Compiler Theory CS447a: Quiz 3. **UWO, November 18, 2004.**

Student ID number:

Guidelines. The quiz is closed book and all notes are forbidden. The duration of the quiz is 40 minutes. Exercises 1 and 2 are worth 8 points each, whereas each of the other two is worth 12 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

Exercise 2. We consider the following translation scheme.

S	$\longmapsto$	$A\,\{C.i := A.s\}\,C$
S	$\longmapsto$	$AB\,\{C.i := A.s\}\,C$
A	$\longmapsto$	**int** $\{A.s :=$ **int**$.val\}$
B	$\longmapsto$	**id** $\{\}$
C	$\longmapsto$	**int** $\{C.s := C.i +$ **int**$.val\}$

where S, A, B, C are non-terminals, **int** and **id** are terminals, $C.i$ is an inherited attribute, $A.s$ and **int**$.val$ are synthesized attributes. Write a YACC program for this translation scheme.

Answer 2

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

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

Write a translation scheme for this syntax-directed definition.

Answer 3

Exercise 4. 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 pseudo-code 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 enter($\mathbf{id}.name$, $T.list$)
- The instruction $T.list$:= makelist($A.attrib$) will create the list consisting of the single item $A.attrib$.
- The instruction $T.list$:= cons($A.attrib$, $T.list$) will add the element $A.attrib$ at the beginning of the list $T.list$.

Answer 4