CSE 431
Lecture 7 — Intro to Theory of Computation

Last time:

Church-Turing Thesis

- TMs precisely capture our intuitive notion of algorithm

High-level view of input encoding

< >

angle brackets not "less than .. greater than"

E_{DFA}, E_{NFA}, A_{DFA}, A_{NFA}, A_{REG}, EQ_{DFA}

$\langle M \rangle$ — $L(M)=\phi$?

$\langle M, w \rangle$ — $w \in L(M)$?

$\langle R, w \rangle$ — $w \in L(R)$?

$\langle M_1, M_2 \rangle$ — $L(M_1)=L(M_2)$?

Context-Free Grammars

Given by

V - finite set of variables

Σ - alphabet (terminals)

R set of rules of form

$$A \to w \qquad A \in V,\ w \in (V \cup \Sigma)^*$$

$S \in V$ start symbol

$A \Rightarrow_G^* w$ repeatedly apply rules. from A to get w

$$L(G) = \{ w \in \Sigma^* \mid A \Rightarrow^*_G w \}$$

CSE 311 **Examples**: • Set of binary palindromes

rules:
$S \to \varepsilon$ $\quad S \to 0S0$
$S \to 0$ $\quad S \to 1S1$
$S \to 1$

short hand $\quad S \to \varepsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$

- Strings of balanced parentheses

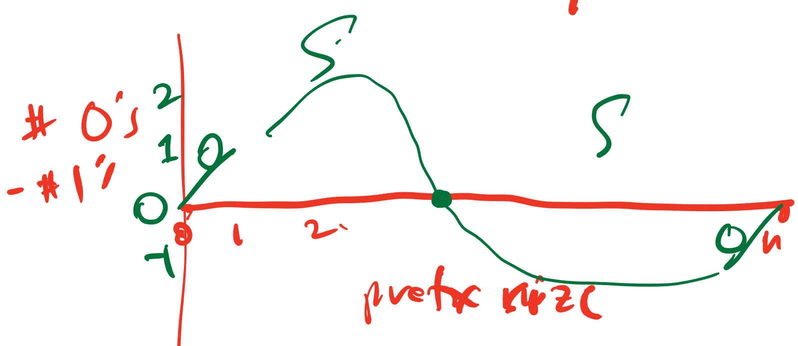
$$S \to \varepsilon \mid (S) \mid SS$$

- strings with equal #'s of 0's and 1's

$$S \to \varepsilon \mid 0S1 \mid 1S0 \mid SS$$

obvious that every string produced has equal #.

- to see that all such strings are produced see that if string begins and ends in same symbol (say 0)



graph starts at 1 and at n-1 it is -1 must cross the axis so $S \to SS$ will generate.

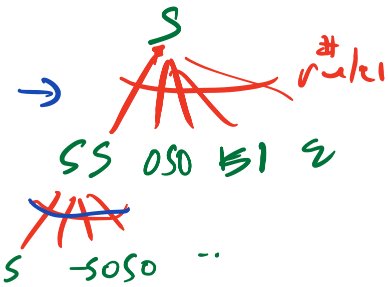
$A_{CFG} = \{\langle G, w\rangle \mid G \text{ is a CFG and } w \in L(G)\}$

Thm A_{CFG} is decidable

Proof "Obvious" thing to try.
(BFS) search through the tree of possible derived strings by applying the rules in all possible ways

if w is found then accept.

not a parse tree →



finite # of possibilities

Problem: Can't tell when to reject.
No bound on # of steps to derive w since rules may repeatedly add and remove characters

Solution: Can convert any CFG to an equivalent CFG that is in a special form for which this algorithm works

Chomsky Normal Form

all rules of form

$A \to BC$ $\qquad B, C \in V \setminus \{S\}$

$A \to a$ $\qquad a \in \Sigma$

$S \to \varepsilon$

Thm There is an algorithm that converts any CFG G to a new CFG G' in Chomsky Normal Form with $L(G') = L(G)$

Proof: Later in the quarter ∎

Note: Each rule
$A \to BC$ increases length by 1
$A \to a$ gets rid of a variable

To produce ε only rule is $S \to \varepsilon$ (if it exists)

To produce w length $|w| = n \geq 1$
need precisely
$2n-1$ steps
- $n-1$ steps to increase length from 1 to n
- n steps to replace vars by terminals

Algorithm for A_{CFG}:

On input $\langle G, w \rangle$
1. Convert G to equivalent Chomsky Normal Form G'
2. Run "obvious" BFS alg for G' and w
   - accept if w found
   - stop after depth $2n-1$ and reject if not ∎

$E_{CFG} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \emptyset \}$

Thm E_{CFG} is decidable

Examples

$S \to 0S0 \mid 1S1$
empty

$S \to 0SA \mid 1S1$
$A \to 0$
empty

$S \to 0BA \mid 1S1$
$A \to 0$, $B \to 1$
not empty.

Proof Idea: keep track of all symbols that can produce strings of terminals.

Call a variable $A \in V$ "productive" iff $A \Rightarrow^* w$ for some $w \in \Sigma^*$

Let P be the set of productive variables

Can compute P as follows:

- Put all variables A with rule $A \to w$ for $w \in \Sigma^*$, into P
- Repeatedly:
  Add $A \in V$ to P if there is a rule $A \to w$ with $w \in (\Sigma \cup P)^*$

This algorithm to compute P will stop since G is finite.

Algorithm for E_{CFG}:

On input $\langle G \rangle$

- Compute P
- if $S \in P$ reject if $S \notin P$ accept.

$\boxtimes$

Now to TMs

$$A_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM that accepts } w\}$$

Thm A_{TM} is Turing-recognizable

This is a weaker statement than decidable

if no, can either reject or run forever

Proof (Turing's idea)

Algorithm: Universal Turing Machine U

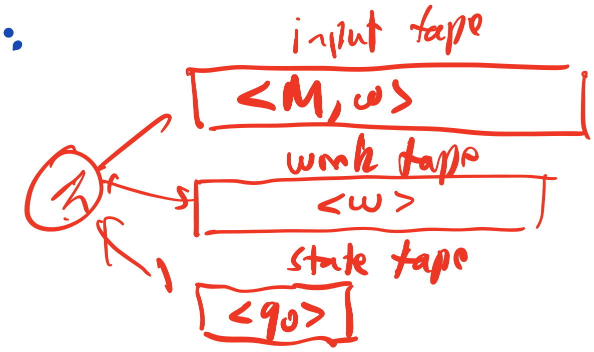
Turing machine simulator

U: On input $\langle M, w \rangle$

TM description/diagram

perform a step-by-step simulation of M on input w.

Details of U:



- start by putting <w> on work tape and <q0> on state tape

- maintain encoding of current tape (and head posit'n) on work tape and current state on state tape
  use δ table on input to figure out how to update work tape and state tape
- accept iff M does
- reject iff M does.

∎

Thm A_{TM} is <u>not</u> decidable

Proof idea diagonalization

Recall Cantor's ideas

Rationals are Countable

dovetailing can list in increasing order of a/b via max(a,b)

1/1	1/2	1/3	1/4	...
2/1	2/2	2/3	2/4	...
3/1	3/2	3/3	3/4	...
4/1	4/2	4/3	4/4	...

Dovetailing

Thm (Cantor) $\mathbb{R}^{[0,1)}$ are not countable

some messy details related to representations with repeating 0's or 9's
so instead we review

Thm (Cantor) The set $\mathcal{P}(\mathbb{N})$ of all subsets of $\mathbb{N}$ is not countable

Proof Suppose that $\mathcal{P}(\mathbb{N})$ is countable with a listing
$$S_0, S_1, S_2, S_3, \ldots$$

Table to represent this

	0	1	2	3	4	5	6
S_0	0	1	1	0	1	0	1
S_1	1	0	1	1	0	1	1
S_2	0	1	1	1	0	1	0
S_3	1	0	0	1	1	1	0
S_4	0	1	1	1	0	1	.
S_5							
$\vdots$							

row is characteristic function of S_i
$$(i,j) = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{if not} \end{cases}$$

flipped diagonal set

Define a set $D \in \mathcal{P}(\mathbb{N})$ by $i \in D$ iff $i \notin S_i$

Observe that $D \neq S_j$ for any j

since they disagree on whether they include j

$\therefore$ Contradiction since listing was supposed to be complete ∎