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ s dvěma stavy, který bude přijímat dosažením koncového stavu. **V kontextu tohoto algoritmu budeme výjimečně předpokládat, že vrchol zásobníku je vpravo.**

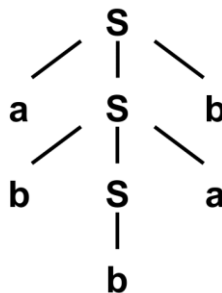
- $Q = \{s_0, r\}$
- $\Gamma = \Sigma \cup N \cup \{\#\}$
- δ :
 - **posunovací pravidla:** $a \in \Sigma \Rightarrow \delta(s_0, a, \epsilon) = \{(s_0, a)\}$

- **redukční pravidla:** $(A, \alpha) \in P \Rightarrow (s_0, A) \in \delta(s_0, \epsilon, \alpha)$
- **ukončující pravidlo:** $\delta(s_0, \epsilon, \#S) = \{(r, \epsilon)\}$
- žádné jiné přechody neexistují
- $Z_0 = \#$
- $F = \{r\}$

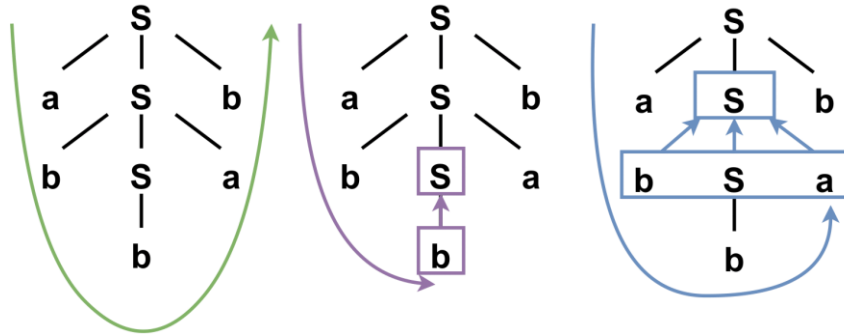
Na začátku máme na zásobníku speciální symbol $\#$. Pokud je na vrcholu zásobníku pravá strana nějakého pravidla (A, α) , je možné ji celou odebrat (používáme rozšířený zásobníkový automat) a nahradit za levou stranu tohoto pravidla. Toto je takzvané **redukční pravidlo**. Je rovněž možné symbol ze vstupu umístit na zásobník, aniž bychom z něj cokoli odstranili. Jde o takzvané **posunovací pravidlo**. Pokud se na vrcholu zásobníku objeví $\#S$, je možné tento řetězec odstranit a přejít do koncového stavu. Toto je **ukončující pravidlo**. Pokud se řetězec $\#S$ objeví na vrcholu zásobníku, jistě jde o obsah celého zásobníku, neboť $\#$ je startující symbol zásobníku, který už nikdy na zásobník neumístíme. Současně je zřejmé, že jsme celý vstup redukovali na pouhý startující neterminál.

Příklad:

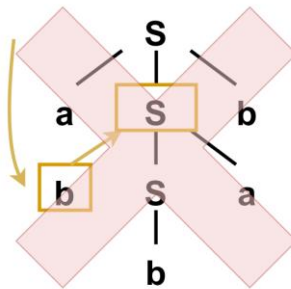
- $G = (\{S\}, \{a, b\}, \{(S, aSb), (S, bSa), (S, a), (S, b)\}, S)$
- $A = (\{s_0, r\}, \{a, b\}, \{a, b, S, \#\}, \delta, s_0, \#, \{r\})$
 - $\delta(s_0, a, \epsilon) = \{(s_0, a)\}$
 - $\delta(s_0, b, \epsilon) = \{(s_0, b)\}$
 - $\delta(s_0, \epsilon, a) = \{(s_0, S)\}$
 - $\delta(s_0, \epsilon, b) = \{(s_0, S)\}$
 - $\delta(s_0, \epsilon, aSb) = \{(s_0, S)\}$
 - $\delta(s_0, \epsilon, bSa) = \{(s_0, S)\}$
 - $\delta(s_0, \epsilon, \#S) = \{(r, \epsilon)\}$
- generování slova $w = abbab$ gramatikou G
 - $S \Rightarrow aSb \Rightarrow abSab \Rightarrow abbab$
 - derivační strom je zobrazený níže



- analýza slova $w = abbab$ automatem A
 - $(s_0, abbab, \#) \vdash (s_0, bbab, \#a) \vdash (s_0, bab, \#ab) \vdash (s_0, ab, \#abb) \vdash (s_0, ab, \#abS)$
 $\vdash (s_0, b, \#abSa) \vdash (s_0, b, \#aS) \vdash (s_0, \epsilon, \#aSb) \vdash (s_0, \epsilon, \#S) \vdash (r, \epsilon, \epsilon)$
 - průběh analýzy je snadné vyčíst z derivačního stromu



- symboly přidáváme na zásobník zleva doprava dle derivačního stromu, jak je naznačeno na **prvním obrázku**
- jakmile na zásobník umístíme symbol vstupu takovým způsobem, že pomocí něj a dříve přidanych symbolů jde dle derivačního stromu provést redukce, tedy **nahrazení celé části stromové úrovně se společným rodičem za neterminál**, provedeme ji, jak je naznačeno na **druhém obrázku** a na **třetím obrázku**
- tzn. neprovádíme redukci například ihned po umístění prvního symbolu *b* na zásobník, neboť symboly *S* a *a* na stejné úrovni derivačního stromu se stejným rodičem ještě na zásobníku nejsou (nemůžeme tedy správně aplikovat pravidlo).



Pokud v textu najdete chybu, nebudete něčemu rozumět nebo budete mít dojem, že by bylo vhodné něco doplnit, kontaktujte na discordu uživatele kocotom.