Jason Downing
Email: jason_downing@student.uml.edu
Foundations of Computer Science
Homework # - Chapter 4 + Chapter 5
12/1/2016

****
Since I started this assignment early, and I had some time left before the assignment was due, I decided to do the following **Extra Credit** problems:

1. ??

These problems are at the end of my PDF in the "**Extra Credit**" section.
****

4.2 Consider the problem of determining whether a DFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable.

We can describe this problem as the language:
$AB_{DFA,REX} = \{\langle D, R\rangle | D$ is a DFA, R is a regular expression and $L(D) = L(R)\}$. The following Turing Machine (TM) M decides $AB_{DFA,REX}$:

$M =$ "On input $\langle D, R\rangle$:

1. Convert the regular expression R to an equivalent DFA A using the procedure that is given in Theorem 1.28.
2. Use the TM C for deciding $AB_{DFA,REX}$ in Theorem 4.5, on input $\langle D, A\rangle$.
3. If R accepts, *accept*.
4. If R rejects, *reject*."

4.3 Let $ALL_{DFA} = \{\langle A\rangle | A \text{ is a } DFA \text{ and } L(A) = \sum^*\}$. Show that ALL_{DFA} is decidable.

Let $ALL_{DFA} = \{\langle A\rangle | A$ is a DFA that recognizes $\sum^*$. The TM M decides ALL_{DFA}:
M = "On input $\langle A\rangle$ where A is a DFA:

1. Construct DFA B that recognizes $\overline{L(A)}$ which is described in the 1.10 exercise.
2. Run TM T from Theorem 4.4 on the input $\langle B\rangle$, where T will decide E_DFA.
3. If T accepts, *accept*.
4. If T rejects, *reject*."

4.4 Let $A\epsilon_{CFG} = \{\langle G\rangle | G \text{ is a } CFG \text{ that generates } \epsilon\}$. Show that $A\epsilon_{CFG}$ is decidable.

Let $A\epsilon_{CFG} = \{\langle G\rangle | G \text{ is a } CFG \text{ that generates } \epsilon\}$. The following TM M will decide $A\epsilon_{CFG}$:
M = "On input $\langle G\rangle$ where G is a CFG:

1. Run TM S from Theorem 4.6 on input $\langle G, \epsilon\rangle$, where S is a decider for $A\epsilon_{CFG}$.
2. If S accepts, *accept*.
3. If S rejects, *reject*."

4.6 Let X be the set $\{1, 2, 3, 4, 5\}$ and let Y be the set $\{6, 7, 8, 9, 10\}$. We describe the functions $f : X \to Y$ and $g : X \to Y$ in the following tables. Answer each part and give a reason for each negative answer.

n	$f(n)$
1	6
2	7
3	6
4	7
5	6

n	$g(n)$
1	10
2	9
3	8
4	7
5	6

a. Is f one-to-one?
Answer: No, f is not one-to-one because we can find a case that shows it is not. Which is when $f(1) = f(3)$, as well as $f(2) = f(4)$ and $f(3) = f(5)$.

b. Is f onto?
Answer: No, f is not onto because there is no case that exists where $x \in X$ is $f(x) = 10$.

c. Is f a correspondence?
Answer: No, f is not a correspondence because f is not one-to-one and onto.
d. Is g one-to-one?
Answer: Yes, g is one-to-one.

e. Is g onto?
Answer: Yes, g is onto.

f. Is g a correspondence?
Answer: Yes, g is a correspondence because g is one-to-one and onto.

4.7 Let $\mathcal{B}$ be the set of all infinite sequences over $\{0, 1\}$. Show that $\mathcal{B}$ is uncountable using a proof by diagonalization.

We can assume that B is countable, and that a correspondence $f : \mathcal{N} \to \mathcal{B}$ exists. We can then construct x in $\mathcal{B}$ that does not pair with anything in $\mathcal{N}$. We can then let $x = x_1, x_2, ...$ Let $x_i = 0$ if we find that $f(i)_i) = 1$, and we can also say that $x_i = 1$ if we find that $f(i)_i = 0$ where $f(i)_i$ is the ith bit of $f(i)$. This will let us make sure that x is not in $f(i)$ for any i because it is different then the $f(i$ in the ith symbol. As a result, a contradiction will occur, and this proves that $\mathcal{B}$ is uncountable.

4.8 Let $T = \{(i, j, k)|i, j, k \ \in \ \mathcal{N}\}$. Show that T is countable.

We can demonstrate that T is one-to-one with the following function: $f : T \to \mathcal{N}$. We can let $f(i, j, k) = 2^i 3^j 5^k$. The function f is one-to-one because when $a \neq b$, $f(a) \neq f(b)$. As a result, T is countable.

5.1 Show that EQ_{CFG} is undecidable.

We can show that EQ_{CFG} is undecidable by showing a contradiction for EQ_{CFG} that is decidable. We can construct a decider, D, for $ALL_{CFG} = \{\langle G\rangle | G \text{ is a } CFG \text{ and } L(G) = \sum^*\}$ that is as follows:

$D =$ "On input $\langle G\rangle$:

1. Construct a CFG C such that $L(C) = \sum^*$.
2. Run the decider for EQ_{CFG} on the input $\langle G, C\rangle$
3. If it accepts, we *accept*.
4. If it rejects, we *reject*."

5.2 Show that EQ_{CFG} is co-Turing-recognizable.

We can show that EQ_{CFG} is co-Turing-recognizable by showing a Turing Machine (TM) T which will recognize the complement of EQ_{CFG}:

$T =$ "On input $\langle G, H\rangle$:

1. Generate the strings $x \in \sum^*$ lexicographically.
2. Test each string x and see whether $x \in L(G)$ and $x \in L(H)$ are true, using the algorithm for A_{CFG}.
3. If we find that one of the tests accepts, and the other rejects, then we mark it as *accept*.
4. Otherwise, we continue."

5.3 Find a match in the following instance of the Post Correspondence Problem:

$$\{[\frac{ab}{abab}], [\frac{b}{a}], [\frac{aba}{b}], [\frac{aa}{a}]\}$$

One possible match is:

$$[\frac{ab}{abab}] \quad [\frac{ab}{abab}] \quad [\frac{aba}{b}] \quad [\frac{b}{a}] \quad [\frac{b}{a}] \quad [\frac{aa}{a}] \quad [\frac{aa}{a}]$$

5.4 If $A \leq_m B$, and B is a regular language, does that imply that A is a regular language? Why, or why not?

Answer: No, this does not implty that A is a regular language. One example that shows it is not a regular language is: $\{a^n b^n c^n | \ n \ \geq 0\} \leq_m \{a^n b^n | \ n \ \geq 0\}$

The reduction is what first tests whether the input is a member of $\{a^n b^n c^n | \ n \ \geq 0\}$. If it is a member, then it outputs the string ab, and if it is not a member then it just outputs the string a.

****

EXTRA CREDIT SECTION BEGINS HERE

****