IX

NINTH LECTURE OF AUTOMATA & FORMAL LANGUAGES

25 October 2023

RECAP: REGULAR EXPRESSIONS

Lecture VIII page 9

Let Σ be an alphabet. Among the finite strings over the set $\Sigma \cup \{\varnothing, \varepsilon, (,), +, ^+, ^*\}$; we shall define the notion of *regular expressions over* Σ by recursion:[10]

(1) The symbol $\varnothing$ is a regular expression;

(2) the symbol ε is a regular expression;

(3) every $a \in \Sigma$ is a regular expression,

(4) if R and S are regular expressions, then $(R + S)$ is a regular expression;

(5) if R and S are regular expressions, then (RS) is a regular expression;

(6) if R is a regular expression, then R^+ is a regular expression;

(7) if R is a regular expression, then R^* is a regular expression;

(8) nothing else is a regular expression.

sions by recursion:

(1) If $E = \varnothing$, then $\mathcal{L}(E) = \varnothing$;

(2) if $E = \varepsilon$, then $\mathcal{L}(E) = \{\varepsilon\}$;

(3) if $E = a$ for $a \in \Sigma$, then $\mathcal{L}(E) = \{a\}$;

(4) if R and S are regular expressions, then $\mathcal{L}((R + S)) = \mathcal{L}(R) \cup \mathcal{L}(S)$;

(5) if R and S are regular expressions, then $\mathcal{L}((RS)) = \mathcal{L}(R)\mathcal{L}(S)$;

(6) if R is a regular expression, then $\mathcal{L}(R^*) = \mathcal{L}(R)^*$;

(7) if R is a regular expression, then $\mathcal{L}(R^+) = \mathcal{L}(R)^+$.[11]

Lecture VIII page 10:

We now associate languages to regular expressions by recursion:

(1) If $E = \varnothing$, then $\mathcal{L}(E) = \varnothing$;

(2) if $E = \varepsilon$, then $\mathcal{L}(E) = \{\varepsilon\}$;

(3) if $E = a$ for $a \in \Sigma$, then $\mathcal{L}(E) = \{a\}$;

(4) if R and S are regular expressions, then $\mathcal{L}((R + S)) = \mathcal{L}(R) \cup \mathcal{L}(S)$;

(5) if R and S are regular expressions, then $\mathcal{L}((RS)) = \mathcal{L}(R)\mathcal{L}(S)$;

(6) if R is a regular expression, then $\mathcal{L}(R^*) = \mathcal{L}(R)^*$;

(7) if R is a regular expression, then $\mathcal{L}(R^+) = \mathcal{L}(R)^+$.[11]

THEOREM

Let $L \subseteq \mathbb{W}$ be a language. Then the following are eq.:

(i) L is essentially regular

(ii) there is a regex R s.t. $\mathcal{L}(R) = L$.

We will only prove (ii) $\Rightarrow$ (i).

In order to prove (ii) $\Rightarrow$ (i) only need that if L is regular, then so is L^+.

Lecture VIII page 11:

Start of proof of that closure property:
[If L regular, then L^+ regular]

Let G be a regular grammar, $L = \mathcal{L}(G)$.
Need to find grammar G^+ s.t. $\mathcal{L}(G^+) = L^+$.

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

no S on RHS of any rule

By previous discussion, can assume w.l.o.g. G is ε-adequate.

$P^+ := P \cup \{A \to aS \,;\, A \to a \in P\}$

$G^+(\Sigma, V, P^+, S)$

Claim (proof in Lecture IX):

$$\mathcal{L}(G^+) = L^+$$

Proof of $L^+ = \mathcal{L}(G^+)$.

(1) $L^+ \subseteq \mathcal{L}(G^+)$. Let $w \in L^+$, i.e., $w = w_0 \dots w_n$ where $w_i \in L$.

So for i have $S \xrightarrow{G} w_i = v_i a_i$ [Pick v_i, a_i s.t. $w_i = v_i a_i$]

$$S \xrightarrow{G} v_i A_i \xrightarrow[1]{G} w_i$$

uses rule $A_i \to a_i \in P$

$\Longrightarrow A_i \to a_i S \in P^+$

$$S \xrightarrow{G^+} w_i S$$

$$S \xrightarrow{G^+} w_0 S \xrightarrow{G^+} w_0 w_1 S \xrightarrow{G^+} w_0 w_1 w_2 S \xrightarrow{G^+} \dots$$

$$\xrightarrow{G^+} w_0 w_1 \dots w_n = w.$$

(2) $\mathcal{L}(G^+) \subseteq L^+$.

Assume $S \xrightarrow{G^+} w$

Since G was ε-adequate, all instances of S in this derivation (except for the initial S) are due to one of the new rules in P^+.

Let $k > 0$ be the number of S occurrences in the derivation.

Prove the claim by induction on k.

<u>$k=1$</u> $\underbrace{S \xrightarrow{G^+} w}_{\text{no } S}$ $\Longrightarrow S \xrightarrow{G} w$

$\Longrightarrow w \in \mathcal{L}(G) = L \subseteq L^+$. ✓

<u>Ind. step</u> Suppose true for k.

$$\underbrace{\underbrace{S \xrightarrow{G^+} v}_{k \text{ many } S}\, \underbrace{S \xrightarrow{G^+} w}_{\text{no } S}} = vu$$

ε-adequacy used

since vS must come from a new rule $A \to aS$ where $A \to a \in P$

$\Longrightarrow \underbrace{S \xrightarrow{G^+} v}_{k \text{ many } S}$

$\overset{IH}{\Longrightarrow} v \in L^+$

$vS \xrightarrow{G^+} vu$

$\Longrightarrow S \xrightarrow{G^+} u$

$\Longrightarrow S \xrightarrow{G} u$ since no S occurs

$\Longrightarrow u \in \mathcal{L}(G) = L$.

$\Downarrow$

$\Longrightarrow w = vu \in L^+ L \subseteq L^+$. q.e.d.

MINIMAL DETERMINISTIC AUTOMATA

In this entire section, automata will be deterministic.

If L is regular,

$$\emptyset \neq \{n \in \mathbb{N};\ \exists D \text{ with } |Q| = n \ \&\ \mathcal{L}(D) = L\} \subseteq \mathbb{N}.$$

deterministic

If we use "nondeterministic", this number may be diff. See ES#2.

There is a minimal size of det. automata for L, and therefore there are

MINIMAL AUTOMATA

for L, but it is conceivable **BUT ACTUALLY FALSE** that there are substantially different (= non-isomorphic) minimal automata.

We will show:

there is a unique (up to isomorphism) minimal automaton for L.

Lecture VI, page 4

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):

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

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

Let's explore reasons why f might be a homomorphism but not inj. / surj.

If $D = (\Sigma, Q, \delta, q_0, F)$ is a deterministic automaton, we call a state $q \in Q$ *inaccessible* if there is no word w such that $\widehat{\delta}(q_0, w) = q$. We call two states $q, q' \in Q$ *indistinguishable* if for all words w, we have that

$$\widehat{\delta}(q, w) \in F \iff \widehat{\delta}(q', w) \in F. \iff A_q = A_{q'}.$$

A word w such that $\widehat{\delta}(q, w) \in F$ and $\widehat{\delta}(q', w) \notin F$ or vice versa is said to *distinguish* q and q'. Given $q, q' \in Q$ and $a \in \Sigma$ and $\delta(q, a)$ and $\delta(q', a)$ are distinguished by a word w, then q and q' are distinguished by the word aw. If $f : Q \to Q'$ is a homomorphism from an automaton D to an automaton D', then if $p, q \in Q$ are distinguishable, then $f(p) \neq f(q)$. Furthermore, if $q' \in Q'$ is accessible, then $q' \in \mathrm{ran}(f)$.

We write $q \sim q'$ if they are indistinguishable. Note that $\sim$ is an equivalence relation on Q, i.e., reflexive, symmetric, and transitive. We write $[q]$ for the $\sim$-equivalence class of q.

$$A_q := \{w ; \hat{\delta}(q, w) \in F\}$$

If $D = (\Sigma, Q, \delta, q_0, F)$ is a deterministic automaton, we call a state $q \in Q$ *inaccessible* if there is no word w such that $\widehat{\delta}(q_0, w) = q$. We call two states $q, q' \in Q$ *indistinguishable* if for all words w, we have that

$$\widehat{\delta}(q, w) \in F \iff \widehat{\delta}(q', w) \in F.$$

A word w such that $\widehat{\delta}(q, w) \in F$ and $\widehat{\delta}(q', w) \notin F$ or vice versa is said to *distinguish* q and q'. Given $q, q' \in Q$ and $a \in \Sigma$ and $\delta(q, a)$ and $\delta(q', a)$ are distinguished by a word w, then q and q' are distinguished by the word aw. If $f : Q \to Q'$ is a homomorphism from an automaton D to an automaton D', then if $p, q \in Q$ are distinguishable, then $f(p) \neq f(q)$. Furthermore, if $q' \in Q'$ is accessible, then $q' \in \operatorname{ran}(f)$.

We write $q \sim q'$ if they are indistinguishable. Note that $\sim$ is an equivalence relation on Q, i.e., reflexive, symmetric, and transitive. We write $[q]$ for the $\sim$-equivalence class of q. We

Observe

(1) If f is a homomorphism from D to D' and p, q are distinguishable [i.e., wlog $\widehat{\delta}(p,w) \in F$ for $\widehat{\delta}(q,w) \notin F$ some w]

then $f(p) \neq f(q)$.

Sufficient condition for injectivity.

Therefore if D has no indistinguishable states, f is injective.

(2) f homom. from D to D' and $q' \in Q'$ is accessible [$\exists w \ \widehat{\delta}'(q_0', w) = q'$]

then there is some $q \in Q$ s.t. $f(q) = q'$.

$$\left[\ \widehat{\delta}'(q_0', w) = \widehat{\delta}'(f(q_0), w) = f(\widehat{\delta}(q_0, w))\ \right]$$

Sufficient condition for surjectivity.

Therefore, if D' has no inaccessible states, then f is surjective.

Def. A det. automaton D is called IRREDUCIBLE if it has no unacc. or indistinguishable states.

Corollary (to our observations)
If I, I' are irreducible and f is a homom. between I & I', then f is an isomorphism.

THE CLEAN-UP AUTOMATON

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

Let $A \subseteq Q$ be the set of accessible states
and $\delta^* := \delta \upharpoonright A \times \Sigma$

$D^* := (\Sigma, A, \delta^*, q_0, F \cap A)$ clean-up automaton

Observe: (1) $\delta^* : A \times \Sigma \longrightarrow A$
(2) $id : A \longrightarrow Q$ is a homomorphism

So: (3) $\mathcal{L}(D) = \mathcal{L}(D^*)$
(4) D^* has no inaccessible states.

THE QUOTIENT AUTOMATON

Fix $D = (\Sigma, Q, \delta, q_0, F)$.

Indistinguishability $q \sim q'$ is an equivalence relation

$[q] := \{q' ; q \sim q'\}$

$Q/\sim := \{[q] ; q \in Q\}$

$[\delta]([q], a) := [\delta(q, a)]$

$[F] := \{[q] ; q \in F\}$

$D/\sim := (\Sigma, Q/\sim, [\delta], [q_0], [F])$ quotient automaton

Observe: (1) $[\delta]$ is well-defined.
(2) $q \longmapsto [q]$ is a homomorphism from D to $D/\sim$

So: (3) $\mathcal{L}(D) = \mathcal{L}(D/\sim)$.
(4) $D/\sim$ has no indistinguishable states.

Summary

$$D \rightsquigarrow D^* \rightsquigarrow D^*/\sim$$

(1) preserves the language

(2) produces an irreducible automaton s.t.

$$|A/\sim| \leq |Q|$$

Moreover, if D was not irreducible, then

$$|A/\sim| < |Q|.$$

CONSEQUENCE

For any reducible automaton D, there is an irreducible automaton with strictly fewer states.

therefore if n is ~~the~~ minimal # of states, every automaton of size n is irreducible.

THEOREM Any two irreducible automata accepting the same language are isomorphic.

This implies that all minimal automata accepting L are isomorphic and of minimal size.

Proof. $I = (\Sigma, Q, \delta, q_0, F)$ irreducible
$I' = (\Sigma, Q', \delta', q_0', F')$

If $q \in Q$, $q' \in Q'$, I write

$$q \sim q' \quad :\text{iff} \quad \underset{\{w;\, \hat{\delta}(q,w) \in F\}}{\overset{\parallel}{A_q}} = \underset{\{w;\, \hat{\delta}'(q',w) \in F'\}}{\overset{\parallel}{A'_{q'}}}$$

Claim 1 Each $q \in Q$ has $q' \in Q'$ s.t. $q \sim q'$.

Case 1 $q = q_0$. Then $A_{q_0} = \mathcal{L}(I) = \mathcal{L}(I')$
$= A'_{q_0'}$

$$\Longrightarrow q_0 \sim q_0'.$$

Case 2 $q = \delta(q_0, w)$ some $w \in W^+$.

Define $q' := \hat{\delta}'(q_0', w)$.

Subclaim: $A_q = A'_{q'}$:

[if $\hat{\delta}'(q', v) \in F'$ & $\hat{\delta}(q, v) \notin F$,
then $\hat{\delta}'(q_0', wv) \in F'$ & $\hat{\delta}(q_0, wv) \notin F$
$\Longrightarrow wv \in \mathcal{L}(I')$ $\Longrightarrow wv \notin \mathcal{L}(I)$ contr.]

There is no Case 3, since everything that is neither Case 1 or 2 must be inaccessible which doesn't exist.

<u>Claim 2</u> If $q \sim q'$ and $q \sim p'$, then $q' = p'$.

Proof: In Lecture $\underline{X}$.

<u>PREVIEW</u> Claim 1 & 2 together mean that

$$q \longmapsto \text{the unique } q' \text{ s.t. } q \sim q'$$

is a function.
This is a homomorphism; but homomorphisms between irred. automata are isos (as proved earlier).