Sample Questions for Final Exam

1. The alphabet for the following regular expressions includes exactly the letters a, b, c. The parentheses indicate grouping, * indicates Kleene’s star. Consider the regular languages generated by the expressions

\[
L_1 = a^*b^* (a|c) (a^*b)^*
\]

\[
L_2 = (a|b)^* c (a|b)^*
\]

Give a string in the language of \(L_1\) but not in the language of \(L_2\)

Give a string in the language of \(L_2\) but not in the language of \(L_1\)

2. You are given a context free grammar for a new programming language and asked whether you can build a predictive parser for the language.

Explain in 2-3 sentences what is a predictive parser.

Explain briefly what you need to do in order to find out whether one can build a predictive parser for the given grammar.

3. (1) Explain what garbage collection means in the context of computer programming. (2) List one advantage and one disadvantage of using garbage collection.

4. Briefly describe one method for implementing garbage collection. (We discussed more than one such method in class. Pick one and explain its details.)

5. Recall that in C++ we can choose between “row-major” and “row-pointers” memory allocations for (two dimensional) array.

Briefly explain the difference between the two.

Give a code fragment that declares, allocates, and initializes a two dimensional array using row-major allocation.

Give a code fragment that declares, allocates, and initializes a two dimensional array using row-pointers allocation.

6. Study the following ML code and (1) give the values of \(a_1, b_1, a_2, b_2, a_3, b_3\), (2) explain your answer (do the values change? why?)

```ml
val x = 5;
fun f1 y = x+y;
fun f2 z = z * f1 2;
val a1 = f1 5;
val b1 = f2 5;
val x = 10;
val a2 = f1 5;
val b2 = f2 5;
fun f1 y = x-y;
val a3 = f1 5;
val b3 = f2 5;
```
7. Consult the following Prolog code:

\[
\begin{align*}
e(1,2). \\
e(3,1). \\
e(2,1). \\
e(1,1). \\
e(2,3). \\
e(1,3). \\
triangle(A,B,C) :- e(A,B), e(B,C), e(C,A).
\end{align*}
\]

(1) Draw the search tree (as we did in class) for running the query \texttt{triangle(X,Y,Z)} until the third output is produced. Make sure to annotate the tree with binding for variables. In addition, mark potential backtracking points in the tree.

(2) To clarify your answer from above, list the first 3 answers produced by the system on the query \texttt{triangle(X,Y,Z)}. For each answer give the binding for \texttt{X, Y, Z}.

8. Study the following code and (1) give the type of the function \texttt{f} (2) briefly explain how you inferred the type (3) give the value of \texttt{a} and explain how you computed it. (The function \texttt{foldl} is included for your convenience - it is identical to the one covered in class).

\[
\begin{align*}
\text{fun foldl f e [] = e} \\
| \quad \text{foldl f e (h::t) =} \\
| \quad \quad \quad \text{foldl f (f(h,e)) t;}
\end{align*}
\]

\[
\begin{align*}
\text{fun f L = foldl (fn(x,y) => 10*y+x) 0 L;}
\end{align*}
\]

\[
\begin{align*}
\text{val a = f [1,2,3];}
\end{align*}
\]

9. Write a function in ML that takes two sorted integer lists as input and produces a sorted merged list as illustrated by the following examples. The type of the function is:

\[
\begin{align*}
\text{val merge = fn : int list * int list -> int list}
\end{align*}
\]

and when running the function with:

\[
\begin{align*}
\text{val a = merge([1,3,5],[2,3,6]);}
\text{val b = merge([10],[1,3,9]);}
\end{align*}
\]

we get

\[
\begin{align*}
\text{val a = [1,2,3,5,6] : int list}
\text{val b = [1,3,9,10] : int list}
\end{align*}
\]
10. Write a Prolog procedure \texttt{length2sub(L,Out)} whose input \( L \) is a list and whose output \( \text{Out} \) is a subset of \( L \) with exactly two elements. Backtracking using ; should produce all such subsets as illustrated by the following examples.

You may not assume or use any list library functions. If you need such a function you should give the code for it as well.

\[
\begin{align*}
?-& \text{length2sub([a,b,c,a],Out).} \\
& \text{Out = [a, b]} ; \\
& \text{Out = [a, c]} ; \\
& \text{Out = [a, a]} ; \\
& \text{Out = [b, c]} ; \\
& \text{Out = [b, a]} ; \\
& \text{Out = [c, a]} ; \\
& \text{No}
\end{align*}
\]

\[
\begin{align*}
?-& \text{length2sub([1,2],Out).} \\
& \text{Out = [1, 2]} ; \\
& \text{No}
\end{align*}
\]

\[
\begin{align*}
?-& \text{length2sub([1],Out).} \\
& \text{No}
\end{align*}
\]