Semantics

operational semantics

k - framework

$X:$ [1]

$X = X + 1$

$X:$ [2]
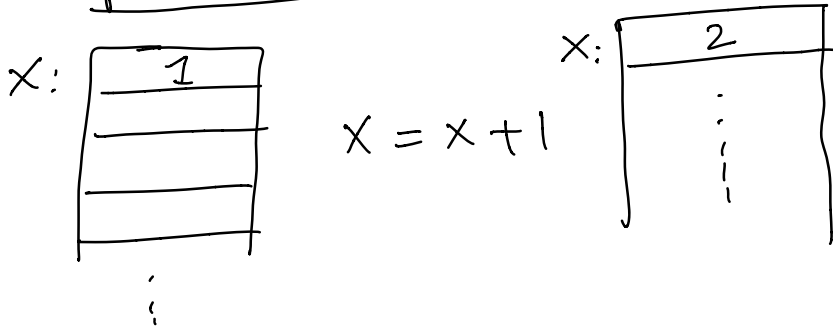
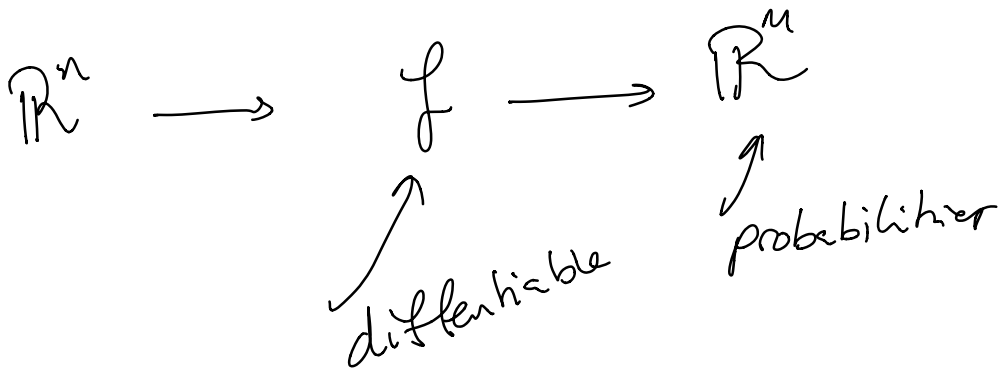In λ calculus, β reduction is an operational semantics

denotational semantics

pixels $\longrightarrow$ Neural network $\longrightarrow$ cat /dog

$\mathbb{R}^n \longrightarrow f \longrightarrow \mathbb{R}^m$

differentiable

probabilities

axiomatic semantics

$$x = x + 1$$

if $x > 0$ then after executing $x = x + 1$

x is still > 0

Semantics with applications

arithmetic expressions

$$a := n \mid x \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2$$

Boolean expressions

$$b := true \mid false \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \underbrace{\neg b_1}_{not} \mid \underbrace{b_1 \wedge b_2}_{AND}$$

program

$$P := x := a \mid skip \mid P_1 ; P_2 \mid$$

$$if\ b\ then\ P_1\ else\ P_2$$

$$\mid while\ b\ then\ P_1$$

$\underline{Var}$ is the set of all variables

a state $s : Var \longrightarrow \mathbb{Z}$ ← integers

E.g. $S = \begin{cases} x \longmapsto 10 \\ y \longmapsto 70 \\ z \longmapsto 200 \\ \vdots \end{cases}$

$[x/0]$

$S[x \longmapsto 0] = \begin{cases} x \longmapsto 0 \\ y \longmapsto 70 \\ z \longmapsto 200 \end{cases}$

Substitution

Semantics of expressions

$[\![a]\!] : State \longrightarrow \mathbb{Z}$

set of all possible states

$\{true, false\}$

$[\![b]\!] : State \longrightarrow \mathbb{B}$

E.g. $[\![x+y]\!] = 0 + 70 = 70$

$[\![x = y]\!] = false$

Natural (or big-step) semantics

$$\langle P, S \rangle \longrightarrow S'$$

program → (arrow to P)

starting state → (arrow to S)

final state → (arrow to S')

| skip | $\langle \text{skip}, S \rangle \longrightarrow S$ |

| asn | $\langle x := a, S \rangle \longrightarrow S[x \rightarrow [\![a]\!](s)]$ |

| sequential comp | $\langle P_1 ; P_2, S \rangle \longrightarrow S''$ |

where $\langle P_1, S \rangle \longrightarrow S'$

$\langle P_2, S' \rangle \longrightarrow S''$

$$\frac{\langle P_1, S \rangle \longrightarrow S' \qquad \langle P_2, S' \rangle \longrightarrow S''}{\langle P_1 ; P_2, S \rangle \longrightarrow S''}$$

if (true)

$$\langle \text{if } b \text{ then } P_1 \text{ else } P_2, S \rangle \longrightarrow S'$$
$$\text{if } [\![b]\!] (S) = \text{true}$$
$$\text{and } \langle P_1, S \rangle \longrightarrow S'$$

$$\langle \text{if } b \text{ then } P_1 \text{ else } P_2, S \rangle \longrightarrow S'$$
$$\text{if } [\![b]\!] (S) = \text{false}$$
$$\text{and } \langle P_2, S \rangle \longrightarrow S'$$

while (true)

$$\langle \text{while } b \text{ do } P, \ s \rangle \longrightarrow s''$$

if $[\![b]\!](s) = \text{true}$

and $\langle P, s \rangle \longrightarrow s'$

and $\langle \text{while } b \text{ do } P, s' \rangle \longrightarrow s''$

while (false)

$$\langle \text{while } b \text{ do } P, \ s \rangle \longrightarrow s$$

if $[\![b]\!](s) = \text{false}$

E.g. $\langle z := x ; \ x := y ; \ y := z \ , \ s_0 \rangle$

$\langle z := x, \ s_0 \rangle \longrightarrow s_1$ $\langle x := y, \ s_1 \rangle \longrightarrow s_2$

$S_0 = \begin{cases} x \mapsto 5 \\ y \mapsto 7 \\ z \mapsto 0 \end{cases}$ $S_1 = \begin{cases} x \mapsto 5 \\ y \mapsto 7 \\ z \mapsto 5 \end{cases}$ $S_2 = \begin{cases} x \mapsto 7 \\ y \mapsto 7 \\ z \mapsto 5 \end{cases}$

Properties

two programs P_1 and P_2 are equivalent iff

forall s, s' $\langle P_1 , s \rangle \longrightarrow s'$ iff

$$\langle P_2 , s \rangle \longrightarrow s'$$

e.g. we can prove that

     while b do P

   is equivalent to

if b then

    P ;

   while b do P

else

   skip ;

Theorem : the semantics are deterministic

assume $\langle P, s \rangle \longrightarrow s'$

if $\langle P, s \rangle \longrightarrow s''$, then $s' = s''$

---

program semantics

$[\![P]\!] : \text{State} \longrightarrow \text{State}$

this is a partial function

$$[\![P]\!](s) = \begin{cases} s' & \text{if } \langle P, s \rangle \longrightarrow s' \\ \text{undef} & \text{o/w} \end{cases}$$

non-termination

E.g. $[\![\text{while true do } x := 1]\!](s) = \text{undef}$

$[\![a]\!] : State \longrightarrow \mathbb{Z}$

$x + y + z$

$\uparrow \quad \uparrow$

Concurrent / Shared memory setting

$\langle P, S \rangle \longrightarrow \langle P', S' \rangle$