Lecture Notes on Eager Products

15-814: Types and Programming Languages
Ryan Kavanagh

Lecture 7
Tuesday, September 25, 2018

1 Introduction

Last time, we added sums to our language. This allowed us to deal with collections of individual "tagged" values. Sometimes we would like to simultaneously consider multiple values. To do this, we introduce *eager products*. These are akin to "pairs" or "tuples" of values.

2 Syntax

We need to extend the syntax for our types and our terms to handle the new constructs:

$\tau ::= \cdots$	
$\mid \tau_1 \otimes \tau_2$	eager product of τ_1 and τ_2
$\mid \mathbf{1}$	nullary product
$e ::= \cdots$	
$\mid \langle e_1, e_2 \rangle$	ordered pair of e_1 and e_2
$\mid \langle\,\rangle$	null tuple
$\mid$ **case** $e\ \{\langle x_1, x_2 \rangle \Rightarrow e'\}$	eager pair destructor
$\mid$ **case** $e\ \{\langle\,\rangle \Rightarrow e'\}$	null tuple destructor

3 Statics

The product type has the following introduction rules:

$$\frac{}{\Gamma \vdash \langle\,\rangle : \mathbf{1}}\ (\text{I-}\mathbf{1}) \qquad \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2\rangle : \tau_1 \otimes \tau_2}\ (\text{I-}\otimes)$$

Its elimination rules are:

$$\frac{\Gamma \vdash e_0 : \mathbf{1} \quad \Gamma \vdash e_1 : \tau}{\Gamma \vdash \textbf{case}\ e_0\ \{\langle\,\rangle \Rightarrow e_1\} : \tau}\ (\text{E-}\mathbf{1})$$

$$\frac{\Gamma \vdash e_0 : \tau_1 \otimes \tau_2 \quad \Gamma, x_1 : \tau_1, x_2 : \tau_2 \vdash e_1 : \tau}{\Gamma \vdash \textbf{case}\ e_0\ \{\langle x_1, x_2\rangle \Rightarrow e_1\} : \tau}\ (\text{E-}\otimes)$$

4 Dynamics

The intended semantics is that we always eagerly evaluate pairs and only eliminate a **case** when both of the paired expressions are values. First, a tuple is a value only when all of its components are values:

$$\frac{}{\langle\,\rangle\ \mathit{val}}\ (\langle\,\rangle\text{-VAL}) \qquad \frac{v_1\ \mathit{val} \quad v_2\ \mathit{val}}{\langle v_1, v_2\rangle\ \mathit{val}}\ (\text{PAIR-VAL})$$

Otherwise, we reduce the components to values from left to right:

$$\frac{e_1 \mapsto e_1'}{\langle e_1, e_2\rangle \mapsto \langle e_1', e_2\rangle}\ (\text{STEP-L}) \qquad \frac{v_1\ \mathit{val} \quad e_2 \mapsto e_2'}{\langle v_1, e_2\rangle \mapsto \langle v_1, e_2'\rangle}\ (\text{STEP-R})$$

In the elimination forms, we step the subjects until they become values:

$$\frac{e_0 \mapsto e_0'}{\textbf{case}\ e_0\ \{\langle\,\rangle \Rightarrow e_1\} \mapsto \textbf{case}\ e_0'\ \{\langle\,\rangle \Rightarrow e_1\}}\ (\text{STEP-SUBJ-1})$$

$$\frac{e_0 \mapsto e_0'}{\textbf{case}\ e_0\ \{\langle x_1, x_2\rangle \Rightarrow e_1\} \mapsto \textbf{case}\ e_0'\ \{\langle x_1, x_2\rangle \Rightarrow e_1\}}\ (\text{STEP-SUBJ-2})$$

Then we simultaneously substitute where applicable:

$$\frac{}{\textbf{case}\ \langle\,\rangle\ \{\langle\,\rangle \Rightarrow e_1\} \mapsto e_1}\ (\text{STEP-CASE-1})$$

$$\frac{\langle v_1, v_2\rangle\ \mathit{val}}{\textbf{case}\ \langle v_1, v_2\rangle\ \{\langle x_1, x_2\rangle \Rightarrow e_1\} \mapsto [v_1, v_2/x_1, x_2]e_1}\ (\text{STEP-CASE-2})$$

5 Desiderata

Our definition satisfies all of our desiderata. The proofs are left as exercises.

Theorem 1 (Type Safety) *Our rules satisfy the progress property, that is, for all* e*,* τ_1*, and* τ_2*,*

1. *if* $\cdot \vdash e : \mathbf{1}$*, then either* e *val or there exists an* e' *such that* $e \mapsto e'$*, and*
2. *if* $\cdot \vdash e : \tau_1 \otimes \tau_2$*, then either* e *val or there exists an* e' *such that* $e \mapsto e'$*.*

They also satisfy the preservation property, that is, for all e*,* e'*,* τ_1*, and* τ_2*,*

1. *if* $\cdot \vdash e : \mathbf{1}$ *and* $e \mapsto e'$*, then* $\cdot \vdash e' : \mathbf{1}$*, and*
2. *if* $\cdot \vdash e : \tau_1 \otimes \tau_2$ *and* $e \mapsto e'$*, then* $\cdot \vdash e' : \tau_1 \otimes \tau_2$*.*

Proof: The proof of progress is by induction on the derivation of $\cdot \vdash e : \tau$. The proof of preservation is by induction on the derivation of $e \mapsto e'$. □

Theorem 2 (Canonical Forms for Eager Products) *Values have the following characterization:*

1. *If* $\cdot \vdash e : \mathbf{1}$ *and* e *val, then* $e \equiv \langle\,\rangle$*.*
2. *If* $\cdot \vdash e : \tau_1 \otimes \tau_2$ *and* e *val, then* $e \equiv \langle v_1, v_2\rangle$*, where* $\cdot \vdash v_i : \tau_i$ *and* v_i *val.*

6 Programming with pairs

To better grasp how these pairs work, let us do a bit of programming. We say that types τ and τ' are **isomorphic**, $\tau \cong \tau'$, if there exist terms $f : \tau \to \tau'$ and $g : \tau' \to \tau$ such that are mutual inverses. The exact meaning of "mutual inverses" is subtle because it requires us to specify what we mean by equality when we say $f(g(x)) = x$ and $g(f(x)) = x$. For our call-by-value language, it will be sufficient to require for certain x that $f(g(x)) \mapsto^* x$ and $g(f(x)) \mapsto^* x$. Explicitly, types τ and τ' are isomorphic if there exist f and g satisfying:

- $\cdot \vdash f : \tau \to \tau'$,
- $\cdot \vdash g : \tau' \to \tau$,
- for all v such that $\cdot \vdash v : \tau$ and v *val*, $g(f(v)) \mapsto^* v$, and
- for all v such that $\cdot \vdash v : \tau'$ and v *val*, $f(g(v)) \mapsto^* v$.

In this case, we say that f and g are **witnesses** to the isomorphism.

6.1 Unit is a unit

Our first observation is that $\mathbf{1}$ is the unit for $\otimes$, i.e., that for all τ,

$$\tau \otimes \mathbf{1} \cong \tau.$$

This isomorphism is witnessed by the following pair of mutual inverses:

$$\begin{aligned} \rho &= \lambda x.\textbf{case}\ x\ \{\langle l, \_\rangle \Rightarrow l\}, \\ \rho^{-1} &= \lambda x.\langle x, \langle\,\rangle\rangle. \end{aligned}$$

We begin by showing that they have the right types. First, we show that $\cdot \vdash \rho : \tau \otimes \mathbf{1} \to \tau$:

$$\dfrac{\dfrac{\dfrac{}{x : \tau \otimes \mathbf{1} \vdash x : \tau \otimes \mathbf{1}}\ (\textsc{Var}) \qquad \dfrac{}{x : \tau \otimes \mathbf{1}, l : \tau, \_ : \mathbf{1} \vdash l : \tau}\ (\textsc{Var})}{x : \tau \otimes \mathbf{1} \vdash \textbf{case}\ x\ \{\langle l, \_\rangle \Rightarrow l\} : \tau}\ (\text{E-}\otimes)}{\cdot \vdash \lambda x.\textbf{case}\ x\ \{\langle l, \_\rangle \Rightarrow l\} : \tau \otimes \mathbf{1} \to \tau}\ (\textsc{Lam})$$

Next, we show $\cdot \vdash \rho^{-1} : \tau \to \tau \otimes \mathbf{1}$:

$$\dfrac{\dfrac{\dfrac{}{x : \tau \vdash x : \tau}\ (\textsc{Var}) \qquad \dfrac{}{x : \tau \vdash \langle\,\rangle : \mathbf{1}}\ (\text{I-}\mathbf{1})}{x : \tau \vdash \langle x, \langle\,\rangle\rangle \tau \otimes \mathbf{1}}\ (\text{I-}\otimes)}{\cdot \vdash \lambda x.\langle x, \langle\,\rangle\rangle : \tau \to \tau \otimes \mathbf{1}}\ (\textsc{Lam})$$

We must also show that these two functions are mutual inverses. This requires us to show that for all values v such that $\vdash v : \tau$, we have the following reduction, where we colour-code the redexes in red:

$$\begin{aligned} \rho(\rho^{-1}(v)) &\equiv (\lambda x.\textbf{case}\ x\ \{\langle l, \_\rangle \Rightarrow l\})((\lambda x.\langle x, \langle\,\rangle\rangle)v) \\ &\mapsto (\lambda x.\textbf{case}\ x\ \{\langle l, \_\rangle \Rightarrow l\})\langle v, \langle\,\rangle\rangle \\ &\mapsto \textbf{case}\ \langle v, \langle\,\rangle\rangle\ \{\langle l, \_\rangle \Rightarrow l\} \\ &\mapsto [v, \langle\,\rangle / l, \_]l \\ &\equiv v. \end{aligned}$$

We must also show that for all values v such that $\vdash v : \tau \otimes \mathbf{1}$, we have by the canonical forms theorem that $v \equiv \langle t, \langle\,\rangle\rangle$ for some value t satisfying $\vdash t : \tau$.

We then have the following reduction:

$$\begin{aligned}
\rho^{-1}(\rho(v)) &\equiv (\lambda x.\langle x, \langle\,\rangle\rangle)((\lambda x.\textbf{case}\ x\ \{\langle l, _\rangle \Rightarrow l\})v) \\
&\mapsto (\lambda x.\langle x, \langle\,\rangle\rangle)(\textbf{case}\ v\ \{\langle l, _\rangle \Rightarrow l\}) \\
&\equiv (\lambda x.\langle x, \langle\,\rangle\rangle)(\textbf{case}\ \langle t, \langle\,\rangle\rangle\ \{\langle l, _\rangle \Rightarrow l\}) \\
&\mapsto (\lambda x.\langle x, \langle\,\rangle\rangle)([t, \langle\,\rangle/l, _]l) \\
&\equiv (\lambda x.\langle x, \langle\,\rangle\rangle)t \\
&\mapsto \langle t, \langle\,\rangle\rangle \\
&\equiv v.
\end{aligned}$$

6.2 One is not two

In general, it is not the case for arbitrary τ that:

$$\tau \cong \tau + \tau.$$

To see why, it is sufficient to take $\tau = \mathbf{1}$ and observe that $\mathbf{1}$ has one value, $\langle\,\rangle$, while $\mathbf{1} + \mathbf{1}$ has two, $l \cdot \langle\,\rangle$ and $r \cdot \langle\,\rangle$. Consequently, no term can induce a surjection from the values of $\mathbf{1}$ to the values of $\mathbf{1} + \mathbf{1}$.

6.3 Distributivity

Products distribute over sums, i.e., for all τ, ρ, and σ:

$$\tau \otimes (\rho + \sigma) \cong \tau \otimes \rho + \tau \otimes \sigma.$$

This isomorphism is witnessed by the following pair of mutual inverses:

$$\begin{aligned}
\xi &= \lambda x.\textbf{case}\ x\ \{\langle t, s\rangle \Rightarrow \textbf{case}\ s\ \{l \cdot u \Rightarrow l \cdot \langle t, u\rangle \\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \mid r \cdot w \Rightarrow r \cdot \langle t, w\rangle\}\}, \\
\xi^{-1} &= \lambda x.\textbf{case}\ x\ \{l \cdot y \Rightarrow \textbf{case}\ y\ \{\langle t, r\rangle \Rightarrow \langle t, l \cdot r\rangle\} \\
&\qquad\qquad\qquad\quad \mid r \cdot y \Rightarrow \textbf{case}\ y\ \{\langle t, s\rangle \Rightarrow \langle t, r \cdot s\rangle\}\}.
\end{aligned}$$

In the case of ξ, we take in a term x of type $\tau \otimes (\rho + \sigma)$ and decompose it into a $t : \tau$ and an $s : \rho + \sigma$. We do case analysis on s to determine if is a left injection or a right injection. If it is a left injection, then we get a term $u : \rho$ and inject the pair $\langle t, u\rangle$ into the left to get a term of type $\tau \otimes \rho + \tau \otimes \sigma$. We proceed symmetrically if s reduces to a left right injection. The definition of ξ^{-1} is similar. The details are left as an exercise.

6.4 Currying

We can *curry* functions, i.e., for all τ, ρ, and σ:

$$\tau \to (\rho \to \sigma) \cong (\tau \otimes \rho) \to \sigma.$$

This isomorphism is witnessed by the following pair of mutual inverses:

$$\zeta = \lambda f.\lambda x.\textbf{case}\ x\ \{\langle t, r\rangle \Rightarrow f t r\},$$
$$\zeta^{-1} = \lambda f.\lambda t.\lambda r.f\langle t, r\rangle.$$

In the ζ case, the intuition is that we must take in a function $f : \tau \to (\rho \to \sigma)$ and produce a function of type $(\tau \otimes \rho) \to \sigma$. We do so by taking in a term x of type $\tau \otimes \rho$, and eliminating it to get terms $t : \tau$ and $r : \rho$ to which we can apply f and ft, respectively.

In the ζ^{-1} case, the intuition is that we must take in a function f : $(\tau \otimes \rho) \to \sigma$ and produce a function of type $\tau \to (\rho \to \sigma)$. To do so, we need to take in a term $t : \tau$ and produce a term of type $\rho \to \sigma$. To produce such a term, we take in a $r : \rho$ and must produce a term of type σ. By pairing together t and r, we get a term $\langle t, r\rangle : \tau \otimes \rho$ to which we can apply f to get a term of type σ.

As we discovered in class, we could instead take the left and right projections out of x:

$$\zeta = \lambda f.\lambda x.f(\textbf{case}\ x\ \{\langle l, \rangle \Rightarrow l\})(\textbf{case}\ x\ \{\langle \_, r\rangle \Rightarrow r\}).$$