Context-free Languages & Pushdown Automata

A pushdown automaton is a sextuple $\langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$ with

Q	a finite set of states
q_0	an element of Q, the initial state
F	a subset of Q
Σ	a finite set of symbols (input alphabet)
Γ	the *stack alphabet*
δ	a map ('afbeelding')
	$\delta : Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q \times \Gamma \cup \{\lambda\})$

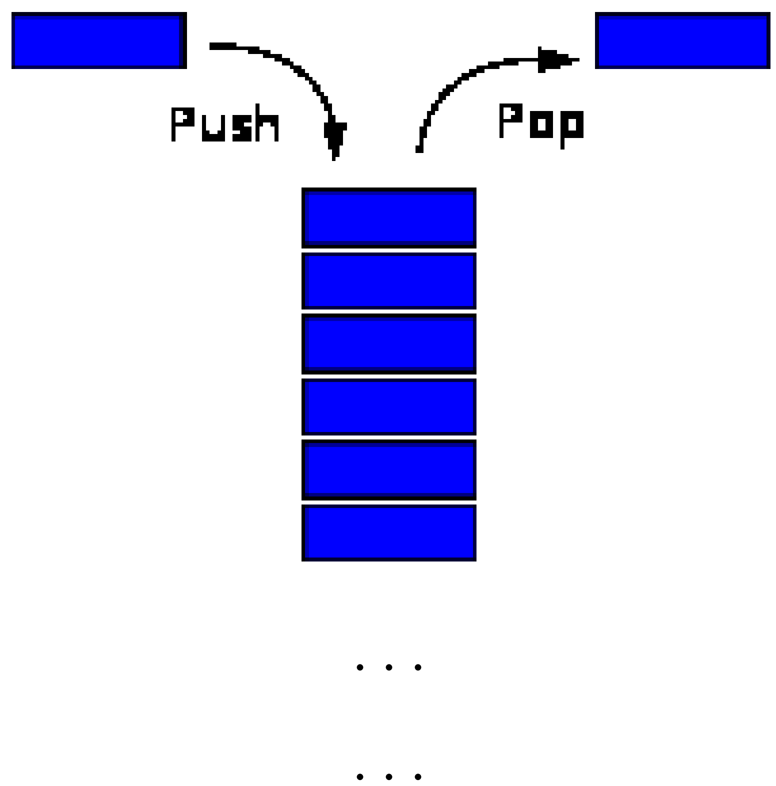
We write e.g. $\delta(q_i, a, A) = \{[q_j, B], [q_k, C]\}$

We understand the $Q, \Sigma, \mathcal{P}, \lambda$. New is the Γ: alfabet of stack symbols

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

A stack

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

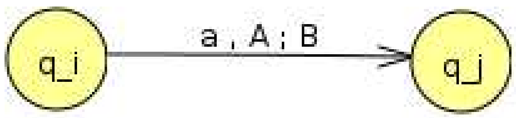


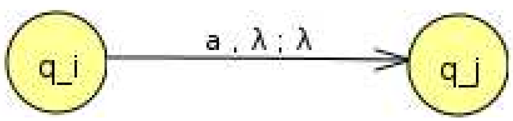
Each item carries an element of Γ

First used by Turing in 1946

Action of a PDA

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

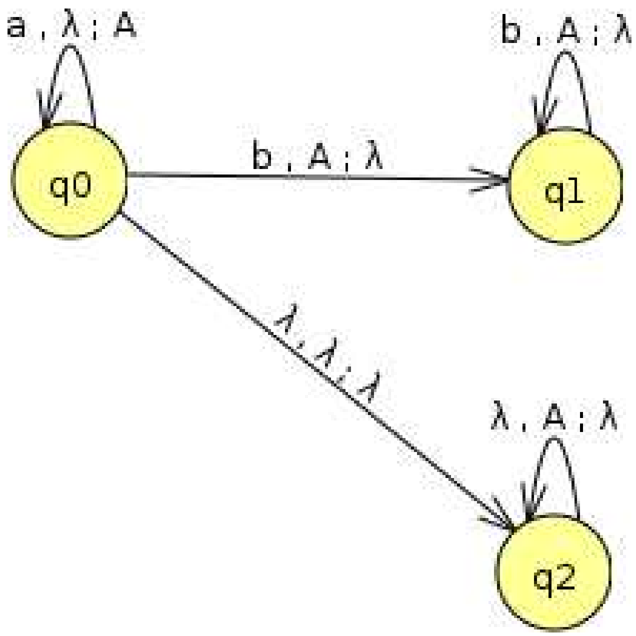




$$L(M) = \{w \in \Sigma^* \mid [q_0, w, \lambda] \vdash^* [q_i, \lambda, \lambda] \ \& \ q_i \in F\}$$

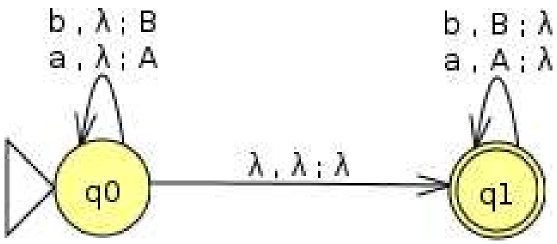
Example of a PDA

Accepts $\{a^n b^n \mid n \geq 0\}$

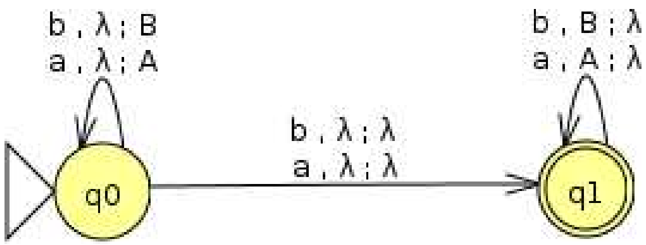


accepts $\{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$

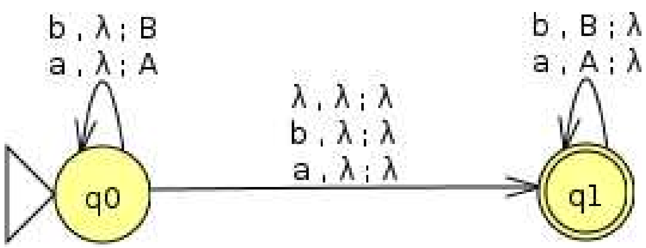
Non-determinism is essential: in general it cannot be eliminated



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

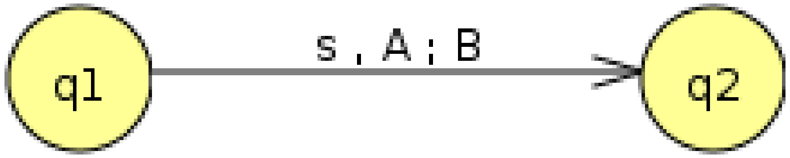


accepts $\{u\sigma u^R \in \Sigma^* \mid u \in \Sigma^*,\ \sigma \in \Sigma\}$ the odd palindromes



accepts the palindromes

Definition. A PDA is *atomic* if for all transitions



with $s \in \Sigma \cup \{\lambda\}$ one has A or B is λ

More formally

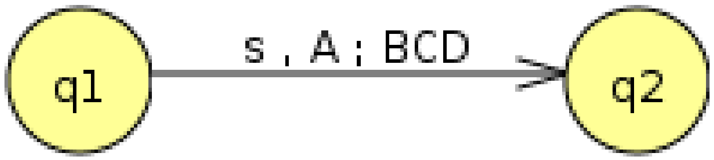
$$[q_2, B] \in \delta(q_1, s, A) \;\Rightarrow\; A \text{ or } B \text{ is } \lambda$$

Proposition. Atomic PDAs are good enough (for accepting languages)

Proof. Replace a transition like above with A, B both not λ by

This is 'atomic' and runs in the right way ■

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

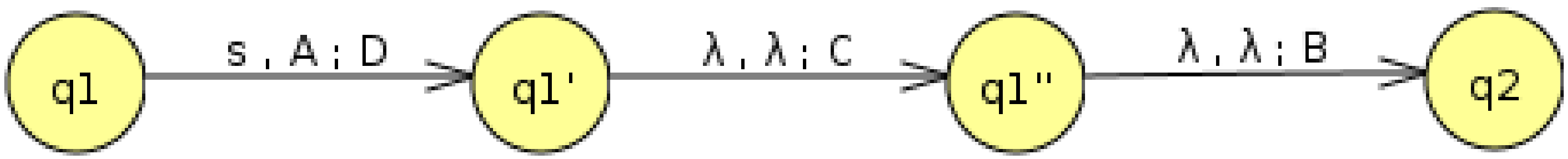


formally $[q_2, BCD] \in \delta(q_1, s, A)$, and in general

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

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

$$w \in L(M) \;\Leftrightarrow\; [q_0, w, \lambda] \vdash^* [q_i, \lambda, \lambda] \;\&\; q_i \in F$$

That is *acceptance by empty stack and final state*

Alternatively we can define

$$w \in L_F(M) \;\Leftrightarrow\; [q_0, w, \lambda] \vdash^* [q_i, \lambda, \alpha] \;\&\; q_i \in F$$

acceptance by final state

or

$$w \in L_E(M) \;\Leftrightarrow\; [q_0, w, \lambda] \vdash^+ [q_i, \lambda, \lambda]$$

acceptance by empty stack

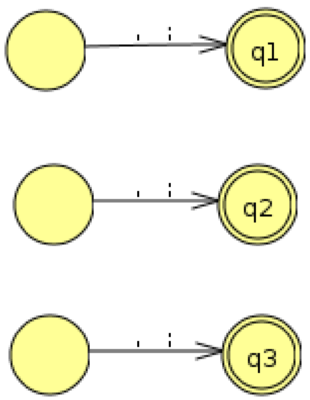
Proposition Let M be a PDA.

(i) For each $L \in L_F(M)$ there is an M' such that $L \in L(M')$.

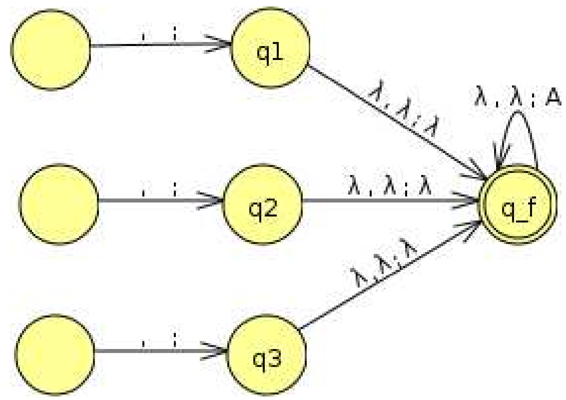
(ii) For each $L \in L_E(M)$ there is an M' such that $L \in L(M')$.

Proof.

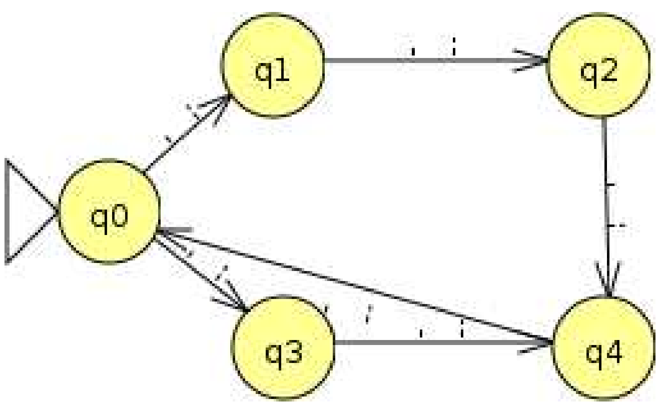
(i) Transform an M like



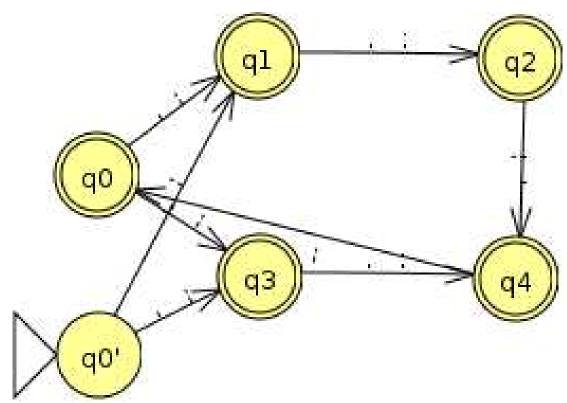
into



(ii) Transform an M like



into



∎

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

Proof. Do as exercise. ■

Theorem Let L be a language over Σ^*.

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. ■

Let L be a context-free language over Σ^*

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

$$\begin{array}{lcl} S & \to & u_1 A u_2 \\ A & \to & v_1 A v_2 \\ A & \to & w \end{array}$$

a grammar loop!

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

$$S \Rightarrow u_1 A u_2 \Rightarrow u_1 v_1 A v_2 u_2 \Rightarrow \cdots u_1 v_1^i A v_2^i u_2 \Rightarrow u_1 v_1^i w v_2^i u_2 \in L \blacksquare$$

$\{a^k b^k \mid k>0\}$ is context-free hence satisfies the pumping lemma

$\{a^k b^k c^k \mid k>0\}$ is not context-free it violates the pumping lemma

See running examples in JFLAP

Note that in logic one has

$$A \rightarrow B \quad \not\Rightarrow \quad B \rightarrow A$$

$$A \rightarrow B \quad \Rightarrow \quad \neg B \rightarrow \neg A$$