Automata & Complexity

Jörg Endrullis

Vrije Universiteit Amsterdam

2018

Looking Back

Previous subjects relevant for this lecture:

- Context-free languages
- Context-free grammars
- Parsing algorithms (CYK and LL)

Pushdown Automata

Goal

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

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, and
- has only finitely many states.

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, and
- 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 Automata

Goal

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

Nondeterministic finite automata (NFA's):

- no memory except for the current state, and
- 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.

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.

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.

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

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.

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

A transition reads the topmost element of the stack

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \ldots$$

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.

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

A transition reads the topmost element of the stack

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \ldots$$

and exchanges it with zero or more new elements:

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

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.

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

A transition reads the topmost element of the stack

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \dots$$

and exchanges it with zero or more new elements:

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

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

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.

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

A transition reads the topmost element of the stack

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \dots$$

and exchanges it with zero or more new elements:

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

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

Initially, stack contains one symbol: **stack start symbol** $z \in \Gamma$.

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

Language Accepted by a Pushdown Automaton

When a NPDA reads a words, we need to keep track of:

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

Language Accepted by a Pushdown Automaton

When a NPDA reads a words, we need to keep track of:

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

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

When a NPDA reads a words, we need to keep track of:

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

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.

This is denoted by $(q,\ \alpha w,\ bu) \vdash (q',\ w,\ vu)$.

Language Accepted by a Pushdown Automaton

When a NPDA reads a words, we need to keep track of:

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

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.

This is denoted by $(q,\ \alpha w,\ bu) \vdash (q',\ w,\ vu)$.

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. It is often empty.

Drawing Pushdown Automata

The transition graph for a NPDA contains

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

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

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

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

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 $L(M) = \{\, a^n b^n \mid n \geq 0 \,\}$.

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

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.

Drawing Pushdown Automata

The transition graph for a NPDA contains

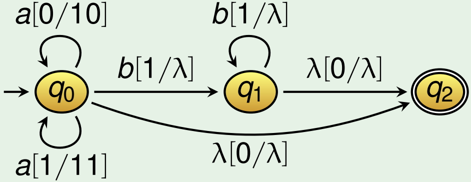
for every $(q', v) \in \delta(q, \alpha, b)$ an arrow $q \xrightarrow{\alpha[b/v]} q'$.

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

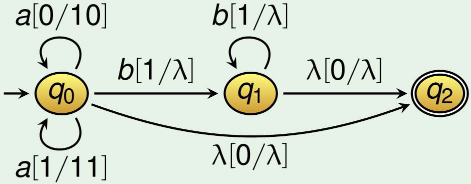
$Q = \{q_0, q_1, q_2\}$ $\Sigma = \{a, b\}$ $\Gamma = \{0, 1\}$ $z = 0$ $F = \{q_2\}$

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.



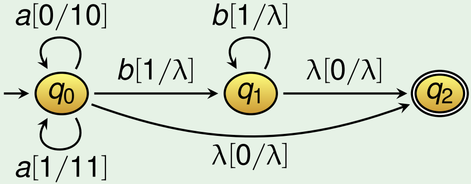
Example Computation



Stepwise reading of the word *aabb*:

$$(q_0, aabb, 0)$$

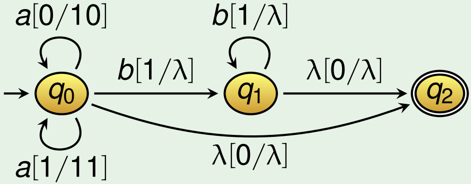
Example Computation



Stepwise reading of the word *aabb*:

$$(q_0, aabb, 0) \vdash (q_0, abb, 10)$$

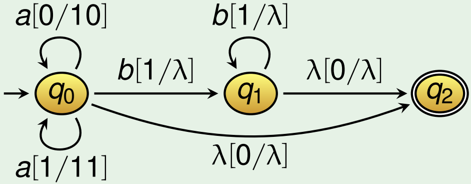
Example Computation



Stepwise reading of the word *aabb*:

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

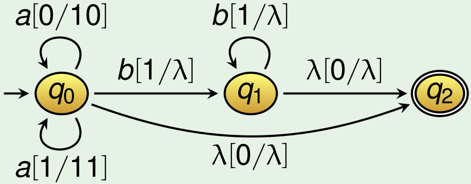
Example Computation



Stepwise reading of the word *aabb*:

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

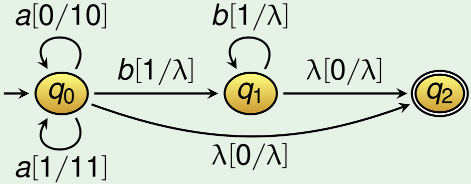
Example Computation



Stepwise reading of the word *aabb*:

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

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

(Groups of two, 2 minutes)

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

Exercises

(Groups of two, 2 minutes)

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

(Groups of two, 2 minutes)

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

Hint: define

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

$$\Sigma = \{a, b\} \qquad F = \{q_2\}$$

Exercises

(Groups of two, 2 minutes)

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

(Groups of two, 2 minutes)

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

Hint: define

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

$$\Sigma = \{a, b\} \qquad F = \{q_2\}$$

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

Deterministic Pushdown Automaton

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

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

Deterministic Pushdown Automaton

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

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

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

Deterministic Pushdown Automaton

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

- $\delta(q, \alpha, b)$ contains at most one element
- If $\delta(q, \lambda, b) \neq \emptyset$, then $\delta(q, a, b) = \emptyset$ 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 language allows for efficient parsing.

Exercises

Which of these languages are deterministic context-free?

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

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.

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)

Context-Free Languages and NPDA's

Theorem

A language L is context-free
$\iff$ there exists an NPDA M with $L(M) = L$.

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

Construction

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

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

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:

$$Q = \{q_0, q_1, q_2\} \qquad \Sigma = T$$

$$F = \{q_2\} \qquad \Gamma = V \cup T \cup \{z\}$$

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:

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

A derivation in G starts with S: $\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$.

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:

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

A derivation in G starts with S: $\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$.

We simulate a leftmost reduction:

$$\delta(q_1, \lambda, A) = \{\, (q_1, x) \mid A \to x \in P \,\} \qquad (A \in V)$$
$$\delta(q_1, a, a) = \{\, (q_1, \lambda) \,\} \qquad (a \in T)$$

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:

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

A derivation in G starts with S: $\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$.

We simulate a leftmost reduction:

$$\delta(q_1, \lambda, A) = \{\, (q_1, x) \mid A \to x \in P \,\} \qquad (A \in V)$$
$$\delta(q_1, a, a) = \{\, (q_1, \lambda) \,\} \qquad (a \in T)$$

We stop when the stack is z: $\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$.

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:

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

A derivation in G starts with S: $\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$.

We simulate a leftmost reduction:

$$\delta(q_1, \lambda, A) = \{\, (q_1, x) \mid A \to x \in P \,\} \qquad (A \in V)$$
$$\delta(q_1, a, a) = \{\, (q_1, \lambda) \,\} \qquad (a \in T)$$

We stop when the stack is z: $\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$.

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

Example

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

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

Example

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

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

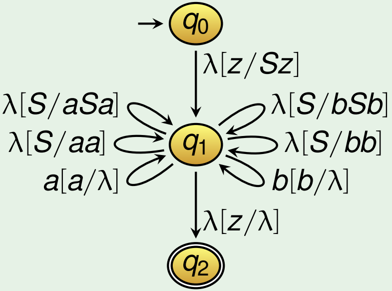
Translating this grammar into an NPDA yields:

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:

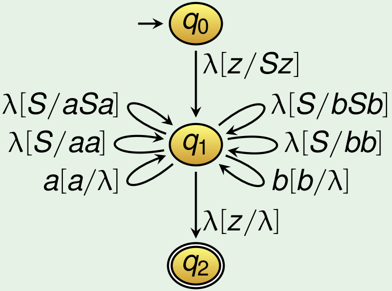


Example

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:



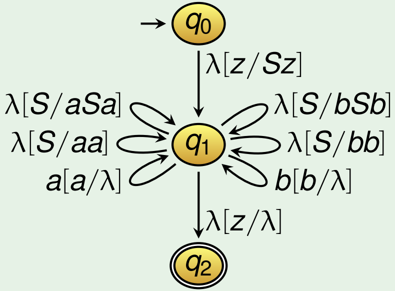
$(q_0, abba, z)$

Example

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

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

Translating this grammar into an NPDA yields:



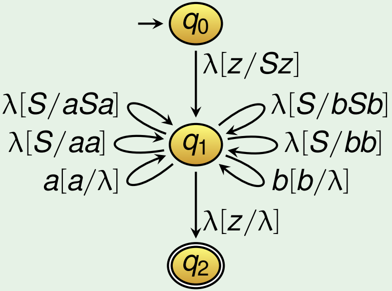
$(q_0, abba, z) \vdash (q_1, abba, Sz)$

Example

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

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

Translating this grammar into an NPDA yields:



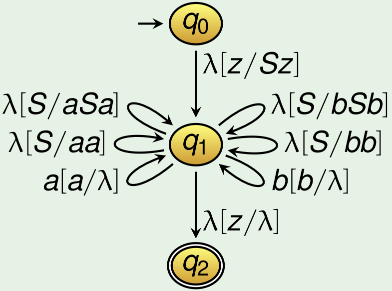
$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz)$

Example

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

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

Translating this grammar into an NPDA yields:



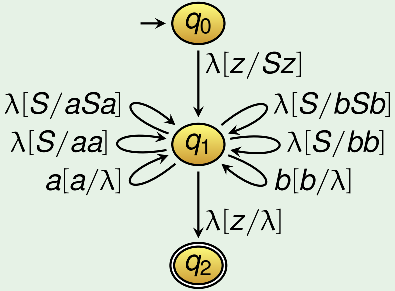
$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz) \vdash (q_1, bba, Saz)$

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:



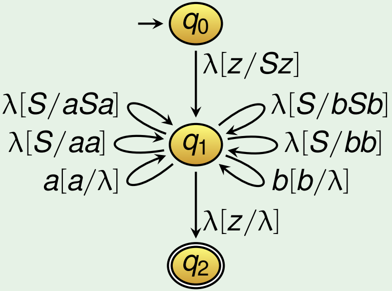
$$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz) \vdash (q_1, bba, Saz)$$
$$\vdash (q_1, bba, bbaz)$$

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:



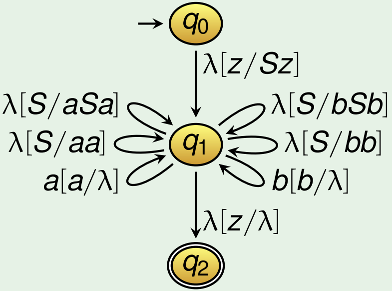
$$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz) \vdash (q_1, bba, Saz)$$
$$\vdash (q_1, bba, bbaz) \vdash (q_1, ba, baz)$$

Example

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

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

Translating this grammar into an NPDA yields:



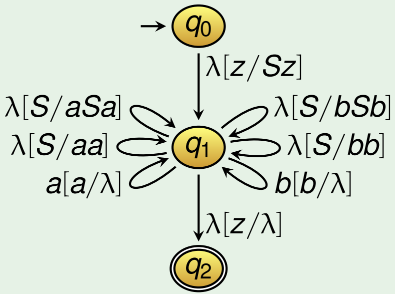
$$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz) \vdash (q_1, bba, Saz)$$
$$\vdash (q_1, bba, bbaz) \vdash (q_1, ba, baz) \vdash (q_1, a, az)$$

Example

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

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

Translating this grammar into an NPDA yields:



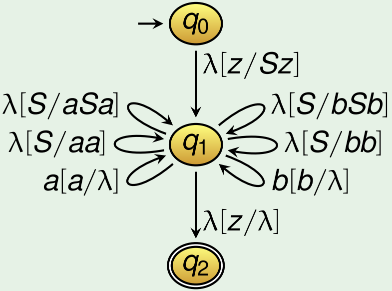
$$(q_0, abba, z) \vdash (q_1, abba, Sz) \vdash (q_1, abba, aSaz) \vdash (q_1, bba, Saz)$$
$$\vdash (q_1, bba, bbaz) \vdash (q_1, ba, baz) \vdash (q_1, a, az)$$
$$\vdash (q_1, \lambda, z)$$

Example

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

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

Translating this grammar into an NPDA yields:



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

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

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

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

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.)

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

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 $q \in F$, $s \in \Gamma$.

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

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.)

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

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.)

Question

How to transform an NPDA into an equivalent NPDA such that

- $F = \{ q_f \}$, and
- q_f can only be reached when the stack is empty.

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 NPDA's to Context-Free Grammars

Construction

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

From NPDA's to Context-Free Grammars

Construction

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

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

From NPDA's to Context-Free Grammars

Construction

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

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

From NPDA's to Context-Free Grammars

Construction

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

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 $(q', \lambda) \in \delta(q, \alpha, b)$, then
  $(q\, b\, q') \to \alpha$ in P.
- If $(q', c_1 \cdots c_n) \in \delta(q, \alpha, b)$ 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** $r_1, \ldots, r_n \in Q$

From NPDA's to Context-Free Grammars

Construction

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

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

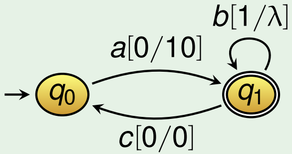
The set P contains the following rules:

- If $(q', \lambda) \in \delta(q, \alpha, b)$, then
  $(q\, b\, q') \to \alpha$ in P.
- If $(q', c_1 \cdots c_n) \in \delta(q, \alpha, b)$ 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 $L(G) = L(M)$.

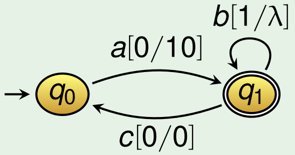
Example

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



Example

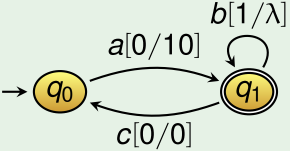
Consider the following NPDA with stack starting symbol $z = 0$:



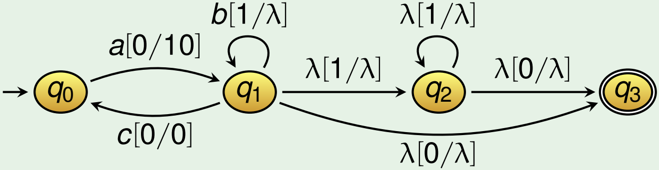
Ensure that the final state is only be reached with empty stack:

Example

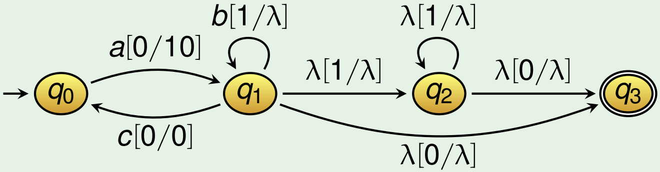
Consider the following NPDA with stack starting symbol $z = 0$:



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



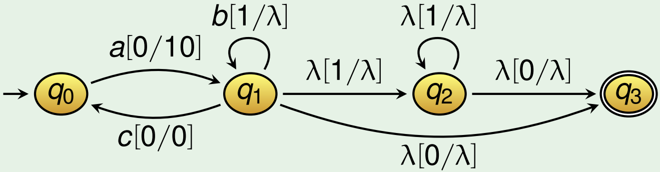
Example



The resulting context-free grammar is:

$(q_0\, 0$

Example

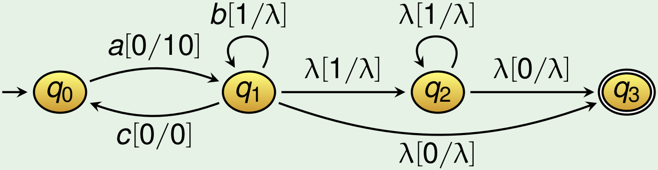


The resulting context-free grammar is:

$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2)$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

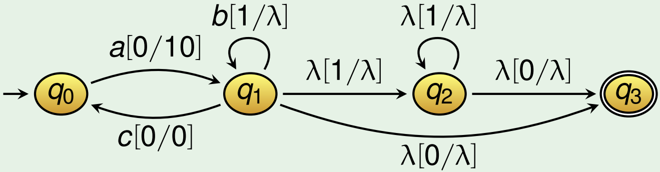


The resulting context-free grammar is:

$$(q_0\,0\,r_2) \longrightarrow a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2)$$
$$(q_1\,0$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

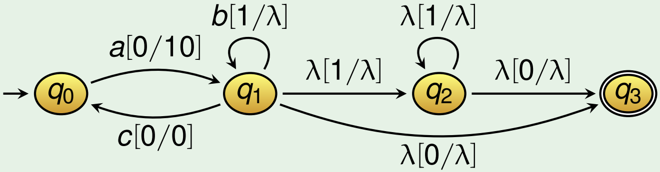


The resulting context-free grammar is:

$$(q_0\, 0\, r_2) \rightarrow a\, (q_1\, 1\, r_1)\, (r_1\, 0\, r_2)$$
$$(q_1\, 0\, r_1) \rightarrow c\, (q_0\, 0\, r_1)$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

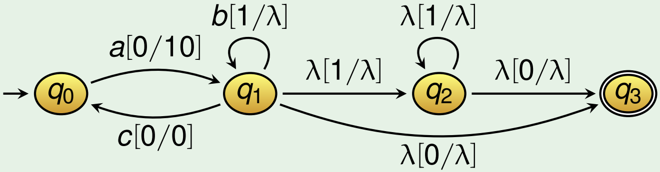


The resulting context-free grammar is:

$$(q_0\, 0\, r_2) \rightarrow a\, (q_1\, 1\, r_1)\, (r_1\, 0\, r_2)$$
$$(q_1\, 0\, r_1) \rightarrow c\, (q_0\, 0\, r_1)$$
$$(q_1\, 1$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

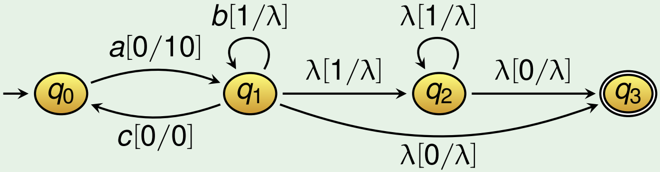


The resulting context-free grammar is:

$$(q_0\, 0\, r_2) \to a\, (q_1\, 1\, r_1)\, (r_1\, 0\, r_2)$$
$$(q_1\, 0\, r_1) \to c\, (q_0\, 0\, r_1)$$
$$(q_1\, 1\, q_1) \to b$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example



The resulting context-free grammar is:

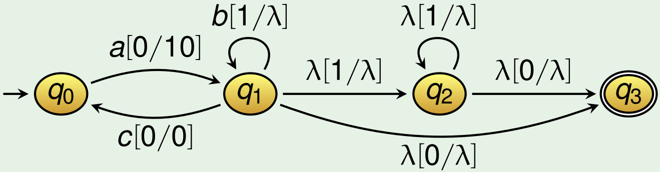
$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2) \qquad (q_1\,0$$

$$(q_1\,0\,r_1) \to c\,(q_0\,0\,r_1)$$

$$(q_1\,1\,q_1) \to b$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

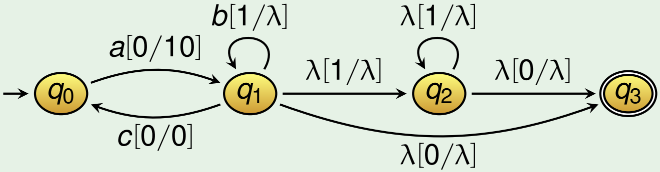


The resulting context-free grammar is:

$$(q_0\, 0\, r_2) \to a\, (q_1\, 1\, r_1)\, (r_1\, 0\, r_2) \qquad (q_1\, 0\, q_3) \to \lambda$$
$$(q_1\, 0\, r_1) \to c\, (q_0\, 0\, r_1)$$
$$(q_1\, 1\, q_1) \to b$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example



The resulting context-free grammar is:

$(q_0\, 0\, r_2) \rightarrow a\,(q_1\, 1\, r_1)\,(r_1\, 0\, r_2)$

$(q_1\, 0\, r_1) \rightarrow c\,(q_0\, 0\, r_1)$

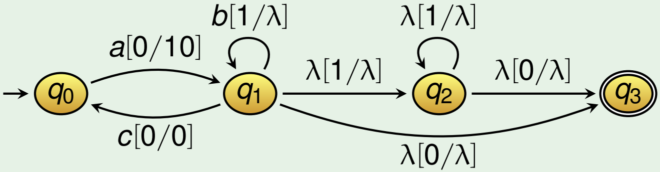
$(q_1\, 1\, q_1) \rightarrow b$

$(q_1\, 0\, q_3) \rightarrow \lambda$

$(q_1\, 1$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

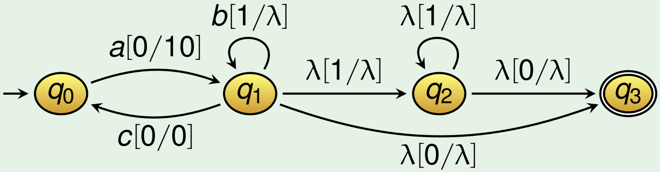


The resulting context-free grammar is:

$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2) \qquad (q_1\,0\,q_3) \to \lambda$$
$$(q_1\,0\,r_1) \to c\,(q_0\,0\,r_1) \qquad (q_1\,1\,q_2) \to \lambda$$
$$(q_1\,1\,q_1) \to b$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example



The resulting context-free grammar is:

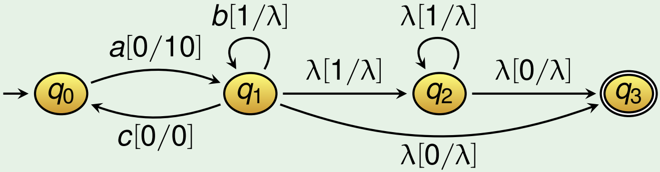
$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2) \qquad (q_1\,0\,q_3) \to \lambda$$

$$(q_1\,0\,r_1) \to c\,(q_0\,0\,r_1) \qquad (q_1\,1\,q_2) \to \lambda$$

$$(q_1\,1\,q_1) \to b \qquad (q_2\,1$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example

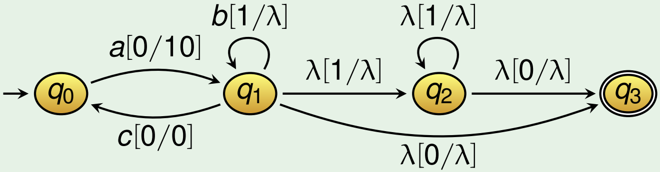


The resulting context-free grammar is:

$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2) \qquad (q_1\,0\,q_3) \to \lambda$$
$$(q_1\,0\,r_1) \to c\,(q_0\,0\,r_1) \qquad (q_1\,1\,q_2) \to \lambda$$
$$(q_1\,1\,q_1) \to b \qquad (q_2\,1\,q_2) \to \lambda$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example



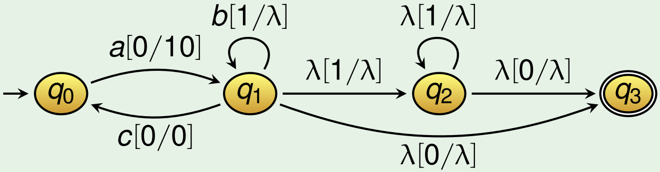
The resulting context-free grammar is:

$$(q_0\,0\,r_2) \rightarrow a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2)$$
$$(q_1\,0\,r_1) \rightarrow c\,(q_0\,0\,r_1)$$
$$(q_1\,1\,q_1) \rightarrow b$$

$$(q_1\,0\,q_3) \rightarrow \lambda$$
$$(q_1\,1\,q_2) \rightarrow \lambda$$
$$(q_2\,1\,q_2) \rightarrow \lambda$$
$$(q_2\,0$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example



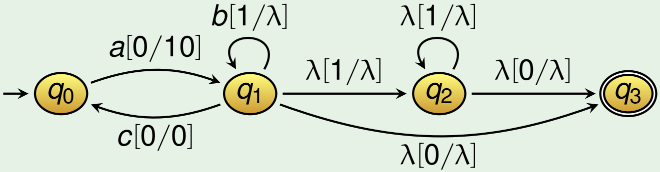
The resulting context-free grammar is:

$$(q_0\, 0\, r_2) \to a\, (q_1\, 1\, r_1)\, (r_1\, 0\, r_2)$$
$$(q_1\, 0\, r_1) \to c\, (q_0\, 0\, r_1)$$
$$(q_1\, 1\, q_1) \to b$$

$$(q_1\, 0\, q_3) \to \lambda$$
$$(q_1\, 1\, q_2) \to \lambda$$
$$(q_2\, 1\, q_2) \to \lambda$$
$$(q_2\, 0\, q_3) \to \lambda$$

for all $r_1, r_2 \in \{q_0, q_1, q_2, q_3\}$.

Example



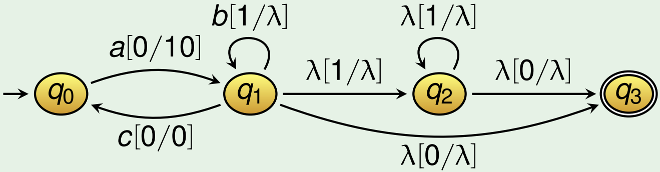
The resulting context-free grammar is:

$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2)$$
$$(q_1\,0\,r_1) \to c\,(q_0\,0\,r_1)$$
$$(q_1\,1\,q_1) \to b$$

$$(q_1\,0\,q_3) \to \lambda$$
$$(q_1\,1\,q_2) \to \lambda$$
$$(q_2\,1\,q_2) \to \lambda$$
$$(q_2\,0\,q_3) \to \lambda$$

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

Example



The resulting context-free grammar is:

$$(q_0\,0\,r_2) \to a\,(q_1\,1\,r_1)\,(r_1\,0\,r_2) \qquad (q_1\,0\,q_3) \to \lambda$$
$$(q_1\,0\,r_1) \to c\,(q_0\,0\,r_1) \qquad (q_1\,1\,q_2) \to \lambda$$
$$(q_1\,1\,q_1) \to b \qquad (q_2\,1\,q_2) \to \lambda$$
$$(q_2\,0\,q_3) \to \lambda$$

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

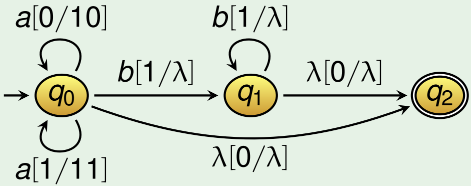
$$(q_0\,0\,q_3) \Rightarrow a\,(q_1\,1\,q_2)\,(q_2\,0\,q_3) \Rightarrow a\,(q_2\,0\,q_3) \Rightarrow a$$

$$(q_0\,0\,q_3) \Rightarrow a\,(q_1\,1\,q_1)\,(q_1\,0\,q_3) \Rightarrow ab\,(q_1\,0\,q_3)$$
$$\Rightarrow abc\,(q_0\,0\,q_3) \Rightarrow^+ abca$$

Exercise

(Groups of two, 2 minutes)

Construct a context-free grammar for the following NPDA.
The stack starting symbol is $z = 0$.



Use the grammar to show that *aabb* is in the language.

Looking Back

- Pushdown automata (with stack)
- Deterministic context-free languages
- Equivalence of NPDA's and context-free grammars

Looking Forward

Read:

- Linz 7.1–7.3
- Lewis & Papadimitriou 3.4 (optional)

Do the following exercises:

- Linz 7.1: 3a, 4c,g, 5, 11
- Linz 7.2: 5, 6, 15
- Linz 7.3: 3, 5, 8, 16, 18

Following lecture:

- Properties of context-free languages
- Turing machines