Fundamentele Informatica 3

voorjaar 2018

`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi3/`

Rudy van Vliet
kamer 140 Snellius, tel. 071-527 2876
rvvliet(at)liacs(dot)nl

college 1, 5 februari 2018

Herhaling onderwerpen FI2

7. Turing Machines

Oud FI3 vs nieuw FI3

- keuze vs verplicht
- 6 EC vs 3 EC
- vier onderwerpen vs drie onderwerpen...

Practische Informatie

- hoorcollege: maandag, 11.00–12.45 (zaal B03)

werkcollege (Lieuwe Vinkhuijzen):
    maandag, 13.30–15.15 (zaal 312)
    van 5 februari – 26 maart 2018

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

- hoofdstuk 7–9

Practische Informatie

• tentamens: maandag 16 april 2018, 14.00–17.00
maandag 28 mei 2018, 14.00–17.00

• Eén huiswerkopgave (individueel)
Niet verplicht, maar . . .
eindcijfer = tentamencijfer + cijferhuiswerkopgave
cijferhuiswerkopgave ≤ 0.4
eindcijfer ≤ 10.0

Practische Informatie

Website

`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi3/`

- slides, N.B. . . .
- overzicht van behandelde stof
- antwoorden van bepaalde opgaven
- huiswerkopgave
- errata

Overview

7. Turing machines

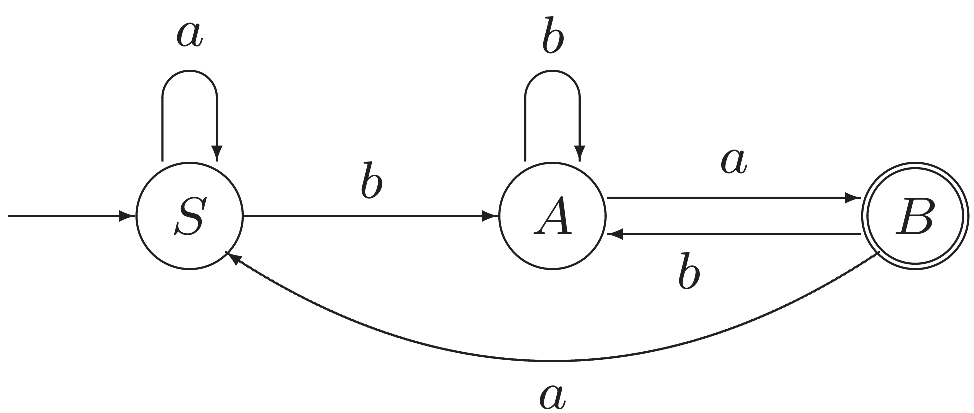
8. Recursive(ly enumerable) languages / general grammars

9. Undecidable problems

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

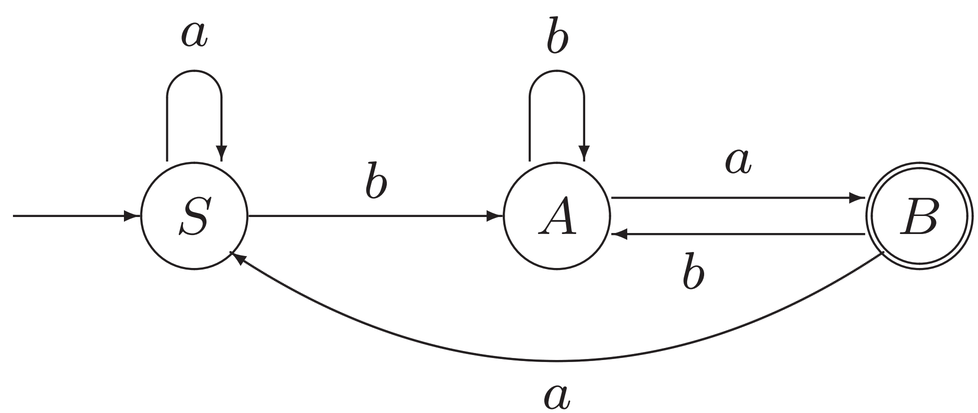
4.2. Context-Free Grammars

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

4.3. Regular Languages and Regular Grammars

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

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

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

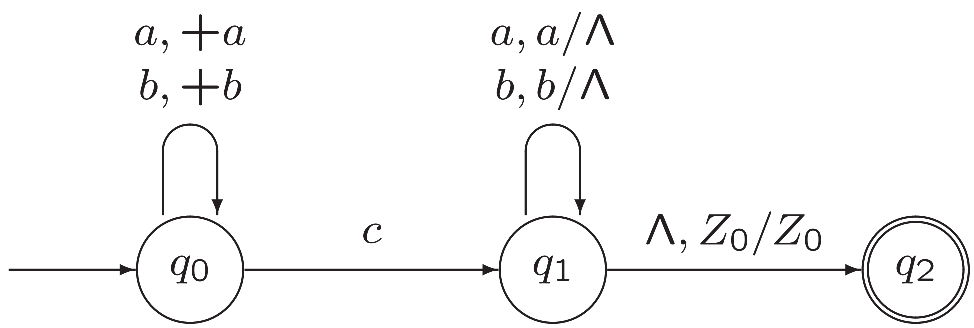
4.4. Derivation Trees

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*

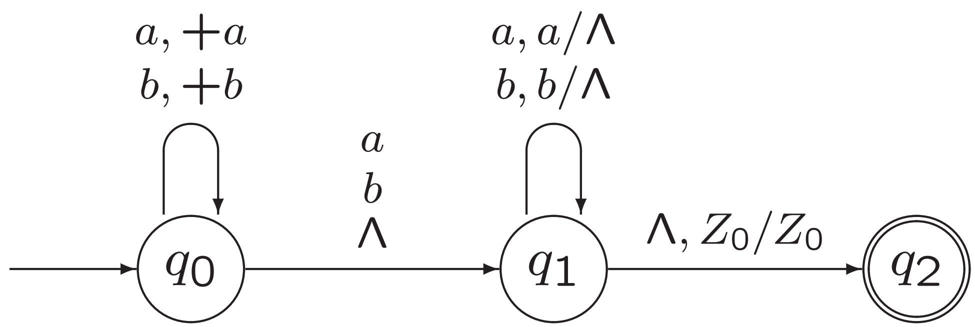


Fundamentele Informatica 2

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)

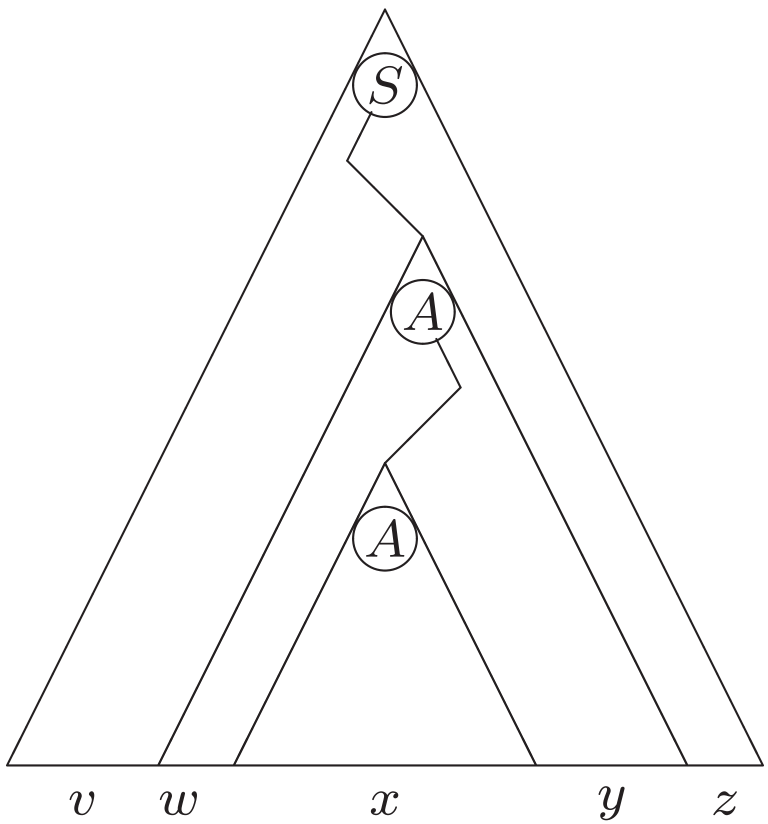
5.2. Deterministic Pushdown Automata

5.3. A PDA from a Given CFG

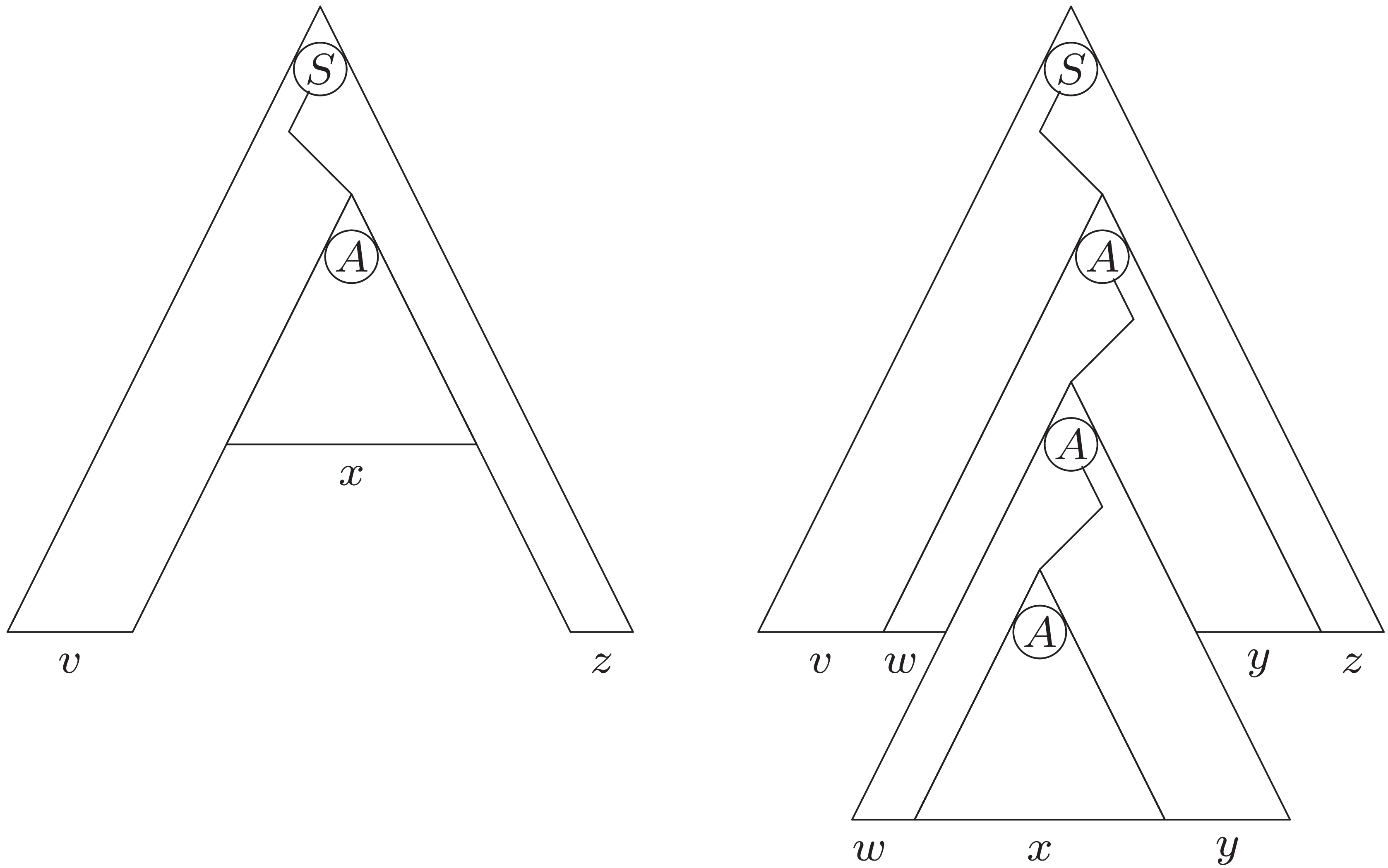
5.4. A CFG from a Given PDA

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
AnBnCn $= \{a^i b^i c^i \mid i \geq 0\}$, $L = \{xcx \mid x \in \{a,b\}^*\}$

7. Turing Machines

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

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

Just like FA, but unlike PDA

- by default TM is deterministic

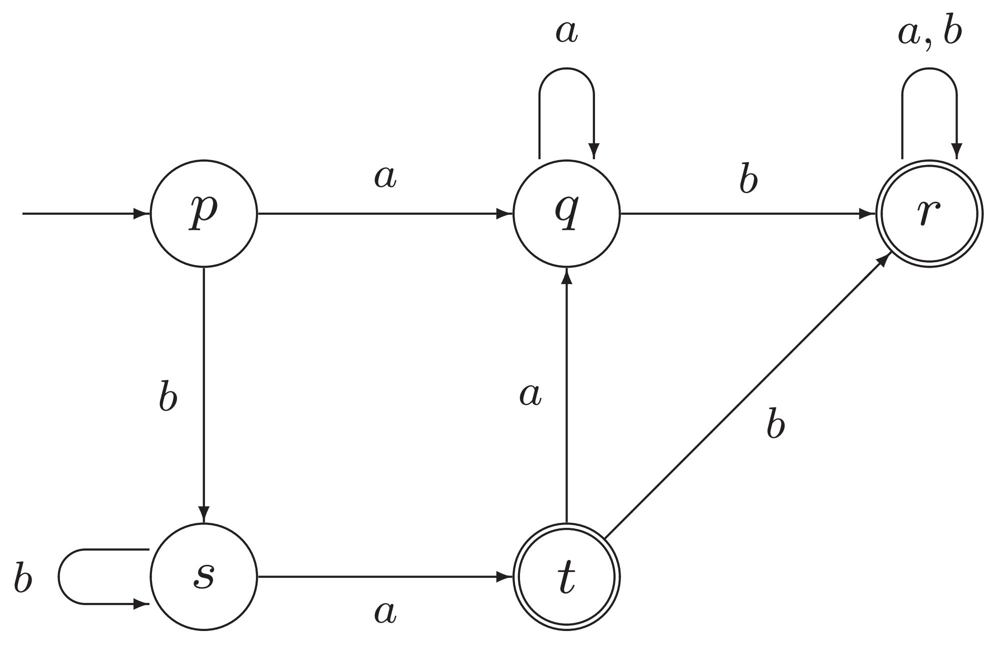
Unlike FA and PDA, Turing machine

- 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, . . .)
- might not decide to halt

* = just like human computer

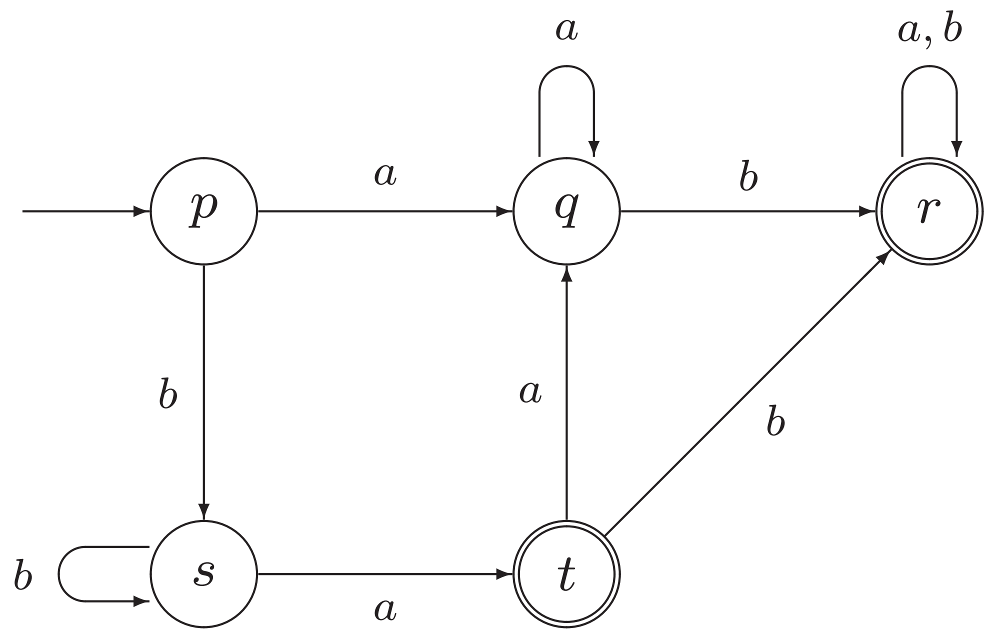
Example.

An FA Accepting $L = \ldots$



Example.

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



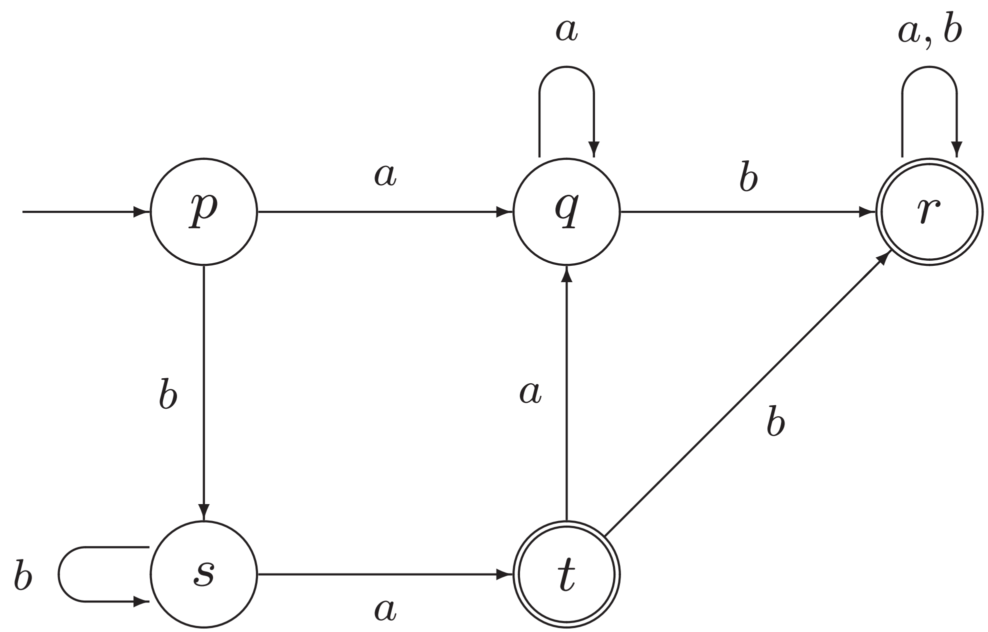
Why two accepting states?

7.2. Turing Machines as Language Acceptors

Example 7.3. A TM Accepting a Regular Language

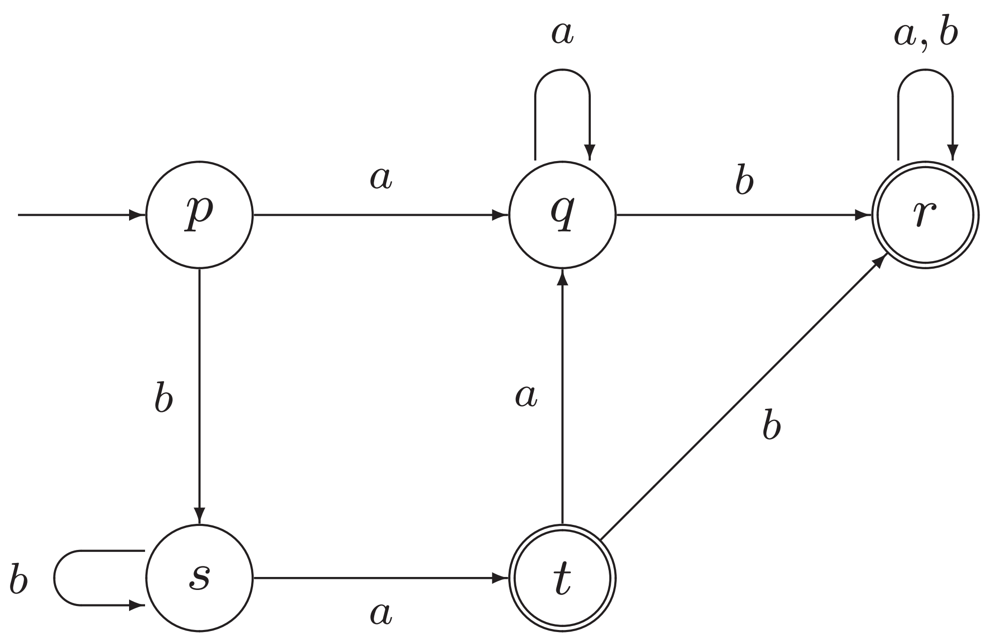
$L = \{a,b\}^*\{ab\}\{a,b\}^* \cup \{a,b\}^*\{ba\}$

First a finite automaton:



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:



Δ vs Λ
Λ vs $\{\Lambda\}$ vs $\emptyset$

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 $XX = \{xx \mid x \in \{a,b\}^*\}$

$$\underline{\Delta}aabaab$$

$$\ldots$$

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

$$\begin{array}{l}
\underline{\Delta}\, a\ a\ b\ a\ a\ b \\
\Delta A a\ b\ a\ a\ b \\
\Delta A a\ b\ a\ a B \\
\Delta A A b\ a\ a B \\
\Delta A A b\ a A B \\
\Delta A A B a A B \\
\Delta A A B A A B \\
\quad \ldots
\end{array}$$

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

$$
\begin{array}{l}
\underline{\Delta}\, a\ a\ b\ a\ a\ b \\
\Delta A a\ b\ a\ a\ b \\
\Delta A a\ b\ a\ a B \\
\Delta A A\ b\ a\ a B \\
\Delta A A\ b\ a A B \\
\Delta A A B a A B \\
\Delta A A B A A B \\
\Delta a\ a\ b A A B \\
\Delta A a\ b A A B \\
\quad\ldots
\end{array}
$$

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

$$\underline{\Delta}\, a\, a\, b\, a\, a\, b$$
$$\Delta A\, a\, b\, a\, a\, b$$
$$\Delta A\, a\, b\, a\, a\, B$$
$$\Delta A A\, b\, a\, a\, B$$
$$\Delta A A\, b\, a\, A\, B$$
$$\Delta A A B\, a\, A\, B$$
$$\Delta A A B\, A\, A\, B$$
$$\Delta\, a\, a\, b\, A\, A\, B$$
$$\Delta A\, a\, b\, A\, A\, B$$
$$\Delta A\, a\, b\, \Delta\, A\, B$$
$$\Delta A A\, b\, \Delta\, A\, B$$
$$\Delta A A\, b\, \Delta \Delta\, B$$
$$\Delta A A B \Delta \Delta\, B$$
$$\Delta A A B \Delta \Delta \Delta$$

Exercise 7.4.

For each of the following languages, draw a transition diagram for a Turing machine that accepts that language.

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

b. $\{a^i b^j \mid i < j\}$

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

d. $\{a^i b^j \mid i \neq j\}$

Exercise 7.4.

For each of the following languages, draw a transition diagram for a Turing machine that accepts that language.

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

We could also use the portion of the tape to the right of the input, to simulate the stack of a deterministic pushdown automaton (works for any deterministic PDA!)

Exercise.

Draw a transition diagram for a Turing machine accepting

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

Exercise 7.5.

For each part below, draw a transition diagram for a TM that accepts $AEqB = \{x \in \{a,b\}^* \mid n_a(x) = n_b(x)\}$ by using the approach that is described.

a. Search the string left-to-right for an a. As soon as one is found, replace it by X, return to the left end, and search for b. Replace it by X. Return to the left end and repeat these steps until one of the two searches is unsuccessful.

b. Begin at the left and search for either an a or a b. When one is found, replace it by X and continue to the right searching for the opposite symbol. When it is found, replace it by X and move back to the left end. Repeat these steps until one of the two searches is unsuccessful.

Exercise 7.6.

Draw a transition diagram for a TM accepting *Pal*, the language of palindromes over $\{a, b\}$, using the following approach.
Look at the leftmost symbol of the current string, erase it but remember it, move to the rightmost symbol and see if it matches the one on the left; if so, erase it and go back to the left end of the remaining string.
Repeat these steps until either the symbols are exhausted or the two symbols on the ends don't match.

Exercise 7.9.

Describe the language (a subset of $\{1\}^*$) accepted by the TM in Figure 7.37 (on the blackboard).