Context-free Languages & Pushdown Automata

Let G be given by

$$V = \{\mathtt{S}, \mathtt{A}, \mathtt{B}\}$$
$$\Sigma = \{\mathtt{a}, \mathtt{b}\}$$

$$\begin{array}{lcl} \mathtt{S} & \rightarrow & \mathtt{aB} \mid \mathtt{bA} \\ \mathtt{A} & \rightarrow & \mathtt{a} \mid \mathtt{aS} \mid \mathtt{BAA} \\ \mathtt{B} & \rightarrow & \mathtt{b} \mid \mathtt{bS} \mid \mathtt{ABB} \end{array}$$

Show that $\mathtt{ababba} \in L(G)$ but $\mathtt{baaababba} \notin L(G)$

Characterize $L(G)$; give motivation.

$G = \langle V, \Sigma, R, S \rangle$ with $S \in V$ and R a set of rules of the form

$$\boxed{\texttt{A}{\rightarrow}\sigma\texttt{B} \text{ or } \texttt{A}{\rightarrow}\lambda}$$

where $A, B \in V$ and $\sigma \in \Sigma$. Such grammar is called *regular*

Many books are a bit more liberal

$$\boxed{\texttt{A}{\rightarrow}\sigma\texttt{B} \text{ or } \texttt{A}{\rightarrow}\lambda \text{ or } \texttt{A}{\rightarrow}\sigma}$$

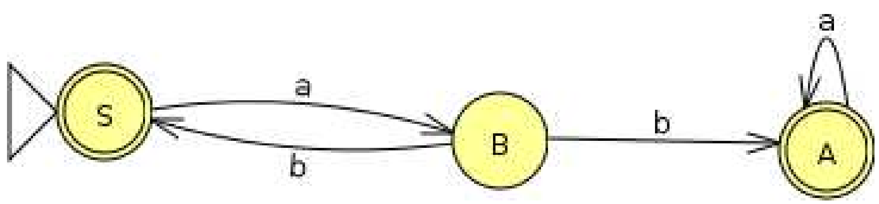
But that is essentially the same

$\texttt{A}{\rightarrow}\sigma$ can be simulated by $\texttt{A}{\rightarrow}\sigma\texttt{A}'$ and $\texttt{A}'{\rightarrow}\lambda$

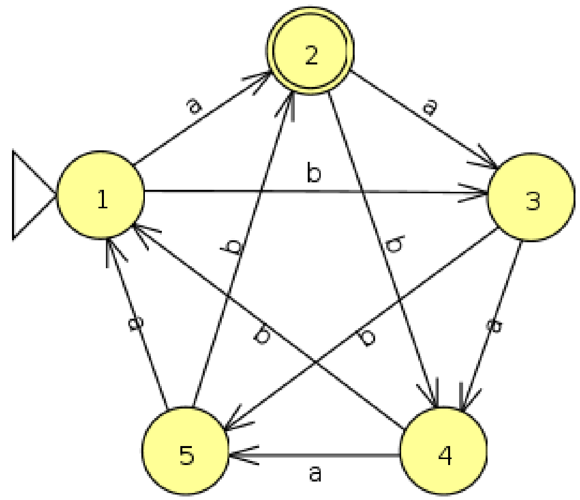
If G is a regular grammar, then $L(G)$ is a regular language.

Example
$$\begin{array}{lcl} S & \rightarrow & \mathtt{a}B \mid \lambda \\ B & \rightarrow & \mathtt{b}S \mid \mathtt{b}A \\ A & \rightarrow & \mathtt{a}A \mid \lambda \end{array}$$

NFA



Conversely, given a DFA



Can we find a regular grammar generating the same language?

Yes:

S=1	$\to$	a2	|	b3		
2	$\to$	a3	|	b4	|	λ
3	$\to$	a4	|	b5		
4	$\to$	a5	|	b1		
5	$\to$	a1	|	b2		

The *context-sensitive grammars* have production rules of the form

$$uXv \to uwv,$$

with $u, v \in (\Sigma \cup V)^*$ arbitrary and $w \in (\Sigma \cup V)^*$ with $w \neq \lambda$
or $S \to \lambda$ if S doesn't occur on the right of a rule

For the *phrase-structure grammars* production rules are of the form

$$uXv \to uwv,$$

with $u, v \in (\Sigma \cup V)^*$ and $w \in (\Sigma \cup V)^*$ arbitrary.

Grammar/language	rule	condition
regular	$A \to \sigma B \mid \lambda$	$A, B \in V$, $\sigma \in \Sigma$
context-free	$A \to w$	$A \in V$, $w \in (V \cup \Sigma)^*$
context-sensitive	$uAv \to uwv$ or $S \to \lambda$	$A \in V$, $u,v,w \in (V \cup \Sigma)^*$, $w \neq \lambda$ if S 'not on the right'
phrase-structure/enumerable	$uAv \to uwv$	$A \in V$, $u,v,w \in (V \cup \Sigma)^*$

Given the grammar G

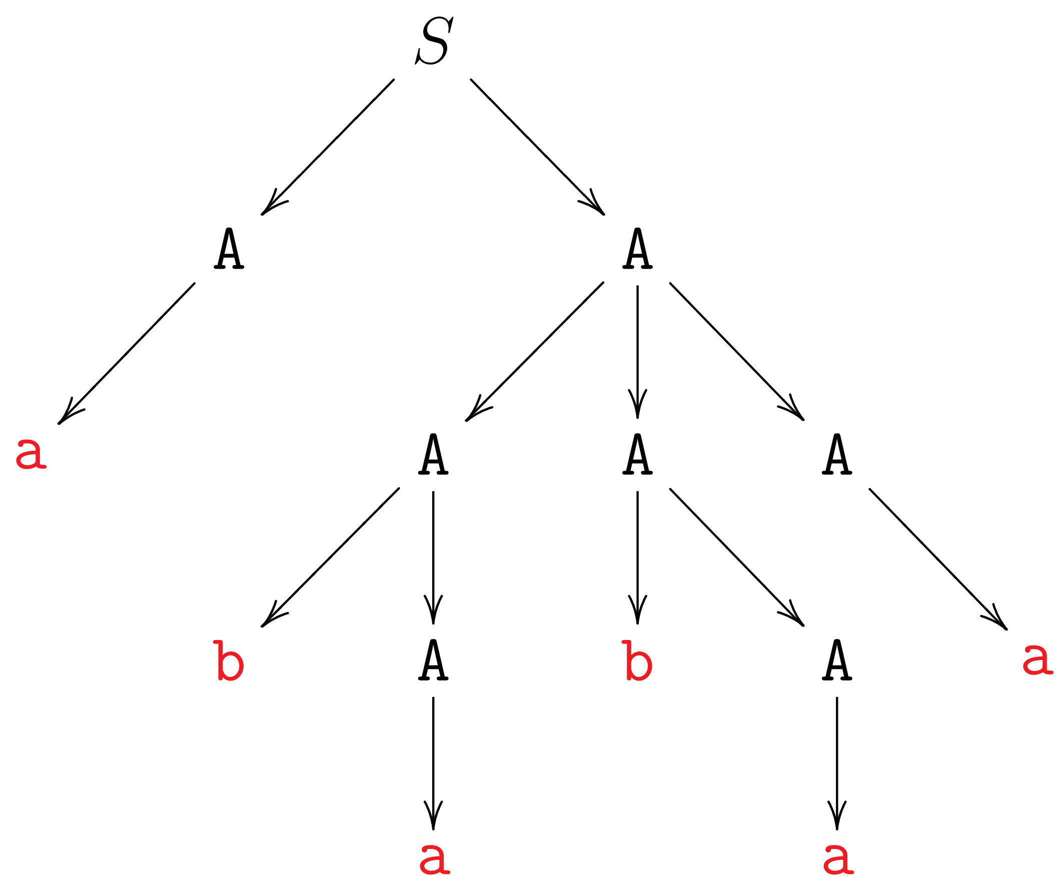
$$\mathtt{S} \rightarrow \mathtt{AA}$$

$$\mathtt{A} \rightarrow \mathtt{AAA} \mid \mathtt{bA} \mid \mathtt{Ab} \mid \mathtt{a}$$

We have

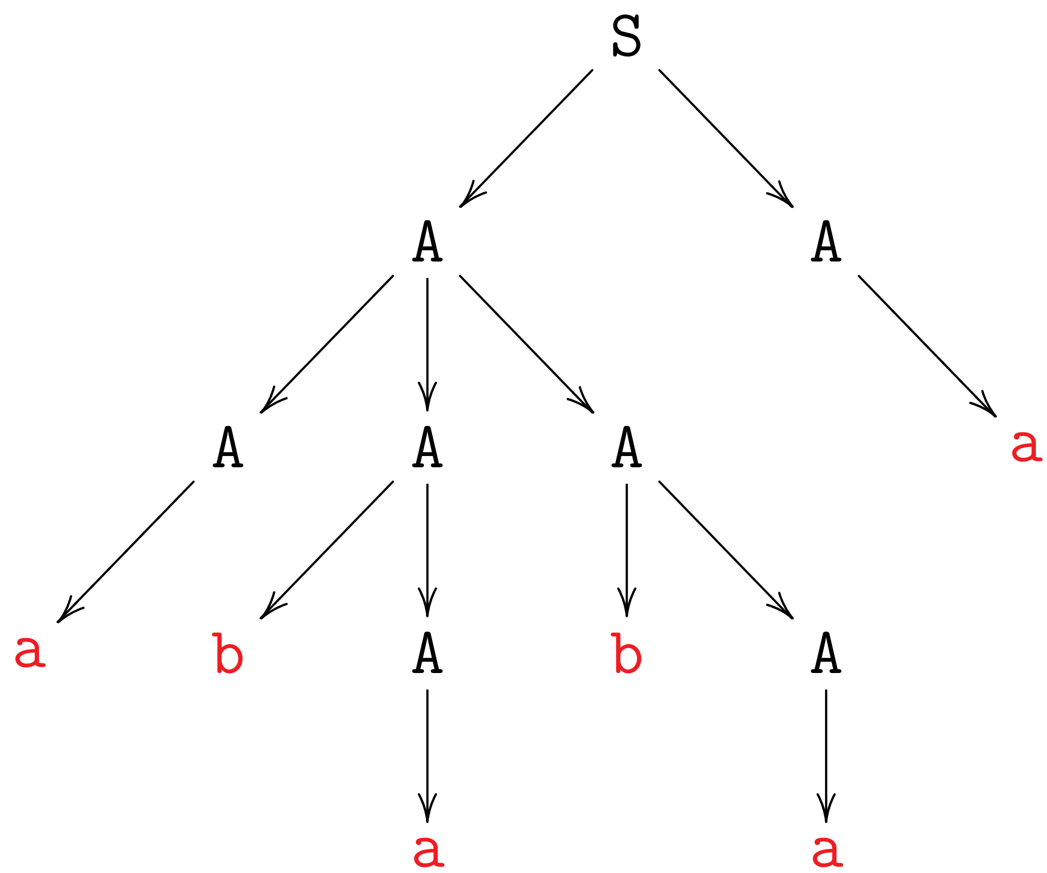
$$\begin{aligned} S &\Rightarrow \mathtt{AA} \\ &\Rightarrow \mathtt{aA} \\ &\Rightarrow \mathtt{aAAA} \\ &\Rightarrow \mathtt{abAAA} \\ &\Rightarrow \mathtt{abaAA} \\ &\Rightarrow \mathtt{ababAA} \\ &\Rightarrow \mathtt{ababaA} \\ &\Rightarrow \mathtt{ababaa} \end{aligned} \qquad \begin{aligned} S &\Rightarrow \mathtt{AA} \\ &\Rightarrow \mathtt{AAAA} \\ &\Rightarrow \mathtt{aAAA} \\ &\Rightarrow \mathtt{abAAA} \\ &\Rightarrow \mathtt{abaAA} \\ &\Rightarrow \mathtt{ababAA} \\ &\Rightarrow \mathtt{ababaA} \\ &\Rightarrow \mathtt{ababaa} \end{aligned} \qquad \begin{aligned} S &\Rightarrow \mathtt{AA} \\ &\Rightarrow \mathtt{Aa} \\ &\Rightarrow \mathtt{AAAa} \\ &\Rightarrow \mathtt{AAbAa} \\ &\Rightarrow \mathtt{AAbaa} \\ &\Rightarrow \mathtt{AbAbaa} \\ &\Rightarrow \mathtt{Ababaa} \\ &\Rightarrow \mathtt{ababaa} \end{aligned}$$

$$
\begin{aligned}
\mathtt{S} &\Rightarrow \mathtt{AA} \\
&\Rightarrow \mathtt{aA} \\
&\Rightarrow \mathtt{aAAA} \\
&\Rightarrow \mathtt{abAAA} \\
&\Rightarrow \mathtt{abaAA} \\
&\Rightarrow \mathtt{ababAA} \\
&\Rightarrow \mathtt{ababaA} \\
&\Rightarrow \mathtt{ababaa}
\end{aligned}
$$



Derivation trees (2)

$$
\begin{aligned}
\mathtt{S} &\Rightarrow \mathtt{AA} \\
&\Rightarrow \mathtt{AAAA} \\
&\Rightarrow \mathtt{aAAA} \\
&\Rightarrow \mathtt{abAAA} \\
&\Rightarrow \mathtt{abaAA} \\
&\Rightarrow \mathtt{ababAA} \\
&\Rightarrow \mathtt{ababaA} \\
&\Rightarrow \mathtt{ababaa}
\end{aligned}
$$



M is a DFA over Σ if $M = (Q, \Sigma, q_0, \delta, F)$ with

Q	is a finite set of *'states'*
Σ	is a finite *alphabet*
$q_0 \in Q$	is the *initial* state
$F \subseteq Q$	is a finite set of *final* states
$\delta : Q \times \Sigma \to Q$	is the *transition* function (often given by a table)

Reading function $\hat{\delta} : Q \times \Sigma^* \to Q$

$$\begin{aligned} \hat{\delta}(q, \lambda) &= q \\ \hat{\delta}(q, a) &= \delta(q, a) \\ \hat{\delta}(q, aw) &= \hat{\delta}(\delta(q, a), w) \end{aligned}$$

Language accepted by M, notation $L(M)$:

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

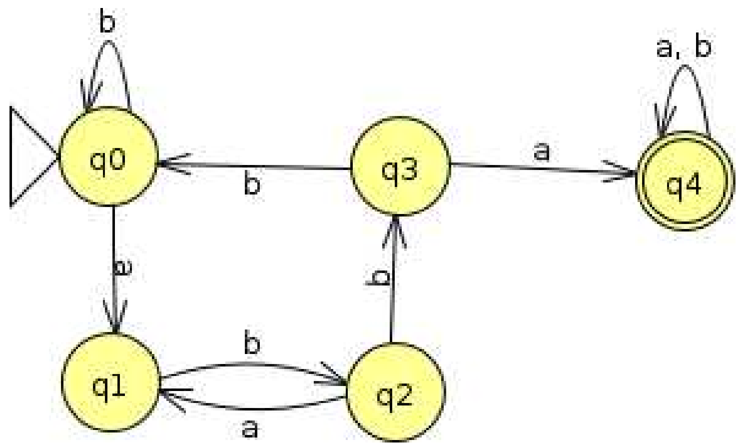
Reading words $w \in \Sigma^*$ in a DFA

Computation for $\hat{\delta}(q, w)$ in the example $w = abba$:

$$
\begin{aligned}
[q, abba] &\vdash [\delta(q, a), bba] \\
&\vdash [\delta(\delta(q, a), b), ba] \\
&\vdash [\delta(\delta(\delta(q, a), b), b), a] \\
&\vdash [\delta(\delta(\delta(\delta(q, a), b), b), a), \lambda]
\end{aligned}
$$

Example transition table for δ with $Q = \{0, 1, 2, 3, 4\}$, $\Sigma = \{a, b\}$, $q_0 = 0$, and $F = \{4\}$

δ	a	b
0	1	0
1	1	2
2	1	3
3	4	0
4	4	4



We have $\hat{\delta}(0, abba) = 4 \in F$ and $[0, abba] \vdash^* [4, \lambda]$, hence $abba \in L(M)$
Similarly $\hat{\delta}(0, baba) = 1 \notin F$; so even if $[0, baba] \vdash^* [1, \lambda]$ we have $baba \notin L(M)$.
Even if $\hat{\delta}(1, bba) = 4 \in F$ and $[1, bba] \vdash^* [4, \lambda]$ we have $bba \notin L(M)$.

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*

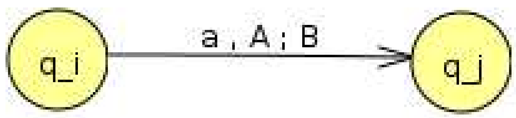
δ a map ('afbeelding')

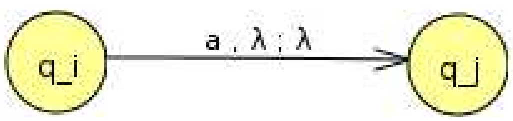
$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q \times \Gamma \cup \{\lambda\})$$

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

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

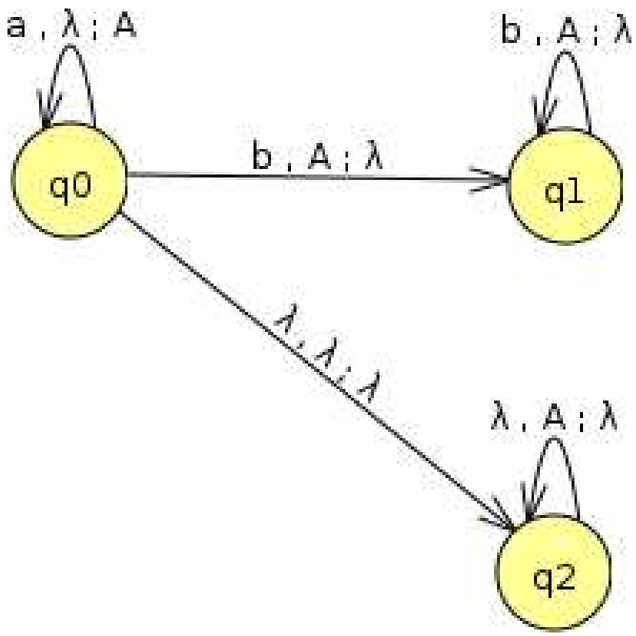
$[q', B] \in \delta(q, a, A)$ and you can pop A and do push B





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

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



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