Fundamentele Informatica 1 (I&E)

najaar 2015

`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi1ie/`

Rudy van Vliet
kamer 124 Snellius, tel. 071-527 5777
rvvliet(at)liacs(dot)nl

college 9, 24 november 2015

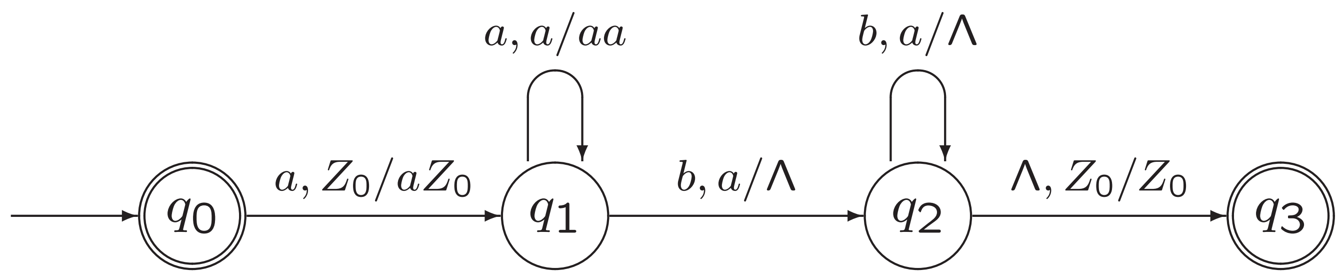
5. Pushdown Automata

5.1. Definitions and Examples

5.2. Deterministic Pushdown Automata

Example 5.3. A PDA Accepting the Language *AnBn*

$$AnBn = \{a^i b^i \mid i \geq 0\}$$



A slide from lecture 8:

Definition 5.1. A Pushdown Automaton

A *pushdown automaton* (PDA)
is a 7-tuple $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, where

Q is a finite set of states.
Σ and Γ are finite sets, the *input* and *stack* alphabet.
q_0, the initial state, is an element of Q.
Z_0, the initial stack symbol, is an element of Γ.
A, the set of accepting states, is a subset of Q.
δ, the transition function, is a function from $Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma$
to the set of finite subsets of $Q \times \Gamma^*$.

In principle, Z_0 may be removed from the stack,
but often it isn't.

Exercise.

Give transition diagrams for PDAs accepting each of the following languages.

a. *Balanced* $= \{$balanced strings of brackets [and]$\}$

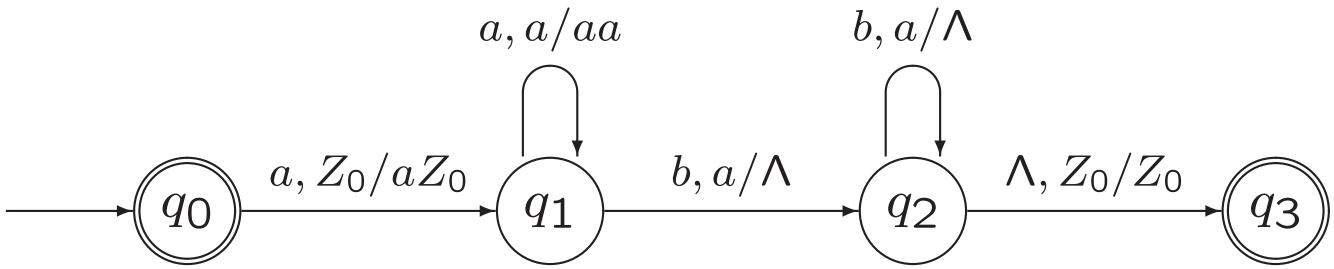
b. *AEqB* $= \{x \in \{a, b\}^* \mid n_a(x) = n_b(x)\}$

5.2. Deterministic Pushdown Automata

reg. languages	FA	reg. grammar	reg. expression
determ. cf. languages	DPDA		
cf. languages	PDA	cf. grammar	
re. languages	TM	unrestr. grammar	

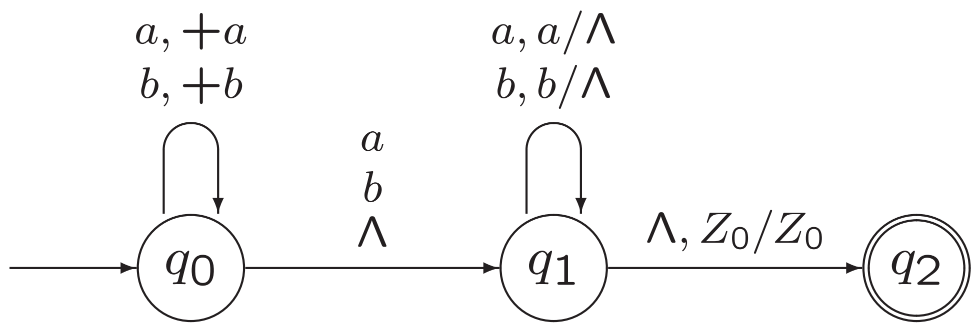
Example 5.3. A PDA Accepting the Language *AnBn*

$$AnBn = \{a^i b^i \mid i \geq 0\}$$



Example 5.7. A Pushdown Automaton Accepting *Pal*

$$Pal = \{y \in \{a,b\}^* \mid y = y^r\} = \{xx^r, xax^r, xbx^r \mid x \in \{a,b\}^*\}$$



Definition 5.10. A Deterministic Pushdown Automaton

A pushdown automaton $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ is *deterministic* if it satisfies both of the following conditions.

1. For every $q \in Q$, every $\sigma \in \Sigma \cup \{\Lambda\}$, and every $X \in \Gamma$, the set $\delta(q, \sigma, X)$ has at most one element.

2. For every $q \in Q$, every $\sigma \in \Sigma$, and every $X \in \Gamma$, the two sets $\delta(q, \sigma, X)$ and $\delta(q, \Lambda, X)$ cannot both be nonempty.

A language L is a *deterministic context-free language* (DCFL) if there is a deterministic PDA (DPDA) accepting L.

2. (in other words): For every $q \in Q$ and every $X \in \Gamma$, if $\delta(q, \Lambda, X)$ is not empty, then $\delta(q, \sigma, X)$ is empty for every $\sigma \in \Sigma$.

Example 5.11. A DPDA Accepting *Balanced*

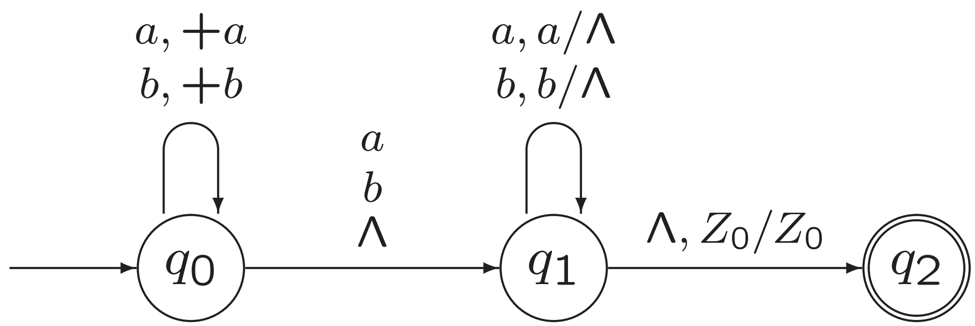
Balanced = {balanced strings of brackets [and]}

Example 5.13. Two DPDAs accepting *AEqB*

$AEqB = \{x \in \{a, b\}^* \mid n_a(x) = n_b(x)\}$

Example 5.7. A Pushdown Automaton Accepting *Pal*

$$Pal = \{y \in \{a,b\}^* \mid y = y^r\} = \{xx^r, xax^r, xbx^r \mid x \in \{a,b\}^*\}$$



Theorem 5.16.

The language *Pal* cannot be accepted by a deterministic pushdown automaton.

The proof of this result does not have to be known for the exam.

Exercise 5.16.

Show that if L is accepted by a PDA, then L is accepted by a PDA that never crashes (i.e., . . .).

Exercise 5.16.

Show that if L is accepted by a PDA, then L is accepted by a PDA that never crashes (i.e., for which the stack never empties and no configuration is reached from which there is no move defined).

A slide from lecture 8:

Stack in PDA contains symbols from certain alphabet.

Usual stack operations: pop, top, push

Extra possiblity: replace top element X by string α

$\alpha = \Lambda$	pop
$\alpha = X$	top
$\alpha = YX$	push
$\alpha = \beta X$	push*
$\alpha = \ldots$	

Top element X is required to do a move!

A slide from lecture 8:

Definition 5.2. Acceptance by a PDA

If $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ and $x \in \Sigma^*$,
the string x is accepted by M if

$$(q_0, x, Z_0) \vdash_M^* (q, \Lambda, \alpha)$$

for some $\alpha \in \Gamma^*$ and some $q \in A$.

A language $L \subseteq \Sigma^*$ is said to be accepted by M,
if L is precisely the set of strings accepted by M;
in this case, we write $L = L(M)$.

Sometimes a string accepted by M, or a language accepted by M, is said to be accepted *by final state*.

Exercise 5.17.

Show that if L is accepted by a PDA, then L is accepted by a PDA in which every move
$*$ either pops something from the stack (i.e., removes a stack symbol without putting anything else on the stack);
$*$ or pushes a single symbol onto the stack on top of the symbol that was previously on top;
$*$ or leaves the stack unchanged.

Hence, each action on the stack due to a move in the PDA has one of the following forms:
$*$ either X/Λ (with $X \in \Gamma$),
$*$ or X/YX (with $X, Y \in \Gamma$),
$*$ or X/X (with $X \in \Gamma$).

A slide from lecture 6:

Theorem 4.9.

If L_1 and L_2 are context-free languages over an alphabet Σ, then

$$L_1 \cup L_2, \quad L_1 L_2 \quad \text{and } L_1^*$$

are also CFLs.

Proof. . .

Exercise 5.19.

Suppose M_1 and M_2 are PDAs accepting L_1 and L_2, respectively. For both the languages L_1L_2 and L_1^*, describe a procedure for constructing a PDA accepting the language.

In each case, nondeterminism will be necessary. Be sure to say precisely how the stack of the new machine works; no relationship is assumed between the stack alphabets of M_1 and M_2.

Answer begins with:
Let $M_1 = (Q_1, \Sigma, \Gamma_1, q_{01}, Z_{01}, A_1, \delta_1)$
and let $M_2 = (Q_2, \Sigma, \Gamma_2, q_{02}, Z_{02}, A_2, \delta_2)$.