CS335: 3. Semantic Analysis   Due Date: March 3, 2019

1. Consider the following grammar, that represents a simple program having variable declarations followed by a sequence of assignment statements:

$$
\begin{array}{rcl}
P & \rightarrow & L\ S \\
L & \rightarrow & L\ D\ \mid\ \epsilon \\
D & \rightarrow & T\ V \\
V & \rightarrow & V\ \text{id}\ \mid\ \text{id} \\
S & \rightarrow & S\ A\ \mid\ A \\
A & \rightarrow & \text{id} = E \\
E & \rightarrow & E\ +\ E\ \mid\ E\ *\ E \mid\ (E) \\
 & \mid & \text{id} \mid\ \text{float-const}\ \mid\ \text{int-const} \\
T & \rightarrow & \text{int}\ \mid\ \text{float}
\end{array}
$$

We want to write a translator to estimate the total power consumption of any program accepted by the above grammar. Assume the following characteristics for the target machine:

- Power consumption for execution for primitive operations on **int** arguments:

Operation	**Description**	**Cost**
store (=)	store value in memory	s
loadV	load a variable's value from memory	v
loadC	load a constant	i
+	addition	a
*	multiplication	m

- For **float** arguments, the power consumption for any operation is 4 times the corresponding consumption for **int** operators.
- If an operation has mixed arguments (int-s and float-s), it will be considered a float operation (e.g. 3 * 4.0 will be float multiplication).
- For assignments, it is an error to assign float value to an int variable on LHS.
- The target machine has an infinite supply of registers, so that a variable need to be loaded only once. Assume write through policy (if a variable is already loaded in register, write will update both the memory location as well as the register).
- The power requirement to read from and write to a register is 0 (negligible).

(a) Create a Syntax Directed Definition (SDD) to compute power consumption for input programs in the language.

(b) What is the cost of the following program with your estimator:

```
float x z
int y
x = (y + 3) * 5.0
z = y + x
x = y * y
```

(c) Mention the type (synthesized/inherited) of every attribute that you use. Is your SDD S-attributed or L-attributed or none? Justify your answer.

2. For the infix-expression grammar:

$$\boxed{\begin{array}{lcll} \mathrm{E} & \rightarrow & \mathrm{E} + \mathrm{T} & | \;\; \mathrm{T} \\ \mathrm{T} & \rightarrow & \mathrm{T} * \mathrm{F} & | \;\; \mathrm{F} \\ \mathrm{F} & \rightarrow & (\,\mathrm{E}\,) & | \;\; \mathrm{id} \end{array}}$$

(a) Write a syntex directed definition (SDD) to translate the accepted infix expressions into *infix expressions without redundant paranthesis.* Use the natural precedence and associativity rules (* and + associate to the left, parenthesis has highest precedence followed by *, then +).
For example `( ( a * ( b + c ) ) * ( d ) )` will be translated as `a * ( b + c ) * d`.

(b) Is your SDD S-attributed, L-attributed or neither? Explain the reason.

The End