18 Denotational Semantics

Wednesday, November 13, 2024 12:10 PM

- Another approach to specify semantics (meaning) of a programming language
- Most* widely used.

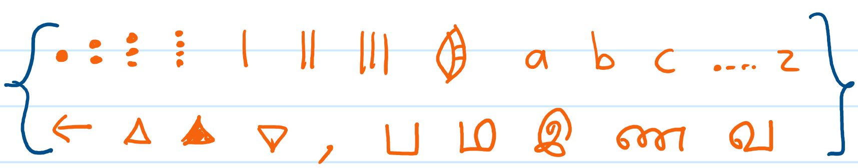
IDEA:

For each syntactic entity in the programming language
Define a function that maps that entity to a known Mathematical Object.

M: Syntactic Entity → Known Math Object.

EXAMPLE:

Alphabet (Terminal)



- Constants (Literals)

Grammar:

DI → • | : | ⁝ | ⋮

DV → | | || | |||

C → DI | DV | DI DV | θ

Example strings:

"|" 5   "θ" 0   "•" 1

"⁝||" 13   "•|||" 16

Meaning:

$M_{DI}(\cdot) = 1$ $M_{DV}(|) = 5$

$M_{DI}(:) = 2$ $M_{DV}(||) = 10$

$M_{DI}(⁝) = 3$ $M_{DV}(|||) = 15$

$M_{DI}(⋮) = 4$

$M_C(⁝||) = M_{DI}(⁝) + M_{DV}(||) = 3 + 10 = 13$

$M_C(\cdot|||) = M_{DI}(\cdot) + M_{DV}(|||) = 1 + 15 = 16$

$M_C(DI) = M_{DI}(DI)$

$M_C(DV) = M_{DV}(DV)$

$M_c(DI) = M_{DI}(DI)$

$M_c(DV) = M_{DV}(DV)$

$M_c(DI\ DV) = M_{DI}(DI) + M_{DV}(DV)$

$M_c(\text{Ø}) = 0$

$= 1 + 15$

$= 16$

Grammar.

$NUM \rightarrow C \mid NUM, C$

Example strings

"1,1" 105 "11,Ø" 260 "·,1,:1" 507 "·,Ø,Ø" 400

Meaning

$M_{Num}(c) = M_c(c)$

$M_{Num}(NUM, C) = M_c(c) + 20 \cdot M_{Num}(NUM)$

[Base 20 number system "Vigesimal" System.]

$M_{Num}(\cdot,1,:1) = M_c(:1) + 20 \cdot M_{Num}(\cdot,1)$

$= 7 + 20(M_c(1) + 20 \cdot M_{Num}(\cdot))$

$= 7 + 20 \cdot 5 + 20 \cdot 20 \cdot M_c(\cdot)$

$= 7 + 20 \cdot 5 + 20^2 \cdot 1$

$= 507$

$M_{Num}(\cdot, \text{Ø}, \text{Ø})$

$= M_c(\text{Ø}) + 20 \cdot M_c(\text{Ø}) + 20^2 \cdot M_c(\cdot)$

$= 400$

- Variables

Grammar

$V \rightarrow a \mid b \mid c \mid d \mid e \mid \ldots \mid z$

Meaning:

The meaning of a variable is dependent on STATE

DEF: State: a set of $\langle V, \text{Value} \rangle$ pairs where V are unique.

i.e. a "mapping" from variable names to values

Let our values be integers

e.g. $s = \{\langle a, 8\rangle \langle b, 7\rangle \langle z, 0\rangle\}$

$M_V(V, s) = val$ s.t. $\langle V, val\rangle \in S$

e.g.

$M(\ldots)$

e.g.

$M_V(b, \{\langle a,8\rangle \langle b,7\rangle \langle z,0\rangle\}) = 7$

- Expressions.

Grammar.

Expr → V | Num | Expr △ Expr | Expr ▲ Expr | Expr ▽ Expr.

"a" ":|" ":|△:" "||▽c" "b▽a▲:|"

Meaning:

$M_{Expr}(V,S) = M_V(V,S)$

$M_{Expr}(Num,S) = M_{Num}(Num)$

$M_{Expr}(Expr_1 \triangle Expr_2, S) = M_{Exp}(Expr_1, S) + M_{Expr}(Expr_2, S)$

$M_{Expr}(Epr_1 \blacktriangle Expr_2, S) = M_{Exp}(Expr_1, S) \cdot M_{Expr}(Expr_2, S)$

$M_{Expr}(Epr_1 \triangledown Expr_2, S) = M_{Exp}(Expr_1, S) - M_{Expr}(Expr_2, S)$

÷ a ** b a^b

E.g.

$M_{Expr}(b \triangledown a \blacktriangle :|, S)$ $S = \{\langle a,7\rangle \langle b,4\rangle \langle z,8\rangle\}$

$M_{Expr}(b \triangledown a, s) \cdot M_{Exp}(:|, s) =$

$M_{Expr}(b,s) - (M_{Expr}(a,s) \cdot M_{Exp}(:|, s)) =$

$M_V(b,s) - (M_V(a,s) \cdot M_{Num}(:|)) =$

$4 - (7 \cdot 8)$

$4 - 56$

-52

- Assignment.

Grammar

A → V ← Expr "x←:" "y←:▲x△:||"

Meaning.

$M_A(V \leftarrow Expr, s) = s'$

$s' = (s - \{\langle V, val\rangle\}) \cup \{(V, M_{Expr}(Expr, S))\}$

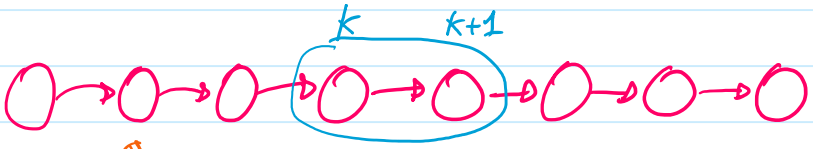
e.g.

$M_A(x \leftarrow :, s) =$ $\quad s = \{\langle x\ 7\rangle, \langle y\ 4\rangle \langle z\ 9\rangle\}$

$s' = (s - \{\langle x\ 7\rangle\}) \cup \{\langle x, M_{Exp}(:, s)\rangle\}$

$\{\langle x, 3\rangle\}$

$s' = \{\langle x\ 3\rangle \langle y\ 4\rangle \langle z\ 9\rangle\}$



y ← 0

x ← : ▲ :|

y ← y Δ •

z ← :| ▽ x

Conditionals.

Grammar :

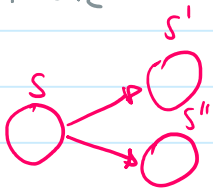
$I \rightarrow$ @ Expr ⊔ block ⊔ $block_2$

"@ x ▽ | ⊔ x ← : ▲ : ⊔ x ← 0"

Expr, $block_1$, $block_2$

Meaning :

M_I (@ Expr ⊔ $block_1$ ⊔ $block_2$, s) = s'



$$s' \begin{cases} M(block_1, s) & \text{if } M_{Expr}(Expr, s) \neq 0 \\ M(block_2, s) \end{cases}$$

E.g.

M_I (@ x ▽ | ⊔ x ← : Δ : ⊔ x ← 0, s) $\quad s\{\langle x\ 6\rangle \langle y\ 9\rangle\}$

depends on $M_{Exp}(x \triangledown 1, s)$

$M_V(x, s) - M_{Num}(1)$

$6 \quad - \quad 5 \quad = 1$

$= M_A(x \leftarrow : \blacktriangle :, s)$

$\hookrightarrow = M_A(x \leftarrow \vdots\Delta\vdots, s)$

$= \{\langle x, M_{Expr}(\vdots\Delta\vdots, s)\rangle \langle y\ 9\rangle$

$= \{\langle x\ 5\rangle\langle y\ 9\rangle\}$

- Iteration:

Grammar.

W → ௵ Expr ௳ block "௵ y ௳ y ← y ∇∘ "

Meaning

M_W (௵ Expr ௳ block, s) = s'

$$s' \begin{cases} s & \text{if } M_{Expr}(Expr, s) = Q \\ M_W(\text{௵ Expr ௳ block}, M(block, s)) \end{cases}$$