Ministry of Higher Education and Scientific Research

Salahaddin University / Erbil ***Subject: Compilers***
College of Engineering ***Time: 90 Minutes***
Dept. of Software Engineering ***Mid Year Exam 2011-2012*** ***Lecturer: Amanj Sherwany***

The highest obtainable mark is 17, the minimum passing mark is 8.5

Q1: (4 points)

For the the following regular expression:

$$a(bc)^*a \mid ac(c)^*$$

1. Construct a Deterministic Finite Automaton (DFA) for the above regular expression. **(3 points)**
2. Show each step of the lexer on the string *aaaccabca*. Be sure to show the values of the important internal variables of the recognizer. There will be repeated calls to the lexer to get all tokens from the string. **(1 point)**

* * *

Q2: (4 points)

Consider the following context-free grammar G_0:

$$E \rightarrow E, E$$
$$E \rightarrow E = E$$
$$E \rightarrow E[E]$$
$$E \rightarrow (E)$$
$$E \rightarrow \mathrm{x}$$

1. The grammar G_0 is ambiguous. Explain what this concept means, and give an example which shows that $G_0$is ambiguous. **(2 point)**
2. Assume that ($E[E]$) has the highest priority, (=) has the second-highest priority, and (,) has the lowest priority. Rewrite G_0 to an equivalent G_1 which expresses these properties. **(1 points)**
3. Assume that (=) is right-associative and (,) is left-associative. Rewrite G_1 to an equivalent grammar G_2 which expresses these properties. **(1 points)**

* * *

Q3: (6 points)

Consider the following augmented grammar, where S' is the start symbol and \$ is the special end-of-input symbol.

P_0	$S' \rightarrow S\$$
P_1	$S \rightarrow A\ B$
P_2	$S \rightarrow \mathrm{a\ c}$
P_3	$S \rightarrow \mathrm{x}\ A\ \mathrm{c}$
P_4	$A \rightarrow \mathrm{a}$
P_5	$B \rightarrow \mathrm{b}$
P_6	$B \rightarrow -$

1. Show that this grammar is not *SLR* (or, show that the *SLR* construction will fail). **(1 points)**
2. Rewrite the grammar to an equivalent grammar that is *SLR*. (Hint: clone or eliminate the non-terminal *A*.) Construct an *SLR* parsing table for the new grammar, including the intermediate *LR(0)* automation with states and transitions, and the *FIRST* and *FOLLOW* sets for the non-terminals. **(4 points)**
3. Show how an *LR* parser step by step (including changes in the stack and remaining tokens) parses the string *xac* using your *SLR* table. **(1 point)**

* * *

Q 4: (3 points)

Knowing that C-like languages use static scoping:

1. You are writing a C-compiler for a device that has a *2.4 GHz i7* CPU and only *256 MB* of RAM. Describe an *efficient* way of processing nested scopes for your compiler. Motivate your answer. **(2 points)**
2. Identify the *scopes* and the *variable* declarations for each scope of the following C program. **(1 point)**

```
int x;
int foo(int y)
{
     int z;
     if (y != 0) {
          double x = 3.14;
          z = bar(x);
     } else
          z = x;
     return z;
}

int bar(void)
{
     int x = 2;
     return x;
}
```

Good Luck