

Lecture 16

Finite State Machines (continued)

Kenneth M. Anderson
Foundations of Software Engineering
CSCI 5828 - Spring Semester, 1999

Today's Lecture

- Continue to explore issues related to Finite State Machines
- Present a FSM-like language called SDL
- Discuss Homework 2

Advantages of FSM Model

- Simple
- Obvious graphical representation
- Easy to Build Support Tools
 - Transformers
 - Transform FSM Model into other representations
 - Analyzers
 - Will this FSM run forever? Is it possible for it to halt? Are the state sequences infinite?

Shortcomings of FSM Model

- Theoretical Limit on Computational Power
 - FSM has no “memory”
- State Space Explosion for Large Problems
- State Space Explosion for Composed FSMs
 - States are multiplicative
- Inherently Synchronous
 - FSM in single, global state at each time instant

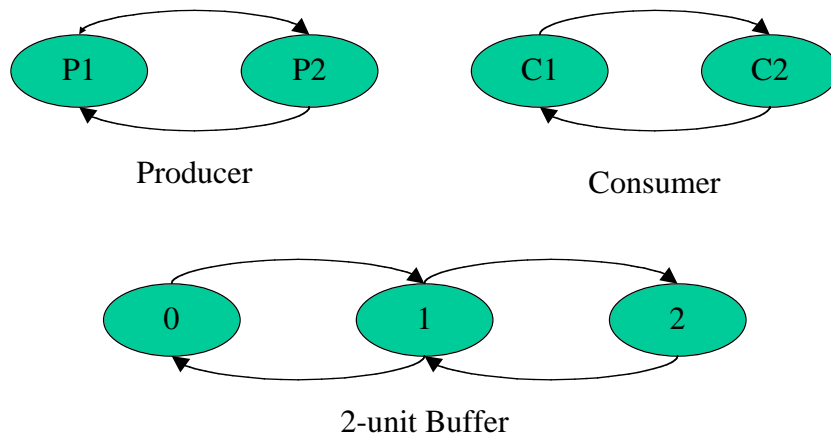
Levels of Complexity

- Turing Machine
 - Unbounded tape
- Linear-Bounded Automata
 - Bounded tape
- Push-Down Automata
 - stack
- Finite State Machines
 - limited computational power but its simple to understand and program
- Programming Languages
 - Execution Semantics
- Context Sensitive Langs.
 - Language Semantics
- Context Free Grammars
 - Syntax
- Regular Expressions
 - Lexical Structure

An FSM-Based Tool: SDL

- Used Widely for Telephony Applications
- Extended FSMs
 - Modularity
 - Channel
- Tools
 - Analysis
 - Simulation
 - Code-generation

Producer/Consumer Example



Homework 2

- Use a Finite State Machine and a Petri Net to specify the cruise control system of an automobile
 - Retrieve the assignment from the Website
 - Start on the FSM part now! (Don't wait!)
 - We will discuss Petri Nets next week
 - You will have the weekend to work on the Petri Net part before turning the assignment in