PDA - Final part

SITE :  http://www.sir.blois.univ-tours.fr/~mirian/
Let
\[ P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \]
be a PDA. The language accepted by \( P \) by final state is:
\[ L(P) = \{ w \mid (q_0, w, Z_0) \xrightarrow{*} (q, \epsilon, \alpha), q \in F \} \]
for some state \( q \) in \( F \) and any input stack string \( \alpha \).

Starting in the initial ID with \( w \) waiting on the input, \( P \) consumes \( w \) from the input and enters an accepting state. The contents of the stack at that time is irrelevant.
PDA- Acceptance by empty stack

Let

\[ P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \]

be a PDA. The **language accepted by** \( P \) **by empty stack** is:

\[ N(P) = \{ w \mid (q_0, w, Z_0) \vdash (q, \epsilon, \epsilon) \} \]

where \( q \) is any state

\( N(P) \) is the set of inputs \( w \) that \( P \) can consume at the same time empty the stack.
From final state to empty stack

Let $P_N$ be a PDA by empty stack.

Let $P_F$ be a PDA by final state.

Theorem:
If $L = N(P_N)$ for some PDA $P_N$, then there exist a PDA $P_F$, such that

$$L = L(P_F)$$
Equivalence of PDA and CFG

A language is

generated by a CFG

iff it is

accepted by a PDA by empty stack

iff it is

accepted by a PDA by final state
Deterministic PDA

A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is deterministic iff

1. $\delta(q, a, X)$ is always empty or a singleton
2. If $\delta(q, a, X)$ is nonempty, then $\delta(q, \epsilon, X)$ must be empty

Example. $L_{w,cw} = \{wcw \mid w \in \{0,1\}^*\}$ is recognized by the following PDA

- $q_1 \xrightarrow{c, 1/1} q_1$
- $q_1 \xrightarrow{c, 0/0} q_1$
- $q_1 \xrightarrow{c, Z_0/Z_0} q_1$
- $q_2 \xrightarrow{0, Z_0/0Z_0} q_1$
- $q_2 \xrightarrow{1, Z_0/1Z_0} q_1$
- $q_2 \xrightarrow{0, 0/0} q_2$
- $q_2 \xrightarrow{0, 1/01} q_2$
- $q_2 \xrightarrow{1, 0/10} q_2$
- $q_2 \xrightarrow{1, 1/11} q_2$
- $q_2 \xrightarrow{\epsilon, Z_0/Z_0} q_2$
- $q_2 \xrightarrow{0, 0/\epsilon} q_3$
- $q_2 \xrightarrow{1, 1/\epsilon} q_3$
- $q_3 \xrightarrow{\epsilon, Z_0/Z_0} q_3$
Importance of DPDA

While PDA are by definition nondeterministic, the deterministic subcase is quite important.

**Parses in general behave like DPDA**

A DPDA can accept languages like $L_{wcw}$ that are not regular, but there are CFL (like $L_{wwr}$) that cannot be accepted by a DPDA.

Languages accepted by DPDA all have unambiguous grammar

**Theorem:** If $L$ is the language accepted by some DPDA $P$, then $L$ has an unambiguous CFG.

The DPDA languages are not exactly equal the subset of CFL that are not inherently ambiguous.

Example: $L_{wwr}$ has an unambiguous grammar and it is not a DPDA language.
Important properties

Let

- NFA: Nondeterministic finite automaton
- DFA: Deterministic finite automaton
- RE: Regular expression
- DDPA: Deterministic pushdown automaton
- DPA: Nondeterministic pushdown automaton
- CFL: Context free language

1. We know that if $L_{Regular}$ is a regular language, then there exist NFA, DFA and RE such that

$$L(NFA) = L(DFA) = L(RE) = L_{Regular}$$

2. $L(PDA) = CFL$

3. $L(DPDA) = L_1$ such that $L_1$ has an unambiguous CFG

4. Regular languages $\subseteq L(DPDA) \subseteq CFL$