COL352 Lecture 14

Contents

Continuation of lecture 13.

1 Recap

Discussion in previous class about Myhill-Nerode theorem.

2 Definitions

3 Content

Given DFA $D = (Q, \Sigma, \delta, q_0, A)$ without unreachable states.
We want DFA $D^* = (Q^*, \Sigma, \delta^*, q_0^*, A^*)$ with min number of states that recognizes $\mathcal{L}(D)$.
We know $\sim_D$ refines $=_{\mathcal{L}(D)}$, and $\sim_{D^*}$ is identical to $=_{\mathcal{L}(D)}$.

Definition 1

$C_q = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) = q\}$.

Note that none of the C_q's are empty since D has no unreachable states.
We then know that these are equivalence classes of $\sim_D$.
Let x_q be an arbitrary string in C_q, i.e., $\hat{\delta}(q_0, x_q) = q$.

Definition 2

$\equiv$ is an equivalence relation on Q defined as $q \equiv q'$ if $C_q, C_{q'}$ are in the same equivalence class of $=_{\mathcal{L}(D)}$.

Claim 0.1

$q \equiv q' \iff x_q =_{\mathcal{L}(D)} x'_q$

Proof. Forward direction: Obvious by the definition of $\equiv$ and $x_q \in C_q, x_{q'} \in C_{q'}$

Backward direction: Follows from the fact that $\sim_D$ refines $=_{\mathcal{L}(D)}$ and the equivalence class of x_q wrt $\sim_D$ is C_q, and similarly for $C_{q'}$. □

Claim 0.2

If $q \equiv q'$, then $\forall a \in \Sigma, \delta(q, a) \equiv \delta(q', a)$.

Proof. $q \equiv q' \iff x_q =_L x_{q'}$ from the previous claim.

$x_q =_L x_{q'} \implies x_q a =_L x_{q'} a$ (as done in last class) $\implies \hat{\delta}(q_0, x_q a) \equiv \hat{\delta}(q_0, x_{q'} a) \implies \delta(\hat{\delta}(q_0, x_q), a) \equiv \delta(\hat{\delta}(q_0, x_{q'}), a) \implies \delta(q, a) \equiv \delta(q', a)$. □

Claim 0.3

$q \equiv q'$ iff $\forall z \in \Sigma^*, \hat{\delta}(q,z), \hat{\delta}(q',z)$ are both in A or both not in A.

Proof. $q \equiv q' \iff x_q =_L x_{q'} \iff \forall z$ either both $x_q z, x_{q'} z$ are in L or both not in L $\iff \forall z, \hat{\delta}(q_0, x_q z), \hat{\delta}(q_0, x_{q'} z)$ both in A or both not in A iff ... iff $\hat{\delta}(q,z), \hat{\delta}(q',z)$ both in A or both not in A. ☐

Having determined $\equiv$, D^* can be constructed as:

1. Q^* : one state from each equivalence class of $\equiv$.
2. rep : $Q \to Q^*$ where $\text{rep}(q) =$ the representative of the equivalence class of q under $\equiv$.
3. $q_0^* = \text{rep}(q_0)$.
4. $\delta^*(r,a) = \text{rep}(\delta(r,a))$.
5. $A^* = \{\text{rep}(q) \mid q \in A\}$.

The invariant is $\hat{\delta}^*(q_0^*, x) = \text{rep}(\hat{\delta}(q,x))$.

Question 1

How do we construct this equivalence relation $\equiv$?

Answer. Recall that $q \equiv q'$ iff $\forall z \in \Sigma^*, \hat{\delta}(q,z), \hat{\delta}(q',z)$ are both in A or both not in A.

Definition 3

$q \equiv_n q'$ if for all $z \in \Sigma^*$ of length $\leq n$, $\hat{\delta}(q,z), \hat{\delta}(q',z)$ are both in A or both not in A.

Note 1

Note that each $\equiv_n$ refines $\equiv_{n-1}$, and $\equiv$ refines $\equiv_n$. Also, $q \equiv q \iff \forall n, q \equiv_n q'$.

Also note that $\equiv_0$ is easy to find: $q \equiv_0 q'$ iff either both q, q' are in A or both are not in A, so we can use $\equiv_0$ as an equivalent statement for either both q, q' being in A or not being in A.

Claim 0.4

$q \equiv_n q'$ iff $q \equiv_{n-1} q' \wedge \forall a \in \Sigma, \delta(q,a) \equiv_{n-1} \delta(q',a)$.

Proof. Forward direction:

$$q \equiv_n q' \implies (\forall z, |z| \leq n \implies \hat{\delta}(q,z) \equiv_0 \hat{\delta}(q',z))$$

Specialising to $|z| \leq n-1$, we have $q \equiv_{n-1} q'$.

Now continuing, we have for z' of length $\leq n-1$, $\hat{\delta}(q, az') = \hat{\delta}(q', az') \iff \hat{\delta}(\delta(q,a), z') \equiv_0 \hat{\delta}(\delta(q',a), z')$, so $\delta(q,a) \equiv_{n-1} \delta(q',a)$.

Backward direction:

If $|z| \leq n-1$, use $q \equiv_{n-1} q'$ to get $\hat{\delta}(q,z) \equiv_0 \hat{\delta}(q',z)$. If $|z| = n$, then $z = az'$ for some $a \in \Sigma$, $z' \in \Sigma^*$, and $|z'| \leq n-1$. So we have $\hat{\delta}(q,z) = \hat{\delta}(q,az') = \hat{\delta}(\delta(q,a),z') \equiv_0 \hat{\delta}(\delta(q',a),z')$(either both are or are not in A) $= \hat{\delta}(q',z)$, so $q \equiv_n q'$. ☐

Question 2

Can we make an algorithm to find the equivalence classes?

Answer.

function PARTITION($D = (Q, \Sigma, \delta, q_0, A)$)
  $n \leftarrow 0$
  $\equiv_0 = \{(q, q') \in Q \times Q \mid \text{ both } q, q' \in A \text{ or both } q, q' \notin A\}$
  while True **do**
    $n \leftarrow n + 1$
    $\equiv_n \leftarrow \{(q, q') \in \equiv_{n-1} \wedge \forall a \in \Sigma, (\delta(q, a), \delta(q', a)) \in \equiv_{n-1}\}$.
    if $\equiv_n = \equiv_{n-1}$ **then**
      break
    end if
  end while
  return $\equiv_n$
end function

Claim 0.5

The above algorithm terminates.

Proof. For all $i < n$, number of equivalence classes of $\equiv_i$ is more than the number of equivalence classes of $\equiv i - 1$. But $\forall i$, the number of equivalence classes of $\equiv_i$ is at most $|Q|$. So we terminate in at most $|Q| + 1$ iterations. □

Claim 0.6

If $\equiv_n$ is identical to $\equiv_{n-1}$, then they are identical to $\equiv$.

Proof. Intuition for why we will do this – the number of equivalence classes is non-decreasing, so there will be an N such that $\equiv = \equiv_N$.

Now coming to the proof, we claim the following:

Claim 0.7

If $\equiv_n$ is identical to $\equiv_{n-1}$ then $\equiv_{n+1}$ is identical to $equiv_n$.

Proof. Use the definition $q \equiv_n q'$ iff ($q \equiv_{n-1} q'$ and $\forall a \in \Sigma, \delta(q, a) \equiv_{n-1} \delta(q', a)$).

So we have

$$q \equiv_{n+1} q' \iff (q \equiv_n q' \text{ and } \forall a \in \Sigma, \delta(q, a) \equiv_n \delta(q', a) \quad \iff (q \equiv_{n-1} q' \text{ and } \forall a \in \Sigma, \delta(q, a) \equiv_{n-1} \delta(q', a) \iff q \equiv_n q'$$

□

Corollary 1

If $\equiv_n$ is identical to $\equiv_{n-1}$, then $\forall m \geq n, \equiv_m$ is identical to $\equiv_n$

Corollary 2

If $\equiv_n$ is identical to $\equiv_{n-1}$, then $\equiv$ is identical to $\equiv_n$.

Proof. Intersection of states that stabilize. □

□

Example done in class for DFA minimization using this algorithm.
We will need an arbitrarily large number of steps.
For an example, consider the obvious DFA for $\{z \in \Sigma^* \mid |z| \geq n\}$.

3.1 Context-Free Languages

Motivating example:

Example 1

Inductively define L as the smallest language satisfying the following:

1. $\epsilon \in L$.
2. If $x, y \in L$ then $x \cdot y \in L$.
3. If $x \in L$, then $0x1 \in L$.

Note 2

This is something like balanced parenthesized expressions. We needed the smallest language thing because any superset of L also works.

Claim 0.8

$x \in L \iff$ the number of 0s in $x =$ number of 1s in x, and for all y (prefixes of x), the number of 0s in y is at least the number of 1s in y.

Proof. Exercise. $\square$

Claim 0.9

This language is not regular.

Proof. Use the pumping lemma on $0^n 1^n$ or give another proof using the Myhill-Nerode theorem. $\square$

This is something like a grammar.
In what follows, S is the initial non-terminal.
$S \to \epsilon$
$S \to SS$
$S \to 0S1$
We can make parse trees using this.