2009 Paper 5 Question 2

Compiler Construction

Consider an ML-like language in which the set of values includes functions and these functions may have nested definitions.

(a) Explain how a function value may be represented at run-time, both in a syntax-tree interpreter and in compiled code. What is the importance of the word "nested" above? [3 marks]

(b) Give an example program that gives different results in static and dynamic scoping. In each case, explain how an interpreter or compiled code might perform function application. [3 marks]

(c) Considering only static scoping from now on, explain what restrictions are necessary to implement function values without using an auxiliary heap. Explain the notions of static and dynamic chains, giving an example of a situation in which they differ (i.e. reference to the wrong one would access the wrong variable). Illustrate how updating (by assignment to) a free variable can be implemented. [5 marks]

(d) Give an alternative heap-based implementation for function values when the restrictions in part (c) do not hold. Explain how free variables (and particularly their update) can now be implemented. [4 marks]

(e) Java normally holds local variables in stack frames and instance variables on the heap. Consider a Java-like language with nested classes, and a possible program of the form:

```
class C {
  int f(int x)
   { class D { int addx(int y) { return x+y; }
               void updtx(int y) { x = y; }
             }
     ...
   }
};
```

Explain a possible implementation of such nested classes based on your answers above; also give and justify a restriction on methods like `updtx` that eases the cost of implementation. [5 marks]