Finite Automata Theory and Formal Languages
Föreläsning 2 - Central concepts of automata theory

Erik Sjöström

March 22, 2016

1 Alphabets

Definition 1.1. *An alphabet is a finite, non-empty set of symbols, usually denoted by* Σ
The number of symbols in Σ *is donoted as* $|\Sigma|$

Notation. *We will use* $a, b, c, ...$ *to denote symbols.*

Anmärkning.
Alphabets will represent the observable events of the automata.

Exempel 1.1.
Some alphabets:

- *on/off-switch:* $\Sigma = \{Push\}$
- *simple vending machine:* $\Sigma = \{5\ kr,\ choc\}$
- *complex vending machine* $\Sigma = \{5\ kr,\ 10\ kr,\ choc,\ big\ choc\}$
- *parity counter* $\Sigma = \{p_0, p_1\}$

2 Strings or Words

Definition 2.1. *Strings/Words are finite sequences of symbols from some alphabet.*

Notation. *We will use* $w, x, y, z, ...$ *to denote words.*

Anmärkning.
Words will represent the behaviour of an automaton.

Exempel 2.1.
Some behaviours:

- *on/off-switch: Push, Push, Push, Push*
- *simple vending machine: 5 kr, choc, 5 kr, choc, 5 kr, choc*
- *parity counter:* p_0p_1 *or* $p_0p_0p_0p_1p_1p_0$

Anmärkning.
Some words do NOT represent behaviour

Exempel 2.2.
simple vending machine: choc, choc, choc

3 Inductive Definition of Σ^*

Definition 3.1. *Σ^* is the set of all words for a given alphabet Σ.*

This can be described inductively in at least 2 different ways:

- Base case: $\epsilon \in \Sigma^*$
- Inductive step: if $a \in \Sigma$ and $x \in \Sigma^*$ then $ax \in \Sigma^*$
- We will usually work with this definition.

Or:

- Base case: $\epsilon \in \Sigma^*$
- Inductive step: if $a \in \Sigma$ and $x \in \Sigma^*$ then $xa \in \Sigma^*$

We can (recursively) *define* functions over Σ^* and (inductively) *prove* properties about those functions.

4 Concatenation

Definition 4.1. *Given the string x and y, the concatenation xy is defined as:*

$$\begin{aligned} \epsilon y &= y \\ (ax')y &= a(x'y) \end{aligned}$$

Observe that in general $xy \neq xy$

Exempel 4.1.
If $x = 010$, and $y = 11$, then $xy = 01011$, and $yx = 11010$

Lemma. *If Σ has more than one symbol then concatenation is not commutative.*

5 Prefix and Suffix

Definition 5.1. *Given x and y words over a certain alphabet Σ:*

- *x is a prefix of y iff there exists z such that $y = xz$*
- *x is a suffix of y iff there xists z such that $y = zx$*

Anmärkning.
$\forall x$, ϵ is both a prefix and suffix of x.

Anmärkning.
$\forall x$, x is both a prefix and suffix of x.

6 Length and Reverse

Definition 6.1. *The length function $|\ |: \Sigma^* \to \mathbb{N}$ is defined as:*

$$\begin{aligned} |\epsilon| &= 0 \\ |ax| &= 1 + |x| \end{aligned}$$

Exempel 6.1.
$|01010| = 5$

Definition 6.2. *The* reverse *function* $rev(\,) : \Sigma^* \to \Sigma^*$ *is defined as:*

$$
\begin{aligned}
rev(\epsilon) &= \epsilon \\
rev(ax) &= rev(x)a
\end{aligned}
$$

Exempel 6.2.
$rev(a_1 ... a_n) \ = a_n ... a_1$

7 Power

7.1 Of a string

Definition 7.1. *We define* x^n *as follows:*

$$
\begin{aligned}
x^0 &= \epsilon \\
x^{n+1} &= xx^n
\end{aligned}
$$

Exempel 7.1.
$(010)^3 \ = (010010010)$

7.2 Of an alphabet

Definition 7.2. *We define* Σ^n*, the set of words over* Σ *with length* n*, as follows:*

$$
\begin{aligned}
\Sigma^0 &= \{\epsilon\} \\
\Sigma^{n+1} &= \{ax \mid a \in \Sigma, x \in \Sigma^n\}
\end{aligned}
$$

Exempel 7.2.
$\{0,1\}^3 \ = \{000, 001, 010, 011, 100, 101, 110, 111\}$

Notation.

$$
\begin{aligned}
\Sigma^* &= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 ... \\
\Sigma^+ &= \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 ...
\end{aligned}
$$

8 Some properties

The following properties can be proved by induction:

Lemma. *Concatenation is associative:* $\forall x, y, z$*:* $x(yz) = (xy)z$

Lemma. $\forall x$*:* $x\epsilon = \epsilon x = x$

Lemma. $\forall x$*:* $|x^n| = n * |x|$

Lemma. $\forall \Sigma$*:* $|\Sigma| = |\Sigma|^n$

Lemma. $\forall x$*:* $rev(rev(x)) = x$

Lemma. $\forall x, y$*:* $rev(xy) = rev(y)rev(x)$

9 Languages

Definition 9.1. *Given an alphabet Σ, a language $\mathcal{L}$ is a subset of Σ^*, that is, $\mathcal{L} \subseteq \Sigma^*$*

Anmärkning.
If $\mathcal{L} \subseteq \Sigma^*$ and $\Sigma \subseteq \Delta$ then $\mathcal{L} \subseteq \Delta^*$

Anmärkning.
A language can be either finite or infinite.

Exempel 9.1.
Some languages:

- *Swedish, English, French, Spanish...*
- *Any programming language*
- *$\emptyset$, $\{\epsilon\}$, and Σ^* are languages over any Σ*
- *The set of prime Natural numbers: $\{1, 3, 5, 7, 11, ...\}$*

10 Some Operations on Languages

Definition 10.1. *Given $\mathcal{L}, \mathcal{L}_1, \mathcal{L}_2$ languages, we define the following languages:*

- **Union, intersection:** The same as for any set.
- **Concatenation:** $\mathcal{L}_1\mathcal{L}_1 = \{x_1x_2 \mid x_1 \in \mathcal{L}_1, x_2 \in \mathcal{L}_2\}$
- **Closure:** $\mathcal{L}^* = \bigcup_{n\in\mathbb{N}} \mathcal{L}^n$ where $\mathcal{L}^0 = \{\epsilon\}$, $\mathcal{L}^{n+1} = \mathcal{L}^n\mathcal{L}$

Anmärkning.
$\emptyset^* = \{\epsilon\}$
$\mathcal{L}^* = \mathcal{L}^0 \cup \mathcal{L}^1 \cup \mathcal{L}^2 \cup ... = \{\epsilon\} \cup \{x_1...x_n \mid n > 0, x_i \in \mathcal{L}\}$

Notation. $\mathcal{L}^* = \mathcal{L}^1 \cup \mathcal{L}^2 \cup \mathcal{L}^3 \cup ...$

Exempel 10.1.
Let $\mathcal{L} = \{aa, b\}$, then

$$\begin{aligned}
&\mathcal{L}^0 = \{\epsilon\} \\
&\mathcal{L}^1 = \mathcal{L} \\
&\mathcal{L}^2 = \mathcal{L}\mathcal{L} = \{aaaa, aab, baa, bb\} \\
&\mathcal{L}^3 = \mathcal{L}^2\mathcal{L} \\
&\vdots \\
&\mathcal{L}^* = \{\epsilon, aa, b, aaaa, aab, baa, bb, ...\}
\end{aligned}$$

11 How to Prove the Equality of Languages?

Given the languages $\mathcal{L}$ and $\mathcal{M}$, how can we prove that $\mathcal{L} = \mathcal{M}$

A few possibilities:

- Languages are sets so we prive that $\mathcal{L} \subseteq \mathcal{M}$ and $\mathcal{M} \subseteq \mathcal{L}$
- Transitivity of equality: $\mathcal{L} = \mathcal{L}_1 = ... = \mathcal{L}_m = \mathcal{M}$
- We can reason about the elements in the language:

> **Exempel 11.1.**
> $\{a(ba)^n \mid n \geq 0\} = \{(ab)^n a \mid n \geq 0\}$ *can be proved by induction on* n.

12 Algebraic Laws for Languages

Laws of concatenation:

- Associativity: $\mathcal{L}(\mathcal{M}\mathcal{N}) = (\mathcal{L}\mathcal{M})\mathcal{N}$
- Not commutative: $\mathcal{L}\mathcal{M} \neq \mathcal{M}\mathcal{L}$
- Distributivity: $\mathcal{L}(\mathcal{M} \cup \mathcal{N}) = \mathcal{L}\mathcal{M} \cup \mathcal{L}\mathcal{N}$
- Distributivity: $(\mathcal{M} \cup \mathcal{N})\mathcal{L} = \mathcal{M}\mathcal{L} \cup \mathcal{N}\mathcal{L}$
- Identity: $\mathcal{L}\{\epsilon\} = \{\epsilon\}\mathcal{L} = \mathcal{L}$
- Annihilator: $\mathcal{L}\emptyset = \emptyset\mathcal{L} = \emptyset$
- Other Rules:
  - $\emptyset^* = \{\epsilon\}^* = \{\epsilon\}$
  - $\mathcal{L}^+ = \mathcal{L}\mathcal{L}* = \mathcal{L}^*\mathcal{L}$
  - $(\mathcal{L}^*)^* = \mathcal{L}^*$

Anmärkning.
While:

$$\mathcal{L}(\mathcal{M} \cap \mathcal{N}) \subseteq \mathcal{L}\mathcal{M} \cap \mathcal{L}\mathcal{N} \qquad \textit{and} \qquad (\mathcal{M} \cap \mathcal{N})\mathcal{L} \subseteq \mathcal{M}\mathcal{L} \cap \mathcal{N}\mathcal{L}$$

both hold, in general

$$\mathcal{L}\mathcal{M} \cap \mathcal{L}\mathcal{N} \subseteq \mathcal{L}(\mathcal{M} \cap \mathcal{N}) \qquad \textit{and} \qquad \mathcal{M}\mathcal{L} \cap \mathcal{N}\mathcal{L} \subseteq (\mathcal{M} \cap \mathcal{N})\mathcal{L}$$

don't.

> **Exempel 12.1.**
> *Consider the case where:*
> $$\mathcal{L} = \{\epsilon, a\}, \mathcal{M} = \{a\}, \mathcal{N} = \{aa\}$$
> *Then*
> $$\mathcal{L}\mathcal{M} \cap \mathcal{L}\mathcal{N} = \{aa\}$$
> *but*
> $$\mathcal{L}(\mathcal{M} \cap \mathcal{N}) = \mathcal{L}\emptyset = \emptyset$$

13 Functions between Languages

Definition 13.1. *A function $f : \Sigma^* \to \Delta^*$ between 2 languages should satisfy:*

$$f(\epsilon) = \epsilon$$
$$f(xy) = f(x)f(y)$$

Intuitively, $f(a_1...a_n) = f(a_1)...f(a_n)$

Anmärkning.
$f(a) \in \Delta^*$ *if* $a \in \Sigma$