UVa - cs302: Theory of Computation Spring 2008

Exam1 - Comments

Due: 11 March 2008

Corrected, 12 March 2008.

Score Distributions

Question/Part	Target	0	1, 2	3	4	5	6	7	8	9-12	13-17	18-19	20
1a	5	0	2	10	15	58							
1b	5	0	4	40	38	3							
1c	5	1	43	17	17	7							
1d	5	0	4	9	50	22							
2a	5	0	11	0	18	56							
2b	5	0	19	2	6	58							
2c	5	0	16	0	1	68							
2d	5	3	36	7	27	12							
3a	7	2	4	24	4	8	7	36					
3b	7	1	16	17	5	2	0	39	5				
3c	7	6	11	29	21	0	1	17					
4a	8	0	3	1	7	7	19	4	44				
4b	5	1	18	12	9	45							
4c	8	3	4	2	5	7	11	8	45				
5a	20	1	3	1	8	21	4	0	11	7	6	4	19
5b		77	1	0	0	3	0	0	0	4			

	≤ 50	51–60	61–70	71–80	81–90	> 90
Total	6	13	23	13	18	12

Problem 1: Definitions. (20) For each question, provide a correct, clear, precise, and concise answer from the perspective of a theoretical computer scientist.

a. What is a *language*?

Answer: A set of strings.

b. What is a *computer*?

Answer: Something that can take an input, perform processing by carrying out a sequence of precisely-defined steps, and produce an output.

c. What is a *problem*?

Answer: A specification of a set of possible inputs and property of a desired output.

d. What does it mean for a machine to *recognize a language*?

Answer: The language recognized by a machine is the set of strings the machine accepts. A machine, M, recognizes a language, L, if M accepts every string in L and no strings that are not in L.

Problem 2: Language Classification. (20)

For each described language, your answer should identify the simplest machine that can recognize the given language. The choices for machines (from simplest to least simple) are:

(A) Deterministic Finite Automaton with no cycles
(B) Deterministic Finite Automaton
(C) Deterministic Push-Down Automaton
(D) Nondeterministic Push-Down Automaton
(E) None of the Above

Your answer should include one of the five letters to identify the machine, and a brief justification. For these questions, you are *not* required to actually construct the machine or provide a proof for a full credit answer; a correct intuitive explanation is sufficient.

a. $\{w | w$ describes the position of a chess board in which White can win $\}$

Answer: A — the language is finite, so it can be recognized by a DFA with no cycles.

b. The language of all strings generated by the grammar: $S \rightarrow 0S \mid 1S \mid \epsilon$.

Answer: B — the grammar generates the language $\{0, 1\}^*$. This is infinite, so cannot be recognized by A, but it is regular so it can be recognized by B (a DFA with one state, which accepts, and transitions back to that state on both inputs).

c. $\left\{0^i 1^{2i} 0^{3i} | i \geq 0\right\}$

Answer: E — the language is not a context-free language so it cannot be recognized by an NDPDA. A formal proof would use the pumping lemma to prove the language is not context-free. Here, it is enough to argue that recognizing the language requires counting more than one thing — we need to count the number of 0s at the beginning, then count that there are twice as many 1s, but can't do this by popping the stack, since we still need to know i for the 0s at the end.

d. The language of all strings in $\{a, b\}^*$ that do not include any substring of the form $a^i b^i$ where $i \geq 1$.

Answer: B — all we need to do is determine if ab occurs in the string! We can do this with a DFA (but it must have cycles, since there are infinitely long strings in the language).

Problem 3: Problematic Proofs. (20)

Each of these "proofs" claims to prove a conjecture that is false. For each proof, identify the *first step* that is wrong, and briefly and clearly explain why.

a. **False Theorem:** The language

$$L = \{w | w \in \{0,1\}^* \text{ and the number of 1s in } w \text{ is odd } \}$$

is non-regular.

Claimed Proof. We use proof-by-contradiction to show L is not regular.

1. Assume L is regular.

2. Since L is regular, the pumping lemma should be satisfied.

3. Choose $s = 1$, which satisfies the requirements for the pumping lemma since it is in L (it has an odd number of 1s).

4. The only way to satisfy the pumping lemma requirements for dividing $s = xyz$ is to make $x = \epsilon$, $y = 1$, and $z = \epsilon$ since the pumping lemma requires $|y| > 0$.

5. But, this division leads to a contradiction of the pumping lemma, since for $i = 2$, $xy^iz = 11$ which is not in L (it has an even number of 1s).

6. Since the pumping lemma is not satisfied, the assumption must be invalid.

7. Thus, L is not regular.

Answer: Step 3. To use the pumping lemma, we need to find a string s with $|s| \geq p$ for any p. The length of 1 is one, so it doesn't work if $p > 1$.

b. **False Theorem:** All regular languages include the empty string.

Proof by induction. We induce on the size of the language.

1. *Basis*: A language of size 1 includes the empty string: the language $\{\epsilon\}$ is a regular language and it includes the empty string.
2. *Induction*: We assume that all regular languages of size n include the empty string, and show that all regular languages of size $n+1$ include the empty string.

2a. Suppose L is a language of size $n+1$.
2b. For some language M of size n, $L = M \cup w$ where w is some string not in M.
2c. By the induction hypothesis, M includes the empty string, so $L = M \cup w$ also includes the empty string.

Answer: Step 1. The induction is on the size of the language, so the basis needs to show the property holds for *all* languages of size 1, not just one language of size 1. This isn't true since there are languages of size 1 that do not include the empty string.

c. **False Theorem:** If A and B are context-free languages, then $A \cap B$ is a context-free language.

Proof by construction.

1. Since A and B are context-free languages, there exist deterministic pushdown automata $P_A = (Q_A, \Sigma, \Gamma, \delta_A, q_{0A}, F_A)$ and $P_B = (Q_B, \Sigma, \Gamma, \delta_B, q_{0B}, F_B)$ that recognize A and B respectively.

2. We can construct the deterministic pushdown automaton $P_{AB} = (Q_{AB}, \Sigma, \Gamma_{AB}, \delta_{AB}, q_{0AB}, F_{AB})$ that recognizes the intersection of A and B by simulating both P_A and P_B simultaneously:

2a. $Q_{AB} = Q_A \times Q_B$
2b. $\Gamma_{AB} = \Gamma_A \times \Gamma_B$
2c. $\delta_{AB}((q_a, q_b), \sigma, (\gamma_a, \gamma_b)) = ((r_a, r_b), (y_a, y_b))$
where $\delta_A(q_a, \sigma, \gamma_a) = (r_a, y_a)$ and $\delta_B(q_b, \sigma, \gamma_b) = (r_b, y_b)$
2d. $q_{0AB} = (q_0A, q_0B)$
2e. $F_{AB} = \{(q_a, q_b) | q_a \in F_A \wedge q_b \in F_B\}$

3. Since P_{AB} accepts only if both P_A and P_B would accept, the language recognized by P_{AB} is the intersection of the languages recognized by P_A and P_B.

Answer: Step 1. We cannot assume that there exist *deterministic* pushdown automata that recognize A and B. There exists a *nondeterministic* pushdown automaton that can recognize any context-free language, but (as proved in class) there are context-free languages (such as ww^R) that cannot be recognized by a *deterministic* pushdown automaton. The rest of the proof is (essentially) correct (except for not explaining how to deal with transitions that push or pop ϵ): the intersection of two deterministic context-free languages is indeed a deterministic context-free language, and we could prove this by constructing a DPDA that recognizes their intersection.

Problem 4: Proofs. (20) For each part, write a clear, concise, and convincing proof. You may use the pumping lemma for regular languages, pumping lemma for context-free languages, as well as the closure properties for regular and context-free languages.

a. Prove that the language $\left\{0^i 1^j | i, j \geq 0 \wedge i \neq j\right\}$ is not regular.

Answer: The easiest way to prove this is to prove that its complement,

$$\bar{L} = \left\{0^i 1^j | i, j \geq 0 \wedge i = j\right\} \cup (0 \mid 1)^* 10 (0 \mid 1)^*$$

is non-regular. Since regular languages are closed under complement, this proves the given language is also non-regular. We prove by contradiction using the pumping lemma: Assume $\bar{L}$ is regular, so the pumping lemma holds for some pumping length p. Choose $s = 0^p 1^p$ which is in the language. The pumping lemma says we can divide $s = xyz$ where $|xy| \leq p$, so xy must be within the first p symbols of s which contain only 0s. Since the pumping lemma requires $|y| > 0$, we know y must contain one or more 0s. Pumping y increases the number of 0s, but not the number of 1s, so will produce a string of the form $0^{p+k} 1^p$ where $k = |y| i > 1$ that is not in the language.

We could also prove the given language is non-regular, but need to be very careful here since we cannot assume $|y| = 1$. Proof by contradiction. Assume the language is regular, so the pumping lemma holds for some pumping length p. Choose $s = 0^p 1^p$ which is not in the language. The pumping lemma says we can divide $s = xyz$ where $|xy| \leq p$, so xy must be within the first p symbols of s which contain only 0s. Since the pumping lemma requires $|y| > 0$, we know y must contain one or more 0s. Pumping y increases the number of 0s, but not the number of 1s, so will produce a string that is in the language. (Note that we go from a string not in the language to one that is in the language here; it is much tougher to write a correct proof that goes from a string in the language to one that is not in the language since it is difficult to argue that i and j will be made exactly equal by pumping.)

b. Prove that the language $\{a^i b^j a^j b^i\}$ is context-free where $i, j \geq 0$.

Answer: We can prove a language is context-free by constructing an NDPDA or CFG that recognizes the language. Here is a CFG:

$$S \rightarrow aSb \mid X$$
$$X \rightarrow bXa \mid \epsilon$$

c. Prove that the language $\left\{0^n 1^{n^2}\right\}$ (the number of 1s is the square of the number of 0s) is not context-free.

Answer: We prove by contradiction using the pumping lemma for context-free languages. Assume L is context-free, so there is some pumping length p for which the pumping lemma for CFLs holds.

Choose $s = 0^p 1^{p^2}$. According to the pumping lemma, there is some division $s = uvxyz$ such that uv^ixy^iz is in L for any $i \geq 0$. Note that for whatever division we choose $v = 0^a 1^b$ and $y = 0^c 1^d$ with no constraints on $a, b, c, d \geq 0$, and $uxz = 0^f 1^g$. So, for a string to be in the language the number of 1s must equal the square of the number of 0s: $b + d + g = (a + c + f)^2$. This holds initially (since $s \in L$) for $i = 1$. For a given i, we need $ib + id + g = i(b + d) + g = (i(a + c) + f)^2$. This cannot be satisfied unless $b + d = a + c = 0$, but it is not possible for then all to be zero since the pumping lemma requires $|v| > 0$ or $|y| > 0$.

Problem 5: Closure. (20) Define $H(L)$ as the set of even-length strings in L. That is,

$$H(L) = \{w | w \in L \text{ and } |w| = 2k \text{ for some } k \geq 0\}$$

a. If L is a regular language, is $H(L)$ a regular language? (State clearly "Yes" or "No", and support your answer with a convincing proof.)

Answer:

Yes. The easiest proof uses the intersection closure property for regular languages. The language of even-length strings is regular: $L_{even} = (00 \cup 01 \cup 10 \cup 11)^*$. For any language L, $H(L) = L \cap L_{even}$. The intersection of two regular languages is a regular language.

We could also prove this by construction. Since L is regular, there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes L. We can construct a DFA $M_H = (Q_H, \Sigma, \delta_H, q_{0H}, F_H$ that recognizes $H(L)$ as:

$Q_H = Q \times \{even, odd\}$ — states in M_H keep track of the state in M, and whether an even or odd number of symbols have been seen so far.

δ_H is defined by:

$$\delta_H((q, even), x) = (\delta(q, x), odd)$$
$$\delta_H((q, odd), x) = (\delta(q, x), even)$$

$q_{OH} = (q_0, even)$

$F_H = \{(q_f, even) | q_f \in F\}$

b. [Bonus] If L is a context-free language, is $H(L)$ a context-free language? (State clearly "Yes" or "No", and support your answer with a convincing proof.)

Answer:

Yes. The easiest proof would use the theorem that the intersection of a context-free language and a regular language is a context-free language. From this, it follows that $H(L)$ is closed for context-free languages since $H(L)$ is the intersection of L with the regular language of even-length strings (as explained in part a). But, we didn't prove that theorem, so will prove this without using it. (From this proof, it should be clear how to prove the more general theorem.)

Proof by construction. Since L is a CFL, there exists a NDPDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$. We show how to construct a machine $M_H = (Q_H, \Sigma, \Gamma, \delta_H, q_{0H}, F_H)$ that recognizes $H(L)$. The basic idea is to make two copies the states of Q, corresponding to having seen an odd or even number of input symbols so far, and duplicating all the transition rules as in part a.

$Q_H = Q \times \{even, odd\}$ — states in M_H keep track of the state in M, and whether an even or odd number of symbols have been seen so far.

δ_H is defined by:

$\delta_H((q, even), x, \gamma) = ((q_r, odd), \beta)$ if $\delta(q, x, \gamma) \rightarrow (q_r, \beta)$
$\delta_H((q, odd), x, \gamma) = ((q_r, even), \beta)$ if $\delta(q, x, \gamma) \rightarrow (q_r, \beta)$
$\delta_H((q, t), \epsilon, \gamma) = ((q_r, t), \beta)$ if $\delta(q, \epsilon, \gamma) \rightarrow (q_r, \beta)$

$q_{OH} = (q_0, even)$
$F_H = \{(q_f, even) | q_f \in F\}$