

# CS603 Organization of Programming Languages

## Lecture 6 Supplement

University of Alabama  
Department of Computer Science  
Spring 2004

In class on Wednesday, we worked through the judgments for

```
(val x 1)
(set x 2)
```

which have the corresponding “judgment” forms:

```
VAL(VAR(X),LITERAL(1))
EXP(SET(VAR(X),LITERAL(2)))
```

Note that the form for the “set” expression is slightly different than the one I gave in class. If we look at code chunks 12a and 12b in the textbook, we note that “SET” is not a top-level element, but an “EXP” element. If we evaluate the “val” and the “set” as if the interpreter has just started up, i.e. both the global value environment and the function environment are empty, then the state and term binding when beginning evaluation of the “set” statement is  $\xi = \{x \mapsto 1\}$ ,  $\phi = \{\}$ ,  $e = \text{SET}(\text{VAR}(x), \text{LITERAL}(2))$  and the judgment is:

$$\frac{\frac{x \notin \text{dom } \rho \quad x \in \text{dom } \xi \quad \langle \text{LITERAL}(2), \{x \mapsto 1\}, \{\}, \{\} \rangle \Downarrow \langle 2, \{x \mapsto 1\}, \{\}, \{\} \rangle}{\langle \text{SET}(\text{VAR}(x), \text{LITERAL}(2)), \{x \mapsto 1\}, \{\}, \{\} \rangle \Downarrow \langle 2, \{x \mapsto 2\}, \{\}, \{\} \rangle} \text{LITERAL}}{\langle \text{EXP}(\text{SET}(\text{VAR}(x), \text{LITERAL}(2))), \{x \mapsto 1\}, \{\} \rangle \rightarrow \langle \{x \mapsto 2\}, \{\} \rangle} \text{GLOBALASSIGN EVALEXP}$$

Note that the EVALEXP judgment adds an empty argument environment. This is so there aren’t two sets of judgments for expressions—one when an expression is evaluated at the top-level and another for when an expression is evaluated inside a function.