The following slides have been adapted from the material of the course

Principles for Software Composition

http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/start

by

Roberto Bruni (University of Pisa, Italy)

http://www.di.unipi.it/~bruni/

Used with permission of the author

Previously on MaSC

- IMP: a core language for imperative programming; syntax, op. semantics, rule induction..
- Lambda-notation
- Today we look at HOFL: a core language for typed functional programming

HOFL Syntax

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \ \mathsf{op} \ t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

HOFL Syntax

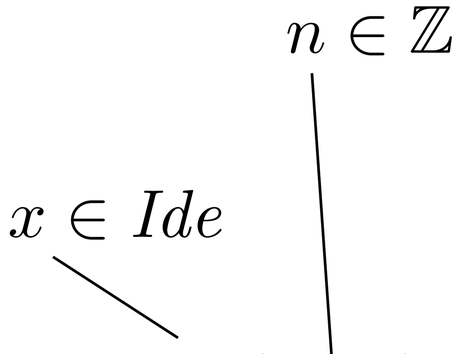
$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \quad \text{ordinary ops} \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

HOFL Syntax

$x \in Ide$

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \quad \text{ordinary ops} \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec}\ x.\ t
\end{array}
$$

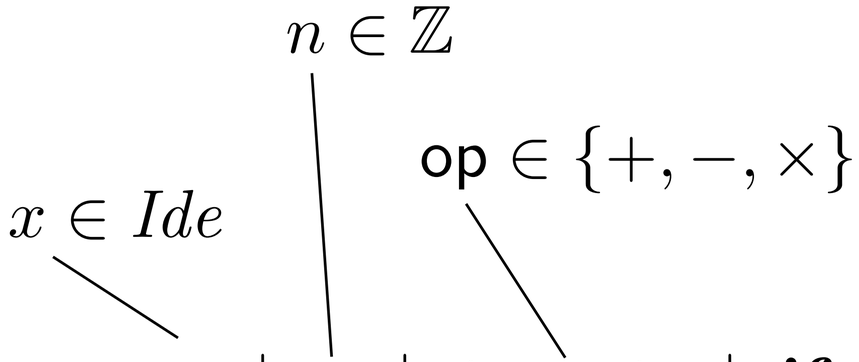
HOFL Syntax



$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

ordinary ops

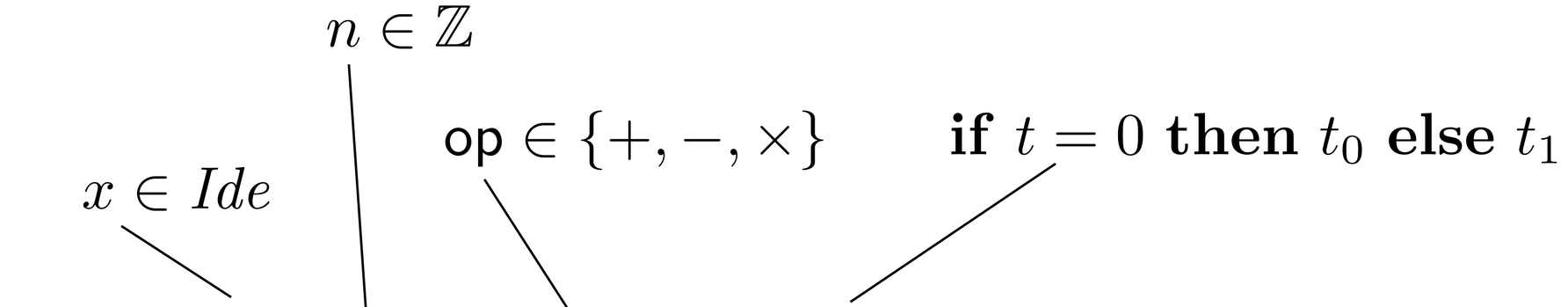
HOFL Syntax



$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \ \mathsf{op} \ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1 \\
& \mid & (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \mathbf{rec}\ x.\ t
\end{array}
$$

ordinary ops

HOFL Syntax



$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0\ \mathsf{op}\ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1 \\
& \mid & (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \mathbf{rec}\ x.\ t
\end{array}
$$

ordinary ops

HOFL Syntax

$n \in \mathbb{Z}$

$\mathsf{op} \in \{+, -, \times\}$

$x \in Ide$

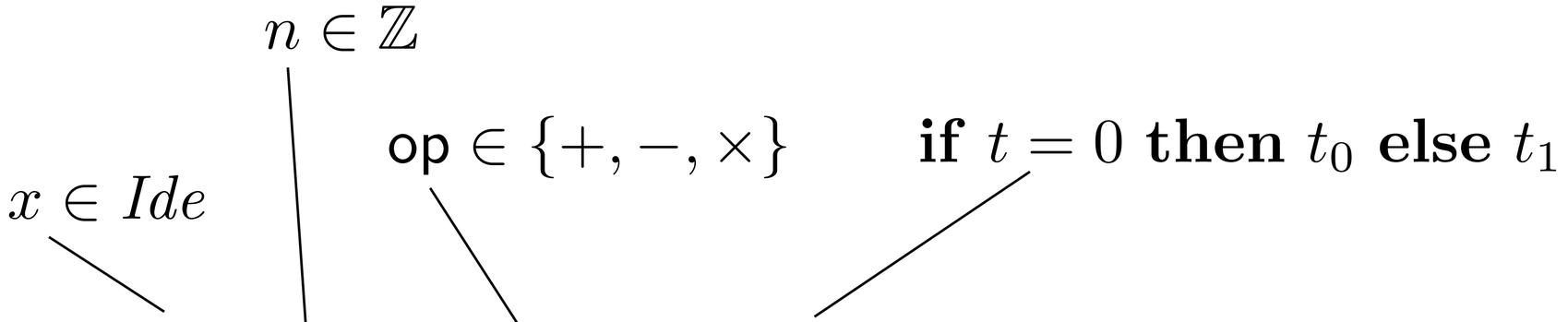
$\mathbf{if}\ t = 0\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1$

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0\ \mathsf{op}\ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1 \\
& \mid & (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \mathbf{rec}\ x.\ t
\end{array}
$$

ordinary ops

pairing, projections

HOFL Syntax



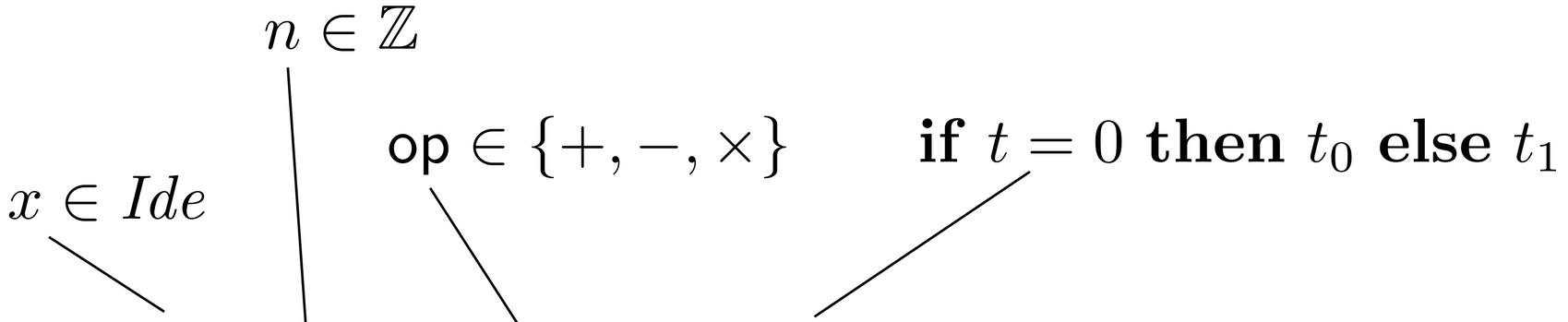
$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0\ \mathsf{op}\ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1 \\
& \mid & (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \mathbf{rec}\ x.\ t
\end{array}
$$

ordinary ops

pairing, projections

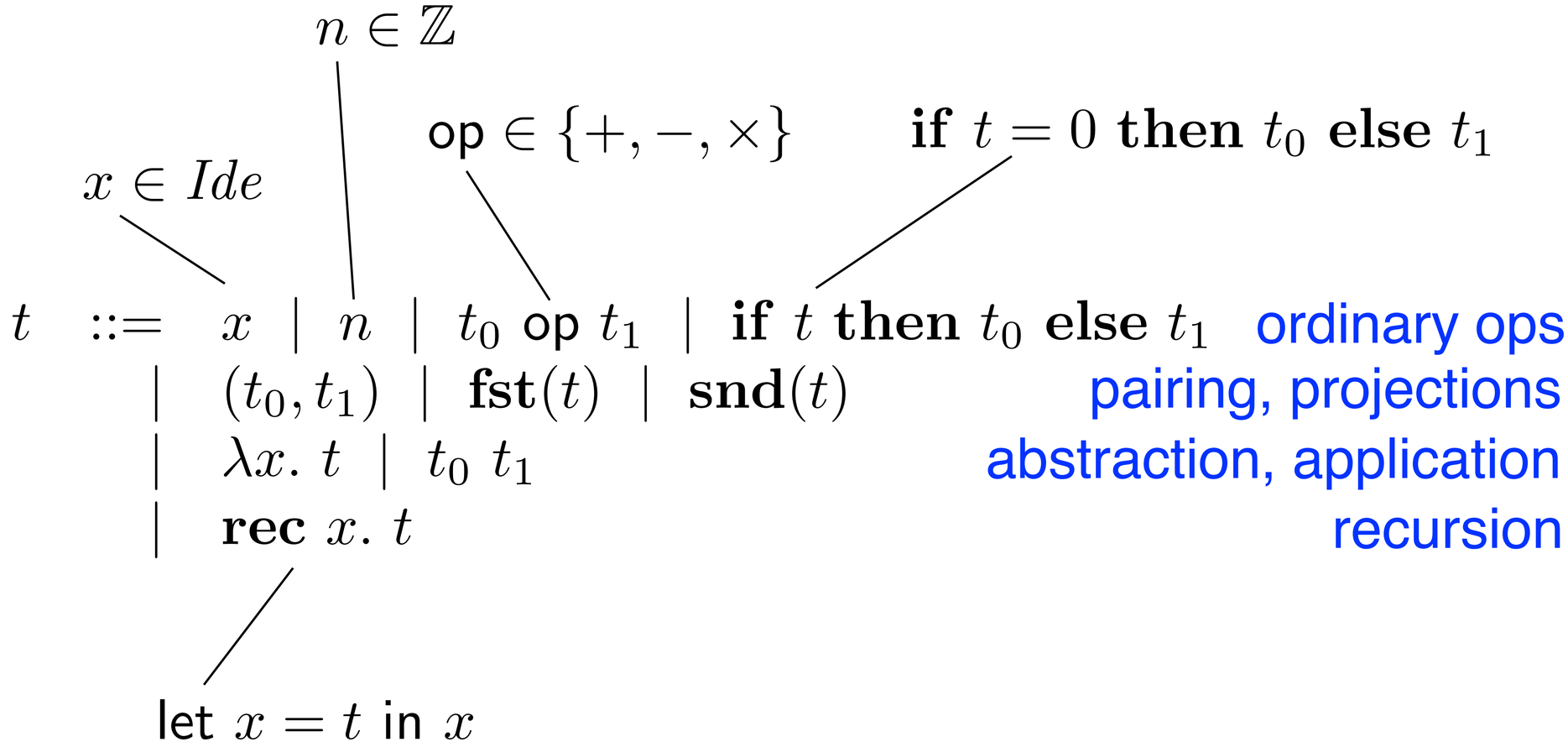
abstraction, application

HOFL Syntax

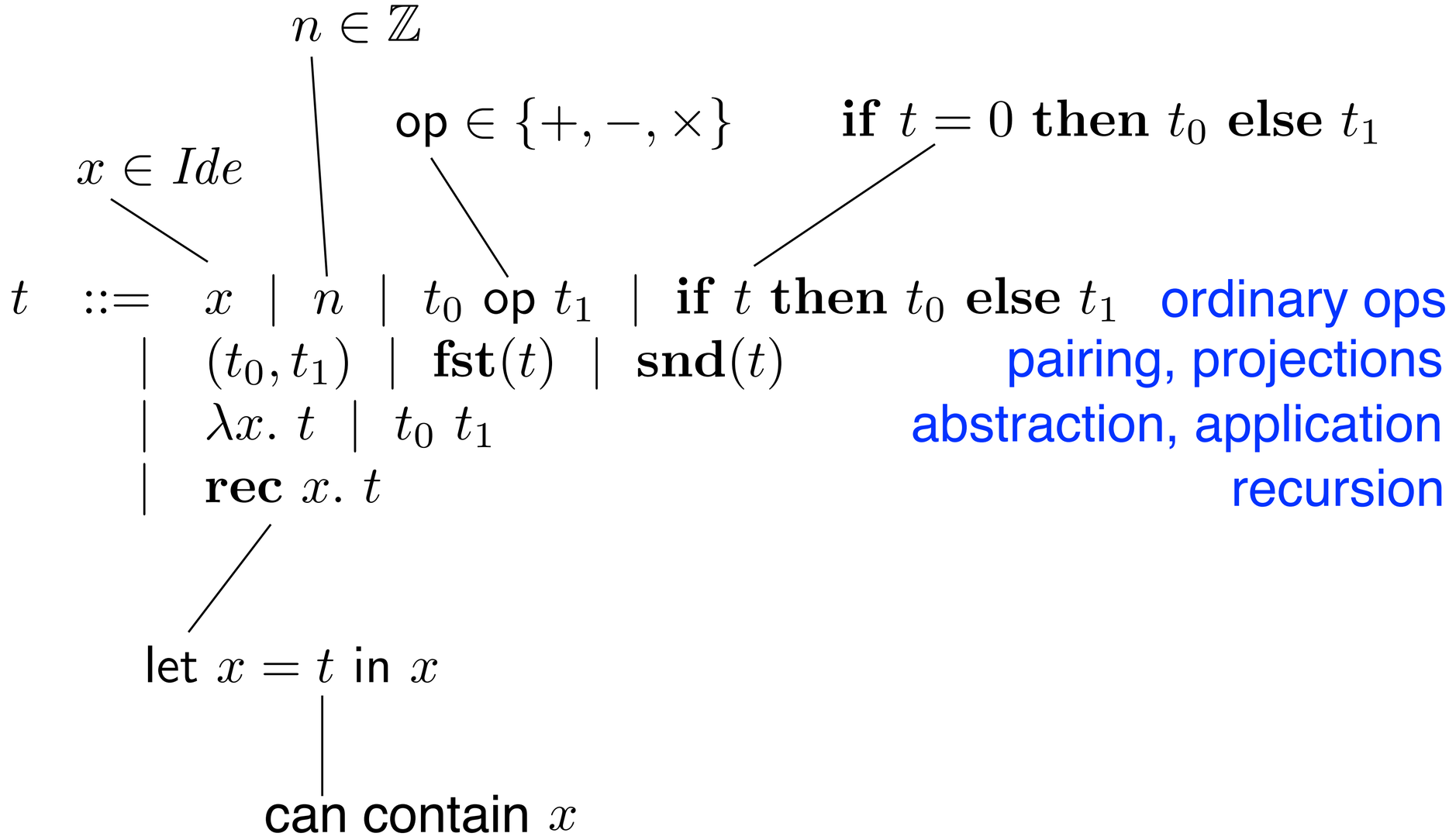


$$
\begin{array}{rcll}
t & ::= & x \mid n \mid t_0\ \mathsf{op}\ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1 & \text{ordinary ops} \\
& \mid & (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t) & \text{pairing, projections} \\
& \mid & \lambda x.\ t \mid t_0\ t_1 & \text{abstraction, application} \\
& \mid & \mathbf{rec}\ x.\ t & \text{recursion}
\end{array}
$$

HOFL Syntax


$n \in \mathbb{Z}$
$x \in Ide$
$\mathsf{op} \in \{+, -, \times\}$
$\mathbf{if}\ t = 0\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1$
$t ::= x \mid n \mid t_0\ \mathsf{op}\ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1$
ordinary ops
$\mid (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t)$
pairing, projections
$\mid \lambda x.\ t \mid t_0\ t_1$
abstraction, application
$\mid \mathbf{rec}\ x.\ t$
recursion
$\mathsf{let}\ x = t\ \mathsf{in}\ x$

HOFL Syntax


$n \in \mathbb{Z}$
$x \in Ide$
$\mathsf{op} \in \{+, -, \times\}$
$\mathbf{if}\ t = 0\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1$
$t ::= x \mid n \mid t_0\ \mathsf{op}\ t_1 \mid \mathbf{if}\ t\ \mathbf{then}\ t_0\ \mathbf{else}\ t_1$
ordinary ops
$\mid (t_0, t_1) \mid \mathbf{fst}(t) \mid \mathbf{snd}(t)$
pairing, projections
$\mid \lambda x.\ t \mid t_0\ t_1$
abstraction, application
$\mid \mathbf{rec}\ x.\ t$
recursion
let $x = t$ in x
can contain x

Exercise

$$\mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

guess the meaning of the above pre-term

Exercise

$$\mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

guess the meaning of the above pre-term

factorial

Exercise

$$\mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

guess the meaning of the above pre-term

factorial

```
fact x = if x then 1
         else x * (fact (x-1))
```

Exercise

$$\mathbf{rec}\ rep.\ \lambda n.\ \lambda f.\ \lambda x.\ \mathbf{if}\ n\ \mathbf{then}\ x\ \mathbf{else}\ f\ (rep\ (n-1)\ f\ x)$$

guess the meaning of the above pre-term

Exercise

$$\textbf{rec}\ rep.\ \lambda n.\ \lambda f.\ \lambda x.\ \textbf{if}\ n\ \textbf{then}\ x \\ \textbf{else}\ f\ (rep\ (n-1)\ f\ x)$$

guess the meaning of the above pre-term

$$rep\ n\ f\ x = f^n\ x$$

Exercise

$$\textbf{rec}\ rep.\ \lambda n.\ \lambda f.\ \lambda x.\ \textbf{if}\ n\ \textbf{then}\ x\ \textbf{else}\ f\ (rep\ (n-1)\ f\ x)$$

guess the meaning of the above pre-term

$$rep\ n\ f\ x = f^n\ x$$

```
rep n f x = if n then x
            else f (rep (n-1) f x)
```

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ 

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ 

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ ✅

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$ ✅

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ ✅

$1 + (0, 5)$

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$ ✅

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ ✓

$1 + (0, 5)$ ✗

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$ ✓

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ ✓

$1 + (0, 5)$ ✗

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$ ✓

$\textbf{fst}(3)$

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ ✓

$1 + (0, 5)$ ✗

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$ ✓

$\textbf{fst}(3)$ ✗

Pre-terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\ t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\ t
\end{array}
$$

they are called pre-terms, why?

$x + 1$ ✓

$\textbf{if } x \textbf{ then } x + 1 \textbf{ else } x - 1$ ✓

✗ $1 + (0, 5)$

✗ $\textbf{fst}(3)$

we need a type system

HOFL types

Types Syntax

$$\tau \ ::= \ int \ \mid \ \tau_0 * \tau_1 \ \mid \ \tau_0 \rightarrow \tau_1$$

$\mathcal{T}$ set of all types

Types Syntax

$$\tau \ ::= \ int \ \mid \ \tau_0 * \tau_1 \ \mid \ \tau_0 \rightarrow \tau_1$$

$\mathcal{T}$ set of all types

example types:

Types Syntax

$$\tau \ ::= \ int \ \mid \ \tau_0 * \tau_1 \ \mid \ \tau_0 \to \tau_1$$

$\mathcal{T}$ set of all types

example types: $int * int$ $int * (int \to int)$ $(int * int) \to int$

Types Syntax

$$\tau \ ::= \ int \ \mid \ \tau_0 * \tau_1 \ \mid \ \tau_0 \to \tau_1$$

$\mathcal{T}$ set of all types

example types: $int * int$ $int * (int \to int)$ $(int * int) \to int$

assume variables are typed $Ide = \{Ide_\tau\}_{\tau \in \mathcal{T}}$

Types Syntax

$$\tau \ ::= \ int \ \mid \ \tau_0 * \tau_1 \ \mid \ \tau_0 \to \tau_1$$

$\mathcal{T}$ set of all types

example types: $int * int$ $int * (int \to int)$ $(int * int) \to int$

assume variables are typed $Ide = \{Ide_\tau\}_{\tau \in \mathcal{T}}$

$\widehat{\cdot} : Ide \to \mathcal{T}$

Types Syntax

$\tau \ ::= \ int \ \mid \ \tau_0 * \tau_1 \ \mid \ \tau_0 \to \tau_1$

$\mathcal{T}$ set of all types

example types: $int * int$ $int * (int \to int)$ $(int * int) \to int$

assume variables are typed $Ide = \{Ide_\tau\}_{\tau \in \mathcal{T}}$

$\widehat{\cdot} : Ide \to \mathcal{T}$

$\widehat{x}$ denotes the type of x

Type judgements

reads “has type”

formulas: $t : \tau$

types are assigned to pre-terms
using a set of inference rules
(structural induction of HOFL syntax)

Type judgements

formulas: $t : \tau$ reads “has type”

types are assigned to pre-terms
using a set of inference rules
(structural induction of HOFL syntax)

Type system

Type system

$$\frac{}{x : \widehat{x}}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{}{t_0 \text{ op } t_1 :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{}{(t_0, t_1) :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{}{\textbf{fst}(t) :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{}{\textbf{snd}(t) :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

$$\frac{}{\lambda x.\ t :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

$$\frac{x : \tau_0 \quad t : \tau_1}{\lambda x.\ t : \tau_0 \to \tau_1}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

$$\frac{x : \tau_0 \quad t : \tau_1}{\lambda x.\ t : \tau_0 \to \tau_1} \qquad \frac{}{t_1\ t_0 :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

$$\frac{x : \tau_0 \quad t : \tau_1}{\lambda x.\ t : \tau_0 \to \tau_1} \qquad \frac{t_1 : \tau_0 \to \tau_1 \quad t_0 : \tau_0}{t_1\ t_0 : \tau_1}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

$$\frac{x : \tau_0 \quad t : \tau_1}{\lambda x.\ t : \tau_0 \to \tau_1} \qquad \frac{t_1 : \tau_0 \to \tau_1 \quad t_0 : \tau_0}{t_1\ t_0 : \tau_1}$$

$$\frac{}{\textbf{rec } x.\ t :}$$

Type system

$$\frac{}{x : \widehat{x}} \qquad \frac{}{n : int} \qquad \frac{t_0 : int \quad t_1 : int}{t_0 \text{ op } t_1 : int} \qquad \frac{t : int \quad t_0 : \tau \quad t_1 : \tau}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 : \tau}$$

$$\frac{t_0 : \tau_0 \quad t_1 : \tau_1}{(t_0, t_1) : \tau_0 * \tau_1} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{fst}(t) : \tau_0} \qquad \frac{t : \tau_0 * \tau_1}{\textbf{snd}(t) : \tau_1}$$

$$\frac{x : \tau_0 \quad t : \tau_1}{\lambda x.\ t : \tau_0 \to \tau_1} \qquad \frac{t_1 : \tau_0 \to \tau_1 \quad t_0 : \tau_0}{t_1\ t_0 : \tau_1}$$

$$\frac{x : \tau \quad t : \tau}{\textbf{rec } x.\ t : \tau}$$

Well-formed terms

$$
\begin{array}{rcl}
t & ::= & x \mid n \mid t_0 \text{ op } t_1 \mid \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \\
& \mid & (t_0, t_1) \mid \textbf{fst}(t) \mid \textbf{snd}(t) \\
& \mid & \lambda x.\, t \mid t_0\ t_1 \\
& \mid & \textbf{rec } x.\, t
\end{array}
$$

$$
\tau ::= int \mid \tau_0 * \tau_1 \mid \tau_0 \to \tau_1
$$

$\mathcal{T}$ set of all types

a pre-term t is *well formed* if $\exists \tau \in \mathcal{T}.\ t : \tau$

i.e. if we can assign a type to it

also called *well-typed* or *typeable*

T_τ set of all well-formed terms of type τ

Type checking

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$fact : int \to int$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\frac{f : int \to int \qquad \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \qquad \dfrac{x : int \qquad \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int}{\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{x : int \quad 1 : int \quad (x \times (f(x-1))) : int}{\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int}}{\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{}{1 : int} \quad \dfrac{x : int \qquad f(x-1) : int}{(x \times (f(x-1))) : int}}{\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int}}{\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{}{1 : int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{f : int \to int \qquad x-1 : int}{f(x-1) : int}}{(x \times (f(x-1))) : int}}{\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int}}{\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{x : int \quad 1 : int}{x - 1 : int}}{f(x-1) : int}}{(x \times (f(x-1))) : int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{}{1 : int}}{\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int}}{\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$\dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{}{1 : int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{\dfrac{\widehat{f} = int \to int}{f : int \to int} \quad \dfrac{\dfrac{\widehat{x} = int}{x : int} \quad \dfrac{}{1 : int}}{x - 1 : int}}{f(x-1) : int}}{(x \times (f(x-1))) : int}}{\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int}}{\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ (x \times (f(x-1))) : int \to int}}{fact : int \to int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int \to int}{f}\ .\lambda\ \underset{int}{x}\ .\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (\quad f \quad (x-1))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{f}\ .\lambda\ \underset{int}{x}\ .\mathbf{if}\ \underset{int}{x}\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (\quad f \quad (x-1))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{\sqcup f}\ .\lambda\ \underset{int}{\sqcup x}\ .\mathbf{if}\ \underset{int}{\sqcup x}\ \mathbf{then}\ \underset{int}{\sqcup 1}\ \mathbf{else}\ x \times (\quad f \quad (x-1))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{f}\ .\lambda\ \underset{int}{x}\ .\mathbf{if}\ \underset{int}{x}\ \mathbf{then}\ \underset{int}{1}\ \mathbf{else}\ \underset{int}{x} \times (\quad f \quad (x-1))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{f}\ .\lambda\ \underset{int}{x}\ .\mathbf{if}\ \underset{int}{x}\ \mathbf{then}\ \underset{int}{1}\ \mathbf{else}\ \underset{int}{x} \times (\ \underset{int\to int}{f}\ \ (x-1))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{f}\ .\lambda\ \underset{int}{x}\ .\mathbf{if}\ \underset{int}{x}\ \mathbf{then}\ \underset{int}{1}\ \mathbf{else}\ \underset{int}{x} \times (\ \underset{int\to int}{f}\ (\underset{int}{x} - 1))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{f}.\lambda\ \underset{int}{x}.\mathbf{if}\ \underset{int}{x}\ \mathbf{then}\ \underset{int}{1}\ \mathbf{else}\ \underset{int}{x} \times (\ \underset{int\to int}{f}\ (\underset{int}{x} - \underset{int}{1}))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underbrace{f}_{int\to int}.\lambda\ \underbrace{x}_{int}.\mathbf{if}\ \underbrace{x}_{int}\ \mathbf{then}\ \underbrace{1}_{int}\ \mathbf{else}\ \underbrace{x}_{int} \times (\ \underbrace{f}_{int\to int}\ (\underbrace{\underbrace{x}_{int} - \underbrace{1}_{int}}_{int}))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underbrace{f}_{int\to int}.\lambda\ \underbrace{x}_{int}.\mathbf{if}\ \underbrace{x}_{int}\ \mathbf{then}\ \underbrace{1}_{int}\ \mathbf{else}\ \underbrace{x}_{int} \times (\underbrace{\underbrace{f}_{int\to int}\ \underbrace{(\underbrace{x}_{int} - \underbrace{1}_{int})}_{int}}_{int})$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underbrace{f}_{int\to int}.\lambda\ \underbrace{x}_{int}.\mathbf{if}\ \underbrace{x}_{int}\ \mathbf{then}\ \underbrace{1}_{int}\ \mathbf{else}\ \underbrace{\underbrace{x}_{int} \times (\underbrace{\underbrace{f}_{int\to int}\ (\underbrace{\underbrace{x}_{int} - \underbrace{1}_{int}}_{int})}_{int})}_{int}}_{int}$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underset{int\to int}{f}.\lambda\ \underset{int}{x}.\mathbf{if}\ \underset{int}{x}\ \mathbf{then}\ \underset{int}{1}\ \mathbf{else}\ \underset{int}{x} \times (\ \underset{int\to int}{f}\ (\underset{int}{x} - \underset{int}{1}))$$

$x - 1$: int ; $f\ (x-1)$: int ; $x \times (f\ (x-1))$: int ; $\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$: int ; $\mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$: int

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underbrace{f}_{int\to int}.\lambda\ \underbrace{x}_{int}.\mathbf{if}\ \underbrace{x}_{int}\ \mathbf{then}\ \underbrace{1}_{int}\ \mathbf{else}\ \underbrace{x}_{int} \times (\ \underbrace{f}_{int\to int}\ (\underbrace{\underbrace{x}_{int} - \underbrace{1}_{int}}_{int}))$$

Example

variables are tagged with (declared) types
we deduce the type of terms by structural recursion

$$fact \triangleq \mathbf{rec}\ f : int \to int.\ \lambda x : int.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

more concisely

$$fact \stackrel{\text{def}}{=} \mathbf{rec}\ \underbrace{f}_{int\to int}.\lambda\ \underbrace{x}_{int}.\mathbf{if}\ \underbrace{x}_{int}\ \mathbf{then}\ \underbrace{1}_{int}\ \mathbf{else}\ \underbrace{x}_{int} \times (\ \underbrace{f}_{int\to int}\ (\underbrace{\underbrace{x}_{int} - \underbrace{1}_{int}}_{int})) \quad : int \to int$$

Type inference

Example

types of variables are not given
type rules are used to derive type constraints (type equations)
whose solutions (via unification) define the principal type

Example

types of variables are not given
type rules are used to derive type constraints (type equations)
whose solutions (via unification) define the principal type

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x + 2)))$$

Example

types of variables are not given
type rules are used to derive type constraints (type equations)
whose solutions (via unification) define the principal type

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

intuitively

Example

types of variables are not given
type rules are used to derive type constraints (type equations)
whose solutions (via unification) define the principal type

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

intuitively

$$t\ 0 \equiv (0, (t\ 2)) \equiv (0, (2, (t\ 4))) \equiv \cdots \equiv (0, (2, (4, \ldots)))$$

sequence of all even numbers

Example

types of variables are not given
type rules are used to derive type constraints (type equations)
whose solutions (via unification) define the principal type

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

intuitively

$$t\ 0 \equiv (0, (t\ 2)) \equiv (0, (2, (t\ 4))) \equiv \cdots \equiv (0, (2, (4, \ldots)))$$

sequence of all even numbers

we can type sequence of integers of fixed length
we have no type for sequences of any/infinite length

Example

types of variables are not given
type rules are used to derive type constraints (type equations)
whose solutions (via unification) define the principal type

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

Haskell

```
Prelude> let p x = (x, p (x+2))

<interactive>:…:5: error:
    • Occurs check: cannot construct the infinite type: b ~ (t, b)
      Expected type: t -> b
        Actual type: t -> (t, b)
    • Relevant bindings include
        p :: t -> b (bound at <interactive>:…:5)
```

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ p.\quad \lambda\ x\ .\ (\,x\,,(\quad p\quad (x + \underset{int}{\square}2)))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ p.\quad \lambda\ x\ .\ (\ x\ ,(\quad p\quad (\underset{int}{x} + \underset{int}{2})))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ p.\quad \lambda \underset{int}{\underline{x}}\ .\ (\underset{int}{\underline{x}}, (\quad p \quad (\underset{int}{\underline{x}} + \underset{int}{\underline{2}})))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ p.\quad \lambda \underset{int}{x}.\ (\underset{int}{x}, (\quad p \quad (\underbrace{\underset{int}{x} + \underset{int}{2}}_{int})))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ p.\quad \lambda \underset{int}{x}\ .\ (\underset{int}{x}, (\underset{int\to\tau_4}{p}\ \underbrace{(\underset{int}{x} + \underset{int}{2})}_{int}))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ \underset{int\to\tau_4}{p}.\quad \lambda\ \underset{int}{x}\ .\ (\underset{int}{x}, (\quad \underset{int\to\tau_4}{p}\quad (\underbrace{\underset{int}{x} + \underset{int}{2}}_{int})))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \mathbf{rec}\ \underbrace{p}_{int\to\tau_4}.\quad \lambda\ \underbrace{x}_{int}\ .\ (\ \underbrace{x}_{int}\ , \underbrace{(\ \underbrace{p}_{int\to\tau_4}\quad \underbrace{(\ \underbrace{x}_{int} + \underbrace{2}_{int}\ )}_{int}}_{\tau_4}))$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \textbf{rec}\ \underbrace{p}_{int \to \tau_4}.\quad \lambda\ \underbrace{x}_{int}\ .\ \underbrace{(\ \underbrace{x}_{int}\ , \underbrace{(\ \underbrace{p}_{int \to \tau_4}\quad \underbrace{(\ \underbrace{x}_{int} + \underbrace{2}_{int}\ )}_{int}\ )}_{\tau_4}\ )}_{int * \tau_4}$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

more concisely

$$t = \textbf{rec}\ \underset{int\to\tau_4}{p}.\quad \lambda\ \underset{int}{x}\ .\ (\underset{int}{x}, (\quad \underset{int\to\tau_4}{p}\quad (\underbrace{\underset{int}{x} + \underset{int}{2}}_{int})))$$

$$\underbrace{\underbrace{\underbrace{(int\to(int*\tau_4)) = (int\to\tau_4)}_{\tau_4 = (int*\tau_4)}}_{int*\tau_4}}_{\tau_4}$$

fail (occur check)

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau \quad \nwarrow_{\widehat{p}=\tau} \quad \lambda x.\ (x, (p\ (x+2))) : \tau$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau \quad \nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau \quad \nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

$$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau \quad \nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

$$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$$

$$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau \quad \nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

$$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$$

$$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$$

$$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$$

Example

$$t \stackrel{\text{def}}{=} \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \mathbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau \quad \nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

$$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$$

$$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$$

$$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$$

$$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

$$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$$

$$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$$

$$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$$

$$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$$

$$\nwarrow_{\tau_5=int}\ x : int,\ 2 : int$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$$

$$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$$

$$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$$

$$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$$

$$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$$

$$\nwarrow_{\tau_5=int}\ x : int,\ 2 : int$$

$$\nwarrow^{*}_{\widehat{x}=int}\ \square$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$

$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$

$\nwarrow_{\tau_5=int}\ x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}\ \square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$

$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$

$\nwarrow_{\tau_5=int}\ x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}\ \square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$

$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$

$\nwarrow_{\tau_5=int}\ x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}\ \square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$$\begin{array}{ll}
\nwarrow_{\widehat{p}=\tau} & \lambda x.\ (x, (p\ (x+2))) : \tau \\
\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1} & (x, (p\ (x+2))) : \tau_2 \\
\nwarrow_{\tau_2=\tau_3*\tau_4} & x : \tau_3,\ (p\ (x+2)) : \tau_4 \\
\nwarrow_{\widehat{x}=\tau_3} & (p\ (x+2)) : \tau_4 \\
\nwarrow & p : \tau_5 \to \tau_4,\ (x+2) : \tau_5 \\
\nwarrow_{\widehat{p}=\tau_5\to\tau_4} & (x+2) : \tau_5 \\
\nwarrow_{\tau_5=int} & x : int,\ 2 : int \\
\nwarrow^{*}_{\widehat{x}=int} & \square
\end{array}$$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$\nwarrow_{\widehat{p}=\tau}$ $\lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}$ $(x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}$ $x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}$ $(p\ (x+2)) : \tau_4$

$\nwarrow$ $p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}$ $(x+2) : \tau_5$

$\nwarrow_{\tau_5=int}$ $x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}$ $\square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$\nwarrow_{\widehat{p}=\tau}$ $\lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}$ $(x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}$ $x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}$ $(p\ (x+2)) : \tau_4$

$\nwarrow$ $p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}$ $(x+2) : \tau_5$

$\nwarrow_{\tau_5=int}$ $x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}$ $\square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$\nwarrow_{\widehat{p}=\tau}$ $\lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}$ $(x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}$ $x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}$ $(p\ (x+2)) : \tau_4$

$\nwarrow$ $p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}$ $(x+2) : \tau_5$

$\nwarrow_{\tau_5=int}$ $x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}$ $\square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$\nwarrow_{\widehat{p}=\tau}$ $\lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}$ $(x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}$ $x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}$ $(p\ (x+2)) : \tau_4$

$\nwarrow$ $p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}$ $(x+2) : \tau_5$

$\nwarrow_{\tau_5=int}$ $x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}$ $\square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$$\left.\begin{array}{l} \widehat{p} = \tau = \tau_1 \to \tau_2 \\ \widehat{p} = \tau_5 \to \tau_4 \end{array}\right\} \begin{array}{l} \tau_1 = \tau_5 = int \\ \tau_2 = \tau_4 \end{array}$$

$\nwarrow_{\widehat{p}=\tau}$ $\lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}$ $(x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}$ $x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}$ $(p\ (x+2)) : \tau_4$

$\nwarrow$ $p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}$ $(x+2) : \tau_5$

$\nwarrow_{\tau_5=int}$ $x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}$ $\square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$$\left.\begin{array}{l} \widehat{p} = \tau = \tau_1 \to \tau_2 \\ \widehat{p} = \tau_5 \to \tau_4 \end{array}\right\} \begin{array}{l} \tau_1 = \tau_5 = int \\ \tau_2 = \tau_4 \end{array}$$

$\lambda x.\ (x, (p\ (x+2))) : \tau$ ↖ $\widehat{p}=\tau$

$(x, (p\ (x+2))) : \tau_2$ ↖ $\tau=\tau_1\to\tau_2$, $\widehat{x}=\tau_1$

$x : \tau_3,\ (p\ (x+2)) : \tau_4$ ↖ $\tau_2=\tau_3*\tau_4$

$(p\ (x+2)) : \tau_4$ ↖ $\widehat{x}=\tau_3$

$p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$ ↖

$(x+2) : \tau_5$ ↖ $\widehat{p}=\tau_5\to\tau_4$

$x : int,\ 2 : int$ ↖ $\tau_5=int$

$\square$ ↖* $\widehat{x}=int$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$$\left.\begin{array}{l} \widehat{p} = \tau = \tau_1 \to \tau_2 \\ \widehat{p} = \tau_5 \to \tau_4 \end{array}\right\} \begin{array}{l} \tau_1 = \tau_5 = int \\ \tau_2 = \tau_4 \end{array}$$

$\nwarrow_{\widehat{p}=\tau}\ \lambda x.\ (x, (p\ (x+2))) : \tau$

$\nwarrow_{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1}\ (x, (p\ (x+2))) : \tau_2$

$\nwarrow_{\tau_2=\tau_3*\tau_4}\ x : \tau_3,\ (p\ (x+2)) : \tau_4$

$\nwarrow_{\widehat{x}=\tau_3}\ (p\ (x+2)) : \tau_4$

$\nwarrow\ p : \tau_5 \to \tau_4,\ (x+2) : \tau_5$

$\nwarrow_{\widehat{p}=\tau_5\to\tau_4}\ (x+2) : \tau_5$

$\nwarrow_{\tau_5=int}\ x : int,\ 2 : int$

$\nwarrow^{*}_{\widehat{x}=int}\ \square$

Example

$$t \stackrel{\text{def}}{=} \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2)))$$

$$t = \textbf{rec}\ p.\ \lambda x.\ (x, (p\ (x+2))) : \tau$$

$$\left.\begin{array}{l} \widehat{x} = \tau_1 \\ \widehat{x} = \tau_3 \\ \widehat{x} = int \end{array}\right\} \tau_1 = \tau_3 = int$$

$$\left.\begin{array}{l} \widehat{p} = \tau = \tau_1 \to \tau_2 \\ \widehat{p} = \tau_5 \to \tau_4 \end{array}\right\} \begin{array}{l} \tau_1 = \tau_5 = int \\ \tau_2 = \tau_4 \end{array}$$

$$\left.\begin{array}{l} \tau_2 = \tau_4 \\ \tau_2 = \tau_3 * \tau_4 \end{array}\right\} \text{fail! (occur check)}$$

$$\begin{array}{ll}
\nwarrow^{\widehat{p}=\tau} & \lambda x.\ (x, (p\ (x+2))) : \tau \\
\nwarrow^{\tau=\tau_1\to\tau_2,\ \widehat{x}=\tau_1} & (x, (p\ (x+2))) : \tau_2 \\
\nwarrow^{\tau_2=\tau_3*\tau_4} & x : \tau_3,\ (p\ (x+2)) : \tau_4 \\
\nwarrow^{\widehat{x}=\tau_3} & (p\ (x+2)) : \tau_4 \\
\nwarrow & p : \tau_5 \to \tau_4,\ (x+2) : \tau_5 \\
\nwarrow^{\widehat{p}=\tau_5\to\tau_4} & (x+2) : \tau_5 \\
\nwarrow^{\tau_5=int} & x : int,\ 2 : int \\
\nwarrow^{*\ \widehat{x}=int} & \square
\end{array}$$

HOFL semantics

But first... capture-avoiding substitution

Free variables

$$\mathrm{fv}(n) \stackrel{\text{def}}{=} \varnothing$$

$$\mathrm{fv}(x) \stackrel{\text{def}}{=} \{x\}$$

$$\mathrm{fv}(t_0 \ \mathrm{op}\ t_1) \stackrel{\text{def}}{=} \mathrm{fv}(t_0) \cup \mathrm{fv}(t_1)$$

$$\mathrm{fv}(\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1) \stackrel{\text{def}}{=} \mathrm{fv}(t) \cup \mathrm{fv}(t_0) \cup \mathrm{fv}(t_1)$$

$$\mathrm{fv}((t_0, t_1)) \stackrel{\text{def}}{=} \mathrm{fv}(t_0) \cup \mathrm{fv}(t_1)$$

$$\mathrm{fv}(\textbf{fst}(t)) \stackrel{\text{def}}{=} \mathrm{fv}(t)$$

$$\mathrm{fv}(\textbf{snd}(t)) \stackrel{\text{def}}{=} \mathrm{fv}(t)$$

$$\mathrm{fv}(\lambda x.\ t) \stackrel{\text{def}}{=} \mathrm{fv}(t) \setminus \{x\}$$

$$\mathrm{fv}((t_0\ t_1)) \stackrel{\text{def}}{=} \mathrm{fv}(t_0) \cup \mathrm{fv}(t_1)$$

$$\mathrm{fv}(\textbf{rec } x.\ t) \stackrel{\text{def}}{=} \mathrm{fv}(t) \setminus \{x\}$$

Substitutions

$$n[^t/_x] = n$$

$$y[^t/_x] \stackrel{\text{def}}{=} \begin{cases} t & \text{if } y = x \\ y & \text{if } y \neq x \end{cases}$$

$$(t_0 \text{ op } t_1)[^t/_x] \stackrel{\text{def}}{=} t_0[^t/_x] \text{ op } t_1[^t/_x] \quad \text{with op} \in \{+, -, \times\}$$

$$(\textbf{if } t' \textbf{ then } t_0 \textbf{ else } t_1)[^t/_x] \stackrel{\text{def}}{=} \textbf{if } t'[^t/_x] \textbf{ then } t_0[^t/_x] \textbf{ else } t_1[^t/_x]$$

$$(t_0, t_1)[^t/_x] \stackrel{\text{def}}{=} (t_0[^t/_x], t_1[^t/_x])$$

$$\textbf{fst}(t')[^t/_x] \stackrel{\text{def}}{=} \textbf{fst}(t'[^t/_x])$$

$$\textbf{snd}(t')[^t/_x] \stackrel{\text{def}}{=} \textbf{snd}(t'[^t/_x])$$

$$(t_0\ t_1)[^t/_x] \stackrel{\text{def}}{=} (t_0[^t/_x]\ t_1[^t/_x])$$

$$(\lambda y.\ t')[^t/_x] \stackrel{\text{def}}{=} \lambda z.\ (\ t'[^z/_y][^t/_x]\ ) \quad \text{for } z \notin \text{fv}(\lambda y.\ t') \cup \text{fv}(t) \cup \{x\}$$

$$(\textbf{rec } y.\ t')[^t/_x] \stackrel{\text{def}}{=} \textbf{rec } z.\ (t'[^z/_y][^t/_x]) \quad \text{for } z \notin \text{fv}(\textbf{rec } y.\ t') \cup \text{fv}(t) \cup \{x\}$$

Types are respected

TH. $x_0 : \tau_0$, $t_0 : \tau_0$ $\quad t : \tau \quad \Rightarrow \quad t[{}^{t_0}/_{x_0}] : \tau$

proof omitted
(by structural induction
of the stronger assertion $t[{}^{\widetilde{t}}/_{\widetilde{x}}] : \tau$)

Big-step operational semantics

Statements

Big step operational semantics

Statements

$$t \to c$$

Big step operational semantics

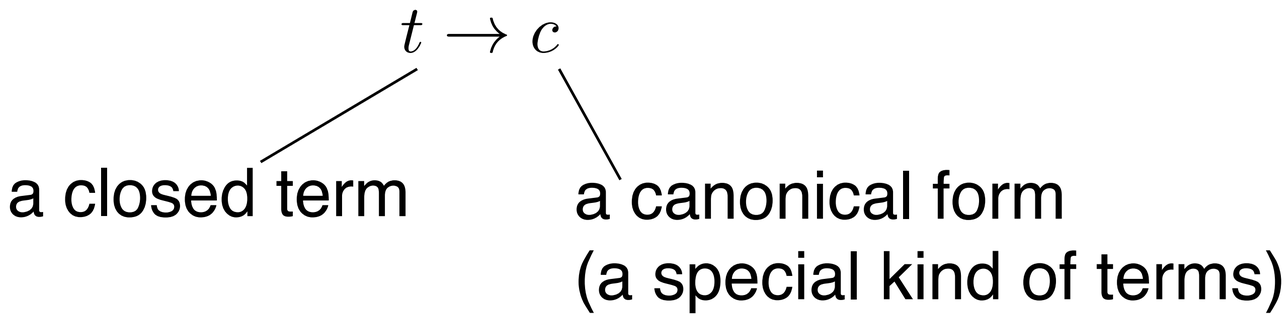
Statements

$$t \to c$$

a closed term

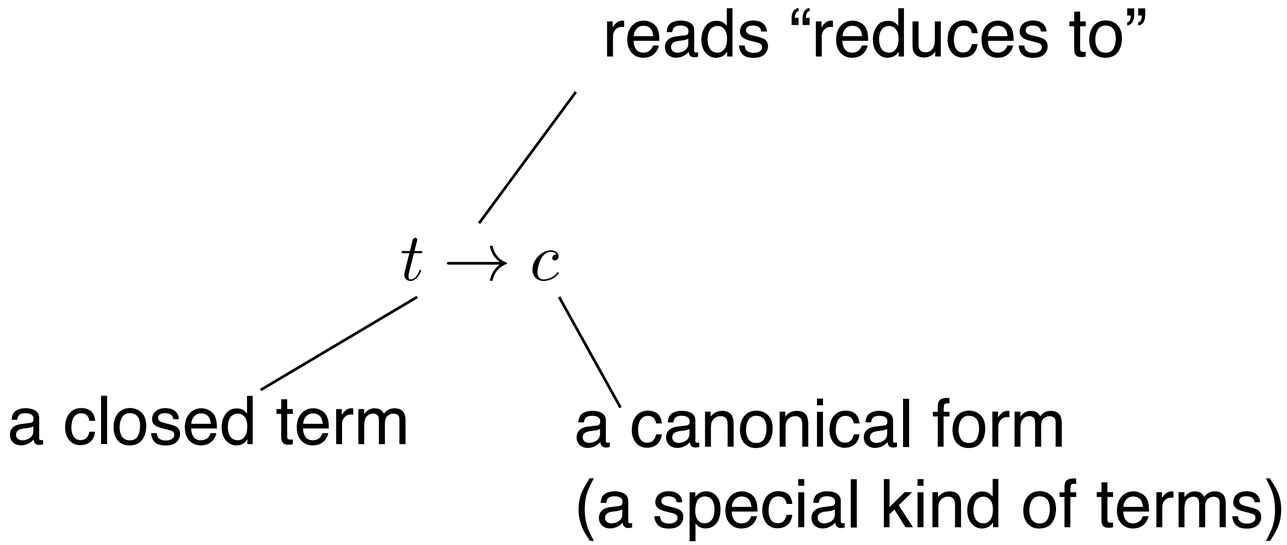
Big step operational semantics

Statements



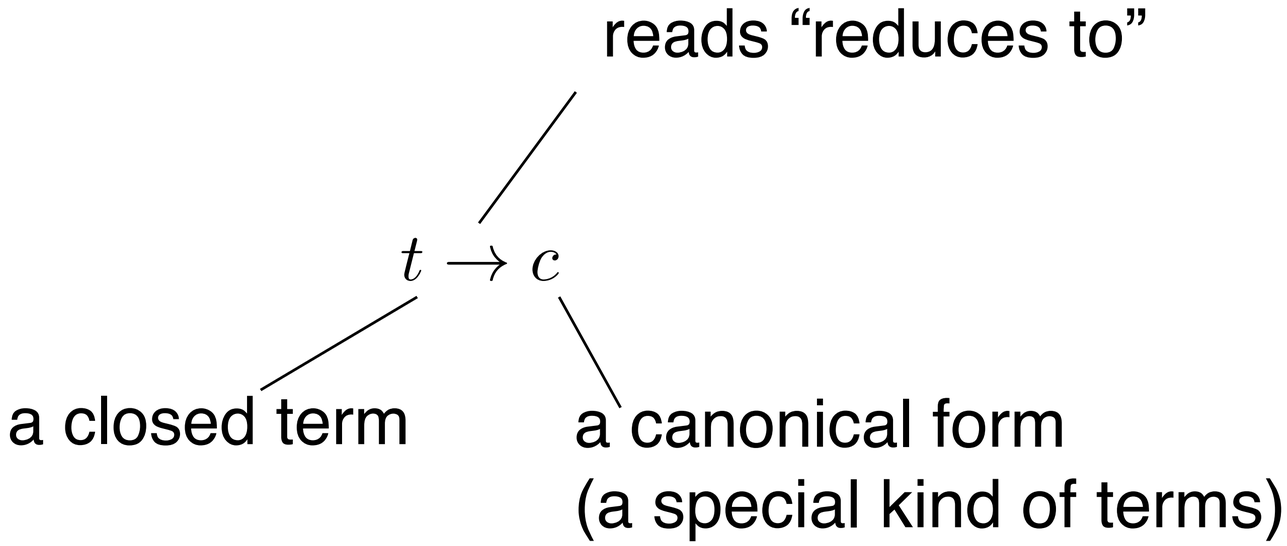
Big step operational semantics

Statements



Big step operational semantics

Statements



Big step operational semantics

computation of canonical form
(by term manipulation)

Canonical forms

set of canonical forms $C_\tau \subseteq T_\tau$
with type τ

Canonical forms

set of canonical forms $C_\tau \subseteq T_\tau$
with type τ

$$\frac{}{n \in C_{int}}$$

Canonical forms

set of canonical forms $C_\tau \subseteq T_\tau$
with type τ

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}}$$

Canonical forms

set of canonical forms $C_\tau \subseteq T_\tau$
with type τ

(laziness)
not required to be
in canonical forms

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}}$$

Canonical forms

set of canonical forms $C_\tau \subseteq T_\tau$
with type τ

(laziness)
not required to be
in canonical forms

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

Canonical forms

set of canonical forms $C_\tau \subseteq T_\tau$
with type τ

(laziness)
not required to be
in canonical forms

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}}$$

t not necessarily
a closed term

$$\frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$$1 + 2 \times 3$$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

$(1, 2)$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

$(1, 2)$ 

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

$(1, 2)$ 

$(1 + 2, 2 - 1)$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

$(1, 2)$ 

$(1 + 2, 2 - 1)$ 

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

$(1, 2)$ 

$(1 + 2, 2 - 1)$ 

$\mathbf{fst}(1, 2)$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ 

$(1, 2)$ 

$(1 + 2, 2 - 1)$ 

$\mathbf{fst}(1, 2)$ 

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

$\lambda x.\ 1$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

$\lambda x.\ 1$ ✓

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

$\lambda x.\ 1$ ✓

$\lambda x.\ 1 + 2 \times 3$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

$\lambda x.\ 1$ ✓

$\lambda x.\ 1 + 2 \times 3$ ✓

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

$\lambda x.\ 1$ ✓

$\lambda x.\ 1 + 2 \times 3$ ✓

$\lambda x.\ \mathbf{fst}(1, 2)$

Canonical forms?

$$\frac{}{n \in C_{int}} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \in C_{\tau_0 * \tau_1}} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.\, t \text{ closed}}{\lambda x.\, t \in C_{\tau_0 \to \tau_1}}$$

$1 + 2 \times 3$ ✗

$(1, 2)$ ✓

$(1 + 2, 2 - 1)$ ✓

$\mathbf{fst}(1, 2)$ ✗

$\mathbf{if}\ 0\ \mathbf{then}\ 0\ \mathbf{else}\ 0$ ✗

$\lambda x.\ 1$ ✓

$\lambda x.\ 1 + 2 \times 3$ ✓

$\lambda x.\ \mathbf{fst}(1, 2)$ ✓

Operational semantics: axioms

$$\frac{c \in C_\tau}{c \to c}$$

Operational semantics: axioms

$$\frac{c \in C_\tau}{c \to c}$$

i.e., expanding the various cases

Operational semantics: axioms

$$\frac{c \in C_\tau}{c \to c}$$

i.e., expanding the various cases

$$\frac{}{n \to n}$$

Operational semantics: axioms

$$\frac{c \in C_\tau}{c \to c}$$

i.e., expanding the various cases

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)}$$

Operational semantics: axioms

$$\frac{c \in C_\tau}{c \to c}$$

i.e., expanding the various cases

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

Operational semantics: axioms

$$\frac{c \in C_\tau}{c \to c}$$

i.e., expanding the various cases

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

integers, pairs and abstractions
are already in canonical form

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$t_0 \text{ op } t_1 \to$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0}$$

$$\frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \textbf{fst}(t) \to$$

$$\frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1} \qquad \frac{t \to (t_0, t_1) \quad t_1 \to c_1}{\textbf{snd}(t) \to c_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\textbf{rec } x.\, t \to \qquad \frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1} \qquad \frac{t \to (t_0, t_1) \quad t_1 \to c_1}{\textbf{snd}(t) \to c_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\frac{t\left[{}^{\textbf{rec } x.\, t}/_{x}\right] \to c}{\textbf{rec } x.\, t \to c} \qquad \frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1} \qquad \frac{t \to (t_0, t_1) \quad t_1 \to c_1}{\textbf{snd}(t) \to c_1}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\frac{t\left[{}^{\textbf{rec } x.\, t}/_{x}\right] \to c}{\textbf{rec } x.\, t \to c} \qquad \frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1} \qquad \frac{t \to (t_0, t_1) \quad t_1 \to c_1}{\textbf{snd}(t) \to c_1}$$

$$(t_1\ t_0) \to$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\frac{t\left[{}^{\textbf{rec } x.\, t}/_{x}\right] \to c}{\textbf{rec } x.\, t \to c} \qquad \frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1} \qquad \frac{t \to (t_0, t_1) \quad t_1 \to c_1}{\textbf{snd}(t) \to c_1}$$

$$\frac{t_1 \to \lambda x.\, t_1' \quad t_1'\left[{}^{t_0}/_{x}\right] \to c}{(t_1\ t_0) \to c}$$

Lazy op semantics

$$\frac{}{n \to n} \qquad \frac{t_0 : \tau_0 \quad t_1 : \tau_1 \quad t_0, t_1 \text{ closed}}{(t_0, t_1) \to (t_0, t_1)} \qquad \frac{\lambda x.\, t : \tau_0 \to \tau_1 \quad \lambda x.t \text{ closed}}{\lambda x.\, t \to \lambda x.\, t}$$

$$\frac{t_0 \to n_0 \quad t_1 \to n_1}{t_0 \text{ op } t_1 \to n_0 \underline{\text{op}}\, n_1} \qquad \frac{t \to 0 \quad t_0 \to c_0}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_0} \qquad \frac{t \to (t_0, t_1) \quad t_0 \to c_0}{\textbf{fst}(t) \to c_0}$$

$$\frac{t\left[{}^{\textbf{rec } x.\, t}/_{x}\right] \to c}{\textbf{rec } x.\, t \to c} \qquad \frac{t \to n \quad n \neq 0 \quad t_1 \to c_1}{\textbf{if } t \textbf{ then } t_0 \textbf{ else } t_1 \to c_1} \qquad \frac{t \to (t_0, t_1) \quad t_1 \to c_1}{\textbf{snd}(t) \to c_1}$$

$$\frac{t_1 \to \lambda x.\, t_1' \quad t_1'\left[{}^{t_0}/_{x}\right] \to c}{(t_1\ t_0) \to c} \text{ (lazy)}$$

Example

$$t \triangleq \lambda x.\ \mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \ \nwarrow\ t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

Example

$$t \triangleq \lambda x.\ \mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \ \nwarrow\ t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\mathbf{if}\ \ldots}\ (\mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

Example

$$t \triangleq \lambda x.\ \mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \ \nwarrow\ t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\mathbf{if}\ \ldots}\ (\mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

$$= \mathbf{if}\ \mathbf{fst}(1,2)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(1,2) \to c$$

Example

$$t \triangleq \lambda x.\ \mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \ \nwarrow\ t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\mathbf{if}\ \ldots}\ (\mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

$$= \mathbf{if}\ \mathbf{fst}(1,2)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(1,2) \to c$$

$$\nwarrow\ \mathbf{fst}(1,2) \to n\ ,\ n \neq 0\ ,\ \mathbf{snd}(1,2) \to c$$

Example

$$t \triangleq \lambda x.\ \textbf{if}\ \textbf{fst}(x)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \nwarrow t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\textbf{if}\ \ldots} (\textbf{if}\ \textbf{fst}(x)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

$$= \textbf{if}\ \textbf{fst}(1,2)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ \textbf{fst}(1,2) \to n\ ,\ n \neq 0\ ,\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ (1,2) \to (n_1, n_2)\ ,\ n_1 \to n\ ,\ n \neq 0\ ,\ \textbf{snd}(1,2) \to c$$

Example

$$t \triangleq \lambda x.\ \textbf{if}\ \textbf{fst}(x)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \nwarrow t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\textbf{if}\ \ldots} (\textbf{if}\ \textbf{fst}(x)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

$$= \textbf{if}\ \textbf{fst}(1,2)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ \textbf{fst}(1,2) \to n\ ,\ n \neq 0\ ,\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ (1,2) \to (n_1, n_2)\ ,\ n_1 \to n\ ,\ n \neq 0\ ,\ \textbf{snd}(1,2) \to c$$

$$\nwarrow^{*}_{n_1=1, n_2=2, n=1}\ \textbf{snd}(1,2) \to c$$

Example

$$t \triangleq \lambda x.\ \textbf{if}\ \textbf{fst}(x)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \nwarrow t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\textbf{if}\ \ldots} (\textbf{if}\ \textbf{fst}(x)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

$$= \textbf{if}\ \textbf{fst}(1,2)\ \textbf{then}\ 1\ \textbf{else}\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ \textbf{fst}(1,2) \to n\ ,\ n \neq 0\ ,\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ (1,2) \to (n_1, n_2)\ ,\ n_1 \to n\ ,\ n \neq 0\ ,\ \textbf{snd}(1,2) \to c$$

$$\nwarrow^{*}_{n_1=1,n_2=2,n=1}\ \textbf{snd}(1,2) \to c$$

$$\nwarrow\ (1,2) \to (n_3, n_4)\ ,\ n_4 \to c$$

Example

$$t \triangleq \lambda x.\ \mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x) : (int * int) \to int$$

$$t\ (1,2) \to c \ \nwarrow\ t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$$

$$\nwarrow_{x'=x,\ t'=\mathbf{if}\ \ldots} (\mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$$

$$= \mathbf{if}\ \mathbf{fst}(1,2)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(1,2) \to c$$

$$\nwarrow\ \mathbf{fst}(1,2) \to n\ ,\ n \neq 0\ ,\ \mathbf{snd}(1,2) \to c$$

$$\nwarrow\ (1,2) \to (n_1, n_2)\ ,\ n_1 \to n\ ,\ n \neq 0\ ,\ \mathbf{snd}(1,2) \to c$$

$$\nwarrow^{*}_{n_1=1,n_2=2,n=1}\ \mathbf{snd}(1,2) \to c$$

$$\nwarrow\ (1,2) \to (n_3, n_4)\ ,\ n_4 \to c$$

$$\nwarrow^{*}_{n_3=1,n_4=2,c=2} \square$$

Example

$t \triangleq \lambda x.\ \mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x) : (int * int) \to int$

$t\ (1,2) \to c \nwarrow t \to \lambda x'.\ t'\ ,\ t'[{}^{(1,2)}/_{x'}] \to c$

$\nwarrow_{x'=x,\ t'=\mathbf{if}\ \ldots} (\mathbf{if}\ \mathbf{fst}(x)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(x))[{}^{(1,2)}/_{x}] \to c$

$= \mathbf{if}\ \mathbf{fst}(1,2)\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{snd}(1,2) \to c$

$\nwarrow \mathbf{fst}(1,2) \to n\ ,\ n \neq 0\ ,\ \mathbf{snd}(1,2) \to c$

$\nwarrow (1,2) \to (n_1, n_2)\ ,\ n_1 \to n\ ,\ n \neq 0\ ,\ \mathbf{snd}(1,2) \to c$

$\nwarrow^{*}_{n_1=1,n_2=2,n=1} \mathbf{snd}(1,2) \to c$

$\nwarrow (1,2) \to (n_3, n_4)\ ,\ n_4 \to c$

$\nwarrow^{*}_{n_3=1,n_4=2,c=2} \square$ $\qquad t\ (1,2) \to 2$

Example

$$t \triangleq \mathbf{rec}\ x.\ x$$

Example

$$t \triangleq \mathbf{rec}\ x.\ \underbrace{x}_{\tau}$$

Example

$$t \triangleq \mathbf{rec}\ \underbrace{x}_{\tau}.\ \underbrace{x}_{\tau}$$

Example

$$t \triangleq \underbrace{\mathbf{rec}\ \underbrace{x}_{\tau}.\ \underbrace{x}_{\tau}}_{\tau}$$

Example

$$t \triangleq \underbrace{\mathbf{rec}\ \underbrace{x}_{\tau}.\ \underbrace{x}_{\tau}}_{\tau} : \tau$$

Example

$$t \triangleq \underbrace{\mathbf{rec}\ \underbrace{x}_{\tau}.\ \underbrace{x}_{\tau}}_{\tau} : \tau$$

$$\mathbf{rec}\ x.\ x \to c \ \nwarrow\ x\left[{}^{\mathbf{rec}\ x.\ x}/_{x}\right] \to c$$

Example

$$t \triangleq \underbrace{\mathbf{rec}\ \underbrace{x}_{\tau}.\ \underbrace{x}_{\tau}}_{\tau} : \tau$$

$$\mathbf{rec}\ x.\ x \to c \nwarrow x\left[{}^{\mathbf{rec}\ x.\ x}/_{x}\right] \to c$$

$$= \mathbf{rec}\ x.\ x \to c$$

Example

$$t \triangleq \underbrace{\mathbf{rec}\ \underbrace{x}_{\tau}.\ \underbrace{x}_{\tau}}_{\tau} : \tau$$

$$\mathbf{rec}\ x.\ x \to c \nwarrow x\left[{}^{\mathbf{rec}\ x.\ x}/_{x}\right] \to c$$

$$= \mathbf{rec}\ x.\ x \to c$$

same goal from which we started
no other option to explore:
divergence!

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$fact \to c \nwarrow (\lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f(x-1)))[^{fact}/_{f}] \to c$$

Example

$$fact \triangleq \textbf{rec}\ f.\ \lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (f\ (x-1))$$

$$fact \to c \nwarrow (\lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (\overbrace{f(x-1)}^{fact}))[{}^{fact}/_{f}] \to c$$

$$= \lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (\overbrace{(\textbf{rec} f.\ \ldots)}(x-1)) \to c$$

Example

$$fact \triangleq \textbf{rec}\ f.\ \lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (f\ (x-1))$$

$$fact \to c \nwarrow (\lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (\overbrace{f(x-1)}^{fact}))[^{fact}/_{f}] \to c$$

$$= \lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (\overbrace{(\textbf{rec} f.\ \ldots)}(x-1)) \to c$$

$$\nwarrow_{c=\lambda x.\ \textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ x \times (fact(x-1))}\ \square$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \ \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[^1/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^1/_x] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[{}^{1-1}/_{x''}] \to n_2$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[{}^{1-1}/_{x''}] \to n_2$$

$$\nwarrow^{*}_{x''=x,t''=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1-1}/_{x}] \to n_2$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[{}^{1-1}/_{x''}] \to n_2$$

$$\nwarrow^{*}_{x''=x,t''=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1-1}/_{x}] \to n_2$$

$$= \mathbf{if}\ 1-1\ \mathbf{then}\ 1\ \mathbf{else}\ (1-1) \times (fact\ ((1-1)-1)) \to n_2$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[^{1-1}/_{x''}] \to n_2$$

$$\nwarrow^{*}_{x''=x,t''=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^{1-1}/_{x}] \to n_2$$

$$= \mathbf{if}\ 1-1\ \mathbf{then}\ 1\ \mathbf{else}\ (1-1) \times (fact\ ((1-1)-1)) \to n_2$$

$$\nwarrow\ 1-1 \to 0\ ,\ 1 \to n_2$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[^{1-1}/_{x''}] \to n_2$$

$$\nwarrow^{*}_{x''=x,t''=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^{1-1}/_{x}] \to n_2$$

$$= \mathbf{if}\ 1-1\ \mathbf{then}\ 1\ \mathbf{else}\ (1-1) \times (fact\ ((1-1)-1)) \to n_2$$

$$\nwarrow\ 1-1 \to 0\ ,\ 1 \to n_2$$

$$\nwarrow^{*}_{n_2=1}\ \square$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[^{1-1}/_{x''}] \to n_2$$

$$\nwarrow^{*}_{x''=x,t''=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[^{1-1}/_{x}] \to n_2$$

$$= \mathbf{if}\ 1-1\ \mathbf{then}\ 1\ \mathbf{else}\ (1-1) \times (fact\ ((1-1)-1)) \to n_2$$

$$\nwarrow\ 1-1 \to 0\ ,\ 1 \to n_2$$

$$\nwarrow^{*}_{n_2=1}\ \square$$

$$c = n_1 \underline{\times} n_2 = 1 \underline{\times} 1 = 1$$

Example

$$fact \triangleq \mathbf{rec}\ f.\ \lambda x.\ \mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (f\ (x-1))$$

$$(fact\ 1) \to c \nwarrow\ fact \to \lambda x'.\ t'\ ,\ t'[{}^{1}/_{x'}] \to c$$

$$\nwarrow^{*}_{x'=x,t'=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1}/_{x}] \to c$$

$$= \mathbf{if}\ 1\ \mathbf{then}\ 1\ \mathbf{else}\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow\ 1 \to n\ ,\ n \neq 0\ ,\ 1 \times (fact\ (1-1)) \to c$$

$$\nwarrow^{*}_{n=1,c=n_1 \underline{\times} n_2}\ 1 \to n_1\ ,\ (fact\ (1-1)) \to n_2$$

laziness evident here

$$\nwarrow_{n_1=1}\ fact \to \lambda x''.\ t''\ ,\ t''[{}^{1-1}/_{x''}] \to n_2$$

$$\nwarrow^{*}_{x''=x,t''=\mathbf{if}\ldots}\ (\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ x \times (fact\ (x-1)))[{}^{1-1}/_{x}] \to n_2$$

$$= \mathbf{if}\ 1-1\ \mathbf{then}\ 1\ \mathbf{else}\ (1-1) \times (fact\ ((1-1)-1)) \to n_2$$

$$\nwarrow\ 1-1 \to 0\ ,\ 1 \to n_2$$

$$\nwarrow^{*}_{n_2=1}\ \square$$

$$c = n_1 \underline{\times} n_2 = 1 \underline{\times} 1 = 1$$

Properties of the semantics

Termination

termination?

Termination

termination? $\forall t.\ \exists c.\ t \to c?$

Termination

termination? $\forall t.\ \exists c.\ t \to c?$ 

Termination

termination? $\forall t.\ \exists c.\ t \to c?$ ❌ $\mathbf{rec}\ x.\ x$

Determinacy?

determinacy?

Determinacy?

determinacy? $\forall t.\ \forall c_1, c_2.\ t \to c_1 \wedge t \to c_2 \Rightarrow c_1 = c_2\,?$

Determinacy?

determinacy? $\forall t.\ \forall c_1, c_2.\ t \to c_1 \wedge t \to c_2 \Rightarrow c_1 = c_2\,?$ 

Determinacy?

determinacy? $\forall t.\ \forall c_1, c_2.\ t \to c_1 \wedge t \to c_2 \Rightarrow c_1 = c_2\,?$ 

$$P(t \to c) \triangleq \forall c_1.\ t \to c_1 \Rightarrow c_1 = c$$

Determinacy?

determinacy? $\forall t.\ \forall c_1, c_2.\ t \to c_1 \wedge t \to c_2 \Rightarrow c_1 = c_2\,?$ 

$$P(t \to c) \triangleq \forall c_1.\ t \to c_1 \Rightarrow c_1 = c$$

by rule induction (try by yourself)

Subject reduction

subject reduction?

Subject reduction

subject reduction? $\forall t.\ \forall c.\ \forall \tau.\ t \to c \wedge t : \tau \Rightarrow c : \tau\ ?$

Subject reduction

subject reduction? $\forall t.\ \forall c.\ \forall \tau.\ t \to c \wedge t : \tau \Rightarrow c : \tau\ ?$ 

Subject reduction

(statically assigned types do not change at runtime)

subject reduction? $\forall t.\ \forall c.\ \forall \tau.\ t \to c \wedge t : \tau \Rightarrow c : \tau\ ?$ 

Subject reduction

(statically assigned types do not change at runtime)

subject reduction? $\forall t.\ \forall c.\ \forall \tau.\ t \to c \wedge t : \tau \Rightarrow c : \tau$?

$$P(t \to c) \triangleq \forall \tau.\ t : \tau \Rightarrow c : \tau$$

Subject reduction

(statically assigned types do not change at runtime)

subject reduction? $\forall t.\ \forall c.\ \forall \tau.\ t \to c \wedge t : \tau \Rightarrow c : \tau$?

$$P(t \to c) \triangleq \forall \tau.\ t : \tau \Rightarrow c : \tau$$

by rule induction (try by yourself)

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence?

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence? ❌

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence? ❌

$$2 \equiv_{\mathrm{op}} 1 + 1$$

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence? ❌

$$2 \equiv_{\mathrm{op}} 1 + 1$$

$$\lambda x.\ 2 \not\equiv_{\mathrm{op}} \lambda x.\ 1 + 1$$

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence? ❌

$$2 \equiv_{\mathrm{op}} 1 + 1$$

$$\lambda x.\ 2 \not\equiv_{\mathrm{op}} \lambda x.\ 1 + 1$$

$$\lambda x.\ 2\ ,\ \lambda x.\ 1 + 1 \in C_{\tau \to int}$$

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence? ❌

$$2 \equiv_{\mathrm{op}} 1 + 1$$

$$\lambda x.\ 2 \not\equiv_{\mathrm{op}} \lambda x.\ 1 + 1$$

$$\lambda x.\ 2\ ,\ \lambda x.\ 1 + 1 \in C_{\tau \to int}$$

$$\lambda x.\ 2 \to \lambda x.\ 2$$

Congruence?

$$t_1 \equiv_{\mathrm{op}} t_2 \quad \text{iff} \quad \forall c.\ (t_1 \to c \Leftrightarrow t_2 \to c)$$

is it a congruence? ❌

$$2 \equiv_{\mathrm{op}} 1 + 1$$

$$\lambda x.\ 2 \not\equiv_{\mathrm{op}} \lambda x.\ 1 + 1$$

$$\lambda x.\ 2\ ,\ \lambda x.\ 1 + 1 \in C_{\tau \to int}$$

$$\lambda x.\ 2 \to \lambda x.\ 2$$

$$\lambda x.\ 1 + 1 \to \lambda x.\ 1 + 1$$

Lazy vs Eager semantics

Lazy vs Eager

$$\frac{t_1 \to \lambda x.\, t_1' \quad t_1'\left[{}^{t_0}/_{x}\right] \to c}{(t_1\ t_0) \to c} \quad \text{(lazy)}$$

$$\frac{t_1 \to \lambda x.\, t_1' \quad t_0 \to c_0 \quad t_1'\left[{}^{c_0}/_{x}\right] \to c}{(t_1\ t_0) \to c} \quad \text{(eager)}$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int$$

$$\frac{x : \tau \qquad y : \tau}{}$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int$$

$$x : \tau$$
$$y : \tau$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'[^{\mathbf{rec}\ y.\ y}/_{x'}] \to c$$

lazy

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int$$

$$x : \tau$$
$$y : \tau$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[{}^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1}\ 1\left[{}^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int$$

$$x : \tau$$
$$y : \tau$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[{}^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1}\ 1\left[{}^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

$$= 1 \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int \qquad \begin{array}{l} x : \tau \\ y : \tau \end{array}$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[{}^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1}\ 1\left[{}^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

$$= 1 \to c$$

$$\nwarrow_{c=1}\ \square$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int \qquad \begin{array}{l} x : \tau \\ y : \tau \end{array}$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[{}^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1}\ 1\left[{}^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

$$= 1 \to c$$

$$\nwarrow_{c=1}\ \square$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ \mathbf{rec}\ y.\ y \to c'\ ,\ t'\left[{}^{c'}/_{x'}\right] \to c$$

eager

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int \qquad \begin{array}{l} x : \tau \\ y : \tau \end{array}$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1}\ 1\left[^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

$$= 1 \to c$$

$$\nwarrow_{c=1}\ \square$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ \mathbf{rec}\ y.\ y \to c'\ ,\ t'\left[^{c'}/_{x'}\right] \to c$$

eager

$$\nwarrow_{x'=x,\ t'=1}\ \mathbf{rec}\ y.\ y \to c'\ ,\ 1\left[^{c'}/_{x}\right] \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int \qquad \begin{array}{l} x : \tau \\ y : \tau \end{array}$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[{}^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1}\ 1\left[{}^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

$$= 1 \to c$$

$$\nwarrow_{c=1}\ \square$$

$$t \to c \ \nwarrow\ \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ \mathbf{rec}\ y.\ y \to c'\ ,\ t'\left[{}^{c'}/_{x'}\right] \to c$$

eager

$$\nwarrow_{x'=x,\ t'=1}\ \mathbf{rec}\ y.\ y \to c'\ ,\ 1\left[{}^{c'}/_{x}\right] \to c$$

$$\nwarrow\ \mathbf{rec}\ y.\ y \to c'\ ,\ 1\left[{}^{c'}/_{x}\right] \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ 1)\ (\mathbf{rec}\ y.\ y) : int \qquad \begin{array}{l} x : \tau \\ y : \tau \end{array}$$

$$t \to c \nwarrow \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ t'\left[{}^{\mathbf{rec}\ y.\ y}/_{x'}\right] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=1} \quad 1\left[{}^{\mathbf{rec}\ y.\ y}/_{x}\right] \to c$$

$$= 1 \to c$$

$$\nwarrow_{c=1} \square$$

$$t \to c \nwarrow \lambda x.\ 1 \to \lambda x'.\ t'\ ,\ \mathbf{rec}\ y.\ y \to c'\ ,\ t'\left[{}^{c'}/_{x'}\right] \to c$$

eager

$$\nwarrow_{x'=x,\ t'=1} \quad \mathbf{rec}\ y.\ y \to c'\ ,\ 1\left[{}^{c'}/_{x}\right] \to c$$

$$\nwarrow \mathbf{rec}\ y.\ y \to c'\ ,\ 1\left[{}^{c'}/_{x}\right] \to c$$

divergence!

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$

$$= (1 \times 2) + (1 \times 2) \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$

$$= (1 \times 2) + (1 \times 2) \to c$$

$$\nwarrow_{c=c_1 \pm c_2}\ (1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$

$= (1 \times 2) + (1 \times 2) \to c$

$\nwarrow_{c=c_1 \pm c_2}\ (1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2$

$\nwarrow^{*}_{c_1=2, c_2=2}\ \square$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$

$$= (1 \times 2) + (1 \times 2) \to c$$

$$\nwarrow_{c=c_1 \underline{+} c_2}\ (1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2$$

$$\nwarrow^{*}_{c_1=2, c_2=2}\ \square \qquad\qquad c = c_1 \underline{+} c_2 = 2 \underline{+} 2 = 4$$

Lazy vs Eager

$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$

$t \to c \ \nwarrow \ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x} \ (x + x)[^{1\times 2}/_{x}] \to c$

$= (1 \times 2) + (1 \times 2) \to c$ evaluated twice

$\nwarrow_{c=c_1 \underline{+} c_2} \boxed{(1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2}$

$\nwarrow^{*}_{c_1=2, c_2=2} \ \square \qquad c = c_1 \underline{+} c_2 = 2 \underline{+} 2 = 4$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$$

$$= (1 \times 2) + (1 \times 2) \to c$$

evaluated twice

$$\nwarrow_{c=c_1 \underline{+} c_2} \boxed{(1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2}$$

$$\nwarrow^{*}_{c_1=2, c_2=2} \square$$

$$c = c_1 \underline{+} c_2 = 2 \underline{+} 2 = 4$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ 1 \times 2 \to c'\ ,\ t'[^{c'}/_{x'}] \to c$$

eager

Lazy vs Eager

$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$

$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$

lazy $\quad \nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$

$= (1 \times 2) + (1 \times 2) \to c$ evaluated twice

$\nwarrow_{c=c_1 \underline{+} c_2} \boxed{(1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2}$

$\nwarrow^{*}_{c_1=2, c_2=2} \square \qquad c = c_1 \underline{+} c_2 = 2 \underline{+} 2 = 4$

$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ 1 \times 2 \to c'\ ,\ t'[^{c'}/_{x'}] \to c$

eager $\quad \nwarrow_{x'=x,\ t'=x+x}\ 1 \times 2 \to c'\ ,\ (x + x)[^{c'}/_{x}] \to c$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$$

$$= (1 \times 2) + (1 \times 2) \to c$$

evaluated twice

$$\nwarrow_{c=c_1 \underline{+} c_2} \boxed{(1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2}$$

$$\nwarrow^{*}_{c_1=2, c_2=2} \square \qquad c = c_1 \underline{+} c_2 = 2 \underline{+} 2 = 4$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ 1 \times 2 \to c'\ ,\ t'[^{c'}/_{x'}] \to c$$

eager

$$\nwarrow_{x'=x,\ t'=x+x}\ 1 \times 2 \to c'\ ,\ (x + x)[^{c'}/_{x}] \to c$$

$$\nwarrow^{*}_{c'=2}\ (x + x)[^{2}/_{x}] \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$$

$$= (1 \times 2) + (1 \times 2) \to c$$

evaluated twice

$$\nwarrow_{c=c_1 \underline{+} c_2} \boxed{(1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2}$$

$$\nwarrow^{*}_{c_1=2, c_2=2} \square \qquad c = c_1 \underline{+} c_2 = 2 \underline{+} 2 = 4$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ 1 \times 2 \to c'\ ,\ t'[^{c'}/_{x'}] \to c$$

eager

$$\nwarrow_{x'=x,\ t'=x+x}\ 1 \times 2 \to c'\ ,\ (x + x)[^{c'}/_{x}] \to c$$

$$\nwarrow^{*}_{c'=2}\ (x + x)[^{2}/_{x}] \to c$$

$$= 2 + 2 \to c$$

Lazy vs Eager

$$t \triangleq (\lambda x.\ x + x)\ (1 \times 2) : int \qquad x : int$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ t'[^{1\times 2}/_{x'}] \to c$$

lazy

$$\nwarrow_{x'=x,\ t'=x+x}\ (x + x)[^{1\times 2}/_{x}] \to c$$

$$= (1 \times 2) + (1 \times 2) \to c$$

evaluated twice

$$\nwarrow_{c=c_1 \pm c_2} \boxed{(1 \times 2) \to c_1\ ,\ (1 \times 2) \to c_2}$$

$$\nwarrow^{*}_{c_1=2, c_2=2} \square \qquad c = c_1 \pm c_2 = 2 \pm 2 = 4$$

$$t \to c \ \nwarrow\ \lambda x.\ x + x \to \lambda x'.\ t'\ ,\ 1 \times 2 \to c'\ ,\ t'[^{c'}/_{x'}] \to c$$

eager

$$\nwarrow_{x'=x,\ t'=x+x}\ 1 \times 2 \to c'\ ,\ (x + x)[^{c'}/_{x}] \to c$$

$$\nwarrow^{*}_{c'=2}\ (x + x)[^{2}/_{x}] \to c$$

$$= 2 + 2 \to c$$

$$\nwarrow^{*}_{c=4}\ \square$$