Jan Hoffmann (Lect3) [2018/07/05]

Finite Data Structures

Recap

Base language

typ $\tau ::= \mathrm{app}(\tau_1;\tau_2) \mid \tau_1 \to \tau_2$

$\cdots$

exp $e ::= \mathrm{lam}\{\tau\}(x.e) \mid \lambda(x:\tau)e$

$\mathrm{app}(e_1;e_2) \mid e_1(e_2)$

Statics

$\Gamma \vdash e : \tau$

Type Rules

$$\frac{\Gamma, x:\tau \vdash e : \tau'}{\Gamma \vdash \lambda(x:\tau)e : \tau \to \tau'}$$

$$\frac{\Gamma \vdash e_1 : \tau_2 \to \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1(e_2) : \tau}$$

Dynamics

$$\frac{}{\lambda(x)e \ \mathrm{val}}$$

Steps

$$\frac{e_1 \longmapsto e_1'}{e_1(e_2) \longmapsto e_1'(e_2)}$$

$$\frac{e_1 \ \mathrm{val} \quad e_2 \longmapsto e_2'}{e_1(e_2) \longmapsto e_1(e_2')}$$

$$\frac{e_2 \ \mathrm{val}}{(\lambda(x:\tau)e)(e_2) \longmapsto [e_2/x]e}$$

Fin Data Structures

Products

- conjunctive combination of data
- tuples, records, structs, unit

Today: pairs Typ $\tau ::= \ldots$ (extend our types with)

unit unit

abstract syntax: $\mathrm{prod}(\tau_1; \tau_2)$ — concrete syntax: $\tau_1 \times \tau_2$

Exp $e ::=$

triv	$\langle\rangle$	Intro forms
pair$(e_1; e_2)$	$\langle e_1; e_2 \rangle$	
pr[l]e	e.l	elim forms
pr[r]e	e.r	

Statics

Intro:

$$\frac{}{\Gamma \vdash \langle\rangle : \mathrm{unit}}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}\ (\times I)$$

elim:

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash e.l : \tau_1}\ (E_l) \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash e.r : \tau_2}\ (E_r)$$

Jan Hoffmann

Dynamics (for product)

$$\frac{}{\langle\rangle \text{ val}} \qquad \frac{e_1 \text{ val} \quad e_2 \text{ val}}{\langle e_1, e_2\rangle \text{ val}}$$

$$\frac{e_1 \longmapsto e_1'}{\langle e_1, e_2\rangle \longmapsto \langle e_1', e_2\rangle} \qquad \frac{e_1 \text{ val} \quad e_2 \longmapsto e_2'}{\langle e_1, e_2\rangle \longmapsto \langle e_1, e_2'\rangle}$$

$$\frac{e \longmapsto e'}{e.l \longmapsto e'.l} \qquad \frac{e_1 \text{ val} \quad e_2 \text{ val}}{\langle e_1, e_2\rangle.l \longmapsto e_1}$$

$$\frac{e \longmapsto e'}{e.r \longmapsto e'.r} \qquad \frac{e_1 \text{ val} \quad e_2 \text{ val}}{\langle e_1, e_2\rangle.r \longmapsto e_2}$$

Example $\quad (\lambda(x : \text{unit}).\langle x, x\rangle)\,\langle\rangle).l$

$\longmapsto \langle\langle\rangle,\langle\rangle\rangle.l \longmapsto \langle\rangle$

Products are very useful.
We get some things for free

1) multiple fn args
$(\tau_1 \times \tau_2) \longrightarrow \tau$

2) multiple return val
$\tau_1 \longrightarrow \tau_1 \times \tau_2$

Jan Hoffmann [2018/07/05] (4)

Sum Types

- disjunctive combination of data
- enums, options, void

Nullary & binary sums Typ τ ::=

	void	void
	$\mathsf{sum}(\tau_1;\tau_2)$	$\tau_1 + \tau_2$
	abstract syntax	concrete syntax

Exp e ::=

intro forms

$\mathsf{in}[l]\{\tau_1;\tau_2\}(e)$	$l \cdot e$
$\mathsf{in}[r]\{\tau_1;\tau_2\}(e)$	$r \cdot e$

(we include type info here because we don't want to get stuck on something like $l \cdot \langle\rangle : \mathsf{unit} + ?$)

elim forms

$\mathsf{case}(e; x_1.e_1, x_2.e_2)$	case e { $l.x_1 \longrightarrow e_1$ $r.x_2 \longrightarrow e_2$ }
$\mathsf{abort}\{\tau\}(e)$	case e { }

Statics (for sum type)

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash in[l]\{\tau_1,\tau_2\}(e) : \tau_1 + \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash in[r]\{\tau_1,\tau_2\}(e) : \tau_1 + \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \quad \Gamma, x_1 : \tau_1 \vdash e_1 : \tau \quad \Gamma, x_2 : \tau_2 \vdash e_2 : \tau}{\Gamma \vdash case(e; x_1.e_1; x_2.e_2) : \tau}$$

$$\frac{\Gamma \vdash e : void}{\Gamma \vdash case\, e\{\} : \tau}$$

$$(abort\{\tau\}(e))$$

Dynamics (for sum type)

$$\frac{e\ val}{l \cdot e\ val} \qquad \frac{e\ val}{r \cdot e\ val}$$

Steps

$$\frac{e \longmapsto e'}{l \cdot e \longmapsto l \cdot e'}$$

$$\frac{e \longmapsto e'}{case(e; x_1.e_1; x_2.e_2) \longmapsto case(e'; x_1.e_1; x_2.e_2)}$$

$$\frac{e\ val}{case(l \cdot e; x_1.e_1; x_2.e_2) \longmapsto [e/x_1]e_1}$$

(other step rules are similar)

Examples

- $\mathrm{bool} = \mathrm{unit} + \mathrm{unit}$
- $\mathrm{Enum} = \mathrm{unit} + \cdots + \mathrm{unit}$
- Option Type

$$\mathrm{opt}\ \tau = \mathrm{unit} + \tau$$

$$\mathrm{null} = l \cdot \langle\rangle$$

$$\mathrm{just}\ e = r \cdot e$$

$$\mathrm{ifnull}\ e\ \{\mathrm{null} \hookrightarrow e_1 \mid \mathrm{just}(x_2) \hookrightarrow e_2\}$$

$$:= \mathrm{case}(e; x_1.e_1; x_2.e_2)$$