DFA Minimization and Applications

Monday, October 15, 2007
Reading: Stoughton 3.12

CS235 Languages and Automata

Department of Computer Science
Wellesley College

Goals for Today

- Answer any PS3 questions you might have.
- Some information about Thursday's midterm.
- DFA minimization
- Some applications of regular expressions and automata:
  - Automata design using Forlan
  - Efficient String Searching
  - Pattern Matching with Regular Expressions

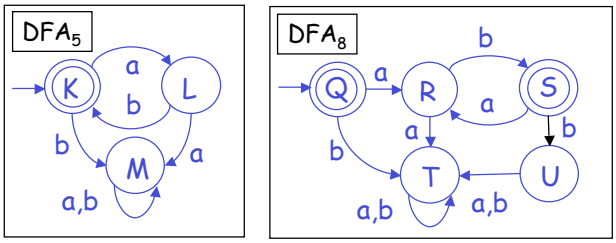
The Midterm

- The midterm will be an in-class exam during the lecture period on Thu. Oct. 18.
- It will cover material through today's lecture (#17) and problem sets 1 – 3.
- It is open book/open notes.
- Problems will be easier than most problem set problems.
- Study all the problem set solutions!
- You are encouraged to form study groups!
- Where to find practice problems?
  - Stoughton's course web site: Exercise Sets and Exams
  - Sipser Chapter 1 Exercises (not Problems), some of which have solutions

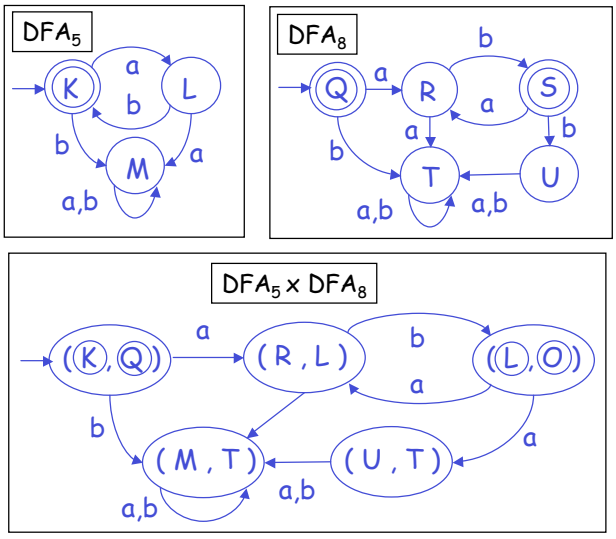
Midterm Problems you Should Prepare For

- A simple proof by induction.
- A simple proof by contradiction/diagonalization.
- Reasoning about logic, sets, relations, functions, bijections, and closures (reflexive, symmetric, transitive).
- Reasoning about sizes and containment of sets/languages.
- Reading/writing simple SML functions, possibly using the Forlan functions you've used in assignments.
- Understanding the formal definition of FAs and DFAs (i.e., a quadruple of states, start state, accepting state, and transitions).
- Simplifying regular expressions (only *very* simple simplifications)
- Converting between English descriptions of languages, regular expressions, FAs, EFAs, NFAs, DFAs.
- Operations involving products of DFAs: intersection, union, equivalence.

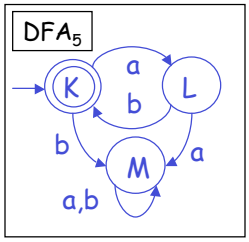
Are DFA_5 and DFA_8 Equivalent?

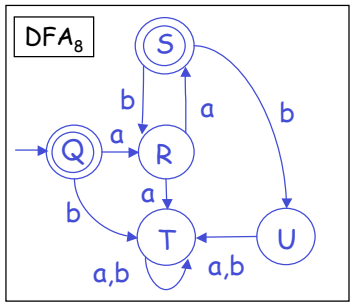


Are DFA_5 and DFA_8 Equivalent? Take Their Product!



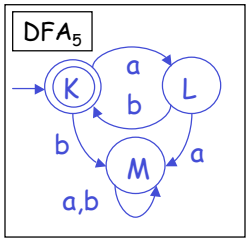
Minimal DFAs

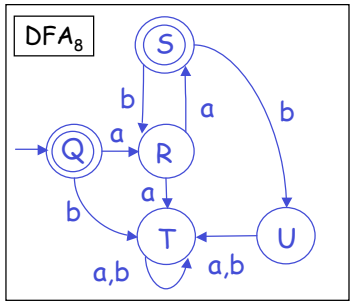




- A DFA is minimal if it has the smallest number of states of any DFA accepting its language.
- Is DFA_5 minimal?
- Is DFA_8 minimal?

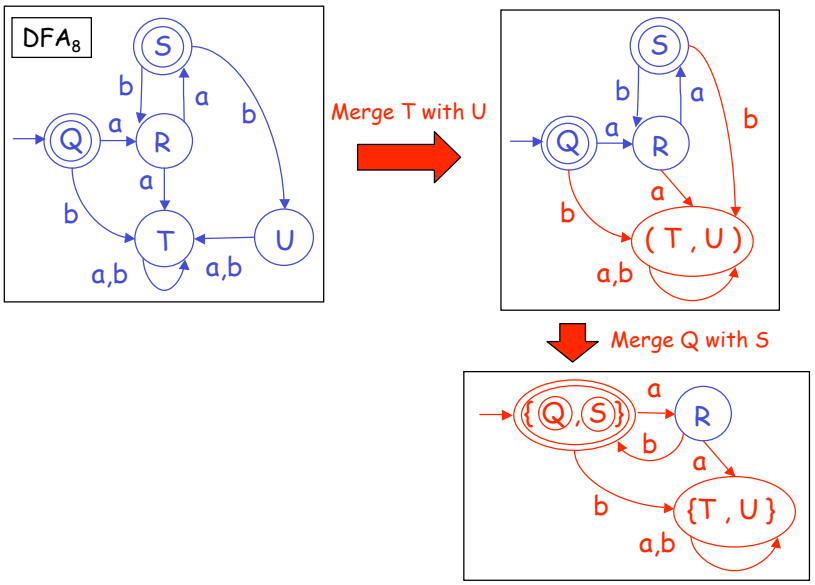
State Merging



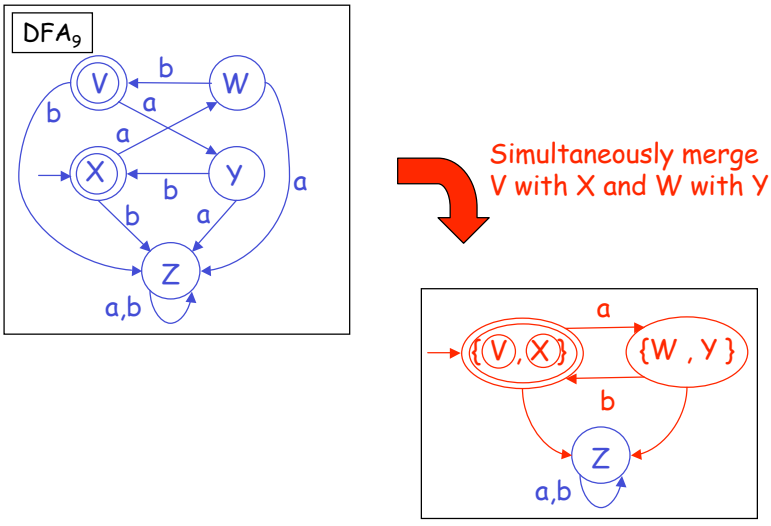


- A DFA is not minimal iff two states can be merged to form a single state without changing the meaning of the DFA.
- Accepting states and non-accepting states can never be merged.
- Which states in DFA_8 can be merged?

State Merging in DFA_8



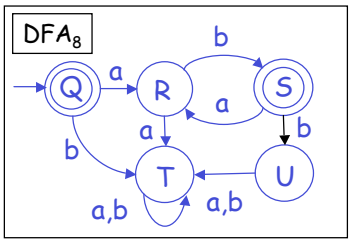
States Can't Always be Merged Iteratively



DFA Minimization Algorithm: Step 1

List all pairs of states than **must not** be merged
= pairs of one accepting and one non-accepting state.

Other pairs **might** be mergeable – they are considered mergeable until proven otherwise



Unmergeable

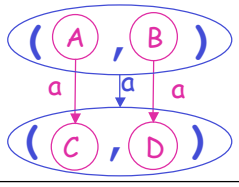
(Q, R), (R, Q),
(Q, T), (T, Q),
(Q, U), (U, Q),
(S, R), (R, S),
(S, T), (T, S),
(S, U), (U, S)

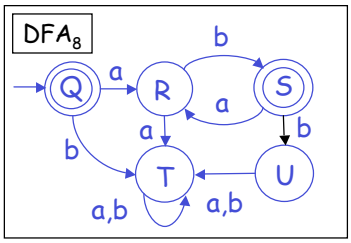
Might-Be-Mergeable

(Q, Q), (R, R), (S, S),
(T, T), (U, U),
(Q, S), (S, Q),
(R, T), (T, R),
(R, U), (U, R),
(T, U), (U, T)

DFA Minimization Algorithm: Step 2

Move from Might-Be-Mergeable to Unmergeable any pair (A,B) such that there is a transition to a (C,D) in Unmergeable:





Unmergeable

(Q, R), (R, Q), (R, U),
(Q, T), (T, Q), (U, R),
(Q, U), (U, Q), (R, T),
(S, R), (R, S), (T, R),
(S, T), (T, S),
(S, U), (U, S)

Might-Be-Mergeable

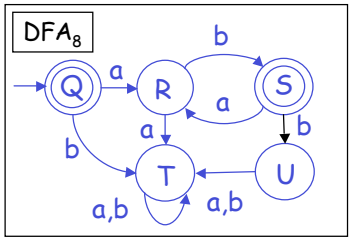
(Q, Q), (R, R), (S, S),
(T, T), (U, U),
(Q, S), (S, Q),
~~(R, T), (T, R),~~
~~(R, U), (U, R),~~
(T, U), (U, T)

DFA Minimization Algorithm: Step 3

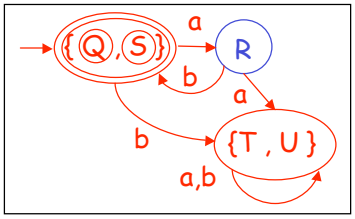
Merge the non-trivial pairs remaining int Might-Be-Mergeable

Might-Be-Mergeable

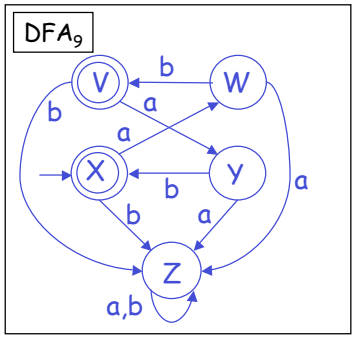
(Q, Q), (R, R), (S, S),
(T, T), (U, U),
(Q, S), (S, Q),
(T, U), (U, T)



Merge Q with S and T with U



DFA Minimization: More Practice



Designing DFAs with Forlan

Using Forlan, design a minimal DFA that accepts any string of as and bs that is one of the following:

- Any string beginning with two as.
- Any string ending in two bs.
- Any string containing the substring abab.
- Any string whose third-to-last character is b.