## Recap of Undecidability Proof

- The Question: Are there languages that are not decidable by any Turing machine (TM)?
  - i.e. Are there problems that cannot be solved by any algorithm?
- Consider the language:
  $A_{TM}$ = {<M,w> | M is a TM and M accepts w}
  (Recall that <A,B,…> is just a string encoding the objects A, B, …)
- What can we say about $A_{TM}$?



## $A_{TM}$ is Turing-recognizable

- $A_{TM}$ is Turing-recognizable: Recognizer TM R for $A_{TM}$:
  On input string <M,w>:
  Simulate M on w.
  ACCEPT <M,w> if M halts & accepts w;
  REJECT <M,w> if M halts & rejects
  (Loop (& thus reject <M,w>) if M ends up looping).
  R accepts <M,w> iff M accepts w, i.e. L(R) = $A_{TM}$

Yeah, but is it decidable?!!

# Is $A_{TM}$ decidable?

- No, $A_{TM}$ = {<M,w> | M is a TM and M accepts w} is undecidable! 1-slide Proof (by Contradiction):
  1. Assume $A_{TM}$ is decidable ⇒ there's a decider H, L(H) = $A_{TM}$
  2. H on <M,w> = ACC if M accepts w  
     REJ if M rejects w (halts in $q_{REJ}$ or loops on w)
  3. Construct new TM D: On input <M>:  
     Simulate H on <M,<M>> (here, w = <M>)  
     If H accepts, then REJ input <M>  
     If H rejects, then ACC input <M>
  4. What happens when D gets <D> as input?  
     D rejects <D> if H accepts <D,<D>> if D accepts <D>  
     D accepts <D> if H rejects <D,<D>> if D rejects <D>

  Either way: Contradiction! D cannot exist ⇒ H cannot exist  
  Therefore, $A_{TM}$ is not a decidable language.



# Undecidability Proof uses Diagonalization

Input strings

List of TMs

| | <$M_1$> | <$M_2$> | <$M_3$> | … |
|---|---|---|---|---|
| $M_1$ | ACC | REJ | *loop* | … |
| $M_2$ | REJ | *loop* | ACC | … |
| $M_3$ | ACC | ACC | REJ | … |
| : | : | : | : | |

If H exists →

D outputs opposite of diagonal

| | <$M_1$> | <$M_2$> | <$M_3$> | … | <D> |
|---|---|---|---|---|---|
| $M_1$ | <u>ACC</u> | REJ | **REJ** | … | ACC |
| $M_2$ | REJ | **<u>REJ</u>** | ACC | … | ACC |
| $M_3$ | ACC | ACC | <u>REJ</u> | … | REJ |
| : | : | : | : | : | : |
| D | <u>REJ</u> | <u>ACC</u> | <u>ACC</u> | … | ?? |

D on <$M_i$> accepts if and only if $M_i$ on <$M_i$> rejects.  
So, D on <D> will accept if and only if D on <D> rejects!  
A contradiction ⇒ H cannot exist!  
Therefore, $A_{TM}$ is not a decidable language.

## One Last Concept: Reducibility

- How do we show a new problem B is undecidable?
- Idea: Show that $A_{TM}$ is <u>reducible to</u> the new problem B
  - What does this mean and how do we show this?
- Show that if B was decidable, then you can use the decider for B as a *subroutine* to decide $A_{TM}$
  - Contradiction, therefore B must also be undecidable



## The Halting Problem is Undecidable (Turing, 1936)

- Halting Problem: Does TM M halt on input w?
  - Equivalent language: $A_H$ = { <M,w> | TM M halts on input w}
  - Need to show $A_H$ is undecidable
  - We know $A_{TM}$ = {<M,w> | TM M accepts w} is undecidable
- Show $A_{TM}$ is reducible to $A_H$ (Theorem 5.1 in text)
  - Suppose $A_H$ is decidable $\Rightarrow$ there's a decider $M_H$ for $A_H$
  - Then, we can construct a decider $D_{TM}$ for $A_{TM}$:

    On input <M,w>, run $M_H$ on <M,w>.
    - If $M_H$ rejects, then REJ (this takes care of M looping on w)
    - If $M_H$ accepts, then simulate M on w until M halts
    - If M accepts, then ACC input <M,w>; else REJ

    $L(D_{TM}) = A_{TM} \Rightarrow A_{TM}$ is decidable! Contradiction $\Rightarrow A_H$ is undecidable
- E.g. 2: Show $E_{TM}$ = {<M> | M is a TM and L(M) = $\varnothing$} is undecidable (see Theorem 5.2 in the text)

## Are There Languages That Are Not Even Recognizable?

- $A_{TM}$ and $A_H$ are undecidable but Turing-recognizable
  - Are there languages that are not even Turing-recognizable?
- What happens if both A and $\overline{A}$ are Turing-recognizable?
  - There exist TMs M1 and M2 that recognize A and $\overline{A}$
  - Can construct a decider for A! On input w:

  1. Simulate M1 and M2 on w one step at a time, alternating between them.
  2. If M1 accepts, then ACC w and halt; if M2 accepts, REJ w and halt.
- A and $\overline{A}$ are both Turing-recognizable iff A is decidable
- Corollary: $\overline{A_{TM}}$ and $\overline{A_H}$ are not Turing-recognizable
  - If they were, then $A_{TM}$ and $A_H$ would be decidable



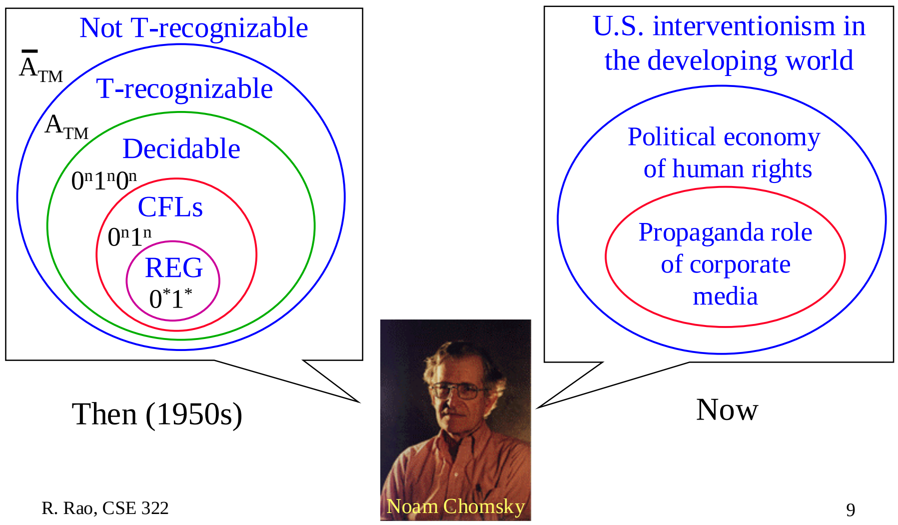
## The Chomsky Hierarchy of Languages

Increasing generality →

| **Language** | Regular | Context-Free | Decidable | Turing-Recognizable |
|---|---|---|---|---|
| **Computational Models** | DFA, NFA, RegExp | PDA, CFG | Deciders – TMs that halt for all inputs | TMs that may loop for strings not in language |
| **Examples** | $(0\cup1)^*11$ | $\{0^n1^n \mid n \geq 0\}$, Palindromes | $\{0^n1^n0^n \mid n \geq 0\}$, $A_{DFA}$, $A_{CFG}$ | $A_{TM}$, $A_H$ |

(Chomsky also studied context-sensitive languages (CSLs, e.g. $a^nb^nc^n$) , a subset of decidable languages recognized by linear-bounded automata (LBA))

# The Chomsky Hierarchy – Then & Now…



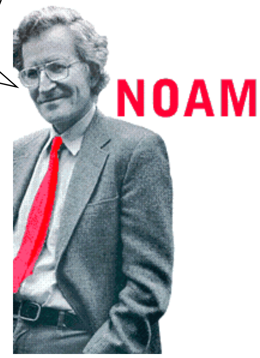



# Final Exam

- Details regarding the Final Exam
  - When: Monday, Dec. 16, 2002 from 8:30-10:20 a.m.
  - Where: This classroom EE1 037.
  - What will it cover?
    - Chapters 0-4 and Chapter 5: pages 171-176.
    - Emphasis will be on material covered after midterm (Chapter 2 and beyond)
    - You may bring 1 page of notes (8 ½” x 11” sheet!)
    - Approximately 6 questions
  - How do I ace it?
    - Practice, practice, practice!
    - See class website for sample final exam and solutions


I believe the Final exam is decidable!


Stay cool 'n' keep pumpin'!


I believe the world's problems are politically decidable.
NOAM


I believe my next movie will be unrecognizable.