CSE547T Class 9

Jeremy Buhler

February 15, 2017

1 Automating the Myhill-Nerode Theorem

Last time, we showed that the smallest DFA accepting a language L was closely related to the equivalence classes of I_L. Can we build this DFA mechanically?

- Suppose we can construct *any* DFA accepting L.
- (DFA may be very large, built via subset construction, etc.)
- Can we then derive a minimal DFA for L?
- Why solve this problem?
  1. You've built a DFA and you want to use it over and over again.
  2. You are space-constrained.
  3. You like optimization.

We will describe an algorithm for minimizing a DFA.

2 A New Definition and A Good Idea

- Previously, we defined a relation I_M on strings induced by a DFA M.
- Recall that xI_My iff $\delta^*(q_0, x) = \delta^*(q_0, y)$.
- In other words, xI_My if x and y take M to the same state.
- Since I_M is always refinement of I_L for $L = L(M)$, this meant that xI_Ly. But what if I_M is a *proper* refinement of I_L?
- Implies that there exist strings x and y such that $\delta^*(q_0, x) = p$ and $\delta^*(q_0, y) = q \neq p$, and yet xI_Ly.
- But then, for any other x', y' that take M to p and q respectively, $x'I_Ly'$ as well. States p and q are therefore "the same" from the point of view of I_L.
- **Defn**: states p and q of a DFA M are *indistinguishable* if $\delta^*(p, z)$ and $\delta^*(q, z)$ are either both accepting or both non-accepting for every $z \in \Sigma^*$.

- A pair of states p, q that is not indistinguishable is said to be *distinguishable*, and in particular may be distinguished by some string z s.t. exactly one of $\delta^*(p, z)$ and $\delta^*(q, z)$ is in A.
- **Example**:

How does this definition help us?

- Suppose states p and q of M are indistinguishable.
- Then for all strings x, $\delta^*(p, x) \in A$ iff $\delta^*(q, x) \in A$.
- But in this case, no harm in redirecting all edges into p to point to q instead – $L(M)$ is unchanged!

- Now delete unused state p, and the result is a strictly smaller DFA!
- **Observation**: if I_M has more classes than I_L, there must be a pair of indistinguishable states in M, which can be shrunk to a single state as above.
- When no more shrinking is possible, I_M is identical to I_L, and so (by the theorem last time) the final DFA is minimal among all DFAs for its language.

Conclude that we can find a minimal DFA equivalent to M by successively collapsing pairs of indistinguishable states in M until no more collapsing is possible.

3 The Algorithm (Simple Version)

Here follows a simple, slow, but polynomial-time algo for identifying pairs of indistinguishable states.

- Goal is to *mark* every pair of states $(p, q) \in M$ that are distinguishable.
- Runs iteratively until no more marking is possible.
- I am deliberately making it slow to simplify explanation.

```
MarkAll(M = (Q, Σ, q0, A, δ))
  for all pairs (p, q) ∈ Q × Q do
    if exactly one of p, q is in A
      mark(p,q)
  do                                          ▷ discover all distinguished pairs
    changed ← false
    for all unmarked pairs (p, q) ∈ Q × Q do
      for all a ∈ Σ do
        if pair (δ(p, a), δ(q, a)) is marked
          mark(p,q)
          changed ← true
  while changed
```

Last step: collapse the DFA by combining sets of indistinguishable states into one.

- Number the states of M arbitrarily.
- For each state q, let $\chi(q)$ be the lowest-numbered state that is indistinguishable from q.
- For all $p \in Q, a \in \Sigma$ s.t. $\delta(p, a) = q$, set $\delta(p, a) = \chi(q)$.
- $\chi(q_0)$ becomes new start state.
- Finally, remove all states not reachable from $\chi(q_0)$.

4 Example

Here is a DFA to be minimized.

Let's run the marking algorithm.

Finally, collapse states to form new minimal DFA.

5 Correctness (Incomplete)

Thm: the MarkAll algorithm marks a state pair (p, q) iff p, q are distinguishable states of M.

- Will prove one direction and leave other as a (very similar) exercise.
- I claim that all distinguishable pairs (p, q) get marked eventually.
- Let (p, q) be distinguishable states, and let z be a string of minimum length s.t. (WLOG) $\delta^*(p, z) \in A$ but $\delta^*(q, z) \notin A$.
- **Claim**: MarkAll marks (p, q) after at most $|z|$ iterations of the main loop.
- Proceed by induction on $|z|$.
- **Bas**: if $z = \varepsilon$, then by definition, we have that $\delta^*(p, \varepsilon) = p \in A$ but $\delta^*(q, \varepsilon) = q \notin A$.
- Hence, initialization step of MarkAll marks pair (p, q) before we ever run the main loop.
- **Ind**: In general, suppose $z = ay$.
- Let $r = \delta(p, a)$ and $s = \delta(q, a)$.
- By assumption, $\delta^*(p, ay) = \delta^*(r, y) \in A$, while $\delta^*(q, ay) = \delta^*(s, y) \notin A$.
- Hence, r and s are distinguishable by a string of length at most $|z| - 1$.
- By IH, (r, s) will be marked after at most $|z| - 1$ loop iterations.
- Hence, in the $|z|$th iteration, if pair (p, q) is not already marked, the main loop will mark it.
- Conclude that (p, q) is marked after at most $|z|$ loop iterations.

Technically, we must also show that collapsed DFA M' is equivalent to old M, but I won't do it carefully.

- Easy to prove that $L(M) \subseteq L(M')$ using mapping from q to $\chi(q)$ for each state on path.
- To prove that $L(M') \subseteq L(M)$, need only demonstrate that

$$\delta^*_{M'}(\chi(q), x) = \chi(\delta^*_M(q, x)).$$

6 Cost?

How expensive is DFA minimization?

- Let n be size of input DFA.
- Naively, time could be $O(|\Sigma|n^4)$ – one pass through main loop takes time $O(|\Sigma|n^2)$, and we might mark only one of $\Theta(n^2)$ pairs per pass.
- Can lower cost substantially if we check state pairs in right order (Huffman and Moore, 1954-56)
- *Idea*: a pair (p, q) is shown to be distinguished either
  - directly, if only one of them is accepting, or
  - indirectly, if one of the $|\Sigma|$ pairs $(\delta(p, a), \delta(q, a))$ is proven to be distinguished.
- Suppose we build a graph of size $O(n^2)$ whose nodes are pairs and with an edge from each (p, q) to the pairs $(\delta(p, a), \delta(q, a))$ for each $a \in \Sigma$.
- Resulting graph is sparse – only $O(|\Sigma|n^2)$ edges.
- Start by marking the directly distinguished pairs.
- Then, for each distinguished pair, mark its predecessors, then their predecessors, etc in (say) BFS order, until all reachable pairs have been marked. This takes time $O(V + E)$, or $O(|\Sigma|n^2)$, since it visits every state pair at most once.
- Finally, transfer the markings to a table like in the original algorithm, and do the collapsing step in time $O(|\Sigma|n^2)$.
- **Conclude**: can minimize in only $O(|\Sigma|n^2)$ time.
- *Practical hint*: you need not represent the graph explicitly to traverse it. Just keep around the queue from BFS as your "work list" of nodes to check.

Can we do even better?

- Hopcroft (1971): minimization can be done in time $O(|\Sigma|n \log n)$.
- Hopcroft's algorithm avoids explicitly enumerating pairs.
- Instead, it starts with all states in one big set, then progressively splits that set, finally producing partition corresponding to the minimal automaton.
- Uses a slightly fancy "partition refinement" data structure to do splitting operations quickly.
- See the Wikipedia entry on DFA minimization for pseudocode and explanation.