York University
CSE 2001 Fall 2017 – Assignment 3 of 4
Instructor: Jeff Edmonds

Sorry. You MUST work in a pair.

Family Name: ______________________ Given Name: ______________________

Student #: ______________________ Email: ______________________

Family Name: ______________________ Given Name: ______________________

Student #: ______________________ Email: ______________________

Section to which to return the test (circle one): A: 9:00, E: 4:00

1) Easy CFG	15	
2) Tuple CFG	25	
3) Parsing CFG	30	
4) Undecidable	$30 = 5 + 5 + 20$ (+10)	
0) Art	2	
Total	112 marks	

Keep your answers short and clear.

0) (2 marks) Art therapy question: When half done the exam, draw a picture of how you are feeling.

1. You are to give me a context free grammar to generate the language of all tuples of tuples and characters $\{a, b, c\}$. For example, $\langle a, a, \langle b, c, \langle b \rangle \rangle, a, \langle \rangle \rangle$. Note that the terminal symbols are the characters 'a', 'b', 'c', '$\langle$', '$\rangle$', and ','. Note the tuples can be of arbitrary lengths. Hint, use the following nonterminal symbols:

   - T to represent a tuple. (The start symbol).
   - L to represent a list of tuples and characters $\{a, b, c\}$. For example, "$a, a, \langle b, c, \langle b \rangle \rangle, a, \langle \rangle$".
   - I to represent one item, namely either one tuple or one character from $\{a, b, c\}$.

   Be sure that the brackets are formed in matching pairs and that the commas are formed to appear singly between items.

   Demonstrate your grammar by giving a parsing of the string $\langle a, \langle \rangle, b \rangle$

2. Parsing: If possible, write pseudo code for parsing the following grammar.

$$\begin{aligned} S &\Rightarrow A\ a\ S \\ &\Rightarrow A\ b \\ &\Rightarrow A \end{aligned}$$

   A parsing can be presented as a little picture of the parse tree or as a tuple as done in the assignment.

3. Consider alphabets $\Sigma_1 = \{a, b, c\}$ and $\Sigma_2 = \{p, q, r, s, t\}$.

   Σ_1^* consists of all finite strings over Σ_1. Similarly Σ_2^*. We want to determine whether or not Σ_1^* and Σ_2^* have the same *size*. One way of proving that they do is to set up a bijection between them. This can be done, but it is tricky.

   Clearly $|\Sigma_1^*| \leq |\Sigma_2^*|$. Hence, what remains is to determine whether or not $|\Sigma_1^*| \geq |\Sigma_2^*|$. This is true if and only if there is a mapping (encoding) $f : \Sigma_2^* \rightarrow \Sigma_1^*$ such that each string in Σ_2^* is mapped to a unique string in Σ_1^*. (It might not be a bijection because some strings in Σ_1^* might not get mapped to.) In other words, can you use strings over Σ_1 to *name* all strings over Σ_2.

   If you think that such mapping exists, explain why and give pseudo code for computing f. If you think that no such mappings f exists, carefully explain why. Recall that Jeff says that a set is countably-infinite in size if and only if each element in the set has a unique finite description.

4. The Halting Problem is Undecidable

   (a) Use first order logic to state that problem P is computable. Might the TM mentioned in this sentence fail to halt on some input?

   (b) Suppose I give you as an oracle a Universal Turing Machine. With this extra help, does this change with whether you can solve the Halting problem?

   (c) Suppose you think it undignified to feed a TM M a description "M" of itself. Instead, of making M's nemesis be I_M = "M", lets instead define $I_M = F(M)$ where $F(M)$ is the description of what the TM M fears the most. For example, $F(M_{Sherlock\ Homes})$ = "*Moriarty*" and $F(M_{Super\ Man})$ = "*Kryptonite*".

      i. Suppose $F(M)$ is distinct for each TM M, i.e. $\forall M, M',\ M \neq M' \Rightarrow F(M) \neq F(M')$. Using this new nemesis input, give the proof that there is a problem P_{hard} that is uncomputable. This is done by giving the first order logic statement and then playing the game.
      (Six quick sentences, i.e. I removed all the chat from the posted proof.)
      If you have memorized the proof in the slides and you put it here unchanged you will get 60%.

      ii. (Bonus Question so no marks for a blank):
      Suppose $F(M)$ is not distinct for each TM M, i.e. $\exists M, M',\ M \neq M'$ and $F(M) = F(M')$. Suppose we want P_{hard} to be a language, i.e. its output is in $\{Yes, No\}$. What does wrong in your previous proof?