Theory of Computation

Lesson 17a - More Examples (continued)

Review

Last Time:

1. Concepts
   - Turing Machines
   - Language L(M) of machine M
   - Deciding vs. Recognizing
   - Hierarchy of languages. Have seen: regular $\subsetneq$ context-free $\neq$ decidable $\subseteq$ recognizable
   - Tabular and diagram representations
   - Tint
2. Examples
   - Shifting
   - Finding the left end of a blank tape
   - Deciding: Even-length words
   - Deciding: More a's than b's
   - Deciding $\{w \# w \mid w \in \{a, b\}^*\}$

Example: deciding ww

Exercise. Design a Turing machine with $\Sigma = \{a, b\}$ that decides $\{w\,w \mid w \in \{a, b\}^*\}$ ⟵ not context-free, only $w\,w^R$ context-free

Difficulty. No # to mark middle and separate left and right sides.

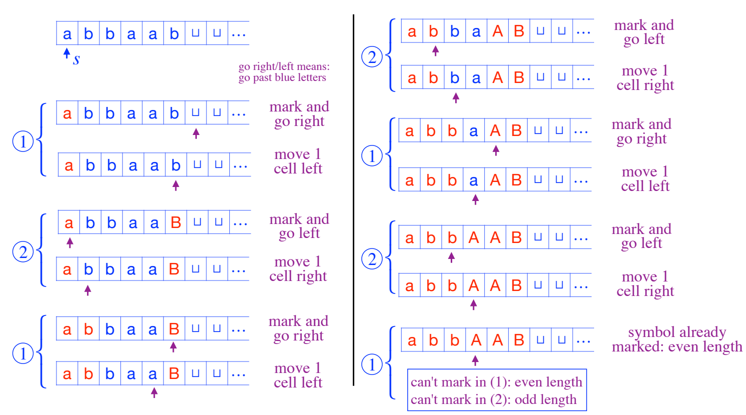
Idea. Find the middle. Use $\Gamma = \{\sqcup, a, b, a, b, A, B, \times\}$

On input string s :

1. Check if $|s|$ is even
   - if no, reject
   - if yes, write 2nd half in uppercase
     a a b a → a a B A
     a b a b → a b A B
     head under first symbol of 2nd half

2. Compare left and right halves
   - similar to $w \# w$ problem (crossing out with ×)

Example: finding the middle



Turing Machines Examples

Poll 17.1

How do we get the effect of "crossing off" with a Turing machine?

(a) We add that feature to the model.

(b) We use a tape alphabet $\Gamma = \{a, b, c, \cancel{a}, \cancel{b}, \cancel{c}, \sqcup\}$. ⟵ correct

(c) All Turing machines come with an eraser.

Theory of Computation

Lesson 17b - Stay-put and two-way machines

Stay-put Machine

Stay-put machine: like standard but allows transitions that don't move the head.

transitions: $(q, a)\ (p, b, \text{move})$

where move $\in \{\text{L}, \text{R}, \text{S}\}$

don't move ("stay put")

Claim: stay-put model is equivalent to standard model

- recognize same languages
- decide same languages

Stay-put Machine

To show

(1) Anything standard does, stay-put can do.

True as each standard is a stay-put that never uses S move.

(2) Anything stay-put does, standard can do.

Given stay-put $M = (Q, \Sigma, \Gamma, \delta, s, q_{accept}, q_{reject})$

Construct standard $\hat{M}$ that simulates M.

$$\hat{M} = (\hat{Q}, \Sigma, \Gamma, \hat{\delta}, s, q_{accept}, q_{reject})$$

- Keep standard transitions (with L, R) of M
- Replace each transition $(q, a)\ (p, b, S)$ of M with $(q, a)\ (\hat{p}, b, R)$

thus $\hat{Q} = Q \cup \{\hat{q} \mid q \in Q\}$ for each state q have copy $\hat{q}$

- For every $\hat{p}$ and every $x \in \Gamma$ add $(\hat{p}, x)\ (p, x, L)$

Two-way Machine

Two-way infinite tape

…			a	b	a	b	b		…
			s						

- Tape is infinite in both directions
- Input is written anywhere on the tape
- Computation begins in state s with head under first input symbol

Claim: two-way model is equivalent to standard one-way model

- recognize same languages
- decide same languages

Two-way Machine

(1) Everything one-way can do, two-way can also do.

Idea: ~~Every one-way is two-way that never moves left of cell 0.~~

Not quite.

0	1	2	3	4	5	6	…
a	b	a	b	b			…
s							

…	-3	-2	-1	0	1	2	3	4	5	6	…
…				a	b	a	b	b			…
				s							

Can we feed the two-way the same input and same δ and get the same results? No, due to special handling of left end.

Two-way Machine

Solution.

M

a	b	a	b	b			…
s							

$\widehat{M}$

…			a	b	a	b	b		…
			$\hat{s}$						

pre processing

…		$	a	b	a	b	b		…
			s						

For every state q of M add new transition: $(q, \$)\ (q, \$, R)$

Two-way Machine

Given $M = (Q, \Sigma, \Gamma, \delta, s, q_{accept}, q_{reject})$

have $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\Gamma}, \hat{\delta}, \hat{s}, q_{accept}, q_{reject})$

$\widehat{\Gamma} = \Gamma \cup \{\$\}$ where $\$ \notin \Gamma$ is new symbol

$\widehat{Q} = Q \cup \{\hat{s}\}$ where $\hat{s} \notin Q$ is new state

$$\hat{\delta} = \delta \cup \{(\hat{s}\ x)\ (\hat{s}\ x\ L) \mid x \in \Sigma\} \cup \{(\hat{s}\ \sqcup)\ (s\ \$\ R)\} \cup \{(q\ \$)\ (q\ \$\ R) \mid q \in Q - \{q_{accept}, q_{reject}\}\}$$

Two-way Machine

Poll 17.2 When simulating a 1-way machine M with a 2-way machine, why did we add \$ to the tape alphabet

(a) For no reason, it was unnecessary.

(b) So we could easily locate the beginning of input of M.

(c) So the simulation of M would be faster.

(d) So we could simulate the left end handling of M. ← correct

Two-way Machine

(2) Everything two-way can do, one-way can also do.

M

…	-3	-2	-1	0	1	2	3	4	5	6	…
…				a	b	a	b	b			…
				↑ s							

Goal: given two-way M, construct one-way $\widehat{M}$ that simulates M.

How to represent two-way tape on one-way tape?

One method: (See textbook.)

$\widehat{M}$

0	1	2	3	4	5	…
a	b	a	b	b		…
↑ $\hat{s}$						

preprocessing →

0	-1	1	-2	2	-3	3	-4	4	-5	5	…
a		b		a		b		b			…
↑											

Two-way Machine

(2) Everything two-way can do, one-way can also do. Take 2

M

…	-3	-2	-1	0	1	2	3	4	5	6	…
…				a	b	a	b	b			…
				↑ s							

Other method: view one way $\widehat{M}$ as having tape divided into two tracks

$\widehat{M}$

0	1	2	3	4	5	…
a	b	a	b	b		…
↑ $\hat{s}$						

preprocessing →

	0	1	2	3	4	5	6	…
\$	a	b	a	b	b			…
								…
	-1	-2	-3	-4	-5	-6	-7	…
	↑							

Two-way Machine

Other method: view one way $\widehat{M}$ as having tape divided into two tracks

$\widehat{M}$

0	1	2	3	4	5	…
a	b	a	b	b		…
↑ $\hat{s}$						

preprocessing →

	0	1	2	3	4	5	6	…
\$	a	b	a	b	b			…
								…
	-1	-2	-3	-4	-5	-6	-7	…
↑								

Tape symbols of $\widehat{M}$ include pairs

$\binom{a}{b}$ for all $a, b \in \Gamma$

$$\widehat{\Gamma} = \Gamma \cup \{\$\} \cup \Gamma \times \Gamma$$

Two-way Machine

Work of $\widehat{M}$

(a) Preprocessing: prepare the tape

a	b	a	b	b			…
↑ $\hat{s}$							

Shift input to the right, write \$ in first cell

\$	a	b	a	b	b		…
	↑						

Then make tracks $\quad (q_{tracks}\ x) \left(q_{tracks} \binom{x}{\sqcup} R \right) \quad (x \in \Sigma)$

\$	a	b	a	b	b	$\sqcup$	$\sqcup$	…
	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$			
↑								

Two-way Machine

(b) Simulate two-way M on 2-track tape

Two states q_{up} and q_{low} for each state q of M

…	-3	-2	-1	0	1	2	3	4	5	…
…		a	b	a	b	a	b	b		…
			↑ q							

	0	1	2	3	4	5	…
\$	a	b	a	b	b		…
	b	a					…
	-1	-2	-3	-4	-5	-6	…
	↑ q_{low}						

…	-3	-2	-1	0	1	2	3	4	5	…
…		a	b	a	b	a	b	b		…
					↑ q					

	0	1	2	3	4	5	…
\$	a	b	a	b	b		…
	b	a					…
	-1	-2	-3	-4	-5	-6	…
		↑ q_{up}					

Two-way Machine

For each transition of M

$q \quad a \quad p \quad b \quad$ move

$\widehat{M}$ has

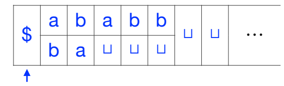
$$\left.\begin{array}{l} q_{up} \begin{pmatrix} a \\ x \end{pmatrix} \; p_{up} \begin{pmatrix} b \\ x \end{pmatrix} \; \text{move} \\[2ex] q_{low} \begin{pmatrix} x \\ a \end{pmatrix} \; p_{low} \begin{pmatrix} x \\ b \end{pmatrix} \; \text{opposite move} \end{array}\right\} \text{for all } x \in \Gamma$$

Two-way Machine

Boundary conditions

$q_{\text{up}} \quad \$ \quad q_{\text{low}} \quad \$ \quad \text{R}$

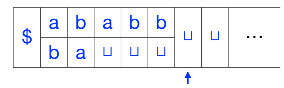
$q_{\text{low}} \quad \$ \quad q_{\text{up}} \quad \$ \quad \text{R}$



Extend tracks to the right

$q_{up} \quad \sqcup \quad q_{up} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \; \text{S}$

$q_{low} \quad \sqcup \quad q_{low} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \; \text{S}$



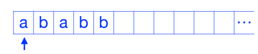
Theory of Computation

Lesson 17c - Multi-tape Machines

Turing Machine Models

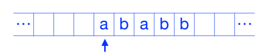
Previously

- Standard

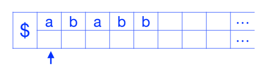


- Stay-put — move $\in$ {L, R, S}

- 2-way infinite tape



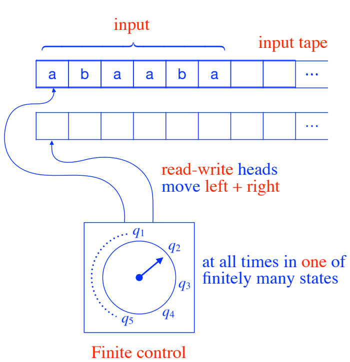
- 2-track machine



Have shown: machine models are equivalent

Multi-tape Turing Machines

Two-tape Turing Machine:



Computation begins
- with input on tape 1 (left end)
- in starting state
- each head under first cell of its tape

Transitions have the form

$$q \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \; p \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \begin{pmatrix} \text{move 1} \\ \text{move 2} \end{pmatrix}$$

(textbook: q, a_1, a_2, p, b_1, b_2, move1, move2)

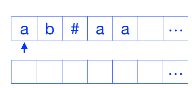
Multi-tape Turing Machines

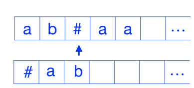
Exercise. Design a 2-tape Turing machine that decides $\{ w \# w \mid w \in \{a, b\}^* \}$

(on 1-tape TM takes time $O(n^2)$)

not context-free, only $w \# w^R$ context-free

Idea. Given input $w_1 \# w_2$

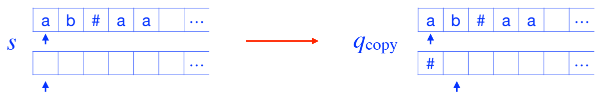




Copy w_1 to tape 2, then compare w_2 (tape 1) with w_1 (tape 2)

Deciding $w \# w$ with two tapes

(1) Write # on tape 2, switch to state q_{copy}

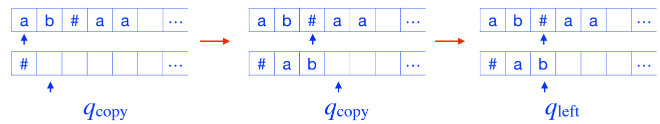


$s \begin{pmatrix} a \\ \sqcup \end{pmatrix} \; q_{copy} \begin{pmatrix} a \\ \# \end{pmatrix} \begin{pmatrix} S \\ R \end{pmatrix}$ $\qquad$ $s \begin{pmatrix} \# \\ \sqcup \end{pmatrix} \; q_{check} \begin{pmatrix} \# \\ \# \end{pmatrix} \begin{pmatrix} R \\ R \end{pmatrix}$

$s \begin{pmatrix} b \\ \sqcup \end{pmatrix} \; q_{copy} \begin{pmatrix} b \\ \# \end{pmatrix} \begin{pmatrix} S \\ R \end{pmatrix}$ $\qquad$ $s \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \; q_{reject} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \begin{pmatrix} S \\ S \end{pmatrix}$

Deciding $w \# w$ with two tapes

(2) Copy from tape 1 to tape 2 until # reached on tape 1

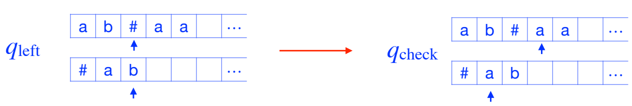


$q_{copy} \begin{pmatrix} a \\ \sqcup \end{pmatrix} \; q_{copy} \begin{pmatrix} a \\ a \end{pmatrix} \begin{pmatrix} R \\ R \end{pmatrix}$ $\qquad$ $q_{copy} \begin{pmatrix} \# \\ \sqcup \end{pmatrix} \; q_{left} \begin{pmatrix} \# \\ \sqcup \end{pmatrix} \begin{pmatrix} S \\ L \end{pmatrix}$

$q_{copy} \begin{pmatrix} b \\ \sqcup \end{pmatrix} \; q_{copy} \begin{pmatrix} b \\ b \end{pmatrix} \begin{pmatrix} R \\ R \end{pmatrix}$ $\qquad$ $q_{copy} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \; q_{reject} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \begin{pmatrix} S \\ S \end{pmatrix}$

Deciding $w \# w$ with two tapes

(3) Move head 2 back to # on the left.

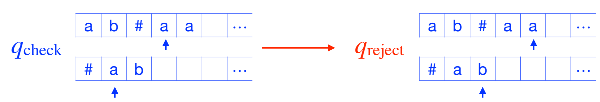


$q_{left} \begin{pmatrix} \# \\ a \end{pmatrix} \; q_{left} \begin{pmatrix} \# \\ a \end{pmatrix} \begin{pmatrix} S \\ L \end{pmatrix}$ $\qquad$ $q_{left} \begin{pmatrix} \# \\ \# \end{pmatrix} \; q_{check} \begin{pmatrix} \# \\ \# \end{pmatrix} \begin{pmatrix} R \\ R \end{pmatrix}$

$q_{left} \begin{pmatrix} \# \\ b \end{pmatrix} \; q_{left} \begin{pmatrix} \# \\ b \end{pmatrix} \begin{pmatrix} S \\ L \end{pmatrix}$

Deciding $w \# w$ with two tapes

(4) Move to the right on both tapes comparing symbols
- reject if discrepancy
- accept if both heads reach ⊔ at the same time



$q_{check} \begin{pmatrix} a \\ a \end{pmatrix} \; q_{check} \begin{pmatrix} a \\ a \end{pmatrix} \begin{pmatrix} R \\ R \end{pmatrix}$

$q_{check} \begin{pmatrix} b \\ b \end{pmatrix} \; q_{check} \begin{pmatrix} b \\ b \end{pmatrix} \begin{pmatrix} R \\ R \end{pmatrix}$

$q_{check} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \; q_{accept} \begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix} \begin{pmatrix} S \\ S \end{pmatrix}$

for $x \in \{a, b, \#, \sqcup\}$, $y \in \{a, b, \sqcup\}$ with $x \neq y$:

$q_{check} \begin{pmatrix} x \\ y \end{pmatrix} \; q_{reject} \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} S \\ S \end{pmatrix}$

Deciding $w \# w$ with two tapes

Epilogue. Deciding $\{w \# w \mid w \in \{a, b\}^* \}$

Running time on input of length n :

- 1-tape Turing Machine: $O(n^2)$
- 2-tape Turing Machine: $O(n)$