Automata & Formal Languages
Sixth Lecture

26 October 2024

RECAP

Lecture IV

Regular derivations have a VARIABLE SEQUENCE.

Chapter 2 : REGULAR LANGUAGES

Regular grammar $A, B \in V$ & $a \in \Sigma$

		# letters	# variable	length
TERMINAL RULES	$A \to a$	+1	−1	0
NONTERMINAL RULES	$A \to aB$	+1	0	+1

$$S \xrightarrow{G} w$$

1. There is precisely terminal rule: at the very end.
2. There are precisely $|w|-1$ nonterminal rules.

$$S \to a_0 A_0 \to a_0 a_1 A_1 \to a_0 a_1 a_2 A_2 \to \dots$$
$$\dots \to a_0 a_1 a_2 \dots a_{n-1} A_{n-1} \to a_0 \dots a_n$$

Lecture V

Similarly, an automaton computation has a STATE SEQUENCE.

§ 2.2 Deterministic Automata

Fix Σ.

$D = (\Sigma, Q, \delta, q_0, F)$ is called deterministic automaton if:

1. Q is a finite set of states
2. $q_0 \in Q$ start state,
3. $F \subseteq Q \setminus \{q_0\}$ accept states
4. $\delta : Q \times \Sigma \to Q$ transition function

RECAP 2 — Homomorphisms.

Prop 2.5 If $f: D \longrightarrow D'$ homomorphism, then $\mathcal{L}(D) = \mathcal{L}(D')$.

Application W.l.o.g., $q_0 \notin \text{ran}(\delta)$.

---

§ 2.5 Closure properties

Closure under complementation:

If $L \in \mathcal{L}$, then so is $W^+ \setminus L$.

Prop 2.17 The class of languages accepted by an automaton is closed under complementation.

[HOMEWORK: Think about automata that allow $q_0 \in F$ and how the proof of P 2.17 proves that we get an automaton for $W \setminus \mathcal{L}(D)$!]

Proof of T 2.17 $D = (\Sigma, Q, \delta, q_0, F)$

Idea Flip "accept" & "reject".

Problem $q_0 \in Q \setminus F$, but q_0 is not allowed as "accept" state.

Solution $F := (Q \setminus F) \setminus \{q_0\}$.

$\bar{D} := (\Sigma, Q, \delta, q_0, F)$

Claim $L(\bar{D}) = \Sigma^+ \setminus L(D)$

[Use application to assume w.l.o.g. $q_0 \notin \mathrm{ran}(\delta)$]

"$\subseteq$": $w \in L(\bar{D}) \Longrightarrow \hat{\delta}(q_0, w) \in F = (Q \setminus F) \setminus \{q_0\}$

$$\Longrightarrow \begin{cases} \hat{\delta}(q_0, w) \neq q_0 & \Longrightarrow w \neq \varepsilon \\ \hat{\delta}(q_0, w) \notin F & \Longrightarrow w \notin L(D) \end{cases} \Longrightarrow w \in \Sigma^+ \setminus L(D)$$

"$\supseteq$" Suppose we $W^+ \setminus L(D)$.

$w \neq \varepsilon$ $\quad \hat{\delta}(q_0, w) \notin F$.

Since $w \neq \varepsilon$ & $q_0 \notin \text{ran}(\delta) \Rightarrow \hat{\delta}(q_0, w) \neq q_0$

Thus $\hat{\delta}(q_0, w) \in (Q \setminus F) \setminus \{q_0\} = \emptyset$. qed

Remark This would be even easier if we didn't have the wlg that $q_0 \notin F$.

Then just flipping proves that the flipped automaton gives $W \setminus L(D)$.

[Once again: the issue of ε.]

PRODUCT AUTOMATA

There is an alternative proof for union and intersection that can be instructive in certain contexts. Given nonempty sets Q and Q', as well as $F \subseteq Q$ and $F' \subseteq Q'$, we let

$$F \wedge F' := \{(q, q') \in Q \times Q' ; q \in F \text{ and } q' \in F'\} = F \times F' \text{ and}$$
$$F \vee F' := \{(q, q') \in Q \times Q' ; q \in F \text{ or } q' \in F'\}.$$

We can now give a product construction of two automata: if $D = (\Sigma, Q, \delta, q_0, F)$ and $D' = (\Sigma, Q', \delta', q_0', F')$ are two automata, we define

$$\delta \times \delta' : \Sigma \times (Q \times Q') \to Q \times Q' : (a, (q, q')) := (\delta(a, q), \delta'(a, q')).$$

This allows us to define product automata for intersection and union as follows:

$$D \wedge D' := (\Sigma, Q \times Q', \delta \times \delta', (q_0, q_0'), F \wedge F'),$$
$$D \vee D' := (\Sigma, Q \times Q', \delta \times \delta', (q_0, q_0'), F \vee F').$$

Proposition 2.18. For any automata D and D', we have

$$\mathcal{L}(D \wedge D') = \mathcal{L}(D) \cap \mathcal{L}(D') \text{ and}$$
$$\mathcal{L}(D \vee D') = \mathcal{L}(D) \cup \mathcal{L}(D').$$

Proof. By definition (and induction), $\widehat{\delta \times \delta'}(w, (q, q')) = (\widehat{\delta}(w, q), \widehat{\delta'}(w, q'))$. Therefore,

$$\begin{aligned} w \in \mathcal{L}(D \wedge D') &\iff \widehat{\delta \times \delta'}(w, (q_0, q_0')) \in F \wedge F' \\ &\iff (\widehat{\delta}(w, q_0), \widehat{\delta'}(w, q_0')) \in F \times F' \\ &\iff \widehat{\delta}(w, q_0) \in F \text{ and } \widehat{\delta'}(w, q_0') \in F' \\ &\iff w \in \mathcal{L}(D) \text{ and } w \in \mathcal{L}(D'), \end{aligned}$$

and similarly for $D \vee D'$. Q.E.D.

HOMEWORK: Check that you understand this proof!

Remark The product construction is quite flexible. [Try to define F s.t. the product accepts precisely $w \in \mathcal{L}(D) \setminus \mathcal{L}(D')$ or $w \in \mathcal{L}(D') \setminus \mathcal{L}(D)$ or ...]

GOAL

L is regular iff there is an automaton D s.t. $L = \mathcal{L}(D)$.

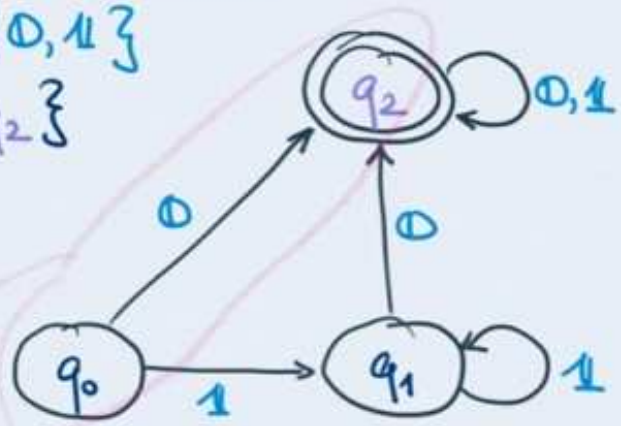
Example from Lecture V:

GRAPHICAL REPRESENTATION:

$\Sigma = \Sigma_{01} = \{0, 1\}$

$Q = \{q_0, q_1, q_2\}$

$F = \{q_2\}$



$\delta(q_0, 0) = q_2$

P

$q_0 \to 0\,q_2$ | $q_0 \to 0$

$q_0 \to 1\,q_1$

$q_1 \to 0\,q_2$ | $q_1 \to 0$

$q_1 \to 1\,q_1$

$q_2 \to 0\,q_2$ | $q_2 \to 0$

$q_2 \to 1\,q_2$ | $q_2 \to 1$

ACCEPT STATES

$G = (\Sigma, Q, q_0, P)$

More generally $D = (\Sigma, Q, \delta, q_0, F)$ automaton

Define $G = (\Sigma, Q, q_0, \mathcal{P})$ with

$$\mathcal{P} = \{ p \to aq \;;\; \delta(a,p) = q \} \cup \{ p \to a \;;\; \delta(a,p) \in F \}$$

Then: $\mathcal{L}(D) = \mathcal{L}(G)$.

[If $w \in \mathcal{L}(D)$, the state seq. of the accepting computation yields a variable seq. for a G-derivation producing w.

If $w \in \mathcal{L}(G)$, then the variable seq. of the derivation is the state seq. of the automaton.

Since the der. ends in a terminal wd, the final state is in F.]

[Theorem 2.6: Every language accepted by automaton is regular.]

Remark: This construction cannot be inverted since the fact that we $L(G)$ does not determine the derivation uniquely, and therefore does not determine the variable seq. uniquely.

Could have $A \to \textcircled{0}B$ $\quad B \neq C$ $\quad A \to \textcircled{0}C$

So $S(\textcircled{0}, A)$ is not uniquely determined.

Grammars only produce POSSIBLE computations and only use of them words to accept.

[§2.3] A tople $N = (\Sigma, Q, \delta, q_0, F)$ is called NONDETERMINISTIC AUTOMATON if

1. Q is a finite set of states
2. $q_0 \in Q$ start state
3. $F \subseteq Q \setminus \{q_0\}$ accept states
4. $\delta : Q \times \Sigma \longrightarrow \mathcal{P}(Q)$ POWER SET OF Q

Interpretation: next possible states

STATE SET SEQUENCE

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

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

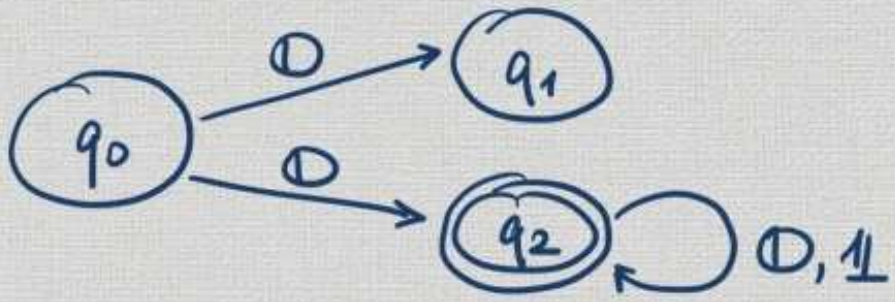
$$\mathcal{L}(N) := \{w;\ \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

REMINDER

§ 2.2 Deterministic Automata

Fix Σ. $D = (\Sigma, Q, \delta, q_0, F)$ is called deterministic automaton if

1. Q is a finite set of states
2. $q_0 \in Q$ start state
3. $F \subseteq Q \setminus \{q_0\}$ accept states
4. $\delta : Q \times \Sigma \longrightarrow Q$ transition functions



$F = \{q_2\}$

STATE SET SEQUENCE	$w_0 = 010$		$w_1 = 101$	
0	ε	$\{q_0\}$	ε	$\{q_0\}$
1	0	$\{q_1, q_2\}$	1	$\emptyset$ MACHINE BREAKS
2	01	$\emptyset \cup \{q_2\} = \{q_2\}$ MACHINE BREAKS	10	$\emptyset$
3	010	$\{q_2\}$	101	$\emptyset$
$\hat{\delta}(q_0, w)$				
$F \cap \hat{\delta}(q_0, w)$		$\{q_2\} \neq \emptyset$		$\emptyset$

HOMEWORK : Do a few more examples like this !

Theorem TFAE:

(i) L is regular

(ii) $L = \mathcal{L}(D)$ for D det. automaton

(iii) $L = \mathcal{L}(N)$ for N nondet. automaton

Proof (ii) $\Rightarrow$ (i) is T2.6.

(i) $\Rightarrow$ (iii) Suppose $G = (\Sigma, V, S, P)$ regular

Build N s.t. $\mathcal{L}(G) = \mathcal{L}(N)$.

Idea Invert the T2.6 construction.

Let $H \notin \Sigma \cup V$ "halt" Let $Q := V \cup \{H\}$

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

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

By construction $H \in \hat{\delta}(S, w) \iff$ there is $S \xrightarrow{G} w$

So $\mathcal{L}(N) = \mathcal{L}(G)$

(iii) $\Rightarrow$ (i) SUBSET CONSTRUCTION

If $N = (\Sigma, Q, \delta, q_0, F)$ is a nondeterministic automaton

define $D = (\Sigma, 2^Q, \Delta, \{q_0\}, G)$ (power set)

$$\Delta(X, a) = \bigcup \{\delta(p, a), p \in X\}$$

$$X \in G \iff X \cap F \neq \emptyset$$

Then by construction: state seq. of D correspond to state set seq of N, so $\mathcal{L}(D) = \mathcal{L}(N)$

qed