Lecture 18—June 1, 2006
Reducibility

Instructor: Neil Rhodes
Show that the following language is undecidable:

- $T = \{<M>| M \text{ is a TM that accepts } w^R \text{ whenever it accepts } w\}$

Step one: decide on undecidable language to reduce to $T$
- $\leq_M T$

Step two: Decide on two languages for a new TM, $M'$ to accept:
- $(<M>,w) \in \rightarrow M' \in T$. So, if $M$ accepts $w$, $M'$ accepts $L_1$, where $L_1$ contains $w^R$
  
  whenever it contains $w$. Let $L_1 = \{\}$
- $(<M>,w) \notin \rightarrow M' \notin T$. So, if $M$ doesn’t accept $W$, $M'$ accepts $L_2$, where $L_2$ contains some string whose reverse is not in $L_2$. Let $L_2 = \{\}$

Step three: write the description for the TM $M'$ that takes input $w'$
- step a) If $w' = \rightarrow$, accept
- step b) If $w' = \rightarrow$, run $M$ on $w$. If it accepts, accept
- step c) Otherwise, reject

Step four: describe the mapping function $f$:
- $f$ will, given $M$ and $w$, create the TM $M'$ as described above

Step five: summary
- We’ve reduced $T$ to $T$ because:
  - If $(<M>,w) \in , f(<M>,w) = M'$, and $M'$ accepts $L_1 = \{$, so $<M'> \in T$
  - If $(<M>,w) \notin , f(<M>,w) = M'$, and $M'$ accepts $L_2 = \{$, so $<M'> \notin T$
Consider the problem of determining whether a two-tape Turing machine ever writes a nonblank symbol on its second tape when it is run on input \( w \). Formulate this problem as a language and show that it is undecidable.

- \( L = \{<M, w>| M \text{ is a TM that}\} \)
- Reduce what to \( L \)?
- What will \( M' \) do?
Decidability example

Consider the problem of determining whether a two-tape Turing machine ever writes a nonblank symbol on its second tape when it is run on any input. Formulate this problem as a language and show that it is undecidable.

- L = \{ <M> | M is a TM that \}
- Reduce what to L?
- What will M’ do?
Decidability example

Consider the problem of determining whether a single-tape Turing machine ever writes a blank symbol over a nonblank symbol during the course of its computation on any input string. Formulate this problem as a language and show that it is undecidable.

- \( L = \{<M>| \text{M is a TM that}\} \}
- Reduce what to \(L\)?
- What will \(M'\) do?
Decidability example

Consider the problem of determining whether a single-tape Turing machine on any input ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and show that it is undecidable.

- $L = \{<M> | M \text{ is a TM that} \}$
- Reduce what to $L$?
- What will $M'$ do?
Decidability example

Consider the problem of determining whether a single-tape Turing machine on any input ever writes a blank on the left-most tape cell. Formulate this problem as a language and show that it is undecidable.

- \( L = \{<M>| M \text{ is a TM that} \} \)
- Reduce what to \( L \)?
- What will \( M' \) do?
Decidability example

A *useless state* is a state that is never entered on any input string.

Consider the problem of determining whether a single-tape Turing machine has any useless states. Formulate this problem as a language and show that it is undecidable.

- \( L = \{ <M> | \text{M is a TM that} \} \)
- Reduce what to \( L \)?
- What will \( M' \) do?
Decidability example

Consider the problem of determining whether a single-tape Turing machine on any input halts leaving the tape with the contents “011111”. Formulate this problem as a language and show that it is undecidable.

- $L = \{<M>| M \text{ is a TM that } \}$
- Reduce what to $L$?
- What will $M'$ do?
Decidability example

Consider the problem of determining whether a single-tape Turing machine on input \( w \) enters all the states of the machine. Formulate this problem as a language and show that it is undecidable.

- \( L = \{<M>| M \text{ is a TM that} \} \)

Reduce what to \( L \)?

What will \( M' \) do?
Decidability example

Consider the problem of determining whether a single-tape Turing machine on input \( w \) ever moves the tape head right then immediately left then immediately right then immediately left. Formulate this problem as a language and show that it is undecidable.

- \( L = \{<M, w>| M \text{ is a TM that} \} \)
- Reduce what to \( L \)?
- What will \( M' \) do?
Reducing \( \cap_{\text{CFG}} \) to \( \text{ALL}_{\text{CFG}} \)

\( \text{ALL}_{\text{CFG}} = \{ G | \text{G is a CFG and } L(G) = \Sigma^* \} \)

- Create \( G' = (G_1 \cap G_2)^c \)
- Check whether \( G' \in \text{ALL}_{\text{CFG}} \). If yes, \( G_1 \) and \( G_2 \) don’t intersect. If no, they do intersect.

- Is this valid? Is \( G_1 \cap G_2 \) a CFG? Is the complement of a CFL a CFL?
  - No

- However, look at \( \text{PCP} \leq \cap_{\text{CFG}} \leq \text{ALL}_{\text{CFG}} \)
  - The grammars used in the reduction of PCP were deterministic!

  - So, actually, the reduction is:

\[ G' = G_1^c \cup G_2^c = (G_1 \cap G_2)^c \]

- \( G_1^c \) is DCFG
- Union is CFG