CSE 311 Foundations of Computing I

Lecture 23

Finite State Machines

Autumn 2012

Announcements

- Reading assignments
  - 7th Edition, Sections 13.3 and 13.4
  - 6th Edition, Section 12.3 and 12.4
  - 5th Edition, Section 11.3 and 11.4

Lecture highlights: Database and relations

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

Databases consist of collection of n-ary relations

Relational algebra: $\bowtie, \Pi, \sigma$

Relational Databases: Keys

- An attribute is a **key** if all its values in the database are always distinct

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

Which attribute is the key?
Why is Student_Name not a key?

Relational Databases: Relationships

STUDENT

Student_Name	ID_Number	Office
Knuth	328012098	022
Von Neuman	481080220	555
Russell	238082388	022
Einstein	238001920	022
Newton	1727017	333
Karp	348882811	022
Bernoulli	2921938	022

PROJECT

PRJ_ID	Project_Name	Due_date
P331	"Flying cyphers"	11/2012
P004	"Virtual induction"	12/2012
P901	"Binary bots"	12/2012

WORKS_ON is a relationship between students and project

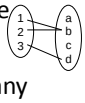
ID_Number	PRJ_ID
2921938	P004
2921938	P901
1727017	P901

Who works on what?

Types of Relationships in Relational Databases

- one-one:

- many-one



- many-many

What type is WORKS_ON?

ID_Number	PRJ_ID
2921938	P004
2921938	P901
1727017	P901

Finite state machines

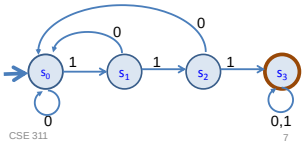
States

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



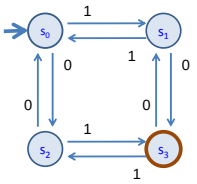
Applications of Finite State Machines (a.k.a. Finite Automata)

- 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

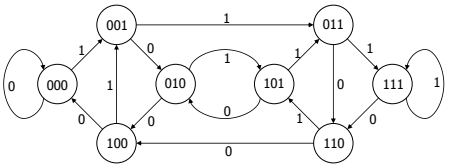
Applications of Finite State Machines (a.k.a. Finite Automata)

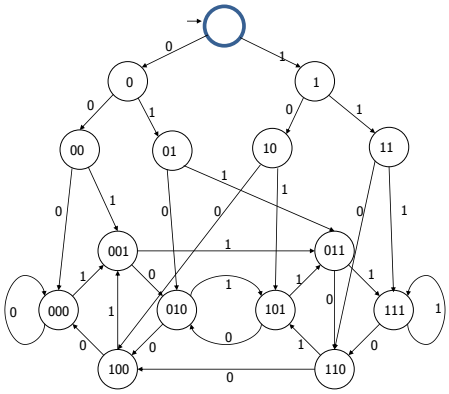
- Formal verification of systems
  - Is an unsafe state reachable?
- Computer games
  - FSMs provide worlds to explore
- Minimization algorithms for FSMs can be extended to more general models used in
  - Text prediction
  - Speech recognition

What language does this machine recognize?



3 Bit Shift register





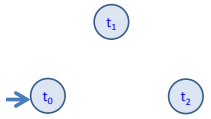
Design a DFA that accepts strings with a 1 three positions from the end

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

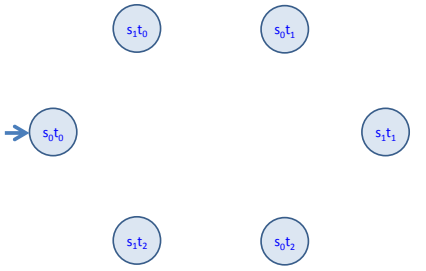
Strings over {0, 1, 2}*

M_1: Strings with an even number of 2's

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



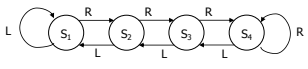
Recognize strings with an even number of 2's and a mod 3 sum of 0



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

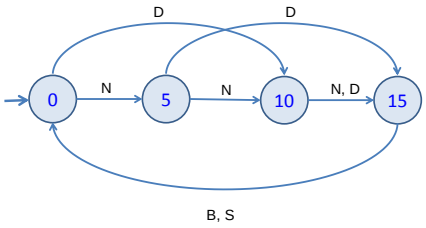
"Tug-of-war"



Vending Machine

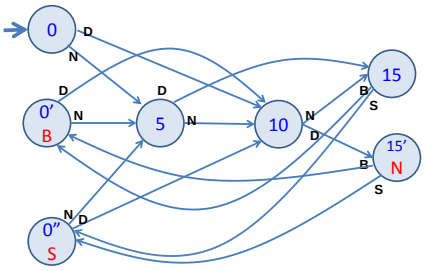
Enter 15 cents in dimes or nickels
Press S or B for a candy bar

Vending Machine, Version 1



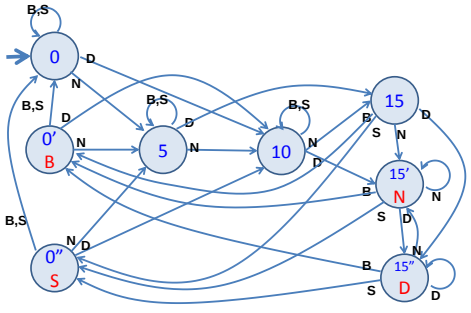
Basic transitions on N (nickel), D (dime), B (butterfinger), S (snickers)

Vending Machine, Version 2



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

Vending Machine, Final Version



Adding additional "unexpected" transitions