Computability and Complexity
COSC 4200

Equivalence of Context-Free Grammars and Pushdown Automata

Theorem

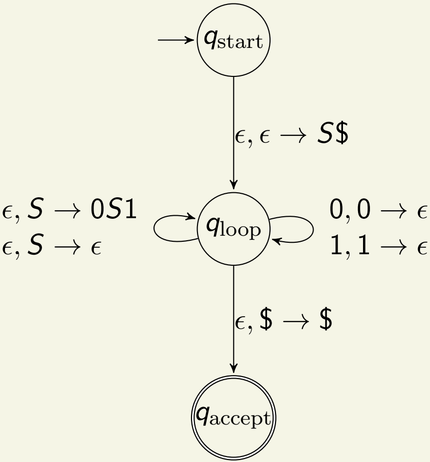
A language is context-free if and only if it is accepted by some pushdown automaton.

There are two directions:

1. Show that for every CFG G, there is an equivalent PDA M with $L(M) = L(G)$.
2. Show that for every PDA M, there is an equivalent CFG G with $L(G) = L(M)$.

Converting a CFG into a PDA

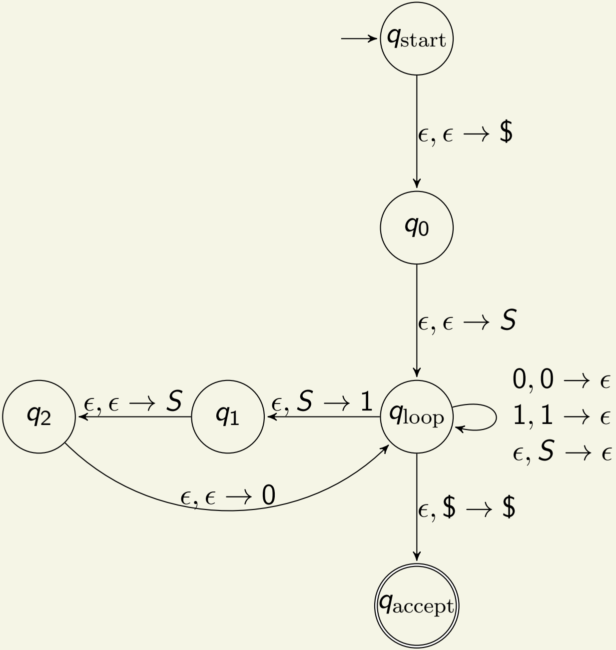
Consider the grammar $S \to 0S1 \mid \epsilon$.
The idea is to build the following PDA:



On q_{loop}, we have

- $a, a \to \epsilon$ for each $a \in \Sigma$
- $\epsilon, A \to \omega$ for each rule $A \to \omega$.

The actual PDA uses more states:



Example

Derivation in CFG:

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0011$$

Computation in PDA:

$$\begin{aligned}
(q_{\text{start}}, 0011, \epsilon) &\stackrel{(2)}{\to} (q_{\text{loop}}, 0011, S\$) \\
&\stackrel{(3)}{\to} (q_{\text{loop}}, 0011, 0S1\$) \\
&\to (q_{\text{loop}}, 011, S1\$) \\
&\stackrel{(3)}{\to} (q_{\text{loop}}, 011, 0S11\$) \\
&\to (q_{\text{loop}}, 11, S11\$) \\
&\to (q_{\text{loop}}, 11, 11\$) \\
&\to (q_{\text{loop}}, 1, 1\$) \\
&\to (q_{\text{loop}}, \epsilon, \$) \\
&\to (q_{\text{accept}}, \epsilon, \epsilon)
\end{aligned}$$

Theorem

If a language is accepted by a PDA, then it is context-free.

Proof. Let A be accepted by PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$. We will construct a CFG $G = (V, \Sigma, R, S)$ with $L(G) = L(M)$.

Without loss of generality we assume that M has a unique final state q_{accept}. (That is, $F = \{q_{\mathrm{accept}}\}$.) We also assume that the stack is empty when it accepts, and that each operation is a push or a pop, but not both. If a PDA does not satisfy these conditions, it can be modified to meet them.

The variables of G are

$$V = \{A_{p,q} \mid p, q \in Q\}.$$

The start variable is $S = A_{q_0, q_{\mathrm{accept}}}$.

The variables of G are

$$V = \{A_{p,q} \mid p, q \in Q\}.$$

The start variable is $S = A_{q_0, q_{\mathrm{accept}}}$.

The variables of G are

$$V = \{A_{p,q} \mid p, q \in Q\}.$$

The start variable is $S = A_{q_0, q_{\mathrm{accept}}}$.

Idea: $A_{p,q}$ generates all strings w from which

$$(p, w, \epsilon) \xrightarrow{*} (q, \epsilon, \epsilon),$$

that is, all strings that take M from state p with empty stack to state q with empty stack.

The variables of G are

$$V = \{A_{p,q} \mid p, q \in Q\}.$$

The start variable is $S = A_{q_0, q_{\mathrm{accept}}}$.

Idea: $A_{p,q}$ generates all strings w from which

$$(p, w, \epsilon) \xrightarrow{*} (q, \epsilon, \epsilon),$$

that is, all strings that take M from state p with empty stack to state q with empty stack.

Then S will generate all w for which

$$(q_0, w, \epsilon) \xrightarrow{*} (q_{\mathrm{accept}}, \epsilon, \epsilon),$$

that is, all strings which M accepts.

We have the following rules:

1. For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a, b \in \Sigma_\epsilon$, if $(r, t) \in \delta(p, a, \epsilon)$ and $(q, \epsilon) \in \delta(s, b, t)$,

   $$p \xrightarrow{a, \epsilon \to t} r \qquad\qquad s \xrightarrow{b, t \to \epsilon} q$$

   we have the rule

   $$A_{p,q} \to aA_{r,s}b.$$

We have the following rules:

1. For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a, b \in \Sigma_\epsilon$, if $(r, t) \in \delta(p, a, \epsilon)$ and $(q, \epsilon) \in \delta(s, b, t)$,

   $$p \xrightarrow{a,\, \epsilon \to t} r \qquad\qquad s \xrightarrow{b,\, t \to \epsilon} q$$

   we have the rule

   $$A_{p,q} \to aA_{r,s}b.$$

2. For each $p, q, r \in Q$, we have the rule

   $$A_{p,q} \to A_{p,r}A_{r,q}.$$

We have the following rules:

1. For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a, b \in \Sigma_\epsilon$, if $(r, t) \in \delta(p, a, \epsilon)$ and $(q, \epsilon) \in \delta(s, b, t)$,

   $$p \xrightarrow{a,\, \epsilon \to t} r \qquad\qquad s \xrightarrow{b,\, t \to \epsilon} q$$

   we have the rule

   $$A_{p,q} \to aA_{r,s}b.$$

2. For each $p, q, r \in Q$, we have the rule

   $$A_{p,q} \to A_{p,r}A_{r,q}.$$

3. For each $p \in Q$, we have the rule

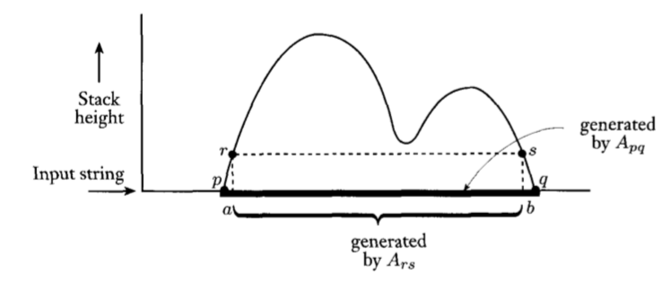
   $$A_{p,p} \to \epsilon.$$

1. For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a, b \in \Sigma_\epsilon$, if $(r, t) \in \delta(p, a, \epsilon)$ and $(q, \epsilon) \in \delta(s, b, t)$,

$$p \xrightarrow{a,\, \epsilon \to t} r \qquad\qquad s \xrightarrow{b,\, t \to \epsilon} q$$

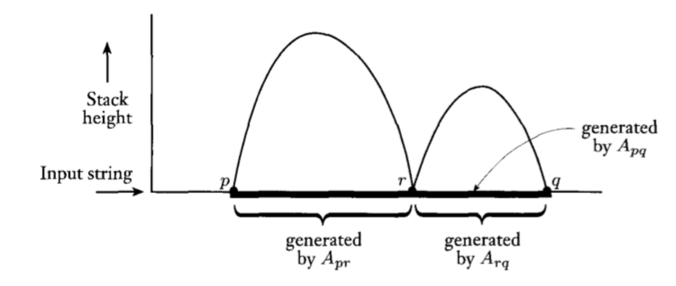
we have the rule

$$A_{p,q} \to aA_{r,s}b.$$



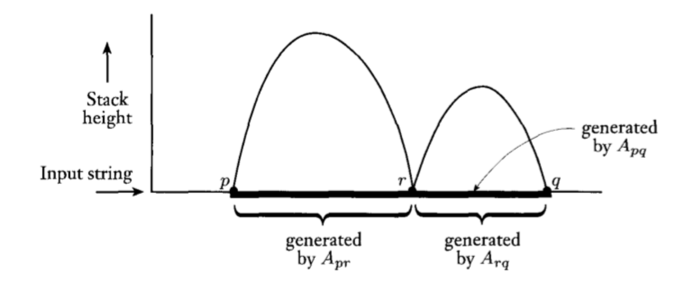
2. For each $p, q, r \in Q$, we have the rule

$$A_{p,q} \to A_{p,r} A_{r,q}.$$



2. For each $p, q, r \in Q$, we have the rule

$$A_{p,q} \to A_{p,r}A_{r,q}.$$



3. For each $p \in Q$, we have the rule

$$A_{p,p} \to \epsilon.$$

Lemma: $A_{p,q}$ generates exactly the strings x for which

$$(p, x, \epsilon) \xrightarrow{*} (q, \epsilon, \epsilon),$$

that is, all strings that take M from state p with empty stack to state q with empty stack.

Lemma: $A_{p,q}$ generates exactly the strings x for which

$$(p, x, \epsilon) \xrightarrow{*} (q, \epsilon, \epsilon),$$

that is, all strings that take M from state p with empty stack to state q with empty stack.

There are two directions:

- If $A_{p,q} \stackrel{*}{\Rightarrow} x$, then $(p, x, \epsilon) \xrightarrow{*} (q, \epsilon, \epsilon)$.
- If $(p, x, \epsilon) \xrightarrow{*} (q, \epsilon, \epsilon)$, then $A_{p,q} \stackrel{*}{\Rightarrow} x$.

Both directions are proved by induction (see book).

The Lemma implies that for all $x \in \Sigma^*$,

$$x \in L(M) \quad \Leftrightarrow \quad (q_0, x, \epsilon) \xrightarrow{*} (q_{\mathrm{accept}}, \epsilon, \epsilon)$$

The Lemma implies that for all $x \in \Sigma^*$,

$$\begin{aligned} x \in L(M) &\Leftrightarrow (q_0, x, \epsilon) \xrightarrow{*} (q_{\mathrm{accept}}, \epsilon, \epsilon) \\ &\Leftrightarrow A_{q_0, q_{\mathrm{accept}}} \overset{*}{\Rightarrow} x \end{aligned}$$

The Lemma implies that for all $x \in \Sigma^*$,

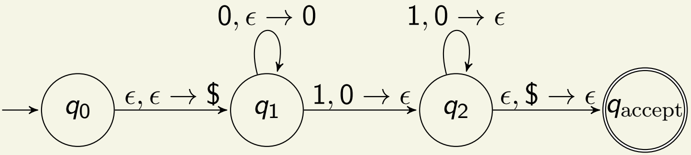
$$\begin{aligned} x \in L(M) &\Leftrightarrow (q_0, x, \epsilon) \xrightarrow{*} (q_{\mathrm{accept}}, \epsilon, \epsilon) \\ &\Leftrightarrow A_{q_0, q_{\mathrm{accept}}} \stackrel{*}{\Rightarrow} x \\ &\Leftrightarrow x \in L(G). \end{aligned}$$

The Lemma implies that for all $x \in \Sigma^*$,

$$\begin{aligned} x \in L(M) &\Leftrightarrow (q_0, x, \epsilon) \xrightarrow{*} (q_{\text{accept}}, \epsilon, \epsilon) \\ &\Leftrightarrow A_{q_0, q_{\text{accept}}} \stackrel{*}{\Rightarrow} x \\ &\Leftrightarrow x \in L(G). \end{aligned}$$

Therefore $L(M) = L(G)$. $\square$

Example

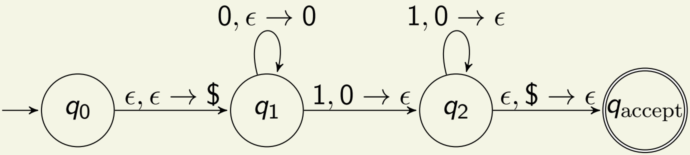


The grammar has 16 variables:

$$
\begin{array}{llll}
A_{q_0,q_0} & A_{q_0,q_1} & A_{q_0,q_2} & A_{q_0,q_{\text{accept}}} \\
A_{q_1,q_0} & A_{q_1,q_1} & A_{q_1,q_2} & A_{q_1,q_{\text{accept}}} \\
A_{q_2,q_0} & A_{q_2,q_1} & A_{q_2,q_2} & A_{q_2,q_{\text{accept}}} \\
A_{q_{\text{accept}},q_0} & A_{q_{\text{accept}},q_1} & A_{q_{\text{accept}},q_2} & A_{q_{\text{accept}},q_{\text{accept}}}
\end{array}
$$

The start variable is $A_{q_0,q_{\text{accept}}}$.

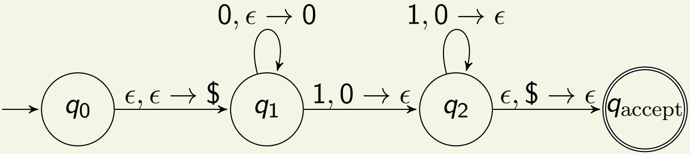
Example



Rules of the first form:

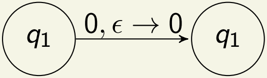
- $A_{q_0, q_{\text{accept}}} \to A_{q_1, q_2}$

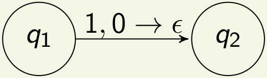
Example



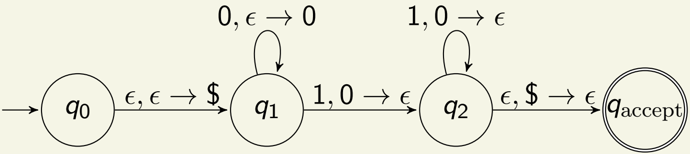
Rules of the first form:

- $A_{q_0, q_{\text{accept}}} \to A_{q_1, q_2}$
- $A_{q_1, q_2} \to 0 A_{q_1, q_1} 1$



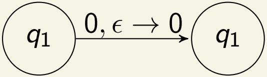


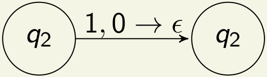
Example



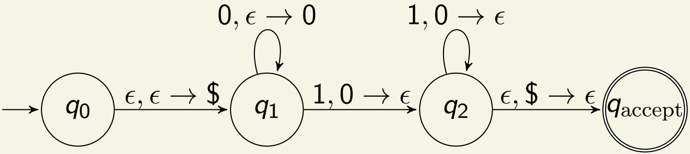
Rules of the first form:

- $A_{q_0, q_{\text{accept}}} \to A_{q_1, q_2}$
- $A_{q_1, q_2} \to 0 A_{q_1, q_1} 1$
- $A_{q_1, q_2} \to 0 A_{q_1, q_2} 1$





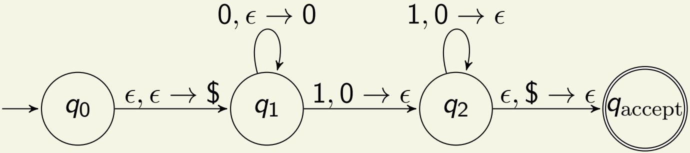
Example



Rules of the first form:

- $A_{q_0, q_{\mathrm{accept}}} \to A_{q_1, q_2}$
- $A_{q_1, q_2} \to 0 A_{q_1, q_1} 1$
- $A_{q_1, q_2} \to 0 A_{q_1, q_2} 1$

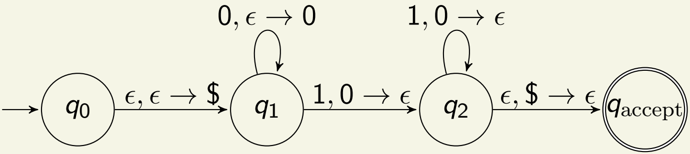
Example



Rules of the second form (64):

- $A_{q_0,q_0} \to$
  $A_{q_0,q_0} A_{q_0,q_0} | A_{q_0,q_1} A_{q_1,q_0} | A_{q_0,q_2} A_{q_2,q_0} | A_{q_0,q_{\mathrm{accept}}} A_{q_{\mathrm{accept}},q_0}$
- $A_{q_0,q_1} \to$
  $A_{q_0,q_0} A_{q_0,q_1} | A_{q_0,q_1} A_{q_1,q_1} | A_{q_0,q_2} A_{q_2,q_1} | A_{q_0,q_{\mathrm{accept}}} A_{q_{\mathrm{accept}},q_1}$
- ...

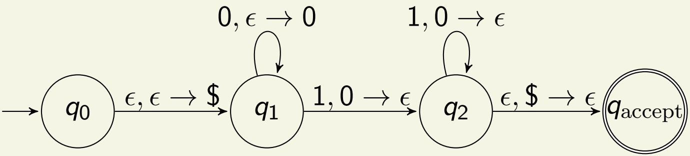
Example



Rules of the third form (4):

- $A_{q_0,q_0} \to \epsilon$
- $A_{q_1,q_1} \to \epsilon$
- $A_{q_2,q_2} \to \epsilon$
- $A_{q_{\text{accept}},q_{\text{accept}}} \to \epsilon$

Example – Input: $w = 0011$



Computation in PDA:

$$
\begin{aligned}
(q_0, 0011, \epsilon) &\to (q_1, 0011, \$) \\
&\to (q_1, 011, 0\$) \\
&\to (q_1, 11, 00\$) \\
&\to (q_2, 1, 0\$) \\
&\to (q_2, \epsilon, \$) \\
&\to (q_{\mathrm{accept}}, \epsilon, \epsilon)
\end{aligned}
$$

Derivation in CFG:

$$
\begin{aligned}
A_{q_0, q_{\mathrm{accept}}} &\Rightarrow A_{q_1, q_2} \\
&\Rightarrow 0A_{q_1, q_2}1 \\
&\Rightarrow 00A_{q_1, q_1}11 \\
&\Rightarrow 0011
\end{aligned}
$$