Context-free Languages & Pushdown Automata
Automata with memory: Pushdown Automata

The limitation of DFAs is that they don’t have a memory. Various (simple) memory models are possible:

Queue First in, first out (like a plastic cup dispenser in a coffee machine)

Stack Last in, first out (like plates in a student restaurant)

Pushdown automata extend automata with a stack

Each item carries an element of stack alphabet $\Gamma$
A pushdown automaton is a sextuple \((Q, \Sigma, \Gamma, \delta, q_0, F)\) with

- \(Q\) a finite set of states
- \(q_0\) an element of \(Q\), the initial state
- \(F\) a subset of \(Q\)
- \(\Sigma\) a finite set of symbols (input alphabet)
- \(\Gamma\) the stack alphabet
- \(\delta\) a map (‘afbeelding’)

\[
\delta : Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q \times \Gamma \cup \{\lambda\})
\]

We write e.g. \(\delta(q_i, a, A) = \{[q_j, B], [q_k, C]\}\)

We understand the \(Q, \Sigma, \mathcal{P}, \lambda\). New is the \(\Gamma\): alphabet of stack symbols

The stack is not mentioned, but it is used in the operation of the PDA!
Action of a PDA

\[ [q', B] \in \delta(q, a, A) \] and you can pop \( A \) and do push \( B \)

\[ L(M) = \{ w \in \Sigma^* \mid [q_0, w, \lambda] \vdash^* [q_i, \lambda, \lambda] \land q_i \in F \} \]
Example of a PDA

Accepts \( \{ a^n b^n \mid n \geq 0 \} \)
Non-deterministic aspect

accepts \( \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\} \)

Non-determinism is essential: in general it cannot be eliminated
Palindromes over $\Sigma = \{a, b\}$

accepts $\{uu^R \in \Sigma^* \mid u \in \Sigma^*\}$ the even palindromes

accepts $\{u\sigma u^R \in \Sigma^* \mid u \in \Sigma^*, \sigma \in \Sigma\}$ the odd palindromes

accepts the palindromes
A PDA is extended if one may push a word over the stack alphabet

formally \([q_2, BCD] \in \delta(q_1, s, A)\), and in general

\[
\delta : Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q \times \Gamma^*)
\]

**Prop.** Extended PDAs accept the same class of languages as PDAs

**Proof.** Replace a transition like above by
In a PDA we have by definition
\[ w \in L(M) \iff [q_0, w, \lambda] \vdash^* [q_i, \lambda, \lambda] \land q_i \in F \]

That is acceptance by empty stack and final state

Alternatively we can define
\[ w \in L_F(M) \iff [q_0, w, \lambda] \vdash^* [q_i, \lambda, \alpha] \land q_i \in F \]

acceptance by final state

or
\[ w \in L_E(M) \iff [q_0, w, \lambda] \vdash^+ [q_i, \lambda, \lambda] \]

acceptance by empty stack
Proposition Let $M$ be a PDA.
(i) For each $L \in L_F(M)$ there is an $M'$ such that $L \in L(M')$.
(ii) For each $L \in L_E(M)$ there is an $M'$ such that $L \in L(M')$.

Proof sketch for (i): Transform an $M$ like

Proposition
(i) PDAs with acceptance by final state accept the same class of languages as PDAs
(ii) PDAs with acceptance by empty stack accept the same class of languages as PDAs
Theorem Let $L$ be a language over $\Sigma^*$. Then the following are equivalent.

(i) $L = L(G)$ for some context-free grammar $G$

(ii) $L = L(M)$ for some PDA $M$

We will not prove this, but some intuition comes from exercises.
Pumping Lemma for context-free languages

Let $L$ be a context-free language over $\Sigma^*$

Then there exists a number $k > 0$
such that every word $z \in L$ with $|z| > k$
can be written as $z = u_1 v_1 w v_2 u_2$ such that

(i) $|v_1 w v_2| \leq k$
(ii) $|v_1| + |v_2| > 0$
(iii) $u_1 v_1^i w v_2^i u_2 \in L$ for all $i \geq 0$

Proofsketch. If there is a sufficiently large word $z \in L$
then in the derivation of $z$ one has

$S \rightarrow u_1 A u_2$
$A \rightarrow v_1 A v_2$ \hspace{1cm} a grammar loop!
$A \rightarrow w$

so that $S \Rightarrow u_1 v_1 w v_2 u_2 = z$ and also

$S \Rightarrow u_1 A u_2 \Rightarrow u_1 v_1 A v_2 u_2 \Rightarrow \cdots u_1 v_1^i A v_2^i u_2 \Rightarrow u_1 v_1^i w v_2^i u_2 \in L$ \blacksquare
• \( L_1 := \{ a^k b^k c^k \mid k \geq 0 \} \) is not context-free  
  because it violates the pumping lemma for CF languages.

• \( L_2 := \{ ww \mid w \in \{a, b\}^* \} \) is not context-free  
  because it violates the pumping lemma for CF languages.

• NB: \( L_3 := \{ a^k b^k \mid k \geq 0 \} \) is context-free hence satisfies the pumping lemma for CF languages.