More Undecidable Problems

Rice’s Theorem
Post’s Correspondence Problem
Some Real Problems
Properties of Languages

- Any set of languages is a property of languages.
- **Example:** The infiniteness property is the set of infinite languages.
Properties of Languages – (2)

◆ As always, languages must be defined by some descriptive device.
◆ The most general device we know is the TM.
◆ Thus, we shall think of a property as a problem about Turing machines.
◆ Let \( L_P \) be the set of binary TM codes for TM’s M such that \( L(M) \) has property P.
Trivial Properties

- There are two (trivial) properties $P$ for which $L_P$ is decidable.
  1. The *always-false property*, which contains no RE languages.
  2. The *always-true property*, which contains every RE language.

- **Rice’s Theorem**: For every other property $P$, $L_P$ is undecidable.
Plan for Proof of Rice’s Theorem

1. **Lemma needed**: recursive languages are closed under complementation.
2. We need the technique known as **reduction**, where an algorithm converts instances of one problem to instances of another.
3. Then, we can prove the theorem.
Closure of Recursive Languages Under Complementation

◆ If $L$ is a language with alphabet $\Sigma^*$, then the *complement* of $L$ is $\Sigma^* - L$.

 Denote the complement of $L$ by $L^c$.

◆ **Lemma**: If $L$ is recursive, so is $L^c$.

◆ **Proof**: Let $L = L(M)$ for a TM $M$.

◆ Construct $M'$ for $L^c$.

◆ $M'$ has one final state, the new state $f$. 
Proof – Concluded

◆ $M'$ simulates $M$.
◆ But if $M$ enters an accepting state, $M'$ halts without accepting.
◆ If $M$ halts without accepting, $M'$ instead has a move taking it to state $f$.
◆ In state $f$, $M'$ halts.
A reduction from language L to language L’ is an algorithm (TM that always halts) that takes a string w and converts it to a string x, with the property that:

\[ x \text{ is in L’ if and only if } w \text{ is in L.} \]
TM’s as *Transducers*

- We have regarded TM’s as acceptors of strings.
- But we could just as well visualize TM’s as having an *output tape*, where a string is written prior to the TM halting.
- Such a TM translates its input to its output.
Reductions – (2)

- If we reduce $L$ to $L'$, and $L'$ is decidable, then the algorithm for $L'$ + the algorithm of the reduction shows that $L$ is also decidable.

- **Used in the contrapositive**: If we know $L$ is not decidable, then $L'$ cannot be decidable.
Reductions – Aside

◆ This form of reduction is not the most general.

◆ **Example**: We “reduced” $L_d$ to $L_u$, but in doing so we had to complement answers.

◆ More in NP-completeness discussion on Karp vs. Cook reductions.
Proof of Rice’s Theorem

◆ We shall show that for every nontrivial property $P$ of the RE languages, $L_P$ is undecidable.
◆ We show how to reduce $L_u$ to $L_P$.
◆ Since we know $L_u$ is undecidable, it follows that $L_P$ is also undecidable.
The Reduction

◆ Our reduction algorithm must take M and w and produce a TM M’.
◆ L(M’) has property P if and only if M accepts w.
◆ M’ has two tapes, used for:
  1. Simulates another TM M_L on the input to M’.
  2. Simulates M on w.
◆ Note: neither M, M_L, nor w is input to M’.
The Reduction – (2)

◆ Assume that $\emptyset$ does not have property P.
  ◦ If it does, consider the complement of P, which would also be decidable by the lemma.
◆ Let L be any language with property P, and let $M_L$ be a TM that accepts L.
◆ $M'$ is constructed to work as follows (next slide).
Design of $M'$

1. On the second tape, write $w$ and then simulate $M$ on $w$.
2. If $M$ accepts $w$, then simulate $M_L$ on the input $x$ to $M'$, which appears initially on the first tape.
3. $M'$ accepts its input $x$ if and only if $M_L$ accepts $x$. 
Action of $M'$ if $M$ Accepts $w$

- Simulate $M$ on input $w$
- Simulate $M_L$ on input $x$

On accept:

Accept iff $x$ is in $M_L$
Design of $M'$ – (2)

◆ Suppose $M$ accepts $w$.
◆ Then $M'$ simulates $M_L$ and therefore accepts $x$ if and only if $x$ is in $L$.
◆ That is, $L(M') = L$, $L(M')$ has property $P$, and $M'$ is in $L_P$. 
Design of $M'$ – (3)

- Suppose $M$ does not accept $w$.
- Then $M'$ never starts the simulation of $M_L$, and never accepts its input $x$.
- Thus, $L(M') = \emptyset$, and $L(M')$ does not have property $P$.
- That is, $M'$ is not in $L_P$. 
Action of $M'$ if $M$ Does not Accept $w$

Simulate $M$ on input $w$

Never accepts, so nothing else happens and $x$ is not accepted
Design of $M'$ – Conclusion

- Thus, the algorithm that converts $M$ and $w$ to $M'$ is a reduction of $L_u$ to $L_P$.
- Thus, $L_P$ is undecidable.
A real reduction algorithm $M, w$ → $M'$ → Hypothetical algorithm for property $P$

Accept iff $M$ accepts $w$
Otherwise halt without accepting

This would be an algorithm for $L_u$, which doesn’t exist
Applications of Rice’s Theorem

◆ We now have any number of undecidable questions about TM’s:
  ▶ Is \( L(M) \) a regular language?
  ▶ Is \( L(M) \) a CFL?
  ▶ Does \( L(M) \) include any palindromes?
  ▶ Is \( L(M) \) empty?
  ▶ Does \( L(M) \) contain more than 1000 strings?
  ▶ Etc., etc.