MCS-265: The Theory of Computation Handout #R2
San Skulrattanakulchai
Gustavus Adolphus College February 14, 2014

Designing Finite Automata

Sipser Ch 1: p41–44

There are two kinds of problems in finite automata (FA) design.

1. Given an FA, describe its language precisely.
2. Given a formal definition of a regular language, design a FA recognizing it.

Solving problems of the first kind is an art. No general principles are known.

We will practice on problems of the second kind. The idea is to work with state diagrams and follow these steps.

1. Get all the machine states. Each state should be describable precisely in English. There must not be overlap between any two states.
2. Pinpoint the final states.
3. Draw the edges.

Exercises Assume $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Design machines recoqnizing strings with the following properties:

(i) starts with `a`

(ii) ends with `bb`

(iii) starts with `a` and ends with `b`

(iv) starts with `a` or ends with `a`

(v) if it starts with `a`, then it doesn't end with `a`

(vi) doesn't start or end with `a` but `a` occurs somewhere

(vii) contains `ab`

(viii) contains `ab` and `ba`

(ix) contains `ab` or `ba`

(x) negate each of the above

Remarks

1. We can always get the complement of a regular language simply by swapping final and nonfinal states. Does this technique work for NFAs?

2. The Product Construction can be used to obtain the language of binary set operations (like union, intersection, set difference, symmetric difference, etc) of two regular languages.