Principles of Programming Languages - Homework II

K. Subramani
LDCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Instructions

1. The homework assignment is to turned in by $11:00$ am. in class on April 7.
2. Each question is worth 3 points.
3. Attempt as many problems as you can. You will be given partial credit, as per the policy discussed in class.

2 Problems

1. Consider the following **C** fragment. Categorize the three assignments as legal or illegal, providing a justification for your answer.

```
{
   int x, *y, z[3];

(1)    &x = (int *)malloc (sizeof(int));
(2)    z= (int *)malloc(sizeof(int)*3);
(3)    *y = 3;

}
```

2. Consider the following **C** declarations:

```
{
  int x[10];
  int y[5];

}
```

   Are x and y type equivalent in **C**? How is the expression $(x == y)$ handled by the **C** compiler?

3. Assume that **C** uses strict type-checking, i.e., there is no type conversion without explicit casting. Further, assume that **C** has a `bool` data type. Use the Hindley-Milner type checking algorithm to derive the most general type for the following **C** function:

```
fact(n)
{
  if (n == 0)
```

```
    return 1;
  else
    return n*fact(n-1);
}
```

4. Consider a variant of the **C** language, called **C**$*$ in which only the **do - while** loop is specified as part of the syntax. Show how you would capture the semantics of **while** and **for** statements, using the **do - while** construct.

5. (i) Briefly explain the difference(s) between the normal and applicative orders of evaluation.

   (ii) Assume that **C** does not use short-circuiting in using the **and** operator. Professor Kurtowski attempts to remedy this situation by writing the following function for the **and** operator:

```
int and( int a, int b)
 {
   return a ? b : 0;
 }
```

   Will the professor's technique work, given the semantics of the **if**-expression in **C**? Justify your answer.