Counting argument

Last time:
We proved that there must be a language that is not Turing-recognizable because

- Countable: Turing-recognizable Languages
- Uncountable: All sets of strings

All sets of strings
Non-recognizable languages exist

• But what do they look like?

• What is a specific example of a language that is not Turing-recognizable? or not Turing-decidable?

• Idea: consider a set that, were it to be Turing-decidable, would have to "talk" about itself.

Diagonalization
Recall $A_{DFA} = \{<B,w> \mid B \text{ is a DFA and } w \text{ is in } L(B)\}$

$A_{TM} = \{<M,w> \mid M \text{ is a TM and } w \text{ is in } L(M)\}$

What is $A_{TM}$?

A. A Turing machine whose input is codes of TMs and strings.
B. A set of pairs of TMs and strings.
C. A set of strings that encode TMs and strings.
D. Not well defined.
E. I don't know.
Define the TM $\text{N} = \text{"On input } <M,w>:\text{"}
1. Simulate $M$ on $w$.
2. If $M$ accepts, accept. If $M$ rejects, reject.
Define the TM $N$ = "On input $<M,w>$:
1. Simulate $M$ on $w$.
2. If $M$ accepts, accept. If $M$ rejects, reject."

What is $L(N)$?
A. $A_{TM}$
B. Some larger set that includes $A_{TM}$
C. $\{<M,w> | M \text{ is a TM and } w \text{ is a string}\}$
D. I don't know.
Define the TM \( N = \) "On input \(<M, w>\):
1. Simulate \( M \) on \( w \).
2. If \( M \) accepts, accept. If \( M \) rejects, reject."

Which statement is true?
A. \( N \) decides \( A_{TM} \)
B. \( N \) recognizes \( A_{TM} \)
C. \( N \) always halts
D. I don't know.
\[ A_{TM} = \{ <M, w> \mid M \text{ is a TM and } w \text{ is in } L(M) \} \]

Define the TM \( N \) = "On input \( <M, w> \):

1. Simulate \( M \) on \( w \).
2. If \( M \) accepts, accept. If \( M \) rejects, reject."

Conclude: \( A_{TM} \) is Turing-recognizable.

Is it decidable?
Diagonalization proof: $A_{TM}$ not decidable  

Assume, towards a contradiction, that it is.

I.e. let $M_{ATM}$ be a Turing machine such that for every TM $M$ and every string $w$,

- Computation of $M_{ATM}$ on $<M, w>$ halts and accepts if $w$ is in $L(M)$.
- Computation of $M_{ATM}$ on $<M, w>$ halts and rejects if $w$ is not in $L(M)$.

What does it mean for $w$ not to be in $L(M)$?

A. $M$ halts on $w$
B. $M$ loops on $w$
C. $M$ accepts $w$
D. $M$ rejects $w$
E. More than one of the above
Suppose $N$ is a TM with $L(N) = \{w \mid w \text{ starts with 0}\}$ and $N$ loops infinitely on all strings not in $L(N)$. What is result of computation of $M_{\text{ATM}}$ on $<N, 11>$?

- A. $M_{\text{ATM}}$ halts and accepts.
- B. $M_{\text{ATM}}$ halts and rejects.
- C. $M_{\text{ATM}}$ loops.
- D. I don't know.

I.e. let $M_{\text{ATM}}$ be a Turing machine such that for every TM $M$ and every string $w$,

- Computation of $M_{\text{ATM}}$ on $<M,w>$ halts and accepts if $w$ is in $L(M)$.
- Computation of $M_{\text{ATM}}$ on $<M,w>$ halts and rejects if $w$ is not in $L(M)$.
Diagonalization proof: $A_{TM}$ not decidable

Assume, towards a contradiction, that $M_{ATM}$ decides $A_{TM}$

Define the TM $D =$ "On input $<M>$:

1. Run $M_{ATM}$ on $<M, <M>>$.  
2. If $M_{ATM}$ accepts, reject; if $M_{ATM}$ rejects, accept."
Assume, towards a contradiction, that $M_{\text{ATM}}$ decides $A_{\text{TM}}$.

Define the TM $D$ = "On input $<M>$:
1. Run $M_{\text{ATM}}$ on $<M, <M>>$.
2. If $M_{\text{ATM}}$ accepts, reject; if $M_{\text{ATM}}$ rejects, accept."

Is $D$ a decider?
A. Yes: it's a TM that always halts.
B. No: it's a well-defined TM but it may loop in step 1.
C. No: it's not even a well-defined TM.
D. I don't know.
Assume, towards a contradiction, that $M_{\text{TM}}$ decides $A_{\text{TM}}$.

Define the TM $D =$ "On input $<M>$:
1. Run $M_{\text{ATM}}$ on $<M, <M>>$.
2. If $M_{\text{ATM}}$ accepts, reject; if $M_{\text{ATM}}$ rejects, accept."

If $M_0$ is a TM with $L(M_0) = \emptyset$, what is result of computation of $D$ with input $<M_0>$?
A. Halt and accept.
B. Halt and reject.
C. Loop.
D. I don't know.
Assume, towards a contradiction, that \( M_{\text{ATM}} \) decides \( A_{\text{TM}} \).

Define the TM \( D \) = "On input \(<M>\):"

1. Run \( M_{\text{ATM}} \) on \(<M, <M>>\).
2. If \( M_{\text{ATM}} \) accepts, reject; if \( M_{\text{ATM}} \) rejects, accept."

If \( M_1 \) is a TM with \( L(M_1) = \Sigma^* \), what is result of computation of \( D \) with input \(<M_1>\)?

A. Halt and accept.
B. Halt and reject.
C. Loop.
D. I don't know.
Diagonalization proof: $A_{TM}$ not decidable \textit{Sipser 4.11}

Assume, towards a contradiction, that $M_{ATM}$ decides $A_{TM}$

Define the TM $D = "On input <M>:  
1. Run $M_{ATM}$ on <M, <M>>.  
2. If $M_{ATM}$ accepts, reject; if $M_{ATM}$ rejects, accept."$

Consider running $D$ on input $<D>$. Because $D$ is a decider:  
- either computation halts and accepts …  
- or computation halts and rejects …
Assume, towards a contradiction, that $A^{\text{TM}}_{\text{TM}}$ decides $A^{\text{TM}}_{\text{TM}}$.

Define the TM $D = \text{"On input } <M>:\n1. \text{ Run } M^{\text{ATM}} \text{ on } <M, <M>>.\n2. \text{ If } M^{\text{ATM}} \text{ accepts, reject; if } M^{\text{ATM}} \text{ rejects, accept.\"}}$

Consider running $D$ on input $<D>$. Because $D$ is a decider:
- either computation halts and accepts …
- or computation halts and rejects …

Diagonalization? Self-reference

"Is $<D>$ an element of $L(D)$?"
Regular

Context-Free

Turing-Decidable

\{a^n b^n | n \geq 0\}

\{a^m b^n | m,n \geq 0\}

\{a^n b^n a^n | n \geq 0\}

??

??

Turing-Recognizable
Next Time

• Another strategy for showing undecidability.

• No diagonalization (or counting).