Problem Set 9

Spring 10

Due: Tues 27 April at 2pm in class before the lecture.

Please follow the homework format guidelines posted on the class web page:

`http://www.cs.uiuc.edu/class/sp10/cs373/`

1. CFG design [**Category**: Comprehension, **Points**: 20]

   Consider this *straight line* programming language without any support for loops or conditional statements: A typical straight line program consists of several *sentences* separated by semicolon(;). Three different kinds of sentences are supported:

   - Variable definitions: using keyword *var* we can define a variable, for example:
     var abc
     defines a variable with name abc. Only letters $a \ldots z$ and $A \ldots Z$ are allowed in variable names.
   - Assignments: using symbol *:=* we can assign value of some *expression* to a variable. Left hand side of this symbol is always a variable name and right hand side is always an expression. An expression is composion of variable names and natural numbers using arithmetic operations +, -, * and /. A single natural number or a single variable name is also an expression. Some samples:
     abc := -23
     a:= 2
     v := 12 * v / abc - 4
     v := uvy
   - Output statements: using keyword *print* we can print a variable value to the output. For example:
     print abc
     prints the value of variable abc to the output.

   The language is case sensitive and the only allowed characters are +, -, *, /, ;, :, =, $a \ldots z$, $A \ldots Z$, $0 \ldots 9$, ␣ (space character), ∼ (new line).

   Design a context free grammer for this language (that is, construct a grammer G such that $L(G)$ is the set of all valid programs in this language. Don't forget to mention all four important pieces of a CFG explicitly). Note that a program is valid if and only if it conforms to the previous defined rules for this language. Becareful not to add any rules by your intuition, for example we never required a variable name to be declared by the keyword *var* before being used, so for example this is a valid program:
   abc :=5; var Acd ;
   Acd := abc * 6 * pqr; print fg
   Checking those kind of restrictions that are usually needed for serious programming languages requires more machinery.

2. CFG decoding [**Category**: Comprehension, **Points**: 20]

(a) Consider the grammar G_1 with the set of productions shown below (S is the start variable). What is $L(G_1)$?

$$S \implies \# \mid 0S1 \mid 1S0$$

(b) Consider the grammar G_2 with the set of productions shown below (S is the start variable). What is $L(G_2)$?

$$\begin{aligned} S &\implies \# \mid ASA \\ A &\implies 0 \mid 1 \end{aligned}$$

(c) Consider the grammar G_3 with the set of productions shown below (S is the start variable). What is $L(G_3)$?

$$\begin{aligned} S &\implies 1B1 \mid 0B0 \mid ASA \\ B &\implies ABA \mid \# \\ A &\implies 0 \mid 1 \end{aligned}$$

(d) What is the relation between $L(G_1)$, $L(G_2)$ and $L(G_3)$?

3. CYK [**Category**: Comprehension, **Points**: 20]

Use CYK algorithm to determine whether or not the given string belongs to the grammar. Your answer should include either "yes" or "no" and a chart that you built using CYK.

(a) Which of the following words belong to $L(G_2)$: *aabbb*, *aabab*?

$$\begin{aligned} S &\implies AP \mid AB \\ E &\implies AP \mid EB \mid b \\ P &\implies EB \\ A &\implies a \\ B &\implies b \end{aligned}$$

(b) Which of the following words belong to $L(G_1)$: *cadba*, *cbaad*?

$$\begin{aligned} S &\implies PE \mid CQ \mid a \\ E &\implies PE \mid CQ \mid a \\ P &\implies EB \\ Q &\implies ED \\ B &\implies b \\ C &\implies c \\ D &\implies d \end{aligned}$$

4. Decidability [**Category**: Proof, **Points**: 20]

   Prove that given a CFG G checking whether or not $L(G) \subseteq a^*b^*$ is a decidable problem.

5. CNF Conversion [**Category**: Proof., **Points**: 20]

   Begin with the grammar G:

$$\begin{array}{lcl} S & \rightarrow & aAa \mid bBb \mid \epsilon \\ A & \rightarrow & C \mid a \\ B & \rightarrow & C \mid b \\ C & \rightarrow & CD \mid \epsilon \\ D & \rightarrow & A \mid B \mid ab \end{array}$$

   (a) Eliminate ϵ-productions, obtaining G_1. (8 Points)

   (b) Eliminate any unit productions in G_1, obtaining G_2. (6 Points)

   (c) Put G_2 into Chomsky Normal Form G_3. (6 Points)

6. Closure Property [**Category**: Proof., **Points**: 20]

   Prove the language $L_s = \{a^n b^n c^m \mid m = n - 1 \textit{ or } m = n + 1\}$ is not context-free using only closure properties. (You may assume that $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.)