

Exercises, CS 336

I. Due Wednesday, January 16

Exercise 6.1(a,f,g,h) (3 pts each) In each case, you can give a simple description in words of the language. The answer to part (e), for example, is the language of all strings that end with a .

6.4(b) (4 pts)

6.6 (4 pts) If your answer is yes, you don't need to give a proof. If your answer is no, "prove your answer" means either find a string x so that x has equal numbers of a 's, b 's, and c 's but cannot be generated by the grammar, or find a string x that is generated by the grammar that does not have equal numbers of a 's, b 's, and c 's (and say which of these two cases is the right one).

6.9(h) (5 pts) Hint: First find a grammar that generates the language $\{a^i b^i \mid i \geq 0\}$, then find another grammar that generates the language $\{a^i b^{2i} \mid i \geq 0\}$. You can do both of these by just thinking about recursive definitions of the two languages. Once you have these two grammars, then figure out how to combine them so as to answer the question.

Total: 25 pts.

II. Due Wednesday, Jan. 31

6.9(d) (5 pts)

6.13 (4 pts) Note that a regular grammar is one in which all productions are of the form $A \rightarrow aB$ or $A \rightarrow a$, where A and B are variables and a is a terminal.

6.19(c,d,e) (2 pts, 3 pts, 1 pt)

6.21 (8 pts) Note that there are 4 questions, two in (a) and two in (b). The answer to each is a number.

6.26(a,e) (4 pts each part) In each case, show the grammar is ambiguous by drawing two different derivation trees for some string. In each case the grammar you're looking for is supposed to generate exactly the same language as the original one and is supposed to be unambiguous.

6.48 (4pts) No proof required.

III. Due date to be specified later

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

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.

Supplementary problem (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).

7.6 (4 pts for each part)

7.13(a) (5 pts)

IV. Due Friday, February 29

7.21(b) (4 pts) Note: this PDA is the standard nondeterministic one, in which the possible moves are (i) Λ -transitions that replace a variable on top of the stack by the right side of a production starting with that variable; or (ii) moves that read a terminal symbol and cancel it with the same symbol on top of the stack.

7.23 (4 pts) The question is not asking about grammars for which the standard nondeterministic top-down PDA can be converted to a deterministic PDA. (We've looked at examples like this in class.) It's asking about grammars for which the standard nondeterministic top-down PDA *IS* deterministic. Think about what would have to be true in order for this to happen.

7.27(b) (5 pts)

8.1(a,b) (5 pts each)

8.10 (9 pts) Give reasons for part (a), which is worth 5 pts; no reasons necessary for (b) and (c), which are worth 2 pts each.

8.11 (2 pts each part) (no reasons necessary)

V. Due date to be set later

9.2(b,c) (2 pts each part)

9.4 (3 pts)

9.6(a,d) (4 pts each part)

9.7 (5 pts)

9.15(b) (5 pts)

VI. Due Monday, April 14

9.29 (3 pts)

9.30 (3 pts)

9.33(b) (3 pts)

10.2 (3 pts each part) No reasons necessary

10.3 (3 pts). The statement is false. Assuming that there is a language $L \subseteq \{a,b\}^*$ that is not recursively enumerable, give an example to illustrate its falsity.

10.11(b,c) (3 pts each) In part (b), answer the question by filling in the blank in the expression $\{a^n \mid \underline{\hspace{2cm}}\}$ with a precise statement about n .

VII. Due Monday, April 28

10.27 (3 pts) Suggestion: proof by contradiction.

10.31(d,e,h,i) (2 pts each) no reasons required.

11.9 (4 pts) Here's what you have to do to answer this question. Starting with a TM T , explain how to obtain a TM T_1 so that T accepts Λ if and only if T_1 accepts the language $\{\Lambda\}$. (In other words, so that $\Lambda \in L(T)$ if and only if $L(T_1) = \{\Lambda\}$.) T_1 can be described in terms of T . In particular, you need to say what T_1 should do if its input is Λ , and what it should do if its input is a nonnull string.

11.10 (b) (3 pts) Here's a way to answer part (a). Let $C = A \cup B$, and let $D = A \cap B$. If $A = B$, then clearly $C = A \cup B = A = B$, and $D = A \cap B = A = B$, so $C \subseteq D$. Conversely, if $A \cup B \subseteq A \cap B$, then since $A \subseteq A \cup B$ and $A \cap B \subseteq B$, it follows that $A \subseteq B$. Similarly, since $B \subseteq A \cup B$ and $A \cap B \subseteq A$, it follows that $B \subseteq A$. Therefore, if $C \subseteq D$, then $A = B$.

Now, use this to answer (b).

11.11 (3 pts each part) I answered (a) in the previous problem. You try to do something similar for (a) in this problem.

I'm not assigning exercise 11.12 for you to turn in, but you should think about this exercise, and answer as many of the parts as you can.