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 8, 20 november 2015

4. Context-Free Languages
4.5. Simplified Forms and Normal Forms
5. Pushdown Automata
5.1. Definitions and Examples

4.5. Simplified Forms and Normal Forms

A slide from lecture 7:

Definition 4.29. Chomsky Normal Form

A context-free grammar is said to be in *Chomsky normal form* if every production is of one of these two types:

$$A \rightarrow BC \text{ (where } B \text{ and } C \text{ are variables)}$$
$$A \rightarrow \sigma \text{ (were } \sigma \text{ is a terminal symbol)}$$

A slide from lecture 7:

Arbitrary CFG may have

- productions $A \to \Lambda$
- productions $A \to B$ (unit productions)
- productions $A \to bc$, $A \to Bc$, $A \to bC$
- productions $A \to \alpha$ with $|\alpha| \geq 3$

A slide from lecture 7:

Converting a CFG to Chomsky Normal Form
Step 1

- Identify *nullable* variables
- Add productions in which nullable variables are removed from right hand side
- Delete Λ-productions

- Delete productions $A \to A$

We cannot generate Λ anymore

Example.

$S \to aSb \mid aBb \quad B \to bB \mid \Lambda$

$S \to SaS \mid B \quad B \to bB \mid \Lambda$

A slide from lecture 7:

Converting a CFG to Chomsky Normal Form
Step 2

- Identify A-*derivable* variables

- For every A-derivable variable B
  and nonunit production $B \to \alpha$, add production $A \to \alpha$

- Delete unit productions

Example.

$S \to aSb \mid B \quad B \to bB \mid b \mid A \quad A \to aBS \mid a$

Arbitrary CFG may have

- productions $A \to \Lambda$
- productions $A \to B$ (unit productions)
- productions $A \to bc$, $A \to Bc$, $A \to bC$
- productions $A \to \alpha$ with $|\alpha| \geq 3$

Converting a CFG to Chomsky Normal Form
Step 3

- Add productions $X_a \to a$

- In every production $A \to \alpha$ with $|\alpha| \geq 2$, replace terminals a by corresponding non-terminals X_a

Example.

$S \to TB \quad T \to aTTb \mid ab \quad B \to bB \mid b$

Arbitrary CFG may have

- productions $A \to \Lambda$
- productions $A \to B$ (unit productions)
- productions $A \to bc$, $A \to Bc$, $A \to bC$
- productions $A \to \alpha$ with $|\alpha| \geq 3$

Converting a CFG to Chomsky Normal Form
Step 4

- Split productions whose right hand sides are too long

Example.

$S \to TB \quad T \to X_aTTX_b \mid X_aX_b \quad B \to X_bB \mid b$
$X_a \to a \quad X_b \to b$

Theorem 4.30.

For every context-free grammar G,
there is another CFG G_1 in Chomsky normal form
such that $L(G_1) = L(G) - \{\Lambda\}$.

What if $\Lambda \notin L(G)$?

Example 4.31. Converting a CFG to Chomsky Normal Form

Let G be CFG with productions

$$\begin{aligned} S &\to TU \mid V \\ T &\to aTb \mid \Lambda \\ U &\to cU \mid \Lambda \\ V &\to aVc \mid W \\ W &\to bW \mid \Lambda \end{aligned}$$

A slide from lecture 7:

Definition 4.13. Regular Grammars.

A context-free grammar $G = (V, \Sigma, S, P)$ is *regular* if every production is of the form

$$A \to \sigma B \quad \text{or} \quad A \to \Lambda,$$

where $A, B \in V$ and $\sigma \in \Sigma$.

A slide from lecture 7:

Definition 4.29. Chomsky Normal Form

A context-free grammar is said to be in *Chomsky normal form* if every production is of one of these two types:

$$A \rightarrow BC \text{ (where } B \text{ and } C \text{ are variables)}$$
$$A \rightarrow \sigma \text{ (were } \sigma \text{ is a terminal symbol)}$$

5. Pushdown Automata

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

just like FA, PDA accepts strings / language

just like FA, PDA has states

just like FA, PDA reads input one letter at a time

unlike FA, PDA has auxiliary memory: a stack

unlike FA, by default PDA is nondeterministic

unlike FA, by default Λ-transitions are allowed in PDA

Why a stack?

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

$$\mathit{SimplePal} = \{xcx^r \mid x \in \{a, b\}^*\}$$

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!

Example 5.3. PDAs Accepting the Languages *AnBn* and *SimplePal*

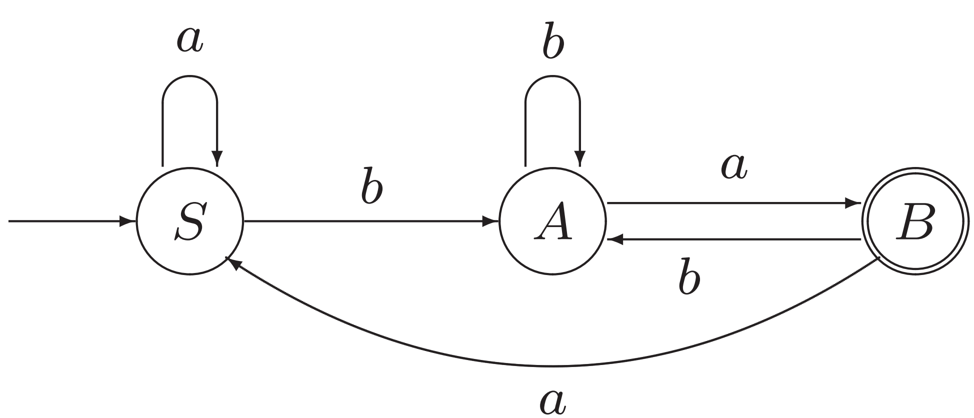
$$\mathit{AnBn} = \{a^i b^i \mid i \geq 0\}$$

$$\mathit{SimplePal} = \{xcx^r \mid x \in \{a, b\}^*\}$$

A slide from lecture 7:

In general: construction of a CFG from a finite automaton.

Example: an FA accepting $\{a,b\}^*\{ba\}$



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 . . . to . . .

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.

Example 5.3. A PDA Accepting the Language *AnBn*

Transition table:

Move Number	State p	Input σ	Stack Symbol X	Move(s) $\delta(p, \sigma, X)$
1	q_0	a	Z_0	(q_1, aZ_0)
2	q_1	a	a	(q_1, aa)
3	q_1	b	a	(q_2, Λ)
4	q_2	b	a	(q_2, Λ)
5	q_2	Λ	Z_0	(q_3, Z_0)
(all other combinations)				none

Notation

configuration for certain input: (q, x, α)

$(p, x, \alpha) \vdash_M (q, y, \beta)$

$(p, x, \alpha) \vdash_M^n (q, y, \beta)$ $(p, x, \alpha) \vdash_M^* (q, y, \beta)$

$(p, x, \alpha) \vdash (q, y, \beta)$ $(p, x, \alpha) \vdash^n (q, y, \beta)$ $(p, x, \alpha) \vdash^* (q, y, \beta)$

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*

Transition table:

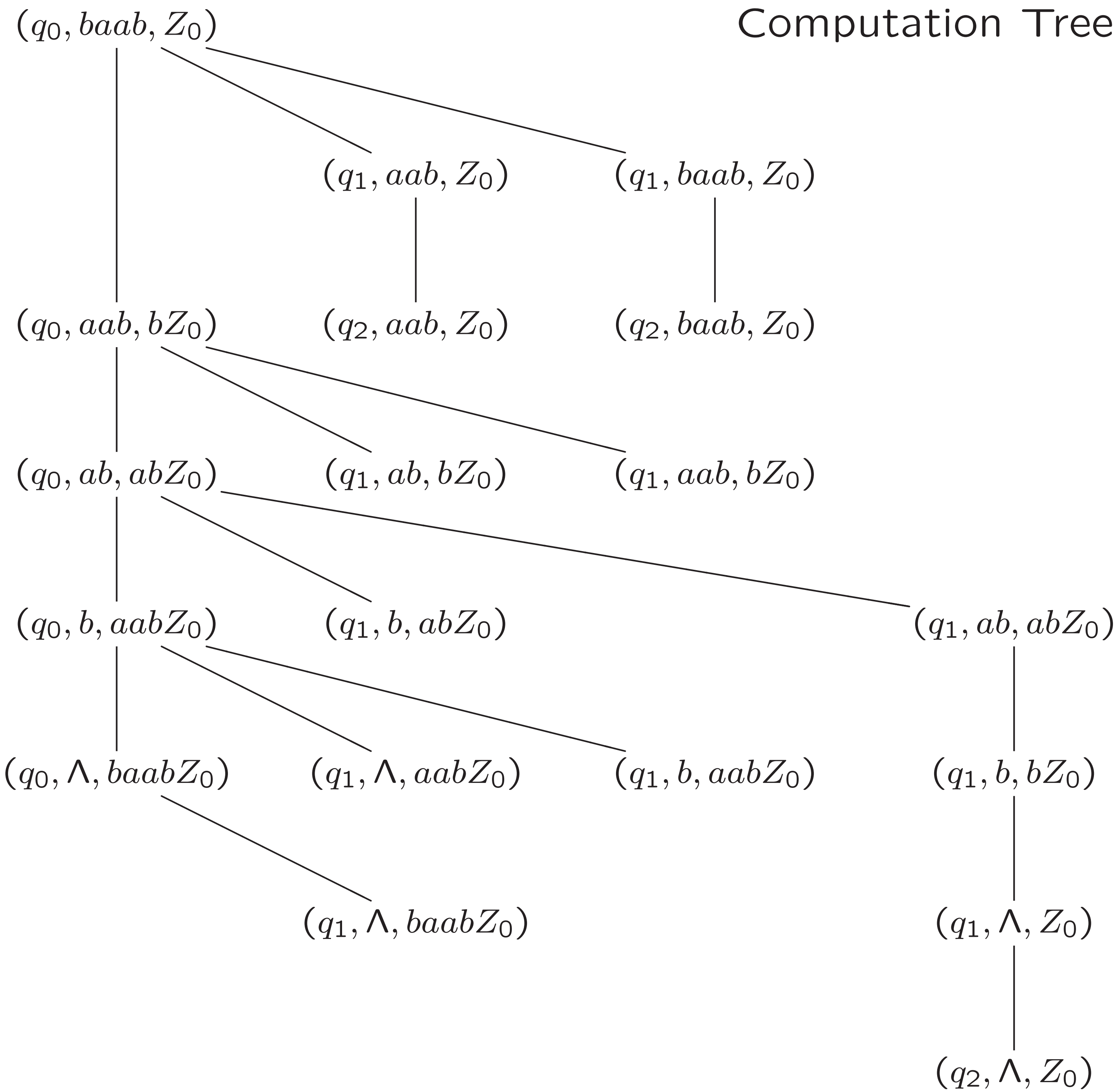
Move Number	State p	Input σ	Stack Symbol X	Move(s) $\delta(p, \sigma, X)$
1	q_0	a	Z_0	(q_1, aZ_0)
2	q_1	a	a	(q_1, aa)
3	q_1	b	a	(q_2, Λ)
4	q_2	b	a	(q_2, Λ)
5	q_2	Λ	Z_0	(q_3, Z_0)
(all other combinations)				none

Computation for $aabb$...

Example 5.7. A Pushdown Automaton Accepting *Pal*

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

Computation Tree



Dinsdag 24 november

Zowel hoorcollege als werkcollege in 405