Name: __________________________

ID #: __________________________

This is a Close Book examination. Only an A4 cheating sheet belonging to you is acceptable. You can write your answers in English or Chinese.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Maximum</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Problem 2</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Problem 3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Problem 4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Problem 5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Problem 6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Problem 7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Problem 8</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
1. (25 pts) Mark by T(=True) or F(=False) each of the following statements. You don’t need to prove it.

(1) A regular language can not be inherently ambiguous.
(2) Every simple grammar is unambiguous.
(3) A linear grammar is regular.
(4) Suppose $L_1 = L(a^*baa^*)$ and $L_2 = L(ab^*)$. Then $L_1/L_2 = L(a^*b^*a^*)$.
(5) If $L_1$ and $L_2$ are two regular languages then the language $(L_1 \cup L_2) \cap (L_1 \cap \overline{L_2})$ is regular.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>T</td>
</tr>
<tr>
<td>(2)</td>
<td>T</td>
</tr>
<tr>
<td>(3)</td>
<td>F</td>
</tr>
<tr>
<td>(4)</td>
<td>F</td>
</tr>
<tr>
<td>(5)</td>
<td>T</td>
</tr>
</tbody>
</table>

2. (15 pts) Consider the following generalized transition graph.

(1) (10 pts) Find an equivalent generalized transition graph with only two states.
(2) (5 pts) What is the language accepted by this graph?

Sol:

(1) $r = a^*(a + b)ab(ab + bb + aa^*(a + b)ab)^*$.

3. (15 pts) Prove that the following languages are not regular where $\Sigma = \{a, b\}$.

(a) $L = \{a^n b^l a^k : k \neq n + l\}$, $k, n, l$ are integers.
(b) This can be done by first solving that $\overline{L} = \{a^n b^m a^k : k = n + l \}$ is not regular as in the previous problem. Then $L$ is not regular; otherwise $\overline{L}$ is regular. The other way to show this is to apply the pumping lemma directly.

Given $m$. We pick $w = a^m b^m a^{2m+1}$ with $|w| \geq m$. Consider $w = xyz$ where $|xy| \leq m$ and $|y| \geq 1$. Then, $y = a^k$, where $1 \leq k \leq m$. Now, we choose $y = a$. Consider the pumping strings

$$w_i = a^{m+(i-1)k} b^m a^{2m+1}.$$ 

When $i = 2$, $w_2 = a^{m+1} b^m a^{2m+1} \notin L$. According to the Pumping Lemma, $L$ is not regular.

(b) If $L = \{w: n_a(w) \neq n_b(w)\}$ is regular, then $\overline{L} = \{w: n_a(w) = n_b(w)\}$ is regular. Note that $L(a^*b^*) \cap \overline{L} = L(a^n b^n)$. If we want to prove $L = \{w: n_a(w) \neq n_b(w)\}$ is not regular, we only need to prove $L(a^n b^n)$ is not regular. This will imply that $L = \{w: n_a(w) \neq n_b(w)\}$ is not regular.

Given $m$. We pick $w = a^m b^m$ with $|w| \geq m$. Let $w = xyz$ where $|xy| \leq m$ and $|y| \geq 1$. Then, $y = a^k$, where $1 \leq k \leq m$. Now, we consider the pumping strings

$$w_i = a^{m+(i-1)k} b^m.$$ 

When $i = 0$, $w_0 = a^{m-k} b^m \notin L(a^n b^n)$. According to the Pumping Lemma, $L(a^n b^n)$ is not regular. Hence, $L = \{w: n_a(w) \neq n_b(w)\}$ is not regular.

(c) Given $m$. We pick $w = a^m b^m a^m b^m a^m b^m a^m$ with $|w| \geq m$. Let $w = xyz$ where $|xy| \leq m$ and $|y| \geq 1$. Then, $y = a^k$, where $1 \leq k \leq m$. Now, we consider the pumping strings

$$w_i = a^{m+(i-1)k} b^m a^m b^m a^m b^m a^m.$$ 

When $i = 0$, $w_0 = a^{m-1} b^m a^m b^m a^m \notin L$. According to the Pumping Lemma, $L$ is not regular.

4. (10 pts)

(a) (5 pts) Give the definition of simple grammar.

(b) (5 pts) Please find a context-free grammar for the language $L = \{a^n b^n : n \geq 1\}$.

Sol:

(a) A context-free grammar $G = (V, T, S, P)$ is said to be a simple grammar or $s$-grammar if all its productions are of the form

$$A \rightarrow ax,$$

where $A \in V, a \in T, x \in V^*$, and any pair $(A, a)$ occurs at most once in $P$. 

3
5. (5 pts) Please show that the following grammar is ambiguous.

\[ S \rightarrow AB, \]
\[ A \rightarrow aaA|\lambda, \]
\[ B \rightarrow Bb|\lambda \]

**Sol:** There are two derivations for \( w = aab \).

\[ S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaB \Rightarrow aab, \]
\[ S \Rightarrow AB \Rightarrow ABb \Rightarrow aaABb \Rightarrow aaAb \Rightarrow aab. \]

6. (10 pts) Eliminate useless productions from

\[ S \rightarrow a|aA|B|C, \]
\[ A \rightarrow aB|\lambda, \]
\[ B \rightarrow Aa, \]
\[ C \rightarrow cCD, \]
\[ D \rightarrow ddd. \]

**Sol:**

\[ S \rightarrow a|aA|B, \]
\[ A \rightarrow aB|\lambda, \]
\[ B \rightarrow Aa. \]

7. (10 pts) Please convert the following grammar to Chomsky Normal Form.

\[ S \rightarrow AB|B|aB, \]
\[ A \rightarrow aab|\lambda, \]
\[ B \rightarrow bbA. \]

**Note:** You need to remove the \( \lambda \)-production and unit-production from the production rules first.

**Sol:**

1° We first remove the \( \lambda \)-production. This yields

\[ S \rightarrow AB|B|aB, \]
\[ A \rightarrow aab, \]
\[ B \rightarrow bbA|bb. \]
This introduces a unit-production, that is not acceptable in our approach. Eliminating
this unit-production is easy and we can derive

\[ S \rightarrow AB|bbA|aB|bb, \]
\[ A \rightarrow aab, \]
\[ B \rightarrow bbA|bb. \]

2° We are now in the place for converting the grammar.

For \( S \rightarrow bbA \), we replace it by

\[ S \rightarrow C_1C_1A, \]
\[ C_1 \rightarrow b. \]

For \( S \rightarrow aB \), we replace it by

\[ S \rightarrow C_2B, \]
\[ C_2 \rightarrow a. \]

For \( S \rightarrow bb \), we replace it by

\[ S \rightarrow C_1C_1. \]

For \( A \rightarrow aab \), we replace it by

\[ A \rightarrow C_2C_2C_1. \]

For \( B \rightarrow bbA \), we replace it by

\[ B \rightarrow C_1C_1A. \]

For \( B \rightarrow bb \), we replace it by

\[ B \rightarrow C_1C_1. \]

3° By introducing new variables \( D_1, D_2, \) and \( D_3 \) for the productions having more than two
variables on the right-hand side, the resulting grammar in Chomsky Normal Form is

\[ S \rightarrow AB|C_1D_1|C_2B|C_1C_1, \]
\[ A \rightarrow C_2D_2, \]
\[ B \rightarrow C_1D_3|C_1C_1, \]
\[ C_1 \rightarrow b, \]
\[ C_2 \rightarrow a, \]
\[ D_1 \rightarrow C_1A, \]
\[ D_2 \rightarrow C_2C_1, \]
\[ D_3 \rightarrow C_1A. \]
8. (10 pts) Consider the language $L$ generated by the following grammar

$$
S \rightarrow AB,
A \rightarrow BB|a,
B \rightarrow AB|b.
$$

(a) (5 pts) Using the CYK algorithm to determine whether the string $w = aab$ is in $L$.

(b) (5 pts) According to the result derived in (a), please find a parsing for the string. Your approach must refer to the CYK algorithm. In other words, you can use the backtracking approach to find the parsing.

**Sol:**

(a) 

$$
V_{11} = \{A\} \quad V_{12} = \emptyset \quad V_{13} = \{S, B\}
$$

$$
V_{22} = \{A\} \quad V_{23} = \{S, B\}
$$

$$
V_{33} = \{S, B\}
$$

Since $S$ is in $V_{13}$, we can conclude $w$ is in $L$.

(b) Using backtracking strategy, we can mark at each step about where the result comes as follow.

For $V_{13} = \{S, B\}$, it is derived from $V_{11}$ and $V_{23}$ with $S \rightarrow AB$ or $B \rightarrow AB$. Let’s start with $S \rightarrow AB$. So, we can get $S \Rightarrow aB$. For $V_{23} = \{S, B\}$, it is derived by $V_{22}$ and $V_{33}$ with $B \rightarrow AB$ or $S \rightarrow AB$. So, $S \Rightarrow aB \Rightarrow aAB$ using $B \rightarrow AB$. Since $V_{22}$ and $V_{33}$ are derived with $A \rightarrow a$ and $B \rightarrow b$, respectively. Therefore, we can get $S \Rightarrow^* aAB \Rightarrow aaB \Rightarrow aab$. 