Fundamentele Informatica 3

voorjaar 2012

`http://www.liacs.nl/home/rvvliet/fi3/`

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

(werk-)college 2a, 13 februari 2012

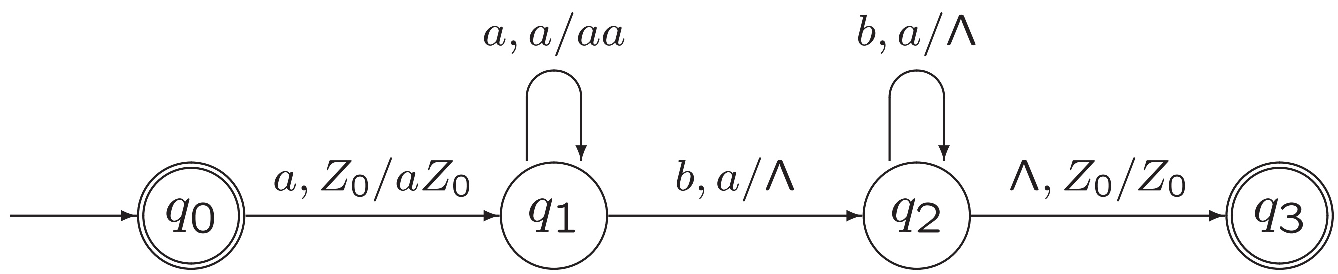
5. Pushdown Automata

5.1. Definitions and Examples

5.2. Deterministic Pushdown Automata

Example 5.3. A PDA Accepting the Language *AnBn*

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



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.

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

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

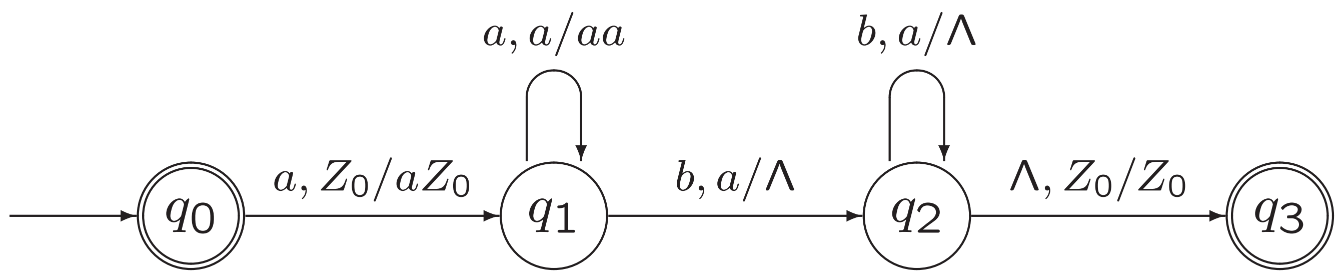
for some $\alpha \in \Gamma^*$ and some $q \in A$.

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*.

Example 5.3. A PDA Accepting the Language *AnBn*

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



Exercise 5.8.

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

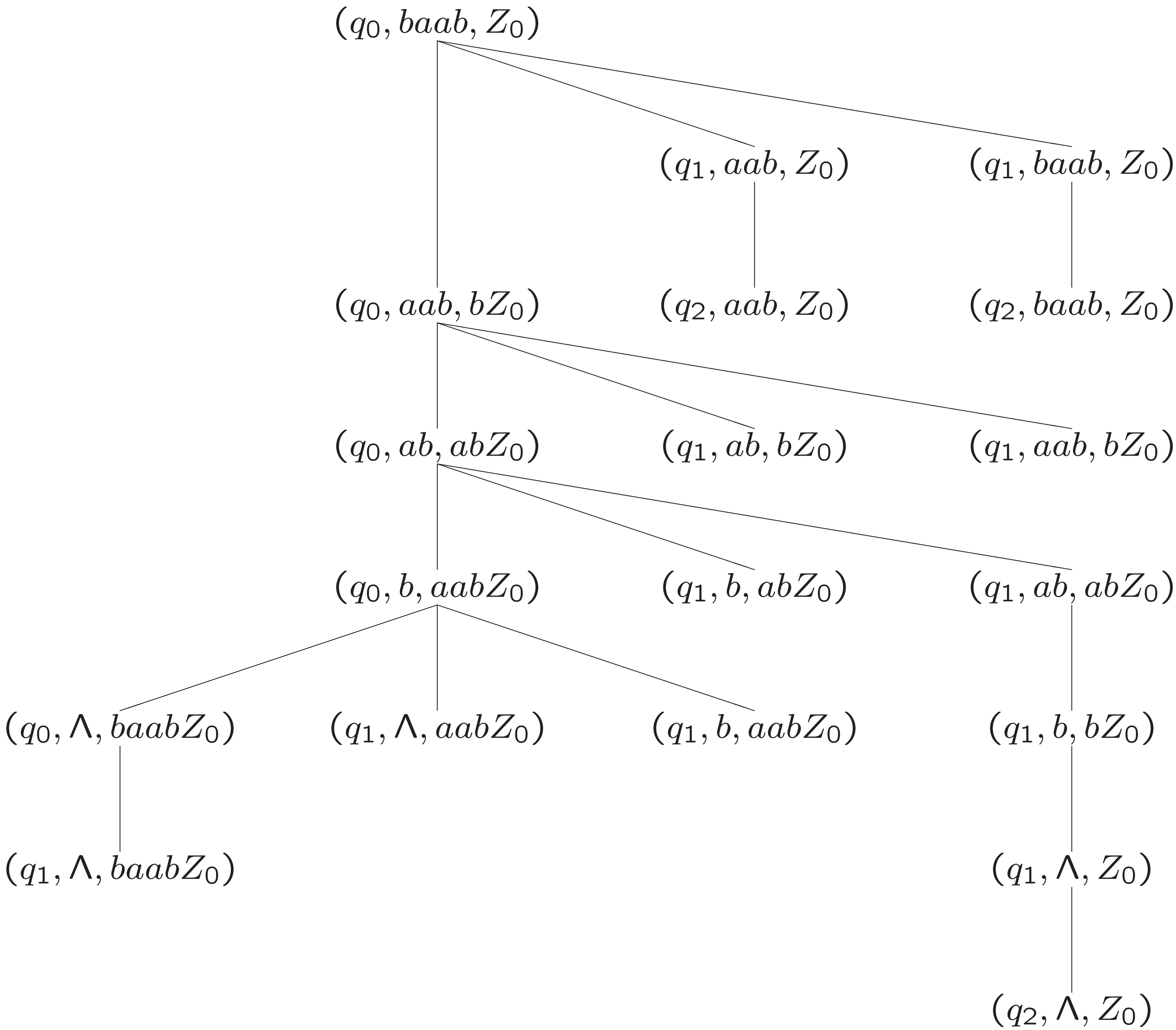
a'. $\{a^i b^{2i} \mid i \geq 0\}$

a. $\{a^i b^j \mid i \leq j \leq 2i\}$

a''. $\{a^i b^j \mid j \leq i \leq 2j\}$

Homework:
b. $\{x \in \{a,b\}^* \mid n_a(x) < n_b(x) < 2n_a(x)\}$

Example 5.7. A Pushdown Automaton Accepting *Pal*


$(q_0, baab, Z_0)$
(q_1, aab, Z_0)
$(q_1, baab, Z_0)$
(q_0, aab, bZ_0)
(q_2, aab, Z_0)
$(q_2, baab, Z_0)$
(q_0, ab, abZ_0)
(q_1, ab, bZ_0)
(q_1, aab, bZ_0)
$(q_0, b, aabZ_0)$
(q_1, b, abZ_0)
(q_1, ab, abZ_0)
$(q_0, \Lambda, baabZ_0)$
$(q_1, \Lambda, aabZ_0)$
$(q_1, b, aabZ_0)$
(q_1, b, bZ_0)
$(q_1, \Lambda, baabZ_0)$
(q_1, Λ, Z_0)
(q_2, Λ, Z_0)

5.2. Deterministic Pushdown Automata

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]}

Theorem 5.16.

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

Sketch of Proof.

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).

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.

Definition 2.20. Strings Distinguishable with Respect to L

If L is a language over the alphabet Σ, and x and y are strings in Σ^*, then x and y are *distinguishable with respect to* L, or L-*distinguishable*, if there is a string $z \in \Sigma^*$ such that either $xz \in L$ and $yz \notin L$, or $xz \notin L$ and $yz \in L$.

Example 2.27. For Every Pair x, y of Distinct Strings in $\{a, b\}^*$, x and y Are Distinghuishable with Respect to *Pal*.

Theorem 5.16.

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

Sketch of Proof.

Assume M is DPDA for *Pal*.

Let moves of M be of forms
$\delta(p, \sigma, X) = \{(q, \Lambda)\}$ or $\delta(p, \sigma, X) = \{(q, \alpha X)\}$

M reads every string $x \in \{a, b\}^*$ completely, with one path.

There exist different strings $r, s \in \{a, b\}^*$, such that for every $z \in \{a, b\}^*$, M treats rz and sz the same way.

There exist different strings $r, s \in \{a, b\}^*$, such that for every $z \in \{a, b\}^*$, M treats rz and sz the same way.

For a string $x \in \{a, b\}^*$, let y_x be a string such height of stack after xy_x is minimal.

Let α_x be stack after xy_x.
(state, top stack symbol) determines how suffix z is treated.

Infinitely many strings xy_x.

Finitely many pairs (q, X)

Different $r = uy_u$ and $s = vy_v$ arrive at same pair (q, X).

For any suffix z, rz and sz are treated the same:
$rz \in$ *Pal* $\iff$ $sz \in$ *Pal*.