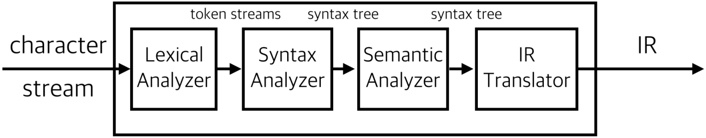
COSE312: Compilers

Lecture 11 — Semantic Analysis (1)

Hakjoo Oh
2017 Spring

Semantic Analysis




```
int a[10] = {...};
int x = rand();
int y = 1;
if (x > 0) {
  if (x < 15) {
    if (x < 10) a[x] = "hello" + 1;
    a[x] = 1;
  }
} else {
  y = y / x;
}
```

Syntax vs. Semantics

A programming language is defined with syntax and semantics.

- The syntax is concerned with the grammatical structure of programs.
  - Context-free grammar
- The semantics is concerned with the meaning of grammatically correct programs.
  - Operational semantics: The meaning is specified by the computation steps executed on a machine. It is of intrest how it is obtained.
  - Denotational semantics: The meaning is modelled by mathematical objects that represent the effect of executing the program. It is of interest the effect, not how it is obtained.

The **While** Language

n will range over numerals, **Num**
x will range over variables, **Var**
a will range over arithmetic expressions, **Aexp**
b will range over boolean expressions, **Bexp**
S will range over statements, **Stm**

$$
\begin{array}{lcl}
a & \rightarrow & n \mid x \mid a_1 + a_2 \mid a_1 \star a_2 \mid a_1 - a_2 \\
b & \rightarrow & \texttt{true} \mid \texttt{false} \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \neg b \mid b_1 \wedge b_2 \\
c & \rightarrow & x := a \mid \texttt{skip} \mid c_1 ; c_2 \mid \texttt{if}\ b\ c_1\ c_2 \mid \texttt{while}\ b\ c
\end{array}
$$

Semantics of Arithmetic Expressions

- The meaning of an expression depends on the values bound to the variables that occur in the expression, e.g., $x + 3$.
- A state is a function from variables to values:

$$\mathbf{State} = \mathbf{Var} \rightarrow \mathbb{Z}$$

- The meaning of arithmetic expressions is a function:

$$\mathcal{A} : \mathbf{Aexp} \rightarrow \mathbf{State} \rightarrow \mathbb{Z}$$

$$\begin{array}{rcl}
\mathcal{A}[\![\, a \,]\!] & : & \mathbf{State} \rightarrow \mathbb{Z} \\
\mathcal{A}[\![\, n \,]\!](s) & = & n \\
\mathcal{A}[\![\, x \,]\!](s) & = & s(x) \\
\mathcal{A}[\![\, a_1 + a_2 \,]\!](s) & = & \mathcal{A}[\![\, a_1 \,]\!](s) + \mathcal{A}[\![\, a_2 \,]\!](s) \\
\mathcal{A}[\![\, a_1 \star a_2 \,]\!](s) & = & \mathcal{A}[\![\, a_1 \,]\!](s) \times \mathcal{A}[\![\, a_2 \,]\!](s) \\
\mathcal{A}[\![\, a_1 - a_2 \,]\!](s) & = & \mathcal{A}[\![\, a_1 \,]\!](s) - \mathcal{A}[\![\, a_2 \,]\!](s)
\end{array}$$

Exercise

Add the arithmetic expression $-a$ to our language.

Semantics of Boolean Expressions

- The meaning of boolean expressions is a function:

$$\mathcal{B} : \mathbf{Bexp} \to \mathbf{State} \to \mathbf{T}$$

where $\mathbf{T} = \{true, false\}$.

$$\begin{array}{rcl}
\mathcal{B}[\![b]\!] & : & \mathbf{State} \to \mathbf{T} \\
\mathcal{B}[\![\texttt{true}]\!](s) & = & true \\
\mathcal{B}[\![\texttt{false}]\!](s) & = & false \\
\mathcal{B}[\![a_1 = a_2]\!](s) & = & \mathcal{A}[\![a_1]\!](s) = \mathcal{A}[\![a_2]\!](s) \\
\mathcal{B}[\![a_1 \leq a_2]\!](s) & = & \mathcal{A}[\![a_1]\!](s) \leq \mathcal{A}[\![a_2]\!](s) \\
\mathcal{B}[\![\neg b]\!](s) & = & \mathcal{B}[\![b]\!](s) = false \\
\mathcal{B}[\![b_1 \land b_2]\!](s) & = & \mathcal{B}[\![b_1]\!](s) \land \mathcal{B}[\![b_2]\!](s)
\end{array}$$

Free Variables

The free variables of an arithmetic expression a are defined to be the set of variables occurring in it:

$$
\begin{array}{rcl}
FV(n) & = & \emptyset \\
FV(x) & = & \{x\} \\
FV(a_1 + a_2) & = & FV(a_1) \cup FV(a_2) \\
FV(a_1 \star a_2) & = & FV(a_1) \cup FV(a_2) \\
FV(a_1 - a_2) & = & FV(a_1) \cup FV(a_2)
\end{array}
$$

Exercise

Define free variables of boolean expressions.

Property of Free Variables

Lemma

Let s *and* s' *be two states satisfying that* $s(x) = s'(x)$ *for all* $x \in FV(a)$*. Then,* $\mathcal{A}[\![a]\!](s) = \mathcal{A}[\![a]\!](s')$*.*

Lemma

Let s *and* s' *be two states satisfying that* $s(x) = s'(x)$ *for all* $x \in FV(b)$*. Then,* $\mathcal{B}[\![b]\!](s) = \mathcal{B}[\![b]\!](s')$*.*

Substitution

- $a[y \mapsto a_0]$: the arithmetic expression that is obtained by replacing each occurrence of y in a by a_0.

$$\begin{array}{rcl} n[y \mapsto a_0] & = & n \\ x[y \mapsto a_0] & = & \begin{cases} a_0 & \text{if } x = y \\ x & \text{if } x \neq y \end{cases} \\ (a_1 + a_2)[y \mapsto a_0] & = & (a_1[y \mapsto a_0]) + (a_2[y \mapsto a_0]) \\ (a_1 \star a_2)[y \mapsto a_0] & = & (a_1[y \mapsto a_0]) \star (a_2[y \mapsto a_0]) \\ (a_1 - a_2)[y \mapsto a_0] & = & (a_1[y \mapsto a_0]) - (a_2[y \mapsto a_0]) \end{array}$$

- $s[y \mapsto v]$: the state s except that the value bound to y is v.

$$(s[y \mapsto v])(x) = \begin{cases} v & \text{if } x = y \\ s(x) & \text{if } x \neq y \end{cases}$$

Exercise

Prove that the two concepts of substitutions are related as follows:

Lemma

$\mathcal{A}[\![\ a[y \mapsto a_0]\ ]\!](s) = \mathcal{A}[\![\ a\ ]\!](s[y \mapsto \mathcal{A}[\![\ a_0\ ]\!](s)])$ *for all states* s.

Exercise

Define substitution for boolean expressions: $b[y \mapsto a_0]$. Prove that

$$\mathcal{B}[\![\ b[y \mapsto a_0]\ ]\!](s) = \mathcal{B}[\![\ b\ ]\!](s[y \mapsto \mathcal{A}[\![\ a_0\ ]\!](s)])$$

holds for all states s.

Mid-term Exam

- 4/24 (Mon), 15:30–16:45 (in class)
- Do not be late.
- Coverage: lexical analysis, syntax analysis
- Based on lectures and homework.
- No classes on 4/26 (Wed).