Assignments CS 336 Spring 2009

Assignment 1, due Monday, January 26

1. (8) In each case, describe precisely, using either words or a regular expression, what language is generated by the context-free grammar with the given productions.

   (a) $S \to aS \mid bS \mid \Lambda$

   (b) $S \to SaS \mid b \mid \Lambda$

   (c) $S \to TT \qquad T \to aT \mid Ta \mid b$

   (d) $S \to aT \mid bT \qquad T \to aS \mid bS \mid \Lambda$

2. (3) Exercise 6.4a

3. (4) In each case, the productions in a context-free grammar G are shown. Find a string x in $\{a, b\}^*$ so that $n_a(x) = n_b(x)$ but $x \notin L(G)$.

   (a) $S \to SabS \mid SbaS \mid \Lambda$

   (b) $S \to aSb \mid bSa \mid abS \mid baS \mid Sab \mid Sba \mid \Lambda$

4. (9) In each case find a CFG (i.e., write the productions) generating the language.

   (a) $\{a^i b^j \mid i \leq j\}$

   (b) $\{a^i b^j \mid i < j\}$

   (c) $\{a^i b^{2i} \mid i \geq 0\}$

5. (10) Suppose G is the context-free grammar with productions $S \to aSb \mid ab \mid SS$. Show using mathematical induction that no string in $L(G)$ begins with abb.

Assignment 2, due Monday, February 9

1. (6 pts) Show using mathematical induction on the number of steps in a derivation that every string produced by the context-free grammar with productions

$$S \to a \mid aS \mid bSS \mid SSb \mid SbS$$

   has more a's than b's. In other words, prove the statement: for every $n \geq 1$, if $S \Rightarrow^n x$, then x has more a's than b's.

2. (8 pts) In each part, draw an NFA (which might be an FA) accepting the language generated by the CFG having the given productions.

   (a) $S \to aA \mid bC \qquad A \to aS \mid bB \qquad B \to aC \mid bA \qquad C \to aB \mid bS \mid \Lambda$

(b) $S \rightarrow bS \mid aA \mid \Lambda \qquad A \rightarrow aA \mid bB \mid b \qquad B \rightarrow bS$

3. (5 pts) Give a brief description of an algorithm for starting with a regular grammar and finding an equivalent unambiguous grammar. (You can use any algorithms that we discussed in CS 335.)

4. (2,2,6,3,4) All parts refer to the CFG with productions $S \rightarrow a \mid S + S \mid S * S \mid (S)$.

   (a) How many distinct derivations (not necessarily leftmost or rightmost) does the string $a + (a * a)$ have?

   (b) How many derivation trees are there for the string $(a + (a + a)) + (a + a)$?

   (c) Suppose we define n_i to be the number of distinct derivation trees for the string $a + a + \ldots + a$ in which there are i a's. Then $n_1 = n_2 = 1$. Find a recursive formula for n_i, by first observing that if $i > 1$ the root of a derivation tree has two children labeled S, and then considering all possibilities for the two subtrees.

   (d) How many derivation trees are there for the string $a + a + a + a + a$?

   (e) How many derivation trees are there for the string $a+a+a+a+a+a+a+a+a+a$?

5. exercise 7.1 (2 pts for each string). For each one, show the configuration at each step. So the first one would start $(q_0, bbcbb, Z_0) \vdash \ldots$

6. exercise 7.3 (4 pts) Note that once the PDA leaves the state q_0, there are no more choices of moves. In order to answer this question, you don't need to know how many moves there are after that point—because there's only one possible *sequence* of moves after that.

7. (6 pts) What language (a subset of $\{a, b\}^*$) is accepted by the following PDA, if q_3 is the only accepting state?

Move No.	State	Input	Stack Symbol	Move(s)
1	q_0	a	Z_0	$(q_0, xZ_0), (q_1, aZ_0)$
2	q_0	b	Z_0	$(q_0, xZ_0), (q_1, bZ_0)$
3	q_0	a	x	$(q_0, xx), (q_1, ax)$
4	q_0	b	x	$(q_0, xx), (q_1, bx)$
5	q_1	a	a	(q_1, a)
6	q_1	b	b	(q_1, b)
7	q_1	a	b	$(q_1, b), (q_2, \Lambda)$
8	q_1	b	a	$(q_1, a), (q_2, \Lambda)$
9	q_2	a	x	(q_2, Λ)
10	q_2	b	x	(q_2, Λ)
11	q_2	Λ	Z_0	(q_3, Z_0)

Note that the PDA can stay in state q_0 by pushing an x onto the stack for every input symbol read. It enters q_1 by pushing onto the stack, not an x, but the symbol it has just read. What you have to think about is how the PDA can get to the accepting state (q_3) once it has moved to state q_1. Notice that in q_1 there is always the option of just ignoring the input symbol that is read and leaving the stack alone, but in order to reach q_3 it must eventually take the other option, which changes the state to q_2 (the only state from which the PDA can enter q_3).

Assignment 3, due Monday, March 9

1. Exercise 7.21(b) (5 pts)
2. Exercise 7.24(b) (5 pts)
3. Exercise 7.27(a) (4 pts)
4. Exercise 8.1(a,b,d) (4 pts each)
5. Exercise 8.5(a,b,e,g) (2 pts each—just answers, no proofs necessary)
6. 8.8(a) (5 pts)
7. 8.10 (3 pts each part)
8. 8.11 (2 pts each part, no reasons necessary)

These are problems to look at before the 2nd exam. Some of them may be included on a future assignment.

1. Draw transition diagrams for TMs accepting the following languages:
   (a) $\{x \in \{a, b, c\}^* \mid$ exactly half the symbols of x are a's $\}$
   (b) $\{x \in \{a, b\}^* \mid |x|$ is odd and the middle symbol of x is $a\}$
   (c) $\{x \in \{a, b\}^* \mid$ each symbol of x that appears in a position of the form $3k + 1$ is an $a\}$ (i.e., the first symbol, the 4th symbol, . . .)
2. Exercise 9.7
3. Exercise 9.15d

Below are two assignments that are due at the end of the semester.

Turing Machine Assignment

This assignment is to construct a Turing machine and to test it by using an online Turing machine simulator until you are certain that it works correctly.

The TM is to perform multiplication of two nonnegative integers. This can be described by saying that it computes the function $m : \mathcal{N} \times \mathcal{N} \to \mathcal{N}$ defined by $m(x, y) = x * y$. As usual, natural numbers are to be represented in unary notation, so that the number n is represented by a string of n 1's.

For example, if the initial configuration of the TM is

$$(q_0, \underline{\Delta}111\Delta11)$$

then the final configuration should be

$$(h_a, \underline{\Delta}111111)$$

You can find a TM simulator at

http://www2.mta.ac.il/ amirben/tm.html

(There may be better ones, but I believe that this one works correctly.) You can enter the individual moves one at a time in the space marked "Programming." A move is represented by a 5-tuple:

state, symbol, new state, new symbol, direction

where the 5 things are separated by commas. The states are represented by positive integers, except that the accepting state is represented by A and the reject state by H. The symbols can be anything. A blank is represented by _ (underscore), and a direction is represented by > or <. (As far as I can tell, the tape head must move either to the right or to the left on each move.) The initial state is assumed to be the first state in the first line. For example

1, _, 2, _, >

says that in state 1, if the symbol is blank, go to state 2, don't change the symbol, and move the tape head one square to the right.

As far as I can tell, there's no way to get a printout directly from the simulator, but you should turn in the following:

1. A transition diagram of your Turing machine
2. A brief description of the simulator you used
3. A listing of the lines you entered to program the simulator
4. A listing of several initial configurations that you tried, and the resulting output from each one

Assignment to Consult Other Sources

In this assignment, you are to answer the following questions. You will be able to find the answers by consulting online sources or reference books or journal articles. The general topic is generally known as Kolmogorov Complexity, although it has other names, such as Descriptive Complexity, Chaitin Complexity, etc.

You should turn in your answers to the questions as well as citations indicating where you obtained the answers.

1. What is meant by the Kolmogorov complexity $K(x)$ of a string x?
2. What is meant by an incompressible string over an alphabet?
3. Explain briefly (you don't need to present much technical detail, if any) why there are infinitely many incompressible strings over the alphabet $\{a, b\}$.
4. Explain briefly why it is impossible in general to compute the complexity $K(x)$ of a string x, or to show that a string x is incompressible.
5. What is Berry's paradox?
6. What is Gödel's Incompleteness Theorem?
7. How are Berry's paradox and Gödel's Incompleteness Theorem related to the first 4 questions?
8. What relations do you see between these topics and undecidability results involving Turing machines?

Assignment 4, due sometime next week.

Exercise 9.33c (4 pts), Exercise 9.39b (4 pts), and the following exercise (3 pts each part):

Suppose G is an unrestricted grammar with start symbol T that generates the language $L \subseteq \{a, b\}^*$. In each part, another unrestricted grammar is described. Say what language it generates.

1. The grammar containing all the variables and all the productions of G, two additional variables S (the start variable) and E, and the additional productions

$$S \to ET \qquad E \to \Lambda \qquad Ea \to E \qquad Eb \to E$$

2. The grammar containing all the variables and all the productions of G, four additional variables S (the start variable), L, R, and E, and the additional productions

$$S \to LTR \qquad La \to aL \qquad Lb \to bL \qquad L \to E$$
$$Ea \to E \qquad Eb \to E \qquad ER \to \Lambda$$

(There will probably be a couple more problems added within the next couple of days.)

Assignment 5, due Friday, May 8

11.8 (4 pts) Just say briefly what the reduction is—i.e., given a pair (T, w), how do you construct a TM T_1 so that T accepts w if and only if T_1 accepts x?

11.9 (5 pts)

11.12 (d) (4 pts) The answer is that the problem is unsolvable. Show this, by reducing another unsolvable problem to it.

11.15 (5 pts)