Equivalence and Minimization of Deterministic Finite Automata

CS154

Outline

- Introduction
- Testing of equivalence of states
  - Definitions
- Table-filling algorithm
- Minimization of DFA's
- Questions

Introduction

Basic Idea of Minimization of DFAs

- We can take any DFA and find an equivalent DFA that has the minimum number of states.
  - In fact, this DFA is essentially unique: given any two minimum state DFAs that are equivalent, we can always find a way to rename the states so that the two DFA' become the same.

Introduction

Basic Idea

1. Find equivalent states
   1. Use table-filling algorithm
2. Partition equivalent states derived from table-filling algorithm into blocks and condense DFA.

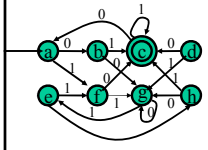
Definitions

- *Equivalent* – When two distinct states p and q can be replaced by a single state that behaves like both p and q.
- *Distinguishable* – When two states are not equivalent. That is, state p is distinguishable from state q if there is at least one string ω such that one of $\delta(p,\omega)$ and $\delta(q,\omega)$ is accepting, and the other is not accepting.

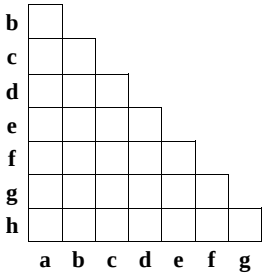
Table-filling Algorithm

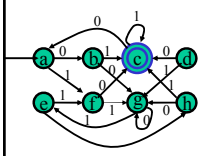
1. Initialize all entries as unmarked and with no dependencies.
2. Mark all pairs of a accepting and non-accepting state.
3. For each unmarked pair p,q and input symbol a:
   1. Let $r = \delta(p,a)$, $s = \delta(q,a)$.
   2. If (r,s) unmarked, add (p,q) to (r,s)'s dependencies,
   3. Otherwise mark (p,q), and recursively mark all dependencies of newly-marked entries.

Table-filling Example

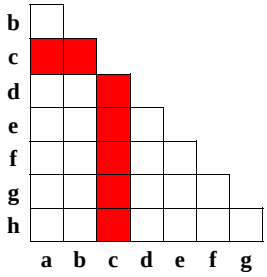


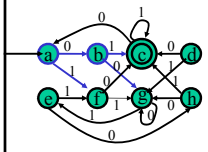
1. Initialize table entries:
   Unmarked, empty list



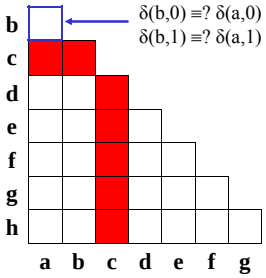


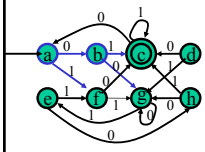
2. Mark pairs of final
   & non-final states



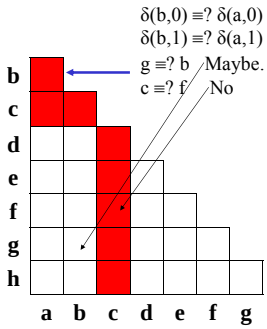


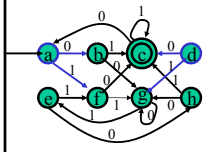
3. For each unmarked
   pair & symbol . . .



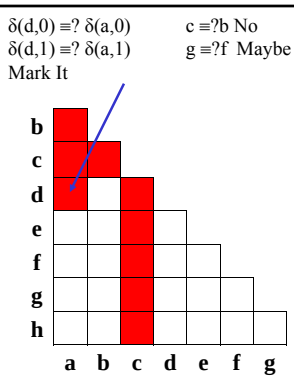


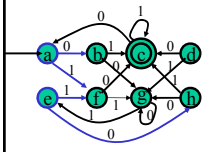
3. For each unmarked
   pair & symbol . . .



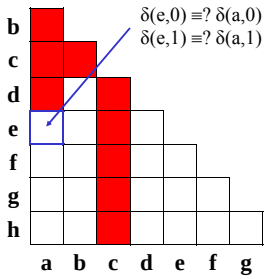


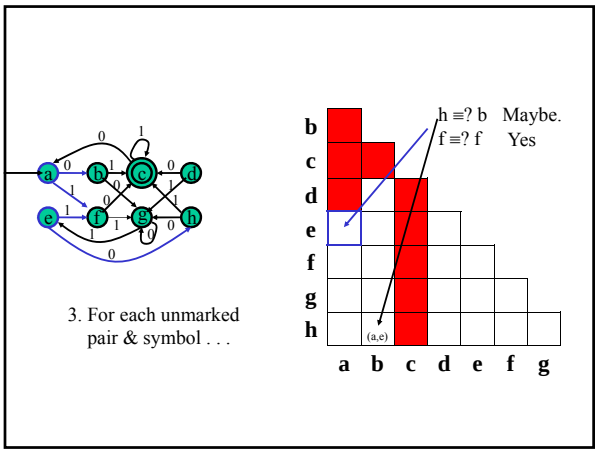
3. For each unmarked
   pair & symbol . . .

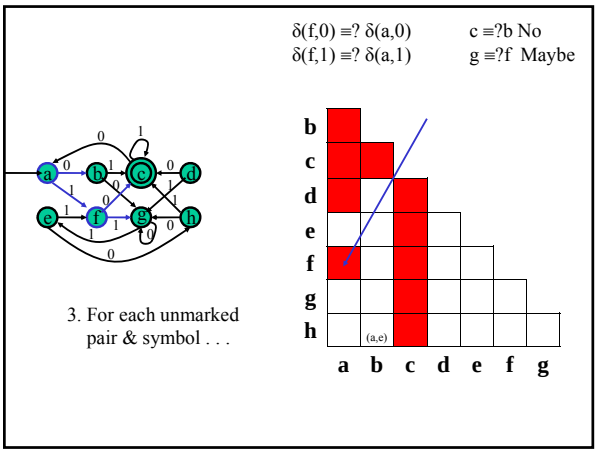


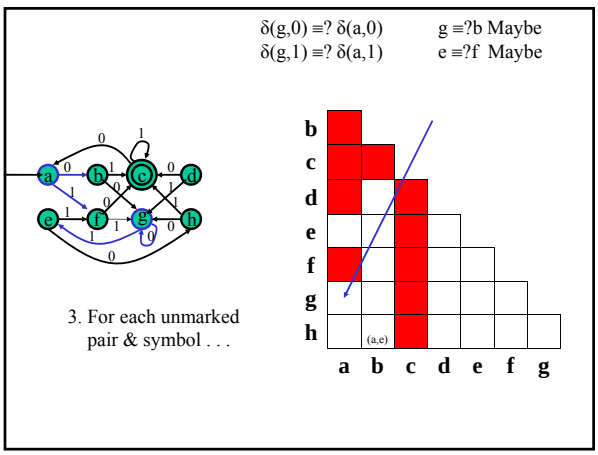


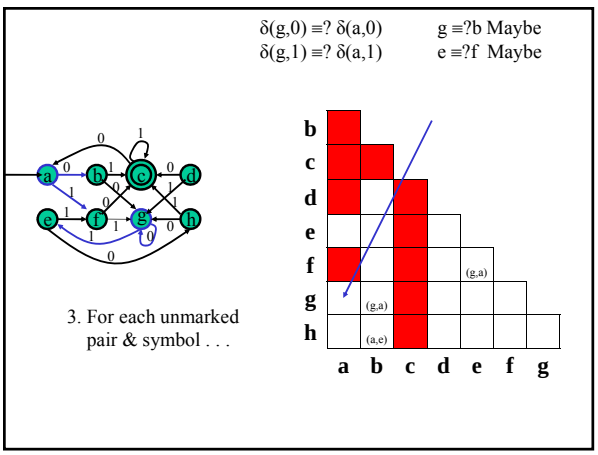
3. For each unmarked
   pair & symbol . . .

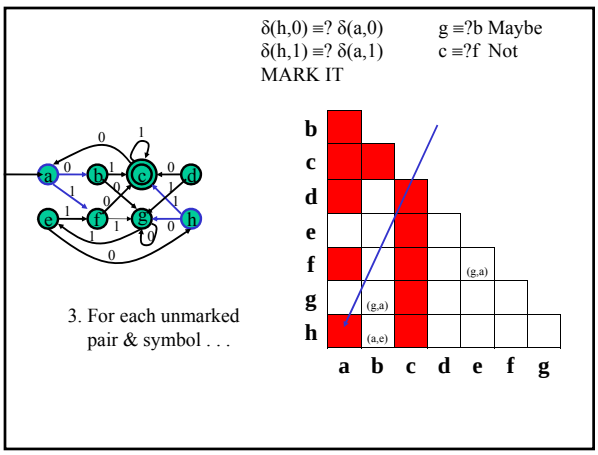


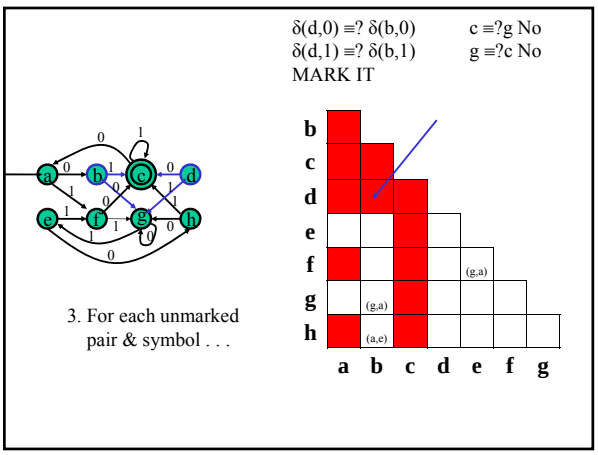

h ≡? b Maybe.
f ≡? f Yes
3. For each unmarked pair & symbol . . .
b
c
d
e
f
g
h
(a,e)
a b c d e f g


δ(f,0) ≡? δ(a,0) c ≡?b No
δ(f,1) ≡? δ(a,1) g ≡?f Maybe
3. For each unmarked pair & symbol . . .
b
c
d
e
f
g
h
(a,e)
a b c d e f g

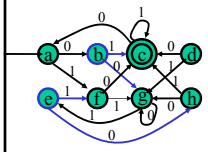

δ(g,0) ≡? δ(a,0) g ≡?b Maybe
δ(g,1) ≡? δ(a,1) e ≡?f Maybe
3. For each unmarked pair & symbol . . .
b
c
d
e
f
g
h
(a,e)
a b c d e f g

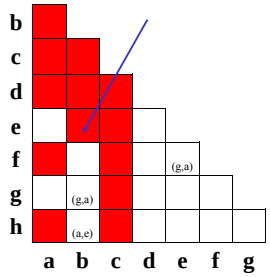

δ(g,0) ≡? δ(a,0) g ≡?b Maybe
δ(g,1) ≡? δ(a,1) e ≡?f Maybe
3. For each unmarked pair & symbol . . .
b
c
d
e
f
g
h
(g,a)
(g,a)
(a,e)
a b c d e f g


δ(h,0) ≡? δ(a,0) g ≡?b Maybe
δ(h,1) ≡? δ(a,1) c ≡?f Not
MARK IT
3. For each unmarked pair & symbol . . .
b
c
d
e
f
g
h
(g,a)
(g,a)
(a,e)
a b c d e f g


δ(d,0) ≡? δ(b,0) c ≡?g No
δ(d,1) ≡? δ(b,1) g ≡?c No
MARK IT
3. For each unmarked pair & symbol . . .
b
c
d
e
f
g
h
(g,a)
(g,a)
(a,e)
a b c d e f g

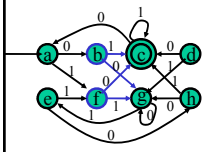
δ(e,0) ≡? δ(b,0) h ≡?g Maybe
δ(e,1) ≡? δ(b,1) f ≡?c No
MARK IT

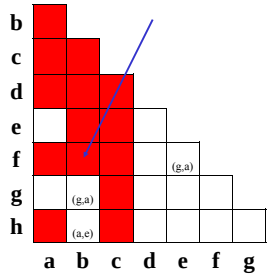




3. For each unmarked pair & symbol . . .

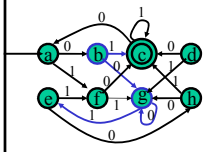
δ(f,0) ≡? δ(b,0) c ≡?g No
δ(f,1) ≡? δ(b,1) g ≡?c No
MARK IT

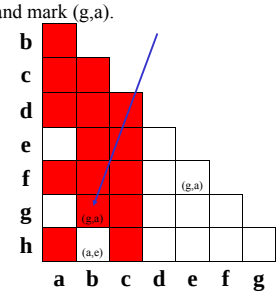




3. For each unmarked pair & symbol . . .

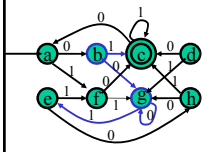
δ(g,0) ≡? δ(b,0) g ≡?g Yes
δ(g,1) ≡? δ(b,1) e ≡?c No
MARK IT – Recursively go back and mark (g,a).

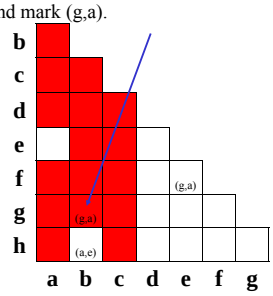




3. For each unmarked pair & symbol . . .

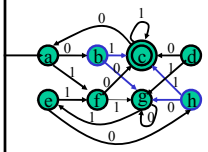
δ(g,0) ≡? δ(b,0) g ≡?g Yes
δ(g,1) ≡? δ(b,1) e ≡?c No
MARK IT – Recursively go back and mark (g,a).

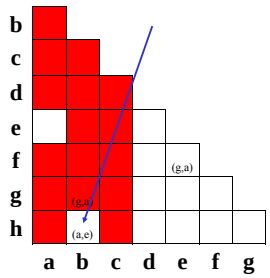




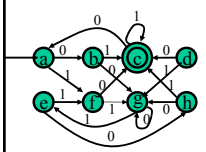
3. For each unmarked pair & symbol . . .

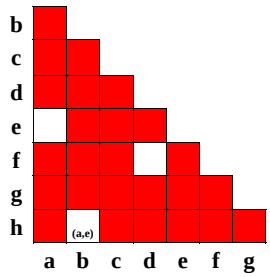
δ(h,0) ≡? δ(b,0) g ≡?g Yes
δ(h,1) ≡? δ(b,1) c ≡?c Yes
Don't Mark It





3. For each unmarked pair & symbol . . .

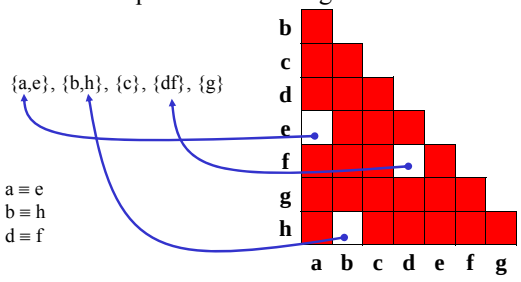


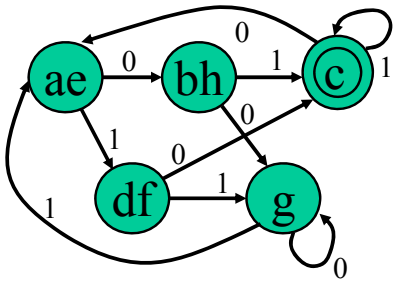


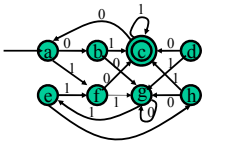
3. For each unmarked pair and symbol, . . .

Minimization of DFA

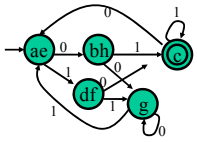
Partition equivalent states together.

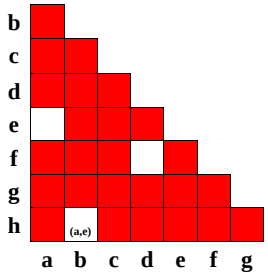






{a,e}, {b,h}, {c}, {df}, {g}





Citations

- J. E. Hopcroft, R. Motwani, and J. D. Ullman. "*Introduction to Automata Theory, Language, and Computation*". Addison--Wesley, 2nd edition, 2001.

Questions?