MTH 222 Theory of Computation

Second Mid Semester Examination (Exercise set A)

Total marks: 25 | October 2002 | Time: $1 + \epsilon$ hours

Name: **Roll Number:**

---

1. Which of the following statements is/are true? (Give an explanation for each in at most two sentences.) **(2 × 5)**
(**Remark:** No credit will be given to a correct guess followed by an improper explanation.)

(a) If the fanout $\phi(G)$ of a CFG G is $\leqslant 2$, then $\mathcal{L}(G)$ may be infinite. **True**

Consider the CFG

$$G := (\{a, b\}, \{S, T\}, S, \{S \to Tb, T \to \epsilon \mid Ta\}).$$

Then

$$\mathcal{L}(G) = \{a^k b \mid k \in \mathbb{Z}_+\}$$

is infinite.

(b) $aabbaa \in \mathcal{L}(G)$, where $G := (\{a, b\}, \{S\}, S, \{S \to b \mid Sa \mid aS \mid SS\})$. **True**

Consider the leftmost derivation:

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaSa \Rightarrow aaSaa \Rightarrow aaSSaa \Rightarrow aabSaa \Rightarrow aabbaa.$$

(c) The CFG G of Part (b) is ambiguous. **True**

Consider the two different parse trees for the following two leftmost derivations of bab:

$$S \Rightarrow SS \Rightarrow bS \Rightarrow baS \Rightarrow bab$$
$$S \Rightarrow SS \Rightarrow SaS \Rightarrow baS \Rightarrow bab$$

(d) $\mathcal{L}(G)$ is the language of the regular expression $a^*bb^*a^*$, where G is the CFG of Part (b). **False**

$bab \in \mathcal{L}(G)$ (See Part (c)), whereas $bab \notin \mathcal{L}(a^*bb^*a^*)$.

(e) The union of infinitely many context-free languages may be non-context-free. **True**

Let $L := \{\alpha_1, \alpha_2, \alpha_3, \ldots\} = \bigcup_{n \in \mathbb{N}} \{\alpha_n\}$ be an (infinite) non-context-free language. Each $\{\alpha_n\}$ is finite and hence regular and hence context-free.

2. Let $\Sigma := \{a, b, c\}$ and $L := \{\alpha c \alpha^R c \alpha \mid \alpha \in \{a, b\}^*\}$.

(a) Show that L is not context-free. **(4)**

Solution Assume that L is context-free and let n be the constant for L prescribed by the stronger version of the pumping lemma. Consider $\alpha := a^n c a^n c a^n \in L$. The pumping lemma gives us the decomposition $\alpha = \beta_1\beta_2\beta_3\beta_4\beta_5$ with $|\beta_2\beta_4| \geqslant 1$ and $|\beta_2\beta_3\beta_4| \leqslant n$. Since $\alpha' := \beta_1\beta_3\beta_5 \in L$, $\beta_2\beta_4$ must not contain the symbol c, i.e., $\beta_2\beta_4$ consists only of a's. The condition $|\beta_2\beta_3\beta_4| \leqslant n$ implies that $\beta_2\beta_4$ can not stretch over all the three runs of a's in α. Therefore, α' lacks the defining property of the strings of L. This contradiction shows that L is not context-free. •

(b) Write L as the intersection of two context-free languages (over Σ). **(4)**

Solution Define

$$
\begin{aligned}
L_1 &:= \{\alpha c \alpha^R c \beta \mid \alpha, \beta \in \{a, b\}^*\}, \\
L_2 &:= \{\beta c \alpha^R c \alpha \mid \alpha, \beta \in \{a, b\}^*\}.
\end{aligned}
$$

Clearly $L = L_1 \cap L_2$. I will now show that L_1 is context-free. Consider the CFG $G := (\Sigma, \{S, U, V\}, S, R)$ for L_1, where the rules in R are:

$$
\begin{aligned}
S &\to UV \\
U &\to c \mid aUa \mid bUb \\
V &\to c \mid Va \mid Vb
\end{aligned}
$$

An analogous CFG defines L_2. •

3. Let $L := \{a^{3k+1}b^{5k-2} \mid k \geqslant 1\} \subseteq \{a,b\}^*$.

(a) Write a CFG G with $\mathcal{L}(G) = L$. **(3)**

Solution The trick is to substitute $k = l + 1$ and write L as

$$L = \{a^{4+3l}b^{5l+3} \mid l \geqslant 0\}.$$

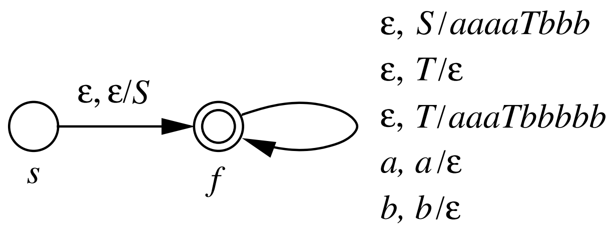
Now it is easy to write a CFG $G := (\{a,b\}, \{S,T\}, S, R)$ for L with the rules:

$$\begin{array}{lcl} S & \to & aaaaTbbb \\ T & \to & \epsilon \mid aaaTbbbbb \end{array}$$

Clearly $\mathcal{L}(T) = \{a^{3l}b^{5l} \mid l \geqslant 0\}$. The rest is obvious. •

(b) Design a PDA M with $\mathcal{L}(M) = L$. **(3)**

Solution A PDA can be designed for L naïvely, i.e., starting from the scratch. Now that we have a CFG for L, it is easier to use the CFG-to-PDA conversion procedure to construct the following PDA with two states:



•

(c) Is the PDA you designed in Part (b) a deterministic PDA? **(1)**

Solution Nope! When the PDA is in the state f and T is at the top of the stack, the PDA may decide to replace it by ϵ or by $aaaTbbbbb$ without consuming any symbol from the input. •

4. [Bonus problem] Let $\Sigma := \{a, b\}$. For $x \in \Sigma$ and $\alpha \in \Sigma^*$ define $\nu_x(\alpha) :=$ the number of occurrences of x in α. Design a PDA M with $\mathcal{L}(M) = \{\alpha \in \Sigma^* \mid \nu_b(\alpha)$ is an (integral) multiple of $\nu_a(\alpha)\}$. **(10)**

Solution Oops! A PDA can not be designed to accept the language in question, call it L, since L is not context-free at all. The intuitive reason why L is not context-free is that the machine will have to keep track of *both* the number of a's and the number of b's read. With a single stack this is impossible. Alternatively, the machine will have to prepare nondeterministically for every $k \in \mathbb{Z}_+$ to handle the case $\nu_b(\alpha) = k\nu_a(\alpha)$. Since there are infinitely many possibilities for k, a finite machine would not be adequate.

We need formal arguments to settle this issue. As usual we will appeal to the pumping lemma – the stronger version makes reasoning easier here.

Assume that L is context-free and let n be the pumping lemma constant for L. Choose an integer $m > n$ (For example, $m := n+1$ will do.) and consider any string $\alpha \in L$ having exactly m occurrences of a and exactly $m!$ (m-factorial) occurrences of b. The pumping lemma provides the decomposition $\alpha = \beta_1\beta_2\beta_3\beta_4\beta_5$ with the relevant properties. Suppose that $\beta_2\beta_4$ consists of exactly r occurrences of a and exactly s occurrences of b. Then $1 \leqslant r + s \leqslant n$, since $1 \leqslant |\beta_2\beta_4| \leqslant |\beta_2\beta_3\beta_4| \leqslant n$.

Case 1: $s = 0$.

In this case $\beta_2\beta_4$ consists only of a's. Then $|\beta_2\beta_4| = r \geqslant 1$ and so we can choose a k large enough, so that $m + kr > m!$. Since $\beta_1\beta_2^{k+1}\beta_3\beta_4^{k+1}\beta_5 \in L$, we have $(m + kr) \mid m!$, which is absurd.

Case 2: $s \geqslant 1$.

$\beta_1\beta_3\beta_5 \in L$ and so $(m - r) \mid (m! - s)$. Since $0 \leqslant r \leqslant n < m$, we have $m - r \in \{1, 2, \ldots, m\}$, i.e., $(m - r) \mid m!$. Therefore, $(m - r) \mid s$, i.e., $m - r \leqslant s$, i.e., $m \leqslant r + s \leqslant n < m$, again a contradiction.

Thus L is not context-free. •

(**Remark:** For integers u, v the phrase "v is an integral multiple of u" is abbreviated as $u \mid v$ to be read as "u divides v". Specifically, we say that $u \mid v$, if (and only if) there exists an integer w with $v = uw$.)

MTH 222 Theory of Computation

Second Mid Semester Examination (Exercise set B)

Total marks: 25 | October 2002 | Time: $1 + \epsilon$ hours

Name: **Roll Number:**

---

1. Which of the following statements is/are true? (Give an explanation for each in at most two sentences.) **(2×5)**
(**Remark:** No credit will be given to a correct guess followed by an improper explanation.)

(a) $aabbaa \in \mathcal{L}(G)$, where $G := (\{a, b\}, \{S\}, S, \{S \to \epsilon \mid Sb \mid aSa\})$. $\boxed{\text{True}}$

Consider the leftmost derivation:

$$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aaSbaa \Rightarrow aaSbbaa \Rightarrow aabbaa.$$

(b) $\mathcal{L}(G)$ is the language of the regular expression $a^*b^*a^*$, where G is the CFG of Part (a). $\boxed{\text{False}}$

Since $S \Rightarrow Sb \Rightarrow aSab \Rightarrow aSbab \Rightarrow abab$, we have $abab \in \mathcal{L}(G)$. But $abab \notin \mathcal{L}(a^*b^*a^*)$.

(c) The grammar of Part (a) is ambiguous. $\boxed{\text{False}}$

In the first step of a leftmost derivation of any $\alpha \in \mathcal{L}(G)$ a unique rule is applicable. That is, the rules $S \to \epsilon$, $S \to aSa$ and $S \to Sb$ are applicable respectively to the pairwise disjoint cases: $\alpha = \epsilon$, α ends with a and α ends with b.

(d) If $\mathcal{L}(G)$ is finite for a CFG G, then the fanout $\phi(G)$ of G is $\leqslant 2$. $\boxed{\text{False}}$

Consider the CFG

$$G := (\{a, b\}, \{S\}, S, \{S \to aba\}).$$

Then $\mathcal{L}(G) = \{aba\}$ is finite, whereas $\phi(G) = 3$.

(e) The intersection of two context-free languages is never context-free. $\boxed{\text{False}}$

The intersection of two regular languages is regular. Regular languages are context-free.
Alternatively, take $L_1 = L_2$ to be a CFL. Then $L_1 \cap L_2$ is evidently context-free.

2. Let $\Sigma := \{a, b, c\}$ and $L := \{\alpha a \alpha^R a \alpha \mid \alpha \in \{b, c\}^*\}$.

(a) Show that L is not context-free. **(4)**

Solution Assume that L is context-free and let n be the constant for L prescribed by the stronger version of the pumping lemma. Consider $\alpha := b^n a b^n a b^n \in L$. The pumping lemma gives us the decomposition $\alpha = \beta_1\beta_2\beta_3\beta_4\beta_5$ with $|\beta_2\beta_4| \geqslant 1$ and $|\beta_2\beta_3\beta_4| \leqslant n$. Since $\alpha' := \beta_1\beta_3\beta_5 \in L$, $\beta_2\beta_4$ must not contain the symbol a, i.e., $\beta_2\beta_4$ consists only of b's. The condition $|\beta_2\beta_3\beta_4| \leqslant n$ implies that $\beta_2\beta_4$ can not stretch over all the three runs of b's in α. Therefore, α' lacks the defining property of the strings of L. This contradiction shows that L is not context-free. •

(b) Write L as the intersection of two context-free languages (over Σ). **(4)**

Solution Define

$$\begin{aligned} L_1 &:= \{\alpha a \alpha^R a \beta \mid \alpha, \beta \in \{b, c\}^*\}, \\ L_2 &:= \{\beta a \alpha^R a \alpha \mid \alpha, \beta \in \{b, c\}^*\}. \end{aligned}$$

Clearly $L = L_1 \cap L_2$. I will now show that L_1 is context-free. Consider the CFG $G := (\Sigma, \{S, U, V\}, S, R)$ for L_1, where the rules in R are:

$$\begin{aligned} S &\rightarrow UV \\ U &\rightarrow a \mid bUb \mid cUc \\ V &\rightarrow a \mid Vb \mid Vc \end{aligned}$$

An analogous CFG defines L_2. •

3. Let $L := \{a^{5k+1}b^{3k-2} \mid k \geqslant 1\} \subseteq \{a,b\}^*$.

(a) Write a CFG G with $\mathcal{L}(G) = L$. **(3)**

Solution The trick is to substitute $k = l + 1$ and write L as

$$L = \{a^{6+5l}b^{3l+1} \mid l \geqslant 0\}.$$

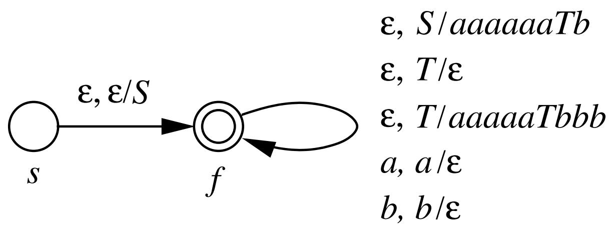
Now it is easy to write a CFG $G := (\{a,b\}, \{S,T\}, S, R)$ for L with the rules:

$$\begin{array}{lcl} S & \to & aaaaaaTb \\ T & \to & \epsilon \mid aaaaaTbbb \end{array}$$

Clearly $\mathcal{L}(T) = \{a^{5l}b^{3l} \mid l \geqslant 0\}$. The rest is obvious. •

(b) Design a PDA M with $\mathcal{L}(M) = L$. **(3)**

Solution A PDA can be designed for L naïvely, i.e., starting from the scratch. Now that we have a CFG for L, it is easier to use the CFG-to-PDA conversion procedure to construct the following PDA with two states:



•

(c) Is the PDA you designed in Part (b) a deterministic PDA? **(1)**

Solution Nope! When the PDA is in the state f and T is at the top of the stack, the PDA may decide to replace it by ϵ or by $aaaaaTbbb$ without consuming any symbol from the input. •

4. [Bonus problem] Let $\Sigma := \{a, b\}$. For $x \in \Sigma$ and $\alpha \in \Sigma^*$ define $\nu_x(\alpha) :=$ the number of occurrences of x in α. Design a PDA M with $\mathcal{L}(M) = \{\alpha \in \Sigma^* \mid \nu_a(\alpha)$ is an (integral) multiple of $\nu_b(\alpha)\}$. **(10)**

Solution Oops! A PDA can not be designed to accept the language in question, call it L, since L is not context-free at all. The intuitive reason why L is not context-free is that the machine will have to keep track of *both* the number of a's and the number of b's read. With a single stack this is impossible. Alternatively, the machine will have to prepare nondeterministically for every $k \in \mathbb{Z}_+$ to handle the case $\nu_a(\alpha) = k\nu_b(\alpha)$. Since there are infinitely many possibilities for k, a finite machine would not be adequate.

We need formal arguments to settle this issue. As usual we will appeal to the pumping lemma – the stronger version makes reasoning easier here.

Assume that L is context-free and let n be the pumping lemma constant for L. Choose an integer $m > n$ (For example, $m := n+1$ will do.) and consider any string $\alpha \in L$ having exactly m occurrences of b and exactly $m!$ (m-factorial) occurrences of a. The pumping lemma provides the decomposition $\alpha = \beta_1\beta_2\beta_3\beta_4\beta_5$ with the relevant properties. Suppose that $\beta_2\beta_4$ consists of exactly r occurrences of b and exactly s occurrences of a. Then $1 \leqslant r + s \leqslant n$, since $1 \leqslant |\beta_2\beta_4| \leqslant |\beta_2\beta_3\beta_4| \leqslant n$.

Case 1: $s = 0$.

In this case $\beta_2\beta_4$ consists only of b's. Then $|\beta_2\beta_4| = r \geqslant 1$ and so we can choose a k large enough, so that $m + kr > m!$. Since $\beta_1\beta_2^{k+1}\beta_3\beta_4^{k+1}\beta_5 \in L$, we have $(m + kr) \mid m!$, which is absurd.

Case 2: $s \geqslant 1$.

$\beta_1\beta_3\beta_5 \in L$ and so $(m - r) \mid (m! - s)$. Since $0 \leqslant r \leqslant n < m$, we have $m - r \in \{1, 2, \ldots, m\}$, i.e., $(m - r) \mid m!$. Therefore, $(m - r) \mid s$, i.e., $m - r \leqslant s$, i.e., $m \leqslant r + s \leqslant n < m$, again a contradiction.

Thus L is not context-free. •

(**Remark:** For integers u, v the phrase "v is an integral multiple of u" is abbreviated as $u \mid v$ to be read as "u divides v". Specifically, we say that $u \mid v$, if (and only if) there exists an integer w with $v = uw$.)