More Finite Automata.

CS154
Chris Pollett
Feb. 8, 2006.

Outline

- Closure under Union
- Nondeterministic Finite Automata
- Formal Definition
- Equivalence

Closure under Union

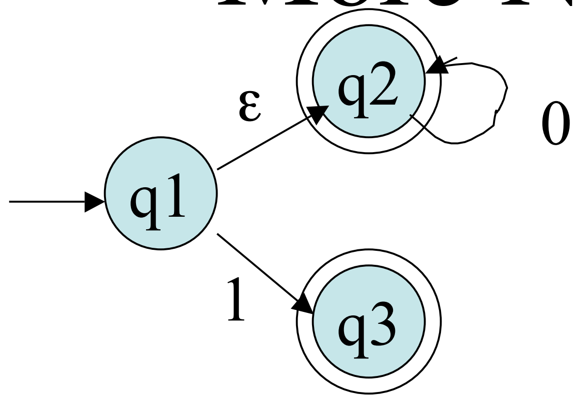
Theorem If A_1 and A_2 two regular languages, so is their union $A_1 \cup A_2$.

Proof: Let $M_1=(Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 =(Q_2, \Sigma, \delta_2, q_2, F_2)$ be the DFAs recognizing A_1 and A_2. We would like make a new DFA, M, which simultaneously simulates both M_1 and M_2 and accepts a string w if either of M_1 and M_2 accepts. To simulate both machines at the same time we use a so-called cartesian product construction. Let $Q = Q_1 \times Q_2$. M's alphabet is Σ like that of M_1 and M_2. Define $\delta((q, q'), a) = (\delta_1(q,a), \delta_2(q',a))$. Let the start state be (q_1, q_2). Finally, let $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

Nondeterminism

- It seems harder to use a similar technique as the last slide to show that the regular language are closed under concatenation.
- This motivates why we'll consider another model of finite automata called nondeterministic finite automata (NFA) a which are slightly more flexible. We'll eventually show the two models are equivalent.
- In a deterministic finite automata, in each state reading a fixed symbols there is only one possible next state. Nondeterministic finite automata relax this condition and allow several possible next states, they are allow transition on the empty string.

More NFA motivation



- Notice we can have more than one transition out of a state, we can have ε-transitions, and we don't need to have a transition from every alphabet symbol from a state.
- We say the NFA accepts w roughly if there is some sequence of transitions beginning with the start state, that processes each character of w and ends in an accept state.
- For instance, the machine above accept ε, 0, 00, 000, 1; but rejects 01, 11, 0001. It rejects 01 because although it can get to state q2 after seeing ε0 = 0, it has nowhere to go when it sees a 1 so it can't process the 1 so it rejects. No other path in the machine processes 01 even this far.

Formal Definition of an NFA

- Recall the power set of a set Q, P(Q), is the set of all subsets of Q.
- A **nondeterministic finite automaton** is a 5-tuple (Q, Σ, δ, q, F) where
  1. Q is a finite set of states,
  2. Σ is an alphabet,
  3. δ: Q x Σ ∪{ε} --> P(Q) is the transition function,
  4. $q_0 \in Q$ is the start state, and
  5. $F \subseteq Q$ is the set of accept states.

Example

- The machine a couple slides back is defined as (Q, Σ, δ, q1, F) where
  1. Q={q1, q2, q3}
  2. Σ = {0, 1}
  3. δ is given by:

     δ(q1, ε)--> {q2}   δ(q2, ε)--> {}   δ(q3, ε)--> {}

     δ(q1, 0)--> {}   δ(q2, 0)--> {q2}   δ(q3, 0)--> {}

     δ(q1, 1) --> {q3}   δ(q2, 1)--> {}   δ(q3, 1)--> {}
  4. q1 is the start state
  5. F = {q2, q3}

Formal Definition of Accepts

- We say M **accepts** $w = y_1 \ldots y_n$, where each y_i is in $\Sigma \cup \{\varepsilon\}$, if there exists a sequence of states $r_0, r_1, \ldots r_n$ such that:
  1. $r_0 = q_0$
  2. $r_{i+1} \in \delta(r_i, y_{i+1})$, for $i=0, \ldots, n-1$
  3. $r_n \in F$.

Equivalence of NFAs and DFAs

Theorem Any language recognized by an NFA is recognized by some DFA.

Proof: Given an NFA N= (Q, Σ, δ, q, F) we want to simulate how it acts on a string w with a DFA, M= (Q', Σ, δ', q', F'). The idea is we want to keep track of what possible states it could be in after reading the first m characters of w. Let Q'= P(Q). The alphabet is the same. For each R∈Q' and a ∈ Σ, let δ'(R,a) = {q ∈Q | q ∈ E(δ(r,a)) for some r ∈R}. Here E(q') is the set of states reachable from q' following only ε transitions. Let q'={q}. Let F' = {R ∈Q'| R contains an accept state of N}.