turing machines review

thursday 27 july 2000

lecture overview:

- state diagrams
- finite automata
- turing machines

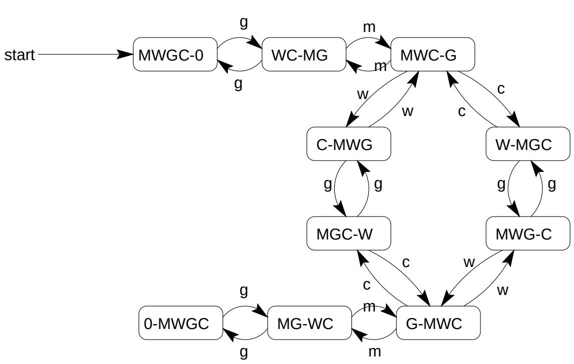
state diagram: example I

A man with a wolf, goat, and cabbage is on the left bank of a river. There is a boat large enough to carry the man and only one of the other three. The man and his entourage wish to cross to the right bank, and the man can ferry each across, one at a time. However, if the man leaves the wolf and goat unattended on either shore, the wolf will surely eat the goat. Similarly, if the goat and cabbage are left unattended, the goat will eat the cabbage.

How can everyone get across the river (without being eaten)?

from [Hopcroft & Ullman, 1979], p.14-15.

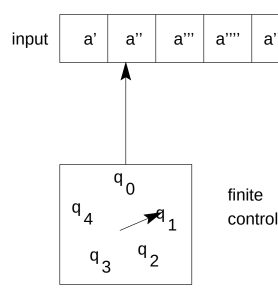
state diagram: example I, continued



key:

M = man	m = man rowing boat alone
W = wolf	w = man rowing boat with wolf
G = goat	g = man rowing boat with goat
C = cabbage	c = man rowing boat with cabbage
0 = empty shore	

finite automata



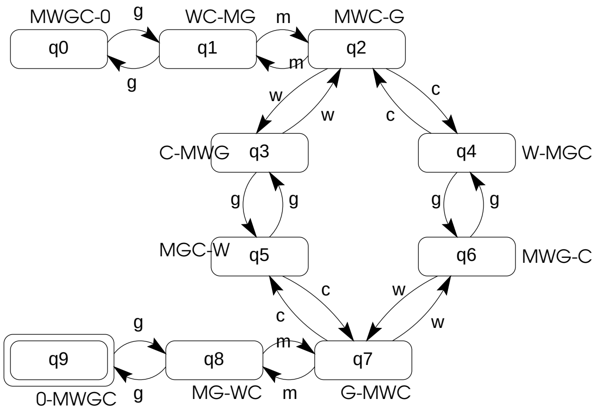
format definition:

$$M = (Q, \Sigma, \delta, q_0, F)$$

where:

Q	=	set of states
Σ	=	input alphabet
δ	=	transition function (DFA): $Q \times \Sigma \to Q$
	or	transition relation (NFA): $Q \times \Sigma^* \to Q$
q_0	=	start state, $q_0 \in Q$
F	=	set of final states, $F \subseteq Q$

finite automata: example I



finite automata: example I, continued

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
$\Sigma = \{c, g, m, w\}$
$F = \{q_9\}$
$\delta =$

state	input	next state
q_0	g	q_1
q_1	g	q_0
q_1	m	q_2
q_2	m	q_1
q_2	w	q_3
q_2	c	q_4
q_3	w	q_2
q_3	g	q_5
q_4	c	q_2
q_4	g	q_6
q_5	g	q_3
q_5	c	q_7
q_6	g	q_4
q_6	w	q_7
q_7	c	q_5
q_7	w	q_6
q_7	m	q_8
q_8	m	q_7
q_8	g	q_9
q_9	g	q_8

finite automata: example II

- define a FA which accepts strings in the following language:

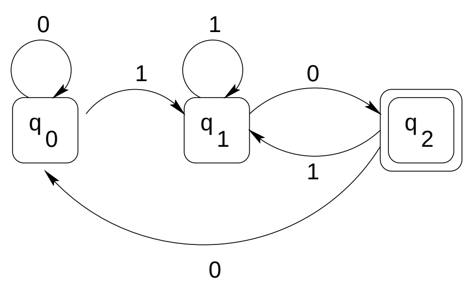
$$L = \{(ab)^*ab\}$$

- rewrite using a binary alphabet:

$$L = \{(01)^*10\}$$

⇒ determine if a given input, n, is divisible by 2, where $n > 0$

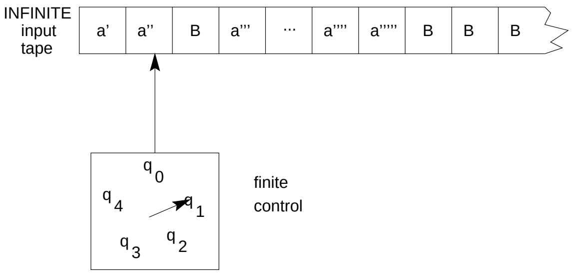
finite automata: example II



$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
$F = \{q_2\}$
$\delta =$

state	input	next state
q_0	0	q_0
q_0	1	q_1
q_1	0	q_2
q_1	1	q_1
q_2	0	q_0
q_2	1	q_1

turing machines



format definition:

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

where:

Q	$=$	set of states
Σ	$=$	input alphabet
Γ	$=$	set of tape symbols
δ	$=$	transition function: $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
q_0	$=$	start state, $q_0 \in Q$
B	$=$	blank symbol, $B \in \Gamma$, $B \notin \Sigma$
F	$=$	set of final states, $F \subseteq Q$

turing machines: example III

define a turing machine to compute *proper subtraction* $m \ominus n$.

$m \ominus n$ is defined to be:

$$= \begin{cases} m - n, & \text{where } m \geq n \\ 0, & \text{otherwise (where } m < n) \end{cases}$$

from [Hopcroft & Ullman, 1979], p.151-153.

turing machines: example III, continued

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, B\}$

$F = \{\}$

$\delta =$

$\delta(q_0, 0) = (q_1, B, R)$
$\delta(q_0, 1) = (q_5, B, R)$
$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, 0) = (q_2, 1, R)$
$\delta(q_2, 0) = (q_2, 1, R)$
$\delta(q_2, 1) = (q_3, 1, L)$
$\delta(q_2, B) = (q_4, B, L)$
$\delta(q_3, 0) = (q_3, 0, L)$
$\delta(q_3, 1) = (q_3, 1, L)$
$\delta(q_3, B) = (q_0, B, R)$
$\delta(q_4, 0) = (q_4, 0, L)$
$\delta(q_4, 1) = (q_4, B, L)$
$\delta(q_4, B) = (q_6, 0, R)$
$\delta(q_5, 0) = (q_5, B, R)$
$\delta(q_5, 1) = (q_5, B, R)$
$\delta(q_5, B) = (q_6, B, R)$

turing machines: example III, still continued

in English:
M starts with 0^m10^n on its tape and halts with $0^{m \ominus n}$ on its tape.

algorithm:
(1) replace leading 0 by B
(2) search right for a 1 followed by a 0
(3) change the 0 to 1
(4) search left for a B
(5) go to (1)
stop when:
(i) in (2), B is found after 1 instead of 0 — then replace $n + 1$ 1's with one 0 and n B's, leaving $m - n$ 0's on the tape
or
(ii) in (1), no 0 is found — then replace all remaining 1's and 0's with B

turing machines: example III, example

$m = 4, n = 2$

step	tape
0	$0000100BBB$...
1	$B000100BBB$...
2	$B000110BBB$...
3	$BB00110BBB$...
4	$BB00111BBB$...
5	$BBB0111BBB$...

the $n = 2$ 0's in $0^m10^n = 0^410^2$ have all been changed to 1's, and $n + 1 = 3$ of the $m = 4$ 0's have been changed to B.

step	tape
6	$BBB0011BBB$...
7	$BBB00B1BBB$...
8	$BBB00BBBBB$...

the 3 1's are replaced by one 0 and 2 B's, leaving $m - n = 4 - 2 = 2$ 0's on the tape.

for next lecture:

- read von Neumann (1945) *First Draft of a Report on the EDVAC* (in green book)

- Backus, J. (1978) Can Programming Be Liberated from the von Neumann Style? *Communications of the ACM*, vol. 21, pp. 613-641. (in green book)

- McCarthy, J. (1979) History of Lisp. *History of Programming Languages*, R. Wexelblat (ed.), Academic Press, 1981. (handout, or on web)