PDA-to-CFG notes

CSE 30151 Spring 2016

February 2016

Recap

We have already seen how Context Free Grammars (CFGs) and Pushdown Automata (PDAs) are two sides of the same coin, but operate on a different level:

- a CFG generates a string by constructing a tree, as it applies its rules.
- a PDA has to go from left-to-right in order to accept a string

Example

$$L = \{\mathtt{0}^n\mathtt{1}^n, n \geq 0\}$$

The CFG G is:

$$S \rightarrow \mathtt{0}S\mathtt{1} \mid \varepsilon$$

Figure 1 compares the CFG derivation of the string `000111` with the run of the equivalent PDA (Sipser, Figure 2.15).

Converting a PDA to a CFG

Prerequisites for the PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$:

1. Single accept state
2. Empties stack before accepting
3. Each transition either pushes one symbol to the stack, or pops one symbol off the stack, but not both or none.

We construct a CFG G that has the following rules:

1. $\forall p \in Q$ put rule $A_{pp} \rightarrow \varepsilon$
2. $\forall p, q, r \in Q$ put rule $A_{pq} \rightarrow A_{pr}A_{rq}$
3. $\forall p, r, s, q \in Q$ put rule $A_{pq} \rightarrow \mathtt{a}A_{rs}\mathtt{b}$ if
   - $(r, \mathtt{u}) \in \delta(p, \mathtt{a}, \varepsilon)$ and
   - $(q, \varepsilon) \in \delta(s, \mathtt{b}, \mathtt{u})$.

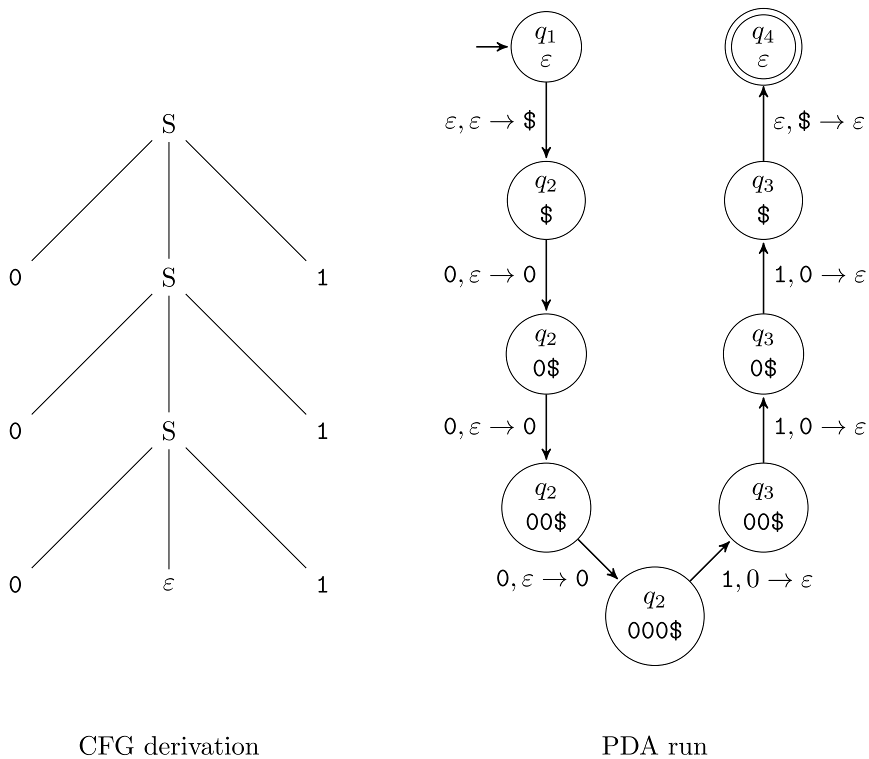
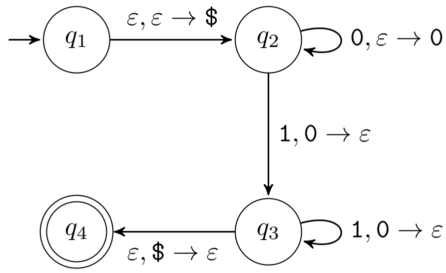


Figure 1: Side-by-side comparison of CFG derivation and PDA run for string 000111. In the PDA run, the stack is shown under each state.

4. The start variable is $A_{q_0 q_{accept}}$

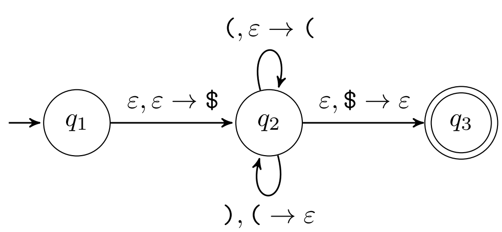
Example (from PDA in fig 2.15)



The produced CFG will be:

$$
\begin{aligned}
A_{11} &\to \varepsilon \\
A_{22} &\to \varepsilon \\
A_{33} &\to \varepsilon \\
A_{44} &\to \varepsilon \\
A_{11} &\to A_{11}A_{11} \mid A_{12}A_{21} \mid A_{13}A_{31} \mid A_{14}A_{41} \\
A_{12} &\to A_{11}A_{12} \mid A_{12}A_{22} \mid A_{13}A_{32} \mid A_{14}A_{42} \\
A_{13} &\to A_{11}A_{13} \mid A_{12}A_{23} \mid A_{13}A_{33} \mid A_{14}A_{43} \\
&\dots \\
A_{42} &\to A_{41}A_{12} \mid A_{42}A_{22} \mid A_{43}A_{32} \mid A_{44}A_{42} \\
A_{43} &\to A_{41}A_{13} \mid A_{42}A_{23} \mid A_{43}A_{33} \mid A_{44}A_{43} \\
A_{44} &\to A_{41}A_{14} \mid A_{42}A_{24} \mid A_{43}A_{34} \mid A_{44}A_{44} \\
A_{23} &\to \mathtt{0}A_{22}\mathtt{1} \mid \mathtt{0}A_{23}\mathtt{1} \\
A_{14} &\to \varepsilon A_{23} \varepsilon \\
S &\to A_{14}
\end{aligned}
$$

Example (The Dyck language)



The produced CFG (after eliminating unreachable non-terminals) will be:

$$
\begin{aligned}
A_{13} &\to \varepsilon A_{22} \varepsilon \\
A_{22} &\to A_{22}A_{22} \mid \varepsilon \mid \mathtt{(}A_{22}\mathtt{)} \\
S &\to A_{13}
\end{aligned}
$$