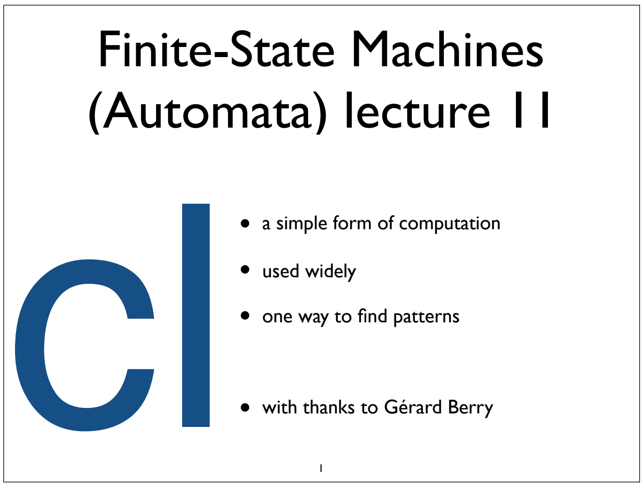
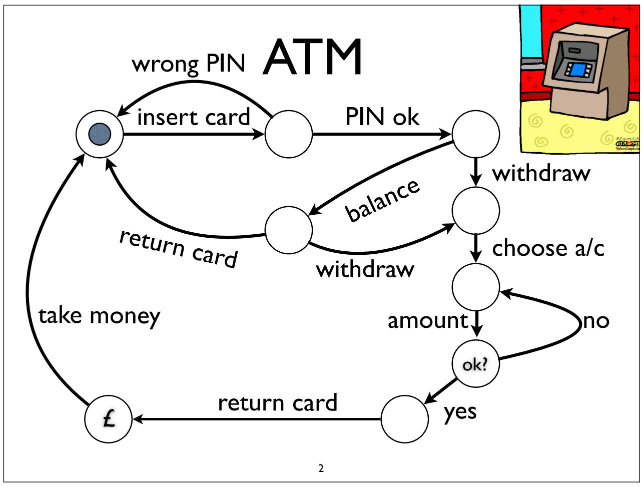


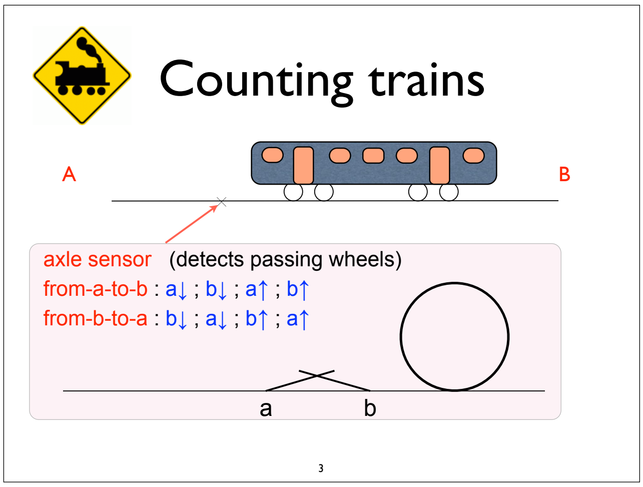
Finite-State Machines (Automata) lecture 11

- a simple form of computation
- used widely
- one way to find patterns
- with thanks to Gérard Berry



ATM

wrong PIN
insert card
PIN ok
withdraw
balance
return card
withdraw
choose a/c
take money
amount
no
ok?
yes
£
return card

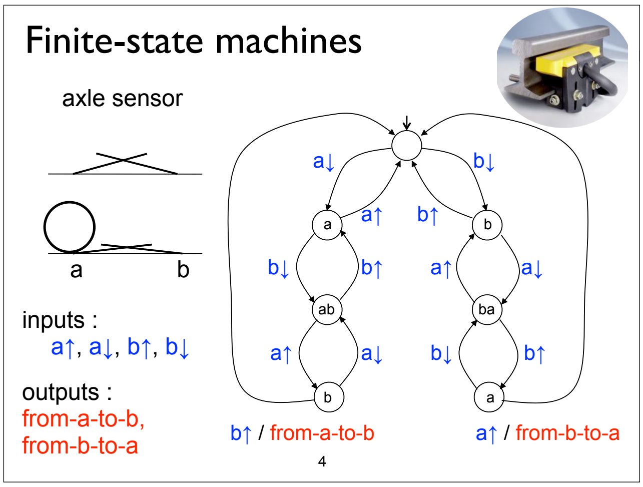


Counting trains

axle sensor (detects passing wheels)

from-a-to-b : a↓ ; b↓ ; a↑ ; b↑

from-b-to-a : b↓ ; a↓ ; b↑ ; a↑



Finite-state machines

axle sensor

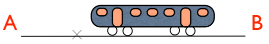
inputs :
a↑, a↓, b↑, b↓

outputs :
from-a-to-b,
from-b-to-a

b↑ / from-a-to-b

a↑ / from-b-to-a

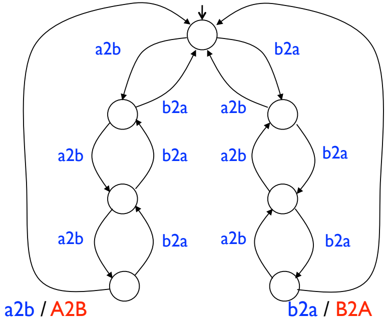
Hierarchical FSMs



carriage counter

inputs :
a2b, b2a

outputs :
A2B, B2A



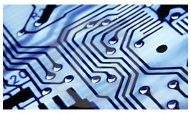
a2b = from-a-to-b

Application Fields

Industry

- real-time control, vending machines, cash dispensers, etc.

Electronic circuits



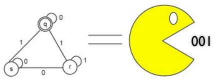
- data path / control path
- memory / cache handling
- protocols, USB, etc.

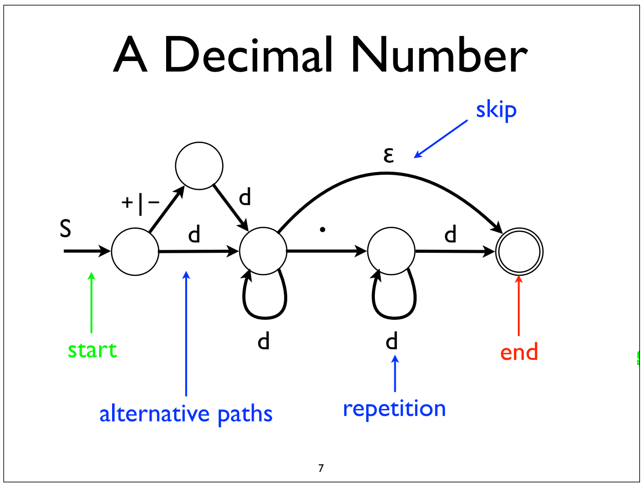
Communication protocols

- initiation and maintenance of communication links
- error detection and handling, packet retransmission

Language analysis

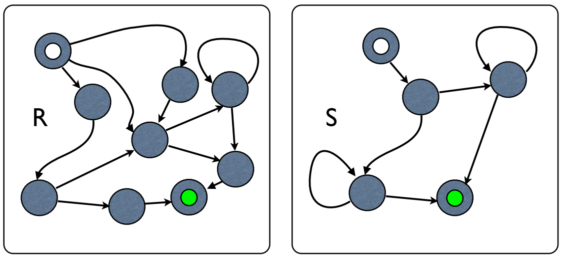
- natural languages
- programming languages
- search engines



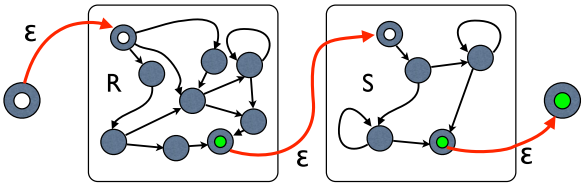


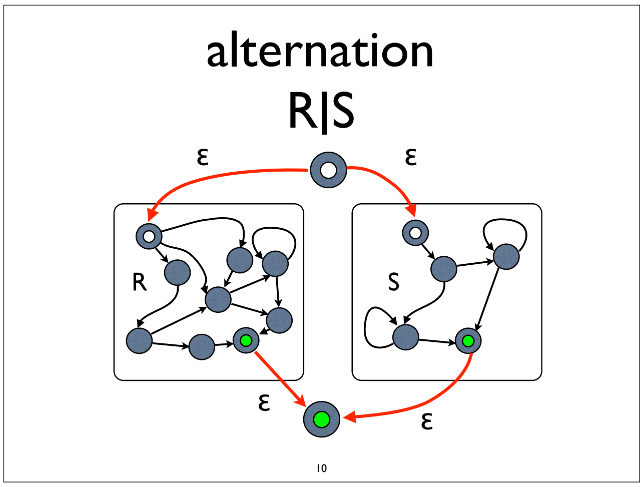
A Decimal Number

finite state machines



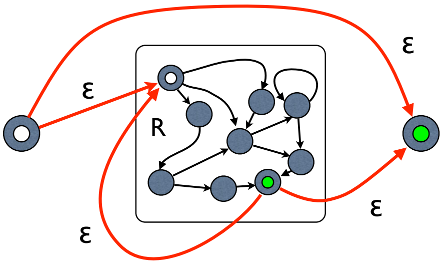
sequence
RS

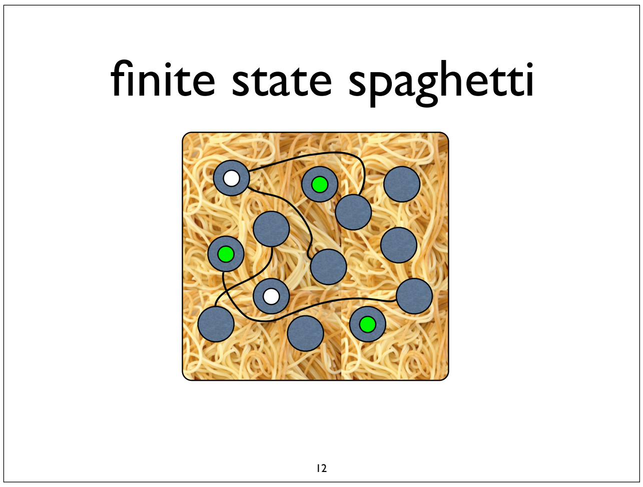




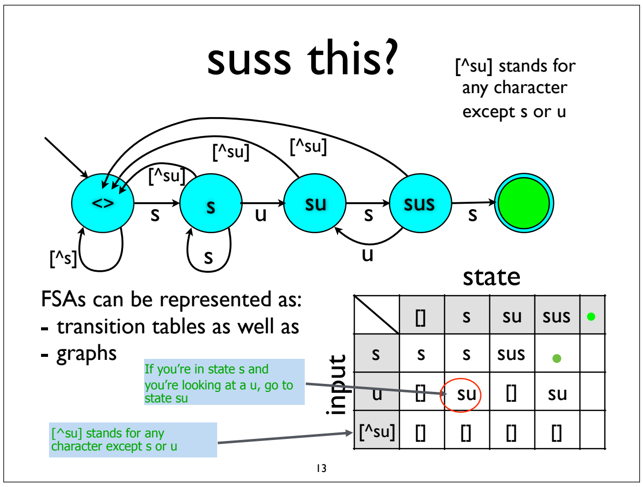
alternation
R|S

iteration
R*





finite state spaghetti



suss this?

[^su] stands for any character except s or u

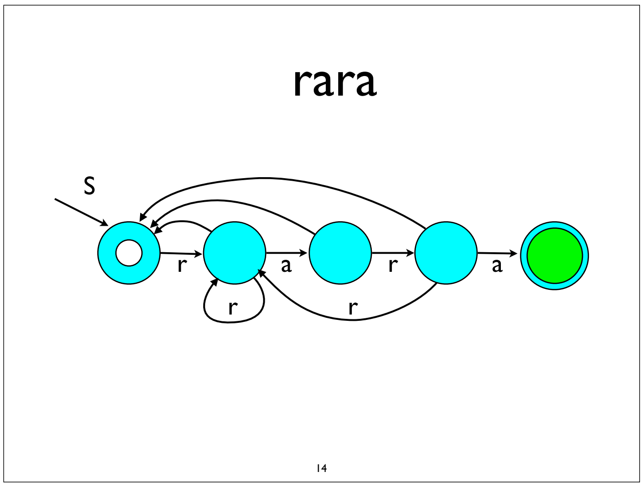
FSAs can be represented as:
- transition tables as well as
- graphs

If you're in state s and you're looking at a u, go to state su

[^su] stands for any character except s or u

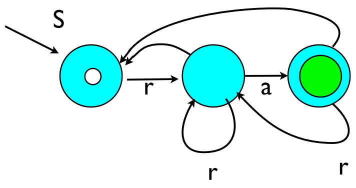
state

input	▯	s	su	sus	●
s	s	s	sus	●	
u	▯	su	▯	su	
[^su]	▯	▯	▯	▯	

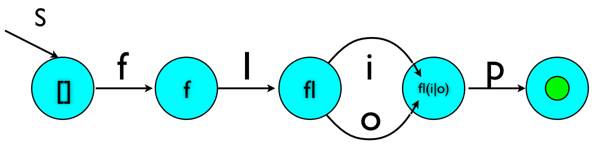


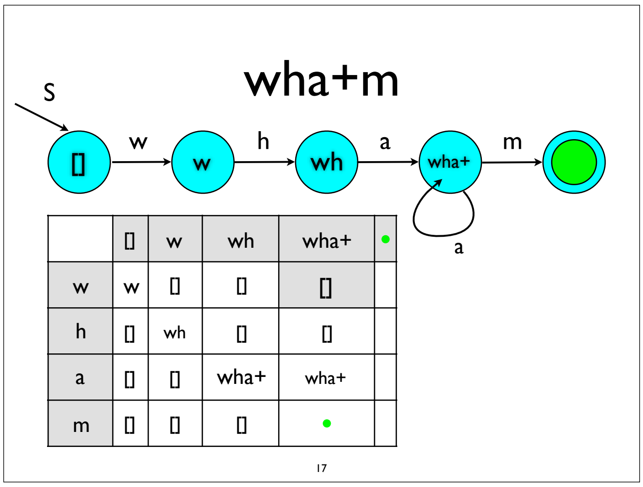
rara

ra(ra)*



(flip)|(flop) = fl(i|o)p





wha+m

	▯	w	wh	wha+	●
w	w	▯	▯	▯	
h	▯	wh	▯	▯	
a	▯	▯	wha+	wha+	
m	▯	▯	▯	●	