CMPT 379 - Summer 2016 - Sample Midterm

(1) (6pts) If the following descriptions define a regular language then write the corresponding regular expression. Otherwise indicate that the language is not regular. Provide regular expressions that are explanatory and compact. Use the usual regular expression operators: $\cdot$ `|` `(` `)` $*$, where the concatenation operator $\cdot$ can be omitted. You can also use the operator ? which stands for zero or one occurrence of the previous symbol or group.

a. All strings of 0's and 1's that represent binary numbers that are equal to the decimal number 6. Leading zeroes must be allowed.

Answer: `0*110`, cut half the points for missing initial `0*`

b. All strings of 0's and 1's that represent binary numbers that are powers of 2. Leading zeroes must be allowed.

Answer: `0*10*`, cut half the points for missing initial `0*`

c. All strings of 0's and 1's that represent Binary Coded Decimal (BCD) numbers (include the empty string). A BCD number is a decimal number where each decimal digit is encoded using a 4-bit representation of its binary value. For example, the BCD number of 2509 is 0010010100001001

Answer: `((0(0|1)(0|1)(0|1))|((100)(0|1)))*`
Following solution is also ok:
`((0000)|(0001)|(0010)|(0011)|(0100)|(0101)|(0110)|(0111)|(1000)|(1001))*`

(2) (8pts) You are given the following ordered list of token definitions:

TOKEN_A	cda^*
TOKEN_B	c^*a^*c
TOKEN_C	c^*b

Provide the tokenized output for the following input strings using the greedy longest match lexical analysis method. Provide the list of tokens and the lexeme values.

a. *cdaaab*

Answer: TOKEN_A (cdaaa), TOKEN_C (b)

b. *cdccc*

Answer: TOKEN_A (cd), TOKEN_B (ccc)

c. *ccc*

Answer: TOKEN_B (ccc)

d. *cdccd*

Answer: TOKEN_A (cd), TOKEN_B (cc), ERROR (illegal token)

(3) (8pts) Consider the following grammar G:

$$S' \rightarrow S$$
$$S \rightarrow aSa \mid bSb \mid aa \mid bb$$

a. Is the CFG G an LL(1) grammar? Provide a reason for your answer.

Answer: No. FIRST(aSa) intersected with FIRST(aa) is non-empty.

b. Consider a slightly modified version of grammar G. Let's call it G':

$$S \rightarrow aSa \mid bSb \mid \epsilon$$

Does this modified grammar G' generate the same language as the original grammar G? Provide a reason for your answer.

Answer: No. It now includes ϵ in the language.

c. Is G' an LL(1) grammar? Provide a reason for your answer.

Answer: No, it is still not an LL(1) grammar because the intersection of FIRST(aSa) and FOLLOW(S) = $\{a, b, \$\}$ is non-empty.

d. Is the CFG G' an SLR(1) grammar? Provide a reason for your answer.

Answer: No. In the closure for $S' \rightarrow \bullet S$ we get two shift-reduce conflicts between $S \rightarrow \epsilon\bullet$ and $S \rightarrow \bullet aSa$ and $S \rightarrow \bullet bSb$. Furthermore, FOLLOW($S$) = $\{a, b\}$ so the conflicts cannot be resolved with one symbol of lookahead in an SLR(1) parser.

(4) (8pts) Consider the family of CFGs G_k with S as the start symbol and k is some arbitrary non-zero positive integer such that $G_1, G_2, G_3, \ldots$ are individual CFGs with the rules:

$$S \rightarrow A\ B$$
$$B \rightarrow C\ A\ A$$
$$C \rightarrow c$$
$$A \rightarrow a_i \quad \text{defines } i \text{ rules, where } i \in [1, k]$$

For example, in G_3 the rules with A as left-hand side are: $A \rightarrow a_1 \mid a_2 \mid a_3$ with three terminal symbols.

a. Provide the number of terminal symbols in a grammar G_k, the number of elements in FIRST(S), and the number of elements in FOLLOW(C).

Answer: $k + 1$, k, and k respectively.

b. If the string $a_3ca_1a_2$ is accepted by grammar G_4 then provide the leftmost derivation that derives it.

Answer: $S \Rightarrow A\ B \Rightarrow a_3\ B \Rightarrow a_3\ C\ A\ A \Rightarrow a_3\ c\ A\ A \Rightarrow a_3\ c\ a_1\ A \Rightarrow a_3\ c\ a_1\ a_2$

c. Can any G_k grammar have a leftmost derivation of the form: $X \Rightarrow^* \alpha X \beta$, where X is any non-terminal in the grammar G_k and α, β is any sequence of terminals or non-terminals? Briefly explain why.

Answer: No. the grammar is not recursive so $X \Rightarrow^*_{lm} \alpha X \beta$ for any X cannot occur.

d. Provide the total number of leftmost derivations possible for a grammar G_k.

Answer: k^3