CSE 311: Foundations of Computing

Fall 2013

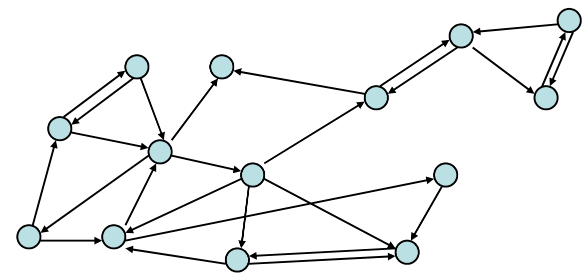
Lecture 22: Finite state machines (aka DFAs)

highlights: directed graphs

G = (V, E)   V – vertices
E – edges, order pairs of vertices

Path: $v_0, v_1, \ldots, v_k$, with (v_i, v_{i+1}) in E

Simple Path
Cycle
Simple Cycle



highlights: relations

Let R be a relation on a set A. There is a path of length n from a to b if and only if $(a,b) \in R^n$

Let R be a relation on a set A. The connectivity relation R* consists of the pairs (a,b) such that there is a path from a to b in R.

Transitive-Reflexive closure: Add the minimum possible number of edges to make the relation transitive and reflexive

The transitive-reflexive closure of a relation R is the connectivity relation R*

n-ary relations

Let $A_1, A_2, \ldots, A_n$ be sets. An **n-ary** relation on these sets is a subset of $A_1 \times A_2 \times \cdots \times A_n$.

relational databases

STUDENT

Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Von Neuman	481080220	555	3.78
Russell	238082388	022	3.85
Einstein	238001920	022	2.11
Newton	1727017	333	3.61
Karp	348882811	022	3.98
Bernoulli	2921938	022	3.21

relational databases

STUDENT

Student_Name	ID_Number	Office	GPA	Course
Knuth	328012098	022	4.00	CSE311
Knuth	328012098	022	4.00	CSE351
Von Neuman	481080220	555	3.78	CSE311
Russell	238082388	022	3.85	CSE312
Russell	238082388	022	3.85	CSE344
Russell	238082388	022	3.85	CSE351
Newton	1727017	333	3.61	CSE312
Karp	348882811	022	3.98	CSE311
Karp	348882811	022	3.98	CSE312
Karp	348882811	022	3.98	CSE344
Karp	348882811	022	3.98	CSE351
Bernoulli	2921938	022	3.21	CSE351

What's not so nice?

relational databases

STUDENT

Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Von Neuman	481080220	555	3.78
Russell	238082388	022	3.85
Einstein	238001920	022	2.11
Newton	1727017	333	3.61
Karp	348882811	022	3.98
Bernoulli	2921938	022	3.21

TAKES

ID_Number	Course
328012098	CSE311
328012098	CSE351
481080220	CSE311
238082388	CSE312
238082388	CSE344
238082388	CSE351
1727017	CSE312
348882811	CSE311
348882811	CSE312
348882811	CSE344
348882811	CSE351
2921938	CSE351

Better

database operations: projection

Find all offices: $\Pi_{\text{Office}}(\text{STUDENT})$

Office
022
555
333

Find offices and GPAs: $\Pi_{\text{Office,GPA}}(\text{STUDENT})$

Office	GPA
022	4.00
555	3.78
022	3.85
022	2.11
333	3.61
022	3.98
022	3.21

database operations: selection

Find students with GPA > 3.9 : $\sigma_{GPA>3.9}(STUDENT)$

Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Karp	348882811	022	3.98

Retrieve the name and GPA for students with GPA > 3.9:
$\Pi_{Student\_Name,GPA}(\sigma_{GPA>3.9}(STUDENT))$

Student_Name	GPA
Knuth	4.00
Karp	3.98

database operations: natural join

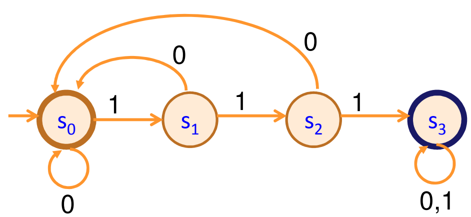
Student ⋈ Takes

Student_Name	ID_Number	Office	GPA	Course
Knuth	328012098	022	4.00	CSE311
Knuth	328012098	022	4.00	CSE351
Von Neuman	481080220	555	3.78	CSE311
Russell	238082388	022	3.85	CSE312
Russell	238082388	022	3.85	CSE344
Russell	238082388	022	3.85	CSE351
Newton	1727017	333	3.61	CSE312
Karp	348882811	022	3.98	CSE311
Karp	348882811	022	3.98	CSE312
Karp	348882811	022	3.98	CSE344
Karp	348882811	022	3.98	CSE351
Bernoulli	2921938	022	3.21	CSE351

finite state machines

- States
- Transitions on inputs
- Start state and final states
- The language recognized by a machine is the set of strings that reach a final state

State	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_3
s_3	s_3	s_3



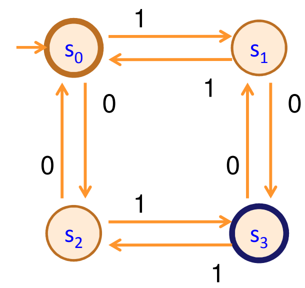
applications of FSMs (a.k.a. finite automata)

- Implementation of regular expression matching in programs like `grep`
- Control structures for sequential logic in digital circuits
- Algorithms for communication and cache-coherence protocols
  - Each agent runs its own FSM
- Design specifications for reactive systems
  - Components are communicating FSMs

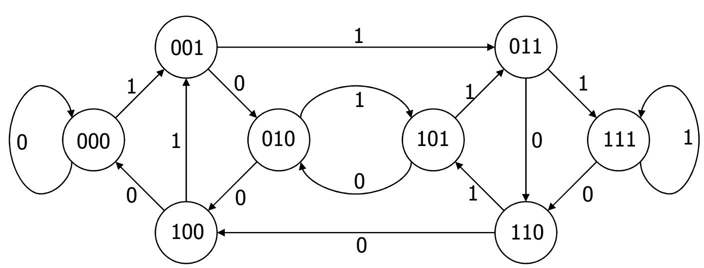
applications of FSMs (a.k.a. finite automata)

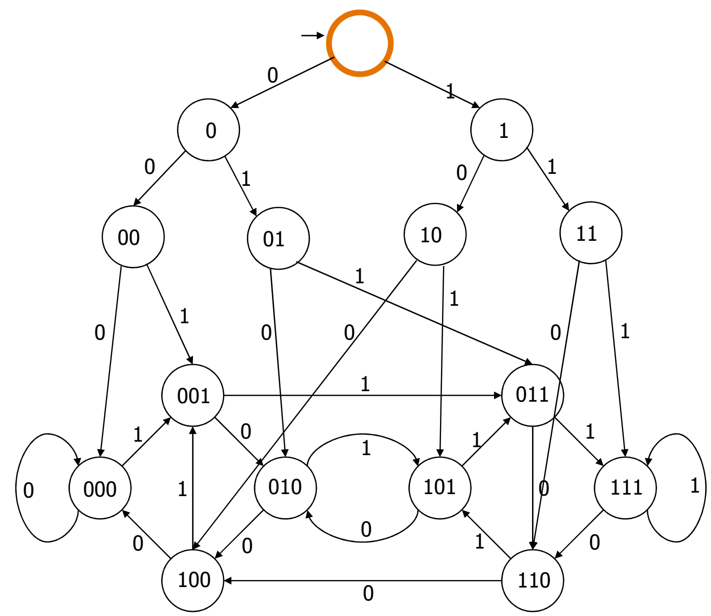
- Formal verification of systems
  - Is an unsafe state reachable?
- Computer games
  - FSMs provide worlds to explore
- Minimization algorithms for FSMs can be extended to more general models used in
  - Text prediction
  - Speech recognition

what language does this machine recognize?



3 bit shift register





goal: FSM that accepts strings with a 1 three positions from the end

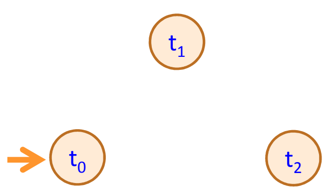
question

How does the size of a DFA to recognize "10^{th} character is a 1" compare with the size of a DFA to recognize "10^{th} character from the end is 1"?

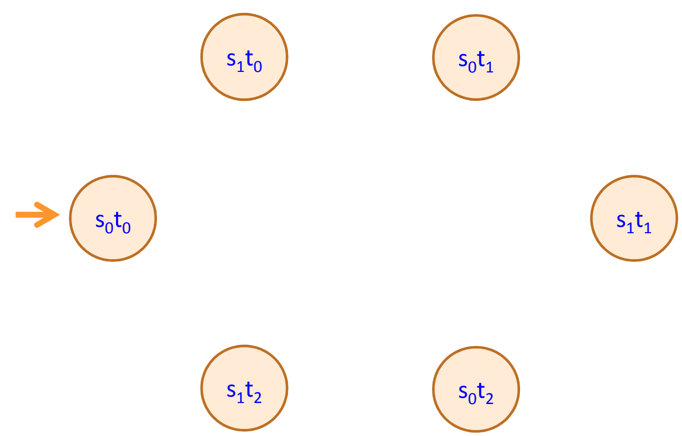
strings over {0, 1, 2}*

M_1: Strings with an even number of 2's

M_2: Strings where the sum of digits mod 3 is 0



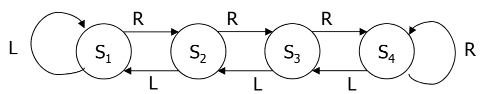
strings with an even number of 2's and a mod 3 sum of 0



state machines with output

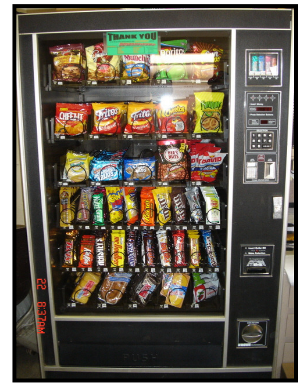
	Input		Output
State	L	R	
s_1	s_1	s_2	Beep
s_2	s_1	s_3	
s_3	s_2	s_4	
s_4	s_3	s_4	Beep

"Tug-of-war"

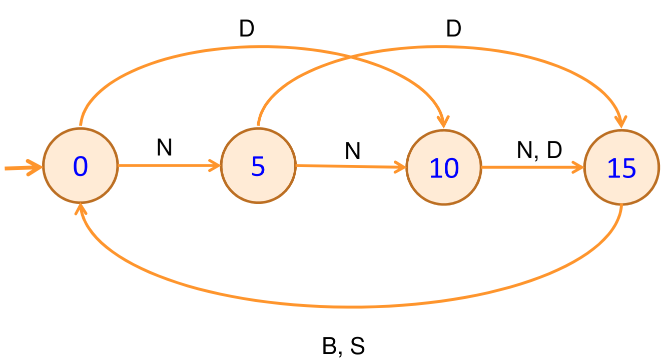


vending machine

Enter 15 cents in dimes or nickels
Press S or B for a candy bar

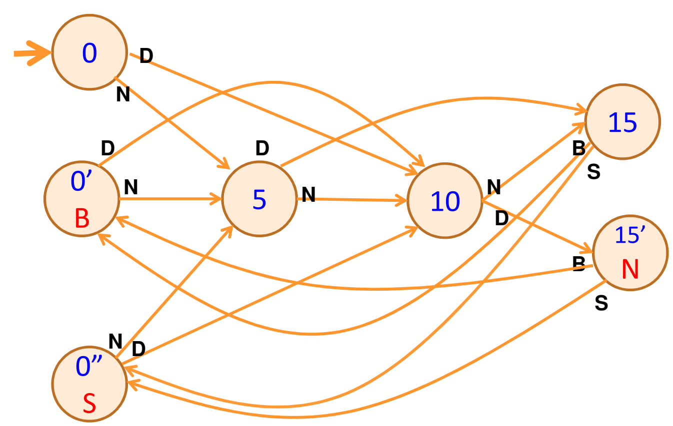


vending machine, v0.1



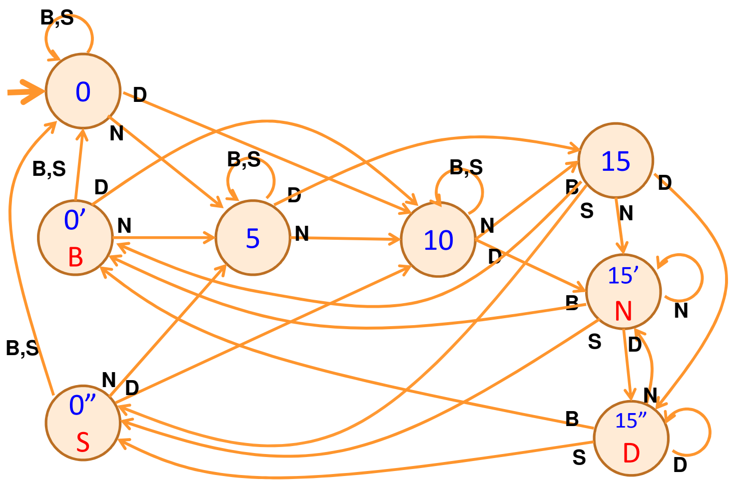
Basic transitions on N (nickel), D (dime), B (butterfinger), S (snickers)

vending machine, v0.2



Adding output to states: N – Nickel, S – Snickers, B – Butterfinger

vending machine, v1.0



Adding additional "unexpected" transitions