CpSc 421 Homework 8 **Solutions**

Extra Credit

Attempt up to three of the problems below.

1. (**15 points**) Let $G = (V, \Sigma, R, Expr)$ be a CFG with variables $V = \{Expr, Factor, Term\}$, and terminals $\Sigma = \{\textsf{CONSTANT}, \textsf{IDENTIFIER}, \textsf{PLUS}, \textsf{TIMES}, \textsf{LPAREN}, \textsf{RPAREN}\}$ and rules:

$$\begin{array}{rcl|l|l} Expr & \rightarrow & Term & Expr\,\textsf{PLUS}\,Term & \\ Term & \rightarrow & Factor & Term\,\textsf{Times}\,Factor & \\ Factor & \rightarrow & \textsf{IDENTIFIER} & \textsf{CONSTANT} & \textsf{LPAREN}\,Expr\,\textsf{RPAREN} \end{array}$$

Here are regular expressions for the terminals:

$$\begin{array}{rcl} \textsf{CONSTANT} & \equiv & (\texttt{0} \cup \texttt{1} \cup \ldots \cup \texttt{9})^+ \\ \textsf{IDENTIFIER} & \equiv & (\texttt{a} \cup \texttt{b} \cup \ldots \cup \texttt{z} \cup \texttt{A} \cup \texttt{B} \cup \ldots \cup \texttt{Z})^* \\ \textsf{PLUS} & \equiv & \texttt{+} \\ \textsf{TIMES} & \equiv & \texttt{*} \\ \textsf{LPAREN} & \equiv & \texttt{(} \\ \textsf{RPAREN} & \equiv & \texttt{)} \end{array}$$

Whitespace between terminals is ignored.

For each string below, either show that it is generated by G by drawing a parsetree or showing a derivation, or explain why it is not generated by G.

(a) `2*a+b`
Solution
$Expr \rightarrow Expr + Term \rightarrow Term + Factor \rightarrow Term * Factor + b \rightarrow Factor * a + b \rightarrow 2 * a + b$.

(b) `a+2*b`
Solution
$Expr \rightarrow Expr + Term \rightarrow Term + Term * Factor \rightarrow Factor + Factor * b \rightarrow a + 2 * b$.

(c) `a-1`
Solution
This is not generated by G because $- \notin \Sigma$.

(d) `(aardvark+2)*antelope`
Solution
$Expr \rightarrow Term \rightarrow Term * Factor \rightarrow Factor * \texttt{antelope} \rightarrow (Expr) * \texttt{antelope} \rightarrow (Expr + Term) * \texttt{antelope} \rightarrow (Term + Factor) * \texttt{antelope} \rightarrow (Factor + 2) * \texttt{antelope} \rightarrow (\texttt{aardvark} + 2) * \texttt{antelope}$.

(e) `2x + 3*(y+z)`
Solution
Not generated by G because G does not permit two consecutive $Factor$ to be concatenated, which is required to generate $2x$.

2. (**20 points**), Sipser, problem 2.27

Let $G = (V, \Sigma, R, S)$ be the following grammar:

$$\begin{array}{rcl} \mathit{STMT} & \to & \mathit{ASSIGN} \mid \mathit{IfThen} \mid \mathit{IfThenElse} \\ \mathit{IfThen} & \to & \texttt{if condition then } \mathit{STMT} \\ \mathit{IfThenElse} & \to & \texttt{if condition then } \mathit{STMT} \texttt{ else } \mathit{Stmt} \\ \mathit{ASSIGN} & \to & \texttt{a:=1} \end{array}$$

$$\begin{array}{rcl} \Sigma & = & \{\texttt{if}, \texttt{condition then}, \texttt{else}, \texttt{a:=1}\} \\ V & = & \{\mathit{STMT}, \mathit{IfThen}, \mathit{IfThenElse}, \mathit{ASSIGN}\} \end{array}$$

G is a natural-looking grammar for a fragment of a programming language, but G is ambiguous.

(a) (**1**0 points) Show that G is ambiguous.
Solution

STMT $\to$
`if condition then STMT else STMT` $\to$
`if condition then if condition then STMT else STMT` $\to$
`if condition then if condition then` $a := 1$ `else` $a := 1$
and
STMT $\to$
`if condition then STMT` $\to$
`if condition then if condition then STMT else STMT` $\to$
`if condition then if condition then` $a := 1$ `else` $a := 1$

(b) (**1**0 points) Give a new, unambiguous grammar for the same language.
Solution

3. (**32 points**) For each language below, either show that it is contex-free or prove that it is not. Please give a short explanation of how any CFG or PDA that you use for your solution works.

$$C_1 = \{\mathtt{a}^i \mathtt{b}^j \mathtt{c}^k \mid i \le j \le k\}$$

Solution
C_1 is not context-free. Let p be a proposed pumping lemma constant, and let $w = a^p b^p c^p \in C_1$. Let $w = uvxyz$ with $|vxy| \le p$ and $|vy| > 0$. Consider two cases: if $vy \in L(a^*b^*)$, then $w' = uv^2xy^2z \notin C_1$ because either $\#a(w') > \#c(w')$ or $\#b(w') > \#c(w')$. If $vy \in L(b^*c^*)$, then $w' = uv^0xy^0z \notin C_1$ because either $\#a(w') > \#c(w')$ or $\#a(w') > \#b(w')$. Therefore, C_1 is not context-free.

$$C_2 = \overline{C_1}$$

Solution
C_2 is context-free. Write C_2 as $A \cup B \cup C$, where $A = L(\Sigma^* ba \Sigma^* \cup \Sigma^* cb \Sigma^* \cup \Sigma^* ca \Sigma^*)$ is regular. B is a subset of $L(a^*b^*c^*)$ where the number of a's exceeds the number of b's, and C is a subset of $L(a^*b^*c^*)$ where the number of b's exceeds the number of c's. The grammar for B is:
$S \to aATC$
$C \to cC|\epsilon$
$A \to aA|\epsilon$
$T \to aTb|\epsilon$
The grammar for C is:
$S \to AbBT$
$A \to aA|\epsilon$
$B \to bB|\epsilon$

$T \rightarrow bTc|\epsilon$
Since C_2 is the union of 3 context-free languages, C_2 is context-free.

$$C_3 = \{\mathsf{a}^i\mathsf{b}^j \mid i \in \{j, 2j\}\}$$

Solution
C_3 is given by the following grammar, and is therefore context-free.
$S \rightarrow E|U$
$E \rightarrow aEb|\epsilon$
$U \rightarrow aaUb|\epsilon$

$$C_4 = \overline{C_3}$$

Solution
Write C_4 as the union of 4 languages, $A \cup L_{a>2b} \cup L_{b>a} \cup L_{a\in(b,2b)}$. A is a regular language of strings that contain a ba substring: $A = L(\Sigma^* ba \Sigma^*)$. The other languages are subsets of $L(a^*b^*)$. $L_{a>2b}$ has more a's than twice the number of b's, and is given by the grammar
$S \rightarrow AT$
$T \rightarrow aaTb|\epsilon$
$A \rightarrow aA|a$

$L_{b>a}$ has more b's than a's and is given by the grammar
$S \rightarrow TB$
$T \rightarrow aTb|\epsilon$
$B \rightarrow bB|b$

$L_{a\in(b,2b)}$ has more a's than b's but less a's than twice the number of b's and is given by the grammar
$S \rightarrow aaaTbb$
$T \rightarrow aATb|\epsilon$
$A \rightarrow a|\epsilon$

Since C_4 is the union of 4 context-free languages, C_4 is context-free.

4. (**40 points**) Let Σ be any finite alphabet with $|\Sigma| \geq 2$. Let

$$D \quad = \quad \{s \in \Sigma^* \mid \exists w \in \Sigma^*.\ s = ww\}$$

(a) (**10 points**) Prove that D is not context-free.
Solution
Without loss of generality, assume that $0, 1 \in \Sigma$. Let p be a proposed pumping lemma constant, and let $s = 0^p 1^p 0^p 1^p \in D$. Let $s = uvxyz$ where $|vxy| \leq p$ and $|vy| > 0$. If $vy \in L(0^*1^*)$ (without loss of generality, assume that vy appear in the first $2p$ symbols of s), then uv^0xy^0z is either of odd length or has a first-half substring ending with 0, which implies it is not in D because the second half substring ends with 1. If $vy \in L(1^*0^*)$, then $uv^2xy^2z \notin D$ for one of four reasons: the length of the string is odd, the first half substring ends with 0, the second half substring begins with 1, or the number of 0's in the two halfs is different. The first case occurs when $|vy|$ is odd, the second case occurs when $|vy|$ has more 0's than 1's, the third case occurs when $|vy|$ has more 1's than 0's, and the fourth case occurs when $|vy|$ has an equal number of 0's and 1's.

(b) (**30 points**) Prove that $\overline{D}$ is context-free. $\overline{D}$ is generated by the grammar below, unioned with a regular language for strings of odd length (i.e. $L(\Sigma(\Sigma\Sigma)^*)$. The key observation is that a PDA/grammar may guess the symbol position i within the first half substring that differs from the symbol in the same position in the second half substring, and that the number of symbols between these two is the length of w, which

is equal to the number of symbols before i in the first half, plus the number of symbols following i in the second half. Thanks to Cedric, Cecilia and others for the grammar.
$S \to S_i S_j$ for any $i, j \in \{1, 2, ..., |\Sigma|\}$ with $i \neq j$
$S_i \to X S_i X | a_i$ for any $i \in \{1, 2, ..., |\Sigma|\}$ and where a_i is the i^{th} symbol in Σ
$X \to a$ for any $a \in \Sigma$

5. (**40 points**) A *type 0 grammar* is like a context-free grammar, except that the rules are of the form $\alpha \to \beta$ where α and β can be arbitrary strings of variables and terminals.

   (a) (**10 points**) Write a type-0 grammar that generates the language

$$\{s \in \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^* \mid \exists n \in \mathbb{Z}^{\geq 0}.\ s = \mathsf{a}^n \mathsf{b}^n \mathsf{c}^n\}$$

   Solution
   Thanks to Flavio for this grammar:
   $S \to aQbc|\epsilon$
   $Q \to aQbT|\epsilon$
   $bTbc \to bbcc$
   $Tb \to bT$

   (b) (**10 points)** Show that every language that is generated by a type 0 grammar is Turing recognizable.
   Solution
   For a given type 0 grammar, a nondeterministic 2-tape TM could store valid grammar rules on one tape and the input string on another. Rules can nondeterministically be applied to a nondeterministically chosen substring of the input, applying the reverse of the grammar rule. The NTM accepts if the input string contains only the initial symbol of the grammar.

   (c) (**20 points)** Show that every language that is Turing recognizable is generated by a type 0 grammar.
   Solution
   Design a type 0 grammar that simulates the reverse computation of a TM. All intermediate strings of the grammar are invariant in that they have exactly one nonterminal, interpreted by symbol as the state of the TM and interpreted by position as the tape-head position. We allow all grammar rules that are the reverse of TM transitions while maintaining the invariant; thus the nonterminal only moves by one string position per grammar rule. The grammar begins with a random string containing the symbol corresponding to the TM accepting state. By assuming that the TM never transitions to the initial state, we allow the grammar to replace the nonterminal symbol for this state to ϵ. By construction, the TM accepts a string iff the corresponding type 0 grammar generates it.

6. (**50 points**) Let $\Sigma = \{\mathtt{1}\}$.

   (a) (**15 points**) Show a language, $F_1 \subseteq \Sigma^*$ such that F_1 is not Turing decidable.

   (b) (**15 points**) Let $F_2 \subseteq \Sigma^*$ be context-free. Show that F_2 is regular.

   (c) (**20 points**) Let $F_3 \subseteq \Sigma^*$ be *any* language. Show that F_3^* is regular.