CMSC 303 Introduction to Theory of Computation, VCU
Assignment 6
Key

Total marks: 52 marks + 3 marks bonus for typing your solutions in LaTeX.

Unless otherwise noted, the alphabet for all questions below is assumed to be $\Sigma = \{a, b\}$. This assignment will get you primarily to practice reductions in the context of decidability.

1. [10 marks] We begin with some mathematics regarding uncountability. Let $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ denote the set of natural numbers.

   (a) [5 marks] Prove that the set of binary numbers has the same size as $\mathbb{N}$ by giving a bijection between the binary numbers and $\mathbb{N}$.

   Solution: Let f be a bijection, where each binomial number maps to its equivalent natural number (for example, $f(0) = 0$, $f(1) = 1$, $f(10) = 2$, etc). We conclude that f is a bijection, as desired.

   (b) [5 marks] Let B denote the set of all infinite sequences over the English alphabet. Show that B is uncountable using a proof by diagonalization.

   Solution: Our proof is analogous to the proof from class that the set of real numbers, $\mathbb{R}$, is uncountable. Specifically, assume for sake of contradiction that B is countable. Then there exists a way to enumerate the elements of B one by one; in other words, assume it is possible to list all elements of B in some infinitely long list L. We now use diagonalization to construct an infinite sequence x over $\{Englishalphabet\}$ such that $x \notin L$. This will give us our desired contradiction. To construct x, simply set the ith bit of x for $i \geq 1$ to the opposite of the ith bit of the ith entry of L. In other words, if the first entry of L reads $abcdefghi\ldots$, we set the first bit of x to a. Hence, for all $i \geq 1$, the ith bit of x will disallow x from equalling the ith entry of L. We conclude that $x \notin L$, as desired.

2. [15 marks] We next move to a warmup question regarding reductions.

   (a) [3 marks] Intuitively, what does the notation $A \leq B$ mean for problems A and B?

   Solution: This means that an algorithm for problem B can be used to solve problem A.

   (b) [3 marks] What is a mapping reduction $A \leq_m B$ from language A to language B? Give both a formal definition, and a brief intuitive explanation in your own words.

   Solution: A mapping reduction from A to B is a computable function $f : \Sigma^* \mapsto \Sigma^*$ satisfying the property that for all $x \in \Sigma^*$, $x \in A$ iff $f(x) \in B$. Intuitively, this means that there is an algorithmic process through which an instance x of problem A can be translated into an instance $f(x)$ of problem B, in such a way that x is a YES-instance of A iff $f(x)$ is a YES-instance of B.

   (c) [3 marks] What is a computable function? Give both a formal definition, and a brief intuitive explanation in your own words.

Solution: A computable function $f : \Sigma^* \mapsto \Sigma^*$ is a function which can be computed in a finite amount of time by some Turing machine. Specifically, the latter takes input x, runs for a finite amount of steps, halts, and outputs $f(x)$ on its tape. Intuitively, a computable function is simply one whose output can be computed by a TM.

(d) [6 marks] Suppose $A \leq_m B$ for languages A and B. Please answer each of the following with a brief explanation.

i. If B is decidable, is A decidable?

Solution: Yes. Since we've reduced problem A to problem B, if we can decide the latter, we can also decide the former.

ii. If A is undecidable, is B undecidable?

Solution: Yes. This is the contrapositive of part 2di above.

iii. If B is undecidable, is A undecidable?

Solution: Not necessarily. This just means that the approach of deciding A by rephrasing it in terms of B is a bad idea, since B is undecidable. It could be, however, that A can be reduced to some *other* language C which *is* decidable.

3. [5 marks] Show that if L = $\{0^n 1^n \mid n \geq 0\}$ is Turing-recogizable and L $\leq \overline{L}$, then L is decidable.

Solution: Suppose that L $\leq \overline{L}$. Then $\overline{L} \leq$ L using the same mapping reduction. Because L is Turing-recognizable, Theorem 5.28 implies that $\overline{L}$ is Turing-recognizable, and then Theorem 4.22 implies that L is decidable.

4. [12 marks] Prove and discuss the following reductions.

(a) [5 marks] Walk through the proof to show that the problem of proving the language of a Turning Machine is a context-free language is undecidable. (Do not use Rice's theorem as a black box and note that this is not the same problem as Theorem 5.13 in the textbook.)

Solution: Using the Theorem 5.3 and text below this from the textbook, we can show that testing if a language is context-free,CF_{TM} is undecidable. Here is the proof: We will use a reduction from A_{TM}. We will let Turing Machine C decide CF_{TM}. We can show that C can be used to decide $A_{TM}, A_{TM} \leq CF_{TM}$. If S decides A_{TM}, S takes input $\langle M, w\rangle$ and we modify M so that:

i. If M accepts w, then M_w accepts $\{0^n 1^n \mid n \geq 0\}$ (A randomly selected context-free language).
ii. If M does not accept w, then M_w accepts $\{0^n 1^n 2^n \mid n \geq 0\}$ (A randomly selected language that is not context-free).

We let C be a Turing Machine that decides CF_{TM} and construct a Turing Machine S to decide A_{TM}. Then S works in the following manner.
S = "On input $\langle M, w\rangle$, where M is a TM and w is a string:

i. Construct the following TM M_w.
   M_w = "On input x:
   A. If x has form $0^n 1^n 2^n$, accept.
   B. If x does not have this form, run M on input w and accept if M accepts w."
ii. Run C on $\langle M_w\rangle$
iii. If C accepts, accept; if C rejects, reject."

Since we cannot decide A_{TM}, this is a contradiction and thus CF_{TM} is undecidable.

(b) [5 marks] Use mapping reductions to prove that L = $\{\langle M\rangle \mid$ M is a Turing Machine and M accepts $\epsilon\}$ is undecidable.

Solution: We can make a Turing Machine S, which upon receiving an input $\langle M, w\rangle$ for A_{TM} it outputs $\langle M', w'\rangle$ for L such that: $\langle M, w\rangle \in A_{TM}$ if and only if $\langle M', w'\rangle \in L$
Define S = "On input $\langle M, w\rangle$:

i. Construct TM M'
   M' = "On input x:
   A. Run M on x
   B. If M accepts, accept
   C. If M rejects, enter an infinite loop."
ii. Output $\langle M', w'\rangle$ where $w' = w$."

(c) [2 marks] How are these two proofs different?

Solution: Mapping reductions allow you to formulate Turing Machine M' for B that acts as a black box for Turing Machine M for the language, A, mapped to B, so that if M' is a decider or recognizer for B, then M is a decider or recognizer for A. If if M is not a decider or recognizer for A, then M' is not a decider or recognizer for B.

5. [10 marks] Use Rice's Theorem if possible to show the following problems are undecidable. If it is not possible to use Rice's Theorem, explain why not.

(a) [5 marks] $M_{1TM} = \{\langle M\rangle$ | M is a TM and L(M) is finite$\}$.

Solution: The property P is "the language is a finite language.", so it is the property of the language that we are looking at. It is non-trivial since, there is at least one machine, M_{in}, such that $\langle M\rangle \in M_{1TM}$, a machine that accepts 0, and at least one machine, M_{notin}, such that $\langle M\rangle \notin M_{1TM}$, a machine that accepts Σ^*.

(b) [5 marks] $M_{2TM} = \{\langle M\rangle$ | M is a TM and L(M) is a subset of $\Sigma^*\}$.

Solution: This is trivial since all languages can be a subset of $\Sigma^*\}$, so this set contains all languages.

6. [Bonus +3 marks] Find a match to the following Post Correspondence Problem set:
$\{ \frac{ab}{abab}, \frac{b}{a}, \frac{aba}{b}, \frac{aa}{a} \}$

Solution: $\{ \frac{ab}{abab}, \frac{ab}{abab}, \frac{aba}{b}, \frac{b}{a}, \frac{b}{a}, \frac{aa}{a}, \frac{aa}{a} \}$

If they have: $\{ \frac{aa}{a}, \frac{aa}{a}, \frac{b}{a}, \frac{ab}{abab} \}$ give them 1 point. They should have used all fractions or dominos.