Compiler Construction WS07/08

Exercise Sheet 7

1 Tree Grammars 5+5 Points

Given the following tree grammar:

\[
\begin{align*}
S & \rightarrow f(g(A, B), A)|f(C, h(B)) \\
A & \rightarrow h(a)|f(g(A, C))|h(g(A, A)) \\
B & \rightarrow g(h(D), A)|b \\
C & \rightarrow g(h(a), b) \\
D & \rightarrow h(g(A, B))|a
\end{align*}
\]

provide two different derivation trees for the input tree

\[f(g(h(a), b), h(g(h(a), h(a))))\]

2 Tree Automata 8+2 Points

a) Write a bottom-up deterministic finite tree automaton which recognizes exactly the trees in \(T(\Sigma)\) for \(\Sigma = \{true, false, and, or, xor, not\}\), where \(\rho(true) = \rho(false) = 0, \rho(not) = 1\) and \(\rho(and) = \rho(or) = \rho(xor) = 2\), which would evaluate to true given the usual rules of logic. Argue for the correctness of your solution.

b) Can the same language also be recognized by a top-down deterministic finite tree automaton? Provide reasonable arguments.
3 Code Generation 6+4 Points

Given a simple machine with the following instructions:

- \( R := M[V_1] \text{ op } M[V_2] \)
- \( R_1 := M[V] \text{ op } R_2 \)
- \( R_1 := R_2 \text{ op } M[V] \)
- \( R_1 := R_2 \text{ op } R_3 \)
- \( R := M[V] \)
- \( M[V] := R \)

a) Create an algorithm that takes a syntax tree of an arithmetic expression as input, labels the nodes of the tree with the corresponding register needs and generates code with a minimal number of needed registers. The result of the arithmetic expression has to be stored again in a register. You can assume that the machine has at least two registers.

b) Generate code for the following syntax trees and label the nodes with the corresponding register needs using your algorithm.

1) 

```
+  
/  
/   
a  b  a  
c   d
```

2) 

```
+  
/  
/   
/   
/   
e  d  a  b  e  f  c  d
```