Assignment 7: Subtyping and Labeled Variants
Model Solution

15-312 Foundations of Programming Languages
Kevin Watkins ⟨kw@cmu.edu⟩

April 28, 2005

Please refer to the assignment itself for the full description and statement of each problem.

§ 1. Width Subtyping for Products **A 1.1.** [10 pts] Type safety is violated. Consider $(1, 2) + 3$, which is well-typed but stuck.

§ 2. Labeled Variants **A 2.1.** [10 pts] There are a couple of different ways of doing this. The most natural seems to be to make the rules for $\mathbf{in}_\ell$ and **case** as *restrictive* as possible, so the subsumption rule ends up doing most of the work.

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathbf{in}_\ell\, e : [\ell : \tau]}$$

$$\frac{\Gamma \vdash e : [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n] \quad \Gamma, (x_1 : \tau_1) \vdash e_1 : \sigma \quad \cdots \quad \Gamma, (x_n : \tau_n) \vdash e_n : \sigma}{\Gamma \vdash (\mathbf{case}\ e\ \mathbf{of}\ \mathbf{in}_{\ell_1}\, x_1 \Rightarrow e_1 \mid \ldots \mid \mathbf{in}_{\ell_n}\, x_n \Rightarrow e_n) : \sigma}$$

A 2.2. [10 pts] Again I'm taking the strategy here of making transitivity do most of the work. For *algorithmic* subtyping, we'd have very different-looking rules (or actually, probably just one rule combining the functionality of all three).

$$\frac{}{[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n] \leq [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n, \ell : \sigma]}$$

$$\frac{\tau_1 \leq \sigma_1 \quad \cdots \quad \tau_n \leq \sigma_n}{[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n] \leq [\ell_1 : \sigma_1, \ldots, \ell_n : \sigma_n]}$$

$$\frac{}{[\ell_1 : \tau_1, \ldots, \ell_i : \tau_i, \ell_{i+1} : \tau_{i+1}, \ldots, \ell_n : \tau_n] \leq [\ell_1 : \tau_1, \ldots, \ell_{i+1} : \tau_{i+1}, \ell_i : \tau_i, \ldots, \ell_n : \tau_n]}$$

A 2.3. [20 pts] The generalization of the canonical forms lemma is

> Canonical Forms *If* $\cdot \vdash v : [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$, *then* $v = \boldsymbol{in}_{\ell_i}\, v'$ *and* $\cdot \vdash v' : \tau_i$, *for some* i *and* v', $1 \leq i \leq n$.

Now trying to prove this immediately by induction fails, because at the subsumption rule, we can't apply the i.h., because we don't know what the left-hand type in the subsumption looks like—it might not even be a labeled

variant type, for all we know.

So we have to know something about labeled variants and subtyping. Specifically, if we know $\sigma \leq [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$, we'd like to know something about σ. There are precise conditions we could prove about σ, something along the lines of 'σ is a labeled variant, the domain of σ is contained in the domain of τ, and $\sigma(\ell) \leq \tau(\ell)$ for every ℓ in the domain of σ.' But that seems unnecessarily complicated.

The following is one possible solution to the problem. There are thousands of others; probably many are simpler. Usually the solution that comes to my mind is not the simplest, but it scales up to larger type systems pretty well.

It turns out that we only really need to know how the subtyping on labeled variants interacts with injection values. The following definition captures the idea:

> DEFINITION *We say v is* directly in τ *iff $v = \boldsymbol{in}_{\ell_i}\ v'$, $\tau = [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$, $\ell = \ell_i$, and $\cdot \vdash v' : \tau_i$, for some v' and i, $1 \leq i \leq n$.*

Now the lemma just says that 'directly in' can be transferred from a subtype to its supertype.

> LEMMA 1 *If $\boldsymbol{in}_\ell\ v$ is directly in τ and $\tau \leq \sigma$, then $\boldsymbol{in}_\ell\ v$ is directly in σ.*

Proof. By rule induction on the derivation of $\tau \leq \sigma$. Note first that by the definition of 'directly in', $\tau = [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$.

Case [$\dfrac{}{\tau \leq \tau}$ (reflexivity)]:
where $\sigma = \tau$
1. $\mathbf{in}_\ell\ v$ is directly in σ ($\sigma = \tau$)

Case [$\dfrac{\tau \leq \tau_1 \quad \tau_1 \leq \sigma}{\tau \leq \sigma}$ (transitivity)]:
1. $\mathbf{in}_\ell\ v$ is directly in τ_1 (i.h. on left premise)
2. $\mathbf{in}_\ell\ v$ is directly in σ (i.h. on right premise)

Now we come to the 'meaty' rules. (Yum!)

Case [$\dfrac{}{[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n] \leq [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n, \ell' : \sigma]}$]:
1a. $\ell = \ell_i$ (defn of 'directly in')
1b. $\cdot \vdash v : \tau_i$ (defn of 'directly in')
2. v is directly in $[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n, \ell' : \sigma]$ (defn of 'directly in')

Case [$\dfrac{\tau_1 \leq \sigma_1 \quad \cdots \quad \tau_n \leq \sigma_n}{[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n] \leq [\ell_1 : \sigma_1, \ldots, \ell_n : \sigma_n]}$]:
1a. $\ell = \ell_i$ (defn of 'directly in')
1b. $\cdot \vdash v : \tau_i$ (defn of 'directly in')
2. $\tau_i \leq \sigma_i$ (premise)
3. $\cdot \vdash v : \sigma_i$ (subsumption on 1b and 2)
4. v is directly in $[\ell_1 : \sigma_1, \ldots, \ell_n : \sigma_n]$ (defn of 'directly in')

The case for the permutation rule is similar.

And that's the proof of the lemma.

It turns out that we need a second lemma:

> LEMMA 2 *If $\sigma \leq [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$, then σ is a labeled variant type.*

The point of the second lemma is just to get us enough information to apply the i.h. to the subsumption rule. It's another simple rule induction on the derivation of $\sigma \leq [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$.

Now we prove that any value of labeled variant type is 'directly in' the labeled variant type.

> LEMMA 3 *If $\cdot \vdash v : [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$ then v is directly in $[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$.*

Proof. By rule induction on the derivation of $\cdot \vdash v : [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$.

Case $\left[\dfrac{\cdot \vdash v : \sigma \quad \sigma \leq [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]}{\cdot \vdash v : [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]}\right]$:

1. σ is a labeled variant type (Lemma 2 on right premise)
2. $\sigma = [\ell'_1 : \tau'_1, \ldots, \ell'_n : \tau'_m]$ (1)
3. v is directly in σ (i.h. on 2 and left premise)
4. v is directly in $[\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$ (Lemma 1 on 3)

Case $\left[\dfrac{\cdot \vdash v' : \tau}{\cdot \vdash \mathbf{in}_\ell\, v' : [\ell : \tau]}\right]$:
where $v = \mathbf{in}_\ell\, v'$

1. $\cdot \vdash v' : \tau$ (premise)
2. v is directly in $[\ell : \tau]$ (defn of 'directly in')

No other rules have conclusions matching the assumption $\cdot \vdash v : [\ell_1 : \tau_1, \ldots, \ell_n : \tau_n]$ that we're inducting over, so we're done.

Now the canonical forms lemma follows immediately, because Lemma 3 is equivalent to it by the definition of 'directly in'. QED.