1998 Paper 5 Question 6

Compiler Construction

Explain how a parse-tree representation of a program may be converted into a stack-based intermediate language giving sketches of code to translate expressions, assignments and the `if-then-else` command; you should also explain how occurrences of a variable in an expression or assignment are translated.

The program may be assumed to conform to the following syntax:

```
E  ->  n | x | E + E | f(E,E)
D  ->  let f(x,x) = {Dseq; Cseq; E} | let x = E
C  ->  x := E; | if E then C else C
Cseq -> C | C Cseq
Dseq -> D | D Dseq
```

with start symbol `Dseq`. Here `n` corresponds to integer constants, `x` corresponds to identifiers used as variable names and `f` corresponds to identifiers used as function names (you may assume these are disjoint). The function declaration construct has the effect of defining a function which, when called, makes declarations, performs commands and then returns the result of its expression; note that therefore functions may be defined within functions, but the above restriction on identifiers means that they cannot be returned as results. [20 marks]