Software Specifications
Minimizing Deterministic State Diagrams

Cain Susko

Queen's University
School of Computing

February 4, 2022

Minimization

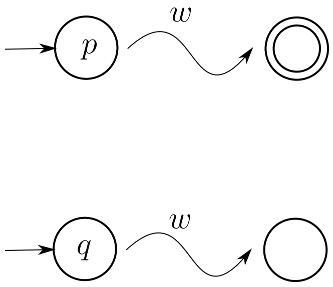
A regular language can be represented by many (very) different Deterministic State Diagrams. Minimization of a Regex or Non-Deterministic State Diagram is intractable–which is to say there is on efficient algorithm to minimize these.

However, because a regular language has a *unique* DFA with a smallest number of states, there is an efficient algorithm to minimize this diagram.

Distinguishable States

We must define that two states p, q of a DFA are **distinguishable**.

If there exists a string w such that it takes p to final state and q to a non-final state; or vice versa.



p, q are **indistinguishable** if and only if for any string w:

- w takes p to final state
- w takes q to a final state

Otherwise, they are Distinguishable

Idea of Algorithm

the main operation of the minimization algorithm is to find all pairs of **distinguishable** states. The reason for this is that, the only way to verify that any states are distinguishable, one must compare 2 states and only 2 states. Any states that are pairwise indistinguishable can be merged into one state (that is, a pair of states that are indistinguishable).

Preprocessing for Algorithm

before performing the minimization algorithm, one must eliminate all states that cannot be reached from the starting state by any string. This is done by using a state reachability algorithm.