Correct Separate and Selective Closure Conversion

Paul Steckler*

Keywords: dataflow analysis, escape analysis, closure conversion, modules, regular trees

Abstract

We describe how to separately analyze a program context and a term substitution in an untyped higher-order language, in order to perform selective closure conversion. Each node of the parse trees for the context and the substitution is annotated by solving constraints; the solutions may be considered regular trees. A translation merges the separate annotations into annotations for the substituted context. Substituting the transformed term into the transformed context is equivalent to transforming the substituted context. Using earlier results for selective closure conversion, the separate transformation is shown correct.

1 Introduction

When compiling separate code modules, it is difficult to know what information will flow from one module to another. In compiling an individual module, an optimizing compiler that makes use of dataflow information may have no knowledge of the dataflow properties of imported information. If the language is also higher-order, procedures from one module may leak into another. For an optimizing closure conversion transformation, cross-module procedure flow is a critical issue, because the analysis and transformation rely on knowledge about which procedures may be applied at which call sites.

In a series of papers, Wand and the present author showed how to analyze and transform higher-order untyped programs so that the transformations could be proved correct [10] [7] [6]. Full correctness proofs are given in [5]. Each of the analyses is *global*, that is, the analyses require the entire program. Two of those

*LFCS, Department of Computer Science, The King's Buildings, University of Edinburgh, Edinburgh EH9 3JZ, Scotland, `steck@dcs.ed.ac.uk`.

analyses supported *selective and lightweight closure conversion* transformations. In the output language of those transformations, closures are given as source-level records. Selectivity means that some procedures in the input program source may not become closures under the transformation. Lightweightness means that bindings for free variables in the procedure may sometimes be omitted from closures, because the required values are available at call sites. Since some procedures do not become closures, and those that do may require additional call-site arguments, each procedure expects to obey a particular application protocol. Under the transformation, each call site is transformed so that procedures flowing to that site will use a particular application protocol. The essential problem of an analysis for selective and lightweight closure conversion is to make sure that all procedures flowing to a call site expect the protocol used at that site.

For example, consider a call site MN. Closure records are applied by using a combinator **app**. If the call site is transformed to **app** MN, the operator M should always evaluate to a closure record. If instead the call site is left unchanged by the transformation (it is still MN), M should always evaluate to ordinary procedures. If procedures may flow across module boundaries, enforcing such restrictions becomes more difficult.

In this report we are more concerned with separate analysis and its soundness than with the closure conversion transformation itself, which has been explained elsewhere. Therefore, we restrict the transformation to selective closure conversion, omitting details relating to lightweightness. Restoring those details to the analysis should be straightforward.

2 Separate analysis

Our model for module composition is *context substitution*. A context is a term with a "hole" in it, which can be filled with a program term. We may write a context as $C[\,]$; the hole is written as $\_$. If we fill the hole with a term substitution M, we obtain the term $C[M]$. Of course, real programming languages have more complex module systems; the methods here should be extendable to them.

Our dataflow analysis is founded on a closure analysis, which conservatively estimates the set of procedures to which a term may evaluate. The usual techniques for performing closure analysis are modified to accomodate separate analysis. For instance, consider the context $C[\,] = (\lambda x.x)\ \_$, and consider the substituted context $C[M]$. Any procedure to which M may evaluate may be passed to the procedure $(\lambda x.x)$ in the context. Now let $C[\,] = \_\ N$, and suppose $M = (\lambda y.y)$. In $C[M]$, any procedure to which N may evaluate may be passed to the procedure $(\lambda y.y)$ in the term substitution. A separate analysis has to contemplate *crossings* of module boundaries.

Once a procedure has crossed a module boundary, it may be applied, and return another procedure from the foreign module. Let MN be a call site in a context, and suppose M may evaluate to the procedure $(\lambda x.\lambda y.P)$ from a term substitution. Then MN may evaluate to $(\lambda y.P)$, which is a procedure from the term substitution. For procedures from the context which leak into the term substitution, the situation is symmetric. Therefore, the separate analysis has to account for the *propagation* of foreign procedures within modules.

When separately analyzing a context, we cannot know what procedures are in the term substitution, and vice-versa. Instead, we model the worst-case behaviors of procedures from foreign modules by introducing a *token* which represents foreign procedures and procedures which may have escaped from the current module. To see how the token models procedure dataflow, consider again the call site MN described in the last paragraph. Suppose the token is associated with the operator, representing procedures from the term substitution. In that case, since the application may evaluate to a foreign procedure, we also associate the token with the application itself.

Besides providing a sound closure analysis, the separate analysis has to adopt some convention about the protocol expected by procedures that cross module boundaries. A procedure that crosses a module boundary may no longer have the bindings for its free variables available. Therefore, even though we do not know which procedures may cross module boundaries, we want all such "escaping" procedures to become closures under the selective closure conversion transformation. Closure avoidance is a strictly local optimization. Requiring escaping procedures to become closures, while allowing local closure avoidance, is the principal difficulty in the separate analysis.

Before the work described here, we already knew how to correctly perform global annotation and substitution. In this report, we show that substitution of a separately annotated and transformed term into a separately annotated and transformed context is equivalent to transforming the substituted context according to the information in a global annotation derived from the separate annotations. Therefore, our earlier correctness results indicate that the separate annotation and transformation are also correct.

3 An example

Let us give an example of how separate analysis provides sufficient information for a sound global analysis. We describe how to annotate a context and term substitution with tags indicating which procedures should become closure records under the closure conversion transformation. This example is meant as illustration; we omit many of the details described later in the report.

Consider the context:

$$\textbf{if } M \textbf{ then } (\_\ (\lambda x.\lambda y.xy)) \textbf{ else } (\lambda z.z)$$

According to the annotation rules, we associate the token with the occurrence of the context hole. Since the application of a foreign procedure may return a foreign procedure, we associate the token with the application in the then-part of the conditional. Because the result of the then-part may be the result of the conditional itself, we also associate the token with the conditional. Now, $(\lambda x.\lambda y.xy)$ is passed to the term substitution, where it may be applied to a foreign procedure, so we associate the token with the formal parameter x and the occurrence of x in the procedure body. The result of such an application will be a substitution instance of $(\lambda y.xy)$, which also may be applied in the term substitution. Therefore, the token is associated with the formal y and with the occurrence of y in the procedure body. Since the token is associated with the operator of the application xy, it is also associated with that application. Let us rewrite the context with a superscript τ representing the token annotation:

$$(\textbf{if } M \textbf{ then } (\_^{\tau}\ (\lambda x^{\tau}.\lambda y^{\tau}.(x^{\tau} y^{\tau})^{\tau}))^{\tau} \textbf{ else } (\lambda z.z))^{\tau}$$

Now we wish to determine which procedures from the context may be applied in the term substitution, and so, by our convention, should become closure records. To do so, we use the token annotations just computed. Since the token appears in operator position at the hole, any procedure that may escape from the operand may be applied in the term substitution. In this case, the potentially escaping procedures are $(\lambda x.\lambda y.xy)$ and $(\lambda y.xy)$. Since the then-part may return a foreign procedure, and the then- and else-parts should agree on protocol, we also want $(\lambda z.z)$ to become a closure record. Let us write a superscript cl next to each procedure occurrence that should become a closure record under the transformation:

$$\textbf{if } M \textbf{ then } (\_\ (\lambda x.(\lambda y.xy)^{cl})^{cl}) \textbf{ else } (\lambda z.z)^{cl}$$

Note that none of the procedures in M needs to be marked *cl*.

Suppose the term substitution is $(\lambda p.p(\lambda q.q))$. Since this term may be applied in the context to a foreign procedure, we associate the token with the formal p and the variable p in the body. Since the application of a foreign procedure may return a foreign procedure, we associate the token with the application $p(\lambda q.q)$. Because $(\lambda q.q)$ may be applied outside the term substitution, the token is associated with the formal parameter q and the variable q in the procedure body:

$$(\lambda p^{\tau}.(p^{\tau}(\lambda q^{\tau}.q^{\tau}))^{\tau})$$

Either of the procedures in this term may be applied in the context, so both are tagged *cl*:

$(\lambda p.p(\lambda q.q)^{cl})^{cl}$

We claim that these separate annotations are sound for the substituted context, in the sense that procedures from the context applied in the term substitution are tagged *cl*; the same holds for procedures flowing from the term to the context. The transformation depends only on the tag information, so here we can disregard the token:

$$\textbf{if } M \textbf{ then } ((\lambda p.p(\lambda q.q)^{cl})^{cl}\ (\lambda x.(\lambda y.xy)^{cl})^{cl}) \textbf{ else } (\lambda z.z)^{cl}$$

Suppose that M evaluates to **true**, so that under call-by-value, the evaluation tree is as in Figure 1. There are two applications here. The procedure $(\lambda p.p(\lambda q.q))$ from the term substitution is applied within the context; appropriately, it is tagged *cl*. The procedure $(\lambda x.(\lambda y.xy))$ is from the context, and it is applied within the term substitution; it also is tagged *cl*. The reader may wish to come back to this example after going through the sections 7, 11, 12, where we describe in detail how to annotate contexts and term substitutions.

The outline of the remainder of the report is:

1. In Section 4, we describe the languages of source terms, contexts, and the output language produced by the transformation. We give a call-by-value evaluator for terms in the source and output languages.

2. In Section 5, we describe an occurrence evaluator, which simulates the term evaluator, but keeps track of particular occurrences of terms and their substitutions.

3. In Section 6, we describe the syntax and semantics of program annotations.

4. In Section 7, we give the constraints which when solved provide sound annotations for a *global* analysis.

5. In Section 8, we describe why we need to weaken the constraints here, compared with similar constraints in previous work on closure conversion.

6. In Section 9, we describe how program annotations correspond to regular trees.

7. In Section 10, we specify the selective closure conversion transformation.

8. In Section 11, we describe how to separately annotate a context and term.

9. In Section 12, we specify a translation for merging the separate annotations for a context and term substitution into annotations for the substituted context. We show that separate annotation and transformation is equivalent to a global annotation and transformation.

$$
\begin{array}{l}
\textbf{if } M \textbf{ then } ((\lambda p.p(\lambda q.q))\ (\lambda x.(\lambda y.xy)))\ \textbf{else}\ (\lambda z.z)\\
\quad M\\
\quad \Downarrow \ \cdots\\
\quad \textbf{true}\\
\\
\quad (\lambda p.p(\lambda q.q))\ (\lambda x.(\lambda y.xy))\\
\qquad (\lambda p.p(\lambda q.q))\\
\qquad \Downarrow\\
\qquad (\lambda p.p(\lambda q.q))\\
\\
\qquad (\lambda x.(\lambda y.xy))\\
\qquad \Downarrow\\
\qquad (\lambda x.(\lambda y.xy))\\
\\
\qquad (\lambda x.(\lambda y.xy))(\lambda q.q) = (p(\lambda q.q))[(\lambda x.(\lambda y.xy))/p]\\
\qquad\quad (\lambda x.(\lambda y.xy))\\
\qquad\quad \Downarrow\\
\qquad\quad (\lambda x.(\lambda y.xy))\\
\\
\qquad\quad (\lambda q.q)\\
\qquad\quad \Downarrow\\
\qquad\quad (\lambda q.q)\\
\\
\qquad\quad (\lambda y.(\lambda q.q)y) = (\lambda y.xy)[(\lambda q.q)/x]\\
\qquad\quad \Downarrow\\
\qquad \Downarrow\ (\lambda y.(\lambda q.q)y)\\
\quad \Downarrow\ (\lambda y.(\lambda q.q)y)\\
\Downarrow\ (\lambda y.(\lambda q.q)y)\\
(\lambda y.(\lambda q.q)y)
\end{array}
$$

Figure 1: Evaluation of the substituted context

10. We describe some related work in Section 13, and present conclusions in Section 14.

4 The language

Our input language Λ_{in} is an untyped λ-calculus with constants, boolean constants, and conditionals:

$$\begin{array}{rcl} M & ::= & x \mid c \mid \mathbf{true} \mid \mathbf{false} \mid \lambda x.M \mid \mathit{PrimOp}\ M \mid MM \mid \\ & & \mathbf{if}\ M\ \mathbf{then}\ M\ \mathbf{else}\ M \end{array}$$

where

$$\mathit{PrimOp} \ ::= \ \mathbf{succ} \mid \mathbf{pred} \mid \mathbf{zero?}$$

We will refer to λ-abstractions as "procedures." Procedures are curried; for convenience we may write $\lambda\vec{x}.M$ for the procedure $\lambda x_1.\ \ldots\ .\lambda x_n.M$. We may write a sequence of variables as $\vec{u}$; the set of variables in a sequence $\vec{u}$ is written as $\lceil\vec{u}\rceil$.

A *context* $C[\,]$ is a Λ_{in} term with a "hole" in it:

$$\begin{array}{rcl} C[\,] & ::= & \_ \mid \lambda x.C[\,] \mid \mathit{PrimOp}\ C[\,] \mid C[\,]\ M \mid M\ C[\,] \mid \\ & & \mathbf{if}\ C[\,]\ \mathbf{then}\ M\ \mathbf{else}\ M \mid \\ & & \mathbf{if}\ M\ \mathbf{then}\ C[\,]\ \mathbf{else}\ M \mid \\ & & \mathbf{if}\ M\ \mathbf{then}\ M\ \mathbf{else}\ C[\,] \end{array}$$

where M is a metavariable ranging over terms in Λ_{in}.

Observe that if we substitute a Λ_{in} term M for the hole in a context $C[\,]$, then the substituted context $C[M]$ is a term in Λ_{in}.

The closure conversion transformation, when applied to terms in Λ_{in}, produces terms in an output language Λ_{clos}, given by the grammar:

$$\begin{array}{rcl} M & ::= & x \mid c \mid \mathbf{true} \mid \mathbf{false} \mid \lambda x.M \mid \mathit{PrimOp}\ M \mid MM \mid \\ & & \mathbf{if}\ M\ \mathbf{then}\ M\ \mathbf{else}\ M \mid [M, \ldots, M] \mid \mathbf{destr}\ M \end{array}$$

where $[M, \ldots, M]$ is a record of terms. If the value of M is a record with n components, and the value of a term N is a curried procedure of n arguments, then the term $\mathbf{destr}\ MN$ applies the procedure to the n record components.

When applied to a context, the closure-conversion transformation produces an *output context*, essentially a term in Λ_{clos} with a hole in it. We omit the grammar.

A *value* is any constant, boolean constant, procedure, or record of values in Λ_{in} or Λ_{clos}.

Both input and output languages use call-by-value evaluation. We define an evaluation relation $\underset{t}{\Longrightarrow}$ between terms and values. Some of the rules defining $\underset{t}{\Longrightarrow}$ are given in Figure 2.

$$x \underset{t}{\Longrightarrow} x \qquad \lambda x.M \underset{t}{\Longrightarrow} \lambda x.M$$

$$\frac{M \underset{t}{\Longrightarrow} \lambda x.Q \quad N \underset{t}{\Longrightarrow} N' \quad Q[N'/x] \underset{t}{\Longrightarrow} V}{MN \underset{t}{\Longrightarrow} V}$$

$$\frac{M \underset{t}{\Longrightarrow} \mathbf{true} \quad N \underset{t}{\Longrightarrow} V}{\mathbf{if}\ M\ \mathbf{then}\ N\ \mathbf{else}\ P \underset{t}{\Longrightarrow} V} \qquad \frac{M \underset{t}{\Longrightarrow} \mathbf{false} \quad P \underset{t}{\Longrightarrow} V}{\mathbf{if}\ M\ \mathbf{then}\ N\ \mathbf{else}\ P \underset{t}{\Longrightarrow} V}$$

$$\frac{\forall i, 1 \leq i \leq n, M_i \underset{t}{\Longrightarrow} V_i}{[M_1, \ldots, M_n] \underset{t}{\Longrightarrow} [V_1, \ldots, V_n]}$$

$$\frac{M \underset{t}{\Longrightarrow} [V_1, \ldots, V_n] \quad N \underset{t}{\Longrightarrow} \lambda x_1 \ldots x_n.Q \quad Q[V_1/x_1, \ldots, V_n/x_n] \underset{t}{\Longrightarrow} W}{\mathbf{destr}\ M\ N \underset{t}{\Longrightarrow} W}$$

Figure 2: Term evaluator

$$\frac{Var(i)}{(i, \psi) \underset{oc}{\Longrightarrow} \psi([\![i]\!])} \qquad \frac{Abs(i)}{(i, \psi) \underset{oc}{\Longrightarrow} (i, \psi)}$$

$$\frac{\begin{array}{l} Cond(i) \\ (i.test, \psi) \underset{oc}{\Longrightarrow} (j, \psi') \\ [\![j]\!] = \mathbf{true} \\ (i.then, \psi) \underset{oc}{\Longrightarrow} (k, \psi'') \end{array}}{(i, \psi) \underset{oc}{\Longrightarrow} (k, \psi'')} \qquad \frac{\begin{array}{l} Cond(i) \\ (i.test, \psi) \underset{oc}{\Longrightarrow} (j, \psi') \\ [\![j]\!] = \mathbf{false} \\ (i.else, \psi) \underset{oc}{\Longrightarrow} (k, \psi'') \end{array}}{(i, \psi) \underset{oc}{\Longrightarrow} (k, \psi'')}$$

$$\frac{\begin{array}{l} App(i) \\ (i.rator, \psi) \underset{oc}{\Longrightarrow} (j, \psi'), \quad Abs(j) \\ (i.rand, \psi) \underset{oc}{\Longrightarrow} (k, \psi'') \\ (j.body, \psi'[[\![j.bv]\!] \mapsto (k, \psi'')]) \underset{oc}{\Longrightarrow} (m, \psi''') \end{array}}{(i, \psi) \underset{oc}{\Longrightarrow} (m, \psi''')}$$

Figure 3: Occurrence evaluator

5 The occurrence evaluator

In our analysis, annotations will be associated with particular program occurrences. Therefore, to show the soundness of our analysis, we want to keep track of program occurrences during computation. To do so, we use an *occurrence evaluator* which simulates the actions of the call-by-value term evaluator. To refer to particular *occurrences* of terms, we assign each parse tree node in the program a finite string from the alphabet

$$\{rator, rand, bv, body, test, then, else\}$$

which describes a path from the root of the parse tree to a given node. So the empty string, written as ϵ, represents the root of the tree; $i.rator$ represents the operator of an application with index i; and so on. Occurrences in contexts are assigned indices in the same way. We similarly provide indices for constants, and for occurrences that appear in environments.

We write $[\![i]\!]_M$ for the subterm of M with index i. The term M that parameterizes $[\![\cdot]\!]$ may be a term as such, or a context. Of course, the same occurrence index may refer to many subterms, depending on the terms or contexts in which they are found. Since it is usually clear in which term an occurrence is found, we will usually omit the term subscript.

The predicates *Var*, *Abs*, *App*, *PrimApp*, and *Cond* test whether an occurrence argument refers to a variable, abstraction (procedure), application, primitive application, or conditional, respectively. The predicate *Const* is true if its occurrence argument refers to an ordinary constant or a boolean constant. The predicate *Hole* tests whether its argument refers to a context hole. As for $[\![\cdot]\!]$, we can think of these predicates as parameterized by a particular term or context. However, we will be informal in our use of such predicates, omitting mention of the term in which an occurrence appears.

We next define the data structures manipulated by the occurrence evaluator.

Definition 1 *The following definitions are mutually referential.*

1. *An* occurrence closure (i, ψ) *is a pair consisting of an occurrence index and an occurrence environment.*
2. *An* occurrence environment ψ *is a finite map from variables to occurrence closures, where for each* $x \in Dom(\psi)$, *if* $\psi(x) = (i, \psi')$ *then* $[\![i]\!]$ *is a value in* Λ_{in}.

We define an evaluation relation $\underset{oc}{\Longrightarrow}$ between occurrence closures that represent terms and occurrence closures that represent values. Representative rules for the occurrence evaluator are given in Figure 3. The actions of the occurrence evaluator simulate the substitutions of the term evaluator [5].

6 Annotations

We annotate each program occurrence with an *environment proposition* and a *value proposition*, which we may think of as an input and an output assertion, respectively.

Value propositions include the following information:

1. A *flow*, which describes the set of procedures to which a given occurrence might evaluate, and which possibly includes a *procedure token*, representing a set of "foreign" procedures, and
2. A *protocol tag*, which indicates an application protocol, either ordinary procedure application, or closure record application.

More formally, we can state:

Definition 2 *The following definitions are mutually referential.*

1. *A* value proposition $\mathcal{P}$ *is a pair consisting of a flow and a protocol tag.*
2. *A* flow ϕ *is a finite set consisting of abstract closures and possibly a procedure token.*
3. *An* abstract closure (i, A) *is a pair consisting of the occurrence index of an abstraction and an environment proposition.*
4. *A* protocol tag π *is an element of the set* $\{id, cl\}$*.*
5. *A* procedure token *is the symbol* τ*.*
6. *An* environment proposition A *is a finite map from variables to value propositions.*

We wish to annotate each occurrence i with an environment proposition A_i and a value proposition $\mathcal{P}_i$ such that if this occurrence i is evaluated in an occurrence environment ψ that satisfies A_i, then the output satisfies the proposition $\mathcal{P}_i$. We now give a semantics for environment propositions and value propositions to formalize these notions of satisfaction.

Define a *protocol assignment* to be a map from occurrences to protocol tags.

Definition 3 *The following definitions are mutually recursive:*

1. *An occurrence environment* ψ *satisfies an environment proposition* A *under a protocol assignment* Π*, as follows:*

$$\psi \models^{\Pi}_{env} A$$

iff $\forall x \in Dom(A)$*,* $x \in Dom(\psi)$ *and* $\psi(x) \models^{\Pi}_{val} A(x)$*.*

2. *An occurrence closure* (i, ψ) *satisfies a proposition* (ϕ, π) *under a protocol assignment* Π*, as follows:*

$$(i, \psi) \models^{\Pi}_{val} (\phi, \pi)$$

iff

   (a) $Var(i) \vee Const(i)$*, or*

   (b) $Abs(i)$*,* $\Pi(i) = \pi$*, and* $\exists A$ *such that* $(i, A) \in \phi$ *and* $\psi \models^{\Pi}_{env} A$*.*

By these definitions, we see that $\models^{\Pi}_{env}$ tells us what it means for an occurrence environment (an input) to satisfy an environment proposition; and $\models^{\Pi}_{val}$ tells us how an occurrence closure satisfies a value propostion.

The set of value propositions is ordered as follows: Say that $(\phi, \pi) \leq (\phi', \pi')$ iff $\phi \subseteq \phi'$ and $\pi = \pi'$. For environment propositions, say that $A \leq A'$ iff $\forall x \in Dom(A')$, $x \in Dom(A)$ and $A(x) \leq A'(x)$.

So:

Lemma 1

1. *If* $(i, \psi) \models^{\Pi}_{val} (\phi, \pi)$ *and* $(\phi, \pi) \leq (\phi', \pi')$*, then* $(i, \psi) \models^{\Pi}_{val} (\phi', \pi')$*.*
2. *If* $A \leq A'$*, then* $\psi \models^{\Pi}_{env} A$ *implies* $\psi \models^{\Pi}_{env} A'$*.*

Proof. By the orderings on environment propositions and value propositions, and the definitions of $\models^{\Pi}_{env}$ and $\models^{\Pi}_{val}$. ∎

7 Local consistency conditions

An *annotation map* Γ associates each occurrence i with an environment proposition A_i and a value proposition $\mathcal{P}_i = (\phi_i, \pi_i)$. We give a set of constraints for annotation maps, presented as *local consistency conditions*, in Figure 4. These are the same constraints given in [10], except that

$$Var(i) \Longrightarrow A_i(\llbracket i \rrbracket) = \mathcal{P}_i$$

$$Abs(i) \Longrightarrow \begin{cases} A_i[\llbracket i.bv \rrbracket \mapsto \mathcal{P}_{i.bv}] \leq A_{i.body}, \\ \{(i, A_i)\} \subseteq \phi_i \end{cases}$$

$$PrimApp(i) \Longrightarrow A_i \leq A_{i.rand}$$

$$App(i) \Longrightarrow \begin{cases} A_i \leq A_{i.rator}, A_{i.rand}, \\ \forall (j, B) \in \phi_{i.rator} \\ \qquad \begin{cases} Abs(j), \\ \mathcal{P}_{i.rand} \leq \mathcal{P}_{j.bv}, \\ \mathcal{P}_{j.body} \leq \mathcal{P}_i \end{cases} \end{cases}$$

$$Cond(i) \Longrightarrow \begin{cases} A_i \leq A_{i.test}, A_{i.then}, A_{i.else}, \\ \mathcal{P}_{i.then}, \mathcal{P}_{i.else} \leq \mathcal{P}_i \end{cases}$$

Figure 4: Local Consistency Conditions

1. invariance sets, which were used to support lightweightness, have been omitted, and
2. the equality constraints on environment propositions have been weakened to inequality constraints.

If these constraints hold for every occurrence in the domain of an annotation map Γ, we say that Γ is *locally consistent.*

Say that an annotation map Γ is *monovariant* iff for every flow ϕ considered in the annotations of Γ, $(j, B) \in \phi$ implies $B = A_j$. In other words, each procedure is represented by one abstract closure.

Similarly to what we showed in [10] for slightly different annotations, we have:

Theorem 1 (Soundness) *Let Γ be a monovariant and locally-consistent annotation map and let Π be the protocol assignment defined by $\forall i,\ \Pi(i) = \pi_i$. Let i be an occurrence index and ψ an occurrence environment such that $\psi \models^{\Pi}_{env} A_i$. Suppose $(i, \psi) \underset{oc}{\Longrightarrow} (j, \psi')$. Then for any subproof $(k, \psi'') \underset{oc}{\Longrightarrow} (m, \psi''')$ of this derivation, $\psi'' \models^{\Pi}_{env} A_k$ and $(m, \psi''') \models^{\Pi}_{val} (\phi_k, \pi_k)$.*

Proof. Induction on the size of the derivation that $(i, \psi) \underset{oc}{\Longrightarrow} (j, \psi')$. ∎

The proof follows the presentation of the soundness theorem in [5, Chapter 3].

8 Weakening the environment proposition constraints

Producing an annotation that satisfies equality constraints on environment propositions is straightforward when analyzing an entire program, but becomes problematic when the annotations for a context and a term are produced separately. For instance, suppose i is the index of an application in a context, where the hole is at the operand. Now suppose we have separate annotation maps for the context and for the term that will be substituted for the hole. We wish to merge the separate annotation maps into a single annotation map that satisfies the constraints we give for entire programs. For this example, we formerly had the constraints that $A_i = A_{i.rator} = A_{i.rand}$. In the substituted context, the index $i.rand$ now corresponds to the substituted term. Since the annotations for the term substitution are computed separately from those for the context, it is impossible to enforce this constraint. Instead, we weaken this and similar constraints, giving the current set of local consistency conditions. A minimality condition on environment propositions for term substitutions, which we state in Section 11, allows satisfaction of the weaked constraints $A_i \leq A_{i.rator}, A_{i.rand}$.

The question arises whether annotations satisfying the weakened constraints are sound. In the proofs of the theorems corresponding to Theorem 1, we used the equality constraints on environment propositions to show that when a given occurrence environment satisfies the environment proposition associated with a term, the environments that arise during computation for the evaluation of subterms satisfy the environment propositions for those subterms. By the second part of Lemma 1, that same property holds for annotations which are locally consistent with respect to the constraints in Figure 4. Therefore, in the proof of Theorem 1, we can now invoke inequality constraints where in earlier such theorems, we invoked equality constraints.

9 Annotations are regular trees

We can associate environment propositions and value propositions with distinct symbols to be used as labels for tree nodes. For example, a value proposition $\mathcal{P}_i$ might have the associated symbol "$\mathcal{P}_i$". Informally, we can say that the environment propositions and value propositions themselves are the labels of nodes of trees, which may be infinite. For each node labelled with environment proposition A, its children are labelled with the value propositions $\{A(x) \mid x \in Dom(A)\}$. For each node labelled with a value proposition $\mathcal{P} = (\phi, \pi)$, its children are labelled with the environment propositions $\{A \mid (i, A) \in \phi\}$. In a given tree, a subtree may appear more than once. For example, suppose that for an environment proposition A, $A(x) = \mathcal{P} = (\phi, \pi)$, and $(i, A) \in \phi$. Then the node labelled $\mathcal{P}$ has both parent

$$Hole(i) \vee Var(i) \vee Const(i) \Longrightarrow \Phi(i) = [\![i]\!]$$

$$Abs(i) \wedge \pi_i = id \Longrightarrow \Phi(i) = \lambda x.\Phi(i.body)$$

$$\begin{aligned} Abs(i) \wedge \pi_i = cl \Longrightarrow \Phi(i) = \ & [(\lambda e[\![i.bv]\!].\, \mathbf{destr}\ e\ (\lambda\vec{u}.\Phi(i.body))), [\vec{u}]] \\ & \quad \mathbf{where}\ e\ \text{fresh} \\ & \quad \text{and}\ \lceil\vec{u}\rceil = FV([\![i]\!]) \end{aligned}$$

$$PrimApp(i) \Longrightarrow \Phi(i) = [\![i.rator]\!]\ \Phi(i.rand)$$

$$App(i) \wedge \pi_{i.rator} = id \Longrightarrow \Phi(i) = \Phi(i.rator)\ \Phi(i.rand)$$

$$App(i) \wedge \pi_{i.rator} = cl \Longrightarrow \Phi(i) = \mathbf{app}\ \Phi(i.rator)\ \Phi(i.rand)$$

$$Cond(i) \Longrightarrow \Phi(i) = \mathbf{if}\ \Phi(i.test)\ \mathbf{then}\ \Phi(i.then)\ \mathbf{else}\ \Phi(i.else)$$

Figure 5: The closure conversion transformation

and child nodes labelled with A.

A *regular tree* is a tree with finitely many different subtrees. Suppose there are finitely many abstract closures in an annotation map Γ, so that there are finitely many different possible flows. In that case, there are finitely many different possible value propositions in Γ. Therefore, there can be only finitely many different subtrees in any tree constructed from the annotations in Γ. Hence, such a tree is regular. As a special case, if the monovariance condition holds for Γ, the annotations describe regular trees.

10 The closure-conversion transformation

Once a program has been annotated, we are ready to perform closure conversion. The transformation in Figure 5 is as in [10], but does not consider lightweightness. Also, to support separate transformation, we add the *Hole* rule to work on occurrences within contexts. Note that Φ relies on the annotation information in a particular annotation map Γ. When it may be unclear which annotation map is used, we may write Φ_Γ.

The transformation is extended to occurrence closures:

$$\hat{\Phi}(i, \psi) = \Phi(i)\{\hat{\Phi} \circ \psi\}$$

where the braces indicate substitution.

We can show:

Theorem 2 (Correctness) *Let Γ be a monovariant and locally-consistent annotation map, and let Π be the protocol assignment defined by $\forall i, \Pi(i) = \pi_i$. Let i be an occurrence index, and ψ an occurrence environment such that $\psi \overset{\Pi}{\models}_{env} A_i$. If $(i, \psi) \underset{oc}{\Longrightarrow} (j, \psi')$ then $\hat{\Phi}(i, \psi) \underset{t}{\Longrightarrow} \hat{\Phi}(j, \psi')$.*

Proof. Induction on the size of the derivation that $(i, \psi) \underset{oc}{\Longrightarrow} (j, \psi')$. ∎

Again, the proof follows the presentation in [5].

11 Separate annotations

So far, we have recapitulated the global analysis and global transformation for closure conversion described in earlier work, restricted to selective closure conversion. Now we show how our method can be extended to perform separate analysis and separate transformation.

To proceed, we need:

Definition 4 *The* flow closure *of a flow ϕ with respect to an annotation map Γ, written $\mathbf{FClos}_\Gamma(\phi)$, is the smallest set closed under the following rules:*

1. *if $\tau \in \phi$ then $\tau \in \mathbf{FClos}_\Gamma(\phi)$*
2. *if $(j, B) \in \phi$ then $(j, B) \in \mathbf{FClos}_\Gamma(\phi)$*
3. *if $(j, B) \in \mathbf{FClos}_\Gamma(\phi)$ then $\phi_{j.body} \subseteq \mathbf{FClos}_\Gamma(\phi)$*

Intuitively, $\mathbf{FClos}_\Gamma(\phi_i)$ represents the procedures in ϕ_i, and all procedures that may escape as a result of applying the procedures in ϕ_i. We may write $\mathbf{FClos}(\phi)$, omitting the associated annotation map when it is understood.

In Figure 6, we give local consistency conditions for contexts and term substitutions. These conditions are *in addition to* the conditions given in Figure 4. That is, for a given occurrence, we have to satisfy the relevant constraints given in both figures.

Let us provide some insight into these additional conditions. Suppose the premise $App(i) \wedge \tau \in \phi_{i.rator}$ holds. As we have suggested, the token τ is intended to represent "foreign" procedures. When such a procedure is applied, it may return another foreign procedure; hence the constraint $\tau \in \phi_i$. By convention, procedures crossing module boundaries should become closures under the transformation, so we have the constraint $\pi_i = cl$. Since the operator may evaluate to a foreign

$$App(i) \wedge \tau \in \phi_{i.rator} \Longrightarrow \begin{cases} \tau \in \phi_i \\ \pi_i = \pi_{i.rand} = cl \\ \forall (j, B) \in \mathbf{FClos}(\phi_{i.rand}) \\ \quad \begin{cases} \pi_{j.bv} = \pi_{j.body} = cl \\ \tau \in \phi_{j.bv} \end{cases} \end{cases}$$

Figure 6: Additional Local Consistency Conditions

procedure, any procedure that is the value of the operand, or escapes from the value of the operand, may be applied outside the current module. Since the operand itself may be a procedure, we require $\pi_{i.rand} = cl$. In complement to this last constraint, we require $\forall (j, B) \in \mathbf{FClos}(\phi_{i.rand})$, $\pi_{j.bv} = cl$. We say "in complement" because when analyzing a foreign module, the escaping procedures given by $\mathbf{FClos}(\phi_{i.rand})$ will be represented by the token τ; those procedures may be applied to operands which, by the last constraint, are tagged *cl*. When applying escaping procedures to operands in the foreign module, procedures from the foreign module may be passed to the escaping procedures. Therefore, we include the constraint $\forall (j, B) \in \mathbf{FClos}(\phi_{i.rand})$, $\tau \in \phi_{j.bv}$.

For contexts only, we impose the additional *hole* constraint:

$$Hole(i) \Longrightarrow \begin{cases} \tau \in \phi_i, \text{ and} \\ \pi_i = cl \end{cases}$$

We need the first constraint because a term substituted into the hole may evaluate to a foreign procedure. The second constraint reflects the assumption that the substituted term is itself tagged *cl*.

Let ϵ be the index of a term substitution. Besides the local consistency conditons, we impose additional global constraints on annotations for term substitutions:

1. A *minimality* constraint, that $A_\epsilon = \emptyset$,

2. *import* constraints, that if $(j, B) \in \mathbf{FClos}(\phi_\epsilon)$ then $\pi_{j.bv} = cl$ and $\tau \in \phi_{j.bv}$, and

3. *export* constraints, that $\pi_\epsilon = cl$, and, if $(j, B) \in \mathbf{FClos}(\phi_\epsilon)$ then $\pi_{j.body} = cl$.

Since we will require the term substitution to be closed, the minimality condition does not conflict with the local condition that requires variables to be in the domain of their associated environment propositions. The import and export constraints are much like those in Figure 6; they handle the case where a term is substituted into a hole in the operand position of an application within a context.

In that case, from the standpoint of the substituted term, the operator may evaluate to a foreign procedure, so the situation is like the premise $App(i) \wedge \tau \in \phi_{i.rator}$ for the constraints in Figure 6. Therefore, we can justify the import and export constraints for similar reasons. As it turns out, the import and export constraints are sufficient to handle other substitution positions within contexts.

12 Merging the separate annotations

We have now provided a way to separately annotate a context and a term substitution. But our soundness and correctness results are based on annotations satisfying the global constraints in Figure 4. We now relate the separate annotations to annotations for the substituted context.

Let $C[\,]$ be a context, and let M be a term substitution. Now let $\Gamma_{\mathcal{C}}$ be an annotation map for $C[\,]$, and let $\Gamma_{\mathcal{M}}$ be an annotation map for M. From these separate annotations, we can derive an annotation map Γ for the substituted context $C[M]$, using the translations given in Figure 7. We define the predicate

$$TApp(i) \equiv App(i) \wedge \tau \in \phi_{i.rator} \wedge \neg Hole(i.rand)$$

As for other predicates, we omit parameterizing *TApp* by a term, as we might if we wished to be more formal.

Note that an occurrence i in the substituted context comes from either the context or from the term substitution. Occurrences from $C[\,]$ will have the same occurrence index in $C[M]$. All occurrences from the term substitution will have an index consisting of some (possibly empty) prefix representing a path from the root of the context parse tree to the hole, followed by the original index of the occurrence in the term.

Observe:

Lemma 2 *Both $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}$ and $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}$ are monotonic with respect to the subset ordering on flows, that is, if $\phi \subseteq \phi'$, then*

1. $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi) \subseteq \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi')$, *and*
2. $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi) \subseteq \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi')$

Proof. By the definitions of the transformations. ∎

Also:

Lemma 3 *If $(\phi, \pi) \leq (\phi', \pi')$, then*

$$
\begin{aligned}
&\mathcal{T}_{map}(\Gamma_{\mathcal{C}}, \Gamma_{\mathcal{M}}) = \Gamma, \text{ where} \\
&\quad Dom(\Gamma) = Dom(\Gamma_{\mathcal{C}}) \cup \{h.i \mid i \in Dom(\Gamma_{\mathcal{M}})\}, \\
&\quad \forall i \in Dom(\Gamma_{\mathcal{C}}),\ i \neq h,\ \Gamma_{\mathcal{C}}(i) = (A, (\phi, \pi)) \Longrightarrow \\
&\qquad\qquad\qquad \Gamma(i) = (\mathcal{T}^{\mathcal{C}}_{env}(A), (\mathcal{T}^{\mathcal{C}}_{flow}(\phi), \pi)), \text{ and} \\
&\quad \forall i \in Dom(\Gamma_{\mathcal{M}}),\ \Gamma_{\mathcal{M}}(i) = (A, (\phi, \pi)) \Longrightarrow \\
&\qquad\qquad\qquad \Gamma(h.i) = (\mathcal{T}^{\mathcal{M}}_{env}(A), (\mathcal{T}^{\mathcal{M}}_{flow}(\phi), \pi))
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{T}^{\mathcal{C}}_{env}(A) = A', \text{ where} \\
&\quad Dom(A') = Dom(A), \text{ and} \\
&\quad \forall x \in Dom(A),\ A(x) = (\phi, \pi) \Longrightarrow A'(x) = (\mathcal{T}^{\mathcal{C}}_{flow}(\phi), \pi)
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{T}^{\mathcal{M}}_{env}(A) = A', \text{ where} \\
&\quad Dom(A') = Dom(A), \text{ and} \\
&\quad \forall x \in Dom(A),\ A(x) = (\phi, \pi) \Longrightarrow A'(x) = (\mathcal{T}^{\mathcal{M}}_{flow}(\phi), \pi)
\end{aligned}
$$

$$
\mathcal{T}^{\mathcal{C}}_{flow}(\phi) = \phi' \text{ where}
$$
$$
\phi' = \begin{cases} \{(j, \mathcal{T}^{\mathcal{C}}_{env}(B)) \mid (j, B) \in \phi\} & \text{if } \tau \notin \phi \\ \\ \{(j, \mathcal{T}^{\mathcal{C}}_{env}(B)) \mid (j, B) \in \phi\} \cup \Theta & \text{if } \tau \in \phi \end{cases}
$$

$$
\mathcal{T}^{\mathcal{M}}_{flow}(\phi) = \phi' \text{ where}
$$
$$
\phi' = \begin{cases} \{(h.j, \mathcal{T}^{\mathcal{M}}_{env}(B)) \mid (j, B) \in \phi\} & \text{if } \tau \notin \phi \\ \\ \{(h.j, \mathcal{T}^{\mathcal{M}}_{env}(B)) \mid (j, B) \in \phi\} \cup \Theta & \text{if } \tau \in \phi \end{cases}
$$

where h is the index of the context hole, and

$$
\begin{aligned}
\Theta = \ & \textstyle\bigcup_{i \in Dom(\Gamma_{\mathcal{M}})} \{\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi'')) \mid TApp(i) \wedge \\
&\qquad\qquad\qquad \Gamma_{\mathcal{M}}(i.rand) = (A, (\phi'', \pi))\} \cup \\
& \textstyle\bigcup_{i \in Dom(\Gamma_{\mathcal{C}})} \{\mathcal{T}^{\mathcal{C}}_{flow}(\mathbf{FClos}(\phi'')) \mid TApp(i) \wedge \\
&\qquad\qquad\qquad \Gamma_{\mathcal{C}}(i.rand) = (A, (\phi'', \pi))\} \cup \\
& \mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})), \text{ where } \Gamma_{\mathcal{M}}(\epsilon) = (\phi^{\mathcal{M}}_{\epsilon}, \pi^{\mathcal{M}}_{\epsilon})
\end{aligned}
$$

Figure 7: Merging the context and term annotations

1. $(\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi), \pi) \leq (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi'), \pi')$*, and*

2. $(\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi), \pi) \leq (\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi'), \pi')$

Proof. By the ordering on value propositions and Lemma 2. ∎

Our main result is:

Theorem 3 (Merged Consistency) *Let $\Gamma_{\mathcal{C}}$ be a monovariant annotation map for a context $C[\,]$, which is locally consistent with respect to the annotations given in Figures 4 and 6 and which obeys the hole constraint, and let $\Gamma_{\mathcal{M}}$ be a monovariant annotation map for a closed term M, which is locally consistent with respect to the annotations given in Figure 4 and 6, and which also obeys the minimality, import and export constraints. Let Γ be an annotation map for the substituted context, constructed according to the translations in Figure 7. Then Γ is a monovariant annotation map for $C[M]$ which is locally consistent with respect to the constraints in Figure 4.*

Proof. By the local consistency of Γ for each occurrence i in $C[M]$. ∎

In the proof, we have to consider not only the syntactic category of each occurrence, but also whether an occurrence and its subterms come from the context or from the term substitution. Therefore, there are many cases. Fortunately, many cases are symmetric with respect to the context and term substitution. A full proof appears in Appendix A.

Theorem 4 (Separate transformation) *Let $\Gamma_{\mathcal{C}}$ be a monovariant annotation map for a context $C[\,]$ with index i, which is locally consistent with respect to the annotations given in Figures 4 and 6, and which obeys the hole constraint, and let $\Gamma_{\mathcal{M}}$ be a monovariant annotation map for a closed term M with index ϵ, which is locally consistent with respect to the annotations given in Figures 4 and 6, and which also obeys the minimality, import and export constraints. Let Γ be an annotation map for $C[M]$, constructed according to the translations in Figure 7. Now let $\Phi_{\Gamma_{\mathcal{C}}}(i)$ be the output context $C'[\,]$. Then $C'[\Phi_{\Gamma_{\mathcal{M}}}(\epsilon)] = \Phi_{\Gamma}(i)$, where i is considered the index of $C[M]$.*

Proof. By the definition of Φ, and the fact that the translation in Figure 7 does not change the tags associated with particular occurrences. ∎

The significance of Theorem 4 is that we can annotate and transform a context and term substitution separately. Indeed, if we are not concerned with local optimization, we can annotate all occurrences in either the context or the term substitution with a tag of *cl*.

13 Related work

The idea of using a token to represent foreign procedures was used by Shivers [4, p. 37–38] for control flow analysis only (no transformation). Shivers considered only a single subprogram, not context substitution. He describes a set of rules for determining when procedures escape. Our set Θ in Figure 7, representing procedures which may escape from a context or a term substitution, is built using similar notions. We have shown not only that the analysis is sound, but also that the resulting transformation is correct.

Appel suggests how the SML/NJ compiler enforces invariants about functions that cross module boundaries [1, Chapter 2].

Consel and Jouvelot performed a separate binding-time analysis using a type and effect system [2]. Their system generates a set of constraints on binding time annotations, which can always be solved.

Tang and Jouvelot devised methods for separate control-flow analysis, first using a type-and-effect system [8], later using an abstract interpretation following Shivers' approach [9]. Both of these analyses depend on type information; such information is not available to our analysis.

Monovariant annotations were observed to correspond to regular trees in [5]. There, we used a translation in the style of Figure 7 to map a solution from one global analysis for closure conversion to another such analysis [5, Chapter 4]; that translation also specified a system of regular equations.

14 Conclusions

We have shown how to perform a separate analysis and transformation for untyped programs, by extending our method for global analysis and transformation. This method should extend to many different program transformations for untyped languages. As Shivers has noted, when there is no information about procedures that cross module boundaries, we have to make worst-case assumptions. Because the resulting analysis provides relatively weak information, we may miss chances for local optimizations. If better information is available, one could obtain a more precise analysis. For example, if we knew that all procedures crossing module boundaries were first-order, we could limit the propagation of procedure tokens, so that closures could be avoided in more cases. One might include such an assertion in programs as a hint to the optimizer.

Acknowledgements

Thanks to Mitch Wand for suggesting a problem which led to the ideas in this paper.

References

[1] Andrew W. Appel. *Compiling with Continuations.* Cambridge University Press, Cambridge, 1992.

[2] Charles Consel and Pierre Jouvelot. Separate polyvariant binding-time analysis. Technical Report CS/E 93-006, Oregon Graduate Institute, 1993.

[3] Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25:96–169, 1983.

[4] Olin Shivers. *Control-Flow Analysis of Higher-Order Languages.* PhD thesis, Carnegie-Mellon University, May 1991.

[5] Paul Steckler. *Correct Higher-Order Program Transformations.* PhD thesis, College of Computer Science, Northeastern University, Boston, Massachusetts, July 1994. Available as Technical Report NU-CCS-94-15.

[6] Paul Steckler and Mitchell Wand. Selective thunkification. In Baudouin Le Charlier, editor, *Proceedings of the 1st International Static Analysis Symposium*, volume 864 of *Lecture Notes in Computer Science*, pages 162–78, Berlin, Heidelberg, and New York, 1994. Springer-Verlag.

[7] Paul Steckler and Mitchell Wand. Tracking available values for lightweight closures (summary). In Neil Jones and Carolyn Talcott, editors, *Proc. Atlantique Workshop on Semantics Based Program Manipulation*, pages 63–70, 1994. Available as DIKU Report No. 94/12, University of Copenhagen.

[8] Yan Mei Tang and Pierre Jouvelot. Control-flow aspects for escape analysis. In *Second International Workshop on Static Analysis (WSA '92)*, Bordeaux, France, September 1992.

[9] Yan Mei Tang and Pierre Jouvelot. Separate abstract interpretation for control-flow analysis. In *Theoretical Aspects of Computer Software (TACS '94)*, volume 789 of *Lecture Notes in Computer Science*, pages 224–43, Berlin, Heidelberg, and New York, 1994.

[10] Mitchell Wand and Paul Steckler. Selective and lightweight closure conversion. In *Conference Record of 21st ACM Symposium on Principles of Programming Languages*, pages 435–45, 1994.

A Proof of Theorem 3

We begin by proving three supporting lemmas. To distinguish annotations from the separate annotation maps for the context and term substitution, we use suggestive superscripts. For instance, for an i in the domain of $\Gamma_{\mathcal{C}}$, the annotation map for the context, we write $A_i^{\mathcal{C}}$ for its environment proposition, $\mathcal{P}_i^{\mathcal{C}}$ for its value proposition, $\phi_i^{\mathcal{C}}$ for its flow, and $\pi_i^{\mathcal{C}}$ for its protocol tag. For an i in the domain of $\Gamma_{\mathcal{M}}$, the comparable annotations are $A_i^{\mathcal{M}}$, $\mathcal{P}_i^{\mathcal{M}}$, $\phi_i^{\mathcal{M}}$, and $\pi_i^{\mathcal{M}}$. When we write an annotation without a superscript, such as ϕ_i, we mean that it is an annotation in the merged annotation map Γ.

To avoid repetition in the statement of the lemmas, call the following the *merger conditions*:

> Let $\Gamma_{\mathcal{C}}$ be an annotation map for a context $C[\,]$ which is locally consistent with respect to the annotations given in Figures 4 and 6, and let $\Gamma_{\mathcal{M}}$ be a monovariant annotation map for a term M which is locally consistent with respect to the annotations given in Figures 4 and 6, and which also obeys the minimality, import and export constraints for terms. Let Γ be an annotation map for the substituted context, constructed according to the translations in Figure 7.

Note that these are the same conditions we stated in Theorem 3.

Lemma 4 *Assume the merger conditions. Let $C[\,]$ be a context, other than the empty context. Suppose that $i.j$ is the index of the the hole in $C[\,]$, so that $i.j$ is also the index of the occurrence of M in $C[M]$. Then in Γ, the annotation map for the substituted context, $A_i \leq A_{i.j}$.*

Proof. By the definition of the translations in Figure 7.

ϵ is the occurrence index for M, so $A_\epsilon^{\mathcal{M}}$ is the environment proposition at ϵ in $\Gamma_{\mathcal{M}}$. By the minimality condition, $A_\epsilon^{\mathcal{M}} = \emptyset$. Therefore in Γ, $A_{i.j} = \mathcal{T}_{env}^{\mathcal{M}}(A_\epsilon^{\mathcal{M}}) = \emptyset$. In $C[\,]$, i must be the index of a term containing the context hole. By the ordering on environment propositions, for any A_i we obtain in Γ, $A_i \leq A_{i.j}$. ∎

Lemma 5 *Assume the merger conditions. If $\forall (j, B) \in \phi_i^{\mathcal{C}}$, $\tau \in \phi_{j.bv}^{\mathcal{C}}$, then in Γ, $\forall (k, C) \in \phi_i$, $\phi_{k.bv}$ contains the set Θ in Figure 7. Similarly, if $\forall (j, B) \in \phi_i^{\mathcal{M}}$, $\tau \in \phi_{j.bv}^{\mathcal{M}}$, then in Γ, $\forall (k, C) \in \phi_{h.i}$, $\phi_{k.bv}$ contains the set Θ.*

Proof. By the definitions of $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$ and $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$, and the local consistency of $\Gamma_{\mathcal{C}}$ and $\Gamma_{\mathcal{M}}$.

There are two statements in the lemma. The proof of each statement is similar. For the first part, observe that $\phi_i = \mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_i)$, and for the second part, we have $\phi_{h.i} = \mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_i)$, where h is the index of the context hole.

The main difficulty here is to show that the property holds for all abstract closures in the transformed flows. By the definition of the translations, there may be more abstract closures in $\mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_i)$ than in $\phi^{\mathcal{C}}_i$.

Suppose $\forall (j,B) \in \phi^{\mathcal{C}}_i$, $\tau \in \phi^{\mathcal{C}}_{j.bv}$. By the definition of $\mathcal{T}^{\mathcal{C}}_{flow}$, $(j,B) \in \phi^{\mathcal{C}}_i$ implies $(j, \mathcal{T}^{\mathcal{C}}_{env}(B)) \in \mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_i) = \phi_i$. Consider any such (j,B). By the local consistency of $\Gamma_{\mathcal{C}}$, $j.bv \in Dom(\Gamma_{\mathcal{C}})$, so $j.bv \in Dom(\Gamma)$, and in Γ, $\phi_{j.bv} = \mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_{j.bv})$. Since $\tau \in \phi^{\mathcal{C}}_{j.bv}$, Θ is contained in $\phi_{j.bv}$. If $\tau \notin \phi^{\mathcal{C}}_i$, there are no other abstract closures in $\mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_i)$, so we are done.

Now suppose $\tau \in \phi^{\mathcal{C}}_i$, and consider the additional subsets of $\mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_i)$, that is, the subsets of Θ in Figure 7. Our strategy will be to show that each such additional subset is the $\mathcal{T}^{\mathcal{M}}_{flow}$- or $\mathcal{T}^{\mathcal{C}}_{flow}$-translation of some flow ϕ, and that for all abstract closures $(j,B) \in \phi$, the token τ is an element of $\phi_{j.bv}$ in either $\Gamma_{\mathcal{C}}$ or $\Gamma_{\mathcal{M}}$. If τ appears in such a ϕ, there will not be any additional abstract closures to consider, because we have already supposed $\tau \in \phi^{\mathcal{C}}_i$.

One additional subset is $\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$, where $\phi^{\mathcal{M}}_{\epsilon}$ is the flow associated with the substituted term in $\Gamma_{\mathcal{M}}$. By the import constraint, $\forall (j,B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$, $\tau \in \phi^{\mathcal{M}}_{j.bv}$. Now we have to consider what abstract closures are in $\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$. By the definition of $\mathcal{T}^{\mathcal{M}}_{flow}$, $(j,B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$ implies $(h.j, \mathcal{T}^{\mathcal{M}}_{env}(B)) \in \mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$, where h is the index of the context hole. For all such (j,B), because $\tau \in \phi^{\mathcal{M}}_{j.bv}$, in Γ, $\phi_{h.j.bv}$ contains Θ.

Another additional subset is:

$$\bigcup_{i \in Dom(\Gamma_{\mathcal{M}})} \{\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi'')) \mid TApp(i) \wedge \Gamma_{\mathcal{M}}(i.rand) = (A, (\phi'', \pi))\}$$

which is the union of subsets, one for each m such that $m \in Dom(\Gamma_{\mathcal{M}})$ and $TApp(m)$. For each such m, we add $\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand}))$. Consider such an m. Since $TApp(m)$, $\tau \in \phi^{\mathcal{M}}_{m}.rator$, so by the constraints in Figure 6, $\forall (j,B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand})$, $\tau \in \phi^{\mathcal{M}}_{j.bv}$. By the definition of $\mathcal{T}^{\mathcal{M}}_{flow}$, $(j,B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand})$ implies $(h.j, \mathcal{T}^{\mathcal{M}}_{env}(B)) \in \mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand}))$. For all such (j,B), because $\tau \in \phi^{\mathcal{M}}_{j.bv}$, in Γ we have that $\phi_{h.j.bv}$ contains Θ.

A third possible subset is:

$$\bigcup_{i \in Dom(\Gamma_{\mathcal{C}})} \{\mathcal{T}^{\mathcal{C}}_{flow}(\mathbf{FClos}(\phi'')) \mid TApp(i) \wedge \Gamma_{\mathcal{C}}(i.rand) = (A, (\phi'', \pi))\}$$

We may use an argument similar to the one we just used for the last additional subset.

Now consider the other statement in the lemma. Suppose $\forall (j, B) \in \mathbf{FClos}(\phi_i^{\mathcal{M}})$, $\tau \in \phi_{j.bv}^{\mathcal{M}}$. By the definition of $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$, $(j, B) \in \mathbf{FClos}(\phi_i^{\mathcal{M}})$ implies $(h.j, \mathcal{T}_{\mathit{env}}^{\mathcal{C}}(B)) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi)$, where h is the index of the hole in the context. Consider any such (j, B). By the local consistency of $\Gamma_{\mathcal{M}}$, $j.\mathit{bv} \in \mathit{Dom}(\Gamma_{\mathcal{M}})$, so $h.j.\mathit{bv} \in \mathit{Dom}(\Gamma)$. The flow for $h.j.bv$ in Γ is $\phi_{h.j.bv} = \mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_{j.bv})$. Since $\tau \in \phi_{j.bv}^{\mathcal{M}}$, Θ is contained in $\phi_{j.bv}$. If $\tau \notin \mathbf{FClos}(\phi_i^{\mathcal{M}})$, we are done.

If $\tau \in \mathbf{FClos}(\phi_i^{\mathcal{M}})$, use an argument similar to the one we used for the first part of the lemma for the additional subsets. ∎

Since a flow closure is the union of flows from an annotation map, we have:

Corollary 1 *Assume the merger conditions. If $\forall (j, B) \in \mathbf{FClos}(\phi_i^{\mathcal{C}})$, $\tau \in \phi_{j.bv}^{\mathcal{C}}$, then in Γ, $\forall (k, C) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_i^{\mathcal{C}}))$, $\phi_{k.bv}$ contains the set Θ in Figure 7. Similarly, if $\forall (j, B) \in \mathbf{FClos}(\phi_i^{\mathcal{M}})$, $\tau \in \phi_{j.bv}^{\mathcal{M}}$, then in Γ, $\forall (k, C) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\mathbf{FClos}(\phi_i^{\mathcal{M}}))$, $\phi_{k.bv}$ contains Θ.*

Lemma 6 *Assume the merger conditions. If $\forall (j, B) \in \phi_i^{\mathcal{C}}$, $\pi_{j.bv}^{\mathcal{C}} = cl$, then in Γ, $\forall (k, C) \in \phi_i$, $\pi_{k.bv} = cl$. Similarly, if $\forall (j, B) \in \phi_i^{\mathcal{M}}$, $\pi_{j.bv}^{\mathcal{M}} = cl$, then in Γ, $\forall (k, C) \in \phi_{h.i}$, $\pi_{k.bv} = cl$.*

Proof. By the definitions of $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$ and $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$, and the local consistency of $\Gamma_{\mathcal{C}}$ and $\Gamma_{\mathcal{M}}$.

As in the previous lemma, there are two parts here. For purposes of the first part, note that $\phi_i = \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_i^{\mathcal{C}})$, and for the second part, we have $\phi_{h.i} = \mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_i^{\mathcal{M}})$.

Consider the first part of the lemma. By the definition of $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$, $(j, B) \in \phi_i^{\mathcal{C}}$ implies $(j, \mathcal{T}_{\mathit{env}}^{\mathcal{C}}(B)) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_i^{\mathcal{C}})$. By assumption, $(j, B) \in \phi_i^{\mathcal{C}}$ implies $\pi_{j.bv}^{\mathcal{C}} = cl$. By the definitions of the translations in Figure 7, the tag associated with an occurrence in $\Gamma_{\mathcal{C}}$ is the same tag associated with that occurrence in Γ. Therefore, the required condition on tags holds for those abstract closures in ϕ_i directly derived from abstract closures in $\phi_i^{\mathcal{C}}$. If $\tau \notin \phi_i^{\mathcal{C}}$, we are done.

If $\tau \in \phi_i^{\mathcal{C}}$, we need to consider the other abstract closures that are in ϕ_i. In that case, by the definitions of $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$ and $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$, $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_i^{\mathcal{C}})$ also contains the sets making up the set Θ in Figure 7. Therefore, the other abstract closures are elements of $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$- and $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$-transforms of flow closures from either $\Gamma_{\mathcal{C}}$ or $\Gamma_{\mathcal{M}}$. Note that the only additional abstract closures in ϕ_i we need to consider are those derived directly from such flow closures, since we already assume $\tau \in \phi_i^{\mathcal{C}}$.

The first subset of Θ is $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\mathbf{FClos}(\phi_\epsilon^{\mathcal{M}}))$. By the import constraint for term substitutions, $\forall (j, B) \in \mathbf{FClos}(\phi_\epsilon^{\mathcal{M}})$, $\pi_{j.bv}^{\mathcal{M}} = cl$. Note that occurrences have the same tags in Γ as in $\Gamma_{\mathcal{M}}$, so for all abstract closures in $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\mathbf{FClos}(\phi_\epsilon^{\mathcal{M}}))$ derived directly from those in $\mathbf{FClos}(\phi_\epsilon^{\mathcal{M}})$, the result holds.

Consider the subset of Θ:

$$\bigcup_{i \in Dom(\Gamma_{\mathcal{M}})} \{\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi'')) \mid \mathit{TApp}(i) \wedge \Gamma_{\mathcal{M}}(i.\mathit{rand}) = (A, (\phi'', \pi))\}$$

For each m such that $m \in Dom(\Gamma_{\mathcal{M}})$ and $\mathit{TApp}(m)$, we include $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{m.\mathit{rand}}))$ in this subset. For such an m, since $\mathit{TApp}(m)$, $\tau \in \phi^{\mathcal{M}}_{m}.\mathit{rator}$, so by the constraints in Figure 6, $\forall (j, B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{m.\mathit{rand}})$, $\pi^{\mathcal{M}}_{j.bv} = \mathit{cl}$. Again we note that the translations do not change the tags associated with occurrences.

A similar argument holds for the subset of Θ:

$$\bigcup_{i \in Dom(\Gamma_{\mathcal{C}})} \{\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\mathbf{FClos}(\phi'')) \mid \mathit{TApp}(i) \wedge \Gamma_{\mathcal{C}}(i.\mathit{rand}) = (A, (\phi'', \pi))\}$$

The second part of the lemma follows from similar reasoning about $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{i})$. ∎

Again, we observe that flow closures are the union of flows from an annotation map, giving:

Corollary 2 *Assume the merger conditions. If* $\forall (j, B) \in \mathbf{FClos}(\phi^{\mathcal{C}}_{i})$, $\pi^{\mathcal{C}}_{j.bv} = \mathit{cl}$*, then in* Γ, $\forall (k, C) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{C}}_{i}))$, $\pi_{k.bv} = \mathit{cl}$*. Similarly, if* $\forall (j, B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{i})$, $\pi^{\mathcal{M}}_{j.bv} = \mathit{cl}$*, then in* Γ, $\forall (k, C) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{i}))$, $\pi_{k.bv} = \mathit{cl}$.

In a substituted context $C[M]$, we may identify an occurrence i as originating in either the context $C[\,]$ or in the substituted term M. We write $\mathcal{C} \rightsquigarrow i$ or $\mathcal{M} \rightsquigarrow i$ to indicate that i originates in the context or substituted term, respectively. We can extend this notation and write, for example, $\mathcal{C} \rightsquigarrow i, j$ if both i and j originate in the context.

If $\mathcal{C} \rightsquigarrow i$, then the same occurrence has index i in the context. If $\mathcal{M} \rightsquigarrow i$, then i must be of the form $h.j$, where h is the index of the hole in the context, and j is the index of the occurrence in M. In the proof below, we will always use h as the index of the context hole.

We can now give the proof of Theorem 3, p. 19, by showing the local consistency of the merged annotation map Γ for every occurrence in its domain.

case $\mathit{Var}(i)$

The only constraint is that $A_i([\![i]\!]) = \mathcal{P}_i$.

Suppose $\mathcal{C} \rightsquigarrow i$. By the local consistency of $\Gamma_{\mathcal{C}}$, $A^{\mathcal{C}}_{i}([\![i]\!]) = \mathcal{P}^{\mathcal{C}}_{i}$. Since $i \in Dom(\Gamma_{\mathcal{C}})$, by the merging translations in Figure 7, $A_i = \mathcal{T}^{\mathcal{C}}_{\mathit{env}}(A^{\mathcal{C}}_{i})$. Since $[\![i]\!] \in Dom(A^{\mathcal{C}}_{i})$, we have $[\![i]\!] \in Dom(A_i)$, and $A_i([\![i]\!]) = (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i}), \pi^{\mathcal{C}}_{i})$, where $(\phi^{\mathcal{C}}_{i}, \pi^{\mathcal{C}}_{i}) = A^{\mathcal{C}}_{i}([\![i]\!])$. But by the translations in Figure 7, the value proposition for i in Γ is also $(\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i}), \pi^{\mathcal{C}}_{i})$.

Suppose instead $\mathcal{M} \leadsto i$, and let $i = h.j$, where j is the index of the same occurrence in the term. By the local consistency of $\Gamma_{\mathcal{M}}$, $A_j^{\mathcal{M}}([\![j]\!]) = \mathcal{P}_j^{\mathcal{M}}$. Since $j \in Dom(\Gamma_{\mathcal{M}})$, by the merging translations, $A_i = A_{h.j} = \mathcal{T}_{env}^{\mathcal{M}}(A_j^{\mathcal{M}})$. Now, $[\![j]\!] \in Dom(A_j^{\mathcal{M}})$, and $[\![j]\!]_M = [\![i]\!]_{C[]}$, so $[\![i]\!] \in Dom(A_i)$, and $A_i([\![i]\!]) = (\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_j^{\mathcal{M}}), \pi_j^{\mathcal{M}})$. By the merging translations, the value proposition for $h.j = i$ in Γ is also $(\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_j^{\mathcal{M}}), \pi_j^{\mathcal{M}})$.

case $Abs(i)$

First we show that $A_i[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}] \leq A_{i.body}$.

Suppose that $\mathcal{C} \leadsto i, i.body$. Since $\Gamma_{\mathcal{C}}$ is locally consistent, $A_i^{\mathcal{C}}[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}^{\mathcal{C}}] \leq A_{i.body}^{\mathcal{C}}$, so by Lemma 2, $\mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}}[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}^{\mathcal{C}}]) \leq \mathcal{T}_{env}^{\mathcal{C}}(A_{i.body}^{\mathcal{C}})$. By the definition of the translations, $\mathcal{T}_{env}^{\mathcal{C}}(A_{i.body}^{\mathcal{C}}) = A_{i.body}$, and

$$\begin{aligned}
& \mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}}[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}^{\mathcal{C}}]) \\
= \;& \mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}}[[\![i.bv]\!] \mapsto (\phi_{i.bv}^{\mathcal{C}}, \pi_{i.bv}^{\mathcal{C}})]) \\
= \;& \mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}})[[\![i.bv]\!] \mapsto (\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{i.bv}^{\mathcal{C}}), \pi_{i.bv}^{\mathcal{C}})] \\
= \;& A_i[[\![i.bv]\!] \mapsto (\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{i.bv}^{\mathcal{C}}), \pi_{i.bv}^{\mathcal{C}})] \\
= \;& A_i[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}]
\end{aligned}$$

Suppose that $\mathcal{M} \leadsto i, j.body$. Since $\Gamma_{\mathcal{M}}$ is locally-consistent, $A_i[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}] \leq A_{i.body}$, and again by Lemma 2, $\mathcal{T}_{env}^{\mathcal{M}}(A_i[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}]) \leq \mathcal{T}_{env}^{\mathcal{M}}(A_{i.body})$. We have $\mathcal{T}_{env}^{\mathcal{C}}(A_{i.body}^{\mathcal{M}}) = A_{i.body}$, and

$$\begin{aligned}
& \mathcal{T}_{env}^{\mathcal{M}}(A_i^{\mathcal{M}}[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}^{\mathcal{M}}]) \\
= \;& \mathcal{T}_{env}^{\mathcal{M}}(A_i^{\mathcal{M}}[[\![i.bv]\!] \mapsto (\phi_{i.bv}^{\mathcal{M}}, \pi_{i.bv}^{\mathcal{M}})]) \\
= \;& \mathcal{T}_{env}^{\mathcal{M}}(A_i^{\mathcal{M}})[[\![i.bv]\!] \mapsto (\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_{i.bv}^{\mathcal{M}}), \pi_{i.bv}^{\mathcal{M}})] \\
= \;& A_i[[\![i.bv]\!] \mapsto (\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_{i.bv}^{\mathcal{M}}), \pi_{i.bv}^{\mathcal{M}})] \\
= \;& A_i[[\![i.bv]\!] \mapsto \mathcal{P}_{i.bv}]
\end{aligned}$$

The other possibility is that $\mathcal{C} \leadsto i$ but $\mathcal{M} \leadsto i.body$. In this case, the context hole has occurrence index $i.body$, so the result holds by Lemma 4.

Next, we show that $\{(i, A_i)\} \subseteq \phi_i$.

Suppose $\mathcal{C} \leadsto i$. By the local consistency of $\Gamma_{\mathcal{C}}$, $\{(i, A_i^{\mathcal{C}})\} \subseteq \phi_i^{\mathcal{C}}$. Now, $\Gamma(i) = (A_i, (\phi_i, \pi_i))$, and by the merging translation, $(\phi_i, \pi_i) = (\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_i^{\mathcal{C}}), \pi_i^{\mathcal{C}})$. Observe that by the definition of $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$, if $(i, A_i^{\mathcal{C}}) \in \phi_i^{\mathcal{C}}$ then $(i, \mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}})) = (i, A_i) \in \phi_i$, so the constraint is satisfied.

Suppose $\mathcal{M} \leadsto i$, and that j is the index of the same occurrence in the term substitution, so $i = h.j$. By the local consistency of $\Gamma_{\mathcal{M}}$, $\{(j, A_j^{\mathcal{M}})\} \subseteq \phi_j^{\mathcal{M}}$. We have $\Gamma(i) = (A_i, (\phi_i, \pi_i))$, and $(\phi_i, \pi_i) = (\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_j^{\mathcal{M}}), \pi_j^{\mathcal{M}})$. By the definition of $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$, if $(j, A_j^{\mathcal{M}}) \in \phi_j^{\mathcal{M}}$, then $(h.j, \mathcal{T}_{env}^{\mathcal{M}}(A_j^{\mathcal{M}})) = (i, A_i) \in \phi_i$.

case $PrimApp(i)$

We have only one constraint, that $A_i \leq A_{i.rand}$.

If $\mathcal{C} \rightsquigarrow i, i.rand$, then by the local consistency of $\Gamma_{\mathcal{C}}$, $A_i^{\mathcal{C}} \leq A_{i.rand}^{\mathcal{C}}$. By Lemma 2, $\mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}}) \leq \mathcal{T}_{env}^{\mathcal{C}}(A_{i.rand}^{\mathcal{C}})$, that is, $A_i \leq A_{i.rand}$.

Similarly, suppose $\mathcal{M} \rightsquigarrow i, i.rand$. The occurrences i and $i.rand$ in the substituted context correspond to occurrences j and $j.rand$ in the term substitution. Therefore, $i = h.j$ and $i.rand = h.j.rand$. By the local consistency of $\Gamma_{\mathcal{M}}$, $A_j^{\mathcal{M}} \leq A_{j.rand}^{\mathcal{M}}$. By Lemma 2, $\mathcal{T}_{env}^{\mathcal{M}}(A_j^{\mathcal{C}}) \leq \mathcal{T}_{env}^{\mathcal{M}}(A_{j.rand}^{\mathcal{M}})$, so $A_i \leq A_{i.rand}$.

If $\mathcal{C} \rightsquigarrow i$ and $\mathcal{M} \rightsquigarrow i.rand$, $i.rand$ is the index of the context hole, and the result holds by Lemma 4.

case $App(i)$

There are several constraints that we wish to show are satisfied. Because there are so many constraints, and for each constraint so many subcases, we provide extensive guideposts in the text for the reader's benefit.

Constraint: $A_i \leq A_{i.rator}, A_{i.rand}$.

Suppose $\mathcal{C} \rightsquigarrow i, i.rator$. By the local consistency of $\Gamma_{\mathcal{C}}$, $A_i^{\mathcal{C}} \leq A_{i.rator}^{\mathcal{C}}$, so by Lemma 2, $\mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}}) \leq \mathcal{T}_{env}^{\mathcal{C}}(A_{i.rator}^{\mathcal{C}})$. Therefore, $A_i \leq A_{i.rator}$. A similar argument holds for A_i and $A_{i.rand}$ if $\mathcal{C} \rightsquigarrow i, i.rand$.

Suppose $\mathcal{M} \rightsquigarrow i, i.rator$. The occurrences i and $i.rator$ in the substituted context correspond to occurrences j and $j.rator$ in the term substitution. Therefore, $i = h.j$ and $i.rator = h.j.rator$. By the local consistency of $\Gamma_{\mathcal{M}}$, $A_j^{\mathcal{M}} \leq A_{j.rator}^{\mathcal{M}}$, so by Lemma 2, $\mathcal{T}_{env}^{\mathcal{C}}(A_j^{\mathcal{M}}) \leq \mathcal{T}_{env}^{\mathcal{C}}(A_{j.rator}^{\mathcal{M}})$. By the definition of the translations, $A_i \leq A_{i.rator}$. Use a similar argument for A_i and $A_{i.rand}$ if $\mathcal{M} \rightsquigarrow i, i.rand$.

Now suppose $\mathcal{C} \rightsquigarrow i$, but $\mathcal{M} \rightsquigarrow i.rator$ (note that the converse is not possible). In this case, $i.rator = h$, and the result holds by Lemma 4. Use a similar argument if $\mathcal{C} \rightsquigarrow i$ and $\mathcal{M} \rightsquigarrow i.rand$.

Constraint: $\forall (j, B) \in \phi_{i.rator},\ Abs(j)$.

If $\mathcal{C} \rightsquigarrow i.rator$, then $\phi_{i.rator} = \mathcal{T}_{flow}^{\mathcal{C}}(\phi_{i.rator}^{\mathcal{C}})$. Similarly, if $\mathcal{M} \rightsquigarrow i.rator$, then $\phi_{i.rator} = \mathcal{T}_{flow}^{\mathcal{M}}(\phi_{i.rator}^{\mathcal{M}})$. By the definitions of flow and abstract closure, for all abstract closures (k, C) in all flows in both annotation maps, $Abs(k)$. Observe that $\mathcal{T}_{flow}^{\mathcal{C}}$ and $\mathcal{T}_{flow}^{\mathcal{M}}$ are mutually recursive. Consider an abstract closure (j, B) in a flow that is the result of one those translations. By the definitions of the translations, either (1), j appears in an abstract closure in some flow in $\Gamma_{\mathcal{C}}$, or (2), $j = h.k$, for some k that appears in a flow in $\Gamma_{\mathcal{M}}$. In the case of (1), j is an index for an occurrence in both the context and the substituted context. In the case of (2), the occurrence k in the term has index $h.k$ in the substituted context. In either case, $Abs(j)$.

Constraint: $\forall (j,B) \in \phi_{i.rator}$, $\mathcal{P}_{i.rand} \leq \mathcal{P}_{j.bv}$.

We want to show that $\forall (j,B) \in \phi_{i.rator}$, $\phi_{i.rand} \leq \phi_{j.bv}$ and $\pi_{i.rand} = \pi_{j.bv}$.

Suppose $\mathcal{C} \leadsto i, i.rator, i.rand$. Then by the local consistency of $\Gamma_{\mathcal{C}}$, $\forall (j,B) \in \phi^{\mathcal{C}}_{i.rator}$, $\mathcal{P}^{\mathcal{C}}_{i.rand} \leq \mathcal{P}^{\mathcal{C}}_{j.bv}$, or equivalently, $\forall (j,B) \in \phi^{\mathcal{C}}_{i.rator}$, $(\phi^{\mathcal{C}}_{i.rand}, \pi^{\mathcal{C}}_{i.rand}) \leq (\phi^{\mathcal{C}}_{j.bv}, \pi^{\mathcal{C}}_{j.bv})$.

The abstract closures in $\phi_{i.rator}$ may represent procedures from either the context or the term. Consider first those abstract closures in $\phi_{i.rator}$ that have their source in $\phi^{\mathcal{C}}_{i.rator}$, and so represent procedures in the context. By the definition of $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}$, if $(j,B) \in \phi^{\mathcal{C}}_{i.rator}$, then $(j, \mathcal{T}^{\mathcal{C}}_{env}(B)) \in \phi_{i.rator}$. For all such (j,B), $j.bv \in Dom(\Gamma_{\mathcal{C}})$, so $\mathcal{P}_{j.bv} = (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{j.bv}), \pi^{\mathcal{C}}_{j.bv})$. Therefore, for all such (j,B),

$$\begin{aligned} \mathcal{P}_{i.rand} &= (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}), \pi^{\mathcal{C}}_{i.rand}) \\ &\leq (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{j.bv}), \pi^{\mathcal{C}}_{j.bv}) \\ &= \mathcal{P}_{j.bv} \end{aligned}$$

where the ordering holds by Lemma 3.

The other abstract closures (j,B) in $\phi_{i.rator}$ represent procedures from either the context or the term; they are present because $\tau \in \phi^{\mathcal{C}}_{i.rator}$. Therefore, the additional abstract closures in $\phi_{i.rator}$ are exactly those in the set Θ in Figure 7. We will show that:

1. $\forall (j,B) \in \Theta$, $\Theta \subseteq \phi_{j.bv}$,
2. $\forall (j,B) \in \Theta$, $\pi_{i.rand} = \pi_{j.bv}$, and
3. $\phi_{i.rand} \subseteq \Theta$, hence by (1), $\forall (j,B) \in \Theta$, $\phi_{i.rand} \subseteq \phi_{j.bv}$.

We show (1) and (2) together. Consider each of the specified subsets of Θ. The first subset is:

$$\bigcup_{i \in Dom(\Gamma_{\mathcal{M}})} \{\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi'')) \mid \mathit{TApp}(i) \wedge \Gamma_{\mathcal{M}}(i.rand) = (A, (\phi'', \pi))\} \cup$$

This subset is the union of other subsets, one for each m such that $m \in Dom(\Gamma_{\mathcal{M}})$ and $\mathit{TApp}(m)$. For each such m, we have the subset $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand}))$. Consider such an m. Since $\mathit{TApp}(m)$, $\tau \in \phi^{\mathcal{M}}_{m}.{}_{rator}$, so by the constraints in Figure 6, $\forall (j,B) \in \mathbf{FClos}_{\Gamma_{\mathcal{M}}}(\phi^{\mathcal{M}}_{i.rand})$, $\tau \in \phi_{j.bv}$ and $\pi_{j.bv} = \mathit{cl}$. By Corollary 1 $\forall (j,B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand}))$, $\phi_{j.bv}$ contains the set Θ. Similarly, by Corollary 2, $\forall (j,B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{m.rand}))$, $\pi_{j.bv} = \mathit{cl}$. We can reason along these same lines to get similar results for the other two specified subsets of Θ. Since we have assumed that $\tau \in \phi_{i.rator}$, by the constraints in Figure 6, $\pi_{i.rand} = \mathit{cl}$, giving the desired result.

Now we show (3). By the definition of flow closure, $\phi^{\mathcal{C}}_{i.rand} \subseteq \mathbf{FClos}(\phi^{\mathcal{C}}_{i.rand})$. Therefore, by Lemma 2, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}) \subseteq \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{C}}_{i.rand}))$. Since $\tau \in \phi^{\mathcal{C}}_{i.rand}$, and the operand is not the context hole, we have $\mathit{TApp}(i)$. Therefore, by the definition of Θ in Figure 7, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{C}}_{i.rand}))$ is contained in Θ, hence $\phi_{i.rand} = \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}) \subseteq \Theta$.

Suppose $\mathcal{M} \rightsquigarrow i, i.rator, i.rand$. This subcase is essentially identical to the last, observing the symmetries between the context and the term substitution.

Now suppose $\mathcal{C} \rightsquigarrow i, i.rator$ but $\mathcal{M} \rightsquigarrow i.rand$. Then $i.rand$ is the index of the hole in the context, and in the substituted context, it is the index of the substituted term. Considered alone, the term substitution has index ϵ.

Now, $\phi_{i.rator} = \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$ and $\phi_{i.rand} = \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$. We want to show that $\forall (j,B) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$, $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}) \subseteq \phi_{j.bv}$ and $\pi^{\mathcal{M}}_{\epsilon} = \pi_{j.bv}$.

Our strategy will be to show:

1. $\forall (j,B) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$, $\phi_{j.bv}$ contains the set Θ,
2. $\forall (j,B) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$, $\pi_{j.bv} = \pi_{i.rand}$, and
3. $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$ is contained in Θ, so $\forall (j,B) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$, $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}) \subseteq \phi_{j.bv}$.

We first show (1). By the local consistency of $\Gamma_{\mathcal{C}}$, since $\mathit{Hole}(i.\mathit{rand})$, $\tau \in \phi^{\mathcal{C}}_{i.rand}$. Therefore, by the local consistency of $\Gamma_{\mathcal{C}}$, $\forall (j,B) \in \phi^{\mathcal{C}}_{i.rator}$, $\tau \in \phi^{\mathcal{C}}_{j.bv}$. By Lemma 5, we conclude that $\forall (j,B) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$, $\phi_{j.bv}$ contains the set Θ.

Now we show (2). By the export constraints, $\pi_{\epsilon} = cl$, and by the definition of the translations, $\pi_{\epsilon} = \pi_{i.rand}$. Since $\mathit{Hole}(i.\mathit{rand})$, $\pi^{\mathcal{C}}_{i.rand} = cl$, so by the local consistency of $\Gamma_{\mathcal{C}}$, $\forall (j,B) \in \phi^{\mathcal{C}}_{i.rator}$, $\pi^{\mathcal{C}}_{j.bv} = cl$. Therefore, by Lemma 6, we have $\forall (j,B) \in \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rator})$, $\pi_{j.bv} = cl$.

To show (3): Observe that $\phi^{\mathcal{M}}_{\epsilon} \subseteq \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$. By Lemma 2, $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}) \subseteq \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$. Since $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$ is contained in Θ, so is $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$.

The last possibility we need to consider for this constraint is where $\mathcal{C} \rightsquigarrow i, i.rand$ but $\mathcal{M} \rightsquigarrow i.rator$. If this holds, then $i.rator$ is the index of the context hole, and in the substituted context, it is the index of the substituted term. We have $\phi_{i.rator} = \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$ and $\phi_{i.rand} = \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand})$. We wish to show that $\forall (j,B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}) \subseteq \phi_{j.bv}$ and $\pi^{\mathcal{C}}_{i.rand} = \pi_{j.bv}$.

Our plan will be familiar by now. We will show that:

1. $\forall (j,B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\phi_{j.bv}$ contains the set Θ,
2. $\forall (j,B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\pi_{j.bv} = \pi_{i.rand}$, and

3. $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand})$ is contained in Θ, hence $\forall (j, B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}) \subseteq \phi_{j.bv}$.

By the import constraints for term substitutions, if $(j, B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$, then $\tau \in \phi^{\mathcal{M}}_{j.bv}$. Since $\phi^{\mathcal{M}}_{\epsilon} \subseteq \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$, if $(j, B) \in \phi^{\mathcal{M}}_{\epsilon}$, then $\tau \in \phi^{\mathcal{M}}_{j.bv}$. By Lemma 5, $\forall (j, B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\phi_{j.bv}$ contains Θ. This is our goal (1) from above.

Since $\mathit{Hole}(i.rator)$, by the hole constraint, $\tau \in \phi^{\mathcal{C}}_{i.rator}$. By the local conditions in Figure 6, we have $\pi_{i.rand} = \mathit{cl}$. By the import constraints, if $\forall (j, B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$, $\pi^{\mathcal{M}}_{j.bv} = \mathit{cl}$. Therefore, $\forall (j, B) \in \phi^{\mathcal{M}}_{\epsilon}$, $\pi^{\mathcal{M}}_{j.bv} = \mathit{cl}$. By Lemma 6, $\forall (j, B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\pi^{\mathcal{M}}_{j.bv} = \mathit{cl}$. This is goal (2).

We have that $\mathit{App}(i)$, $\tau \in \phi^{\mathcal{C}}_{i.rator}$, and $i.rand$ is not the index of the context hole. Therefore, $\mathit{TApp}(i)$, so $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{C}}_{i.rand}))$ is contained in Θ. Now $\phi^{\mathcal{C}}_{i.rand} \subseteq \mathbf{FClos}(\phi^{\mathcal{C}}_{i.rand})$, so by Lemma 2, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}) \subseteq \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{C}}_{i.rand}))$. Hence, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand})$ is contained in Θ, so $\forall (j, B) \in \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon})$, $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i.rand}) \subseteq \phi_{j.bv}$. This is goal (3).

Constraint: $\forall (j, B) \in \phi_{i.rator}$, $\mathcal{P}_{j.body} \leq \mathcal{P}_i$.

Suppose $\mathcal{C} \rightsquigarrow i, i.rator, i.rand$. Again, we need to consider what abstract closures appear in $\phi_{i.rator}$.

Consider those abstract closures in $\phi_{i.rator}$ that have their source in $\phi^{\mathcal{C}}_{i.rator}$, representing procedures in the context. By the local consistency of $\Gamma_{\mathcal{C}}$, $\forall (j, B) \in \phi^{\mathcal{C}}_{i.rator}$, $\mathcal{P}^{\mathcal{C}}_{j.body} \leq \mathcal{P}^{\mathcal{C}}_i$, that is, $\phi^{\mathcal{C}}_{j.body} \subseteq \phi^{\mathcal{C}}_i$ and $\pi^{\mathcal{C}}_{j.body} = \pi^{\mathcal{C}}_i$. By the definition of $\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}$, if $(j, B) \in \phi^{\mathcal{C}}_{i.rator}$, then $(j, \mathcal{T}^{\mathcal{C}}_{\mathit{env}}(B)) \in \phi_{i.rator}$. For all such (j, B), $j.body \in \mathit{Dom}(\Gamma_{\mathcal{C}})$, so if $j.body \neq h$, then $\mathcal{P}_{j.body} = (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{j.body}), \pi^{\mathcal{C}}_{j.body})$. For all such (j, B),

$$\begin{aligned} \mathcal{P}_{j.body} &= (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{j.body}), \pi^{\mathcal{C}}_{j.body}) \\ &\leq (\mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_{i}), \pi^{\mathcal{C}}_{i}) \\ &= \mathcal{P}_i \end{aligned}$$

where the ordering holds by Lemma 3.

We also need to consider the possibility that for some (j, B), $j.body = h$, the index of the context hole. Suppose $j.body = h$. Then $\mathcal{P}_{j.body} = (\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}), \pi^{\mathcal{M}}_{\epsilon})$, so we want to show that $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}) \subseteq \phi_i$ and $\pi^{\mathcal{M}}_{\epsilon} = \pi_i$. Since $\mathit{Hole}(j.body)$, by the local consistency of $\Gamma_{\mathcal{C}}$, $\tau \in \phi^{\mathcal{C}}_{j.body}$. Therefore, by the local consistency of $\Gamma_{\mathcal{C}}$, $\tau \in \phi^{\mathcal{C}}_i$. Now, $\phi_i = \mathcal{T}^{\mathcal{C}}_{\mathit{flow}}(\phi^{\mathcal{C}}_i)$. Therefore, ϕ_i contains the set Θ in Figure 7, so $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})) \subseteq \phi_i$. Since $\phi^{\mathcal{M}}_{\epsilon} \subseteq \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$, by Lemma 2, $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}) \subseteq \mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$. Therefore, $\mathcal{T}^{\mathcal{M}}_{\mathit{flow}}(\phi^{\mathcal{M}}_{\epsilon}) \subseteq \phi_i$. Also by the local consistency of $\Gamma_{\mathcal{C}}$, since $\mathit{Hole}(i)$, $\pi^{\mathcal{C}}_{j.body} = \mathit{cl}$, so $\pi^{\mathcal{C}}_i = \mathit{cl}$. By the export constraint, $\pi^{\mathcal{M}}_{\epsilon} = \mathit{cl}$. But $\pi^{\mathcal{M}}_{\epsilon} = \pi_{j.body}$.

The other abstract closures (j, B) in $\phi_{i.rator}$ represent procedures from either the context or the term; they are present because $\tau \in \phi^{\mathcal{C}}_{i.rator}$. In that case, by the local consistency of $\Gamma_{\mathcal{C}}$, $\tau \in \phi_i^{\mathcal{C}}$. Therefore, the additional abstract closures in $\phi_{i.rator}$ are exactly those in the set Θ in Figure 7, so we wish to show that $\forall (j,B) \in \Theta$, $\phi_{j.body} \subseteq \phi_i$ and $\pi_{j.body} = \pi_i$. To do so, we will show:

1. $\forall (j,B) \in \Theta$, $\phi_{j.body} \subseteq \Theta$,

2. $\forall (j,B) \in \Theta$, $\pi_{j.body} = \pi_i$, and

3. $\Theta \subseteq \phi_i$, so by (1), $\forall (j,B) \in \Theta$, $\phi_{j.body} \subseteq \phi_i$.

To see that (1) holds, note that every subset of Θ specified in Figure 7 is the $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$- or $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$-transform of the *flow closure* of some flow ϕ in $\Gamma_{\mathcal{C}}$ or $\Gamma_{\mathcal{M}}$. That is, for such a ϕ, $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi)) \subseteq \Theta$ (or $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\mathbf{FClos}(\phi)) \subseteq \Theta$). For concreteness in our presentation, we illustrate the result by picking one such flow, say $\phi_k^{\mathcal{C}}$ from $\Gamma_{\mathcal{C}}$; the result applies to flows from either annotation map, *mutatis mutandis*.

We want to show $\forall (j,B) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$, $\phi_{j.body} \subseteq \Theta$. Suppose $(j,B) \in \mathbf{FClos}(\phi_k^{\mathcal{C}})$. Then by the definition of flow closure, $\phi_{j.body}^{\mathcal{C}} \subseteq \mathbf{FClos}(\phi_k^{\mathcal{C}})$. By Lemma 2, $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{j.body}^{\mathcal{C}}) \subseteq \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}_{\Gamma_{\mathcal{C}}}(\phi_k^{\mathcal{C}}))$. Therefore, $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{j.body}^{\mathcal{C}}) \subseteq \Theta$. So far, we have shown that $\forall (j,B) \in \mathbf{FClos}(\phi_k^{\mathcal{C}})$, $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{j.body}^{\mathcal{C}}) \subseteq \Theta$. This is almost, but not quite, what we wanted to show.

Suppose $(j,B) \in \mathbf{FClos}(\phi_k^{\mathcal{C}})$. Then $(j, \mathcal{T}_{\mathit{env}}^{\mathcal{C}}(B)) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$. For such j, we just showed that $\phi_{j.body} = \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{j.body}^{\mathcal{C}}) \subseteq \Theta$. What other abstract closures may appear in $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$? If $\tau \not\in \mathbf{FClos}(\phi_k^{\mathcal{C}})$, then there are no other abstract closures. If $\tau \in \mathbf{FClos}(\phi_k^{\mathcal{C}})$, then all abstract closures in Θ are in $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$. But our choice of $\phi_k^{\mathcal{C}}$ was arbitrary, so $\forall (j,B) \in \Theta$, $\phi_{j.body} \subseteq \Theta$.

To show (2), we use a similar argument. Since we assume that $\tau \in \phi_{i.rator}^{\mathcal{C}}$, by the local consistency of $\Gamma_{\mathcal{C}}$, $\pi_i = \mathit{cl}$. Therefore, we want to show that $\forall (j,B) \in \Theta$, $\pi_{j.body} = \mathit{cl}$. Again, we point out that every specified subset of Θ is the $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}$- or $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}$-transform of the *flow closure* of some flow ϕ in $\Gamma_{\mathcal{C}}$ or $\Gamma_{\mathcal{M}}$. Such a ϕ is associated with an operand $m.rand$ for a call site m such that $TApp(m)$, in either $\Gamma_{\mathcal{C}}$ or $\Gamma_{\mathcal{M}}$, or $\phi = \phi_\epsilon^{\mathcal{M}}$. For either possibility, by the local consistency of $\Gamma_{\mathcal{C}}$ and $\Gamma_{\mathcal{M}}$, $\forall (j,B) \in \mathbf{FClos}(\phi)$, $\pi_{j.body} = \mathit{cl}$. We arbitrarily pick one such flow, say $\phi_k^{\mathcal{C}}$ from $\Gamma_{\mathcal{C}}$; again, the result will apply to flows from either annotation map.

We want to show $\forall (j,B) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$, $\pi_{j.body} = \mathit{cl}$. If $(j,B) \in \mathbf{FClos}(\phi)$, then $(j, \mathcal{T}_{\mathit{env}}^{\mathcal{C}}(B)) \in TFlowContext(\mathbf{FClos}(\phi))$. For all abstract closures in $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$ which have their origin in $\mathbf{FClos}(\phi_k^{\mathcal{C}}$, $\pi_{j.body} = \pi_{j.body}^{\mathcal{C}} = \mathit{cl}$, since the translations act as the identity on tags. Again, we are concerned with what other abstract closures may appear in $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$. If $\tau \in \mathbf{FClos}(\phi_k^{\mathcal{C}})$, then the abstract closures in Θ are also in $\mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\mathbf{FClos}(\phi_k^{\mathcal{C}}))$. Again, the arbitrariness of our choice $\phi_k^{\mathcal{C}}$ indicates that $\forall (j,B) \in \Theta$, $\pi_{j.body} = \mathit{cl}$.

Now we show (3). Since $\tau \in \phi_i$, by the definition of the translations in Figure 7, $\Theta \subseteq \phi_i$.

Suppose $\mathcal{M} \rightsquigarrow i, i.rator, i.rand$. The result holds by reasoning similar to that used in the preceding subcase, taking into account the symmetries of the context and the term substitution.

Now suppose $\mathcal{C} \rightsquigarrow i, i.rator$ but $\mathcal{M} \rightsquigarrow i.rand$. Then $i.rand$ is the index of the context hole, and the index of the substituted term in the substituted context. We want to show that $\forall (j, B) \in \mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_{i.rator})$, $(\phi_{j.body}, \pi_{j.body}) \leq (\phi_i, \pi_i)$, Now, $(j, B) \in \phi^{\mathcal{C}}_{i.rator}$ implies $(j, \mathcal{T}^{\mathcal{C}}_{env}(B)) \in \mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_{i.rator})$. By the local consistency of $\Gamma_{\mathcal{C}}$, $\forall (j, B) \in \phi^{\mathcal{C}}_{i.rator}$, $\mathcal{P}^{\mathcal{C}}_{j.body} \leq \mathcal{P}^{\mathcal{C}}_i$. Therefore, by Lemma 3, for all such j,

$$\begin{aligned}
\mathcal{P}_{j.body} &= (\mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_{j.body}), \pi^{\mathcal{C}}_{j.body}) \\
&\leq (\mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_i), \pi^{\mathcal{C}}_i) \\
&= \mathcal{P}_i
\end{aligned}$$

Since the transformation does not change the tags associated with occurrences, also for all such j, $\pi^{\mathcal{C}}_{j.body} = \pi_{j.body}$ and $\pi^{\mathcal{C}}_i = \pi_i$. This gives the desired result for those abstract closures in $\mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_{i.rator})$ derived directly from the abstract closures $\phi^{\mathcal{C}}_{i.rator})$.

If $\tau \in \phi^{\mathcal{C}}_{i.rator}$, then $\Theta \subseteq \mathcal{T}^{\mathcal{C}}_{flow}(\phi^{\mathcal{C}}_{i.rator})$. Here we can use the same reasoning as in the subcase for $\mathcal{C} \rightsquigarrow i.rator, i.rand, i$ to show that $\forall (j, B) \in \Theta$, $\phi_{j.body} \subseteq \phi_i$ and $\pi_{j.body} = \pi_i$.

Finally, suppose $\mathcal{C} \rightsquigarrow i, i.rand$ but $\mathcal{M} \rightsquigarrow i.rator$. It must be that $i.rator$ is the index of the context hole, and also the index of the term substitution in the substituted context. Therefore, we want to show that $\forall (j, B) \in \mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{\epsilon})$, $(\phi_{j.body}, \pi_{j.body}) \leq (\phi_i, \pi_i)$. Since $Hole(i.rator)$, by the local consistency of $\Gamma_{\mathcal{C}}$, $\tau \in \phi^{\mathcal{C}}_{i.rator}$, so $\tau \in \phi^{\mathcal{C}}_i$. Therefore, by the definition of Θ, $\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})) \subseteq \phi_i$. Suppose $(j, B) \in \phi^{\mathcal{M}}_{\epsilon}$. By the definition of flow closure, $\phi^{\mathcal{M}}_{j.body} \subseteq \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$. By Lemma 2, $\mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{j.body}) \subseteq \mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$, so $\mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{j.body}) \subseteq \phi_i$. Now, since $(j, B) \in \phi^{\mathcal{M}}_{\epsilon}$, also $(h.j, \mathcal{T}^{\mathcal{M}}_{env}(B)) \in \mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{\epsilon})$. Therefore, we have that $\forall (j, B) \in \phi^{\mathcal{M}}_{\epsilon}$, $\mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{j.body}) \subseteq \phi_i$.

Again, this is not enough, because there may be abstract closures in $\mathcal{T}^{\mathcal{M}}_{flow}(\mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon}))$ not directly derived from $\phi^{\mathcal{M}}_{\epsilon}$. If $\tau \in \phi^{\mathcal{M}}_{\epsilon}$, then $\mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{\epsilon})$ also contains the set Θ in Figure 7. As we showed before for similar conditions, $\forall (j, B) \in \Theta$, $\phi_{j.body} \subseteq \phi_i$.

Now we have to deal with the tags. Since $Hole(i.rator)$, $\tau \in \phi^{\mathcal{C}}_{i.rator}$, so $\pi^{\mathcal{C}}_i = cl$. So we want to show that $\forall (j, B) \in \mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{\epsilon})$, $\pi_{j.body} = cl$. By the export constraints, $\forall (j, B) \in \mathbf{FClos}(\phi^{\mathcal{M}}_{\epsilon})$, $\pi_{j.body} = cl$. Since the translations do not change tags associated with particular occurrences, for those abstract closures (j, B) in $\mathcal{T}^{\mathcal{M}}_{flow}(\phi^{\mathcal{M}}_{\epsilon})$ derived directly from abstract closures in $\phi^{\mathcal{M}}_{\epsilon}$, $\pi_{j.body} = cl$.

More formally, we can state that if $(j, B) \in \phi_\epsilon^{\mathcal{M}}$, then $(h.j, \mathcal{T}_{env}^{\mathcal{M}}(B)) \in \mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_\epsilon^{\mathcal{M}})$, and $\pi_{h.j.body} = cl$.

If $\tau \in \phi_\epsilon^{\mathcal{M}}$, then $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_\epsilon^{\mathcal{M}})$ also contains the abstract closures in Θ. As we showed before for similar conditions, $\forall (j, B) \in \Theta$, $\pi_{j.body} = cl$.

case $Cond(i)$

First we show that $A_i \leq A_{i.test}, A_{i.then}, A_{i.else}$.

Suppose $\mathcal{C} \leadsto i, i.test$. By the local consistency of $\Gamma_{\mathcal{C}}$, $A_i^{\mathcal{C}} \leq A_{i.test}^{\mathcal{C}}$, and by Lemma 2, $\mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{C}}) \leq \mathcal{T}_{env}^{\mathcal{C}}(A_{i.test}^{\mathcal{C}})$. Therefore, $A_i \leq A_{i.test}$. Use similar reasoning for A_i and $A_{i.then}$ if $\mathcal{C} \leadsto i, i.then$, and for A_i and $A_{i.else}$ if $\mathcal{C} \leadsto i, i.else$.

Suppose $\mathcal{M} \leadsto i, i.test$. By the local consistency of $\Gamma_{\mathcal{M}}$, $A_i^{\mathcal{M}} \leq A_{i.test}^{\mathcal{M}}$, and by Lemma 2, $\mathcal{T}_{env}^{\mathcal{C}}(A_i^{\mathcal{M}}) \leq \mathcal{T}_{env}^{\mathcal{C}}(A_{i.test}^{\mathcal{M}})$, so $A_i \leq A_{i.test}$. Use similar reasoning for A_i and $A_{i.then}$ if $\mathcal{M} \leadsto i, i.then$, and for A_i and $A_{i.else}$ if $\mathcal{M} \leadsto i, i.else$.

Now suppose $\mathcal{C} \leadsto i$, but $\mathcal{M} \leadsto i.test$. The test-part of the conditional must be the substituted term itself. By the minimality condition, $A_{i.test}^{\mathcal{M}} = \emptyset$. So $\mathcal{T}_{env}^{\mathcal{M}}(A_{i.test}^{\mathcal{M}}) = A_{i.test} = \emptyset$, and the result holds by Lemma 4. Use a similar argument if $\mathcal{C} \leadsto i$ and $\mathcal{M} \leadsto i.then$, or $\mathcal{C} \leadsto i$ and $\mathcal{M} \leadsto i.else$.

We now show that $\mathcal{P}_{i.then} \leq \mathcal{P}_i$ (the reasoning to show that $\mathcal{P}_{i.else} \leq \mathcal{P}_i$ is similar).

Suppose $\mathcal{C} \leadsto i, i.then$. By the local consistency of $\Gamma_{\mathcal{C}}$, $\phi_{i.then}^{\mathcal{C}} \subseteq \phi_i^{\mathcal{C}}$, so

$$\begin{aligned} \phi_{i.then} &= \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_{i.then}^{\mathcal{C}}) \\ &\subseteq \mathcal{T}_{\mathit{flow}}^{\mathcal{C}}(\phi_i^{\mathcal{C}}) \\ &= \phi_i \end{aligned}$$

where the set inclusion holds by Lemma 2. Also by the local consistency of $\Gamma_{\mathcal{C}}$, $\pi_{i.then}^{\mathcal{C}} = \pi_i^{\mathcal{C}}$. Since the translations give occurrences in the substituted context the same tags as in the context, $\pi_{i.then} = \pi_i$. We can use similar reasoning if $\mathcal{M} \leadsto i, i.then$, allowing for the translation between occurrence indices in the term and in the substituted context.

Now suppose $\mathcal{C} \leadsto i$ but $\mathcal{M} \leadsto i.then$. It must be that $i.then$ is the index of the context hole, and so is the index of the substituted term in the substituted context. Therefore, $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_\epsilon^{\mathcal{M}}) = \phi_{i.then}$. Since $Hole(i.then)$, by the local consistency of $\Gamma_{\mathcal{C}}$, $\tau \in \phi_{i.then}^{\mathcal{C}}$, so $\tau \in \phi_i^{\mathcal{C}}$. Therefore, by the definition of Θ in Figure 7, ϕ_i contains $\mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\mathbf{FClos}(\phi_\epsilon^{\mathcal{M}}))$. Since $\phi_\epsilon^{\mathcal{M}} \subseteq \mathbf{FClos}(\phi_\epsilon^{\mathcal{M}})$,

$$\begin{aligned} \phi_{i.then} &= \mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\phi_\epsilon^{\mathcal{M}}) \\ &\subseteq \mathcal{T}_{\mathit{flow}}^{\mathcal{M}}(\mathbf{FClos}(\phi_\epsilon^{\mathcal{M}})) \\ &\subseteq \phi_i \end{aligned}$$

as desired. By the local consistency of $\Gamma_{\mathcal{C}}$, $\phi^{\mathcal{C}}_{i.then} = cl$, so $\phi^{\mathcal{C}}_{i} = cl$. Therefore, $\phi_i = cl$. By the export constraint, $\pi^{\mathcal{M}}_{\epsilon} = cl$, so $\pi_{i.then} = cl$. ∎