Announcements

- Pick up graded H/W #7
- Pick up (partial) solutions to sample final
- Final exam
  - Monday, June 5, 2:30-4:20pm, here
- Review session
  - Sunday, June 4, 3:00-4:00pm, EE1 045
- No puzzle today
  - I think you've had enough of them!

Wrap-up

Atri Rudra

June 2

Q4 in H/W #8

- This question will not be graded
  - Is a solved problem in Sipser
  - I apologize for the blunder
  - H/W #8 will out of a total of 48 (earlier was 60)
- Solutions to H/W #8
  - Will be online by today evening

Last lecture

- \( A_{TM} = \{ \langle M, w \rangle | M \text{ accepts } w \} \)
- \( A_{TM} \) is undecidable
- Proof by contradiction (diagonalization)
- Assume there is a decider \( H \) for \( A_{TM} \)

As H is a decider...

Questions?
Up next…

- $A_{CFG} = \{ (G, w) | G$ is a CFG that generates $w \}$
- All context-free languages are also decidable
- If we have time:
  - Some more undecidable languages
  - One language that is not Turing-recognizable

End of quarter report…

What we studied in this course…

- Models!
  - Not really…

Models of Computation

- The machine series
- The grammar series
- The language series
  - Studied different variants
  - Established their relative "powers"

The machine series

- Deciders vs "normal" TMs

The grammar series (Chomsky hierarchy)

- Regular grammars
  - Context-free Grammars
  - Unrestricted grammars
The language series

Decidable \{0^n1^n\}

Regular \Sigma^* \epsilon \Sigma

CFLs \{0^n|w|\}

Turing Recognizable

A_{TM}

Your impression of the course…

Hopefully…

What next?

- Computational Complexity
  - This course builds the platform
  - CSE 431, offered every spring

In the “big” picture…

What the heck do I mean?

- Computability vs Complexity
  - TM can decide if \( w \in L \) in \( 2^{2|w|} \) time for any \( w \)
  - \( L \) is computable
- Sure it is a decider but is it any “good”?
Computational Complexity

Worry about resource consumption
- How much time does an algorithm take?
  - Exp. time vs. poly time vs. linear time
  - How much space does an algorithm take?
  - How much randomness does an algorithm use?

The P vs NP question
- In a nutshell: if it is easy to verify a proof is it also easy to come up with a short proof?
- Solve it and earn a million $!

So now the game is...

- Given a problem
  - Is there an efficient algorithm?
  - Or is there “no” efficient algorithm?
- Questions?

Thanks!