INSTRUCTOR'S SOLUTION

**THEORY OF COMPUTATION**
**CSCI 320, section 11, class # 54495**
**Test # 2**
May 6, 2019
instructor: Bojana Obrenić

**<u>NOTE:</u> It is the policy of the Computer Science Department to issue a failing grade in the course to any student who either gives or receives help on any test.**

**To pass this test, you are required to follow in full the test protocol described below:**

**Queens College photo-ID** is required;

this is a **closed-book** test, to which it is **forbidden** to bring any material except pencils (pens) and erasers—in particular, bringing any electronic device is a direct violation of the test protocol;

**student name** has to be written **clearly** on **each page** of the problem set during the **first five minutes of the test**—there is a penalty of **at least 1 point** for each missing name;

**answers** should be written only in the space marked "**Answer:** " that follows the statement of the problem (unless stated otherwise);

your **handwriting** must be legible so as to leave no ambiguity whatsoever as to what exactly you have written;

any problem to which you give **two or more (different) answers** receives the **grade of zero** automatically;

if you have written something into the answer space by **mistake**, **cross it out completely or erase it** and it will not be graded;

**scratch** should never be written in the answer space, but may be written on the (empty) back of the problem pages, the content of which *will not be graded;*

when requested, **hand in** the problem set;

**once you leave** the classroom, you cannot come back to the test;

You may work on as many (or as few) problems as you wish. Good luck.

time: 75 minutes.
**full credit: 75 points (corresponds to normalized score of 100%.)**
$C$: 30 points (corresponds to normalized score of 40%.)

| problem: | 01 | 02 | 03 | 04 | 05 | 06 | 07 | normalized score |
|---|---|---|---|---|---|---|---|---|
| grade [points]: | | | | | | | | % |

INSTRUCTOR'S SOLUTION

**Problem 1** [ **15 points** ] Let $L$ be the language accepted by the pushdown automaton:
$M = (Q, \Sigma, \Gamma, \delta, s, F)$ where: $Q = \{s, x\}$;
$\Sigma = \{a, b, c, g\}; \Gamma = \{B, E, R, T, U\}; F = \{x\}$; and $\delta$ is defined by the following transition set:

$$\begin{array}{lll} [s, c, \lambda, s, BUTTER] & [x, a, U, x, \lambda] & [x, g, R, x, \lambda] \\ [s, g, \lambda, s, BEER] & [x, b, B, x, \lambda] & [x, g, T, x, \lambda] \\ [s, a, \lambda, x, \lambda] & [x, c, E, x, \lambda] & \\ [s, b, \lambda, x, T] & & \end{array}$$

(Recall that $M$ is defined so as to accept by final state and empty stack.) Furthermore, if an arbitrary stack string, say $X_1 \ldots X_n \in \Gamma^*$ where $n \geq 2$, is pushed on the stack by an individual transition, then the leftmost symbol $X_1$ is pushed first, while the rightmost symbol $X_n$ is pushed last.)

**(a)** List four distinct strings that belong to $L$. If this is impossible, state it and explain why it is so.

**Answer:**

$$a, bg, gagccb, gbggccb$$

**(b)** Write a complete formal definition of a context-free grammar that generates $L$. If such a grammar does not exist, prove it.

**Answer:** $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c, g\}$, $V = \{S\}$, and $P$ is:

$$S \to cSgcggab \mid gSgccb \mid a \mid bg$$

**Note: This problem continues on the following page.**

**Problem 1 cont'd:**

INSTRUCTOR'S SOLUTION

Let $Q$ be a property of recursively enumerable languages, whose value for any language $Y$ is defined as follows:

$$Q(Y) \iff (Y \cap L \neq \emptyset)$$

(where $L$ is defined at the beginning of this problem.)

**(c)** Is $Q(\emptyset)$ true? Prove your answer.

**Answer:** No.

$$Q(\emptyset) \iff (\emptyset \cap L \neq \emptyset)$$

but

$$\emptyset \cap L = \emptyset$$

for every set $L$, hence $Q(\emptyset)$ is false.

**(d)** Is $Q(\Sigma^*)$ true? ($\Sigma$ is defined at the beginning of this problem.) Prove your answer.

**Answer:** Yes.

$$Q(\Sigma^*) \iff (\Sigma^* \cap L \neq \emptyset)$$

which is true since

$$\Sigma^* \cap L = L$$

and $L$ is not empty (as is shown in the answers to parts (a) and (b).)

**(e)** Is $Q(\boldsymbol{a^*})$ true? Prove your answer.

**Answer:** Yes.

$$Q(\boldsymbol{a^*}) \iff (\boldsymbol{a^*} \cap L \neq \emptyset)$$

Since $a \in \boldsymbol{a^*}$ but also $a \in L$ (as is stated in the answer to part (a)), indeed $\boldsymbol{a^*} \cap \boldsymbol{L}$ is not empty.

**(f)** Is property $Q$ nontrivial? Prove your answer.

**Answer:** Yes. As is shown in the answers to parts (a) and (b), $Q(\emptyset)$ is false, but $Q(\Sigma^*)$ is true. Hence, by definition, $Q$ is nontrivial.

INSTRUCTOR'S SOLUTION

**Problem 2** [ **15 points** ] Let $L$ be the language accepted by the pushdown automaton:
$M = (Q, \Sigma, \Gamma, \delta, s, F)$ where: $Q = \{s, t, x\}$;
$\Sigma = \{a, b, c, g\}; \Gamma = \{A, C, E, L, R, T\}; F = \{x\}$; and
$\delta$ is defined by the following transitions set:

| | | |
|---|---|---|
| $[s, a, \lambda, s, TREACLE]$ | $[x, a, A, x, \lambda]$ | $[x, c, C, x, \lambda]$ |
| $[t, c, \lambda, t, TART]$ | $[x, a, L, x, \lambda]$ | $[x, g, T, x, \lambda]$ |
| $[s, b, \lambda, t, \lambda]$ | $[x, b, E, x, \lambda]$ | |
| $[t, g, \lambda, x, R]$ | $[x, b, R, x, \lambda]$ | |

(Recall that $M$ is defined so as to accept by final state and empty stack.) Furthermore, if an arbitrary stack string, say $X_1 \dots X_n \in \Gamma^*$ where $n \geq 2$, is pushed on the stack by an individual transition, then the leftmost symbol $X_1$ is pushed first, while the rightmost symbol $X_n$ is pushed last.)

**(a)** List four distinct strings that belong to $L$. If this is impossible, state it and explain why it is so.

**Answer:**

$$bg, bcggbag, bccggbaggbag, abgbacabbg$$

**(b)** Write a complete formal definition of a context-free grammar that generates $L$. If such a grammar does not exist, prove it.

**Answer:** $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c, g\}$, $V = \{S, W\}$, and $P$ is:

$$S \rightarrow aSbacabbg \mid bW$$
$$W \rightarrow cWgbag \mid g$$

**Note: This problem continues on the following page.**

**Problem 2 cont'd:**



**(c)** Explain how to construct an algorithm that solves the following problem:

INPUT: Finite automaton $F$ that accepts a language $L_F$, and string $w$;

OUTPUT: **yes** if $w \in L_F \cup L$, where $L$ is the language defined at the beginning of this problem;
**no** otherwise.

If this algorithm does not exist, prove it.

**Answer:** Simulate $L_F(w)$ and accept if $L_F$ accepts. Otherwise, simulate $M(w)$ and decide exactly as $M$ decides.

**(d)** Explain how to construct an algorithm that solves the following problem:

INPUT: Turing machine $T$ that decides a language $L_T$, and string $w$;

OUTPUT: **yes** if $w \in L_T \setminus L$, where $L$ is the language defined at the beginning of this problem;
**no** otherwise.

If this algorithm does not exist, prove it.

**Answer:** Simulate $T(w)$ (which must terminate since $T$ decides) and reject if $T$ rejects. Otherwise, simulate $M(w)$ and decide the opposite of what $M$ decides.

**Problem 3** [ **18 points** ] Consider the Turing machine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ such that: $Q = \{s, p, x\}$; $\Sigma = \{0, 1\}$; $\Gamma = \{B, 0, 1\}$; $F = \{s\}$; and $\delta$ is defined by the following transition set:

$$\begin{array}{lll} [s,0,s,0,R] & [p,0,p,0,R] & [x,0,x,0,R] \\ [s,1,p,1,R] & [p,1,x,1,R] & [x,1,s,1,R] \\ [s,B,s,B,R] & & \end{array}$$

($M$ has an one-way infinite tape (infinite to the right only.) $B$ is the designated blank symbol. $M$ accepts by final state.)

Let $L$ be the set of strings which $M$ accepts.
Let $L_R$ be the set of strings which $M$ rejects.
Let $L_\infty$ be the set of strings on which $M$ diverges.

**(a)** List four distinct strings that belong to $L_\infty$. If this is impossible, state it and explain why it is so.

**Advice for Answer:** $M$ cannot halt in state $s$, so it does not accept any strings. If the number of 1's is divisible by 3, $M$ diverges, otherwise it halts in one of the two non-final states.

**Answer:**

$$\lambda, 0, 001110, 010101$$

**(b)** List four distinct strings that belong to $L_R$. If this is impossible, state it and explain why it is so.

**Answer:**

$$1, 01, 10, 11$$

**(c)** List four distinct strings that belong to $L$. If this is impossible, state it and explain why it is so.

**Answer:** Impossible—$L$ is empty and there are no strings to list.

INSTRUCTOR'S SOLUTION

**(d)** Write a regular expression that defines $L_\infty$. If such a regular expression does not exist, prove it.

**Answer:**

$$\mathbf{(0 \cup 10^*10^*1)^*}$$

**(e)** Write a regular expression that defines $L_R$. If such a regular expression does not exist, prove it.

**Answer:**

$$\mathbf{(0 \cup 10^*10^*1)^* \, (1 \cup 10^*1) \, 0^*}$$

**Note: This problem continues on the following page.**

**Problem 3 cont'd:**

INSTRUCTOR'S SOLUTION

**(f)** Write a regular expression that defines $L$. If such a regular expression does not exist, prove it.

**Answer:**

$$\emptyset$$

**(g)** Define the language (if any) which $M$ decides. Prove your answer. ($M$ is the machine defined at the beginning of this problem.)

**Answer:** $M$ does not decide any language, since it diverges on some input strings—exactly those in $L_\infty$.

**(h)** Explain how to construct an algorithm that solves the following problem:

INPUT: A Turing machine $T$ that accepts a language $L_T$.

OUTPUT: **yes** is $L \subseteq L_T$;
**no** otherwise.

($L$ is the language defined at the beginning of this problem.)

If this algorithm does not exist, prove it.

**Answer:** This algorithm always returns answer **yes**, since $L = \emptyset$ (as is stated in the answer to part (f)), and thus $L$ is a subset of every set (including $L_T$.)

INSTRUCTOR'S SOLUTION

**Problem 4** [ **20 points** ] Consider the Turing machine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ such that: $Q = \{s, p, x, v, z, w\}$; $\Sigma = \{a, b, c, g\}$; $\Gamma = \{B, a, b, c, g, A, K, D, G\}$; $F = \{w\}$; and $\delta$ consists of the following transition set:

$$\begin{array}{lll}
[s,a,p,A,R] & [p,a,x,D,R] & [x,a,x,a,R] \\
[s,b,s,b,R] & [p,b,p,b,R] & [x,b,x,b,R] \\
[s,c,p,K,R] & [p,c,x,G,R] & [x,c,x,c,R] \\
[s,g,s,g,R] & [p,g,p,g,R] & [x,g,x,g,R] \\
[s,B,s,B,R] & [p,B,p,B,R] & [x,B,v,B,L]
\end{array}$$

$$\begin{array}{ll}
[v,a,v,a,L] & [z,a,z,a,L] \\
[v,b,v,b,L] & [z,b,z,b,L] \\
[v,c,v,c,L] & [z,c,z,c,L] \\
[v,g,v,g,L] & [z,g,z,g,L] \\
[v,D,z,D,L] & [z,K,w,K,R]
\end{array}$$

($M$ has an one-way infinite tape (infinite to the right only.) $B$ is the designated blank symbol. $M$ accepts by final state.)

Let $L$ be the set of strings which $M$ accepts.
Let $L_R$ be the set of strings which $M$ rejects.
Let $L_\infty$ be the set of strings on which $M$ diverges.

**(a)** List four distinct strings that belong to $L_\infty$. If this is impossible, state it and explain why it is so.

**Advice for Answer:** $M$ diverges unless it finds at least two occurrences of symbols belonging to $\{a, c\}$. If such two symbols are found, then $M$ accepts if the leftmost two such symbols are $c$ and $a$ in that order, from left to right. Otherwise, $M$ rejects.

**Answer:**

$$\lambda, b, g, a$$

**(b)** List four distinct strings that belong to $L$. If this is impossible, state it and explain why it is so.

**Answer:**

$$ca, caa, bcab, gca$$

**(c)** List four distinct strings that belong to $L_R$. If this is impossible, state it and explain why it is so.

**Answer:**

$$aa, ac, cc, baa$$

**(d)** Write a regular expression that defines $L$. If such a regular expression does not exist, prove it.

**Answer:**

$$(\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{c} \, (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{a} \, (\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c} \cup \boldsymbol{g})^*$$

**(e)** Write a regular expression that defines $L_\infty$. If such a regular expression does not exist, prove it.

**Answer:**

$$(\boldsymbol{b} \cup \boldsymbol{g})^* \cup (\boldsymbol{b} \cup \boldsymbol{g})^* \, (\boldsymbol{a} \cup \boldsymbol{c}) \, (\boldsymbol{b} \cup \boldsymbol{g})^*$$

**Note: This problem continues on the following page.**

**Problem 4 cont'd:**



**(f)** Write a regular expression that defines $L_R$. If such a regular expression does not exist, prove it.

**Answer:**

$$\begin{array}{c} (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{a} \, (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{a} \, (\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c} \cup \boldsymbol{g})^* \\ \cup \\ (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{a} \, (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{c} \, (\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c} \cup \boldsymbol{g})^* \\ \cup \\ (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{c} \, (\boldsymbol{b} \cup \boldsymbol{g})^* \, \boldsymbol{c} \, (\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c} \cup \boldsymbol{g})^* \end{array}$$

**(g)** List four distinct strings that belong to $(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c} \cup \boldsymbol{g})^*$ such that the machine $M$ (defined at the beginning of this problem) must write symbol $G$ on its tape if invoked on any of these four strings (as input.) If this is impossible, state it and explain why it is so.

**Answer:**

$$ac, ccaa, bac, gcca$$

**(h)** Explain how to construct an algorithm that solves the following problem:

INPUT: Turing machine $T$, input string $w$, and a tape symbol $X$;

OUTPUT: **yes** if the machine $T$ writes symbol $X$ if invoked on input $w$;
**no** otherwise.

If this algorithm does not exist, prove it.

**Answer:** This algorithm does not exist—it is undecidable whether a Turing Machine writes a given symbol on its tape.

**Problem 5** [ **12 points** ] Let $L$ be the language over the alphabet $\Sigma = \{a, b, c, g\}$ that contains exactly those strings which satisfy all of the following properties:

1. the length of the string is equal to $8n + 1$ for some natural number $n$;
2. the middle symbol is different from $a$;
3. the second (from the left) symbol is $g$;
4. the third (from the left) symbol is $a$;
5. the next-to-last (i.e, second from the right) symbol is $b$;
6. the last (i.e, first from the right) symbol is $c$.

Write a complete formal definition of a context-free grammar that generates the language $L$. If such a grammar does not exist, prove it.

INSTRUCTOR'S SOLUTION

**Answer:** $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c, g\}$, $V = \{S, A, Z\}$, and $P$ is:

$$\begin{aligned}
&S \rightarrow ZgaZ\,A\,ZZbc \\
&A \rightarrow ZZZZ\,A\,ZZZZ \mid b \mid c \mid g \\
&Z \rightarrow a \mid b \mid c \mid g
\end{aligned}$$

**Problem 6** [ **16 points** ] Let $\Sigma = \{a, b, c, g\}$, and let $L$ be the set of exactly those strings over $\Sigma$ whose form is:

$$g^m a^{2n} b^\ell g^{2\ell} c^n g^{2m+2k}$$

where $k, \ell, m, n \geq 0$ are natural numbers.

**(a)** Write a complete formal definition of a context-free grammar that generates $L$. If such a grammar does not exist, prove it.

**Answer:** $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c, g\}$, $V = \{S, A, B, D, E\}$, and $P$ is:

$$\begin{aligned}
&S \to AB \\
&A \to gAgg \mid D \\
&D \to aaDc \mid E \\
&E \to bEgg \mid \lambda \\
&B \to ggB \mid \lambda
\end{aligned}$$

INSTRUCTOR'S SOLUTION

**(b)** Draw a state-transition graph of a finite-state automaton that accepts $L$. If such an automaton does not exist, prove it.

**Answer:** This automaton does not exist, since $L$ is not regular. To prove this, we show that Pumping Lemma does not hold for $L$.

Observe that all element strings of $L$ satisfy the following characteristic property:

number of $a$'s is equal to twice the number of $c$'s.

Assume the opposite, that $L$ is regular. Let $\xi$ be the constant as in the Pumping Lemma for $L$. Let $n > \xi$; then the string $w = a^{2n}c^n$ belongs to $L$, as it is obtained from the template by setting $m = \ell = k = 0$.

In any "pumping" decomposition such that $a^{2n}c^n = uvx$, we have: $|uv| \leq \xi < n < 2n$. Hence, the "pumping" substring $v$ consists entirely of $a$'s, say $v = a^j$, where $j > 0$, since the "pumping" substring $v$ cannot be empty. Pump up once, obtaining the string:

$$w_1 = a^{2n+j}c^n$$

Since $2n + j > 2n$, string $w_1$ violates the stated characteristic property and thus $w_1 \notin L$, in violation of the Pumping Lemma.

**Problem 7** [ **16 points** ] Let $\Sigma = \{a, b, c, g\}$, and let $L$ be the set of exactly those strings over $\Sigma$ whose form is:

$$g^m a^\ell b^n g^{2\ell} c^n g^{2m+2k}$$

where $k, \ell, m, n \geq 0$ are natural numbers.

**(a)** Write a complete formal definition of a context-free grammar that generates $L$. If such a grammar does not exist, prove it.

**Answer:** Such a grammar does not exist, since $L$ is not context-free. To prove this, we show that Pumping Lemma does not hold for $L$.

Observe that every element string of $L$ satisfies the following characteristic property:

the number of $b$'s is equal to the number of $c$'s and number of $a$'s is equal to one half of the number of $g$'s between $b$'s and $c$'s.

Assume the opposite, that $L$ is context-free. Let $\eta$ be the constant as in the Pumping Lemma for $L$. Select $n$ and $\ell$ such that $n > \eta, \ell > \eta$ and consider a string $w = a^\ell b^n g^{2\ell} c^n$, which is obtained from the template for $L$ by setting $m = k = 0$. $w$ must pump, because its length is at least $5\eta > \eta$. In any pumping decomposition of $w_1$, the pumping window either is entirely within one of the four segments: $a^\ell$ or $b^n$ or $g^{2\ell}$ or $c^n$, or it spans two adjacent segments, since the pumping window is too short to extend into more than two segments, because its length is less than $\eta$, and thereby less than $n$ and less than $\ell$. Thus, if we pump up once, there will be at least one of the segments where pumping occurs and at least one that is not adjacent to it where the pumping does not occur. The number of occurrences of at least one of the letters will have increased, while the related number of occurrences of another letter will not have changed. The new string $w_1$ will violate the characteristic property and thus will not belong to $L$. Since $L$ violates the Pumping Lemma, $L$ is not context free.

INSTRUCTOR'S SOLUTION

**(b)** Draw a state-transition graph of a finite-state automaton that accepts $L$. If such an automaton does not exist, prove it.

**Answer:** This automaton does not exist, since $L$ is not regular. If $L$ was regular, then $L$ would be context free. However, $L$ is not context free, as is shown in the answer to part (a). Thus, $L$ cannot be regular.