

Finite State Machines

Thursday March 20

Motivation – where?

- Regular expressions : generalized search (and replace)
 - Editors, sed/awk, perl
- Simple parsers for languages
- Control systems of all sorts
 - GUI interaction model
 - Elevator, vending machine, car-building robot
- Compilers, interpreters

Motivation – why

- Provides a simple model of computation
 - Lots of nice results
 - Most are easy to prove!
 - **But:** only so powerful
- Occurs a lot
 - And not just in computer science!

What

- Informal examples (blackboard)
- Notice:
 - Start state
 - End state(s)
 - Nodes
 - (labeled) transitions

Formal definition

- A **Finite State Machine** (FSM) is a tuple $(Q, \Sigma, \delta, s, F)$ where
 - Q – alphabet of state symbols
 - Σ – alphabet of input symbols
 - $\delta : Q \times \Sigma \rightarrow Q$ - transition function (total or partial)
 - $s \in Q$ – starting state
 - $F \subseteq Q$ – final state(s)

Full Example

- See blackboard again

FSM as computation

- (see board drawing)
- FSM with input x :
 - String x placed on tape one ‘letter’ per cell
 - Read head on leftmost cell
 - Current state = s
 - FSM started

FSM as computation (cont)

- Execute cycle:
 - Symbol under head read (current symbol). If no symbol, terminate.
 - Compute next state (using current state and current symbol). If next state is undefined abort – only occurs if state function partial
 - Move head right
 - Current state := next state
- Repeat

FSM as computation (cont)

- Note that tape is read-only, one-way and finite
- Note that controller ‘memory’ also finite, but read/write
- Additionally: define the “output” of a FSM to be its last state (including undefined if necessary)

Languages

- Definition: Let M be a FSM. We say that M accepts a language L (of finite strings from M 's input alphabet Σ) if and only if
 - For all t in L , FSM on t ends in a final state
 - For all t not in L , FSM on t does not end in a final state

Motivation revisited

- Regular expressions correspond exactly to languages accepted by FSM
- **Most** control systems are FSM!
 - Exposing FSM of a GUI controller leads to much simpler explanation, design, coding, ...