Announcements

Pset 2 being graded

Pset 3 out

Edit distance

$$\{w\#x^{rev} : w, x \in \{a, b, c\}^* \text{ and } ED(w, x) \leq 1\}$$

includes $w\#w^{rev}$ for all w

(insertion) include $w\#x^{rev}$ if $w = w_1 w_2 \cdots w_n$

and $x = w_1 w_2 \cdots w_{i-1} \wedge w_i \cdots w_n$

(deletion)

$- \; x = w_1 w_2 \cdots w_{i-1} w_{i+1} \cdots w_n$

$$\{w\#w^{rev} : w \in \{0,1\}^*\}$$

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

✓ abc#cba

✓ abc#caba   ✗ abc#caaba

$\Downarrow$

abc ⇒ abac

Midterm 2 Wednesday

75 min (same as before)

6 problems (same as before)

out of 180 pts (30 pts each)

full credit is 150.

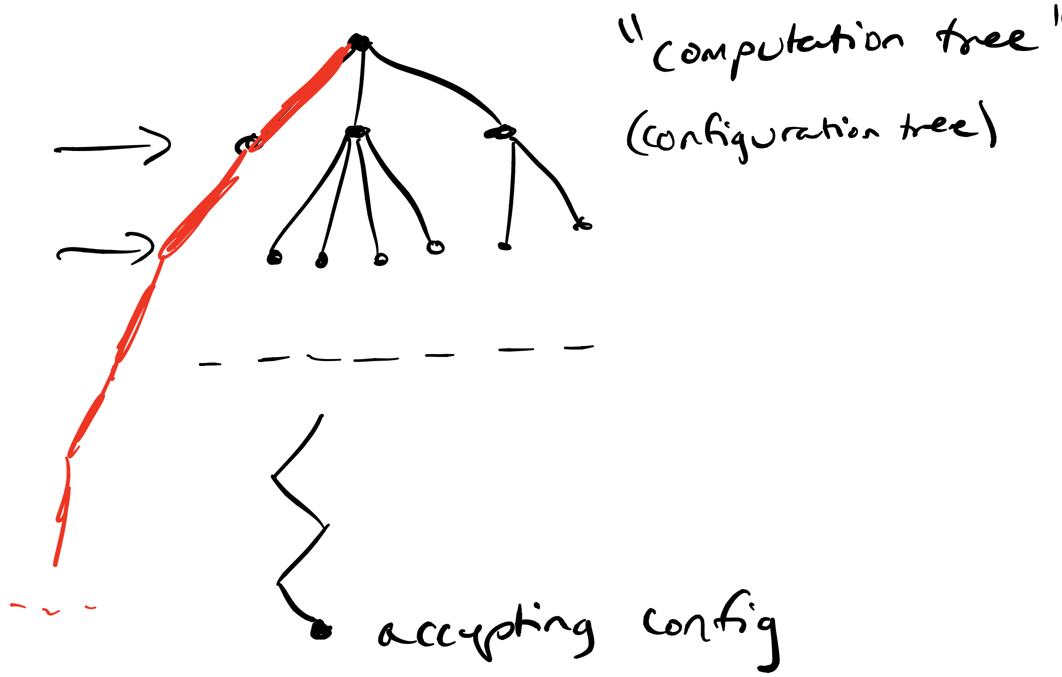
Nondeterministic TMs

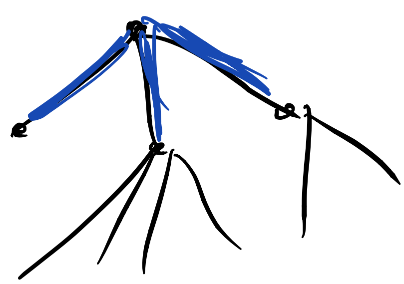
$\delta : Q \times \Gamma \to \underline{\mathcal{P}}(Q \times \Gamma \times \{L, R\})$

accept a string w if and only if there is some sequence of nondet. choices that get to q_{accept}.

Say that a NTM is a decider if <u>all</u> computation paths halt.

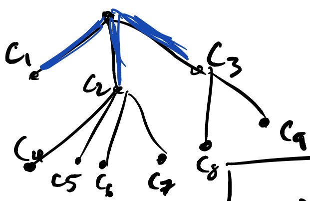
NTM $\rightarrow$ DTM





If NTM was a decider, then (1) if accepting config in tree, then DTM will find it, or (2) if otherwise, then all computations halt (by defn of NTM decider).

If NTM was a recognizer then (1) if accepting config in tree, ________________, or (2) if otherwise, then we run forever (but ok b/c the string was not accepted by the NTM).



input (no modify)

$\# C_1 \# C_2 \# C_3 \# \ldots \#\# C_4 \# C_5 \# \ldots$

C_i contain tape contents, tape head position, and which state of NTM.

tape contents $= w_1 \ldots w_n$
tape head pos $= i$
state $= q$

Config $= w_1 w_2 \ldots w_{i-1} q w_i \ldots w_n$

0, 1, 2, 3, 4, 5, ...]

0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4,
5, 5, 5, 5, 5, ...

5, 4, 3, 2, 1, 0, 10, 9, 8, 7, 6, ...

An **enumerator** is a TM enumerates a **language** L if the TM starts with empty tape and writes

$$\# w_1 \# w_2 \# w_3 \# w_4 \# \ldots$$

where each $w_i \in L$; and in a finite amount of time, fixing a w_i, it will be printed.

Say that L is **enumerable** if some enumerator enumerates it.

$$\Sigma_1^{*} = \{\varepsilon, 0, 1, 00, 01, \ldots\}$$

$$\boxed{\#\#0\#1\#00\#01\#10\# \ldots}$$

Thm: Recognizable and enumerable languages are the same.

(Enumerator $\Rightarrow$ Recognizer)

"On input w:

1. Keep asking the enumerator for another string x.
2. If $w = x$, accept.
   Otherwise, ask for another string."

(Recognizer $\Rightarrow$ Enumerator)

Let R recognize L.

Idea: generate 1 string w in Σ_1^*.

If R accepts w, then have the enumerator print it.

w is the 470th string outputted by the Σ_1^* enumerator.

"On empty input:

1. Initialize $i=1$.
2. While $i \geq 1$:
   (a) Run the first i strings in Σ_1^* for i steps each.
   (b) If any are accepted by R, then print them on the tape.
   (c) $i = i+1$."

"running in parallel"

Q: Can a TM, given 2 binary integers, write their sum to the tape?

| 1 0 1 ⊔ ⊔ - - -

| 1 0 1 1 ⊔ - - -

| 1 1 0

Compute $n+1$, $n-1$

Can we compute $n+m$? Repeated addition by 1.

Can we compute $n \times m$? Repeated addition by m.

Can we compute n^m? ——— multiplication by n.

It appears any mathematical operation a computer can do a TM can also do.

Intuitive notion of an algorithm - procedure that always halts and can be implemented on a "typical" computer.

Models of Computation

Turing Machine (Alan Turing 1936)
λ-calculus (Church 1930s)
Rewriting systems (Post 1930s)

Church-Turing thesis - our intuitive notion of algorithms and TMs are equivalent.

Encodings

$$A_{DFA} = \{ \underbrace{\langle M, w \rangle}_{\text{encoding}} : M \text{ is a DFA and } w \in L(M) \}.$$

$\langle M \rangle$

$$\# \underbrace{q_0\, q_1\, q_2\, q_3 \,---\,}_{\text{states}} q_m \# p_1\, p_2 \,--\, p_k \#\; q_1\, a\, q_3 \;---\# \; q_0 \# q_2\, q_3\, q_7 \,---\#$$

Claim: ADFA is decidable.

D = "On input $\langle M, w \rangle$ where M is a DFA and w is a string:

1. Run M on w.
2. If M ended in a final state, accept. Otherwise, reject."

Recognizable but not decidable:

$$A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM and } w \in L(M) \}.$$

$A_{NFA} = \{\langle M, w\rangle : M$ is an NFA and $w \in L(M)\}$.

is decidable.

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

1. Convert M into an equivalent DFA D.
2. Run the decider for A_{DFA} on $\langle D, w\rangle$.
3. Output what the decider says."