Automata Theory :: Pushdown Automata

Jörg Endrullis

Vrije Universiteit Amsterdam

Pushdown Automata

Goal

A class of automata that accepts the context-free languages.

Nondeterministic finite automata (NFA's):

- no memory except for the current state
- has only finitely many states

We need some form of infinite memory to accept languages like

$$\{\, a^n b^n \mid n \geq 0 \,\}$$

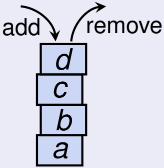
Pushdown automataon

A pushdown automaton has a **stack** of unlimited size.

Pushdown Automata

Next to the input alphabet Σ, there is now a **stack alphabet** Γ.

A **stack** is a finite sequence of elements from Γ:



We write **stacks as words**

dcba

with the **top-element on the left.**

Elements added or removed only on the top of the stack.

A transition reads the topmost element of the stack

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Q \times \Gamma^*}$$

and exchanges it with zero or more new elements.

The nondeterministic choice $\delta(q, \alpha, b)$ must always be finite!

Pushdown Automata

A **nondeterministic pushdown automaton** (**NPDA**) is a tuple

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

- Q is a finite set of states
- Σ is a finite input alphabet
- Γ is a finite stack alphabet
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Q \times \Gamma^*}$

  the transition function, where $\delta(q, \alpha, b)$ is always finite
- $q_0 \in Q$ the starting state
- $z \in \Gamma$ the stack starting symbol
- $F \subseteq Q$ a set of final states

Initially, the stack content is z.

If $(q', v) \in \delta(q, \alpha, b)$, this means that

- from state q with input αw and stack bu

the automaton can do a transition to

- state q' with input w and stack vu.

Language Accepted by a Pushdown Automaton

A **configuration** (q, w, u) of an NPDA consists of:

- current state $q \in Q$
- input word $w \in \Sigma^*$
- current stack $u \in \Gamma^*$

The **step relation** on configurations is defined by

$$(q,\ \alpha w,\ bu) \vdash (q',\ w,\ vu)$$

whenever $(q', v) \in \delta(q, \alpha, b)$.

We write $\vdash^*$ for computation (zero or more steps).

The **language generated by** NPDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ is

$$L(M) = \{\, w \in \Sigma^* \mid (q_0, w, z) \vdash^* (q', \lambda, u) \text{ where } q' \in F \,\}.$$

Note: no condition on the stack u at the end.

Drawing Pushdown Automata

The transition graph for a NPDA contains

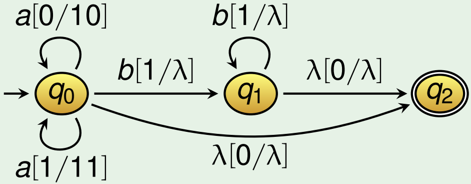
for every $(q', v) \in \delta(q, \alpha, b)$ an arrow $q \xrightarrow{\alpha[b/v]} q'$

We construct NPDA M with $L(M) = \{ a^n b^n \mid n \geq 0 \}$.

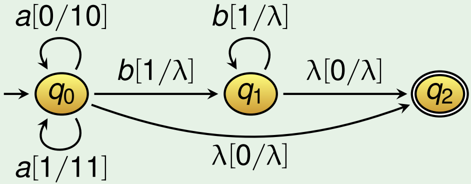
$Q = \{q_0, q_1, q_2\}$ $\Sigma = \{a, b\}$ $\Gamma = \{0, 1\}$ $z = 0$ $F = \{q_2\}$

Intuition:

- In q_0 a stack $1^k 0$ means: we have read k a's.
- In q_1 a stack $1^k 0$ means: we still have to read k b's.



Example Computation

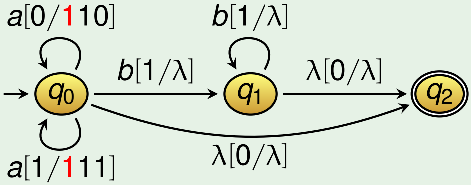


Stepwise reading of the word *aabb*:

$$
\begin{aligned}
(q_0, aabb, 0) &\vdash (q_0, abb, 10) \\
&\vdash (q_0, bb, 110) \\
&\vdash (q_1, b, 10) \\
&\vdash (q_1, \lambda, 0) \\
&\vdash (q_2, \lambda, \lambda)
\end{aligned}
$$

Exercises (1)

Draw an NPDA M with $L(M) = \{\, a^n b^{2n} \mid n \geq 0 \,\}$.



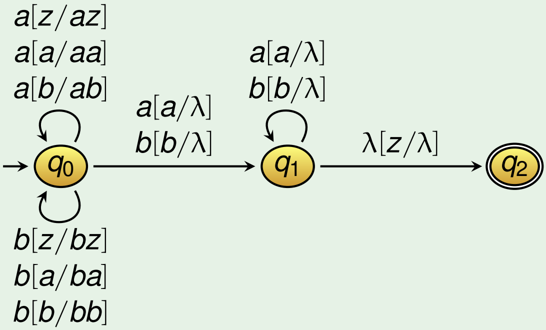
Exercises (2)

Draw an NPDA M with $L(M) = \{ ww^R \mid w \in \{a, b\}^+ \}$.

Hint: define

$$Q = \{ q_0, q_1, q_2 \} \qquad \Gamma = \{ a, b, z \}$$
$$\Sigma = \{ a, b \} \qquad F = \{ q_2 \}$$

We construct the NDPA as follows:



Exercises (3)

Is there an NPDA M with $L(M) = \{\, ww \mid w \in \{a, b\}^+ \,\}$?

No!

This language is not context-free.

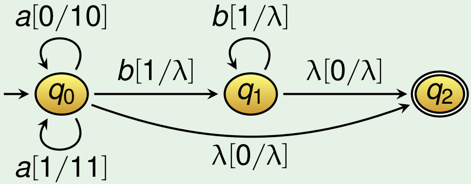
Acceptance with Empty Stack

Acceptance with Empty Stack

All automata we have seen so far had the following property:

$$(q_0, w, z) \vdash^* (q', w', u') \quad \Longrightarrow \quad (q' \in F \iff u' = \lambda)$$

They reach an accepting state if and only if the stack is empty.



Acceptance with Empty Stack

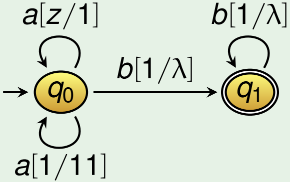
Empty stack language of NPDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ is

$$L_\lambda(M) = \{\, w \in \Sigma^* \mid (q_0, w, z) \vdash^* (q', \lambda, \lambda) \,\}.$$

(No need for final states in this definition.)

Example

Consider the following NPDA M:



What is the language accepted by this automaton?

$$L(M) = \{\, a^n b^m \mid n \geq m \geq 1 \,\}$$

The empty stack language of M is

$$L_\lambda(M) = \{\, a^n b^n \mid n \geq 1 \,\}$$

For a language L the following two are equivalent:

- There is an NDPA M with $L(M) = L$.
- There is an NDPA M with $L_\lambda(M) = L$.

From Final States to Acceptance with Empty Stack

From Final States to Acceptance with Empty Stack

Every NPDA M be transformed into NPDA N such that

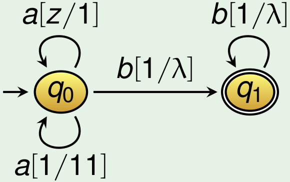
- it has **a single final state** $F = \{ q_f \}$,
- **final state is reached if and only if the stack is empty**,
- $L(M) = L(N) = L_\lambda(N)$.

We add fresh states $\{ \widehat{q_0}, q_e, q_f \}$ to Q and stack element $\hat{z}$ to Γ.

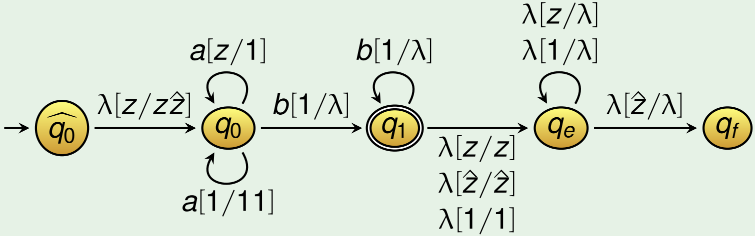
- Add a transition $\widehat{q_0} \xrightarrow{\lambda[z/z\hat{z}]} q_0$.
  (Intuition: $\hat{z}$ marks the bottom of the stack.)
- Add transitions $q \xrightarrow{\lambda[s/s]} q_e$ for every $q \in F$, $s \in \Gamma$.
- Add transitions $q_e \xrightarrow{\lambda[s/\lambda]} q_e$ for every $s \in \Gamma \setminus \{\hat{z}\}$.
  (Intuition: q_e empties the stack.)
- Add transition $q_e \xrightarrow{\lambda[\hat{z}/\lambda]} q_f$.
  (Intuition: switch to final state q_f when stack is empty.)
- Define $\widehat{q_0}$ as starting state and $F = \{ q_f \}$.

From Final States to Acceptance with Empty Stack

Consider the NPDA M:

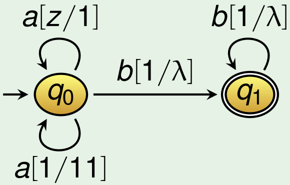


Transform it into NPDA N such that $L(M) = L(N) = L_\lambda(N)$:

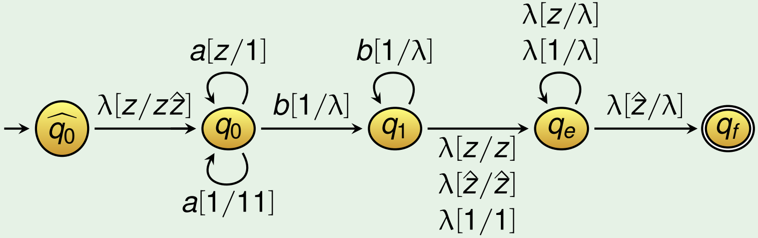


From Final States to Acceptance with Empty Stack

Consider the NPDA M:



Transform it into NPDA N such that $L(M) = L(N) = L_\lambda(N)$:



This NPDA N reaches the final state $\iff$ the stack is empty.

From Acceptance with Empty Stack To Final States

From Acceptance with Empty Stack To Final States

Every NPDA M be transformed into NPDA N such that

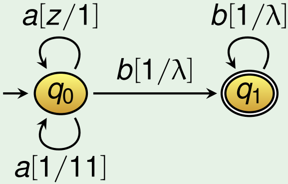
- it has **a single final state** $F = \{ q_f \}$,
- **final state is reached if and only if the stack is empty**,
- $L_\lambda(M) = L(N) = L_\lambda(N)$.

We add fresh states $\{ \widehat{q_0}, q_f \}$ to Q and stack element $\hat{z}$ to Γ.

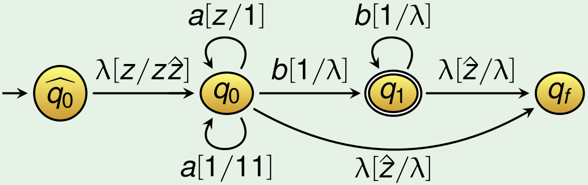
- Add a transition $\widehat{q_0} \xrightarrow{\lambda[z/z\hat{z}]} q_0$.
  (Intuition: $\hat{z}$ marks the bottom of the stack.)
- Add transition $q \xrightarrow{\lambda[\hat{z}/\lambda]} q_f$ for every state $q \in Q \setminus \{ \widehat{q_0}, q_f \}$.
  (Intuition: switch to final state q_f when stack is empty.)
- Define $\widehat{q_0}$ as starting state and $F = \{ q_f \}$.

From Acceptance with Empty Stack To Final States

Consider the NPDA M:

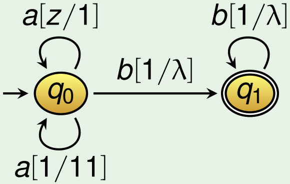


Transform it into NPDA N such that $L_{\lambda}(M) = L(N) = L_{\lambda}(N)$:

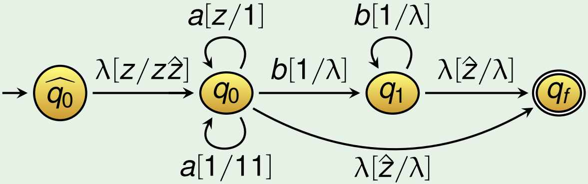


From Acceptance with Empty Stack To Final States

Consider the NPDA M:



Transform it into NPDA N such that $L_{\lambda}(M) = L(N) = L_{\lambda}(N)$:



This NPDA N reaches the final state $\iff$ the stack is empty.

Pushdown Automata & Context-Free Languages

Context-Free Languages and NPDA's

Theorem

A language L is context-free

$\iff$ there exists an NPDA M with $L(M) = L$.

Proof.

We need to prove two directions:

- $(\Rightarrow)$ Translate context-free grammars into NPDA's.
- $(\Leftarrow)$ Translate NPDA's into context-free grammars.

$\square$

From Context-Free Grammars to NPDA’s

From Context-Free Grammars to NPDA's

Construction

Let $G = (V, T, S, P)$ be a context-free grammar.

Idea: simulate leftmost derivation on the stack

We construct an NPDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ as follows:

$$Q = \{ q_0, q_1, q_2 \} \qquad \Sigma = T$$
$$F = \{ q_2 \} \qquad \Gamma = V \cup T \cup \{ z \}$$

We add transitions simulating a leftmost derivation:

$$\begin{array}{ll} q_0 \xrightarrow{\lambda[z/Sz]} q_1 & \\ q_1 \xrightarrow{\lambda[A/x]} q_1 & \text{for every } A \to x \in P \\ q_1 \xrightarrow{a[a/\lambda]} q_1 & \text{for every } a \in T \\ q_1 \xrightarrow{\lambda[z/\lambda]} q_2 & \end{array}$$

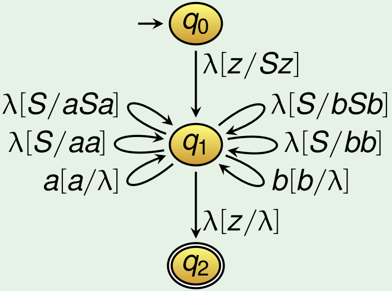
Then $L(M) = L(G)$.

Example

The language $\{ ww^R \mid w \in \{a, b\}^+ \}$ is generated by the grammar

$$S \rightarrow aSa \mid bSb \mid aa \mid bb$$

Translating this grammar into an NPDA yields:



$$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz) \vdash (q_1, bba, Saz)$$
$$\vdash (q_1, bba, bbaz) \vdash (q_1, ba, baz) \vdash (q_1, a, az)$$
$$\vdash (q_1, \lambda, z) \vdash (q_2, \lambda, \lambda)$$

From NPDA’s to Context-Free Grammars

From NPDA's to Context-Free Grammars

Construction

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ be an NPDA. Transform M s.t.

Assumption: $F = \{ q_f \}$ and q_f reachable only with empty stack.

We define a context-free grammar (V, T, S, P) as follows:

$$T = \Sigma \qquad V = \{ (q\, b\, q') \mid q, q' \in Q, b \in \Gamma \} \qquad S = (q_0\, z\, q_f)$$

Intuition: $(q\, b\, q') \Rightarrow^+ w \iff (q, w, b) \vdash^+ (q', \lambda, \lambda)$.

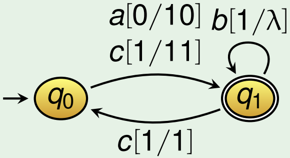
The set P contains the following rules:

- If $q \xrightarrow{\alpha[b/\lambda]} q'$, then
  $(q\, b\, q') \to \alpha$ in P.
- If $q \xrightarrow{\alpha[b/c_1 \cdots c_n]} q'$ with $n \geq 1$, then
  $(q\, b\, r_n) \to \alpha\, (q'\, c_1\, r_1)\, (r_1\, c_2\, r_2) \cdots (r_{n-1}\, c_n\, r_n)$ in P
  for **all** $r_1, \ldots, r_n \in Q$

Then we have $L(G) = L(M)$.

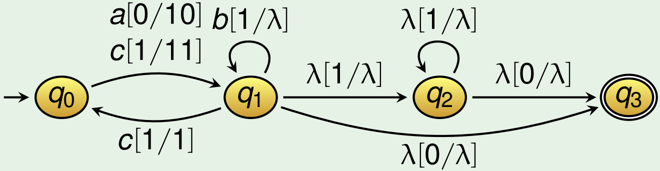
Example

Consider the following NPDA with stack starting symbol $z = 0$:

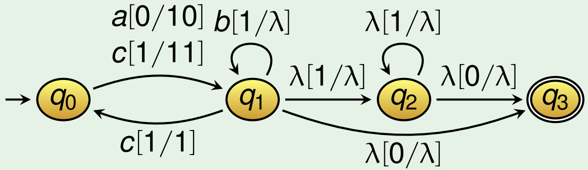


Ensure that the final state is only be reached with empty stack.

We already know how to transform acceptance with final states to acceptance with empty stack. **Here no fresh start state is needed;** the symbol 0 always remains at the bottom.



Example



The resulting context-free grammar is:

$$
\begin{array}{llll}
(q_0\,0\,r_2) \xrightarrow{1} a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2) & & (q_1\,0\,q_3) \xrightarrow{5} \lambda \\
(q_0\,1\,r_2) \xrightarrow{2} c\,(q_1\,1\,r_1)\,(r_1\,1\,r_2) & & (q_1\,1\,q_2) \xrightarrow{6} \lambda \\
(q_1\,1\,r_1) \xrightarrow{3} c\,(q_0\,1\,r_1) & & (q_2\,1\,q_2) \xrightarrow{7} \lambda \\
(q_1\,1\,q_1) \xrightarrow{4} b & & (q_2\,0\,q_3) \xrightarrow{8} \lambda
\end{array}
$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$. Here $S = (q_0\,0\,q_3)$.

$$
\begin{array}{rl}
 & (q_0, accb, \; 0) \\
\vdash & (q_1, ccb, \; 10) \\
\vdash & (q_0, cb, \; 10) \\
\vdash & (q_1, b, \; 110) \\
\vdash & (q_1, \lambda, \; 10) \\
\vdash & (q_2, \lambda, \; 0) \\
\vdash & (q_3, \lambda, \; \lambda)
\end{array}
$$

$$
\begin{aligned}
(q_0\,0\,q_3) &\overset{1}{\Rightarrow} \underline{a\,(q_1\,1\,q_2)\,(q_2\,0\,q_3)} \overset{3}{\Rightarrow} ac\,\underline{(q_0\,1\,q_2)}\,(q_2\,0\,q_3) \\
&\overset{2}{\Rightarrow} acc\,\underline{(q_1\,1\,q_1)\,(q_1\,1\,q_2)}\,(q_2\,0\,q_3) \\
&\overset{4}{\Rightarrow} acc\underline{b}\,(q_1\,1\,q_2)\,(q_2\,0\,q_3) \overset{6}{\Rightarrow} accb\,(q_2\,0\,q_3) \overset{8}{\Rightarrow} accb
\end{aligned}
$$

Deterministic Pushdown Automata

Deterministic Pushdown Automata

A **deterministic pushdown automaton** (**DPDA**) is an NPDA such that

- $\delta(q, \alpha, b)$ contains at most one element
- If $\delta(q, \lambda, b) \neq \varnothing$, then $\delta(q, a, b) = \varnothing$ for every $a \in \Sigma$.

A language L is **deterministic context-free** if there exists a DPDA M with $L(M) = L$.

A deterministic context-free L allows for efficient parsing.

Exercises

Which of these languages are deterministic context-free?

- $\{ a^n b^n \mid n \geq 0 \}$
- $\{ ww^R \mid w \in \{a, b\}^+ \}$
- $\{ wcw^R \mid w \in \{a, b\}^+ \}$

Conclusion

Not all context-free languages are deterministic context-free.

Theorem

It is decidable if two DPDA's generate the same language.

(Géraud Sénizergues, 1997)