Introduction to Theoretical Computer Science

Oct 25

Markus Bläser

Saarland University

Organization

CMS:

- `cms.sic.saarland/ti2324`
- Registration until Saturday, October 28th, 23:59

Lectures:

- Wednesdays 14:15–16:00, Fridays 8:30–10:00
- Lectures only in person, but will be recorded
- Recordings will be available a few days later
- Lecture notes and slides in the CMS

Organization (2)

Tutorials:

- A new exercise sheet every friday
- Submitted until the following friday at 8:00 in the CMS
- Submissions can be in either English or German
- Groups of up to 3 people, groups have to be formed until November 1st in the CMS
- You can search for group members in the CMS-Forum
- Weekly tutorials tuesdays in English or German
- Select time and language preferences until October 28th in the CMS
- Weekly Office Hours thursdays 14:15–16:00

Organization (3)

Exams:

- Admission: 50% of all regular points on the exercise sheets
- Dates: February 20th and March 20th, 2024.

Words (Chapter A.3)

$\Sigma = \{a, b\}$

$w : \{1, 2, 3\} \to \Sigma$

- Finite set Σ (*alphabet*).
- Elements in Σ are called *symbols* or *letters*.
- A (finite) *word* over Σ is a mapping $w : \{1, \ldots, \ell\} \to \Sigma$.
- Words are also called *strings*.
- ℓ is the *length* of w.
- The *empty word* ε has length 0.

w	1	2	3
	a	b	a

$\varepsilon : \emptyset \to \Sigma$

$w_2 = w(2) = b$

Notation

$$\begin{array}{ccc} a & b & a \\ \shortparallel & \shortparallel & \shortparallel \\ w_1 & w_2 & w_3 \end{array}$$

Words (2)

We can write a word $w : \{1, \ldots, \ell\} \to \Sigma$ as

- a table of values,
- in a compact way as $w(1)w(2)\ldots w(\ell)$,
- even more compact as $w_1 w_2 \ldots w_\ell$.

Words (3)

$\Sigma = \{a, b\}$ $\Sigma^2 = \{aa, ab, ba, bb\}$

$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots$

- Σ^n are all words of length n.
- $\Sigma^* := \bigcup_{n=0}^{\infty} \Sigma^n$ are all finite words.
- $\Sigma^0 = \{\varepsilon\}$
- We identify Σ with Σ^1.

$w(1) = a$

Σ Σ^1

finite set of symbols

set of functions $\{1\} \to \Sigma$

Words (4)

a b a a a $\longrightarrow$ a b a a a

$\begin{array}{c|ccc} w & 1 & 2 & 3 \\ \hline & a & b & a \end{array}$ $\begin{array}{c|cc} x & 1 & 2 \\ \hline & a & a \end{array}$

$\begin{array}{c|ccccc} wx & 1 & 2 & 3 & 4 & 5 \\ \hline & a & b & a & a & a \end{array}$

Concatenation:

- $w : \{1, \ldots, \ell\} \to \Sigma$, $x : \{1, \ldots, k\} \to \Sigma$.
- The concatenation of w and x is a mapping

$$\{1, \ldots, \ell + k\} \to \Sigma$$

$$i \mapsto \begin{cases} w(i) & \text{if } 1 \leq i \leq \ell \\ x(i - \ell) & \text{if } \ell + 1 \leq i \leq \ell + k. \end{cases}$$

- We denote the new word with wx.
- w^i denotes the i-fold concatenation of w with itself.
- $\varepsilon w = w \varepsilon = w$.

$\varepsilon : \emptyset \to \Sigma$

“ $\{1, \ldots, 0\}$

Words (5)

$x = aa$

$y = abaab$

$u = ab$

$v = b$

Subwords:

- x is a *subword* (or *substring*) of y if there are words u and v with $y = uxv$.
- If $u = \varepsilon$, then x is called a *prefix*.
- If $v = \varepsilon$, then x is a *suffix*.

$uxv = ab \vdots aa \vdots b$

$(ux)v$ $u(xv)$

x is a prefix of $aabaa$

Chapter 1: Finte automata

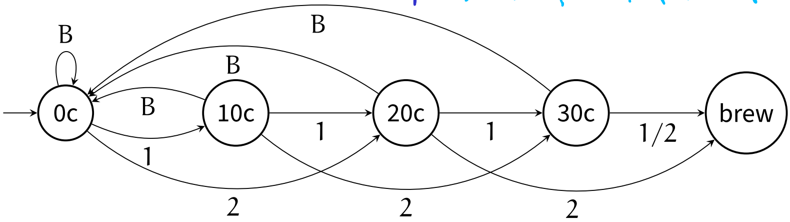
Finite automata

Example: coffee vending machine

- Accepts 10 cent and 20 cent coins.
- If 40 cents or more are inserted, the machine brews a coffee.
- The machine does not give change.
- As long as less than 40 Cent are inserted, one can press the "Money back" button.

.

1	2	3	4	5	6	7	8	9	10
1	2	B	1	2	B	1	1	1	B



"set of actions": words over $\{1, 2, B\}$

BB . . . . B . . . . B — no coffee

1 2 B 1 2 B 1 1 1 B — no coffee

1 1 1 1 — coffee

1 2 1 — coffee

1 2 2 — " "

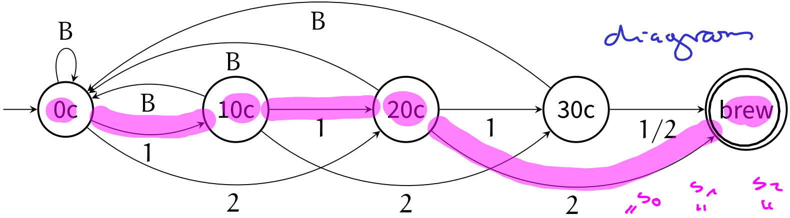
22 — " "

$w = 112$ ← accepted by COFFEE

$112 \in L(\text{COFFEE})$

Computation on w ?

transition diagram



s_0 s_1 s_2 s_3

0c, 10c, 20c, BREW

$\delta(0c, 1) = 10c$

$\delta(10c, 1) = 20c$

$\text{States} \times \Sigma \to \text{States}$

partial function

δ	1	2	B
0c	10c	20c	0c
10c		⋮	
20c			
30c			
brew	—	—	—

table of values

$\delta(20c, 2) = \text{BREW}$

Finite automata

$Q = \{0c, 10c, 20c, 30c, \text{BREW}\}$

$\Sigma = \{1, 2, \text{B}\}$

Table on the previous slide

Definition (1.1)

A finite automaton is described by a 5-tuple $(Q, \Sigma, \delta, q_0, Q_{\mathrm{acc}})$:

1. Q is a finite set, the *set of states*.
2. Σ is a finite set, the *input alphabet*.
3. $\delta : Q \times \Sigma \to Q$ is the *transition function*.
4. $q_0 \in Q$ is the *start state*.
5. $Q_{\mathrm{acc}} \subseteq Q$ is the set of *accepting states*.

"SYNTAX"

$q_0 = 0c$

$Q_{acc} = \{\text{BREW}\}$

Computations

"SEMANTICS"

Definition (1.2)

Let $M = (Q, \Sigma, \delta, q_0, Q_{acc})$ be a finite automaton and $w \in \Sigma^*$, $|w| = n$.

1. As sequence $s_0, \ldots, s_n \in Q$ is a *computation* of M on w, if
   1.1 $s_0 = q_0$,
   1.2 for all $0 \leq \nu < n$: $\delta(s_\nu, w_{\nu+1}) = s_{\nu+1}$,
2. The computation is *accepting*, if in addition $s_n \in Q_{acc}$. Otherwise it is called *rejecting*.

Regular languages

Definition (1.4)

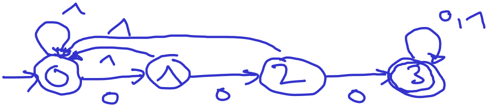
1. A finite automaton $M = (Q, \Sigma, \delta, q_0, Q_{acc})$ *accepts* a word $w \in \Sigma^*$, if there is an accepting computation of M on w. Otherwise M rejects w.
2. $L(M) = \{w \in \Sigma^* \mid M \text{ accepts } w\}$ is the language recognized by M.
3. $A \subseteq \Sigma^*$ is a *regular language*, if there is an M with $A = L(M)$.
4. REG is the set of all regular languages.

How to design a finite automaton?

$L_1 = \{w \in \{0,1\}^* \mid 000 \text{ is a subword of } w\}$.

1010001   100100

We need to remember the last two symbols



1010001

0 0s   1 0s   2 consecutive 0s

Extended transition function

$\delta : Q \times \Sigma \to Q$

$\delta^* : Q \times \Sigma^* \to Q$

Inductive definition on the word length

$$\delta^*(q, \varepsilon) = q$$

$$\delta^*(q, \underbrace{v\overset{\in \Sigma}{\sigma}}_{\ell+1}) = \begin{cases} \delta(\delta^*(q, \overset{\ell}{v}), \sigma) & \text{if } \delta^*(q, v) \text{ is defined} \\ \text{undefined} & \text{otherwise} \end{cases}$$

v is a word of length ℓ

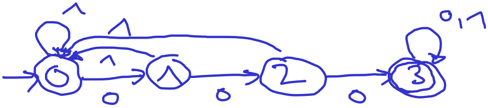
Observation: M accepts $w \iff \delta^*(q_0, w) \in Q_{acc}$

Lemma (1.5)

For all $q \in Q$ *and* $x, y \in \Sigma^*$, *we have*

$$\delta^*(q, xy) = \delta^*(\delta^*(q, x), y),$$

if $\delta^*(q, xy)$ *is defined.*



$$\delta^*(0,\ 0100) =$$

$$\delta(\delta^*(0,010),0)$$

$$= \delta(\delta(\delta^*(0,01),0),0)$$

$$= \delta(\delta(\delta(\delta^*(0,0),1),0),0)$$

$$= \delta(\underbrace{\delta(\underbrace{\delta(\underbrace{\delta(\underbrace{\delta^*(0,\varepsilon)}_{0},0)}_{0},1)}_{1},0)}_{0},0)$$

1

$= 2$