AUTOMATA & FORMAL LANGUAGES

VI

18 October 2023

Lecture V, page 10:

Applying $\hat{\delta}$ to a word w and the start state q_0 produces a unique seq. of states of length $|w|+1$. Call this the STATE SEQUENCE.

Def. A tuple $(\Sigma, Q, \delta, q_0, F)$ is called a (deterministic) automaton

if Σ is an alphabet

Q is a finite set of states

$q_0 \in Q$ start state

cf. page 6 $q_0 \neq F \subseteq Q$ set of accepting states

$\delta : Q \times \Sigma \longrightarrow Q$ transition function

INTUITION: Automaton starts in state q_0 and reads a word $w \in W$ letter-by-letter. For each letter a it reads (while it is in state $q \in Q$), it moves into state $\delta(q,a)$.

At the end, we reached final state q. If $q \in F$, we say ACCEPT! If $q \notin F$, we say REJECT!

Define by recursion on the length of $w \in W$

$$\hat{\delta} : Q \times W \longrightarrow Q$$

$$\hat{\delta}(q, \varepsilon) := q$$

$$\hat{\delta}(q, wa) := \delta(\hat{\delta}(q,w), a)$$

We define $\mathcal{L}(D) := \{ w ;\ \hat{\delta}(q_0, w) \in F \}$

the set of words accepted by D.

Graphical representation of automata by
LABELLED DIRECTED GRAPH

Take automaton $\mathcal{D} = (\Sigma, Q, \delta, q_0, F)$:

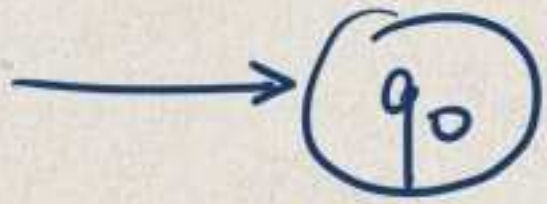
VERTICES: Q states

EDGES: $q \xrightarrow{a} q'$ iff $\delta(q,a) = q'$.

SINGLE CIRCLE: if $q \notin F$

DOUBLE CIRCLE: if $q \in F$

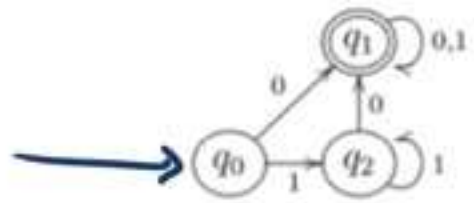
If needed, indicate q_0 by an extra arrow coming out of nowhere



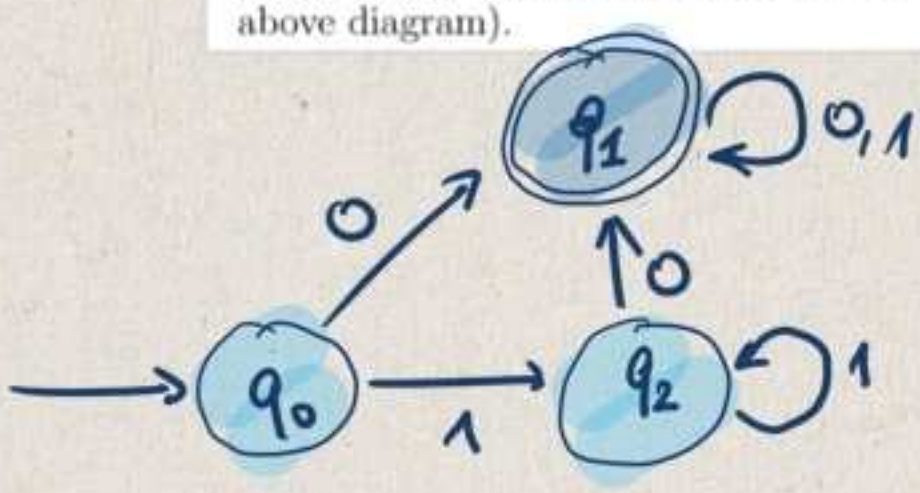
Observe Every node has exactly $|\Sigma|$ many outgoing arrows.

[This follows from the fact that δ is a function. This is different in the case of nondeterministic automata; cf. page 7.]

We graphically represent automata by directed labelled graphs where the vertices are labelled by the elements of Q, each vertex has precisely $|\Sigma|$ immediate successors marked by directed edges labelled with the letters of the alphabet Σ. The vertices labelled with non-elements of F get a single circle, and the vertices labelled with elements of F get a double circle. The following is an example for $\Sigma = \{0, 1\}$, $Q = \{q_0, q_1, q_2\}$, and $F = \{q_1\}$:



Note that we simplify our graphical representations by writing a single arrow with multiple labels if they all have the same source and target (e.g., the arrow from q_1 to itself in the above diagram).



Convention: if multiple arrows connect q, q', draw just one and write all labels next to it.

Example 2.4. The automaton graphically represented above accepts the language

$$L := \{w\,;\, w \text{ contains at least one } 0\}.$$

[Let us analyse which words will result in state q_0, q_1, and q_2, respectively. By this we mean words s such that $\widehat{\delta}(q_0, w) = q_i$. Since q_0 has no incoming edges, the only word that results in q_0 is ε. The state q_2 has two incoming edges, a 1-transition from q_0 which means that the word $\varepsilon 1 = 1$ results in q_2, and then a 1-transition from q_2. So, by induction, any finite sequence consisting entirely of 1s will result in q_2; thus, the words that result in q_2 are precisely the words in $\{1\}^+$. Finally, all 0-transitions lead to q_1, so any word that contains a 0 will always be in state q_1 immediately after reading that 0. However, since both transitions from q_1 lead to q_1, you cannot ever leave that state. In summary, the empty word ends in q_0, any word consisting of 1s results in q_2, and any word that contains a 0 results in q_1. This description covers all possible words. We note that q_1 is the only accepting state, which proves the claim.]

DEFINITION

If $D = (\Sigma, Q, \delta, q_0, F)$ and $D' = (\Sigma, Q', \delta', q_0', F')$ are deterministic automata over the same alphabet Σ, we say that a map $f : Q \to Q'$ is a *homomorphism from* D *to* D' if

(i) for all $q \in Q$ and $a \in \Sigma$, we have that $\delta'(f(q), a) = f(\delta(q, a))$,

(ii) we have $f(q_0) = q_0'$, and

(iii) for all $q \in Q$, $q \in F$ if and only if $f(q) \in F'$.

As usual, bijective homomorphisms are called *isomorphisms* and automata that have an isomorphism between them are called *isomorphic*. Note that if f is a bijection, then f^{-1} satisfies (i) to (iii) and thus is a homomorphism.

Property (i) immediately gives (by induction):

(i*) $\hat{\delta}(f(q), w) = f(\hat{\delta}(q, w))$.

Prop. 2.5 If f is a homomorphism between D & D', then $\mathcal{L}(D) = \mathcal{L}(D')$.

Proof.

$w \in \mathcal{L}(D) \overset{\text{Def.}}{\iff} \hat{\delta}(q_0, w) \in F$

$\overset{\text{(iii)}}{\iff} f(\hat{\delta}(q_0, w)) \in F'$

$\overset{\text{(i*)}}{\iff} \hat{\delta}(f(q_0), w) \in F'$

$\overset{\text{(ii)}}{\iff} \hat{\delta}(q_0', w) \in F'$

$\overset{\text{Def.}}{\iff} w \in \mathcal{L}(D')$.

q.e.d.

[REMARK. Normally in maths, equivalence requires ISOMORPHISM, but Proposition 2.5 only needs the existence of a HOMOMORPHISM.]

Theorem 2.6 (typed notes).
If D is a det. automaton, then $\mathcal{L}(D)$ is regular.

Proof. Let $D = (\Sigma, Q, \delta, q_0, F)$ be the automaton.
Define $G = (\Sigma, Q, P, q_0)$ with P defined as:

(variables are states; start symbol is start state)

$$p \longrightarrow aq \in P \;:\Longleftrightarrow\; \delta(p,a) = q$$

$$p \longrightarrow a \in P \;:\Longleftrightarrow\; \delta(p,a) \in F.$$

Claim: $\mathcal{L}(D) = \mathcal{L}(G)$. Let $w = a_0 \dots a_n$.

"$\subseteq$". If $w \in \mathcal{L}(D)$, let $q_0, q_1, \dots, q_n$ be the state sequence of w, so $\delta(q_i, a_i) = q_{i+1}$, so $q_i \xrightarrow{G}_1 a_i q_{i+1}$

$$q_0 \xrightarrow{G}_1 a_0 q_1 \xrightarrow{G}_1 \dots \xrightarrow{G}_1 a_0 \dots a_{n-1} q_n \xrightarrow{G}_1 w$$

where $q_n \longrightarrow a_n \in P$ since $\delta(q_n, a_n) \in F$.
Thus $w \in \mathcal{L}(G)$.

"$\supseteq$". If $w \in \mathcal{L}(G)$, have derivation

$$q_0 \xrightarrow{G}_1 a_0 q_1 \xrightarrow{G}_1 \dots \xrightarrow{G}_1 a_0 \dots a_{n-1} q_n \xrightarrow{G}_1 w$$

for some variables q_i. By definition of P, these q_i must be the state sequence. Since $\delta(q_n, a_n) \in F$, $w \in \mathcal{L}(D)$ q.e.d.

This proof is so natural that we would assume that you can reverse the construction.

$D \longmapsto G$

Alas, that doesn't quite work:

Suppose G is regular grammar with

$A \longrightarrow aB \in P$ where $B \neq C$

$A \longrightarrow aC$

Then we can't define $\delta(a, A)$ to be either B or C without breaking that δ is a function.

Need to liberalize that part of the definition $\longrightarrow$ NONDETERMINISTIC AUTOMATA.

[Seite 7]

Remark on ε. Since $q_0 \notin F$, our automata can never accept ε.

That's fine since regular grammars don't do that either.

But what if you want that to happen?

If you drop the req. of $q_0 \notin F$, then you get automata that correspond to ESSENTIALLY REGULAR GRAMMARS.

[Same issues as with grammars apply: automaton needs to be "ε-adequate".]

Definition A tuple $N = (\Sigma, Q, \delta, q_0, F)$ is called a **nondeterministic automaton** if

Q is a finite set
Σ is an alphabet
$q_0 \in Q$
$q_0 \notin F \subseteq Q$
$\delta : Q \times \Sigma \longrightarrow \mathcal{P}(Q)$ (power set of Q)

Interpretation:
$\delta(q,a)$ is the set of potential states that could be reached from q when reading a.

Graphical representation

Exactly the same as for deterministic automata, but

$q \xrightarrow{a} q' :\Longleftrightarrow q' \in \delta(q,a)$

and therefore, there could be multiple a-labelled arrows from q to somewhere or none.

Similarly to the STATE SEQ. in the det. case, here we have a STATE SET SEQUENCE.

As before, we define $\hat{\delta}$ by recursion:

$$\hat{\delta} : Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

$$\hat{\delta}(q, \varepsilon) := \{q\}$$

$$\hat{\delta}(q, wa) := \bigcup_{p \in \hat{\delta}(q,w)} \delta(p, a)$$

All possibly reachable states from q by reading wa.

Observe that nondet. automata are more general than det. automata.

[If δ in a nondet. automaton gives use always 1-elt sets, it is deterministic.]

Theorem 2.8 If G is a regular grammar, there is a non-deterministic automaton N s.t. $\mathcal{L}(G) = \mathcal{L}(N)$.

[Note: We still need the relationship between det. & nondet. automata to decide that a characterization term ⟶ Lecture III.]

Proof.

$$G = (\Sigma, V, P, S)$$

$$H \notin \Omega = \Sigma \cup V.$$

$$Q := V \cup \{H\}.$$

special state corresponding to termination.

$$N := (\Sigma, Q, \delta, S, \{H\})$$

where

$$\delta(A,a) := \begin{cases} \{B,\ A \to aB \in P\} & \text{if } A \to a \notin P \\ \{B,\ A \to aB \in P\} \cup \{H\} & \text{if } A \to a \in P \end{cases}$$

Claim $\mathcal{L}(N) = \mathcal{L}(G)$.

Proof of this in lecture VII.