18.404 Recitation 4

Sept 25, 2020

Today's Topics

- Re-explaining Non-CFL Language
  - { $a^ib^jc^k$ | i > j > k }
- Review: A_{TM} is Undecidable
- Proving Decidable
  - { <R, S> | R and S are regular expressions and L(R) ⊆ L(S) }
  - { <R> | R is a regular expression and L(R) is prefix-free }
  - { <D> | D is a DFA that accepts some palindrome }
  - { <D> | D is a DFA that accepts w^R whenever it accepts w }
- Proving T-Recognizable
  - { <M> | M is a TM whose language is non-empty }
- Recap
- Bonus (time-permitting)
  - { <S> | S is a TM whose language is empty } is T-unrecog
  - 2TAPE = { {M, w} | M is a 2-tape TM that writes a non-blank symbol on 2nd tape on w }
    - Prove it is T-recognizable, but not T-decidable

Example: Proving Non-CFL Languages

Prove that $\{ a^i b^j c^k \mid i > j > k \}$ is not a CFL

- $(\forall n \geq 0)\ (uv^n xy^n z \in L)$
- $|vy| \geq 1$
- $|vxy| \leq p$

$$s = a^{p+2} b^{p+1} c^p$$

Review: A_{TM} is Undecidable (1/3)

Proof by Contradiction

A_{TM} = { <M, w> | M is a TM that accepts input w }

- Assume TM H decides A_{TM}
  - H accepts <M, w> iff M accepts w
  - H rejects <M, w> iff M rejects or loops on w
- Will prove that H may never exist due to a contradiction

Review: A_{TM} is Undecidable (2/3)

Recall assuming that H decides A_{TM} = { <M, w> | M accepts w }

Use H to construct a TM D

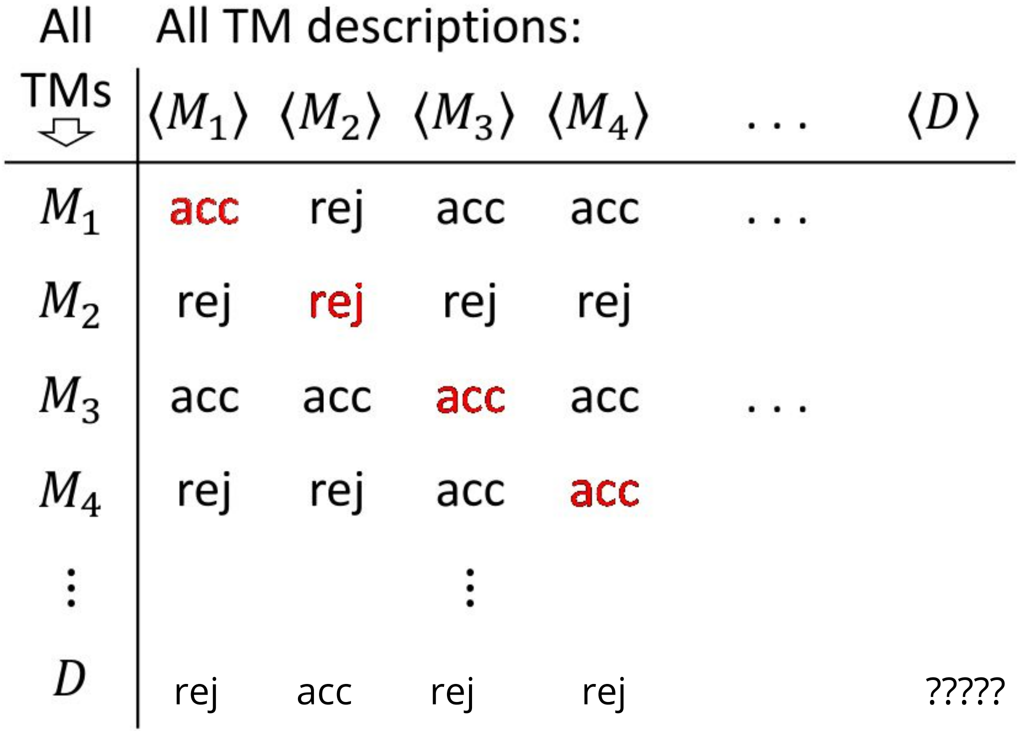
D = “On input <M>

1. Simulate H on input <M, <M>> ie: (<M, w> where w = <M>)
2. Reject if H accepts. Accept if H rejects.”

D accepts <M> iff M does not accept <M>

Contradiction: D accepts <D> iff <D> does not accept <D>

Review: A_{TM} is Undecidable (3/3)



All TM descriptions:

All TMs ⇩	$\langle M_1\rangle$	$\langle M_2\rangle$	$\langle M_3\rangle$	$\langle M_4\rangle$	. . .	$\langle D\rangle$
M_1	acc	rej	acc	acc	. . .	
M_2	rej	rej	rej	rej		
M_3	acc	acc	acc	acc	. . .	
M_4	rej	rej	acc	acc		
$\vdots$			$\vdots$			
D	rej	acc	rej	rej		?????

Proving Decidable

$\{ \langle R, S \rangle \mid R \text{ and } S \text{ are regular expressions and } L(R) \subseteq L(S) \}$

```
D = "on input <R, S>
1. Convert R and S into DFA R' and S' respectively
2. Construct DFA T = R' intersect S'
3. Run EQ_DFA on <R', T> and return accordingly
"
```

Proving Decidable

{ <R> | R is a regular expression and L(R) is prefix-free }

NOT prefix free = {"Star Wars", "b", "ac", "Star Wars is cool!"}

```
D = "on input <R>
1. Construct DFA R' from reg expr R
2. Prune all out-going edges from accept states of R' to create DFA P
      (this filters all suffixed strings out of L(R')
3. Run EQ_DFA on R and P. Accept if EQ_DFA accepts. Reject otherwise.
"
```

Proving Decidable

{ <D> | D is a DFA that accepts some palindrome } palindrome = {w+rev(w)}

D has a palindrome -> intersection of L(D) and palindrome is non empty set

D has no palindrome -> intersection of L(D) and palindrome is empty set

Use construction from HW 2, problem 0.2: $\text{regular language} \cap \text{CFL} = \text{CFL}$

```
F = "on input <D>
1. Use construction from HW 2 to create PDA P that computes:
   CFL = reg lang intersect palindrome
2. Run E_PDA on P. Accept if E_PDA rejects. Reject otherwise.
```

Proving Decidable

{ <D> | D is a DFA that accepts w^R whenever it accepts w }

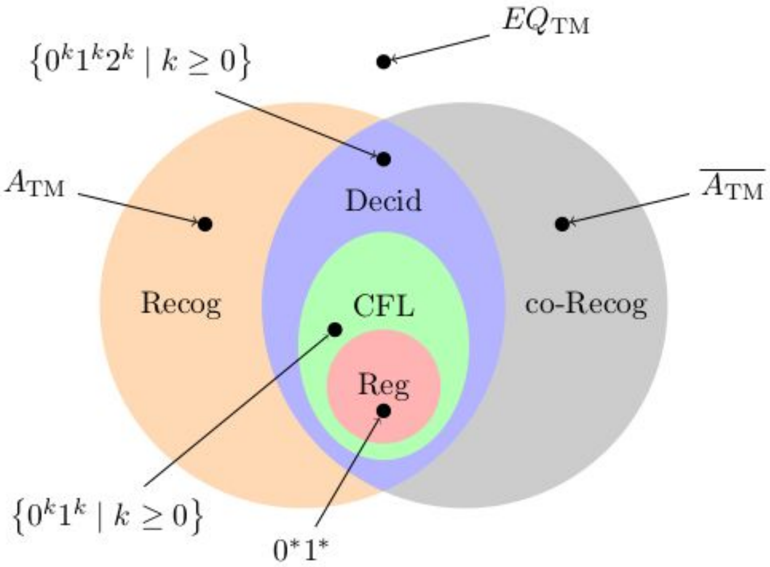
Proving T-Recognizable

{ <M> | M is a TM whose language is non-empty }

```
R = "on input <M>
1. Simulate M on all inputs of $\Sigma^*$ one by one
2. If M accepts any of the inputs, then accept
"
```

If M really has empty language then will iterate forever over Sigma* and never terminate. But this is OK for T-Recog languages.

Recap


EQ_{TM}
$\{0^k1^k2^k \mid k \geq 0\}$
A_{TM}
Decid
$\overline{A_{\text{TM}}}$
Recog
CFL
co-Recog
Reg
$\{0^k1^k \mid k \geq 0\}$
0^*1^*