CS 205
Lecture 17
22/2/21

DFA Minimization

Idea
1. For any regular language L, the DFA M_L obtained from R_L is the unique (up to iso) minimum DFA that accepts L.
2. R_M refines R_L for any DFA M that accepts L.
3. Idea behind minimizing a given DFA M:
   - Merge states of M: merge q and q' iff $\delta(q_0, x) = q$ and $\delta(q_0, y) = q' \Rightarrow xR_L y$

- Given a DFA M (where all states are reachable from q_0)

Def of $\equiv$: Define equivalence reln. $\equiv$ on the states of M:
$$p \equiv q \text{ iff for each } x \in \Sigma^*,\ \delta(p,x) \in F \Leftrightarrow \delta(q,x) \in F$$

- p and q are equivalent if $p \equiv q$
  distinguishable if $p \not\equiv q$

- Need to show: Correctness of $\equiv$ wrt which states can be merged

$$p \equiv q \underset{\text{iff}}{\Leftrightarrow} \forall x \forall y.\ \delta(q_0, x) = p \text{ and } \delta(q_0, y) = q \Rightarrow x R_L y$$

Proof: Easy (Exercise)

Computing $\equiv$: How do we determine if $p \equiv q$, where $p, q \in Q$?

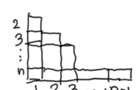
Inductive Definition of $\not\equiv$ (distinguishability)
1. If $p \in F$ and $q \notin F$, or vice versa, then $p \not\equiv q$.
2. If for some a, $\delta(p,a) \not\equiv \delta(q,a)$ then $p \not\equiv q$.

Every other pair is equivalent.

Claim: This defn. is equivalent to the one given earlier

Proof: Exercise.

Minimization Algorithm



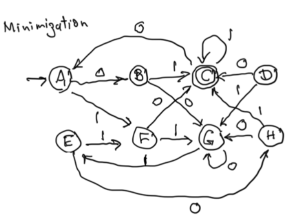
1. Eliminate all unreachable states from M.
2. Distinguish $\leftarrow \{\langle p,q\rangle \mid p \in F \text{ and } q \notin F, \text{ or vice versa}\}$
3. OldDistinguish $\leftarrow \emptyset$ /* initializations */
4. **while** (Distinguish $\neq$ OldDistinguish)
5. OldDistinguish $\leftarrow$ Distinguish
6. **for** every symbol a and for every $\langle p,q\rangle \notin$ Distinguish **do**
7. **if** $\langle \delta(p,a), \delta(q,a)\rangle \in$ Distinguish **then**
8. Distinguish $\leftarrow$ Distinguish $\cup \{\langle p,q\rangle\}$

Time Complexity
- # iteration of while loop: $O(n)$ Why?
- Time taken in each iteration: $O(|\Sigma| n^2)$
- Total time $O(n^3)$

There is an algorithm that runs in time $O(n \log n)$.

Ex DFA Minimization

DFA M



Table

	A	B	C	D	E	F	G
B	X						
C	X	X					
D			X				
E	≡		X				
F			X	≡			
G			X				
H		≡	X				