Fundamentele Informatica 3

voorjaar 2014

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

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

college 1, 3 februari 2014

Herhaling onderwerpen FI2

7. Turing Machines

Practische Informatie

- hoorcollege: maandag 3 feb - 19 mei, zaal 403, 13:45–15:30
werkcollege: dinsdag 4 feb - 13 mei, zaal 403, 13:45–15:30

- boek: John C. Martin, Introduction to Languages and the Theory of Computation, 4th edition

- tentamens: vrijdag 6 juni 2014, 14:00–17:00
dinsdag 8 juli 2014, 14:00–17:00

Practische Informatie

- Drie huiswerkopgaven (individueel)
Niet verplicht, maar . . .
eindcijfer $=$ tentamencijfer $+$ cijferhuiswerkopgaven
waarbij $5.0 \leq$ tentamencijfer $<=$ 10.0
cijferhuiswerkopgaven ≤ 1.0
eindcijfer ≤ 10.0

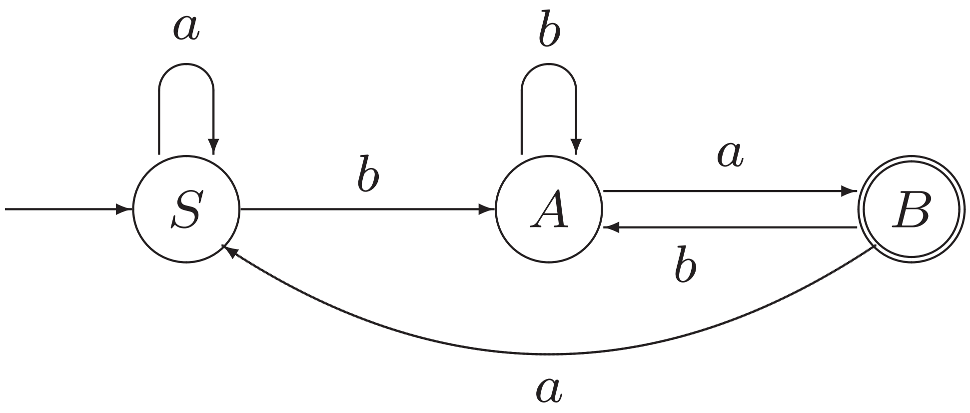
- 6 EC

- Vorig jaar al inhoud verschoven t.o.v. voorjaar 2012:
hoofdstuk 7–10 ipv hoofdstuk 5–9

Fundamentele Informatica 2

2.1. Finite Automata

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



Fundamentele Informatica 2

2.1. Finite Automata

2.4. The Pumping Lemma
AnBn $= \{a^i b^i \mid i \geq 0\}$, *SimplePal* $= \{xcx^r \mid x \in \{a,b\}^*\}$

3.1. Regular Languages and Regular Expressions
$\{a,b\}^*\{ba\}$ vs. $(a+b)^*ba$

3.2. Nondeterministic Finite Automata

3.3. The Nondeterminism in an NFA Can Be Eliminated

3.4/3.5. Kleene's Theorem

Fundamentele Informatica 2

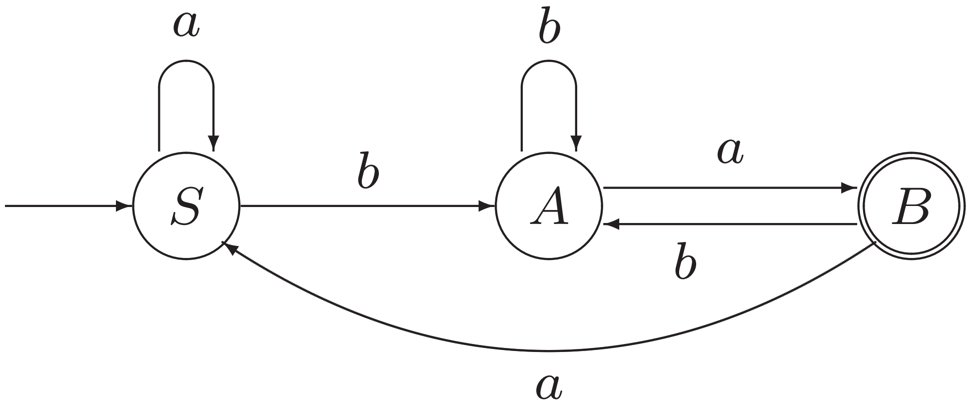
4.2. Context-Free Grammars

$S \rightarrow aSa \mid bSb \mid c$

4.3. Regular Languages and Regular Grammars

$S \rightarrow aS \mid bA \quad A \rightarrow bA \mid aB \quad B \rightarrow bA \mid aS \mid \Lambda$

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



Fundamentele Informatica 2

4.2. Context-Free Grammars

$S \to aSa \mid bSb \mid c$

4.3. Regular Languages and Regular Grammars

$S \to aS \mid bA \quad A \to bA \mid aB \quad B \to bA \mid aS \mid \Lambda$

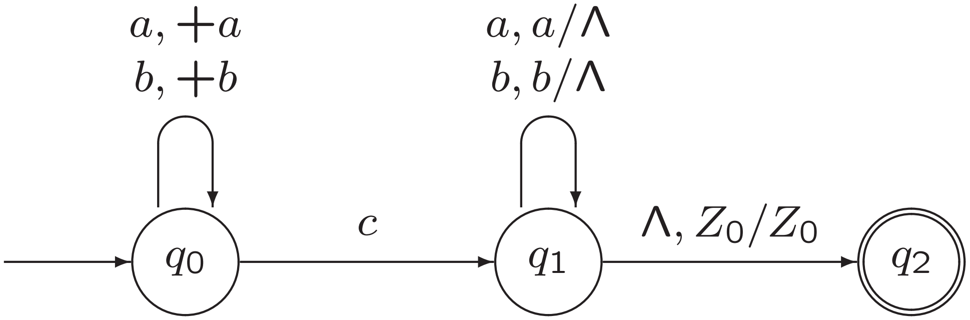
4.4. Derivation Trees

4.5. Simplified Forms and Normal Forms

Fundamentele Informatica 2

5.1. Definitions and Examples (of Pushdown Automata)

Example 5.3. A Pushdown Automaton Accepting *SimplePal*

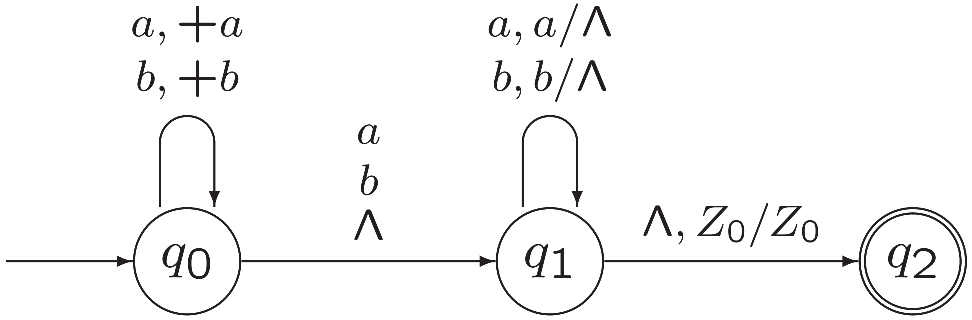


Fundamentele Informatica 2

5.1. Definitions and Examples (of Pushdown Automata)

Example 5.7. A Pushdown Automaton Accepting *Pal*

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



Fundamentele Informatica 2

5.1. Definitions and Examples (of Pushdown Automata)

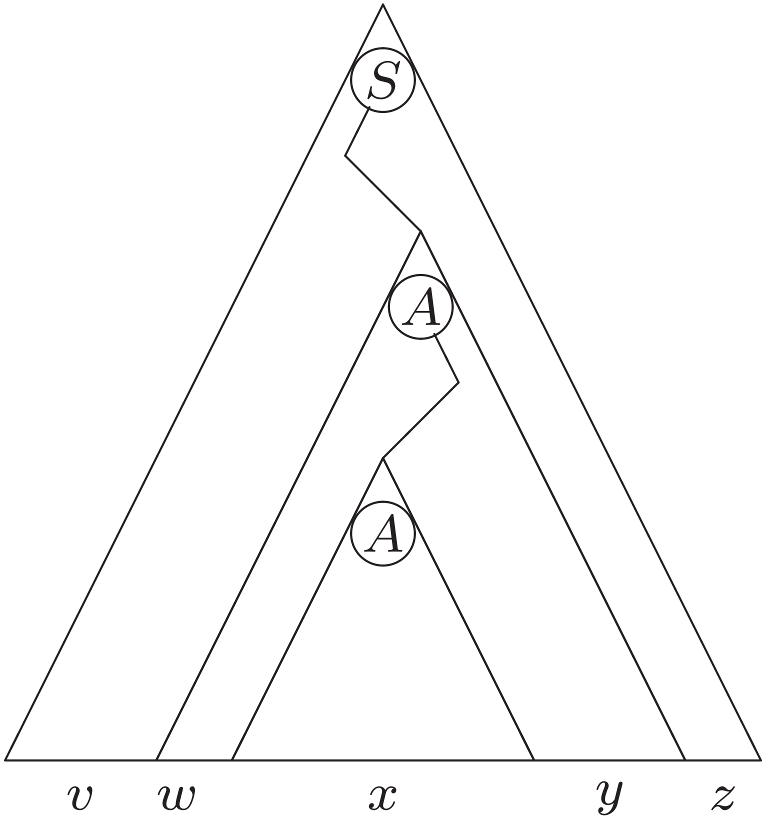
5.2. Deterministic Pushdown Automata

5.3. A PDA from a Given CFG

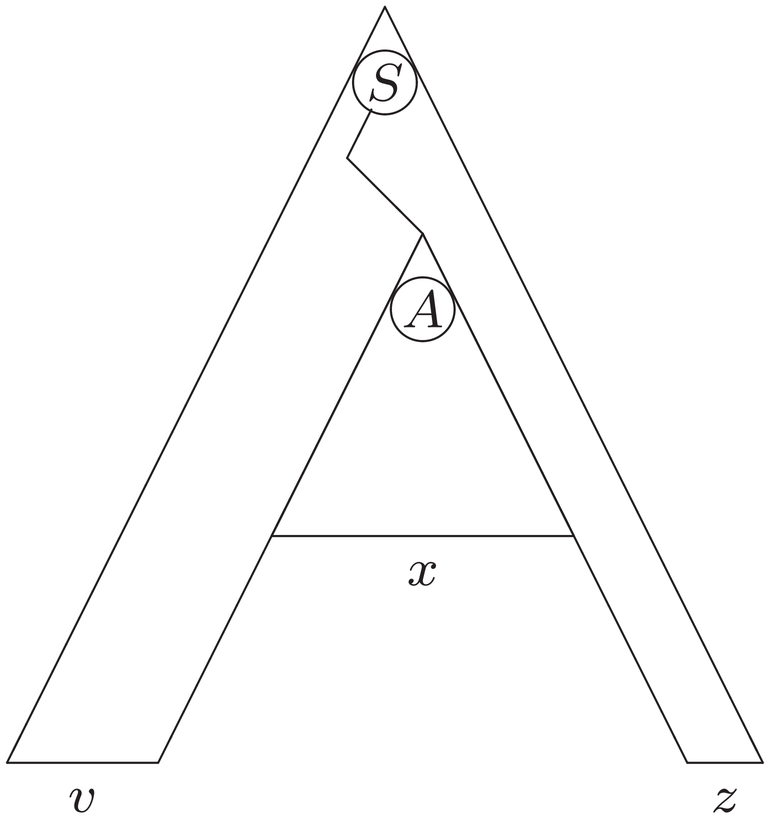
5.4. A CFG from a Given PDA

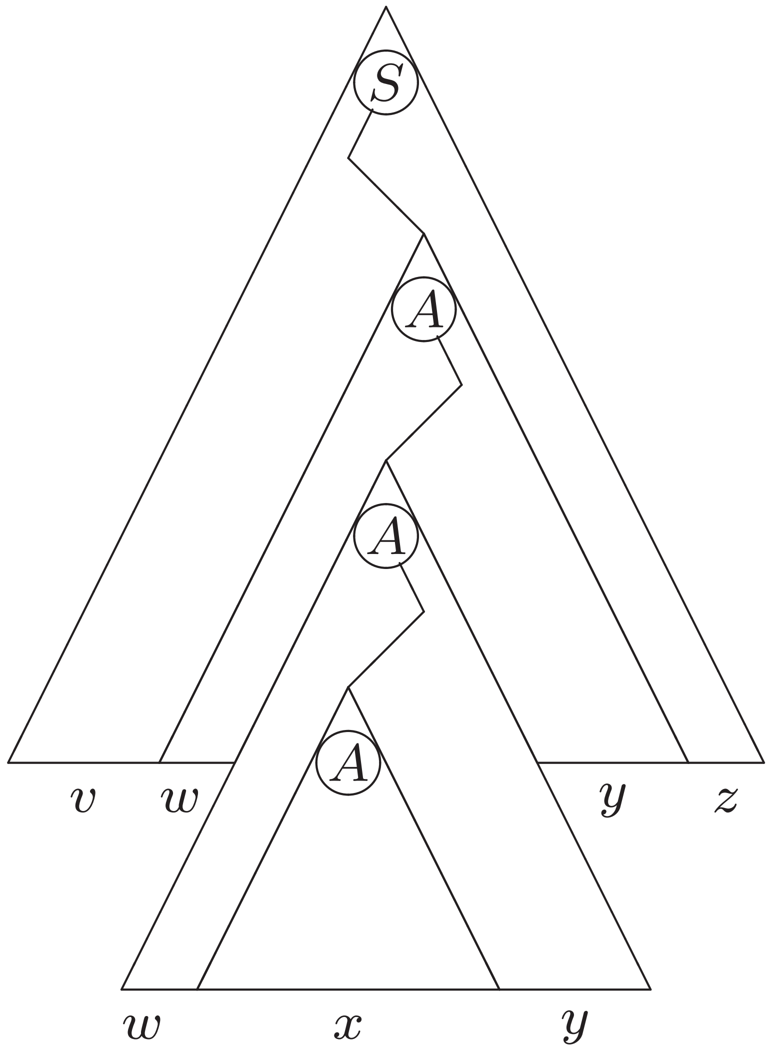
6.1. The Pumping Lemma for Context-Free Languages

FI2: Pumping Lemma for CFLs



FI2: Pumping Lemma for CFLs





Fundamentele Informatica 2

5.1. Definitions and Examples (of Pushdown Automata)

5.2. Deterministic Pushdown Automata

5.3. A PDA from a Given CFG

5.4. A CFG from a Given PDA

6.1. The Pumping Lemma for Context-Free Languages
$\mathit{AnBnCn} = \{a^i b^i c^i \mid i \geq 0\}$, $L = \{xcx \mid x \in \{a,b\}^*\}$

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

7. Turing Machines

7.1. A General Model of Computation

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

$L = \{xcx \mid x \in \{a, b\}^*\}$

Assumptions about a human computer working with a pencil and paper:

1. The only things written on the paper are symbols from some fixed finite alphabet;

2. Each step taken by the computer depends only on the symbol he is currently examining and on his “state of mind” at the time;

3. Although his state of mind may change as a result of his examining different symbols, only a finite number of distinct states of mind are possible.

Actions of a human computer on a sheet of paper:

1. Examining an individual symbol on the paper;

2. Erasing a symbol or replacing it by another;

3. Transferring attention from one symbol to another nearby symbol.

Turing machine

Turing machine has a finite alphabet of symbols.
    (actually two alphabets. . .)
Turing machine has a finite number of states.

Turing machine has a *tape*
    for reading input,
    as workspace,
    and for writing output (if applicable).

Tape is linear, instead of 2-dimensional.
Tape has a left end and is potentially infinite to the right.
Tape is marked off into squares, each of which can hold one symbol.
Tape head is centered on one square of the tape for reading and writing.

A move of a Turing machine consists of:

1. Changing from the current state to another, possibly different state;

2. Replacing the symbol in the current square by another, possibly different symbol;

3. Leaving the tape head on the current square, or moving it one square to the right, or moving it one square to the left if it is not already on the leftmost square.

Just like FA and PDA, Turing machine

- may be used to accept a language
- has a finite number of states

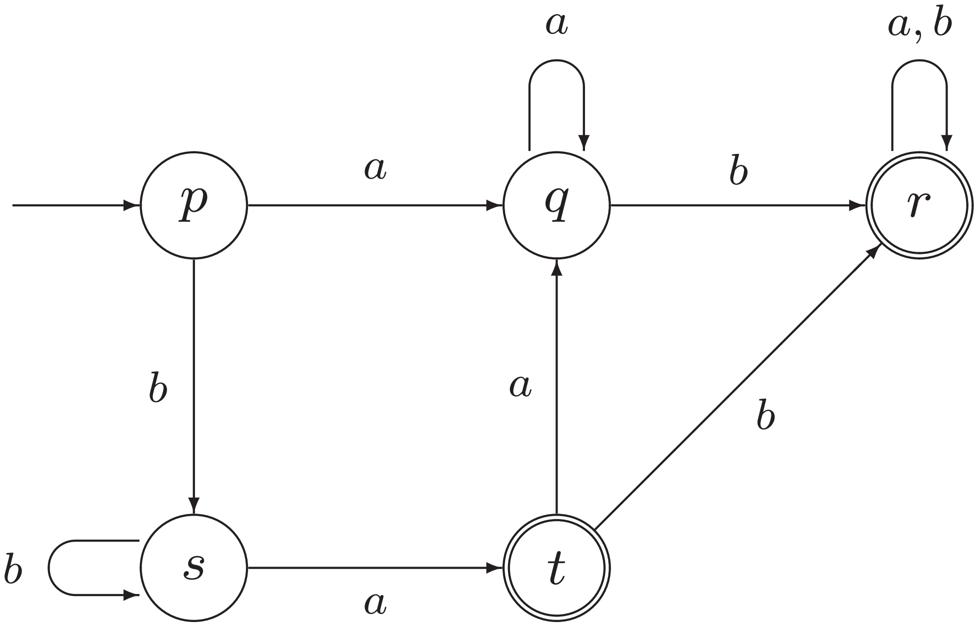
Unlike FA and PDA, Turing machine

- may also be used to compute a function *
- is not restricted to reading input left-to-right *
- does not have to read all input *
- does not have a set of accepting states, but has two *halt* states: one for acceptance and one for rejection (in case of computing a function, . . .)
- may decide not to halt

* = just like human computer

Example.

An FA Accepting $L = \ldots$

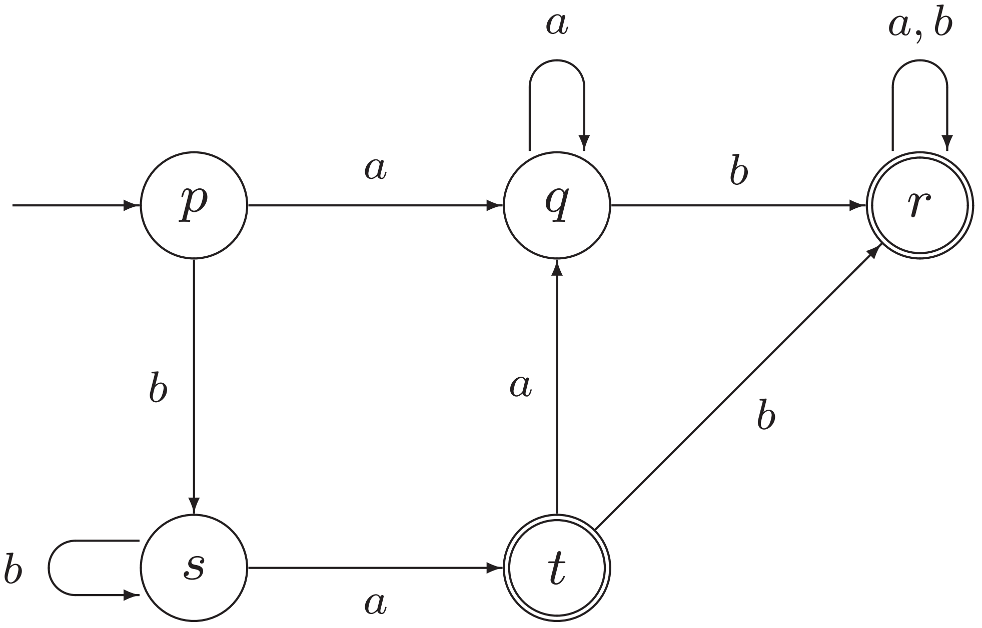


7.2. Turing Machines as Language Acceptors

Example 7.3. A TM Accepting a Regular Language

$L = \{a, b\}^*\{ab\}\{a, b\}^* \cup \{a, b\}^*\{ba\}$

First a finite automaton:



Example 7.3. A TM Accepting a Regular Language

$L = \{a,b\}^*\{ab\}\{a,b\}^* \cup \{a,b\}^*\{ba\}$

First a finite automaton, then a Turing machine

This conversion works in general for FAs.
As a result,
- only moves to the right,
- no modifications of symbols on tape,
- no moves to the reject state, but . . .

In this case,
- we could modify TM, so that it does not always read entire input.

Example 7.5. A TM Accepting $XX = \{xx \mid x \in \{a,b\}^*\}$