Calculating Threesomes, with Blame

Ronald Garcia

September 24, 2013

Abstract

Research on combining static and dynamic types has yielded two approaches to preserving proper tail calls: coercion calculi and threesomes. Coercion calculi elegantly specify space-efficient cast behavior, even when augmented with blame tracking, but implementing their semantics directly is difficult. Threesomes, on the other hand, have a straightforward recursive implementation, but endowing them with blame tracking was a subtle and opaque process. This paper bridges the gap between blame-tracking coercions and threesomes by using the former to produce the latter. We revisit the coercion- and threesome-based variants of Wadler and Findler's Blame Calculus and stepwise transform the coercion calculus into its threesome counterpart. Guided by this process, we construct three new threesome calculi for blame tracking strategies identified by Siek et al. Using these results, we can implement cast-based languages that detect errors earlier, catch more errors, and reflect an intuitive conception of safe and unsafe casts.

1 Introduction

Languages that combine static and dynamic typing have recently been the subject of significant study [Tobin-Hochstadt and Felleisen, 2008, Siek and Vachharajani, 2008, Sergey and Clarke, 2012, Ina and Igarashi, 2011] One particular family of such languages is $\lambda^{\langle T \rangle}$, the lambda calculus extended with a dynamic type $\star$ and casts (Fig. 1). A suite of semantics for it have been proposed (e.g., [Siek and Taha, 2006]), including variants that track and report the source of cast failures (termed "blame tracking") [Wadler and Findler, 2009]. These systems bring more checking to dynamic programs, more flexibility to static programs, and a smooth migration path between the two.

Herman et al. [2007] observed that *higher-order* casts—casts between function types—can cause space leaks (Sect. 2.1). This problem has been addressed using two seemingly distinct approaches. The first approach compiles casts to *coercions* [Henglein, 1994], a low-level representation and machine model that can represent arbitrary strings of casts compactly [Herman et al., 2007, Siek et al., 2009]. The second approach compiles casts to a *threesome* of types, which represents a downcast (from the first type to the second), followed by an upcast (from the second type to the third) [Siek and Wadler, 2010]. Threesomes are combined by *composing* their middle types to produce a new middle type.

These approaches to space efficiency have complementary advantages and disadvantages. Coercion calculi elegantly specify space-efficient cast behavior, even when augmented with blame tracking, but implementing their semantics directly is difficult (Sec. 2.7). Threesomes, on the other hand, have a straightforward recursive implementation, but endowing them with blame tracking was a subtle and opaque process (Sec. 2.2.1). Moreover, the Threesome Calculus [Siek and Wadler, 2010] provides a space-efficient implementation of one blame tracking strategy, but other blame tracking strategies have been developed on top of coercions with useful properties like catching earlier errors and a traditional notion of subtyping to characterize safe casts [Siek et al., 2009]. We and our collaborators desire an effective implementation approach for these other strategies as well.

This paper bridges the gap between blame-tracking coercions and threesomes by using the former to produce the latter. Our **contributions** are as follows:

- We develop three new threesome calculi for blame tracking strategies identified by [Siek et al., 2009]. Using these results, we can implement cast-based languages that detect more errors, detect them earlier, and present an intuitive conception of safe and unsafe casts (Sects. 4–5).

- As a step in our methodology we revisit the coercion- and threesome-based variants of the Blame Calculus [Wadler and Findler, 2009, Siek et al., 2009, Siek and Wadler, 2010]. After developing threesomes without blame, the key challenge in developing the threesome calculus was to introduce blame labels in a manner that reflects the desired blame strategy while preserving space-efficiency. This was originally done by experimenting with blame labels on threesomes, and proven correct post-hoc. In contrast, we start from the corresponding coercion calculus and transform it into a threesome calculus in small, easily verified steps (Sect. 3). This reconstruction lays the groundwork for developing new threesomes, but it also clearly shows that coercions and threesomes are two alternative representations of the same concept, one more suited to specification and the other to implementation. This deeper understanding is a secondary contribution of this work.

2 Background: Casts, Threesomes, and Coercions

This section introduces background on cast calculi, including space efficiency and blame assignment. We then discuss threesomes and coercion calculi, the two means for achieving space efficiency, and their strengths and weaknesses. For now, threesomes are discussed only briefly, as we revisit them in detail later. Coercions are discussed in detail now, since they are the technical starting point for our results.

2.1 Cast Calculi

A cast-based language like $\lambda^{\langle T \rangle}$ (Fig. 1) extends the lambda calculus with a dynamic type $\star$, which represents a universal type, as well as a *cast expression* $\langle T_2 \Leftarrow T_1 \rangle^l \, e$, which attempts to coerce the value of expression e from type T_1 to type T_2. If it succeeds, then computation proceeds with the value at the new type; if it fails, then a *cast error* indicates that the source-language cast with the label l is to blame. Many languages based on such principles have been designed and studied in the literature [Abadi et al., 1991, Wrigstad et al., 2010, Rastogi et al., 2012].

The semantics of casts are straightforward for first-order values. For instance, casting a number to $\star$ and back succeeds:

$$\langle \texttt{Int} \Leftarrow \star \rangle^{l_2} \, \langle \star \Leftarrow \texttt{Int} \rangle^{l_2} \, 42 \longrightarrow^* 42$$

but projecting a $\star$-wrapped integer to the wrong type fails:

$$\langle \texttt{Bool} \Leftarrow \star \rangle^{l_2} \, \langle \star \Leftarrow \texttt{Int} \rangle^{l_1} \, 42 \longrightarrow^* \textbf{blame} \; l_2$$

The *projection* from $\star$ to `Bool` is reported to have failed.

Languages like $\lambda^{\langle T \rangle}$ also support higher-order casts, which apply to function-typed objects, and whose meaning is dictated by the structure of the relevant types. As a basic example, consider the expression:

$$f \equiv \langle \star \to \texttt{Bool} \Leftarrow \star \to \star \rangle^{l_3} \, \lambda x : \star . \langle \star \Leftarrow \texttt{Int} \rangle^{l_2} \, 42$$

The function inside of the l_3 cast can never successfully behave as $\star \to \texttt{Bool}$, and the body of the function is not to blame for this since 42 can always be cast to $\star$. However, this cannot be determined at the point of the l_3 cast without analyzing the body of the function. For this reason, the cast l_3 cannot fail yet, but is ultimately a bad cast.

To detect these errors without analyzing function bodies, higher-order casts adopt the strategy of delaying resolution of the cast, and carry its blame label so that failure can be detected when the function is used [Findler and Felleisen, 2002]. To do so, the cast is split at the point of application and resolved in terms of the first-order strategy:

$$\begin{aligned}
& f \; (\langle \star \Leftarrow \texttt{Int} \rangle^{l_4} \, 7) \\
\longrightarrow \; & \langle \texttt{Bool} \Leftarrow \star \rangle^{l_3} \, ((\lambda x \ldots) \; \langle \star \Leftarrow \star \rangle^{l_3} \, \langle \star \Leftarrow \texttt{Int} \rangle^{l_4} \, 7) \\
\longrightarrow \; & \langle \texttt{Bool} \Leftarrow \star \rangle^{l_3} \, ((\lambda x \ldots) \; \langle \star \Leftarrow \texttt{Int} \rangle^{l_4} \, 7) \\
\longrightarrow \; & \langle \texttt{Bool} \Leftarrow \star \rangle^{l_3} \, \langle \star \Leftarrow \texttt{Int} \rangle^{l_2} \, 42 \longrightarrow \textbf{blame} \; l_3 .
\end{aligned}$$

In short, at the application point, the l_3 cast is split into two first-order casts, one of which is a first-order identity cast that immediately resolves. Then the function is called and its result is projected, yielding a failure that blames the source program's higher-order cast l_3.

Space Efficiency As Herman et al. [2007] observe, delayed higher-order casts can lead to undesirable space consumption, both by accumulating casts around values as well as on the control stack. To demonstrate the first, they present the following example.

```
let rec even(n : Int, k : Dyn→Bool) : Bool =
   if (n = 0) then k(⟨⋆⇐Bool⟩^l1 True)
   else odd(n - 1, ⟨Bool → Bool⇐⋆ → Bool⟩^l2 k)
and odd(n : Int, k : Bool→Bool) : Bool =
   if (n = 0) then k(False)
   else even(n - 1, ⟨⋆ → Bool⇐Bool → Bool⟩^l3 k)
```

Whenever `even` or `odd` is called at non-zero value, another wrapper is placed around the function k. Despite these functions being tail recursive, a trace of this routine shows the accumulating casts.

$$
\begin{aligned}
&\texttt{even(3,k)} \longrightarrow^* \\
&\texttt{odd(2,}\langle \texttt{Bool} \to \texttt{Bool} \Leftarrow \star \to \texttt{Bool}\rangle^{l_2}\ \texttt{k)} \longrightarrow^* \\
&\texttt{even(1,}\langle \star \to \texttt{Bool} \Leftarrow \texttt{Bool} \to \texttt{Bool}\rangle^{l_2} \\
&\qquad \langle \texttt{Bool} \to \texttt{Bool} \Leftarrow \star \to \texttt{Bool}\rangle^{l_3}\ \texttt{k)} \longrightarrow^* \\
&\texttt{odd(0,}\langle \texttt{Bool} \to \texttt{Bool} \Leftarrow \star \to \texttt{Bool}\rangle^{l_2} \\
&\qquad \langle \star \to \texttt{Bool} \Leftarrow \texttt{Bool} \to \texttt{Bool}\rangle^{l_3} \\
&\qquad \langle \texttt{Bool} \to \texttt{Bool} \Leftarrow \star \to \texttt{Bool}\rangle^{l_2}\ \texttt{k)} \longrightarrow \ldots
\end{aligned}
$$

A similar program to the above demonstrates how cast expressions can accumulate on the runtime stack and break proper tail recursion.

To address this expansion, Herman et al. turn to the Coercion Calculus [Henglein, 1994], a low-level formalism for casts that reduces compositions of casts according to its specialized instruction set. More recently, Siek and Wadler [2010] develops threesomes to alleviate the same space leakage issues. Later we show how each approach prevents these space leaks.

Blame Tracking Strategies Wadler and Findler [2009] adapted the notion of blame tracking from higher-order contracts to higher-order casts. Their Blame Calculus introduced a label-passing strategy for detecting cast failures and attributing them to source program casts. Later, Siek et al. [2009] observe that various calculi detect cast failures differently for the same expressions. That work lays out a design space for languages with dynamic casts, varying the cast detection strategy on the following two parameters.

First, some cast calculi detect failures more aggressively than others. Take for example the following expression:

$$\langle \texttt{Bool}\to\texttt{Int} \Leftarrow \star\to\star\rangle^{l_2}\ \langle \star\to\star \Leftarrow \texttt{Int}\to\texttt{Int}\rangle^{l_1}\ (\lambda x : \texttt{Int}.\ x)$$

The top-level type constructors of both casts are consistent, but the inner types, `Int` and `Bool`, conflict. According to the design space, a *lazy* blame tracking strategy evaluates this expression without error. An *eager* strategy, on the other hand, reports an error immediately.

Second, the $\star$ type can be viewed as a universal type to which any other type can be safely coerced. Under that view, $\star$ can be seen as a supertype of all other types, and an arrow types can be related according to traditional subtyping, which is contravariant on the domains and covariant on the ranges of function types. This model, called "D", blames only "downcasts" according to this conception. The blame calculus, in contrast, is designed to conceive of casts as a means at runtime of mediating the barrier between two interacting languages, one statically-typed and one dynamically typed. The blame strategy of the Blame Calculus (which we call "UD" after Siek et al. [2009]) ensures that blame is always allocated to code from the dynamic language[1]. This difference in strategy manifests in the cast calculus. For example, consider the following expression:

$$
\begin{aligned}
&(\langle \star\to\texttt{Int} \Leftarrow \star\rangle^{l_3} \\
&\qquad \langle \star \Leftarrow \texttt{Bool}\to\texttt{Bool}\rangle^{l_2}\ \lambda x : \texttt{Bool}.\ x)\ (\langle \star \Leftarrow \texttt{Int}\rangle^{l_1}\ 1)
\end{aligned}
$$

[1] This is partially achieved by assigning positive and negative *polarities* to blame labels. This aspect is not germane to this work, but straightforward to incorporate.

$x \in \mathsf{Variables},\ k \in \mathsf{Constants},\ B \in \mathsf{BaseTypes},\ l \in \mathsf{Labels}$

types	T	$::=$	$\star \mid B \mid T \to T$
expressions	e	$::=$	$k \mid x \mid \lambda x : T.\ e \mid e\ e \mid \langle T \Leftarrow T \rangle^l\ e$

$$\frac{\Gamma \vdash e : T_1}{\Gamma \vdash \langle T_2 \Leftarrow T_1 \rangle^l\ e : T_2}$$

Figure 1: The $\lambda^{\langle T \rangle}$ Calculus

This expression definitely leads to a cast failure, but while the UD strategy assigns blame with respect to the l_2 cast, the D strategy blames the l_3 downcast.

Siek et al. [2009] present 4 coercion-based calculi—named Lazy UD, Lazy D, Eager UD, and Eager D—that exhibit these different points in the design space of casts while supporting space-efficiency. The Lazy UD variant corresponds directly to the Blame Calculus, which was originally specified directly in terms of space-leaking casts.

2.2 Threesomes

The goal of threesomes is to represent any sequence of casts as exactly two casts, a cast $\langle T_2 \Leftarrow T_1 \rangle$ from a less *precise* type T_1 to a more precise type T_2 followed by a cast $\langle T_3 \Leftarrow T_2 \rangle$ to a less precise type T_3. These two casts are called a threesome because they always invoke three types, the two less precise *outer types* T_3 and T_1, and one most precise *middle type* T_2. For succinctness, such threesomes are written $\langle T_3 \overset{T_2}{\Longleftarrow} T_1 \rangle$.

Informally, a type is more precise than any other type that has the same structure except in positions where specific type structure is replaced with $\star$. This is formalized as a (partial) ordering of types $<:_n$:

$$\frac{}{B <:_n B} \qquad \frac{}{B <:_n \star} \qquad \frac{T_1 <:_n T_3 \quad T_2 <:_n T_4}{T_1 \to T_2 <:_n T_3 \to T_4}$$

For instance, $(\texttt{Int} \to \texttt{Bool}) \to \texttt{Int} <:_n \star \to \texttt{Int}$. Any single cast $\langle T_2 \Leftarrow T_1 \rangle$ can be represented by the threesome $\langle T_2 \overset{T_1 \,\&\, T_2}{\Longleftarrow} T_1 \rangle$ where $T_1 \ \& \ T_2$ is the *greatest lowerbound* of the two types with respect to the ordering relation. To ensure that $\&$ is well defined, the "type" $\bot$ is used to represent type inconsistencies.

$$\frac{}{\bot <:_n T}$$

The definition of $\&$ is as follows:

$$\star \ \& \ T = T \ \& \ \star = T \qquad\qquad B \ \& \ B = B$$
$$(T_1 \to T_2) \ \& \ (T_3 \to T_4) = (T_1 \ \& \ T_3) \to (T_2 \ \& \ T_4) \qquad\qquad T_1 \ \& \ T_2 = \bot \text{ if } T_1 \to\!\!\leftarrow T_2$$

This definition appeals to *shallow inconsistency* $\to\!\!\leftarrow$, which holds when two non-$\star$ types have different topmost type constructors:

$$T_1 \to\!\!\leftarrow T_2 \text{ iff } \lceil T_1 \rceil \neq \lceil T_2 \rceil$$
$$\text{where } \lceil B \rceil = B \text{ and } \lceil T_1 \to T_2 \rceil = \to$$

Its inverse, *shallow consistency*, is written $\leftrightarrow$.

Intuitively, the greatest lowerbound matches the structure of two types, and when a $\star$ in one type meets a static type in the other, the static type is the used. For example, the cast $\langle \texttt{Bool} \to \star \Leftarrow \star \to \texttt{Int} \rangle$ compiles to the threesome $\langle \texttt{Bool} \to \star \overset{\texttt{Bool} \to \texttt{Int}}{\Longleftarrow} \star \to \texttt{Int} \rangle$. The $\bot$ type is the greatest lower bound of two static types whose top-most type constructors don't match. For example, $\texttt{Int} \ \& \ \texttt{Bool} = \bot$, and
$((\texttt{Int} \to \texttt{Bool}) \to \texttt{Int}) \ \& \ (\texttt{Bool} \to \star) = \bot \to \texttt{Int}$.

Threesomes support space efficiency because any sequence of threesomes (and therefore any sequence of casts) can be represented as a single threesome. A sequence of threesomes is collapsed by taking the innermost and outermost types of the sequence as the new outer types, and taking the greatest lowerbound of all intermediate middle types as the new inner type. For example, a pair of threesomes can be reduced to a single threesome, e.g.:

$$\langle \texttt{Bool} \to \star \overset{\texttt{Bool} \to \texttt{Int}}{\Longleftarrow} \star \to \texttt{Int} \rangle\ \langle \star \to \texttt{Int} \overset{\texttt{Bool} \to \bot}{\Longleftarrow} \texttt{Bool} \to \star \rangle\ f \longrightarrow \langle \texttt{Bool} \to \star \overset{\texttt{Bool} \to \bot}{\Longleftarrow} \texttt{Bool} \to \star \rangle\ f.$$

Since greatest lowerbound is associative, this may be computed sequentially from left-to-right, or right-to-left.

Finally, as with traditional casts, higher-order threesomes are split when they appear in a function application, for example:

$$(\langle \texttt{Bool} \to \star \overset{\texttt{Bool}\to\texttt{Int}}{\Longleftarrow} \star \to \texttt{Int} \rangle\ f)\ \texttt{true} \longrightarrow \langle \star \overset{\texttt{Int}}{\Longleftarrow} \texttt{Int} \rangle\ (f\ \langle \star \overset{\texttt{Bool}}{\Longleftarrow} \texttt{Bool} \rangle\ \texttt{true})$$

A cast with a $\bot$ middle type fails immediately:

$$\langle \texttt{Bool} \overset{\bot}{\Longleftarrow} \texttt{Int} \rangle\ 42 \longrightarrow \textbf{blame}.$$

If we consider the even-odd example again, and ignore blame for now, we can compile its casts to threesomes: $\langle \star \Leftarrow \texttt{Bool} \rangle$ becomes $\langle \star \overset{\texttt{Bool}}{\Longleftarrow} \texttt{Bool} \rangle$, $\langle \texttt{Bool} \to \texttt{Bool} \Leftarrow \star \to \texttt{Bool} \rangle$ becomes $\langle \texttt{Bool} \to \texttt{Bool} \overset{\texttt{Bool}\to\texttt{Bool}}{\Longleftarrow} \star \to \texttt{Bool} \rangle$, and $\langle \star \to \texttt{Bool} \Leftarrow \texttt{Bool} \to \texttt{Bool} \rangle$ becomes $\langle \star \to \texttt{Bool} \overset{\texttt{Bool}\to\texttt{Bool}}{\Longleftarrow} \texttt{Bool} \to \texttt{Bool} \rangle$. Then the earlier execution trace demonstrates the space savings of threesomes:

$\texttt{even(3,k)} \longrightarrow^*$
$\texttt{odd(2,}\langle \texttt{Bool} \to \texttt{Bool} \overset{\texttt{Bool}\to\texttt{Bool}}{\Longleftarrow} \star \to \texttt{Bool} \rangle\texttt{ k)} \longrightarrow^*$
$\texttt{even(1,}\langle \star \to \texttt{Bool} \overset{\texttt{Bool}\to\texttt{Bool}}{\Longleftarrow} \star \to \texttt{Bool} \rangle\texttt{ k)} \longrightarrow^*$
$\texttt{odd(0,}\langle \texttt{Bool} \to \texttt{Bool} \overset{\texttt{Bool}\to\texttt{Bool}}{\Longleftarrow} \star \to \texttt{Bool} \rangle\texttt{ k)} \longrightarrow \ldots$

Rather than accumulating like casts, threesomes reduce to a single equivalent threesome at each step. One key property of threesomes that enables this is that their composition is associative. Associativity is important for space-efficiency because it means that threesomes on the call stack can be compressed from left-to-right as a computation proceeds, rather than waiting until a value is returned. This is key to recovering proper tail calls.

2.2.1 Adding Blame to Threesomes is Subtle

Threesomes have some very compelling properties. First, the composition operation for threesomes is easy to implement directly as a recursive function on the middle types. Furthermore, threesomes without blame are easy to understand and explain and proving that they correspond to casts is straightforward.

However, threesomes become more opaque when they are enhanced with support for blame tracking. To support blame tracking, Siek and Wadler [2010] annotates subcomponents of the middle type with blame labels, now calling them *labeled types*, and extends the composition operation to propagate those blame labels. Developing the annotation strategy was a subtle and time-consuming process [Siek, 2011b], involving trial and error.[2] The resulting system was proven equivalent to the coercion calculus after-the-fact, a great achievement, but one is hard-pressed to recognize the blame strategy as encoded in the labeling of the types or the structure of the composition operation. In short, we know that it's correct, but we don't necessarily understand *why*. Nevertheless, threesomes with blame are a compelling implementation model for blame-tracking languages.

Later, we show that we can re-discover the correct labelling and composition strategy for Lazy UD threesomes. The Threesome Calculus provides little guidance for how to make threesomes for other blame-tracking strategies, but our reconstruction paves the way for three *new* threesome-based calculi that correspond to alternative blame-tracking designs, without all the trial and error.

2.3 Lazy UD Coercions

In this section, we present in detail the Lazy UD Coercion Calculus, which provides another space-efficient semantics for the Blame Calculus. Coercions are expressions in a small language that characterizes type casts. Their formal syntax is as follows:

types	T	$::=$	$B \mid \star \mid T \to T$
atomic types	P	$::=$	$B \mid \star$
dynables	G	$::=$	$B \mid \to$
expressions	c	$::=$	$\iota_P \mid G! \mid G?^l \mid c \to c \mid c \circ c \mid \texttt{Fail}^l$

[2] e.g. "We initially tried to label bottom types with a single label ... . However, that approach fails to capture the correct blame tracking behavior." [Siek and Wadler, 2010]

As such, they refer to the types T of the surface language. There are essentially four classes of coercions. The identity coercions ι_P express casts that always succeed. In our system they are only defined at atomic types[3], since identity coercions at other types can be constructed from them. The injection coercions $G!$ express casting a value of some type to the $\star$ type. In particular, the coercions $B!$ express casts from base types to $\star$, while the arrow injection $\rightarrow!$ casts values of type $\star \rightarrow \star$ to $\star$. Coercions for more complex function types $T \rightarrow T$ are constructed from primitive coercions. Finally, the projection coercions $G?^l$ do the opposite, casting values from $\star$. Blame labels are associated with these projections, and complex function type coercions are built from primitive coercions. The arrow coercion constructor $c \rightarrow c$ is used to construct arrow coercions, which represent coercions among function types. Finally, sequences of coercions are composed using the $\circ$ operator. Note that coercions are identified up to associativity of composition. As with the threesome calculus, associativity is vital to space-efficiency.

Coercions are typed according to the type of input value they admit and the output type they are trying to cast it to:

$$\frac{}{\vdash \iota_P : P \Leftarrow P} \qquad \frac{}{\vdash B?^l : B \Leftarrow \star} \qquad \frac{}{\vdash \rightarrow?^l : \star \rightarrow \star \Leftarrow \star} \qquad \frac{}{\vdash B! : \star \Leftarrow B} \qquad \frac{}{\vdash \rightarrow! : \star \Leftarrow \star \rightarrow \star}$$

$$\frac{\vdash c_1 : T_2 \Leftarrow T_1 \quad \vdash c_2 : T_4 \Leftarrow T_3}{\vdash c_1 \rightarrow c_2 : T_1 \rightarrow T_4 \Leftarrow T_2 \rightarrow T_3} \qquad \frac{}{\vdash \mathtt{Fail}^l : T_2 \Leftarrow T_1} \qquad \frac{\vdash c_2 : T_3 \Leftarrow T_2 \quad \vdash c_1 : T_2 \Leftarrow T_1}{\vdash c_2 \circ c_1 : T_3 \Leftarrow T_1}$$

From here on, we only consider well-typed coercions.

2.3.1 Reduction

In order to characterize reduction, we must isolate some subsets of legal coercions:

$$\begin{array}{lrcl} \text{wrappers} & \overline{c} & ::= & G! \mid c^w \rightarrow c^w \mid G! \circ (c^w \rightarrow c^w) \\ \text{normal coercions} & c^w & ::= & \iota_P \mid \mathtt{Fail}^l \mid G! \mid c^w \rightarrow c^w \mid G! \circ (c^w \rightarrow c^w) \\ & & \mid & G?^l \mid \mathtt{Fail}^l \circ G?^l \mid G! \circ G?^l \mid (c^w \rightarrow c^w) \circ G?^l \mid G! \circ (c^w \rightarrow c^w) \circ G?^l \end{array}$$

These subsets capture coercions c^w that are in normal form, as well as coercions $\overline{c}$ that represent delayed (sequences of) casts wrapped around simple program values (e.g. $\langle \star \rightarrow \mathtt{Bool} \Leftarrow \mathtt{Bool} \rightarrow \mathtt{Bool} \rangle^{l_2} \langle \mathtt{Bool} \rightarrow \mathtt{Bool} \Leftarrow \star \rightarrow \mathtt{Bool} \rangle^{l_2} f$). Note that every wrapper is also a normal coercion. Coercion reduction is the compatible closure of the following rules:

$$\begin{array}{ll} (1) & \iota_P \circ c \longrightarrow^\circ c \\ (2) & c \circ \iota_P \longrightarrow^\circ c \\ (3) & B?^l \circ B! \longrightarrow^\circ \iota_B \\ (4) & \rightarrow?^l \circ \rightarrow! \longrightarrow^\circ \iota_\star \rightarrow \iota_\star \\ (5) & (c_1 \rightarrow c_2) \circ (c_3 \rightarrow c_4) \longrightarrow^\circ (c_3 \circ c_1) \rightarrow (c_2 \circ c_4) \\ (6) & \mathtt{Fail}^l \circ \overline{c} \longrightarrow^\circ \mathtt{Fail}^l \\ (7) & c \circ \mathtt{Fail}^l \longrightarrow^\circ \mathtt{Fail}^l \\ (8) & G_1?^l \circ G_2! \longrightarrow^\circ \mathtt{Fail}^l \text{ if } G_1 \neq G_2 \end{array}$$

The ι_P coercion acts as left and right identity. A $\mathtt{Fail}^l$ coercion can only consume a *wrapper* to its right but can consume *any* coercion to its left. This ensures that a rightmost failure will always dominate. For example, $\mathtt{Fail}^{l_1} \circ \mathtt{Int}?^{l_2} \circ \mathtt{Bool}!$ should reduce to $\mathtt{Fail}^{l_2}$, which corresponds to what a standard cast semantics would produce. To ensure this, $\mathtt{Fail}^l \circ G?^l$ never reduces.

Composition distributes over arrow coercion constructors, though the domain coercions are reversed so as to treat the domain contravariantly, while the codomain is treated covariantly.

Casts are resolved or fail whenever a projection meets an injection. If the two are compatible, then the result is an identity, since they annihilate each other. The identity for $\star \rightarrow \star$ is constructed using the arrow coercion constructor and the $\iota_\star$ coercion.

[3]This is a simplification of [Siek et al., 2009], which has ι at all types.

We relate casts to coercions using a compilation function:

$$\begin{aligned}
\langle\!\langle \star \Leftarrow B \rangle\!\rangle^l &= B! \\
\langle\!\langle B \Leftarrow \star \rangle\!\rangle^l &= B?^l \\
\langle\!\langle \star \Leftarrow (\star \to \star) \rangle\!\rangle^l &= \to! \\
\langle\!\langle \star \Leftarrow (T_3 \to T_4) \rangle\!\rangle^l &= \to! \circ (\langle\!\langle T_3 \Leftarrow \star \rangle\!\rangle^l \to \langle\!\langle \star \Leftarrow T_4 \rangle\!\rangle^l) \text{ otherwise} \\
\langle\!\langle (\star \to \star) \Leftarrow \star \rangle\!\rangle^l &= \to?^l \\
\langle\!\langle (T_1 \to T_2) \Leftarrow \star \rangle\!\rangle^l &= (\langle\!\langle \star \Leftarrow T_1 \rangle\!\rangle^l \to \langle\!\langle T_2 \Leftarrow \star \rangle\!\rangle^l) \circ \to?^l \text{ otherwise} \\
\langle\!\langle P \Leftarrow P \rangle\!\rangle^l &= \iota_P \\
\langle\!\langle T_1 \to T_2 \Leftarrow T_3 \to T_4 \rangle\!\rangle^l &= \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \to \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \\
\langle\!\langle T_1 \Leftarrow T_2 \rangle\!\rangle^l &= \texttt{Fail}^l \quad \text{if } T_1 \to\!\!\leftarrow T_2
\end{aligned}$$

For example, the paired casts

$$\langle \star \to \texttt{Bool} \Leftarrow \texttt{Bool} \to \texttt{Bool} \rangle^{l_2} \, \langle \texttt{Bool} \to \texttt{Bool} \Leftarrow \star \to \texttt{Bool} \rangle^{l_2}$$

compile and reduce to the well-typed coercion expression $\vdash (\texttt{Bool}! \circ \texttt{Bool}?^{l_3}) \to \iota_{\texttt{Bool}} : (\star \to \texttt{Bool}) \Leftarrow (\star \to \texttt{Bool})$.

2.4 The $\lambda^{\langle c \rangle}$ Calculus

To use the coercion calculus, we translate $\lambda^{\langle T \rangle}$ programs to a related language $\lambda^{\langle c \rangle}$, the simply-typed lambda calculus with coercions.

The $\lambda^{\langle c \rangle}$ syntax differs from $\lambda^{\langle T \rangle}$ only in that cast expressions are replaced with coercion expressions $\langle c \rangle e$, whose type rule follows:

$$\frac{\Gamma \vdash e : T_1 \qquad \vdash c : T_2 \Leftarrow T_1}{\Gamma \vdash \langle c \rangle e : T_2}$$

The translation from $\lambda^{\langle T \rangle}$ to $\lambda^{\langle c \rangle}$ is simple: just use $\langle\!\langle T \Leftarrow T \rangle\!\rangle^l$ to compile all casts into coercions.

We present a reduction semantics for $\lambda^{\langle c \rangle}$, using a few extra constructs:

simple values	s	::=	$k \mid \lambda x : T.\ e$
values	v	::=	$s \mid \langle \overline{c} \rangle s$
canonicals	w	::=	$v \mid \textbf{blame}\ l$
frames	f	::=	$\Box\ e \mid v\ \Box \mid \langle c \rangle \Box$
contexts	E	::=	$\Box \mid E\,[f]$

The reduction rules for $\lambda^{\langle c \rangle}$ follow:

$$(\lambda x : T.e)\ v \longrightarrow [v/x]\,e \tag{1}$$

$$(\langle c_1^w \to c_2^w \rangle s)\ v \longrightarrow \langle c_2^w \rangle (s\ (\langle c_1^w \rangle v)) \tag{2}$$

$$\langle c_1^w \rangle \langle \overline{c_2} \rangle s \longrightarrow \langle c_1^w \circ \overline{c_2} \rangle s \tag{3}$$

$$\langle \iota_B \rangle s \longrightarrow s \tag{4}$$

$$\langle c_1 \rangle s \longrightarrow \langle c_2^w \rangle s \ \text{ if } \ c_1 \longrightarrow^{o+} c_2^w \tag{5}$$

$$\langle \texttt{Fail}^l \rangle s \longrightarrow \textbf{blame}\ l \tag{6}$$

$$f\,[\textbf{blame}\ l] \longrightarrow \textbf{blame}\ l \tag{7}$$

The expression $f\,[\textbf{blame}\ l]$ indicates plugging the blame expression into the hole in the frame. For example, $(\Box\ e)\,[\textbf{blame}\ l] \equiv (\textbf{blame}\ l)\ e$. The single-step relation on programs is then defined in the standard way using evaluation contexts.

$$\frac{e_1 \longrightarrow e_2}{E\,[e_1] \longmapsto E\,[e_2]}$$

Rule (4) is specialized to ι_B, because there are no simple values s of type $\star$, to which $\iota_\star$ would apply. Rule (5) appeals to coercion calculus reduction (of one or more steps) to normalize a coercion wrapping a simple value s. After composing two coercions, we must normalize the resulting coercion to check if it represents a cast failure. In general, the operational semantics of $\lambda^{\langle c \rangle}$ are parameterized over the definition of coercions and their reduction. By varying that definition, we can change the blame-tracking strategy of the calculus, though one extra reduction rule is needed to support eager blame tracking (see Sec. 5). For simplicity, this semantics does not merge coercions in tail-call position as per Herman et al. [2007], so it is not yet fully space-efficient.

2.5 Space-efficient $\lambda^{\langle c \rangle}$

Full space-efficiency is achieved by composing coercions whenever possible, rather than waiting until a result is being returned. To achieve this, we use a richer notion of evaluation contexts.

$$\begin{array}{lcll} \text{plain frames} & f & ::= & \Box\, e \mid v\, \Box \\ \text{coerced contexts} & C & ::= & E\,[\langle c^w \rangle \Box] \\ \text{plain contexts} & E & ::= & \Box \mid E\,[f] \mid C\,[f] \end{array}$$

Then, reductions are split into *plain reductions* $e \longrightarrow e$ and *coercion reductions* $e \xrightarrow{\langle c \rangle} e$.

(1) $$(\lambda x : T.e)\ v \longrightarrow [v/x]\, e$$

(2) $$(\langle c_1^w \to c_2^w \rangle s)\ v \longrightarrow \langle c_2^w \rangle (s\ (\langle c_1^w \rangle v))$$

(3) $$\langle c_1^w \rangle \langle c_2 \rangle e \xrightarrow{\langle c \rangle} \langle c_1^w \circ c_2 \rangle e$$

(4) $$\langle \iota_B \rangle s \xrightarrow{\langle c \rangle} s$$

(5) $$\langle c \rangle e \xrightarrow{\langle c \rangle} \langle c^w \rangle e \ \text{ if } \ c \longrightarrow^{o+} c^w$$

(6) $$\langle \mathtt{Fail}^l \rangle s \xrightarrow{\langle c \rangle} \textbf{blame}\ l$$

(7) $$f\,[\textbf{blame}\ l] \longrightarrow \textbf{blame}\ l$$

Also, rules (5) and (3) are updated to apply for any coerced expression, not just for coerced simple values.

Finally, the single-step relation on programs is defined as three parts, rather than one:

$$\frac{e_1 \longrightarrow e_2}{E\,[e_1] \longmapsto E\,[e_2]} \qquad \frac{e_1 \longrightarrow e_2}{C\,[e_1] \longmapsto C\,[e_2]} \qquad \frac{e_1 \xrightarrow{\langle c \rangle} e_2}{E\,[e_1] \longmapsto E\,[e_2]}$$

According to these rules, plain reductions can happen in any context. Coercion reductions, on the other hand, can only happen in a *plain context*, i.e. a context that does not have a coercion immediately pending.

2.6 Coercions are space efficient

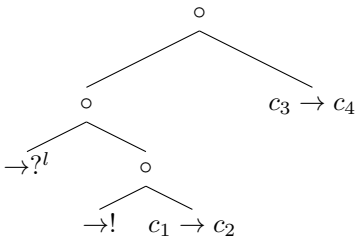
If we consider the even-odd example again, we can compile its casts to coercions: $\langle \star \Leftarrow \mathtt{Bool} \rangle^{l_1}$ becomes $\mathtt{Bool}!$, $\langle \mathtt{Bool} \to \mathtt{Bool} \Leftarrow \star \to \mathtt{Bool} \rangle^{l_2}$ becomes $\langle \mathtt{Bool}! \to \iota \rangle$, and $\langle \star \to \mathtt{Bool} \Leftarrow \mathtt{Bool} \to \mathtt{Bool} \rangle^{l_3}$ becomes $\langle \mathtt{Bool}?^{l_3} \to \iota \rangle$. We can see the space savings in the resulting trace:

$\mathtt{even(3,k)} \longrightarrow^*$
$\mathtt{odd(2,}\langle B! \to \iota \rangle \mathtt{k)} \longrightarrow^*$
$\mathtt{even(1,}\langle B! \circ B?^{l_3} \to \iota \rangle \mathtt{k)} \longrightarrow^*$
$\mathtt{odd(0,}\langle B! \to \iota \rangle \mathtt{k)} \longrightarrow \ldots$

Calling `even` with larger numbers repeats this sequence of the coercions without any further growth.

2.7 Implementing Coercions is Hard

Coercions cleanly and elegantly specify space-efficient cast semantics, which is one reason they were used to characterize the design space of casts in [Siek et al., 2009]. However, they are difficult to implement directly for two reasons. First, coercions are defined *up to associativity*, which can make some redexes harder to find, and can interfere with developing a deterministic reduction strategy; second some coercion compositions remain *inert*, meaning that they do not reduce, but composing with another coercion may warrant a reduction. These two properties interact with each other in troublesome ways. Consider the coercion expression diagrammed below:



The coercion expression $\to ! \circ (c_1 \to c_2)$ is inert, so it does not reduce. However, the coercions $\to ?^l$ and $\to !$ could reduce if composed directly, but in order to do so, the compositions must be reassociated first. The result, $\iota_\star \to \iota_\star$ could then compose with $c_1 \to c_2$. Meanwhile, the two arrow coercions $c_1 \to c_2$ and $c_3 \to c_4$ could also reduce, but are also blocked by how they are associated. One might try to devise an ad hoc reassociation scheme or represent a sequence of coercions as lists, but it would necessarily involve pairwise comparisons, bidirectional search, and splicing into the middle of complex coercion expressions. In short, it remains a challenge to design a redex search and reduction strategy that is complete in the face of inert compositions and associativity, yet simple enough to reason about and implement correctly. This compares quite poorly with threesomes, whose composition operation has a straightforward recursive implementation.

3 Threesomes for Lazy UD

This section presents a way to transform Lazy UD coercions into Lazy UD Threesomes using small easily verified steps. We begin by observing some key properties of coercions, then use those properties to overcome the challenges that associativity and inertness impose on implementing coercions. The strategy we apply and explain here is used later to produce similar results for other coercion calculi.

3.1 Properties of Coercion Reduction

The rest of this work is predicated on several common properties of coercion calculi. We briefly discuss those properties for Lazy UD here.

Proposition 1. *Reduction $\longrightarrow^\circ$ preserves typing.*

Proof. By cases on reduction. □

Proposition 2. *Reduction $\longrightarrow^\circ$ is confluent*

Proof. Among the reduction rules, there are only three pairs of rules that can interact with each other:

1. $\iota_P \circ c$ and $c \circ \iota_P$;
2. $\iota_P \circ c$ and $c \circ \texttt{Fail}^l$;
3. $\texttt{Fail}^l \circ \overline{c}$ with itself when $\overline{c} = C! \circ (c^w \to c^w)$ since we treat composition up to associativity.

In case of 1. and 2. each reduces to the same contractum: ι_P in the first case, $\texttt{Fail}^l$ in the second. In the case of 3., both reduction paths will arrive at $\texttt{Fail}^l$, one in one step and the other in two. □

Proposition 3 (Siek [2011a])**.** *Reduction* $\longrightarrow^\circ$ *is strongly normalizing.*

Proof. Consider as a size metric the lexicographic order on 1) the size of the coercion (raw number of nodes) and 2) the number of function coercions. Then coercion reduction decreases that metric. □

Proposition 4. *c is normal iff c is a* c^w.

Proof. The only-if part is proven by induction on the typing judgment $\vdash c : T_2 \Leftarrow T_1$. The only interesting case is $c_1 \circ c_2$ which can be proven by cases, using the induction hypothesis to deduce that $c_1 = c_1^w$ and $c_2 = c_2^w$.

The if part is proven by induction on the structure of c^w. □

These properties ensure there is a well-defined composition function for normal coercions.

Definition 1. *Let* $\textsc{NormC}^{T_2 \Leftarrow T_1} = \{\, c \mid \vdash c : T_2 \Leftarrow T_1 \,\}$. *Then define* normal composition *as follows:*

$$\odot : \textsc{NormC}^{T_3 \Leftarrow T_2} \times \textsc{NormC}^{T_2 \Leftarrow T_1} \to \textsc{NormC}^{T_3 \Leftarrow T_1}$$
$$c_1^w \odot c_2^w = c_3^w \text{ iff } c_1^w \circ c_2^w \longrightarrow^* c_3^w$$

Given this definition of composition for normal coercions, as well as the fact that cast compilation only produces normal coercions, we can update the definition of $\lambda^{\langle c \rangle}$ to only consider normal coercions and normal composition. Simply replace rules (5) and (3) with the rule:

$$\langle c_1^w \rangle \langle \overline{c_2} \rangle s \xrightarrow{\langle c \rangle} \langle c_1^w \odot \overline{c_2} \rangle s \tag{8a}$$

or in the fully space-efficient case:

$$\langle c_1^w \rangle \langle c_2^w \rangle s \longrightarrow \langle c_1^w \odot c_2^w \rangle s \tag{8b}$$

These rules perform both composition and normalization. However this change does not remove any dependence on the coercion calculus and its difficulties, since $\odot$ is defined in terms of coercion reduction: the composition function exists, but it's still hard to implement.

3.2 Supercoercions

As alluded to in the last section, the key to easily implementing coercions is to focus specifically on their normal forms. In doing so we can replace the characterization of $\odot$ above with an easily implemented notion of composition. Our first step in this process is to replace the set of normal coercion expressions c^w with a new set of composition-free coercions $\ddot{c}$, one for each form that a normal coercion can take (Fig. 2). We call them *supercoercions*, since a single supercoercion may stand for the composition of several traditional coercions.

Supercoercions correspond directly to normal coercions according to the following (reversible) mapping.

$\boxed{\mathcal{N}[\![\cdot]\!] : \textsc{SuperC} \to \textsc{NormC}}$ Supercoercion Meaning

$$\begin{array}{rcl}
\mathcal{N}[\![\iota_P]\!] & \equiv & \iota_P \\
\mathcal{N}[\![\mathtt{Fail}^l]\!] & \equiv & \mathtt{Fail}^l \\
\mathcal{N}[\![\mathtt{Fail}^{l_1 G l_2}]\!] & \equiv & \mathtt{Fail}^{l_1} \circ G?^{l_2} \\
\mathcal{N}[\![G!]\!] & \equiv & G! \\
\mathcal{N}[\![G?^l]\!] & \equiv & G?^l \\
\mathcal{N}[\![G?^l_!]\!] & \equiv & G! \circ G?^l \\
\mathcal{N}[\![\ddot{c}_1 \to \ddot{c}_2]\!] & \equiv & \mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!] \\
\mathcal{N}[\![\ddot{c}_{1 \to \uparrow} \ddot{c}_2]\!] & \equiv & \to! \circ (\mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!]) \\
\mathcal{N}[\![\ddot{c}_{1 \to \downarrow^l} \ddot{c}_2]\!] & \equiv & (\mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!]) \circ \to?^l \\
\mathcal{N}[\![\ddot{c}_{1 \to \updownarrow^l} \ddot{c}_2]\!] & \equiv & \to! \circ (\mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!]) \circ \to?^l
\end{array}$$

We call the mapping from normal coercions to supercoercions $\mathcal{S}[\![\cdot]\!]$, because later it will become different from the inverse of $\mathcal{N}[\![\cdot]\!]$. The type system for supercoercions was directly adapted from how the corresponding normal coercions are typed.

$l \in \text{LABELS}, \quad B \in \text{BASICTYPE}, \quad T \in \text{TYPES}, \quad \ddot{c} \in \text{SUPERC}$

$\text{SUPERC}^{T_2 \Leftarrow T_1} = \{\, \ddot{c} \mid \vdash \ddot{c} : T_2 \Leftarrow T_1 \,\}$

$$
\begin{array}{rcl}
G & ::= & B \mid \to \\
\ddot{c} & ::= & \texttt{Fail}^{l} \mid \texttt{Fail}^{lBl} \mid \texttt{Fail}^{l \to l} \mid G! \mid G?^{l} \mid G\updownarrow^{l} \mid \iota_P \mid \ddot{c} \to \ddot{c} \mid \ddot{c}_{\to\uparrow}\ddot{c} \mid \ddot{c}_{\to\downarrow^{l}}\ddot{c} \mid \ddot{c}_{\to\updownarrow^{l}}\ddot{c}
\end{array}
$$

$$
\frac{}{\vdash \iota_P : P \Leftarrow P} \qquad \frac{}{\vdash \texttt{Fail}^{l} : T_2 \Leftarrow T_1} \qquad \frac{}{\vdash \texttt{Fail}^{l_1 G l_2} : T \Leftarrow \star} \qquad \frac{}{\vdash B! : \star \Leftarrow B}
$$

$$
\frac{}{\vdash B?^{l} : B \Leftarrow \star} \qquad \frac{}{\vdash B\updownarrow^{l} : \star \Leftarrow \star} \qquad \frac{}{\vdash \to! : \star \Leftarrow \star \to \star} \qquad \frac{}{\vdash \to?^{l} : \star \to \star \Leftarrow \star} \qquad \frac{}{\vdash \to\updownarrow^{l} : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \to \ddot{c}_2 : T_1 \to T_4 \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : T_2 \Leftarrow \star \quad \vdash \ddot{c}_2 : \star \Leftarrow T_3}{\vdash \ddot{c}_{1\to\uparrow}\ddot{c}_2 : \star \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : \star \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow \star}{\vdash \ddot{c}_{1\to\downarrow^{l}}\ddot{c}_2 : T_1 \to T_4 \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : \star \Leftarrow \star \quad \vdash \ddot{c}_2 : \star \Leftarrow \star}{\vdash \ddot{c}_{1\to\updownarrow^{l}}\ddot{c}_2 : \star \Leftarrow \star}
$$

Figure 2: LazyUD Supercoercions

Composing supercoercions Based on our understanding of normal coercions, we can easily define composition for supercoercions. Moreover, that composition operation, which we call $\bullet$ is recursively defined. Much like the supercoercion type system, composition is straightforwardly calculated from the definition of composition for traditional coercions:take every pair of type-compatible supercoercions[4], translate them to normal coercions, normalize the result, and finally translate them back to supercoercions i.e.,

$$\ddot{c}_1 \bullet \ddot{c}_2 = \mathcal{S}[\![\mathcal{N}[\![\ddot{c}_1]\!] \odot \mathcal{N}[\![\ddot{c}_2]\!]]\!]$$

In doing so, we rely on the fact that normal arrow coercions compose in a syntax-directed manner.

Proposition 5.
$(c_1^w \to c_2^w) \odot (c_3^w \to c_4^w) = (c_3^w \odot c_1^w) \to (c_2^w \odot c_4^w)$

Proof. Straightforward. □

This fact gives us, for instance, that $(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) = (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4)$. Fig. 3 presents the full definition[5] of $\bullet$. The resulting function is well-defined by virtue of normalization (Prop. 3) and confluence (Prop. 2), which ensure that $\odot$ is well defined for normal coercions.

Version 1: Fig. 3.

Even though supercoercions are defined using normal coercions, they can be used to implement normalization for *arbitrary* coercion expressions. To see this, recall that all every atomic coercion is also a supercoercion, so we can compose them using $\bullet$. Combining this with the translation from normal arrow coercions, we get a translation from non-normal coercions to supercoercions:

$$
\begin{array}{rcl}
\mathcal{S}^\star : \text{COERCE}^{T_2 \Leftarrow T_1} \to \text{SUPERC}^{T_2 \Leftarrow T_1} & & \\
\mathcal{S}^\star[\![c]\!] & = & c \quad \text{if } c \text{ is atomic} \\
\mathcal{S}^\star[\![c_1 \circ c_2]\!] & = & \mathcal{S}^\star[\![c_1]\!] \bullet \mathcal{S}^\star[\![c_2]\!] \\
\mathcal{S}^\star[\![c_1 \to c_2]\!] & = & \mathcal{S}^\star[\![c_1]\!] \to \mathcal{S}^\star[\![c_2]\!]
\end{array}
$$

For example, the problematic coercion from Sec. 2.7 translates directly to its corresponding supercoercion (assuming $\ddot{c}_i = \mathcal{S}^\star[\![c_i]\!]$):

$$
\begin{aligned}
& (\to? \bullet (\to! \bullet (\ddot{c}_1 \to \ddot{c}_2))) \bullet (\ddot{c}_3 \to \ddot{c}_4) \\
=& (\to? \bullet (\ddot{c}_{1\to\uparrow}\ddot{c}_2)) \bullet (\ddot{c}_3 \to \ddot{c}_4) \\
=& (\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) \\
=& (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4)
\end{aligned}
$$

[4]To construct and verify the completeness of this definition, we formalized it and checked coverage using Twelf.

[5]Yes it's large, but don't panic! Things get simpler below.

$\bullet : \text{SUPERC}^{T_3 \Leftarrow T_2} \times \text{SUPERC}^{T_2 \Leftarrow T_1} \to \text{SUPERC}^{T_3 \Leftarrow T_1}$	Supercoercion Composition
$\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P$	$= \quad \ddot{c}$
$G! \bullet G?^l$	$= \quad G\updownarrow^l$
$B?^l \bullet B!$	$= \quad \iota_B$
$\to?^l \bullet \to!$	$= \quad \iota_\star \to \iota_\star$
$B?^{l_1} \bullet B\updownarrow^{l_2}$	$= \quad B?^{l_2}$
$\to?^{l_1} \bullet \to\updownarrow^{l_2}$	$= \quad \iota_\star \to\downarrow^{l_2} \iota_\star$
$B\updownarrow^l \bullet B!$	$= \quad B!$
$\to\updownarrow^l \bullet \to!$	$= \quad \iota_\star \to\uparrow \iota_\star$
$B\updownarrow^{l_1} \bullet B\updownarrow^{l_2}$	$= \quad B\updownarrow^{l_2}$
$\to\updownarrow^{l_1} \bullet \to\updownarrow^{l_2}$	$= \quad \iota_\star \to\updownarrow^{l_2} \iota_\star$
$(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to\downarrow^l \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to\downarrow^l (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to\uparrow \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to\uparrow (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to\uparrow \ddot{c}_2) \bullet (\ddot{c}_3 \to\downarrow^l \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to\updownarrow^l (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to\downarrow^l \ddot{c}_2) \bullet (\ddot{c}_3 \to\uparrow \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to\updownarrow^l \ddot{c}_2) \bullet (\ddot{c}_3 \to\uparrow \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to\uparrow (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to\downarrow^{l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to\updownarrow^{l_2} \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to\downarrow^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4)$
$(\ddot{c}_1 \to\updownarrow^{l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to\updownarrow^{l_2} \ddot{c}_4)$	$= \quad (\ddot{c}_3 \bullet \ddot{c}_1) \to\updownarrow^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4)$
$\to! \bullet (\ddot{c}_1 \to \ddot{c}_2)$	$= \quad (\ddot{c}_1 \to\uparrow \ddot{c}_2)$
$\to! \bullet (\ddot{c}_1 \to\downarrow^l \ddot{c}_2)$	$= \quad (\ddot{c}_1 \to\updownarrow^l \ddot{c}_2)$
$(\ddot{c}_1 \to \ddot{c}_2) \bullet \to?^l$	$= \quad (\ddot{c}_1 \to\downarrow^l \ddot{c}_2)$
$(\ddot{c}_1 \to\uparrow \ddot{c}_2) \bullet \to?^l$	$= \quad (\ddot{c}_1 \to\updownarrow^l \ddot{c}_2)$
$\to?^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2)$	$= \quad (\ddot{c}_1 \to \ddot{c}_2)$
$\to?^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2)$	$= \quad (\ddot{c}_1 \to\downarrow^{l_2} \ddot{c}_2)$
$\to\updownarrow^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2)$	$= \quad (\ddot{c}_1 \to\uparrow \ddot{c}_2)$
$\to\updownarrow^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2)$	$= \quad (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2)$
$(\ddot{c}_1 \to\downarrow^l \ddot{c}_2) \bullet \to!$	$= \quad (\ddot{c}_1 \to \ddot{c}_2)$
$(\ddot{c}_1 \to\updownarrow^l \ddot{c}_2) \bullet \to!$	$= \quad (\ddot{c}_1 \to\uparrow \ddot{c}_2)$
$(\ddot{c}_1 \to\downarrow^{l_1} \ddot{c}_2) \bullet \to\updownarrow^{l_2}$	$= \quad (\ddot{c}_1 \to\downarrow^{l_2} \ddot{c}_2)$
$(\ddot{c}_1 \to\updownarrow^{l_1} \ddot{c}_2) \bullet \to\updownarrow^{l_2}$	$= \quad (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2)$
$G_1?^l \bullet G_2! = G_1\updownarrow^l \bullet G_2!$	$= \quad \texttt{Fail}^l$ if $G_1 \neq G_2$
$B?^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2) = B\updownarrow^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2)$	$= \quad \texttt{Fail}^l$
$(\ddot{c}_1 \to\downarrow^l \ddot{c}_2) \bullet B! = (\ddot{c}_1 \to\updownarrow^l \ddot{c}_2) \bullet B!$	$= \quad \texttt{Fail}^l$
$G_1?^{l_1} \bullet G_2\updownarrow^{l_2} = G_1\updownarrow^{l_1} \bullet G_2\updownarrow^{l_2}$	$= \quad \texttt{Fail}^{l_1 G_2 l_2}$ if $G_1 \neq G_2$
$B?^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2) = B\updownarrow^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2)$	$= \quad \texttt{Fail}^{l_1 \to l_2}$
$(\ddot{c}_1 \to\downarrow^{l_1} \ddot{c}_2) \bullet B\updownarrow^{l_2} = (\ddot{c}_1 \to\updownarrow^{l_1} \ddot{c}_2) \bullet B\updownarrow^{l_2}$	$= \quad \texttt{Fail}^{l_1 B l_2}$
$\ddot{c} \bullet \texttt{Fail}^l$	$= \quad \texttt{Fail}^l$
$\ddot{c} \bullet \texttt{Fail}^{l_1 G l_2}$	$= \quad \texttt{Fail}^{l_1 G l_2}$
$\texttt{Fail}^l \bullet G! = \texttt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \texttt{Fail}^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2)$	$= \quad \texttt{Fail}^l$
$\texttt{Fail}^{l_1} \bullet G?^{l_2} = \texttt{Fail}^{l_1} \bullet G\updownarrow^{l_2}$	$= \quad \texttt{Fail}^{l_1 G l_2}$
$\texttt{Fail}^{l_1} \bullet (\ddot{c}_1 \to\downarrow^{l_2} \ddot{c}_2) = \texttt{Fail}^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2)$	$= \quad \texttt{Fail}^{l_1 \to l_2}$
$\texttt{Fail}^{l_1 G l_2} \bullet G! = \texttt{Fail}^{l_1 \to l_2} \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2)$	$= \quad \texttt{Fail}^{l_1}$
$\texttt{Fail}^{l_1 G l_2} \bullet G\updownarrow^{l_3}$	$= \quad \texttt{Fail}^{l_1 G l_3}$
$\texttt{Fail}^{l_1 \to l_2} \bullet (\ddot{c}_1 \to\updownarrow^{l_3} \ddot{c}_2)$	$= \quad \texttt{Fail}^{l_1 \to l_3}$
$\texttt{Fail}^{l_1 G_1 l_2} \bullet G_2! = \texttt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2)$	$= \quad \texttt{Fail}^{l_2}$ if $G_1 \neq G_2$
$\texttt{Fail}^{l_1 G_1 l_2} \bullet G_2\updownarrow^{l_3}$	$= \quad \texttt{Fail}^{l_2 G_2 l_3}$ if $G_1 \neq G_2$
$\texttt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \to\updownarrow^{l_3} \ddot{c}_2)$	$= \quad \texttt{Fail}^{l_2 \to l_3}$

Figure 3: Lazy UD Supercomposition, Version 1

Then, since we can always map supercoercions back to normal coercions, we immediately see that $\mathcal{N}[\![\mathcal{S}^\star[\![c]\!]]\!]$ implements a recursive normalizer for traditional coercions, overcoming our troubles with associativity and inertness. However, instead of translating back and forth, we can simply use supercoercions as our cast representation throughout. To do so, we update the translation from casts to directly produce supercoercions:

$$\begin{aligned}
\langle\!\langle \star \Leftarrow (T_3 \to T_4) \rangle\!\rangle^l &= (\star \to \star)! \bullet (\langle\!\langle T_3 \Leftarrow \star \rangle\!\rangle^l \to \langle\!\langle \star \Leftarrow T_4 \rangle\!\rangle^l) \\
&= \langle\!\langle T_3 \Leftarrow \star \rangle\!\rangle^l {}_{\to\uparrow} \langle\!\langle \star \Leftarrow T_4 \rangle\!\rangle^l \\
\langle\!\langle (T_1 \to T_2) \Leftarrow \star \rangle\!\rangle^l &= (\langle\!\langle \star \Leftarrow T_1 \rangle\!\rangle^l \to \langle\!\langle T_2 \Leftarrow \star \rangle\!\rangle^l) \bullet (\star \to \star)?^l \\
&= \langle\!\langle \star \Leftarrow T_1 \rangle\!\rangle^l {}_{\to\downarrow^l} \langle\!\langle T_2 \Leftarrow \star \rangle\!\rangle^l \\
&\text{if some } T_i \neq \star
\end{aligned}$$

The other cases are the same as for coercions. Note that source language casts never compile to $B^{\updownarrow}_{\updownarrow}$, $\ddot{c}_{\to\updownarrow}\ddot{c}$, or $\texttt{Fail}^{lGl}$: these supercoercions only arise at runtime.

3.3 Simplifications

Supercoercions, while effective, are not particularly succinct. They consist of 11 forms, including the original coercions. Furthermore, supercoercion composition requires 60 equations. This myriad of constructions and cases implies a large number of data constructor tags compared to the small number of traditional coercions, as well as a number of cases in the recursive composition definition. To address this blow-up, we apply a transformation-based approach to refine our large but correct system into a more streamlined but equivalent counterpart [Burstall and Darlington, 1975].

In this section we apply several simple transformations that together cut down the the set of coercions and equations dramatically. To our delight, the resulting simplified system is in fact nearly exactly the Threesomes of [Siek and Wadler, 2010]. As such, threesomes are not just an alternative to coercions, but literally a streamlined implementation strategy for them.

3.3.1 Primitive arrow coercions are redundant

Consider the set of primitive cast operators, B?, B!, and $B^{\updownarrow}_{\updownarrow}$, and the corresponding operators for arrow types. In the following two analogous equations,

$$\begin{aligned}
B?^l \bullet B! &= \iota_B \\
{\to}?^l \bullet {\to}! &= \iota_\star \to \iota_\star
\end{aligned}$$

the equation for base types produces a single ι_B, while the rule for arrows creates a complex arrow coercion. However, this complex coercion is operationally equivalent to an identity coercion at type $\star \to \star$.

One may notice similar correspondences when looking at the other combinations of these basic operators. For example, consider the equations

$$\begin{aligned}
B^{\updownarrow^l}_{\updownarrow} \bullet B! &= B! \\
{\to}^{\updownarrow^l}_{\updownarrow} \bullet {\to}! &= \iota_{\star\to\uparrow\iota_\star}
\end{aligned}$$

Does the $\to!$ coercion correspond to the $\iota_{\star\to\uparrow\iota_\star}$ coercion? Substituting for $\to!$ on the left side and using the definition of $\bullet$ gives

$${\to}^{\updownarrow^l}_{\updownarrow} \bullet (\iota_{\star\to\uparrow\iota_\star}) = \iota_{\star\to\uparrow\iota_\star}.$$

To confirm that this equivalence holds in general, we check (successfully) that replacing $\to!$ with $\iota_{\star\to\uparrow\iota_\star}$ in every equation preserves the result.

Applying the same logic to the other equations involving basic coercions, we find that the following transformations are sound:

1. Replace $\to!$ with $\iota_{\star\to\uparrow\iota_\star}$;
2. Replace $\to?^l$ with $\iota_{\star\to\downarrow^l\iota_\star}$;

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ — Supercoercion Composition

$$
\begin{array}{rcll}
\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P & = & \ddot{c} & \\
\\
\star\ B! \bullet B?^l & = & B\updownarrow^l & \\
B?^l \bullet B! & = & \iota_B & \\
B?^{l_1} \bullet B\updownarrow^{l_2} & = & B?^{l_2} & \\
B\updownarrow^l \bullet B! & = & B! & \\
B\updownarrow^{l_1} \bullet B\updownarrow^{l_2} & = & B\updownarrow^{l_2} & \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to\downarrow^l \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\downarrow^l (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to\uparrow \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\uparrow (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to\uparrow \ddot{c}_2) \bullet (\ddot{c}_3 \to\downarrow^l \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\updownarrow^l (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to\downarrow^l \ddot{c}_2) \bullet (\ddot{c}_3 \to\uparrow \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to\updownarrow^l \ddot{c}_2) \bullet (\ddot{c}_3 \to\uparrow \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\uparrow (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to\downarrow^{l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to\updownarrow^{l_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\downarrow^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4) & \\
(\ddot{c}_1 \to\updownarrow^{l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to\updownarrow^{l_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\updownarrow^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4) & \\
\\
\star\ B_1?^l \bullet B_2! = B_1\updownarrow^l \bullet B_2! & = & \texttt{Fail}^l \text{ if } B_1 \neq B_2 & \\
B?^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2) = B\updownarrow^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2) & = & \texttt{Fail}^l & \\
(\ddot{c}_1 \to\downarrow^l \ddot{c}_2) \bullet B! = (\ddot{c}_1 \to\updownarrow^l \ddot{c}_2) \bullet B! & = & \texttt{Fail}^l & \\
\\
\star\ B_1?^{l_1} \bullet B_2\updownarrow^{l_2} = B_1\updownarrow^{l_1} \bullet B_2\updownarrow^{l_2} & = & \texttt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 & \\
B?^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2) = B\updownarrow^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2) & = & \texttt{Fail}^{l_1 \to l_2} & \\
(\ddot{c}_1 \to\downarrow^{l_1} \ddot{c}_2) \bullet B\updownarrow^{l_2} = (\ddot{c}_1 \to\updownarrow^{l_1} \ddot{c}_2) \bullet B\updownarrow^{l_2} & = & \texttt{Fail}^{l_1 B l_2} & \\
\\
\ddot{c} \bullet \texttt{Fail}^l & = & \texttt{Fail}^l & \\
\ddot{c} \bullet \texttt{Fail}^{l_1 G l_2} & = & \texttt{Fail}^{l_1 G l_2} & \\
\\
\texttt{Fail}^l \bullet B! = \texttt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \texttt{Fail}^l \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2) & = & \texttt{Fail}^l & \\
\star\ \texttt{Fail}^{l_1} \bullet B?^{l_2} = \texttt{Fail}^{l_1} \bullet B\updownarrow^{l_2} & = & \texttt{Fail}^{l_1 B l_2} & \\
\texttt{Fail}^{l_1} \bullet (\ddot{c}_1 \to\downarrow^{l_2} \ddot{c}_2) = \texttt{Fail}^{l_1} \bullet (\ddot{c}_1 \to\updownarrow^{l_2} \ddot{c}_2) & = & \texttt{Fail}^{l_1 \to l_2} & \\
\\
\star\ \texttt{Fail}^{l_1 B l_2} \bullet B! = \texttt{Fail}^{l_1 \to l_2} \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2) & = & \texttt{Fail}^{l_1} & \\
\star\ \texttt{Fail}^{l_1 B l_2} \bullet B\updownarrow^{l_3} & = & \texttt{Fail}^{l_1 B l_3} & \\
\texttt{Fail}^{l_1 \to l_2} \bullet (\ddot{c}_1 \to\updownarrow^{l_3} \ddot{c}_2) & = & \texttt{Fail}^{l_1 \to l_3} & \\
\\
\star\ \texttt{Fail}^{l_1 G l_2} \bullet B! = \texttt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \to\uparrow \ddot{c}_2) & = & \texttt{Fail}^{l_2} \text{ if } G \neq B & \\
\star\ \texttt{Fail}^{l_1 G l_2} \bullet B\updownarrow^{l_3} & = & \texttt{Fail}^{l_2 B l_3} \text{ if } G \neq B & \\
\texttt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \to\updownarrow^{l_3} \ddot{c}_2) & = & \texttt{Fail}^{l_2 \to l_3} &
\end{array}
$$

Figure 4: LazyUD Supercomposition, Version 2

3. Replace $\to\updownarrow^l$ with $\iota_\star \to\updownarrow^l \iota_\star$.

We can thus remove these coercions and their $\bullet$-equations from further consideration. We update the translation from normal coercions to supercoercions, e.g.:

$$\mathcal{S}[\![\to?^l]\!] \equiv \iota_\star \to\downarrow^l \iota_\star.$$

For compilation from casts to these supercoercions,

$$\langle\!\langle \star \Leftarrow \star \to \star \rangle\!\rangle^l = \iota_\star \to\uparrow \iota_\star = \langle\!\langle \star \Leftarrow \star \rangle\!\rangle^l \to\uparrow \langle\!\langle \star \Leftarrow \star \rangle\!\rangle^l,$$

which has the same form as the equation for all other cases of $\langle\!\langle \star \Leftarrow T_1 \to T_2 \rangle\!\rangle^l$. Thus we can discard the special case for $\langle\!\langle \star \Leftarrow \star \to \star \rangle\!\rangle^l$.

One downside of this transformation is that $\iota_\star \to\uparrow \iota_\star$ is larger than a single $\to!$ coercion. Nevertheless, the former is likely to arise at runtime anyway, the size difference is minor, and the system still guarantees bounded space efficiency for casts.

Version 2: Fig. 4. Updated equations are marked with a red star ($\star$).

3.3.2 Merge base type coercions into type-polymorphic counterparts

The simple base type coercions can be grouped based on their indices. For instance, $B!$ and ι_B, are both indexed on a single base type B, while the coercions $B?^l$ and $B^{l}_{\updownarrow}$ are indexed on a base type B and a blame label l. Looking more closely at these pairings, we see that they can be differentiated based on their typing: for the first two, $\vdash B! : \star \Leftarrow B$ while $\vdash \iota_B : B \Leftarrow B$, while for the second two, $\vdash B?^l : B \Leftarrow \star$ while $\vdash B^{l}_{\updownarrow} : \star \Leftarrow \star$. As such, these four uni-typed coercions can be replaced with two "ad-hoc polymorphic" coercions so long as the type context in which they are used is fully known. We define two coercions which simply carry the indices from the prior coercions: B, and B^l. Each can be typed according to two typing rules:

$$\vdash B : \star \Leftarrow B \qquad\qquad \vdash B : B \Leftarrow B$$
$$\vdash B^l : B \Leftarrow \star \qquad\qquad \vdash B^l : \star \Leftarrow \star$$

Next, we define the meaning of these polymorphic supercoercions in terms of the original ones. Since the typings matter, however, we define the translation over type derivations $\mathcal{D}$:

$$\mathcal{N}' \left[\!\left[\overline{\vdash B : \star \Leftarrow B} \right]\!\right] = \overline{\vdash B! : \star \Leftarrow B}$$
$$\mathcal{N}' \left[\!\left[\overline{\vdash B : B \Leftarrow B} \right]\!\right] = \overline{\vdash \iota_B : B \Leftarrow B}$$
$$\mathcal{N}' \left[\!\left[\overline{\vdash B^l : B \Leftarrow \star} \right]\!\right] = \overline{\vdash B?^l : B \Leftarrow \star}$$
$$\mathcal{N}' \left[\!\left[\overline{\vdash B^l : \star \Leftarrow \star} \right]\!\right] = \overline{\vdash B^{l}_{\updownarrow} : \star \Leftarrow \star}\,.$$

$\mathcal{N}'[\![\cdot]\!]$ is defined recursively as the identity for the other supercoercion typing rules, which carry over directly to this new set of supercoercions. This simplification retains a direct correspondence, though now specifically between typing derivations for supercoercions. We consider this a first correctness criterion.

Proposition 6. *$\mathcal{N}'[\![\cdot]\!]$ is invertible.*

Proof. Immediate. □

In fact, the original transformation from normal coercions to supercoercions $\mathcal{N}[\![\cdot]\!]$ could be expressed in terms of typing derivations, but this would have been superfluous since every coercion corresponded to a unique derivation.

However, we haven't fully established correctness. Sharing indices is not sufficient grounds for reducing the original coercions to these new coercions. We must check that the equations for $\bullet$ are preserved by this interpretation. For example, consider the two supercoercion composition equations:

$$B! \bullet B?^l = B^{l}_{\updownarrow}, \qquad \iota_B \bullet B?^l = B?^l$$

Based on our new supercoercion interpretation, the analogous composition operator over the simpler supercoercions, $\bullet'$, needs only one equation to cover both of them:

$$B \bullet' B^l = B^l$$

However, since composition is defined over type-compatible pairs of coercions, it can in fact be viewed as defined over pairs of *typing derivations*. When viewed in this light, we recover the two equations over two different pairs of type derivations:

$$\overline{\vdash B : \star \Leftarrow B} \bullet' \overline{\vdash B^l : B \Leftarrow \star} = \overline{\vdash B^l : \star \Leftarrow \star}$$
$$\overline{\vdash B : B \Leftarrow B} \bullet' \overline{\vdash B^l : B \Leftarrow \star} = \overline{\vdash B^l : B \Leftarrow \star}$$

Viewed this way, the supercoercion derivations before this transformation correspond exactly to this new set of supercoercion type derivations. However, to our benefit, composition can be oblivious to the relevant typing derivations: the resulting coercion, B^l, stays the same. That means that we can erase the type derivations and still have a well-defined composition function. This is our second correctness criterion: that $\bullet$ is oblivious to type derivations. The following proposition states this more formally. In it, the notation $\mathcal{D} :: \vdash \ddot{c} : T_1 \Leftarrow T_2$ means that $\mathcal{D}$ is a derivation of $\vdash \ddot{c} : T_1 \Leftarrow T_2$.

Proposition 7. *Suppose that:*

1. $\mathcal{D}_1 :: \vdash \ddot{c}_1 : T_3 \Leftarrow T_2$*;*
2. $\mathcal{D'}_1 :: \vdash \ddot{c}_1 : T'_3 \Leftarrow T'_2$*;*
3. $\mathcal{D}_2 :: \vdash \ddot{c}_2 : T_2 \Leftarrow T_1$*; and*
4. $\mathcal{D'}_2 :: \vdash \ddot{c}_2 : T'_2 \Leftarrow T'_1$.

Then
$(\mathcal{D}_1 \bullet' \mathcal{D}_2) :: \vdash \ddot{c}_3 : T_3 \Leftarrow T_1$ *iff* $(\mathcal{D}'_1 \bullet' \mathcal{D}'_2) :: \vdash \ddot{c}_3 : T'_3 \Leftarrow T'_1$.

Proof. Straightforward induction on derivations. □

The important point is that the result of composition, $\ddot{c}_3$ stays the same regardless of the type derivation, so the $\bullet$ doesn't have to worry about the types. To verify that Prop. 7 holds, we simply substitute our new supercoercions into the old equations and verify that the result, at the relevant type, corresponds to the result in the original equation.

Note that since ι_B has been replaced in the new system, $\iota_\star$ is the only primitive identity coercion carried over from the original system. Thus, the relevant equations are specialized:

$$\ddot{c} \bullet' \iota_\star = \ddot{c}, \qquad \iota_\star \bullet' \ddot{c} = \ddot{c}$$

From here on, we drop the $'$s, and simply reuse $\bullet$ and $\ddot{c}$ at each simplification. We can update cast compilation to reflect this simplification. Notice that two different casts now compile to the same thing:

$$\langle\!\langle \star \Leftarrow B \rangle\!\rangle^l = \langle\!\langle B \Leftarrow B \rangle\!\rangle^l = B.$$

Version 3: Fig. 5.

3.3.3 Merge arrow coercions into type-polymorphic counterparts

The analogous compression can be applied to complex arrow coercions, replacing $\ddot{c} \to \ddot{c}$ and $\ddot{c}{\to}{\uparrow}\ddot{c}$ with $\ddot{c} \to \ddot{c}$ and replacing $\ddot{c}{\to}{\downarrow^l}\ddot{c}$ and $\ddot{c}{\to}{\updownarrow^l}\ddot{c}$ with $\ddot{c} \to^l \ddot{c}$. The typing rules for these new supercoercions are as expected, e.g.,:

$$\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \to \ddot{c}_2 : T_1 \to T_4 \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : \star \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow \star}{\vdash \ddot{c}_1 \to^l \ddot{c}_2 : T_1 \to T_4 \Leftarrow \star}$$

$$\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow \star \quad \vdash \ddot{c}_2 : \star \Leftarrow T_3}{\vdash \ddot{c}_1 \to \ddot{c}_2 : \star \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : \star \Leftarrow \star \quad \vdash \ddot{c}_2 : \star \Leftarrow \star}{\vdash \ddot{c}_1 \to^l \ddot{c}_2 : \star \Leftarrow \star}$$

This transformation is subject to the same correctness criteria as above and similarly decreases the set of supercoercions and equations for composition.

Version 4: Fig. 6.

3.3.4 Introduce Optional Data and Coalesce

After merging the base type coercions above, we end up with two distinct supercoercions, B, and B^l, which can also be differentiated on type. However, they can't be merged the same way because they do not both consist of the same subcomponents. With a change of representation we can smooth over this difference.

First, we produce a new form of coercion which features *optional* labels,

$$p ::= \epsilon \mid l$$

where ϵ indicates the absence of a label. Now we can merge both base type coercions into one coercion of the form B^p. The typing rules are as expected, e.g.:

$$\frac{}{\vdash B^\epsilon : \star \Leftarrow B} \qquad \frac{}{\vdash B^\epsilon : B \Leftarrow B}$$

$$\frac{}{\vdash B^l : B \Leftarrow \star} \qquad \frac{}{\vdash B^l : \star \Leftarrow \star}$$

$\bullet : \text{SUPERC}^{T_3 \Leftarrow T_2} \times \text{SUPERC}^{T_2 \Leftarrow T_1} \to \text{SUPERC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\star\ \iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
\star\ B \bullet B & = & B \\
\star\ B \bullet B^l & = & B^l \\
\star\ B^l \bullet B & = & B \\
\star\ B^{l_1} \bullet B^{l_2} & = & B^{l_2} \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to\!\downarrow^l \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\!\downarrow^l (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to\!\uparrow \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\!\uparrow (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to\!\uparrow \ddot{c}_2) \bullet (\ddot{c}_3 \to\!\downarrow^l \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\!\updownarrow^l (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to\!\downarrow^l \ddot{c}_2) \bullet (\ddot{c}_3 \to\!\uparrow \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to\!\updownarrow^l \ddot{c}_2) \bullet (\ddot{c}_3 \to\!\uparrow \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\!\uparrow (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to\!\downarrow^{l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to\!\updownarrow^{l_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\!\downarrow^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to\!\updownarrow^{l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to\!\updownarrow^{l_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to\!\updownarrow^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4) \\
\\
\star\ B_1^l \bullet B_2 & = & \texttt{Fail}^l \text{ if } B_1 \neq B_2 \\
\star\ B^l \bullet (\ddot{c}_1 \to\!\uparrow \ddot{c}_2) & = & \texttt{Fail}^l \\
\star\ (\ddot{c}_1 \to\!\downarrow^l \ddot{c}_2) \bullet B = (\ddot{c}_1 \to\!\updownarrow^l \ddot{c}_2) \bullet B & = & \texttt{Fail}^l \\
\\
\star\ B_1^{l_1} \bullet B_2^{l_2} & = & \texttt{Fail}^{l_1 B l_2} \text{ if } B_1 \neq B_2 \\
\star\ B^{l_1} \bullet (\ddot{c}_1 \to\!\updownarrow^{l_2} \ddot{c}_2) & = & \texttt{Fail}^{l_1 \to l_2} \\
\star\ (\ddot{c}_1 \to\!\downarrow^{l_1} \ddot{c}_2) \bullet B^{l_2} = (\ddot{c}_1 \to\!\updownarrow^{l_1} \ddot{c}_2) \bullet B^{l_2} & = & \texttt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \texttt{Fail}^l & = & \texttt{Fail}^l \\
\ddot{c} \bullet \texttt{Fail}^{l_1 G l_2} & = & \texttt{Fail}^{l_1 G l_2} \\
\\
\star\ \texttt{Fail}^l \bullet B = \texttt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \texttt{Fail}^l \bullet (\ddot{c}_1 \to\!\uparrow \ddot{c}_2) & = & \texttt{Fail}^l \\
\star\ \texttt{Fail}^{l_1} \bullet B^{l_2} & = & \texttt{Fail}^{l_1 B l_2} \\
\texttt{Fail}^{l_1} \bullet (\ddot{c}_1 \to\!\downarrow^{l_2} \ddot{c}_2) = \texttt{Fail}^{l_1} \bullet (\ddot{c}_1 \to\!\updownarrow^{l_2} \ddot{c}_2) & = & \texttt{Fail}^{l_1 \to l_2} \\
\\
\star\ \texttt{Fail}^{l_1 B l_2} \bullet B = \texttt{Fail}^{l_1 \to l_2} \bullet (\ddot{c}_1 \to\!\uparrow \ddot{c}_2) & = & \texttt{Fail}^{l_1} \\
\star\ \texttt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \texttt{Fail}^{l_1 B l_3} \\
\texttt{Fail}^{l_1 \to l_2} \bullet (\ddot{c}_1 \to\!\updownarrow^{l_3} \ddot{c}_2) & = & \texttt{Fail}^{l_1 \to l_3} \\
\\
\star\ \texttt{Fail}^{l_1 G l_2} \bullet B = \texttt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \to\!\uparrow \ddot{c}_2) & = & \texttt{Fail}^{l_2} \text{ if } G \neq B \\
\star\ \texttt{Fail}^{l_1 G l_2} \bullet B^{l_3} & = & \texttt{Fail}^{l_2 B l_3} \text{ if } G \neq B \\
\texttt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \to\!\updownarrow^{l_3} \ddot{c}_2) & = & \texttt{Fail}^{l_2 \to l_3}
\end{array}
$$

Figure 5: LazyUD Supercomposition, Version 3

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B \bullet B & = & B \\
B \bullet B^l & = & B^l \\
B^l \bullet B & = & B \\
B^{l_1} \bullet B^{l_2} & = & B^{l_2} \\
\star\ \ddot{c}_1 \to \ddot{c}_2 \bullet \ddot{c}_3 \to \ddot{c}_4 & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
\star\ \ddot{c}_1 \to \ddot{c}_2 \bullet \ddot{c}_3 \to^l \ddot{c}_4 & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to^l (\ddot{c}_2 \bullet \ddot{c}_4) \\
\star\ \ddot{c}_1 \to^l \ddot{c}_2 \bullet \ddot{c}_3 \to \ddot{c}_4 & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
\star\ \ddot{c}_1 \to^{l_1} \ddot{c}_2 \bullet \ddot{c}_3 \to^{l_2} \ddot{c}_4 & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to^{l_2} (\ddot{c}_2 \bullet \ddot{c}_4) \\
\\
B_1^l \bullet B_2 & = & \mathtt{Fail}^l \text{ if } B_1 \neq B_2 \\
\star\ B^l \bullet \ddot{c}_1 \to \ddot{c}_2 & = & \mathtt{Fail}^l \\
\star\ \ddot{c}_1 \to^l \ddot{c}_2 \bullet B & = & \mathtt{Fail}^l \\
\\
B_1^{l_1} \bullet B_2^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star\ B^{l_1} \bullet \ddot{c}_1 \to^{l_2} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 \to l_2} \\
\star\ \ddot{c}_1 \to^{l_1} \ddot{c}_2 \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\\
\star\ \mathtt{Fail}^l \bullet B = \mathtt{Fail}^l \bullet \ddot{c}_1 \to \ddot{c}_2 & = & \mathtt{Fail}^l \\
\mathtt{Fail}^{l_1} \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star\ \mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \to^{l_2} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 \to l_2} \\
\\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet B = \mathtt{Fail}^{l_1 \to l_2} \bullet \ddot{c}_1 \to \ddot{c}_2 & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\star\ \mathtt{Fail}^{l_1 \to l_2} \bullet \ddot{c}_1 \to^{l_3} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 \to l_3} \\
\\
\star\ \mathtt{Fail}^{l_1 G_1 l_2} \bullet B_2 = \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \to \ddot{c}_2 & = & \mathtt{Fail}^{l_2} \text{ if } G_1 \neq B_2 \\
\mathtt{Fail}^{l_1 G_1 l_2} \bullet B_2^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } G_1 \neq B_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \to^{l_3} \ddot{c}_2 & = & \mathtt{Fail}^{l_2 \to l_3}
\end{array}
$$

Figure 6: LazyUD Supercomposition, Version 4

$\bullet : \text{SUPERC}^{T_3 \Leftarrow T_2} \times \text{SUPERC}^{T_2 \Leftarrow T_1} \to \text{SUPERC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
\star \; B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
\star \; (\ddot{c}_1 \to^{p_1} \ddot{c}_2) \bullet (\ddot{c}_3 \to^{p_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to^{p_2} (\ddot{c}_2 \bullet \ddot{c}_4) \\
\\
\star \; Q_1^l \bullet Q_2^\epsilon & = & \texttt{Fail}^l \text{ if } Q_1 \to\!\!\leftarrow Q_2 \\
\\
\star \; Q_1^{l_1} \bullet Q_2^{l_2} & = & \texttt{Fail}^{l_1 \lceil Q_2 \rceil l_2} \text{ if } Q_1 \to\!\!\leftarrow Q_2 \\
\\
\ddot{c} \bullet \texttt{Fail}^l & = & \texttt{Fail}^l \\
\ddot{c} \bullet \texttt{Fail}^{l_1 G l_2} & = & \texttt{Fail}^{l_1 G l_2} \\
\\
\star \; \texttt{Fail}^l \bullet Q^\epsilon & = & \texttt{Fail}^l \\
\star \; \texttt{Fail}^{l_1} \bullet Q^{l_2} & = & \texttt{Fail}^{l_1 \lceil Q \rceil l_2} \\
\\
\star \; \texttt{Fail}^{l_1 \lceil Q \rceil l_2} \bullet Q^\epsilon & = & \texttt{Fail}^{l_1} \\
\star \; \texttt{Fail}^{l_1 \lceil Q \rceil l_2} \bullet Q^{l_3} & = & \texttt{Fail}^{l_1 \lceil Q \rceil l_3} \\
\\
\star \; \texttt{Fail}^{l_1 G_1 l_2} \bullet Q_2^\epsilon & = & \texttt{Fail}^{l_2} \text{ if } \lceil Q_2 \rceil \neq G_1 \\
\star \; \texttt{Fail}^{l_1 G_1 l_2} \bullet Q_2^{l_3} & = & \texttt{Fail}^{l_2 \lceil Q_2 \rceil l_3} \text{ if } \lceil Q_2 \rceil \neq G_1
\end{array}
$$

Figure 7: LazyUD Supercomposition, Version 5

Similarly, we can merge arrow casts into the form $\ddot{c} \to^p \ddot{c}$. We can now refer to atomic and arrow coercions uniformly

$$Q := B \mid \ddot{c} \to \ddot{c}$$

Then Q^p is essentially an optionally labeled type constructor: either an atomic B^p or an arrow $\ddot{c} \to^p \ddot{c}$. At this point we are close to uncovering our labeled types. In direct analogy with non-$\star$ types, we use $\lceil Q \rceil$ to get Q's topmost constructor, and $Q_1 \to\!\!\leftarrow Q_2$ to say that two coercions have incompatible top-level constructors.

Introducing optional labels further simplifies our definition of composition. For example, the four equations:

$$B \bullet B = B, \qquad B \bullet B^l = B^l,$$
$$B^l \bullet B = B, \qquad B^{l_1} \bullet B^{l_2} = B^{l_2}$$

Become one equation:

$$B^{p_1} \bullet B^{p_2} = B^{p_2}$$

Notice that the function need not even inspect p_2: it simply copies it over to the result. The same thing happens with arrow coercions.

Version 5: Fig. 7.

Using the same strategy, we merge the $\texttt{Fail}^l$ and $\texttt{Fail}^{lGl}$ forms by making the G and l pair into an optional failure "type" annotation.

$$H ::= \epsilon \mid Gl$$
$$\ddot{c} ::= \ldots \mid \texttt{Fail}^{lH}$$

To complete our simplification, we define a helper function $[\cdot]$ that translates a labeled type Q^p into an optional failure annotation.

$$[Q^\epsilon] = \epsilon$$
$$\left[B^l\right] = Bl$$
$$\left[\ddot{c}_1 \to^l \ddot{c}_2\right] = \to l$$

This helper lets us take advantage of both labeled non-$\star$ types Q^p and labeled failure types together. This completes our simplification process, yielding the labeled types and composition operator of Fig. 8.

The resulting labeled types closely mirror those of [Siek and Wadler, 2010], but for a few cosmetic differences:

$l \in \textsc{Labels}, \quad B \in \textsc{BasicType}, \quad T \in \textsc{Types}, \quad \ddot{c} \in \textsc{SuperC}$

$\textsc{SuperC}^{T_2 \Leftarrow T_1} = \{\, \ddot{c} \mid \vdash \ddot{c} : T_2 \Leftarrow T_1 \,\}$

$$
\begin{array}{rcl}
G & ::= & B \mid \rightarrow \\
H & ::= & \epsilon \mid Gl \\
p & ::= & \epsilon \mid l \\
Q^p & ::= & B^p \mid \ddot{c} \rightarrow^p \ddot{c} \\
\ddot{c} & ::= & \iota_\star \mid \texttt{Fail}^{lH} \mid Q^p
\end{array}
$$

$$
\frac{}{\vdash \iota_\star : \star \Leftarrow \star}
\qquad
\frac{}{\vdash \texttt{Fail}^{l^\epsilon} : T_2 \Leftarrow T_1}
\qquad
\frac{}{\vdash \texttt{Fail}^{l_1 G l_2} : T \Leftarrow \star}
$$

$$
\frac{}{\vdash B^\epsilon : B \Leftarrow B}
\qquad
\frac{}{\vdash B^\epsilon : \star \Leftarrow B}
\qquad
\frac{}{\vdash B^l : B \Leftarrow \star}
\qquad
\frac{}{\vdash B^l : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \rightarrow^\epsilon \ddot{c}_2 : T_1 \rightarrow T_4 \Leftarrow T_2 \rightarrow T_3}
\qquad
\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow \star \quad \vdash \ddot{c}_2 : \star \Leftarrow T_3}{\vdash \ddot{c}_1 \rightarrow^\epsilon \ddot{c}_2 : \star \Leftarrow T_2 \rightarrow T_3}
\qquad
\frac{\vdash \ddot{c}_1 : \star \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow \star}{\vdash \ddot{c}_1 \rightarrow^l \ddot{c}_2 : T_1 \rightarrow T_4 \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : \star \Leftarrow \star \quad \vdash \ddot{c}_2 : \star \Leftarrow \star}{\vdash \ddot{c}_1 \rightarrow^l \ddot{c}_2 : \star \Leftarrow \star}
$$

$\boxed{\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \rightarrow \textsc{SuperC}^{T_3 \Leftarrow T_1}}$ Composition

$$
\begin{array}{rcll}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} & \\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} & \\
(\ddot{c}_1 \rightarrow^{p_1} \ddot{c}_2) \bullet (\ddot{c}_3 \rightarrow^{p_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \rightarrow^{p_2} (\ddot{c}_2 \bullet \ddot{c}_4) & \\
\star \quad Q_1^{l_1} \bullet Q_2^{p_2} & = & \texttt{Fail}^{l_1 \lceil Q_2^{p_2} \rceil} & \text{if } Q_1 \rightarrow\!\!\leftarrow Q_2 \\
\star \quad \ddot{c} \bullet \texttt{Fail}^{l_1 H} & = & \texttt{Fail}^{l_1 H} & \\
\star \quad \texttt{Fail}^{l} \bullet Q^p & = & \texttt{Fail}^{l[Q^p]} & \\
\star \quad \texttt{Fail}^{l_1 \lceil Q \rceil l_2} \bullet Q^{p_3} & = & \texttt{Fail}^{l_1[Q^{p_3}]} & \\
\star \quad \texttt{Fail}^{l_1 G l_2} \bullet Q^p & = & \texttt{Fail}^{l_2[Q^p]} & \text{if } \lceil Q \rceil \neq G
\end{array}
$$

where

$$
\begin{array}{rcl}
\lceil Q^\epsilon \rceil & = & \epsilon \\
\left\lceil B^l \right\rceil & = & Bl \\
\left\lceil \ddot{c}_1 \rightarrow^l \ddot{c}_2 \right\rceil & = & \rightarrow l
\end{array}
\qquad
\begin{array}{rcl}
\lceil B \rceil & = & B \\
\lceil \ddot{c} \rightarrow \ddot{c} \rceil & = & \rightarrow .
\end{array}
$$

Figure 8: LazyUD Labeled Types

1. In general $\ddot{c}$ corresponds directly to labeled types P
2. Our $\iota_\star$ corresponds to the labeled type $*$;
3. The B^p and $\ddot{c} \to^p \ddot{c}$ coercions are the same;
4. $\texttt{Fail}^{lH}$ corresponds to $\bot^{lGp}$, with H playing the role of Gp. In the original Threesomes, only the label p was optional. Our derivation reveals, however, that G is only needed if an actual label is present.

3.3.5 Adapting $\lambda^{\langle c \rangle}$ to Threesomes

Having recast coercions first as supercoercions and then as labeled types, we return to implementing the cast calculus using them. The resulting system is a direct remapping of $\lambda^{\langle c \rangle}$, but now with labeled types. As we see, the resulting calculus, called $\lambda^{\langle \ddot{c} \rangle}$, needs type information for the labeled types. As such, we take our labeled types $\vdash \ddot{c} : T_2 \Leftarrow T_1$ and present them as threesomes $\langle T_2 \overset{\ddot{c}}{\Longleftarrow} T_1 \rangle$. Thus we see that $\lambda^{\langle \ddot{c} \rangle}$ is a threesome calculus. The language replaces coercions from $\lambda^{\langle c \rangle}$ with threesomes. The type rule for casted expressions is as follows:

$$\frac{\vdash \ddot{c} : T_2 \Leftarrow T_1 \qquad \Gamma \vdash e : T_1}{\Gamma \vdash \langle T_2 \overset{\ddot{c}}{\Longleftarrow} T_1 \rangle\, e : T_2}$$

Its reduction rules are as follows:

$$(\lambda x : T.e)\ v \longrightarrow [v/x]\, e \tag{1}$$

$$(\langle T_1 \to T_2 \overset{\ddot{c}_1 \overset{\epsilon}{\to} \ddot{c}_2}{\Longleftarrow} T_3 \to T_4 \rangle\ s)\ v \longrightarrow \langle T_2 \overset{\ddot{c}_2}{\Longleftarrow} T_4 \rangle\ (s\ (\langle T_3 \overset{\ddot{c}_1}{\Longleftarrow} T_1 \rangle\ v)) \tag{2}$$

$$\langle B \overset{B^\epsilon}{\Longleftarrow} B \rangle\ s \longrightarrow s \tag{4}$$

$$\langle T_2 \overset{\texttt{Fail}^{l\epsilon}}{\Longleftarrow} T_1 \rangle\ s \longrightarrow \textbf{blame}\ l \tag{6}$$

$$f\ [\textbf{blame}\ l] \longrightarrow \textbf{blame}\ l \tag{7}$$

$$\langle T_3 \overset{\ddot{c}_1}{\Longleftarrow} T_2 \rangle\ \langle T_2 \overset{Q^\epsilon}{\Longleftarrow} T_1 \rangle\ s \longrightarrow \langle T_3 \overset{\ddot{c}_1 \bullet Q^\epsilon}{\Longleftarrow} T_1 \rangle\ s \quad \text{if } (Q, T_1, T_2) \neq (B, B, B) \tag{8}$$

To produce these rules, we simply take the original $\lambda^{\langle c \rangle}$ rules, replace rule (3) and (5) with (8), as discussed in Sec. 3.1, and reconstitute the coercion-based rules with their direct threesome analogues, according to our correspondence. The type information provided by the outer types is specifically needed to support rule (4). In particular, we need to differentiate $\langle B \overset{B^\epsilon}{\Longleftarrow} B \rangle\ s$, which corresponds to $\langle \iota_B \rangle s$ and is a redex, from $\langle \star \overset{B^\epsilon}{\Longleftarrow} B \rangle\ s$, which corresponds to $\langle B! \rangle s$ and is a $\star$-wrapped value. Despite the need for type information, the polymorphic simplifications above are still a win because rules (4) and (8) are the only reduction rules that need ever inspect an outer type. The rest simply pass along components without inspection. Furthermore, the side condition on the rule (8) is only there to ensure deterministic reduction. Without it, reduction is nondeterministic but always produces the same result.

We can similarly reconstitute the fully space-efficient $\lambda^{\langle c \rangle}$ as a threesome calculus, after combining rules (3) and (5) as before.

$$(\lambda x : T.e)\ v \longrightarrow [v/x]\, e \tag{1}$$

$$(\langle T_1 \to T_2 \overset{\ddot{c}_1 \overset{\epsilon}{\to} \ddot{c}_2}{\Longleftarrow} T_3 \to T_4 \rangle\ s)\ v \longrightarrow \langle T_2 \overset{\ddot{c}_2}{\Longleftarrow} T_4 \rangle\ (s\ (\langle T_3 \overset{\ddot{c}_1}{\Longleftarrow} T_1 \rangle\ v)) \tag{2}$$

$$\langle B \overset{B^\epsilon}{\Longleftarrow} B \rangle\ s \xrightarrow{\langle c \rangle} s \tag{4}$$

$$\langle \texttt{Fail}^l \rangle s \xrightarrow{\langle c \rangle} \textbf{blame}\ l \tag{6}$$

$$f\ [\textbf{blame}\ l] \longrightarrow \textbf{blame}\ l \tag{7}$$

$$\langle T_3 \overset{\ddot{c}_1}{\Longleftarrow} T_2 \rangle\ \langle T_2 \overset{\ddot{c}_2}{\Longleftarrow} T_1 \rangle\ s \xrightarrow{\langle c \rangle} \langle T_3 \overset{\ddot{c}_1 \bullet \ddot{c}_2}{\Longleftarrow} T_1 \rangle\ s \tag{8}$$

In this variant, rule (8) has no side condition: it simply combines strings of labeled types as needed.

To support compilation from $\lambda^{\langle T \rangle}$ to $\lambda^{\langle \ddot{c} \rangle}$, we update cast compilation to produce labeled types. Doing so simply involves composing the original cast compilation function with our final $\mathcal{S}[\![\cdot]\!]$ function.

$$\begin{aligned}
\langle\!\langle \star \Leftarrow \star \rangle\!\rangle^l &= \iota_\star \\
\langle\!\langle \star \Leftarrow B \rangle\!\rangle^l = \langle\!\langle B \Leftarrow B \rangle\!\rangle^l &= B^\epsilon \\
\langle\!\langle B \Leftarrow \star \rangle\!\rangle^l &= B^l \\
\langle\!\langle T_1 \to T_2 \Leftarrow T_3 \to T_4 \rangle\!\rangle^l &= \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \to^\epsilon \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \\
\langle\!\langle \star \Leftarrow T_1 \to T_2 \rangle\!\rangle^l &= \langle\!\langle T_1 \Leftarrow \star \rangle\!\rangle^l \to^\epsilon \langle\!\langle \star \Leftarrow T_2 \rangle\!\rangle^l \\
\langle\!\langle T_1 \to T_2 \Leftarrow \star \rangle\!\rangle^l &= \langle\!\langle \star \Leftarrow T_1 \rangle\!\rangle^l \to^l \langle\!\langle T_2 \Leftarrow \star \rangle\!\rangle^l \\
\langle\!\langle T_1 \Leftarrow T_2 \rangle\!\rangle^l &= \texttt{Fail}^{l\epsilon} \text{ if } T_1 \to\!\!\leftarrow T_2
\end{aligned}$$

Then, a source cast $\langle T_2 \Leftarrow T_1 \rangle^l$ is translated to the threesome $\langle T_2 \overset{\langle\!\langle T_2 \Leftarrow T_1 \rangle\!\rangle^l}{\Longleftarrow} T_1 \rangle$.

Alternative: Late Unwrapping As an alternative to threesomes with outer types, one can restructure the $\lambda^{\langle c \rangle}$ calculus so that the labeled types alone are sufficient. The key insight to achieve this is that discharging the ι_B case only matters when a base type value is observed. $\lambda^{\langle c \rangle}$ has no constructs for observing base type values, but suppose an operator $f(k)$ that consumes base-type values. In the original coercion calculus, we could have chosen to delay the unwrapping of the casted expression $\langle \iota_B \rangle k$ until f is applied to it. Carrying this idea forward to $\lambda^{\langle \ddot{c} \rangle}$, we present a new semantics called Latent $\lambda^{\langle \ddot{c} \rangle}$, which embodies this approach. The language has as its cast equivalents expressions $\langle \ddot{c} \rangle e$, and delays unwrapping all B^ϵ coercions.

(1) $$(\lambda x : T.e)\ v \longrightarrow [v/x]\, e$$

(2) $$(\langle \ddot{c}_1 \to^\epsilon \ddot{c}_2 \rangle s)\ v \longrightarrow \langle \ddot{c}_2 \rangle (s\ (\langle \ddot{c}_1 \rangle v))$$

(4) $$f(\langle B^\epsilon \rangle s) \longrightarrow f(s)$$

(6) $$\langle \texttt{Fail}^{l\epsilon} \rangle s \longrightarrow \textbf{blame}\ l$$

(7) $$f\ [\textbf{blame}\ l] \longrightarrow \textbf{blame}\ l$$

(8) $$\langle \ddot{c}_1 \rangle \langle \ddot{c}_2 \rangle s \longrightarrow \langle \ddot{c}_1 \bullet \ddot{c}_2 \rangle s$$

Note that a B^ϵ wrapper may be composed with another coercion via the third reduction rule. In well-typed programs, this always yields the coercion $\ddot{c}_1$ as the result, in accordance with ι_B's semantics.

3.3.6 Summary

Starting with the Lazy UD coercion calculus, we produced a notion of supercoercions, which provides an effective implementation of coercion reduction at the cost of a large number of constructions and cases. Then, by simple stepwise refinement, we have transformed the supercoercions and composition equations into a much smaller system of labeled types, with 3 constructions, 9 equations, and 2 simple helper functions. In this system, every labeled type except for $\iota_\star$ represents multiple supercoercions. Each simplification took a small step and was easy verify.

Re-constructing the Threesome Calculus sheds new light on the relationship between Lazy UD coercions and threesomes, but the key benefit of this process is that it gave us a strategy for developing threesomes corresponding to new blame tracking strategies. In the remainder of this paper, we retrace the same steps to develop new threesome semantics.

4 Lazy D Threesomes

The Lazy D blame strategy retains the laziness of the Blame Calculus (i.e., Lazy UD), but distributes blame in a manner that's consistent with the traditional notion of subtyping, treating $\star$ as the top of the subtype hierarchy. In this section we develop threesomes that implement the Lazy D strategy. To get a Lazy D threesome calculus, simply replace the Lazy UD labeled types in $\lambda^{\langle \ddot{c} \rangle}$ with their Lazy D counterparts.

$$
\begin{array}{rcl}
G & ::= & B \mid T \to T \\
P & ::= & B \mid \star \\
\ddot{c} & ::= & \mathtt{Fail}^{l} \mid \mathtt{Fail}^{lGl} \mid G! \mid G?^{l} \mid G^{\updownarrow^{l}}_{\updownarrow} \mid \iota_P \mid \ddot{c} \to \ddot{c} \\
& \mid & \ddot{c} \to^{(T\to T)l}_{\downarrow} \ddot{c} \mid \ddot{c} \,{}^{T\to T}_{\uparrow}\!\to \ddot{c} \mid \ddot{c} \,{}^{T\to T}_{\uparrow}\!\to^{(T\to T)l}_{\downarrow} \ddot{c}
\end{array}
$$

$$
\frac{}{\vdash \mathtt{Fail}^{l} : T_2 \Leftarrow T_1} \qquad \frac{}{\vdash \mathtt{Fail}^{l_1 G l_2} : T \Leftarrow \star} \qquad \frac{}{\vdash G! : \star \Leftarrow G} \qquad \frac{}{\vdash G?^{l} : G \Leftarrow \star} \qquad \frac{}{\vdash G^{\updownarrow^{l}}_{\updownarrow} : \star \Leftarrow \star}
$$

$$
\frac{}{\vdash \iota_P : P \Leftarrow P} \qquad \frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \to \ddot{c}_2 : T_1 \to T_4 \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \,{}^{T_1\to T_4}_{\uparrow}\!\to^{(T_2\to T_3)l}_{\downarrow} \ddot{c}_2 : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \,{}^{T_1\to T_4}_{\uparrow}\!\to \ddot{c}_2 : \star \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1 \to^{(T_2\to T_3)l}_{\downarrow} \ddot{c}_2 : T_1 \to T_4 \Leftarrow \star}
$$

Figure 9: Lazy D Supercoercions

The Lazy D coercion calculus extends the Lazy UD calculus by adding primitive injections and projections at every type

$$c ::= ... \mid T?^{l} \mid T!$$

In fact, the UD coercions $\to!$ and $\to?^{l}$ correspond exactly to the $(\star \to \star)!$ and $(\star \to \star)?^{l}$ coercions.

In Lazy UD, a cast between a function type and $\star$ compiles to a complex coercion to $\star \to \star$ combined with a primitive cast between $\star$ and $\star \to \star$. Under Lazy D, in contrast, this is replaced with primitive coercions:

$$\langle\!\langle \star \Leftarrow T_1 \to T_2 \rangle\!\rangle^{l} = (T_1 \to T_2)!$$
$$\langle\!\langle T_1 \to T_2 \Leftarrow \star \rangle\!\rangle^{l} = (T_1 \to T_2)?^{l}$$

Now the translations for all injections to and projections from $\star$ have the same structure. The reduction rules for Lazy D must address how complex injections interact with complex projections. This is handled by the following reduction rule:

$$T_1?^{l} \circ T_2! \longrightarrow \langle\!\langle T_1 \Leftarrow T_2 \rangle\!\rangle^{l}$$

This rule subsumes all of the Lazy UD rules that involve injections meeting projections. For Lazy D, however, this rule must be stated in this general format because there are an unbounded number of them. In Lazy UD, the rules can be specialized with respect to the structure of the coercions involved.

The syntax of normal Lazy D coercions differs from Lazy UD specifically in that the primitive cast indices G are extended to be exactly the non-$\star$ types:

$$G ::= B \mid T \to T$$

Lazy D Supercoercions Taking the above differences into account, we can define a set of Lazy D supercoercions (Fig. 9):

The main difference in the Lazy D calculus is that primitive casts are indexed by full-fledged types and the complex arrow coercions have additional type indices. The origin of these arrow-coercion type indices can be seen by looking at their translation back to normal Lazy D coercions:

$$\mathcal{N}[\![\ddot{c}_1 \to^{(T_1\to T_2)l}_{\downarrow} \ddot{c}_2]\!] = (\mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!]) \circ (T_1 \to T_2)?^{l}$$
$$\mathcal{N}[\![\ddot{c}_1 \,{}^{T_1\to T_2}_{\uparrow}\!\to \ddot{c}_2]\!] = (T_1 \to T_2)! \circ (\mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!])$$
$$\mathcal{N}[\![\ddot{c}_1 \,{}^{T_1\to T_2}_{\uparrow}\!\to^{(T_3\to T_4)l}_{\downarrow} \ddot{c}_2]\!] = (T_1 \to T_2)! \circ (\mathcal{N}[\![\ddot{c}_1]\!] \to \mathcal{N}[\![\ddot{c}_2]\!]) \circ (T_3 \to T_4)?^{l}$$

The type indices on arrow coercions correspond to leading projections and trailing injections between arrow types and $\star$. Now that any arrow type can be immediately projected to or injected from $\star$, we must keep track of which type is specifically intended. This affects how blame propagates among coercions.

$\bullet : \text{SUPERC}^{T_3 \Leftarrow T_2} \times \text{SUPERC}^{T_2 \Leftarrow T_1} \to \text{SUPERC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P & = & \ddot{c} \\
\\
G! \bullet G?^{l} & = & G\updownarrow^{l} \\
\\
B?^{l} \bullet B! & = & \iota_B \\
(T_1 \to T_2)?^{l} \bullet (T_3 \to T_4)! & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \to \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \\
B?^{l_1} \bullet B\updownarrow^{l_2} & = & B?^{l_2} \\
(T_1 \to T_2)?^{l_1} \bullet (T_3 \to T_4)\updownarrow^{l_2} & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1} \to^{(T_3 \to T_4) l_2}_{\downarrow} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1} \\
B\updownarrow^{l_1} \bullet B! & = & B! \\
(T_1 \to T_2)\updownarrow^{l} \bullet (T_3 \to T_4)! & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \\
B\updownarrow^{l_1} \bullet B\updownarrow^{l_2} & = & B\updownarrow^{l_2} \\
(T_1 \to T_2)\updownarrow^{l_1} \bullet (T_3 \to T_4)\updownarrow^{l_2} & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4) l_2}{\underset{\downarrow}{\to}} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to^{(T_1 \to T_2) l}_{\downarrow} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to^{(T_1 \to T_2) l}_{\downarrow} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \to^{(T_3 \to T_4) l}_{\downarrow} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4) l}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to^{(T_1 \to T_2) l}_{\downarrow} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \to^{(T_3 \to T_4) l}_{\downarrow} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to^{(T_1 \to T_2) l}_{\downarrow} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \to^{(T_5 \to T_6) l}_{\downarrow} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \to^{(T_5 \to T_6) l}_{\downarrow} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \to^{(T_3 \to T_4) l_1}_{\downarrow} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \to^{(T_7 \to T_8) l_2}_{\downarrow} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_7 \to T_8) l_2}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
\\
(T_1 \to T_2)! \bullet (\ddot{c}_1 \to \ddot{c}_2) & = & \ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2 \\
(T_1 \to T_2)! \bullet (\ddot{c}_1 \to^{(T_3 \to T_4) l}_{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4) l}{\underset{\downarrow}{\to}} \ddot{c}_2 \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (T_1 \to T_2)?^{l} & = & \ddot{c}_1 \to^{(T_1 \to T_2) l}_{\downarrow} \ddot{c}_2 \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) \bullet (T_3 \to T_4)?^{l} & = & \ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4) l}{\underset{\downarrow}{\to}} \ddot{c}_2 \\
\\
(T_1 \to T_2)?^{l} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_2) & = & (\ddot{c}_1 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l}) \to (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_2) \\
(T_1 \to T_2)?^{l} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \to^{(T_5 \to T_6) l_2}_{\downarrow} \ddot{c}_2) & = & (\ddot{c}_1 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l}) \to^{(T_5 \to T_6) l_2}_{\downarrow} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_2) \\
(T_1 \to T_2)\updownarrow^{l} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_2) & = & (\ddot{c}_1 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l}) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_2) \\
(T_1 \to T_2)\updownarrow^{l_1} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \to^{(T_5 \to T_6) l_2}_{\downarrow} \ddot{c}_2) & = & (\ddot{c}_1 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1}) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \to^{(T_5 \to T_6) l_2}_{\downarrow} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1} \bullet \ddot{c}_2) \\
\ddot{c}_1 \to^{(T_1 \to T_2) l}_{\downarrow} \ddot{c}_2 \bullet (T_3 \to T_4)! & = & (\langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l}) \\
\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \to^{(T_3 \to T_4) l}_{\downarrow} \ddot{c}_2 \bullet (T_5 \to T_6)! & = & (\langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l}) \\
\ddot{c}_1 \to^{(T_1 \to T_2) l_1}_{\downarrow} \ddot{c}_2 \bullet (T_3 \to T_4)\updownarrow l_2 & = & (\langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \to^{(T_3 \to T_4) l_2}_{\downarrow} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1}) \\
\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \to^{(T_3 \to T_4) l}_{\downarrow} \ddot{c}_2 \bullet (T_5 \to T_6)\updownarrow^{l_2} & = & (\langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \to^{(T_5 \to T_6) l_2}_{\downarrow} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1})
\end{array}
$$

Figure 10: Lazy D Supercoercions Version 1, Part 1

$$
\begin{array}{rcl}
G_1?^l \bullet G_2! = G_1\updownarrow^l \bullet G_2! & = & \mathtt{Fail}^l \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
B?^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \ddot{c}_2) = B\updownarrow^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \ddot{c}_2) & = & \mathtt{Fail}^l \\
(\ddot{c}_1 \overset{(T_1 \to T_2)l}{\downarrow \;\; \to} \ddot{c}_2) \bullet B! = (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \overset{(T_3 \to T_4)l}{\downarrow} \ddot{c}_2) \bullet B! & = & \mathtt{Fail}^l \\
\\
G_1?^{l_1} \bullet G_2\updownarrow^{l_2} = G_1\updownarrow^{l_1} \bullet G_2\updownarrow^{l_2} & = & \mathtt{Fail}^{l_1 G_2 l_2} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
B?^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \overset{(T_3 \to T_4)l_2}{\downarrow} \ddot{c}_2) = B\updownarrow^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \overset{(T_3 \to T_4)l_2}{\downarrow} \ddot{c}_2) & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
(\ddot{c}_1 \overset{(T_1 \to T_2)l_1}{\downarrow \;\; \to} \ddot{c}_2) \bullet B\updownarrow^{l_2} = (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \overset{(T_3 \to T_4)l}{\downarrow} \ddot{c}_2) \bullet B\updownarrow^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\\
\mathtt{Fail}^l \bullet G! = \mathtt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^l \bullet \ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \ddot{c}_2 & = & \mathtt{Fail}^l \\
\mathtt{Fail}^{l_1} \bullet G?^{l_2} = \mathtt{Fail}^{l_1} \bullet G\updownarrow^{l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \overset{(T_3 \to T_4)l_2}{\downarrow \;\; \to} \ddot{c}_2 = \mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \overset{(T_3 \to T_4)l_2}{\downarrow} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B\updownarrow^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (T_5 \to T_6)\updownarrow^{l_3} = \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1 \overset{T_3 \to T_4}{\uparrow \;\; \to} \overset{(T_5 \to T_6)l_3}{\downarrow} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_5 \to T_6) l_3} \\
\mathtt{Fail}^{l_1 G_1 l_2} \bullet G_2! = \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1 \overset{T_3 \to T_4}{\uparrow \;\; \to} \ddot{c}_2 & = & \mathtt{Fail}^{l_1} \text{ if } G_1 \leftrightarrow G_2 \\
\\
\mathtt{Fail}^{l_1 G_1 l_2} \bullet G_2! = \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \overset{T_1 \to T_2}{\uparrow \;\; \to} \ddot{c}_2 & = & \mathtt{Fail}^{l_2} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
\mathtt{Fail}^{l_1 G_1 l_2} \bullet G_2\updownarrow^{l_3} & = & \mathtt{Fail}^{l_2 G_2 l_3} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
\mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \overset{T_3 \to T_4}{\uparrow \;\; \to} \overset{(T_5 \to T_6)l_3}{\downarrow} \ddot{c}_2 & = & \mathtt{Fail}^{l_2 (T_5 \to T_6) l_3}
\end{array}
$$

Figure 11: Lazy D Supercomposition Version 1, Part 2

Using the same technique as in Sec 3.2, we can define supercoercion composition (Fig. 10 and 11). The directly calculated composition function for Lazy D is more complex than that for Lazy UD because of the intervening type projections and injections that must be compiled and combined for arrow coercions. For example,

$$(\ddot{c}_1 \overset{(T_1 \to T_2)l}{\downarrow \;\; \to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\uparrow \;\; \to} \ddot{c}_4) = (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4)$$

The equivalent Lazy UD equation lacks the cast compilation parts.

4.0.7 Simplification

We can apply the same general approach as in Sec. 3.3 to simplify the definition of supercoercion composition. We find that simplifying primitive coercions is analogous, but simplifying arrow coercions requires greater care and some auxiliary machinery.

Arrow type primitives are redundant Among the Lazy UD coercions, the arrow primitives $\to!$, $\to?^l$, and $\to\updownarrow^l$ were found to be encodable using the arrow coercions. The same is true for Lazy D, though the encoding requires a little more machinery. Recall that the original inspiration for this transformation was based in how an identity coercion for $\star \to \star$ could be represented as $\iota_\star \to \iota_\star$, and each of the above three casts corresponds to one of the other arrow supercoercion constructors. This construction generalizes to arbitrary arrow types. To see how, observe that:

$$(T_1 \to T_2)?^l \bullet (T_1 \to T_2)! = \langle\!\langle T_1 \Leftarrow T_1 \rangle\!\rangle^l \to \langle\!\langle T_2 \Leftarrow T_2 \rangle\!\rangle^l.$$

Now define $\mathcal{I}(T) = \langle\!\langle T \Leftarrow T \rangle\!\rangle^l$ for arbitrary l, or directly:

$$\begin{array}{c} \mathcal{I}(P) = \iota_P \\ \mathcal{I}(T_1 \to T_2) = \mathcal{I}(T_1) \overset{T_1 \to T_2}{\uparrow \;\; \to} \mathcal{I}(T_2) \end{array}$$

Henglein [1994] proves that a coercion constructed from primitive identity coercions is equivalent to an identity coercion at a higher type. With this, we can redefine primitive arrow coercions as expected:

1. Replace $(T_1 \to T_2)!$ with $\mathcal{I}(T_1) \overset{T_1 \to T_2}{\uparrow \;\; \to} \mathcal{I}(T_2)$;

2. Replace $(T_1 \to T_2)?^l$ with $\mathcal{I}(T_1) \to \overset{(T_1 \to T_2)l}{\downarrow} \mathcal{I}(T_2)$;

3. Replace $(T_1 \to T_2)\updownarrow^l$ with $\mathcal{I}(T_1) \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_1 \to T_2)l}{\downarrow} \mathcal{I}(T_2)$.

As with Lazy UD, this change makes some supercoercions larger, but these forms often arise at runtime as casts interact anyway, so the size benefit of the original form is short-lived. Furthermore, casts retain the same asymptotic space bound at run-time, bounded by the size of the types that appear in the program.

Version 2: Fig. 12.

Simplify base type coercions The exact same simplification for Lazy UD can be performed for Lazy D, first replacing the four base type coercions ι_B, $B!$, $B?^l$, and $B\updownarrow^l$ with B and B^l, and then merging B and B^l into the single B^p coercion.

Version 3: Fig. 13.

Simplify arrow type coercions Under Lazy D, arrow supercoercions cannot be merged according to shared common structure. The reason for this is that every variation of arrow cast differs with respect to its associated arrow types and possibly blame label. However, it is possible to merge the structure of these casts by internalizing the associated arrow types and blame labels as optional parameters. By introducing optional structures, we can again unify the structure of arrow supercoercions.

$$H ::= \epsilon \mid Gl$$
$$S ::= \epsilon \mid T \to T$$

Once again, we introduce an optional G-l pair, which will cover the projection label $(T \to T)l$ for arrow casts, as well as the projections on some failure coercions. Now, though, we must also add the optional arrow type S, as the label for arrow casts with injections to $\star$. With these changes arrow casts take the following unified form:

$$\ddot{c}\ {}^{S}\!\to^{H}\ \ddot{c}$$

Version 4: Fig. 14.

Once again, we merge the two `Fail` supercoercions as $\texttt{Fail}^{lH}$, and introduce the Lazy D analogues of $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ to simplify the definition of composition.

$$\begin{array}{rr|rr} [Q^\epsilon] = \epsilon & & \lceil B^p \rceil = B \\ \left[B^l\right] = Bl & & \lceil \ddot{c}_1\ {}^{S}\!\to^{H}\ \ddot{c}_2 \rceil = \to \\ \left[\ddot{c}_1\ {}^{S}\!\to^{Gl}\ \ddot{c}_2\right] = Gl & & \end{array}$$

We use $\lceil T \rceil$ as the analogous operation for types.

We again distinguish the two kinds of supercoercions treated by these operators, which again comprise types with optional labels:

$$Q ::= B^p \mid \ddot{c}\ {}^{S}\!\to^{H}\ \ddot{c}$$

Unlike the Lazy UD case, we can't quite refer to these labeled types uniformly as Q^p, since the arrow coercion annotation H has both a type G and a label l. Nonetheless, we can unambiguously say Q^ϵ to mean either B^ϵ or $\ddot{c}\ {}^{S}\!\to^{\epsilon}\ \ddot{c}$.

This concludes the simplification process, yielding the Lazy D labeled types in Fig. 15.

The Lazy D supercoercions are more complex than their Lazy UD counterparts. This is a side-effect of Lazy D's larger space of coercions. In fact, if we restrict G to only B and $\star \to \star$, it is easy to check that the equations relate directly to those for Lazy UD. For example, the second and third equations, which characterize how arrow supercoercions interact with one another, collapse into one equation, since the intermediate compiled casts (e.g. $\langle\!\langle T_3 \Leftarrow T_2 \rangle\!\rangle^{l_1}$) become $\iota_\star$.

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P & = & \ddot{c} \\
\\
\star\ B! \bullet B?^l & = & B\updownarrow^l \\
B?^l \bullet B! & = & \iota_B \\
B?^{l_1} \bullet B\updownarrow^{l_2} & = & B?^{l_2} \\
B\updownarrow^{l_1} \bullet B! & = & B! \\
B\updownarrow^{l_1} \bullet B\updownarrow^{l_2} & = & B\updownarrow^{l_2} \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\downarrow]{(T_1 \to T_2)l} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow[\downarrow]{(T_1 \to T_2)l} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\downarrow]{(T_3 \to T_4)l} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\downarrow]{(T_1 \to T_2)l} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_3 \to T_4} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_5 \to T_6} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\downarrow]{(T_1 \to T_2)l} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_3 \to T_4} \xrightarrow[\downarrow]{(T_5 \to T_6)l} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \xrightarrow[\downarrow]{(T_5 \to T_6)l} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l_1} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_5 \to T_6} \xrightarrow[\downarrow]{(T_7 \to T_8)l_2} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_7 \to T_8)l_2} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
\\
\star\ B_1?^l \bullet B_2! = B_1\updownarrow^l \bullet B_2! & = & \mathtt{Fail}^l \text{ if } B_1 \neq B_2 \\
\star\ B_1?^{l_1} \bullet B_2\updownarrow^{l_2} = B_1\updownarrow^{l_1} \bullet B_2\updownarrow^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
B?^l \bullet (\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_2) = B\updownarrow^l \bullet (\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_2) & = & \mathtt{Fail}^l \\
B?^{l_1} \bullet (\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l_2} \ddot{c}_2) = B\updownarrow^{l_1} \bullet (\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l_2} \ddot{c}_2) & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
(\ddot{c}_1 \xrightarrow[\downarrow]{(T_1 \to T_2)l} \ddot{c}_2) \bullet B! = (\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l} \ddot{c}_2) \bullet B! & = & \mathtt{Fail}^l \\
(\ddot{c}_1 \xrightarrow[\downarrow]{(T_1 \to T_2)l_1} \ddot{c}_2) \bullet B\updownarrow^{l_2} = (\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l} \ddot{c}_2) \bullet B\updownarrow^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\\
\star\ \mathtt{Fail}^l \bullet B! = \mathtt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^l \bullet \ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_2 & = & \mathtt{Fail}^l \\
\star\ \mathtt{Fail}^{l_1} \bullet B?^{l_2} = \mathtt{Fail}^{l_1} \bullet B\updownarrow^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \xrightarrow[\downarrow]{(T_3 \to T_4)l_2} \ddot{c}_2 = \mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)l_2} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet B! = \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1 \xrightarrow[\uparrow]{T_3 \to T_4} \ddot{c}_2 & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B\updownarrow^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1 \xrightarrow[\uparrow]{T_3 \to T_4} \xrightarrow[\downarrow]{(T_5 \to T_6)l_3} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_5 \to T_6) l_3} \\
\\
\star\ \mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2! = \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_2 & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\star\ \mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2\updownarrow^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \xrightarrow[\uparrow]{T_3 \to T_4} \xrightarrow[\downarrow]{(T_5 \to T_6)l_3} \ddot{c}_2 & = & \mathtt{Fail}^{l_2 (T_5 \to T_6) l_3}
\end{array}
$$

Figure 12: Lazy D Supercomposition, Version 2

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\star\ \iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
\star\ B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \to (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)^l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{(T_5 \to T_6)^l}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^{l_1}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \overset{(T_7 \to T_8)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_7 \to T_8)^{l_2}}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
\\
\star\ B_1^l \bullet B_2^\epsilon & = & \mathtt{Fail}^l \text{ if } B_1 \neq B_2 \\
\star\ B_1^{l_1} \bullet B_2^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star\ B^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) & = & \mathtt{Fail}^l \\
\star\ B^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\star\ (\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^\epsilon = (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^\epsilon & = & \mathtt{Fail}^l \\
\star\ (\ddot{c}_1 \overset{(T_1 \to T_2)^{l_1}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^{l_2} = (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\\
\star\ \mathtt{Fail}^l \bullet B^\epsilon = \mathtt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^l \bullet \ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2 & = & \mathtt{Fail}^l \\
\star\ \mathtt{Fail}^{l_1} \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star\ \mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2 = \mathtt{Fail}^{l_1} \bullet \ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet B^\epsilon = \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_2 & = & \mathtt{Fail}^{l_1} \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)^{l_3}}{\underset{\downarrow}{\to}} \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_5 \to T_6) l_3} \\
\\
\star\ \mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^\epsilon = \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2 & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\star\ \mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)^{l_3}}{\underset{\downarrow}{\to}} \ddot{c}_2 & = & \mathtt{Fail}^{l_2 (T_5 \to T_6) l_3}
\end{array}
$$

Figure 13: Lazy D Supercomposition, Version 3

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
\star\ (\ddot{c}_1\ {}^{S}\!\to^{\epsilon}\ \ddot{c}_2) \bullet (\ddot{c}_3\ {}^{\epsilon}\!\to^{H}\ \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1)\ {}^{S}\!\to^{H}\ (\ddot{c}_2 \bullet \ddot{c}_4) \\
\star\ (\ddot{c}_1\ {}^{S}\!\to_{(T_1 \to T_2)^{l_1}}\ \ddot{c}_2) \bullet (\ddot{c}_3\ {}^{T_3 \to T_4}\!\to^{H}\ \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1)\ {}^{S}\!\to^{H}\ (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
\\
B_1^l \bullet B_2^\epsilon & = & \mathtt{Fail}^l \text{ if } B_1 \neq B_2 \\
B_1^{l_1} \bullet B_2^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star\ B^l \bullet (\ddot{c}_1\ {}^{T_1 \to T_2}\!\to^{\epsilon}\ \ddot{c}_2) & = & \mathtt{Fail}^l \\
\star\ B^{l_1} \bullet (\ddot{c}_1\ {}^{T_1 \to T_2}\!\to_{(T_3 \to T_4)^{l_2}}\ \ddot{c}_2) & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\star\ (\ddot{c}_1\ {}^{S}\!\to_{(T_1 \to T_2)^{l}}\ \ddot{c}_2) \bullet B^\epsilon & = & \mathtt{Fail}^l \\
\star\ (\ddot{c}_1\ {}^{S}\!\to_{(T_1 \to T_2)^{l_1}}\ \ddot{c}_2) \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\\
\star\ \mathtt{Fail}^l \bullet B^\epsilon = \mathtt{Fail}^l \bullet \ddot{c}_1\ {}^{S}\!\to^{\epsilon}\ \ddot{c}_2 & = & \mathtt{Fail}^l \\
\mathtt{Fail}^{l_1} \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star\ \mathtt{Fail}^{l_1} \bullet \ddot{c}_1\ {}^{S}\!\to_{(T_3 \to T_4)^{l_2}}\ \ddot{c}_2 = & = & \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet B^\epsilon = \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1\ {}^{T_3 \to T_4}\!\to^{\epsilon}\ \ddot{c}_2 & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet \ddot{c}_1\ {}^{T_3 \to T_4}\!\to_{(T_5 \to T_6)^{l_3}}\ \ddot{c}_2 & = & \mathtt{Fail}^{l_1 (T_5 \to T_6) l_3} \\
\\
\star\ \mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^\epsilon = \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1\ {}^{T_1 \to T_2}\!\to^{\epsilon}\ \ddot{c}_2 & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1\ {}^{T_3 \to T_4}\!\to_{(T_5 \to T_6)^{l_3}}\ \ddot{c}_2 & = & \mathtt{Fail}^{l_2 (T_5 \to T_6) l_3}
\end{array}
$$

Figure 14: Lazy D Supercomposition, Version 4

The new cases of compilation arise from our first simplification.

$$\langle\!\langle \star \Leftarrow T_1 \to T_2 \rangle\!\rangle^l = \mathcal{I}(T_1)\ {}^{T_1 \to T_2}\!\to^{\epsilon}\ \mathcal{I}(T_2)$$
$$\langle\!\langle T_1 \to T_2 \Leftarrow \star \rangle\!\rangle^l = \mathcal{I}(T_1)\ {}^{\epsilon}\!\to_{(T_1 \to T_2)^l}\ \mathcal{I}(T_2).$$

$l \in \text{LABELS}, \quad B \in \text{BASICTYPE}, \quad T \in \text{TYPES}, \quad \ddot{c} \in \text{SUPERC}$

$\text{SUPERC}^{T_2 \Leftarrow T_1} = \{\, \ddot{c} \mid \vdash \ddot{c} : T_2 \Leftarrow T_1 \,\}$

$$
\begin{array}{rcl}
G & ::= & B \mid T \to T \\
p & ::= & \epsilon \mid l \\
H & ::= & \epsilon \mid Gl \\
S & ::= & \epsilon \mid T \to T \\
Q & ::= & B^p \mid \ddot{c}\ {}^{S}\!\to^{H} \ddot{c} \\
\ddot{c} & ::= & \iota_\star \mid \texttt{Fail}^{lH} \mid Q
\end{array}
$$

$$
\frac{}{\vdash \iota_\star : \star \Leftarrow \star} \qquad \frac{}{\vdash \texttt{Fail}^{l\epsilon} : T_2 \Leftarrow T_1} \qquad \frac{}{\vdash \texttt{Fail}^{l_1 G l_2} : T \Leftarrow \star}
$$

$$
\frac{}{\vdash B^\epsilon : B \Leftarrow B} \qquad \frac{}{\vdash B^\epsilon : \star \Leftarrow B} \qquad \frac{}{\vdash B^l : B \Leftarrow \star} \qquad \frac{}{\vdash B^l : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1\ {}^{\epsilon}\!\to^{\epsilon} \ddot{c}_2 : T_1 \to T_4 \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1\ {}^{(T_1 \to T_4)}\!\to_{(T_2 \to T_3)l}\ \ddot{c}_2 : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1\ {}^{(T_1 \to T_4)}\!\to^{\epsilon} \ddot{c}_2 : \star \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1 : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2 : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1\ {}^{\epsilon}\!\to_{(T_2 \to T_3)l}\ \ddot{c}_2 : T_1 \to T_4 \Leftarrow \star}
$$

$\boxed{\bullet : \text{SUPERC}^{T_3 \Leftarrow T_2} \times \text{SUPERC}^{T_2 \Leftarrow T_1} \to \text{SUPERC}^{T_3 \Leftarrow T_1}}$ Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
(\ddot{c}_1\ {}^{S}\!\to^{\epsilon} \ddot{c}_2) \bullet (\ddot{c}_3\ {}^{\epsilon}\!\to^{H} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1)\ {}^{S}\!\to^{H}\ (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1\ {}^{S}\!\to_{(T_1 \to T_2)l_1}\ \ddot{c}_2) \bullet (\ddot{c}_3\ {}^{T_3 \to T_4}\!\to^{H}\ \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1)\ {}^{S}\!\to^{H}\ (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
\star \quad Q_1 \bullet Q_2 & = & \texttt{Fail}^{l_1 [Q_2]} \text{ if } Q_1 \to\!\leftarrow Q_2,\ [Q_1] = G l_1 \\
\star \quad \ddot{c} \bullet \texttt{Fail}^{l_1 H} & = & \texttt{Fail}^{l_1 H} \\
\star \quad \texttt{Fail}^{l_1 \epsilon} \bullet Q & = & \texttt{Fail}^{l_1 [Q]} \\
\star \quad \texttt{Fail}^{l_1 G l_2} \bullet Q & = & \texttt{Fail}^{l_1 [Q]} \text{ if } \lceil G \rceil = \lceil Q \rceil \\
\star \quad \texttt{Fail}^{l_1 G l_2} \bullet Q & = & \texttt{Fail}^{l_2 [Q]} \text{ if } \lceil G \rceil \neq \lceil Q \rceil
\end{array}
$$

where

$$
\begin{array}{rcl}
\lceil B^\epsilon \rceil = \epsilon & & \\
\lceil B^l \rceil = Bl & & \\
\lceil c_1\ {}^{S}\!\to^{\epsilon} c_2 \rceil = \epsilon & \lceil B^p \rceil = B & \lceil B \rceil = B \\
\lceil c_1\ {}^{S}\!\to^{Gl} c_2 \rceil = Gl & \lceil c_1\ {}^{S}\!\to^{H} c_2 \rceil = \to & \lceil T_1 \to T_2 \rceil = \to
\end{array}
$$

Figure 15: Lazy D Labeled Types

5 Eager Threesomes

As discussed in Sec. 2.1, eager coercion calculi differ from lazy variants in that they detect and report some cast failures earlier. In particular, an arrow coercion $c_1 \to c_2$ cannot be in normal form and have any of its leaves be a $\texttt{Fail}^l$ cast. Such failures percolate upwards. For instance, under Lazy UD, $\texttt{Fail}^l \to \iota_\star$ was normal. That is no longer the case.

To support eager checking, the grammar for normal coercions must change.

$$\begin{array}{lrcl}
\text{wrappers} & \overline{c} & ::= & G! \mid c^v \to c^v \mid G! \circ (c^v \to c^v) \\
\text{value prefix} & c^p & ::= & \texttt{Fail}^l \mid \overline{c} \mid \texttt{Fail}^l \circ (c^v \to c^v) \\
\text{normal values} & c^v & ::= & \iota_P \mid \overline{c} \mid \texttt{Fail}^l \circ (c^v \to c^v) \mid G?^l \mid c^p \circ G?^l \\
\text{normal coercions} & c^w & ::= & \texttt{Fail}^l \mid c^v
\end{array}$$

To characterize the normal eager coercions, we differentiate between normal coercions at the toplevel and *normal values* c^v, which can safely appear under an arrow. In particular, $\texttt{Fail}^l$ by itself cannot appear under an arrow.

During reduction, definitive failures percolate upwards based on two new rules:

$$\texttt{Fail}^l \to c \longrightarrow^\circ \texttt{Fail}^l$$
$$c^v \to \texttt{Fail}^l \longrightarrow^\circ \texttt{Fail}^l$$

Immediate cast inconsistencies below the head type constructor are reported as failures at the top-level, with domain failures taking precedence over codomain failures.

To ensure that reduction is consistent, and that the inner failures of complex expressions take precedence, arrow coercions can only be combined if the subcoercions are known to not be failures.

$$(c_1^v \to c_2^v) \circ (c_3^v \to c_4^v) \longrightarrow^\circ (c_3^v \circ c_1^v) \to (c_2^v \circ c_4^v)$$

Were this not the case, then some coercion expressions could resolve to inconsistent answers, for example:

$$\begin{aligned}(\texttt{Fail}^{l_1} \to c_1) \circ (\texttt{Fail}^{l_2} \to c_2) &\longrightarrow^\circ (\texttt{Fail}^{l_1} \to c_1) \circ \texttt{Fail}^{l_2} \\ &\longrightarrow^\circ \texttt{Fail}^{l_2}\end{aligned}$$

and

$$\begin{aligned}(\texttt{Fail}^{l_1} \to c_1) \circ (\texttt{Fail}^{l_2} \to c_2) &\longrightarrow^\circ (\texttt{Fail}^{l_2} \circ \texttt{Fail}^{l_1} \to c_1 \circ c_2) \\ &\longrightarrow^{\circ *} \texttt{Fail}^{l_1}.\end{aligned}$$

Furthermore, we must restrict rule (6) from the Lazy UD coercion calculus to ensure that a failing coercion never consumes an arrow coercion to its right:

$$\texttt{Fail}^l \circ G! \longrightarrow^\circ \texttt{Fail}^l \tag{6}$$

Without this restriction, reducing a coercion like $\texttt{Fail}^{l_1} \circ (\star \to \texttt{Int}?^{l_2}) \circ (\star \to \texttt{Bool}!)$ from left-to-right could reduce to $\texttt{Fail}^{l_1}$ instead of the proper $\texttt{Fail}^{l_2}$. This restriction preserves associativity of $\circ$ and enables space efficient casts on the call stack.

Finally, to support eager blame tracking the $\lambda^{\langle c \rangle}$ calculus must be extended with one additional rule:

$$\langle \texttt{Fail}^l \circ (c_1^v \to c_2^v) \rangle s \longrightarrow \textbf{blame}\ l \tag{9}$$

Since the coercion calculus can no longer reduce $\texttt{Fail}^l \circ (c_1^v \to c_2^v)$, it is up to $\lambda^{\langle c \rangle}$ to detect it. In this context, the $\texttt{Fail}^l \circ (c_1^v \to c_2^v)$ is not at risk of being composed with another $(c_2^v \to c_3^v)$ cast that could cause a different failure to take precedence over $\texttt{Fail}^l$.

As with the lazy variants, these calculi are confluent and strongly normalizing, so one can define a normal composition operator $c^v \odot c^v$. Because of the new reduction rules, complex arrow coercions interact with composition

$$
\begin{array}{rcl}
\ddot{c} & ::= & \mathtt{Fail}^{l} \mid \ddot{c}^{v} \\
\ddot{c}^{v} & ::= & \mathtt{Fail}^{lGl} \mid G! \mid G?^{l} \mid G\updownarrow^{l} \mid \iota_{P} \mid \ddot{c}^{v} \to \ddot{c}^{v} \\
& \mid & \ddot{c}^{v} \to \overset{(T\to T)l}{\downarrow} \ddot{c}^{v} \mid \ddot{c}^{v} \overset{T\to T}{\uparrow} \to \ddot{c}^{v} \mid \ddot{c}^{v} \overset{T\to T}{\uparrow} \to \overset{(T\to T)l}{\downarrow} \ddot{c}^{v} \\
& \mid & \ddot{c}^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}^{v} \mid \ddot{c}^{v} \overset{\mathtt{F}^{l}}{\to} \overset{(T\to T)l}{\downarrow} \ddot{c}^{v}
\end{array}
$$

$$
\frac{}{\vdash \mathtt{Fail}^{l} : T_2 \Leftarrow T_1} \qquad \frac{}{\vdash \mathtt{Fail}^{l_1 G l_2} : T \Leftarrow \star} \qquad \frac{}{\vdash G! : \star \Leftarrow G} \qquad \frac{}{\vdash G?^{l} : G \Leftarrow \star} \qquad \frac{}{\vdash G\updownarrow^{l} : \star \Leftarrow \star}
$$

$$
\frac{}{\vdash \iota_P : P \Leftarrow P} \qquad \frac{\vdash \ddot{c}_1^{v} : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^{v} : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^{v} \to \ddot{c}_2^{v} : T_1 \to T_4 \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1^{v} : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^{v} : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^{v} \overset{T_1\to T_4}{\uparrow} \to \overset{(T_2\to T_3)l}{\downarrow} \ddot{c}_2^{v} : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1^{v} : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^{v} : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^{v} \overset{T_1\to T_4}{\uparrow} \to \ddot{c}_2^{v} : \star \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1^{v} : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^{v} : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^{v} \to \overset{(T_2\to T_3)l}{\downarrow} \ddot{c}_2^{v} : T_1 \to T_4 \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1^{v} : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^{v} : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2^{v} : T \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1^{v} : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^{v} : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^{v} \overset{\mathtt{F}^{l_1}(T_2\to T_3)l_2}{\to \quad \downarrow} \ddot{c}_2^{v} : T \Leftarrow \star}
$$

Figure 16: Eager D Supercoercions

differently. To capture this difference, we first define an operation $\widehat{\to}$ that combines normal coercions according to whether either is a failure.

$$
\begin{aligned}
\widehat{c^{v} \to c^{v}} &= c^{v} \to c^{v} \\
\widehat{\mathtt{Fail}^{l} \to c^{w}} &= \mathtt{Fail}^{l} \\
\widehat{c^{v} \to \mathtt{Fail}^{l}} &= \mathtt{Fail}^{l}
\end{aligned}
$$

Armed with this, we can succinctly characterize how eager arrow coercions compose.

Proposition 8.
$(c_1 \to c_2) \odot (c_3 \to c_4) = (c_3 \odot c_1) \widehat{\to} (c_2 \odot c_4)$.

Proof. Straightforward. □

5.1 Eager D Threesomes

Applying our calculational approach to the eager calculi largely follows the same pattern. As with the the normal eager coercions, non-failure supercoercions $\ddot{c}^{v}$ must be differentiated from the full set of supercoercions $\ddot{c}$. Thus, the simple arrow supercoercions have the form $\ddot{c}^{v} \to \ddot{c}^{v}$.

Fig. 16 presents the Eager D supercoercions. Eager D restricts all arrow supercoercions to consist of value supercoercion parts $\ddot{c}^{v}$. To represent the notion of a failure coercion followed by an arrow coercion, Eager D extends the Lazy D supercoercions with two failure arrows supercoercions, whose interpretations are as follows:

$$
\begin{aligned}
\mathcal{N}[\![\ddot{c}_1^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2^{v}]\!] &= \mathtt{Fail}^{l} \circ \mathcal{N}[\![\ddot{c}_1^{v} \to \ddot{c}_2^{v}]\!] \\
\mathcal{N}[\![\ddot{c}_1^{v} \overset{\mathtt{F}^{l_1}(T_1\to T_2)l_2}{\to \quad \downarrow} \ddot{c}_2^{v}]\!] &= \mathtt{Fail}^{l_1} \circ \mathcal{N}[\![\ddot{c}_1^{v} \to \ddot{c}_2^{v}]\!] \circ (T_1 \to T_2)?^{l_2}
\end{aligned}
$$

Only two kinds of failure arrows are needed, since a trailing injection to $\star$ would be annihilated by a $\mathtt{Fail}^{l}$ coercion, e.g.:

$$\mathtt{Fail}^{l} \circ (T_1 \to T_2)! \circ (c_1 \to c_2) \longrightarrow^{o} \circ (c_1 \to c_2)$$

Version 1: Fig. 17 and 18.

Composition for Eager D supercoercions introduces some new equations to characterize failure arrow supercoercions. For instance, they annihilate to the left:

$$\ddot{c} \bullet (\ddot{c}_1^v \overset{\mathtt{F}^l}{\to} \ddot{c}_2^v) = (\ddot{c}_1^v \overset{\mathtt{F}^l}{\to} \ddot{c}_2^v)$$

Furthermore, equations that yield arrow supercoercions are defined using failure-detecting auxiliary functions like $\widehat{\ddot{c}^v \to \ddot{c}^v}$. For example:

$$(\ddot{c}_1^v \to \overset{(T_1 \to T_2)^l}{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_4^v) = (\ddot{c}_3^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1^v) \widehat{\to} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4^v)$$

The complex failure-detecting arrow functions are specified in terms of the simple arrow case and $\bullet$ itself, e.g.:

$$\ddot{c}_1^v \to \widehat{\overset{(T_1 \to T_2)^l}{\downarrow}} \ddot{c}_2^v = \ddot{c}_1^v \widehat{\to} \ddot{c}_2^v \bullet (T_1 \to T_2)?^l$$

The full definitions follow:

$$\begin{aligned}
\ddot{c}_1^v \widehat{\to} \ddot{c}_2^v &= \ddot{c}_1^v \to \ddot{c}_2^v & \ddot{c}_1^v \widehat{\overset{T_1 \to T_2}{\uparrow} \to} \ddot{c}_2^v &= \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^v \\
\mathtt{Fail}^l \widehat{\to} \ddot{c}_2 &= \mathtt{Fail}^l & \mathtt{Fail}^l \widehat{\overset{T_1 \to T_2}{\uparrow} \to} \ddot{c}_2 &= \mathtt{Fail}^l \\
\ddot{c}_1^v \widehat{\to} \mathtt{Fail}^l &= \mathtt{Fail}^l & \ddot{c}_1^v \widehat{\overset{T_1 \to T_2}{\uparrow} \to} \mathtt{Fail}^l &= \mathtt{Fail}^l
\end{aligned}$$

$$\begin{aligned}
\ddot{c}_1^v \to \widehat{\overset{(T_1 \to T_2)l_2}{\downarrow}} \ddot{c}_2^v &= \ddot{c}_1^v \to \overset{(T_1 \to T_2)l}{\downarrow} \ddot{c}_2^v & \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \widehat{\to \overset{(T_3 \to T_4)l}{\downarrow}} \ddot{c}_2^v &= \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4)l}{\downarrow} \ddot{c}_2^v \\
\mathtt{Fail}^{l_1} \to \widehat{\overset{(T_1 \to T_2)l_2}{\downarrow}} \ddot{c}_2 &= \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} & \mathtt{Fail}^{l_1} \overset{T_1 \to T_2}{\uparrow} \widehat{\to \overset{(T_3 \to T_4)l_2}{\downarrow}} \ddot{c}_2 &= \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2} \\
\ddot{c}_1^v \to \widehat{\overset{(T_1 \to T_2)l_2}{\downarrow}} \mathtt{Fail}^{l_1} &= \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} & \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to \widehat{\overset{(T_3 \to T_4)l_2}{\downarrow}} \mathtt{Fail}^{l_1} &= \mathtt{Fail}^{l_1 (T_3 \to T_4) l_2}
\end{aligned}$$

$$\begin{aligned}
\ddot{c}_1^v \widehat{\overset{\mathtt{F}^l}{\to}} \ddot{c}_2^v &= \ddot{c}_1^v \overset{\mathtt{F}^l}{\to} \ddot{c}_2^v & \ddot{c}_1^v \overset{\mathtt{F}^{l_1}}{\to} \widehat{\overset{(T_1 \to T_2)l_2}{\downarrow}} \ddot{c}_2^v &= \ddot{c}_1^v \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)l_2}{\downarrow} \ddot{c}_2^v \\
\mathtt{Fail}^{l_1} \widehat{\overset{\mathtt{F}^{l_2}}{\to}} \ddot{c}_2 &= \mathtt{Fail}^{l_1} & \mathtt{Fail}^{l_1} \overset{\mathtt{F}^{l_3}}{\to} \widehat{\overset{(T_1 \to T_2)l_2}{\downarrow}} \ddot{c}_2 &= \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \\
\ddot{c}_1^v \overset{\mathtt{F}^{l_2}}{\to} \widehat{\mathtt{Fail}^{l_1}} &= \mathtt{Fail}^{l_1} & \ddot{c}_1^v \overset{\mathtt{F}^{l_3}}{\to} \widehat{\overset{(T_1 \to T_2)l_2}{\downarrow}} \mathtt{Fail}^{l_1} &= \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2}
\end{aligned}$$

In Eager D, equations that involve failure and arrow coercions may produce a failure arrow, whereas the Lazy D equivalent simply produced failure. For instance,

$$\mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (T_3 \to T_4)! = \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \widehat{\overset{\mathtt{F}^{l_1}}{\to}} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1}$$

Under Lazy D, the result is simply $\mathtt{Fail}^{l_1}$.

5.2 Simplification

As with the other calculi, primitive arrow coercions are redundant and may be replaced with complex arrow coercions.

Version 2: Fig. 19 and 20.

One difference from Lazy D, however, is that $\mathtt{Fail}^{l_1 (T_1 \to T_2) l_2}$ coercions are also redundant. These primitive coercions can be replaced with

$$\mathcal{I}(T_1) \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)l_2}{\downarrow} \mathcal{I}(T_2).$$

Thus, the only remaining primitive failure coercions are $\mathtt{Fail}^l$ and $\mathtt{Fail}^{lBl}$.

Version 3: Fig. 21 and 22.

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P & = & \ddot{c} \\
\\
G! \bullet G?^l & = & G\updownarrow^l \\
\\
B?^l \bullet B! & = & \iota_B \\
(T_1 \to T_2)?^l \bullet (T_3 \to T_4)! & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{\widehat{l}} \to \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \\
B?^{l_1} \bullet B\updownarrow^{l_2} & = & B?^{l_2} \\
\\
(T_1 \to T_2)?^{l_1} \bullet (T_3 \to T_4)\updownarrow^{l_2} & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1} \to^{\widehat{(T_3 \to T_4)l_2}}_{\downarrow} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1} \\
B\updownarrow^{l_1} \bullet B! & = & B! \\
\\
(T_1 \to T_2)\updownarrow^l \bullet (T_3 \to T_4)! & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \overset{\widehat{T_1 \to T_2}}{\uparrow} \to \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \\
B\updownarrow^{l_1} \bullet B\updownarrow^{l_2} & = & B\updownarrow^{l_2} \\
\\
(T_1 \to T_2)\updownarrow^{l_1} \bullet (T_3 \to T_4)\updownarrow^{l_2} & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \overset{T_1 \to T_2}{\uparrow} \to \overset{\widehat{(T_3 \to T_4)l_2}}{\downarrow} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \\
\\
(\ddot{c}_1^v \to \ddot{c}_2^v) \bullet (\ddot{c}_3^v \to \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v)\widehat{\ } \to (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \to \ddot{c}_2^v) \bullet (\ddot{c}_3^v \to^{(T_1 \to T_2)^l}_{\downarrow} \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v) \to^{\widehat{(T_1 \to T_2)^l}}_{\downarrow} (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_3^v \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_4^v) \bullet (\ddot{c}_3^v \to \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v) \overset{\widehat{T_1 \to T_2}}{\uparrow} \to (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^v) \bullet (\ddot{c}_3^v \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v) \overset{T_1 \to T_2}{\uparrow} \to \overset{\widehat{(T_3 \to T_4)^l}}{\downarrow} (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{\mathbb{F}^l}{\to} \ddot{c}_2^v) \bullet (c_3 \to c_4) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v)\widehat{\ } \overset{\mathbb{F}^l}{\to} (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \ddot{c}_2^v) \bullet (c_3 \to^{(T_1 \to T_2)l_2}_{\downarrow} c_4) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v) \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \overset{\widehat{(T_1 \to T_2)l_2}}{\downarrow} (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \to^{(T_1 \to T_2)^l}_{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1^v)\widehat{\ } \to (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_5 \to T_6}{\uparrow} \to \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1^v) \overset{\widehat{T_1 \to T_2}}{\uparrow} \to (\ddot{c}_2^v \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \to^{(T_1 \to T_2)^l}_{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_3 \to T_4}{\uparrow} \to^{(T_5 \to T_6)^l}_{\downarrow} \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1^v) \to^{\widehat{(T_5 \to T_6)^l}}_{\downarrow} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to^{(T_3 \to T_4)l_1}_{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_5 \to T_6}{\uparrow} \to^{(T_7 \to T_8)l_2}_{\downarrow} \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1^v) \overset{T_1 \to T_2}{\uparrow} \to \overset{\widehat{(T_7 \to T_8)l_2}}{\downarrow} (\ddot{c}_2^v \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1^v) \overset{\mathbb{F}^{l_1}}{\to} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \overset{T_3 \to T_4}{\uparrow} \to^{(T_5 \to T_6)l_3}_{\downarrow} \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1^v) \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \overset{\widehat{(T_5 \to T_6)l_3}}{\downarrow} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4^v) \\
\\
(T_1 \to T_2)! \bullet (\ddot{c}_1^v \to \ddot{c}_2^v) & = & \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^v \\
(T_1 \to T_2)! \bullet (\ddot{c}_1^v \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_2^v) & = & \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_2^v \\
(T_1 \to T_2)?^l \bullet (\ddot{c}_1^v \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_2^v) & = & (\ddot{c}_1^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l)\widehat{\ } \to (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_2^v) \\
(T_1 \to T_2)?^l \bullet (\ddot{c}_1^v \overset{T_3 \to T_4}{\uparrow} \to^{(T_5 \to T_6)l_2}_{\downarrow} \ddot{c}_2^v) & = & (\ddot{c}_1^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l) \to^{\widehat{(T_5 \to T_6)l_2}}_{\downarrow} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_2^v) \\
(T_1 \to T_2)\updownarrow^l \bullet (\ddot{c}_1^v \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_2^v) & = & (\ddot{c}_1^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l) \overset{\widehat{T_1 \to T_2}}{\uparrow} \to (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_2^v) \\
(T_1 \to T_2)\updownarrow^{l_1} \bullet (\ddot{c}_1^v \overset{T_3 \to T_4}{\uparrow} \to^{(T_5 \to T_6)l_2}_{\downarrow} \ddot{c}_2^v) & = & (\ddot{c}_1^v \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1}) \overset{T_1 \to T_2}{\uparrow} \to \overset{\widehat{(T_5 \to T_6)l_2}}{\downarrow} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1} \bullet \ddot{c}_2^v) \\
\\
(\ddot{c}_1^v \to \ddot{c}_2^v) \bullet (T_1 \to T_2)?^l & = & \ddot{c}_1^v \to^{(T_1 \to T_2)^l}_{\downarrow} \ddot{c}_2^v \\
(\ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^v) \bullet (T_3 \to T_4)?^l & = & \ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_2^v \\
(\ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to} \ddot{c}_2^v) \bullet (T_1 \to T_2)?^{l_2} & = & \ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \ddot{c}_2^v \\
\\
(\ddot{c}_1^v \to^{(T_1 \to T_2)^l}_{\downarrow} \ddot{c}_2^v) \bullet (T_3 \to T_4)! & = & (\langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1^v)\widehat{\ } \to (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l) \\
(\ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_2^v) \bullet (T_5 \to T_6)! & = & (\langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1^v) \overset{\widehat{T_1 \to T_2}}{\uparrow} \to (\ddot{c}_2^v \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l) \\
(\ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \ddot{c}_2^v) \bullet (T_3 \to T_4)! & = & (\langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1^v) \overset{\mathbb{F}^{l_1}}{\to} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l) \\
(\ddot{c}_1^v \to^{(T_1 \to T_2)^{l_1}}_{\downarrow} \ddot{c}_2^v) \bullet (T_3 \to T_4)\updownarrow l_2 & = & (\langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1^v) \to^{\widehat{(T_3 \to T_4)l_2}}_{\downarrow} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_1}) \\
(\ddot{c}_1^v \overset{T_1 \to T_2}{\uparrow} \to^{(T_3 \to T_4)^l}_{\downarrow} \ddot{c}_2^v) \bullet (T_5 \to T_6)\updownarrow^{l_2} & = & (\langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1^v) \overset{T_1 \to T_2}{\uparrow} \to \overset{\widehat{(T_5 \to T_6)l_2}}{\downarrow} (\ddot{c}_2^v \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1}) \\
(\ddot{c}_1^v \overset{\mathbb{F}^{l_1}}{\to}_{\downarrow} \ddot{c}_2^v) \bullet (T_3 \to T_4)\updownarrow^{l_3} & = & (\langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1^v) \overset{\mathbb{F}^{l_1}}{\to} \overset{\widehat{(T_3 \to T_4)l_3}}{\downarrow} (\ddot{c}_2^v \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l)
\end{array}
$$

Figure 17: Eager D Supercomposition Version 1, Part 1

$$
\begin{array}{rcl}
G_1?^{l} \bullet G_2! = G_1{\updownarrow}^{l} \bullet G_2! & = & \mathtt{Fail}^{l} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
G_1?^{l_1} \bullet G_2{\updownarrow}^{l_2} = G_1{\updownarrow}^{l_1} \bullet G_2{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 G_2 l_2} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
\star\ B?^{l} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^{v}) = B{\updownarrow}^{l} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2^{v} \\
\star\ B?^{l_1} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2^{v}) = B{\updownarrow}^{l_1} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l_1} (T_3 \to T_4) l_2}{\to \quad \downarrow} \ddot{c}_2^{v} \\
(\ddot{c}_1^{v} \to \overset{(T_1 \to T_2) l}{\downarrow} \ddot{c}_2^{v}) \bullet B! = (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l}{\downarrow} \ddot{c}_2^{v}) \bullet B! & = & \mathtt{Fail}^{l} \\
(\ddot{c}_1^{v} \to \overset{(T_1 \to T_2) l_1}{\downarrow} \ddot{c}_2^{v}) \bullet B{\updownarrow}^{l_2} = (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l}{\downarrow} \ddot{c}_2^{v}) \bullet B{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
(\ddot{c}_1^{v} \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2) l_2}{\downarrow} \ddot{c}_2^{v}) \bullet B! & = & \mathtt{Fail}^{l_2} \\
(\ddot{c}_1^{v} \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2) l_2}{\downarrow} \ddot{c}_2^{v}) \bullet B{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_2 B l_3} \\
\\
\ddot{c} \bullet \mathtt{Fail}^{l} & = & \mathtt{Fail}^{l} \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\star\ \ddot{c} \bullet (\ddot{c}_1^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2^{v} \\
\star\ \ddot{c} \bullet (\ddot{c}_1^{v} \overset{\mathtt{F}^{l_2} (T_1 \to T_2) l_1}{\to \quad \downarrow} \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l_2} (T_1 \to T_2) l_1}{\to \quad \downarrow} \ddot{c}_2^{v} \\
\\
\mathtt{Fail}^{l} \bullet G! & = & \mathtt{Fail}^{l} \\
\mathtt{Fail}^{l_1} \bullet G?^{l_2} = \mathtt{Fail}^{l_1} \bullet G{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\star\ \mathtt{Fail}^{l} \bullet (\ddot{c}_1^{v} \to \ddot{c}_2^{v}) = \mathtt{Fail}^{l} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2^{v} \\
\star\ \mathtt{Fail}^{l_1} \bullet (\ddot{c}_1^{v} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2^{v}) = \mathtt{Fail}^{l_1} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l_1} (T_3 \to T_4) l_2}{\to \quad \downarrow} \ddot{c}_2^{v} \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B! & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (T_3 \to T_4)! & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \overset{\mathtt{F}^{l_1}}{\to} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (\ddot{c}_1^{v} \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_2^{v}) & = & (\ddot{c}_1^{v} \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2}) \overset{\mathtt{F}^{l_1}}{\to} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_2^{v}) \\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (T_3 \to T_4){\updownarrow}^{l_3} & = & \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \overset{\mathtt{F}^{l_1} (T_3 \to T_4) l_3}{\to \quad \downarrow} \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (\ddot{c}_1^{v} \overset{T_3 \to T_4}{\uparrow} \to \overset{(T_5 \to T_6) l_3}{\downarrow} \ddot{c}_2^{v}) & = & (\ddot{c}_1^{v} \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2}) \overset{\mathtt{F}^{l_1} (T_5 \to T_6) l_3}{\to \quad \downarrow} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_2^{v}) \\
\\
\mathtt{Fail}^{l_1 G_1 l_2} \bullet G_2! & = & \mathtt{Fail}^{l_2} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
\mathtt{Fail}^{l_1 G_1 l_2} \bullet G_2{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_2 G_2 l_3} \text{ if } G_1 \rightarrow\!\!\leftarrow G_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1^{v} \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2^{v} \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1^{v} \overset{T_3 \to T_4}{\uparrow} \to \overset{(T_5 \to T_6) l_3}{\downarrow} \ddot{c}_2^{v}) & = & \ddot{c}_1^{v} \overset{\mathtt{F}^{l_1} (T_5 \to T_6) l_3}{\to \quad \downarrow} \ddot{c}_2^{v}
\end{array}
$$

Figure 18: Eager D Supercomposition Version 1, Part 2

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P & = & \ddot{c} \\
\\
B! \bullet B?^{l} & = & B\updownarrow^{l} \\
B?^{l} \bullet B! & = & \iota_B \\
B?^{l_1} \bullet B\updownarrow^{l_2} & = & B?^{l_2} \\
B\updownarrow^{l_1} \bullet B! & = & B! \\
B\updownarrow^{l_1} \bullet B\updownarrow^{l_2} & = & B\updownarrow^{l_2} \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \mathbin{\widehat{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_1 \to T_2)l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\widehat{(T_1 \to T_2)l}}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\mathbin{\widehat{\to}}}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_3 \to T_4)l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\mathbin{\widehat{\to}}}} \overset{(T_3 \to T_4)l}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathrm{F}^{l}}{\to} \ddot{c}_2) \bullet (c_3 \to c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\mathrm{F}^{l}}{\mathbin{\widehat{\to}}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathrm{F}^{l_1}}{\to} \ddot{c}_2) \bullet (c_3 \overset{(T_1 \to T_2)l_2}{\underset{\downarrow}{\to}} c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\mathrm{F}^{l_1}}{\to} \overset{(T_1 \to T_2)l_2}{\underset{\downarrow}{}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \mathbin{\widehat{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\mathbin{\widehat{\to}}}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l} \bullet \ddot{c}_1) \to \overset{\widehat{(T_5 \to T_6)l}}{\underset{\downarrow}{}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)l_1}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \overset{(T_7 \to T_8)l_2}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{\widehat{(T_7 \to T_8)l_2}}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
\\
(\ddot{c}_1 \overset{\mathrm{F}^{l_1}}{\to} \overset{(T_1 \to T_2)l_2}{\underset{\downarrow}{}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \overset{\widehat{\mathrm{F}^{l_1}}}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathrm{F}^{l_1}}{\to} \overset{(T_1 \to T_2)l_2}{\underset{\downarrow}{}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)l_3}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \overset{\mathrm{F}^{l_1}}{\to} \overset{\widehat{(T_5 \to T_6)l_3}}{\underset{\downarrow}{}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4)
\end{array}
$$

Figure 19: Eager D Supercomposition Version 2, Part 1

$$
\begin{array}{rcl}
B_1?^{l} \bullet B_2! = B_1{\updownarrow}^{l} \bullet B_2! & = & \mathtt{Fail}^{l} \text{ if } B_1 \neq B_2 \\
B_1?^{l_1} \bullet B_2{\updownarrow}^{l_2} = B_1{\updownarrow}^{l_1} \bullet B_2{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star\ B?^{l} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) = B{\updownarrow}^{l} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2 \\
\star\ B?^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) = B{\updownarrow}^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2 \\
(\ddot{c}_1 \to \overset{(T_1 \to T_2) l}{\downarrow} \ddot{c}_2) \bullet B! = (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l}{\downarrow} \ddot{c}_2) \bullet B! & = & \mathtt{Fail}^{l} \\
(\ddot{c}_1 \to \overset{(T_1 \to T_2) l_1}{\downarrow} \ddot{c}_2) \bullet B{\updownarrow}^{l_2} = (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l}{\downarrow} \ddot{c}_2) \bullet B{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2) l_2}{\downarrow} \ddot{c}_2) \bullet B! & = & \mathtt{Fail}^{l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2) l_2}{\downarrow} \ddot{c}_2) \bullet B{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_2 B l_3} \\
\\
\ddot{c} \bullet \mathtt{Fail}^{l} & = & \mathtt{Fail}^{l} \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 G l_2} & = & \mathtt{Fail}^{l_1 G l_2} \\
\star\ \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2 \\
\star\ \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_1 \to T_2) l_1}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_1 \to T_2) l_1}{\downarrow} \ddot{c}_2 \\
\\
\mathtt{Fail}^{l} \bullet B! & = & \mathtt{Fail}^{l} \\
\mathtt{Fail}^{l_1} \bullet B?^{l_2} = \mathtt{Fail}^{l_1} \bullet B{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star\ \mathtt{Fail}^{l} \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^{l} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2 \\
\mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) = \mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2 \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B! & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\uparrow} \to \ddot{c}_2) & = & (\ddot{c}_1 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2}) \overset{\mathtt{F}^{l_1}}{\to} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_2) \\
\star\ \mathtt{Fail}^{l_1 (T_1 \to T_2) l_2} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\uparrow} \to \overset{(T_5 \to T_6) l_3}{\downarrow} \ddot{c}_2) & = & (\ddot{c}_1 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2}) \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_5 \to T_6) l_3}{\downarrow} (\langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_2) \\
\\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2! & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\uparrow} \to \overset{(T_5 \to T_6) l_3}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_5 \to T_6) l_3}{\downarrow} \ddot{c}_2
\end{array}
$$

Figure 20: Eager D Supercomposition Version 2, Part 2

$$\boxed{\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}} \quad \text{Supercoercion Composition}$$

$$\begin{array}{rcl}
\iota_P \bullet \ddot{c} = \ddot{c} \bullet \iota_P & = & \ddot{c} \\
\\
B! \bullet B?^l & = & B\updownarrow^l \\
B?^l \bullet B! & = & \iota_B \\
B?^{l_1} \bullet B\updownarrow^{l_2} & = & B?^{l_2} \\
B\updownarrow^{l_1} \bullet B! & = & B! \\
B\updownarrow^{l_1} \bullet B\updownarrow^{l_2} & = & B\updownarrow^{l_2} \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \widehat{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\downarrow]{(T_1 \to T_2)^l} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow[\downarrow]{\widehat{(T_1 \to T_2)^l}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow[\uparrow]{\widehat{T_1 \to T_2}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\downarrow]{(T_3 \to T_4)^l} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} \widehat{\to} \xrightarrow[\downarrow]{(T_3 \to T_4)^l} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow{\mathbb{F}^l} \ddot{c}_2) \bullet (c_3 \to c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \widehat{\xrightarrow{\mathbb{F}^l}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow{\mathbb{F}^{l_1}} \ddot{c}_2) \bullet (c_3 \xrightarrow[\downarrow]{(T_1 \to T_2)^{l_2}} c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \xrightarrow{\mathbb{F}^{l_1}} \widehat{\xrightarrow[\downarrow]{(T_1 \to T_2)^{l_2}}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\downarrow]{(T_1 \to T_2)^l} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_3 \to T_4} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \widehat{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)^l} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_5 \to T_6} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} \widehat{\to} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\downarrow]{(T_1 \to T_2)^l} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_3 \to T_4} \xrightarrow[\downarrow]{(T_5 \to T_6)^l} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \to \widehat{\xrightarrow[\downarrow]{(T_5 \to T_6)^l}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow[\uparrow]{T_1 \to T_2} \xrightarrow[\downarrow]{(T_3 \to T_4)^{l_1}} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_5 \to T_6} \xrightarrow[\downarrow]{(T_7 \to T_8)^{l_2}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \xrightarrow[\uparrow]{T_1 \to T_2} \widehat{\to} \xrightarrow[\downarrow]{(T_7 \to T_8)^{l_2}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow{\mathbb{F}^{l_1}} \xrightarrow[\downarrow]{(T_1 \to T_2)^{l_2}} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_3 \to T_4} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \xrightarrow{\mathbb{F}^{l_1}} \widehat{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \xrightarrow{\mathbb{F}^{l_1}} \xrightarrow[\downarrow]{(T_1 \to T_2)^{l_2}} \ddot{c}_2) \bullet (\ddot{c}_3 \xrightarrow[\uparrow]{T_3 \to T_4} \xrightarrow[\downarrow]{(T_5 \to T_6)^{l_3}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \xrightarrow{\mathbb{F}^{l_1}} \widehat{\to} \xrightarrow[\downarrow]{(T_5 \to T_6)^{l_3}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4)
\end{array}$$

Figure 21: Eager D Supercomposition Version 3, Part 1

Base type coercions can be merged as before.

Version 4: Fig. 23.

Version 5: Fig. 24.

Merging arrow coercions is complicated by the addition of the failure arrow, which must be accounted for. We merge our three arrow coercion forms as before, but now the prefix arrow label has 3 forms, adding the blame label l to indicate a failure arrow.

$$S ::= \epsilon \mid G \mid l$$
$$H ::= \epsilon \mid Gl$$

After this change, the arrow coercions have the form

$$\ddot{c}^v \ {}^{S}\!\!\to^{H} \ddot{c}^v,$$

Along with the requisite auxiliary arrow function

$$\ddot{c} \ \widehat{{}^{S}\!\!\to^{H}} \ \ddot{c}.$$

Version 6: Fig. 25.

After merging the primitive failure coercions, we arrive at the our final definition of Eager D Labeled types (Fig. 26). This more complex than the Lazy D ones specifically because failure arrows require some independent treatment, despite our structural unification of arrows. However, Eager D uses the same cast compilation function as for Lazy D: the differences show up at runtime.

To use Eager D supercoercions, the $\lambda^{\langle \ddot{c} \rangle}$ calculus requires, just as $\lambda^{\langle c \rangle}$ did, an additional rule for failure arrows:

$$\langle T_1 \ {}^{\ddot{c}_1}\!\!\xleftarrow{l \to^H}{}^{\ddot{c}_2} T_2 \rangle \ s \longrightarrow \mathbf{blame}\ l.$$

$$
\begin{array}{rcl}
B_1?^{l} \bullet B_2! = B_1{\updownarrow}^{l} \bullet B_2! & = & \mathtt{Fail}^{l} \text{ if } B_1 \neq B_2 \\
B_1?^{l_1} \bullet B_2{\updownarrow}^{l_2} = B_1{\updownarrow}^{l_1} \bullet B_2{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star\ B?^{l} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) = B{\updownarrow}^{l} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2 \\
B?^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) = B{\updownarrow}^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1} (T_3 \to T_4) l_2}{\to} \ddot{c}_2 \\
(\ddot{c}_1 \to \overset{(T_1 \to T_2) l}{\downarrow} \ddot{c}_2) \bullet B! = (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l}{\downarrow} \ddot{c}_2) \bullet B! & = & \mathtt{Fail}^{l} \\
\star\ (\ddot{c}_1 \to \overset{(T_1 \to T_2) l_1}{\downarrow} \ddot{c}_2) \bullet B{\updownarrow}^{l_2} = (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l}{\downarrow} \ddot{c}_2) \bullet B{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1} (T_1 \to T_2) l_2}{\to} \downarrow \ddot{c}_2) \bullet B! & = & \mathtt{Fail}^{l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1} (T_1 \to T_2) l_2}{\to} \downarrow \ddot{c}_2) \bullet B{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_2 B l_3} \\
\\
\ddot{c} \bullet \mathtt{Fail}^{l} & = & \mathtt{Fail}^{l} \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 B l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star\ \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2 \\
\star\ \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^{l_2} (T_1 \to T_2) l_1}{\to} \downarrow \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_2} (T_1 \to T_2) l_1}{\to} \downarrow \ddot{c}_2 \\
\\
\mathtt{Fail}^{l} \bullet B! & = & \mathtt{Fail}^{l} \\
\mathtt{Fail}^{l_1} \bullet B?^{l_2} = \mathtt{Fail}^{l_1} \bullet B{\updownarrow}^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star\ \mathtt{Fail}^{l} \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^{l} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l}}{\to} \ddot{c}_2 \\
\mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) = \mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \overset{(T_3 \to T_4) l_2}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1} (T_3 \to T_4) l_2}{\to} \downarrow \ddot{c}_2 \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B! & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2! & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2{\updownarrow}^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\uparrow} \to \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2 \\
\star\ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\uparrow} \to \overset{(T_5 \to T_6) l_3}{\downarrow} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1} (T_5 \to T_6) l_3}{\to} \downarrow \ddot{c}_2
\end{array}
$$

Figure 22: Eager D Supercomposition Version 3, Part 2

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B \bullet B & = & B \\
B \bullet B^l & = & B^l \\
B^l \bullet B & = & B \\
B^l \bullet B^l & = & B^l \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \mathbin{\widehat{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\widehat{(T_1 \to T_2)^l}}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\widehat{\to}}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\widehat{\to}}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2) \bullet (c_3 \to c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\mathtt{F}^l}{\widehat{\to}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2) \bullet (c_3 \overset{(T_1 \to T_2)^{l_2}}{\underset{\downarrow}{\to}} c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\underset{\downarrow}{\widehat{\to}}} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \mathbin{\widehat{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\widehat{\to}}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)^l}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{\widehat{(T_5 \to T_6)^l}}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^{l_1}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\underset{\uparrow}{\to}} \overset{(T_7 \to T_8)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\underset{\uparrow}{\widehat{\to}}} \overset{(T_7 \to T_8)^{l_2}}{\underset{\downarrow}{\to}} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \overset{\mathtt{F}^{l_1}}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)^{l_3}}{\underset{\downarrow}{\to}} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_5 \to T_6)^{l_3}}{\underset{\downarrow}{\widehat{\to}}} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4) \\
\\
B_1^l \bullet B_2 & = & \mathtt{Fail}^l \text{ if } B_1 \neq B_2 \\
B_1^{l_1} \bullet B_2^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star \; B^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2 \\
\star \; B^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2 \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B = (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B & = & \mathtt{Fail}^l \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^{l_1}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^{l_2} = (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^l}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B & = & \mathtt{Fail}^{l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) \bullet B^{l_3} & = & \mathtt{Fail}^{l_2 B l_3} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 B l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star \; \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2 \\
\star \; \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_1 \to T_2)^{l_1}}{\underset{\downarrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_1 \to T_2)^{l_1}}{\underset{\downarrow}{\to}} \ddot{c}_2 \\
\\
\mathtt{Fail}^l \bullet B & = & \mathtt{Fail}^l \\
\mathtt{Fail}^{l_1} \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star \; \mathtt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2 \\
\mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) = \mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_3 \to T_4)^{l_2}}{\underset{\downarrow}{\to}} \ddot{c}_2 \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2 & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\star \; \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\underset{\uparrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2 \\
\star \; \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\underset{\uparrow}{\to}} \overset{(T_5 \to T_6)^{l_3}}{\underset{\downarrow}{\to}} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_5 \to T_6)^{l_3}}{\underset{\downarrow}{\to}} \ddot{c}_2
\end{array}
$$

Figure 23: Eager D Supercomposition Version 4

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
\\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\frown}{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \to \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_1 \to T_2)^l}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{(T_1 \to T_2)^l}{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_3 \overset{T_1 \to T_2}{\to} \ddot{c}_4) \bullet (\ddot{c}_3 \to \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{(T_3 \to T_4)^l}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^l}{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2) \bullet (c_3 \to c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\mathtt{F}^l}{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2) \bullet (c_3 \overset{(T_1 \to T_2)^{l_2}}{\to} c_4) & = & (\ddot{c}_3 \bullet \ddot{c}_1) \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\to} (\ddot{c}_2 \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{\frown}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\to} \overset{(T_5 \to T_6)^l}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{(T_5 \to T_6)^l}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^l}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^l \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^{l_1}}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_5 \to T_6}{\to} \overset{(T_7 \to T_8)^{l_2}}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_5 \Leftarrow T_3 \rangle\!\rangle^{l_1} \bullet \ddot{c}_1) \overset{T_1 \to T_2}{\to} \overset{(T_7 \to T_8)^{l_2}}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_4 \Leftarrow T_6 \rangle\!\rangle^{l_1} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \overset{\mathtt{F}^{l_1}}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4) \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\to} \ddot{c}_2) \bullet (\ddot{c}_3 \overset{T_3 \to T_4}{\to} \overset{(T_5 \to T_6)^{l_3}}{\to} \ddot{c}_4) & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^{l_2} \bullet \ddot{c}_1) \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_5 \to T_6)^{l_3}}{\to} (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^{l_2} \bullet \ddot{c}_4) \\
\\
B_1^l \bullet B_2 & = & \mathtt{Fail}^l \text{ if } B_1 \neq B_2 \\
B_1^{l_1} \bullet B_2^{l_2} & = & \mathtt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star \ B^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2 \\
\star \ B^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^{l_2}}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_3 \to T_4)^{l_2}}{\to} \ddot{c}_2 \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^l}{\to} \ddot{c}_2) \bullet B = (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^l}{\to} \ddot{c}_2) \bullet B & = & \mathtt{Fail}^l \\
(\ddot{c}_1 \overset{(T_1 \to T_2)^{l_1}}{\to} \ddot{c}_2) \bullet B^{l_2} = (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^{l_1}}{\to} \ddot{c}_2) \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\to} \ddot{c}_2) \bullet B & = & \mathtt{Fail}^{l_2} \\
(\ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_1 \to T_2)^{l_2}}{\to} \ddot{c}_2) \bullet B^{l_3} & = & \mathtt{Fail}^{l_2 B l_3} \\
\\
\ddot{c} \bullet \mathtt{Fail}^l & = & \mathtt{Fail}^l \\
\ddot{c} \bullet \mathtt{Fail}^{l_1 B l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star \ \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2 \\
\star \ \ddot{c} \bullet (\ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_1 \to T_2)^{l_1}}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_1 \to T_2)^{l_1}}{\to} \ddot{c}_2 \\
\\
\mathtt{Fail}^l \bullet B & = & \mathtt{Fail}^l \\
\mathtt{Fail}^{l_1} \bullet B^{l_2} & = & \mathtt{Fail}^{l_1 B l_2} \\
\star \ \mathtt{Fail}^l \bullet (\ddot{c}_1 \to \ddot{c}_2) = \mathtt{Fail}^l \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^l}{\to} \ddot{c}_2 \\
\mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \overset{(T_3 \to T_4)^{l_2}}{\to} \ddot{c}_2) = \mathtt{Fail}^{l_1} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \overset{(T_3 \to T_4)^{l_2}}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \overset{(T_3 \to T_4)^{l_2}}{\to} \ddot{c}_2 \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B & = & \mathtt{Fail}^{l_1} \\
\mathtt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \mathtt{Fail}^{l_1 B l_3} \\
\\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2 & = & \mathtt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^{l_3} & = & \mathtt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\star \ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_1 \to T_2}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_1}}{\to} \ddot{c}_2 \\
\star \ \mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1 \overset{T_3 \to T_4}{\to} \overset{(T_5 \to T_6)^{l_3}}{\to} \ddot{c}_2) & = & \ddot{c}_1 \overset{\mathtt{F}^{l_2}}{\to} \overset{(T_5 \to T_6)^{l_3}}{\to} \ddot{c}_2
\end{array}
$$

Figure 24: Eager D Supercomposition Version 5

$\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}$ Supercoercion Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
\\
\ddot{c}_1 \ {}^{S}\!\to^{\epsilon} \ddot{c}_2 \bullet \ddot{c}_3 \ {}^{\epsilon}\!\to^{H} \ddot{c}_4 & = & (\ddot{c}_3 \bullet \ddot{c}_1) \ \widehat{{}^{S}\!\to^{H}} \ (\ddot{c}_2 \bullet \ddot{c}_4) \\
\ddot{c}_1 \ {}^{S}\!\to^{T_1 \to T_2^l} \ddot{c}_2 \bullet \ddot{c}_3 \ {}^{T_3 \to T_4}\!\to^{H} \ddot{c}_4 & = & (\ddot{c}_3 \bullet \langle\!\langle T_3 \Leftarrow T_1 \rangle\!\rangle^l \bullet \ddot{c}_1) \ \widehat{{}^{S}\!\to^{H}} \ (\ddot{c}_2 \bullet \langle\!\langle T_2 \Leftarrow T_4 \rangle\!\rangle^l \bullet \ddot{c}_4) \\
\\
B_1^l \bullet B_2^\epsilon & = & \texttt{Fail}^l \text{ if } B_1 \neq B_2 \\
B_1^{l_1} \bullet B_2^{l_2} & = & \texttt{Fail}^{l_1 B_2 l_2} \text{ if } B_1 \neq B_2 \\
\star \quad B^{l_1} \bullet \ddot{c}_1 \ {}^{T_1 \to T_2}\!\to^{H} \ddot{c}_2 & = & \ddot{c}_1 \ {}^{l_1}\!\to^{H} \ddot{c}_2 \\
\ddot{c}_1 \ {}^{S}\!\to^{Gl} \ddot{c}_2 \bullet B^\epsilon & = & \texttt{Fail}^l \\
\ddot{c}_1 \ {}^{S}\!\to^{Gl_1} \ddot{c}_2 \bullet B^{l_2} & = & \texttt{Fail}^{l_1 B l_2} \\
\\
\ddot{c} \bullet \texttt{Fail}^l & = & \texttt{Fail}^l \\
\ddot{c} \bullet \texttt{Fail}^{l_1 B l_2} & = & \texttt{Fail}^{l_1 B l_2} \\
\star \quad \ddot{c} \bullet \ddot{c}_1 \ {}^{l_1}\!\to^{H} \ddot{c}_2 & = & \ddot{c}_1 \ {}^{l_1}\!\to^{H} \ddot{c}_2 \\
\\
\texttt{Fail}^l \bullet B^\epsilon & = & \texttt{Fail}^l \\
\texttt{Fail}^{l_1} \bullet B^{l_2} & = & \texttt{Fail}^{l_1 B l_2} \\
\star \ \texttt{Fail}^{l_1} \bullet \ddot{c}_1 \ {}^{\epsilon}\!\to^{H} \ddot{c}_2 = \texttt{Fail}^{l_1} \bullet \ddot{c}_1 \ {}^{G_1}\!\to^{H} \ddot{c}_2 & = & \ddot{c}_1 \ {}^{l_1}\!\to^{H} \ddot{c}_2 \\
\\
\texttt{Fail}^{l_1 B l_2} \bullet B^\epsilon & = & \texttt{Fail}^{l_1} \\
\texttt{Fail}^{l_1 B l_2} \bullet B^{l_3} & = & \texttt{Fail}^{l_1 B l_3} \\
\\
\texttt{Fail}^{l_1 B_1 l_2} \bullet B_2^\epsilon & = & \texttt{Fail}^{l_2} \text{ if } B_1 \neq B_2 \\
\texttt{Fail}^{l_1 B_1 l_2} \bullet B_2^{l_3} & = & \texttt{Fail}^{l_2 B_2 l_3} \text{ if } B_1 \neq B_2 \\
\star \quad \texttt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1 \ {}^{(T_1 \to T_2)}\!\to^{H} \ddot{c}_2 & = & \ddot{c}_1 \ {}^{l_2}\!\to^{H} \ddot{c}_2
\end{array}
$$

Figure 25: Eager D Supercomposition Version 6

The Latent variant simply drops the outer types of the threesome:

$$\langle \ddot{c}_1 \ {}^{l}\!\to^{H} \ddot{c}_2 \rangle s \longrightarrow \textbf{blame}\ l.$$

$l \in \textsc{Labels}, \quad B \in \textsc{BasicType}, \quad T \in \textsc{Types}, \quad \ddot{c} \in \textsc{SuperC}$

$\textsc{SuperC}^{T_2 \Leftarrow T_1} = \{\, \ddot{c} \mid \vdash \ddot{c} : T_2 \Leftarrow T_1 \,\}$

$$
\begin{array}{rcl}
G & ::= & B \mid T \to T \\
p & ::= & \epsilon \mid l \\
H & ::= & \epsilon \mid Gl \\
S & ::= & \epsilon \mid G \mid l \\
\ddot{c} & ::= & \iota_\star \mid \texttt{Fail}^{lH} \mid B^p \mid \ddot{c}^v \ {}^S\!\to^H \ddot{c}^v
\end{array}
$$

$$
\frac{}{\vdash \iota_\star : \star \Leftarrow \star} \qquad \frac{}{\vdash \texttt{Fail}^{l\epsilon} : T_2 \Leftarrow T_1} \qquad \frac{}{\vdash \texttt{Fail}^{l_1 B l_2} : T \Leftarrow \star}
$$

$$
\frac{}{\vdash B^\epsilon : B \Leftarrow B} \qquad \frac{}{\vdash B^\epsilon : \star \Leftarrow B} \qquad \frac{}{\vdash B^l : B \Leftarrow \star} \qquad \frac{}{\vdash B^l : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^v \ {}^\epsilon\!\to^\epsilon \ddot{c}_2^v : T_1 \to T_4 \Leftarrow T_2 \to T_3}
\qquad
\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_2^v \ {}^{(T_1 \to T_4)}\!\to^{(T_2 \to T_3) l} \ddot{c}_1^v : \star \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_2^v \ {}^{(T_1 \to T_4)}\!\to^\epsilon \ddot{c}_1^v : \star \Leftarrow T_2 \to T_3}
\qquad
\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_2^v \ {}^\epsilon\!\to^{(T_2 \to T_3)^l} \ddot{c}_1^v : T_1 \to T_4 \Leftarrow \star}
$$

$$
\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^v \ {}^{l_1}\!\to^\epsilon \ddot{c}_2^v : T \Leftarrow T_2 \to T_3}
\qquad
\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^v \ {}^{l_1}\!\to^{(T_2 \to T_3)^{l_2}} \ddot{c}_2^v : T \Leftarrow \star}
$$

$\boxed{\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}}$ Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
\ddot{c}_1^v \ {}^S\!\to^\epsilon \ddot{c}_2^v \bullet \ddot{c}_3^v \ {}^\epsilon\!\to^H \ddot{c}_4^v & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v) \ \widehat{{}^{S\to H}} \ (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
(\ddot{c}_1^v \ {}^S\!\to^{(T_1 \to T_2) l} \ddot{c}_2^v) \bullet (\ddot{c}_3^v \ {}^{T_3 \to T_4}\!\to^H \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \langle T_3 \Leftarrow T_1 \rangle^l \bullet \ddot{c}_1^v) \ \widehat{{}^{S\to H}} \ (\ddot{c}_2^v \bullet \langle T_2 \Leftarrow T_4 \rangle^l \bullet \ddot{c}_4^v) \\
\\
B_1^{l_1} \bullet B_2^{p_2} & = & \texttt{Fail}^{l_1 \left[B_2^{p_2}\right]} \text{ if } B_1 \neq B_2 \\
\star \quad B^{l_1} \bullet \ddot{c}_1^v \ {}^{T_1 \to T_2}\!\to^H \ddot{c}_2^v & = & \ddot{c}_1^v \ {}^{l_1}\!\to^H \ddot{c}_2^v \\
\ddot{c}_1^v \ {}^S\!\to^{Gl_1} \ddot{c}_2^v \bullet B^{p_2} & = & \texttt{Fail}^{l_1 \left[B^{p_2}\right]} \\
\\
\ddot{c} \bullet \texttt{Fail}^{lH} & = & \texttt{Fail}^{lH} \\
\star \quad \ddot{c} \bullet \ddot{c}_1^v \ {}^{l_1}\!\to^H \ddot{c}_2^v & = & \ddot{c}_1^v \ {}^{l_1}\!\to^H \ddot{c}_2^v \\
\\
\texttt{Fail}^{l_1} \bullet B^{p_2} & = & \texttt{Fail}^{l_1 \left[B^{p_2}\right]} \\
\star \quad \texttt{Fail}^{l_1} \bullet \ddot{c}_1^v \ {}^\epsilon\!\to^H \ddot{c}_2^v = \texttt{Fail}^{l_1} \bullet \ddot{c}_1^v \ {}^{G_1}\!\to^H \ddot{c}_2^v & = & \ddot{c}_1^v \ {}^{l_1}\!\to^H \ddot{c}_2^v \\
\\
\texttt{Fail}^{l_1 B l_2} \bullet B^{p_3} & = & \texttt{Fail}^{l_1 \left[B^{p_3}\right]} \\
\\
\texttt{Fail}^{l_1 B_1 l_2} \bullet B_2^{p_3} & = & \texttt{Fail}^{l_2 \left[B_2^{p_3}\right]} \text{ if } B_1 \neq B_2 \\
\star \quad \texttt{Fail}^{l_1 B l_2} \bullet \ddot{c}_1^v \ {}^{G_1}\!\to^H \ddot{c}_2^v & = & \ddot{c}_1^v \ {}^{l_2}\!\to^H \ddot{c}_2^v
\end{array}
$$

where

$$
\begin{array}{rcl}
\ddot{c}_1^v \ \widehat{{}^{S\to H}} \ \ddot{c}_2^v = \ddot{c}_1^v \ {}^S\!\to^H \ddot{c}_2^v \\
\texttt{Fail}^{l_1} \ \widehat{{}^{S\to\epsilon}} \ \ddot{c}_2 = \\
\ddot{c}_1^v \ \widehat{{}^{S\to\epsilon}} \ \texttt{Fail}^{l_1} = \texttt{Fail}^{l_1} \\
\texttt{Fail}^{l_1} \ \widehat{{}^{S\to(T_1\to T_2)l_2}} \ \ddot{c}_2 = \\
\ddot{c}_1^v \ {}^S\!\to^{\widehat{(T_1\to T_2)l_2}} \texttt{Fail}^{l_1} = \mathcal{I}(T_1) \ {}^{l_1}\!\to^{(T_1\to T_2)l_2} \mathcal{I}(T_2)
\end{array}
\qquad
\begin{array}{l}
\left[B^\epsilon\right] = \epsilon \\
\left[B^l\right] = Bl
\end{array}
$$

Figure 26: Eager D Labeled Types

6 Eager UD Supercoercions

Rather than replay the process for Eager UD Supercoercions, we can treat it as a restriction of the Eager D system. Recall that the major difference between D and UD is the presence of primitive arrow coercions aside from $(\star \to \star)! = \to !$, etc. Restricting the definitions for Eager D in this manner is straightforward and leads to some simplifications to the system. For instance, arrow coercions need not track types G any longer, since the only possible type is $\star \to \star$. As a result, arrow coercions take on the much simpler form $\ddot{c}^v \ {}^p\!\!\to^p \ddot{c}^v$. The Lazy D supercoercion $\texttt{Fail}^{l_1 \to l_2}$ is represented with the arrow coercion $\iota_\star \ {}^{l_1}\!\!\to^{l_2} \iota_\star$. The supercoercions for Eager UD are presented in Fig. 27.

7 Discussion

The primary contributions of this paper are three novel threesome semantics for coercion calculi, each of which supports desirable properties, like catching more errors sooner and a blame strategy based on traditional subtyping. It was previously unknown whether the labeling protocol for threesomes could be adapted to capture D-style blame tracking or the Eager variant of UD. This paper answers that question in the affirmative, and presents the proper protocols and their derivations.

The eager threesome calculi in particular point to some significant efficiencies in our approach. To support consistent blame in the eager systems, failure coercions have interact differently with arrow coercions than in the lazy systems. The result at the threesome level is that composing eager labeled types no longer corresponds to the greatest lower bound of the two types, i.e., $\bot \ \& \ (T_1 \to T_2) = \bot$ does not apply for the type component of eager threesomes. This suggests that the original approach of experimenting with labels on threesomes without blame might have led to a long-lasting dead end.

In addition to new threesome-based semantics, the paper offers a fresh perspective on the original labeled types and Threesome Calculus. Labeled types were developed from scratch, starting from the concept of threesomes without blame, and adding blame labels later. We construct the same concept, but do so from a different starting point, normal coercions, and proceed by stepwise refinement, maintaining correctness at each step. This process reveals that threesomes with blame can be viewed as a streamlined implementation strategy for coercions, and exposes some of the rich structure contained in the original system.

The Blame Calculus is but one point in a rich design space of cast-based languages. If such languages are to have practical impact, they will need to be scaled beyond the basic lambda calculus to support more types (e.g., [Ahmed et al., 2011, Siek and Taha, 2007]). As another step toward full-fledged implementations, we will endeavor to extend our threesome calculi with these features, most likely by first extending their simpler coercion calculus counterparts.

8 Acknowledgements

We thank Jeremy Siek and Éric Tanter for motivating this work, for ongoing experiments with its results, and for discussions about this paper. We also thank anonymous reviewers for their helpful comments about an earlier version of this paper.

$l \in \textsc{Labels}, \quad B \in \textsc{BasicType}, \quad T \in \textsc{Types}, \quad \ddot{c} \in \textsc{SuperC}$

$\textsc{SuperC}^{T_2 \Leftarrow T_1} = \{\, \ddot{c} \mid \vdash \ddot{c} : T_2 \Leftarrow T_1 \,\}$

$$
\begin{array}{rcl}
p & ::= & \epsilon \mid l \\
H & ::= & \epsilon \mid Bl \\
\ddot{c} & ::= & \iota_\star \mid \mathtt{Fail}^{lH} \mid B^p \mid \ddot{c}^v \ {}^p{\to}^p \ \ddot{c}^v
\end{array}
$$

$$\frac{}{\vdash \iota_\star : \star \Leftarrow \star} \qquad \frac{}{\vdash \mathtt{Fail}^{l\epsilon} : T_2 \Leftarrow T_1} \qquad \frac{}{\vdash \mathtt{Fail}^{l_1 B l_2} : T \Leftarrow \star}$$

$$\frac{}{\vdash B^\epsilon : B \Leftarrow B} \qquad \frac{}{\vdash B^\epsilon : \star \Leftarrow B} \qquad \frac{}{\vdash B^l : B \Leftarrow \star} \qquad \frac{}{\vdash B^l : \star \Leftarrow \star}$$

$$\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^v \ {}^\epsilon{\to}^\epsilon \ \ddot{c}_2^v : T_1 \to T_4 \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1^v : \star \Leftarrow \star \quad \vdash \ddot{c}_2^v : \star \Leftarrow \star}{\vdash \ddot{c}_2^v \ {}^\epsilon{\to}^l \ \ddot{c}_1^v : \star \Leftarrow \star} \qquad \frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow \star \quad \vdash \ddot{c}_2^v : \star \Leftarrow T_3}{\vdash \ddot{c}_2^v \ {}^\epsilon{\to}^\epsilon \ \ddot{c}_1^v : \star \Leftarrow T_2 \to T_3}$$

$$\frac{\vdash \ddot{c}_1^v : \star \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow \star}{\vdash \ddot{c}_2^v \ {}^\epsilon{\to}^l \ \ddot{c}_1^v : T_1 \to T_4 \Leftarrow \star}$$

$$\frac{\vdash \ddot{c}_1^v : T_2 \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow T_3}{\vdash \ddot{c}_1^v \ {}^{l_1}{\to}^\epsilon \ \ddot{c}_2^v : T \Leftarrow T_2 \to T_3} \qquad \frac{\vdash \ddot{c}_1^v : \star \Leftarrow T_1 \quad \vdash \ddot{c}_2^v : T_4 \Leftarrow \star}{\vdash \ddot{c}_1^v \ {}^{l_1}{\to}^{l_2} \ \ddot{c}_2^v : T \Leftarrow \star}$$

$\boxed{\bullet : \textsc{SuperC}^{T_3 \Leftarrow T_2} \times \textsc{SuperC}^{T_2 \Leftarrow T_1} \to \textsc{SuperC}^{T_3 \Leftarrow T_1}}$ Composition

$$
\begin{array}{rcl}
\iota_\star \bullet \ddot{c} = \ddot{c} \bullet \iota_\star & = & \ddot{c} \\
\\
B^{p_1} \bullet B^{p_2} & = & B^{p_2} \\
(\ddot{c}_1^v \ {}^{p_1}{\to}^{p_2} \ \ddot{c}_2^v) \bullet (\ddot{c}_3^v \ {}^\epsilon{\to}^{p_3} \ \ddot{c}_4^v) & = & (\ddot{c}_3^v \bullet \ddot{c}_1^v) \ \widehat{{}^{p_1 \to p_3}} \ (\ddot{c}_2^v \bullet \ddot{c}_4^v) \\
\\
B_1^l \bullet B_2^p & = & \mathtt{Fail}^{l\left[B_2^p\right]} \text{ if } B_1 \neq B_2 \\
B^l \bullet (\ddot{c}_1^v \ {}^{p_1}{\to}^{p_2} \ \ddot{c}_2^v) & = & (\ddot{c}_1^v \ {}^l{\to}^{p_2} \ \ddot{c}_2^v) \\
(\ddot{c}_1^v \ {}^{p_1}{\to}^l \ \ddot{c}_2^v) \bullet B^{p_2} & = & \mathtt{Fail}^{l\left[B^{p_2}\right]} \\
\\
\ddot{c} \bullet \mathtt{Fail}^{lH} & = & \mathtt{Fail}^{lH} \\
\ddot{c} \bullet (\ddot{c}_1^v \ {}^{l_1}{\to}^{p_2} \ \ddot{c}_2^v) & = & \ddot{c}_1^v \ {}^{l_1}{\to}^{p_2} \ \ddot{c}_2^v \\
\\
\mathtt{Fail}^l \bullet B^p & = & \mathtt{Fail}^{l\left[B^p\right]} \\
\mathtt{Fail}^l \bullet (\ddot{c}_1^v \ {}^\epsilon{\to}^p \ \ddot{c}_2^v) & = & \ddot{c}_1^v \ {}^l{\to}^p \ \ddot{c}_2^v \\
\\
\mathtt{Fail}^{l_1 B l_2} \bullet B^p & = & \mathtt{Fail}^{l_1\left[B^p\right]} \\
\\
\mathtt{Fail}^{l_1 B_1 l_2} \bullet B_2^p & = & \mathtt{Fail}^{l_2\left[B_2^p\right]} \text{ if } B_1 \neq B_2 \\
\mathtt{Fail}^{l_1 B l_2} \bullet (\ddot{c}_1^v \ {}^\epsilon{\to}^p \ \ddot{c}_2^v) & = & \ddot{c}_1^v \ {}^{l_2}{\to}^p \ \ddot{c}_2^v
\end{array}
$$

where

$$
\begin{array}{rl}
\left[B^\epsilon\right] = \epsilon & \ddot{c}_1^v \ \widehat{{}^{p_1 \to p_2}} \ \ddot{c}_2^v = \ddot{c}_1^v \ {}^{p_1}{\to}^{p_2} \ \ddot{c}_2^v \\
\left[B^l\right] = Bl & \mathtt{Fail}^{\widehat{l_1 \ {}^p{\to}^\epsilon}} \ \ddot{c}_2 = \\
 & \ddot{c}_1^v \ \widehat{{}^{p \to \epsilon} \ \mathtt{Fail}^{l_1}} = \mathtt{Fail}^{l_1} \\
 & \mathtt{Fail}^{\widehat{l_1 \ {}^p{\to}^{l_2}}} \ \ddot{c}_2 = \\
 & \ddot{c}_1^v \ \widehat{{}^{p \to l_2} \ \mathtt{Fail}^{l_1}} = \iota_\star \ {}^{l_1}{\to}^{l_2} \ \iota_\star
\end{array}
$$

Figure 27: Eager UD Labeled Types

References

M. Abadi, L. Cardelli, B. Pierce, and G. Plotkin. Dynamic typing in a statically typed language. *Trans. Programming Languages and Systems*, 13(2):237–268, April 1991.

A. Ahmed, R. B. Findler, J. Siek, and P. Wadler. Blame for all. In *Proc. Symposium on Principles of Programming Languages*, POPL '11, pages 201–214, New York, NY, USA, 2011. ACM.

R. M. Burstall and J. Darlington. Some transformations for developing recursive programs. In *Proc. International Conference on Reliable Software*, pages 465–472, New York, NY, USA, 1975. ACM. doi: 10.1145/800027.808470.

R. B. Findler and M. Felleisen. Contracts for higher-order functions. In *Int. Conf. on Functional Programming*, October 2002.

F. Henglein. Dynamic typing: syntax and proof theory. *Science of Computer Programming*, 22(3):197–230, June 1994.

D. Herman, A. Tomb, and C. Flanagan. Space-efficient gradual typing. In *Trends in Functional Prog.*, page XXVIII, April 2007.

L. Ina and A. Igarashi. Gradual typing for generics. In *Proc. International Conference on Object-oriented Programming Systems Languages and Applications*, OOPSLA '11, pages 609–624, New York, NY, USA, 2011. ACM.

A. Rastogi, A. Chaudhuri, and B. Hosmer. The ins and outs of gradual type inference. In *Proc. Symposium on Principles of Programming Languages*, POPL '12, pages 481–494, New York, NY, USA, 2012. ACM.

I. Sergey and D. Clarke. Gradual ownership types. In *Proc. European Conference on Programming Languages and Systems*, ESOP'12, pages 579–599, Berlin, 2012. Springer-Verlag.

J. Siek. Strong normalization for coercion calculi. Unpublished Manuscript, 2011a.

J. Siek, 2011b. Private Correspondence.

J. Siek and W. Taha. Gradual typing for functional languages. In *Proc. Scheme and Functional Programming Workshop*, Sept. 2006.

J. Siek and W. Taha. Gradual typing for objects. In *Proc. European Conference on Object-Oriented Programming*, ECOOP '07, pages 2–27, Berlin, 2007. Springer-Verlag.

J. Siek and M. Vachharajani. Gradual typing with unification-based inference. In *Proc. Symposium on Dynamic languages*, DLS '08, pages 7:1–7:12, New York, NY, USA, 2008. ACM.

J. Siek and P. Wadler. Threesomes, with and without blame. In *Proc. Symposium on Principles of Programming Languages*, POPL '10, pages 365–376, New York, NY, USA, 2010. ACM.

J. Siek, R. Garcia, and W. Taha. Exploring the design space of higher-order casts. In *Proc. European Symposium on Programming Languages*, ESOP '09, pages 17–31, Berlin, 2009. Springer-Verlag.

S. Tobin-Hochstadt and M. Felleisen. The design and implementation of Typed Scheme. In *Proc. Symposium on Principles of Programming Languages*, POPL '08, pages 395–406, New York, NY, USA, 2008. ACM.

P. Wadler and R. B. Findler. Well-typed programs can't be blamed. In *Proc. European Symposium on Programming Languages*, ESOP '09, pages 1–16, Berlin, 2009. Springer-Verlag.

T. Wrigstad, F. Z. Nardelli, S. Lebresne, J. Östlund, and J. Vitek. Integrating typed and untyped code in a scripting language. In *Proc. Symposium on Principles of Programming Languages*, POPL '10, pages 377–388, New York, NY, USA, 2010. ACM.