Semantics (V)

Side Effects

- Note: Our discussion above assumes there are no side effects caused by evaluating the expressions.
- A side effect occurs during the evaluation of an expression if, in addition to returning a value, the expression alters the state of the program.
- A typical example is the post- and pre- increment: \( x = 1; \quad ++x + x++; \)
- If there are side effects, the above semantics are not specific enough.
- The way to address side effects has two steps:
  1. Update the meaning function. Instead of producing a value, it returns a value and a state.

\[
M : Expression \times State \rightarrow Value \times State
\]

2. The meaning rule should be more specified. Take the binary expression for instance, the meaning rule also defines the order to determine the value of two terms.
   - If the Expression is a Binary, the meaning of \( \text{term1} \) in the current state is first determined, giving a value \( v1 \) and a state \( s1 \). Then the meaning of \( \text{term2} \) in state \( s1 \) is determined, giving a value \( v2 \) and a state \( s2 \). Then Meaning Rule of Binary Expression determines the meaning of the expression by applying the Operator \( op \) to the resulting values \( v1 \) and \( v2 \) in state \( s2 \), and the resulting state is \( s2 \).

- Note: We need to know the definition of side effect and the ways C and Java use to handle it. The implementation of side effects in meaning functions is out of the scope of the course.

Assignment Semantics

- We know that the rhs of an assignment statement is an expression. To determine the meaning of an assignment statement, we need to know the meaning of the rhs expression.
- Now, we know how to determine the meaning of an expression. Then, let’s figure out the way to determine the meaning of an assignment.

- We know that the meaning function of an expression generate a new value and the meaning function of an assignment generates a new state. (Remember: Assignment is a type of statement.)

\[
M : Expression \times State \rightarrow Value \\
M : Statement \times State \rightarrow State
\]
This means the value generated by the rhs expression is used to update the value of the lhs variable of the assignment. In the new state, the lhs variable will be associated with the new value generated by the rhs expression. Expressed this mathematically,

$$M(\text{Assignmenta}, \text{State state}) = \text{state} \cup \{a\cdot \text{target}, M(a\cdot \text{source}, \text{state})\}$$

Here, $\cup$ means overriding union.

If we have a set $X$ which contains three pairs, $\{<a, 1>, <b, 5>, <c, 1>\}$, and a set $Y$ which contains two pairs, $\{<b, 6>, <d, 0>\}$, the overriding union of $X$ and $Y$ is:

\[
X = \{\langle a, 1 \rangle, \langle b, 5 \rangle, \langle c, -1 \rangle\} \\
Y = \{\langle b, 6 \rangle, \langle d, 0 \rangle\} \\
X \cup Y = \{\langle a, 1 \rangle, \langle b, 6 \rangle, \langle c, -1 \rangle, \langle d, 0 \rangle\}
\]

The **overriding union** of $X$ and $Y$, written $X \cup Y$, is the result of replacing in $X$ all pairs $(x,v)$ whose first member matches a pair $(x,w)$ from $Y$ by $(x,w)$ and then adding to $X$ any remaining pairs in $Y$. 