CSE 4111 Computability
Jeff Edmonds
Assignment 4: Reductions for Uncomputable Problems
Due: One week after shown in slides

First Person:
Family Name:
Given Name:
Student #:
Email:

Second Person:
Family Name:
Given Name:
Student #:
Email:

Guidelines:

- You are strongly encouraged to work in groups of two. Do not get solutions from other pairs. Though you are to teach & learn from your partner, you are responsible to do and learn the work yourself. Write it up together. Proofread it.
- Please make your answers clear and succinct. helpful hints.
- Relevant Readings:
  - Slides
  - Any book you used for 2001 on reductions.
- This page should be the cover of your assignment.

Problem Name	Max Mark	
1 Reduction I^R	10	
2 Reduction too far left	10	
3 Algoirthm left	10	
4 Reduction CFG	10	
Total	40	

CSE 4111 Computability
Jeff Edmonds
Assignment 4: Reductions for Uncomputable Problems
Due: One week after shown in slides

1. Let $P = \{\text{“}M\text{”} \mid M$ is a TM that accepts I^R (reverse) whenever it accepts I $\}$, i.e. computational problem P says “yes” on inputs $''M''$ in this set and says “no” on inputs not in this set.

   (a) Use a reduction to prove that this is undecidable.

   (b) Can you directly use Rice's Theorem to prove this?

2. Let $P = \{\langle \text{“}M\text{”}, I\rangle \mid$ TM M on input I never tries to move its head left when it is already on the left hand most cell of the tape. $\}$.

   (a) Use a reduction to prove that this is undecidable.

   (b) Can you directly use Rice's Theorem to prove this?

3. Let $P = \{\langle \text{“}M\text{”}, I\rangle \mid$ TM M on input I never tries to move its head left. $\}$. (Assume that TMs never leave their head stationary.)

   (a) Which other model of computation does a TM that never moves its head left remind you of?

   (b) After the TM has moved its right past the input and moved right for a while on the blank tape what must eventually happen to the state that it is in (i.e. whats written on its black board)?

   (c) Give an algorithm that decides the problem.

4. For each of the following, either give an algorithm for it or prove that it is uncomputable.

   (a) $P_{CFG\ n} = \{\langle G, n\rangle \mid$ all strings of length at most n can be generated by the CFG $G\}$.

   (b) $P_{CFG\ half} = \{G \mid$ for each n, at least half of all strings of length n can be generated by the CFG $G\}$. Assume the terminal alphabet is $\Sigma = \{0, 1\}$. What would the problem be of instead saying “at least half of all strings”?