Finite automata theory and formal languages (DIT321, TMV027)

Nils Anders Danielsson,
partly based on slides by Ana Bove

2019-01-29

Today

Deterministic finite automata.

Inductively defined subsets

Inductively defined subsets

- One can define subsets of (say) Σ^* inductively.
- For instance, for $L \subseteq \Sigma^*$ we can define $L^* \subseteq \Sigma^*$ inductively:

$$\frac{}{\varepsilon \in L^*} \qquad \frac{u \in L \qquad v \in L^*}{uv \in L^*}$$

- Note that there are no constructors.

Inductively defined subsets

- What about recursion?

$$\begin{aligned} &f \in L^* \to Bool \\ &f(\varepsilon) = false \\ &f(uv) = not(f(v)) \end{aligned}$$

- If $\varepsilon \in L$, do we have

$$f(\varepsilon) = f(\varepsilon\varepsilon) = not(f(\varepsilon))?$$

Inductively defined subsets

- Another example:

$$\begin{aligned}
&f \in L^* \to \mathbb{N} \\
&f(\varepsilon) = 0 \\
&f(uv) = |v|
\end{aligned}$$

- If $L = \{ 1, 11 \}$, do we have $f(11) = |\varepsilon| = 0$ or $f(11) = |1| = 1$?
- Recursion of this kind can make sense if, say, every string matches exactly one left-hand side, in exactly one way, and the recursive calls are made on strictly shorter lists.

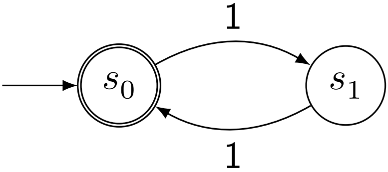
Inductively defined subsets

- Induction works
  (assuming "proof irrelevance").
- $P(\varepsilon) \wedge (\forall u \in L, v \in L^*.\ P(v) \Rightarrow P(uv)) \Rightarrow \forall w \in L^*.\ P(w)$.

DFAs

DFAs

Recall from the first lecture:



- A DFA specifies a language.
- In this case the language $\{ 11 \}^* = \{ \varepsilon, 11, 1111, \dots \}$.

DFAs

A DFA can be given by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$:

- A finite set of states (Q).
- An alphabet (Σ).
- A transition function $(\delta \in Q \times \Sigma \to Q)$.
- A start state $(q_0 \in Q)$.
- A set of accepting states $(F \subseteq Q)$.

Which of the following 5-tuples can be seen as DFAs?

1. $(\mathbb{N}, \{ 0, 1 \}, \delta, 0, \{ 13 \})$,
   where $\delta(n, m) = n + m$.
2. $(\{ 0, 1 \}, \emptyset, \delta, 0, \{ 1 \})$, where $\delta(n, \_) = n$.
3. $(\{ q_0, q_1 \}, \{ 0, 1 \}, \delta, q_0, \{ 1 \})$,
   where $\delta(\_, \_) = q_0$.
4. $(\{ q_0, q_1 \}, \{ 0, 1 \}, \delta, q_0, \{ q_0 \})$,
   where $\delta(\_, \_) = 0$.

Semantics

The language of a DFA

The language $L(A)$ of a DFA $A = (Q, \Sigma, \delta, q_0, F)$ is defined in the following way:

- A transition function for strings is defined by recursion:

$$\hat{\delta} \in Q \times \Sigma^* \to Q$$
$$\hat{\delta}(q, \varepsilon) = q$$
$$\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$$

- The language is $\left\{ w \in \Sigma^* \;\middle|\; \hat{\delta}(q_0, w) \in F \right\}$.

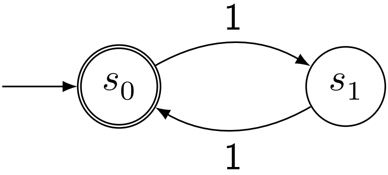
Which strings are members of the language of $(\{ s_0, s_1, s_2, s_3 \}, \{ a, b \}, \delta, s_0, \{ s_0 \})$? Here δ is defined in the following way:

$$
\begin{array}{ll}
\delta(s_0, a) = s_1 & \delta(s_0, b) = s_2 \\
\delta(s_1, a) = s_0 & \delta(s_2, b) = s_0 \\
\delta(\_, \_) = s_3 & \text{(In all other cases.)}
\end{array}
$$

1. ε.
2. aab.
3. aba.
4. $aabbaa$.
5. $abbaab$.
6. $bbaaaa$.

Transition diagrams

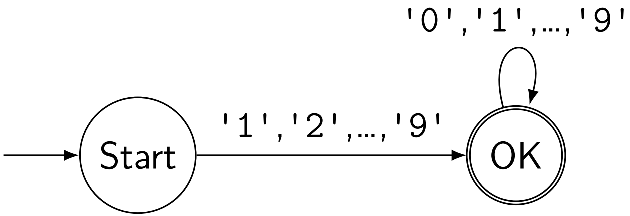
Transition diagrams



- One node per state.
- An arrow "from nowhere" to the start state.
- Double circles for accepting states.
- For every transition $\delta(s_1, a) = s_2$,
  an arrow marked with a from s_1 to s_2.
  - Multiple arrows can be combined.

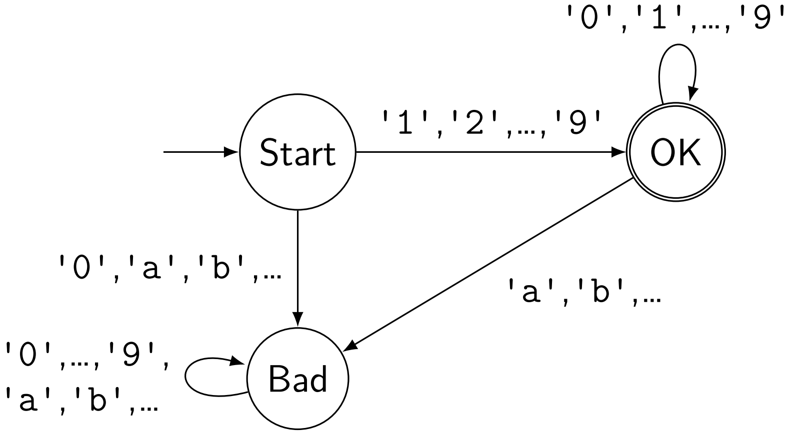
A variant

Diagrams with “missing transitions”:



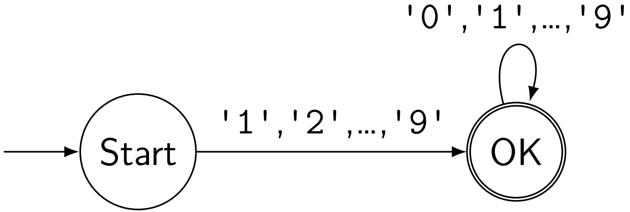
A variant

Every missing transition goes to a new state (that is not accepting):



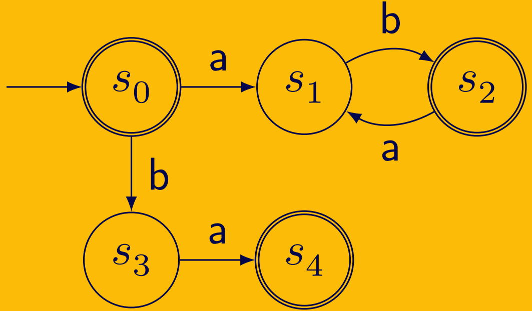
A variant

Note that diagrams with missing transitions do not define the alphabet unambiguously:



The alphabet must be a (finite) superset of { '0', '1', ..., '9' }, but which one?

Which strings are members of the language of the DFA defined by the following transition diagram? The alphabet is $\{ a, b \}$.



1. ε.
2. aa.
3. ab.
4. ba.
5. $abab$.
6. $baba$.

Transition tables

Transition tables

	0	1
$\rightarrow *s_0$	s_2	s_1
s_1	s_2	s_0
s_2	s_2	s_2

- States: Left column.
- Alphabet: Upper row.
- Start state: Arrow.
- Accepting states: Stars.
- Transition function: Table.

Which strings are members of the language of the DFA defined by the following transition table?

	0	1
$\rightarrow s_0$	s_2	s_1
$*s_1$	s_2	s_0
$*s_2$	s_2	s_2

1. ε.
2. 0.
3. 1.
4. 11.
5. 111.
6. 1010.

Constructions

Complement

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$ we can construct a DFA $\overline{A}$ that satisfies the following property:

$$L(\overline{A}) = \overline{L(A)} = \Sigma^* \setminus L(A).$$

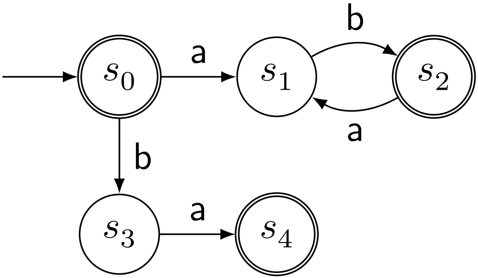
Construction:

$$(Q, \Sigma, \delta, q_0, Q \setminus F).$$

We accept if the original automaton doesn't.

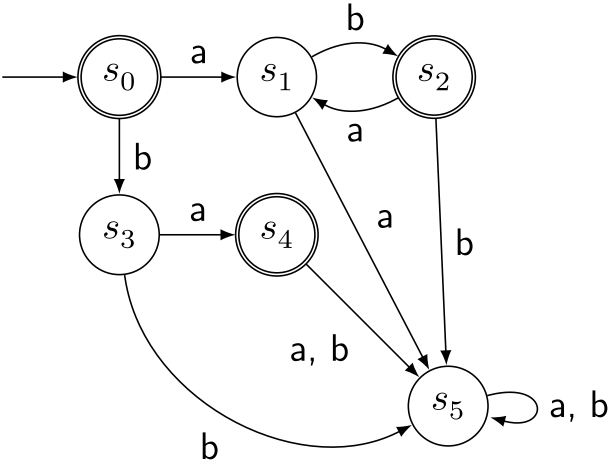
Complement

$A =$



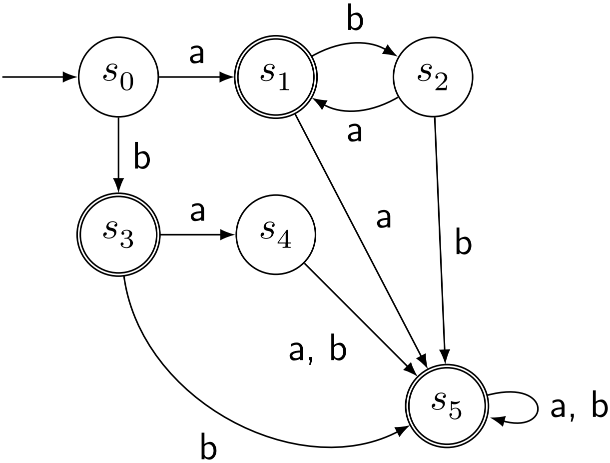
Complement

$A =$



Complement

$\overline{A} =$



Product

Given two DFAs $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ with the same alphabet we can construct a DFA $A_1 \otimes A_2$ that satisfies the following property:

$$L(A_1 \otimes A_2) = L(A_1) \cap L(A_2).$$

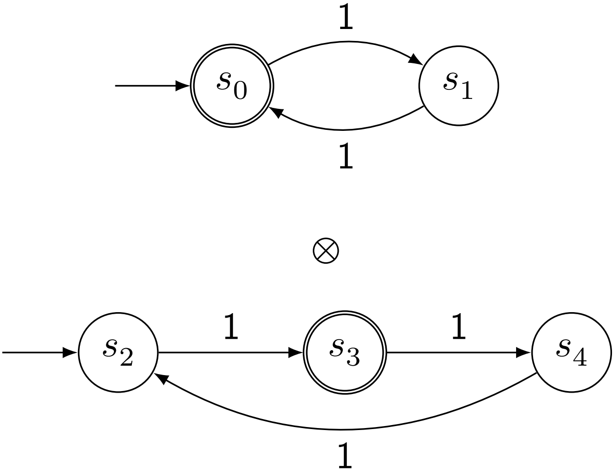
Construction:

$(Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q_{02}), F_1 \times F_2)$, where
$\delta((s_1, s_2), a) = (\delta_1(s_1, a), \delta_2(s_2, a))$.

We basically run the two automatons in parallel and accept if both accept.

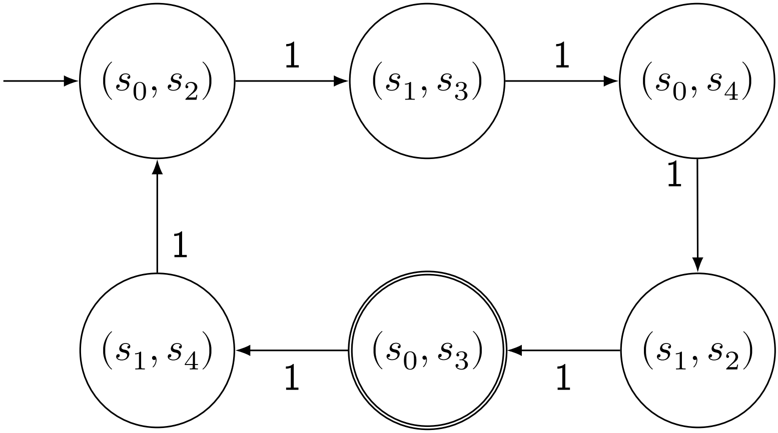
Product

$\{\, 2n \mid n \in \mathbb{N} \,\} \cap \{\, 1 + 3n \mid n \in \mathbb{N} \,\}$
(in unary notation, with ε standing for 0):



Product

$\{ \, 4 + 6n \mid n \in \mathbb{N} \, \}$:



We can also construct a DFA $A_1 \oplus A_2$ that satisfies the following property:

$$L(A_1 \oplus A_2) = L(A_1) \cup L(A_2).$$

The construction is basically that of $A_1 \otimes A_2$, but with a different set of accepting states. Which one?

1. $F_1 \cup F_2$.
2. $F_1 \cap F_2$.
3. $Q_1 \times Q_2$.
4. $F_1 \times Q_2 \ \cup \ Q_1 \times F_2$.
5. $F_1 \times Q_2 \ \cap \ Q_1 \times F_2$.

Regular languages

Regular languages

- A language $M \subseteq \Sigma^*$ is *regular* if there is some DFA A with alphabet Σ such that $L(A) = M$.
- Note that if M and N are regular, then $M \cap N$, $M \cup N$ and $\overline{M}$ are also regular.

Today

- 5-tuples.
- Semantics.
- Transition diagrams.
- Transition tables.
- Constructions.
- Regular languages.

Consultation time

- Today, right after the exercise session, in EL42.
- You decide what you want to work on.

Next lecture

- Nondeterministic finite automata (NFAs).
- The subset construction
  (turns NFAs into DFAs).

- Deadline for the next quiz:
  2019-01-31, 10:00.
- Deadline for the first assignment:
  2019-02-03, 23:59.