CSE 555 Theory of Computation Class 7 (2/5)

Goran Konjevod

Department of Computer Science and Engineering
Arizona State University

ASU, Spring 2008

DFA minimization

Given: DFA M.
Want: Minimum-size DFA M' equivalent to M ($L(M') = L(M)$).

Simple ideas

1. Unreachable (inaccessble) states are unnecessary.
2. Indistinguishable states are redundant

Indistinguishability

States q and q' are **indistinguishable**, if for every string x, $\delta(q, x)$ and $\delta(q', x)$ are either both in F or both not in F.

Negation: q and q' are **distinguishable**, if there exists a string x such that exactly one of the two states $\delta(q, x)$ and $\delta(q', x)$ is in F.

Goal: find all indistinguishable pairs.

Algorithm should achieve the following:

1. For each state $q \in Q$: if q is unreachable, remove q from Q.
2. For each pair q, q' such that q and q' are indistinguishable, identify q and q' and only keep one of them in the DFA.

Correctness claim

Claim: First removing inaccessible states and then merging all sets of indistinguishable states into single states suffices to find a minimal DFA.

Proof: Homework 3, questions 2 and 3.

Standard algorithm

1. For each pair $\{p, q\}$, set $D(\{p, q\}) = 0$.
2. For each pair $\{p, q\}$ with $p \in F$ and $q \notin F$, set $D(\{p, q\}) = 1$.
3. done:=false
4. While not(done):
   1. done:=true
   2. $T := D$
   3. For each pair $\{p, q\}$ with $T(\{p, q\}) = 0$:
   4.   For each a:
   5.   If $T(\delta(p, a), \delta(q, a)\}) = 1$:
   6. $D(\{p, q\}) := 1$
   7.   done:=false.
5. Return(D).

Idea: While loop has at most $n - 2$ iterations.
For loop (4.3) has at most n^2 iterations.
For loop (4.4) has at most Σ iterations.

A better implementation

1. For each pair $\{p, q\}$ set $L(\{p, q\}) = \emptyset$.
2. For each pair $\{p, q\}$ with $p \in F$ and $q \notin F$, set $D(\{p, q\}) = 1$.
3. For each pair $\{p, q\}$ with $p, q \in F$ or $p, q \notin F$:
    1. If $D(\{\delta(p, a), \delta(q, a)\}) = 1$ for some a:
    2. $\quad D(\{p, q\}) := 1$
    3. $\quad$ Recursively set $D(\{p', q'\}) := 1$ for all unmarked pairs $\{p', q'\}$ in $L(\{p, q\})$ and all pairs in those lists, etc.
    4. Else:
    5. $\quad$ For each a:
    6. $\quad\quad$ If $\delta(p, a) \neq \delta(q, a)$ and $\{\delta(p, a), \delta(q, a)\} \neq \{p, q\}$:
    7. $\quad\quad\quad$ Add $\{p, q\}$ to $L(\delta(p, a), \delta(q, a)\})$
4. Return(D).

Recursive call in 3.3: once per pair in a list.
Total entries in all lists over the whole algorithm: $O(n^2)$.

A very simple algorithm

A "generalized NFA": possibly more than one start state.
For a generalized NFA A, $S(A)$: the equivalent DFA made by the subset construction using only reachable states.
Algorithm of Brzozowski:

1. Reverse transitions of M to get a generalized NFA M^R.
2. Eliminate unreachable states.
3. Run subset construction to get a deterministic version $S(M^R)$.
4. Eliminate unreachable states.
5. Reverse again: $(S(M^R))^R$.
6. Run subset construction again: $S((S(M^R))^R)$.

Possibly exponential time, but easy to do by hand.

Most efficient algorithm

Due to Hopcroft, running time $O(n \log n)$.
Difficult to implement.