Compiler Theory CS447b: Quiz 6. **UWO, March 08, 2004.**

Student ID number:

Guidelines. The quiz is closed book and all notes are forbidden. The duration of the quiz is 30 minutes. Exercice 1 is worth 4 points whereas each of the other two is worth 8 points. All answers should be written in the *answer boxes*. No justifications for the answers are needed **unless** explicitly required. You are expected to do this exam on your own without assistance from anyone else in the class. If possible, please avoid pencils and use pens with dark ink. Thank you.

Exercise 1. We consider the following syntax-directed definition

Production	Semantic Rule
$A \longmapsto L\,M$	$L.i := f_1(A.i)$
	$M.i := f_2(L.s)$
	$A.s := f_3(M.s)$
$A \longmapsto Q\,R$	$R.i := f_4(A.i)$
	$Q.i := f_5(R.s)$
	$A.i := f_6(Q.s)$

If this is an L-attributed grammar, then write a translation scheme for it, otherwise explain why this is not an L-attributed grammar.

Answer 1

This is not an L-attributed. Indeed, the action $Q.i := f_5(R.s)$ defines an attribute of Q and thus must be defined before visiting the node corresponding to Q. But, this action also requures the knowledge of a synthetized attribute of R which cannot be known before visiting the node corresponding to R. Since Q is visited before R, we are led to a contradiction with the properties of L-attributed grammars

Exercise 2. Some languages such as PL1 permit a list of names to be given a list of attributes. The grammar G below abstracts the problem.

$$\begin{array}{rcl}
D & \longmapsto & namelist\ \ attriblist; \\
namelist & \longmapsto & \mathbf{id}, namelist \\
namelist & \longmapsto & \mathbf{id} \\
attriblist & \longmapsto & A\ \ attriblist \\
attriblist & \longmapsto & A \\
A & \longmapsto & \mathbf{long} \\
A & \longmapsto & \mathbf{double} \\
A & \longmapsto & \mathbf{single} \\
A & \longmapsto & \mathbf{integer} \\
A & \longmapsto & \mathbf{float}
\end{array}$$

where D, $namelist$, $attriblist$ and A are nonterminals and where **id**, **long**, **double**, **single**, **integer**, **float** and **;** are terminals. As usual **id** is the *identifier* token. For instance each of the following lines is a declaration generated by G.

```
i, j single integer;
u, v, w double float;
```

Give a syntax-directed definition for declarations generated by G. Here are some hints.

- You may simply write N and T instead of $namelist$ and $attriblist$ respectively.
- The code in the actions does **not** need to be C code but may be any pseudocode that is convenient to you.
- You may consider that **id** has an attribute $\mathbf{id}.name$ (as usual) giving the lexical value of **id**.
- Moreover you may assume that every variable in the symbol table may be associated with a list of attributes by an instruction like $\mathsf{enter}(\mathbf{id}.name, T.list)$
- The instruction $T.list := \mathsf{makelist}(A.attrib)$ will create the list consisting of the single item $A.attrib$.
- The instruction $T.list := \mathsf{cons}(A.attrib, T.list)$ will add the element $A.attrib$ at the beginning of the list $T.list$.

Answer 2

D	$\longmapsto$	N T **;**	{ **for id in** $N.list$ enter(**id**.$name$, $T.list$) }
N	$\longmapsto$	**id**, N	{ $N.list$:= cons(**id**.$name$, $N.list$) }
N	$\longmapsto$	**id**	{ $N.list$:= makelist(**id**.$name$) }
T	$\longmapsto$	A T	{ $T.list$:= cons($A.attrib$, $T.list$) }
T	$\longmapsto$	A	{ $T.list$:= makelist($A.attrib$) }
A	$\longmapsto$	**long**	{ $A.attrib$:= **long** }
A	$\longmapsto$	**double**	{ $A.attrib$:= **double** }
A	$\longmapsto$	**single**	{ $A.attrib$:= **single** }
A	$\longmapsto$	**integer**	{ $A.attrib$:= **integer** }
A	$\longmapsto$	**float**	{ $A.attrib$:= **float** }

Exercise 3. Consider the following syntax-directed definition for the point size and height of boxes containing mathematical formulas.

Production	Semantic Rule
$S \longmapsto B$	$B.ps := 10$
	$S.ht := B.ht$
$B \longmapsto B_1\, B_2$	$B_1.ps := B.ps$
	$B_2.ps := B.ps$
	$B.ht := \mathsf{max}(B_1.ht, B_2.ht)$
$B \longmapsto B_1\, \textbf{sub}\, B_2$	$B_1.ps := B.ps$
	$B_2.ps := \mathsf{shrink}(B.ps)$
	$B.ht := \mathsf{disp}(B_1.ht, B_2.ht)$
$B \longmapsto \textbf{text}$	$B.ht := \textbf{text}.h \times B.ps$

Write a translation scheme for this syntax-directed definition.

Answer 3

See course notes, the *Syntax-Directed Translation* Chapter..