A Typed Lambda Calculus with Gradual Intersection Types

Pedro Ângelo
pedro.angelo@fc.up.pt
LIACC, Departamento de Ciência de Computadores,
Faculdade de Ciências, Universidade do Porto,
rua do Campo Alegre s/n, 4169 - 007
Porto, Portugal

Mário Florido
amflorid@fc.up.pt
LIACC, Departamento de Ciência de Computadores,
Faculdade de Ciências, Universidade do Porto,
rua do Campo Alegre s/n, 4169 - 007
Porto, Portugal

ABSTRACT

Intersection types have the power to type expressions which are all of many different types. Gradual types combine type checking at both compile-time and run-time. Here we combine these two approaches in a new typed calculus that harness both of their strengths. We incorporate these two contributions in a single typed calculus and define an operational semantics with type cast annotations. We also prove several crucial properties of the type system, namely that types are preserved during compilation and evaluation, and that the refined criteria for gradual typing holds.

CCS CONCEPTS

• **Theory of computation** → **Type theory**; **Lambda calculus**.

KEYWORDS

typed lambda calculus, intersection types, gradual typing

ACM Reference Format:
Pedro Ângelo and Mário Florido. 2022. A Typed Lambda Calculus with Gradual Intersection Types. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Types have been broadly used to verify program properties and reduce or, in some cases, eliminate run-time errors. Programming languages adopt either static typing or dynamic typing to prevent programs from erroneous behaviour. Static typing is useful for compile-time detection of type errors, while dynamic typing is done at run-time and enables rapid software development. Integration of static and dynamic typing has been a quite active subject of research in the last years under the name of *gradual typing* [15, 16, 23, 24, 38–40].

Intersection types, introduced by [17] and [35] in 1980, give a type theoretical characterization of strong normalization. Several other contributions followed, making intersection types a rich area of study [7, 11, 19, 21, 29, 30, 41], also used in practice in programming language design and implementation [8, 14, 20, 22, 36, 42].

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Although the type inference problem for intersection types is not decidable in general, it becomes decidable for finite rank fragments of the general system [30], e.g. rank 2 intersection types [6, 21, 25, 26].

In this paper, we present a new gradually typed calculus with rank 2 intersection types. To gradually shift type checking to runtime, one needs to annotate lambda-abstractions with the dynamic type, *Dyn*, which matches any type. Therefore, gradual type systems have an intrinsic need for explicit type annotations. Standard gradual types enable to declare every occurrence of formal function parameters as dynamically typed. Our system, using intersection types, enables some occurrences of a formal parameter to be declared as dynamically typed while others as statically typed. This gives a new fine-grained definition of dynamicity which is only possible by the use of intersection types. Thus, the main contributions of our paper are:

(1) a gradual intersection typed calculus, with rank 2 intersection types, which obeys the usual correctness criteria properties for gradual typing [40] (section 4);
(2) a compilation procedure, which inserts run-time casts into the typed code (section 5);
(3) a type safe operational semantics for the whole calculus (section 6).

Intersection types were originally designed as descriptive type assignment systems *à la Curry*, where types are assigned to untyped terms. Prescriptive versions of intersection type systems, supporting terms with type annotations in λ-abstractions, are not trivial [9, 21, 32, 36, 37, 43]. We faced similar problems in our typed calculus to add dynamic type annotations to individual occurrences of formal parameters. As an example consider the following annotated λ-expression, where we need to instantiate σ in order to make the expression well-typed: $(\lambda x : Dyn \wedge (Int \to Int) \,.\, x\, x)\ (\lambda y : \sigma \,.\, y)$. This expression can be typed with Dyn, because $\lambda x : Dyn \wedge (Int \to Int) \,.\, x\, x$ has type $Dyn \wedge (Int \to Int) \to Dyn$ and $\lambda y : \sigma \,.\, y$ may have two types: $(Int \to Int) \to Int \to Int$, with σ equal to $Int \to Int$, and $Int \to Int$, with σ equal to Int. The question now is how to choose the right type for σ. One might be tempted to use the term $\lambda y : (Int \to Int) \wedge Int \,.\, y$, however that would result in the expression being typed as either $(Int \to Int) \wedge Int \to Int \to Int$ or $(Int \to Int) \wedge Int \to Int$, both of which are incorrect. Several solutions have been presented to this problem [9, 32, 36, 37, 43]. Our type system follows the solution of [9], which makes use of parallel terms of the form $M_1 \mid \ldots \mid M_n$, where each M_i, for $i \in 1..n$, is a term with a unique type assigned to it. In the example above, the expression would now be annotated as $(\lambda x : Dyn \wedge (Int \to Int) \,.\, x\, x)\ (\lambda y : Int \to Int \,.\, y \mid \lambda z : Int \,.\, z)$, where the type of the argument is $((Int \to Int) \to Int \to Int) \wedge (Int \to Int)$.

Although originally defined in a programming language context, the logical meaning of the dynamic type is an interesting question. This is especially relevant in the context of intersection type systems, due to the apparent similarities with the ω type [18]: the ω types any program, even ill-typed ones, whereas the *Dyn* type relaxes the type system, allowing ill-typed programs to be type-checked. Our work can be viewed as a first step towards a proof-theoretical characterization of the dynamic type in the context of intersection types. Note that rank 2 intersection types have a decidable type inference problem [6, 21, 25, 26]. So, it should be possible to adapt the type inference algorithm defined in [5] to output the whole syntactic tree of annotated parallel terms, given a partially annotated lambda term as input. This would also enable the use of our calculus as an intermediate code in a gradually typed programming language, avoiding the extra effort of programmers to write several annotated copies of function arguments.

2 RELATED WORK

In [4] we made a first attempt to define a gradual intersection type system. However, this first system had not the type preservation property, due to a naive definition of type annotations with intersection types. So, our first concern was to redesign the system using an existing intersection type system with proper support for type annotations. Intersection-types à la Church [32] tackled this challenge by dividing the calculus into two. Marked-terms encode λ-calculus terms and connect to proof-terms via a variable mark. Proof-terms carry the logical information in the form of proof trees, in which are included the type annotations. Although technically sound and clean, there's a rather large overhead in carrying two distinct terms. Coupled with the indirection arising from the connection between marked and proof-terms, we find this approach too cumbersome for our specific purpose. The issue is that integration of any approach with gradual typing will mean adding a significant level of extra complexity. Branching Types [43] encode different derivations directly into types, by assigning to types a kind that keeps track of the shapes of each derivation. Although an elegant way of dealing with explicit annotations, we found later approaches to allow a more viable integration with gradual typing. Another typed language with intersection types is Forsythe [36]. We did not consider this approach because some terms in this system lack correct typings when fully annotated, e.g. there is no annotated version of $(\lambda x.(\lambda y.x))$ with type $(\tau \to \tau \to \tau) \wedge (\rho \to \rho \to \rho)$. A Typed Lambda Calculus with Intersection Types [9], introduces parallel terms, where each component is annotated, resulting in the typing of the parallel term with an intersection type. Besides allowing type annotations, parallel terms also make easier the definition of dynamic type checking of terms typed by an intersection type. Thus, due mainly to this simplicity and elegant design, we chose [9] as the basis upon which we built our system.

There is also previous work dealing with gradual typing in the presence of intersection types following a set-theoretical approach based on semantic subtyping [12, 13]. By using principles of abstract interpretation, [12] introduces a semantic definition of consistent subtyping. This work does not consider a precision relation, which precludes important properties, such as gradual guarantee [40]. Type inference was not approached in this work, but in [13] the authors refine the work of [12], also introducing a type inference algorithm. However, due to the unrestricted rank of intersection types, this algorithm is not complete. In our paper, we restrict gradual intersection types to rank-2, for which there is a complete type inference algorithm [5]. We are now working on an extension of the algorithm described in [5] to the prescriptive type system described here.

Finally, there are contributions on gradual typing with intersection types using contracts which are also related but intrinsically different from our work. In [27, 44] contracts are implemented as a library, which differs from our approach which relies on the definition of a gradual type system. Furthermore, these contributions employ intersections as a conjunction operator of contracts, whereas we define an intersection type system and a type safe calculus. More recently [34] uses intersection types in the same context, but differently from our work. The main differences are: intersections in [34] are between refinements, limiting the set of types in intersections, and we deal with general intersection types. Besides this [34] is based in a different calculus [33] using strong pairs instead of parallel terms and a non-deterministic operational semantics.

3 INTERSECTION TYPES AND SYNTAX

In the original system [17], intersections are defined as associative, commutative and idempotent. There have been several succeeding contributions that make use of non-idempotent intersections, usually to obtain quantitative information through type derivations [1, 3, 10, 28]. Here we restrict even more the algebraic properties of intersections, following the definition of [9] of a *sequence* $\tau_1 \wedge \ldots \wedge \tau_n$ as an ordered list of base types or arrow types. Therefore, intersections are non-commutative, i.e. the positions of instances cannot be swapped, e.g. $\tau \wedge \rho \neq \rho \wedge \tau$, and non-idempotent, i.e. the duplication or collapsing of instances of the same type is not allowed, e.g. $\tau \wedge \tau \neq \tau$.

Let τ and ρ (possibly with subscripts) range over *monotypes* (where the top level constructor is not the intersection type connective), and σ and υ (possibly with subscripts) range over sequences. Since we allow sequences of size one, σ and υ also range over monotypes. B ranges over base types, such as *Int* and *Bool*, and *Dyn* is the dynamic type. We define the language of types in the following grammar:

$$\begin{array}{rcll} \text{Monotypes} & \tau & ::= & B \mid Dyn \mid \sigma \to \tau \\ \text{Sequence Types} & \sigma & ::= & \tau_1 \wedge \ldots \wedge \tau_n \quad (\text{with } n \geq 1) \end{array}$$

Given a sequence $\tau_1 \wedge \ldots \wedge \tau_n$, each τ_i is called an *element* of the sequence. When we say type we refer to either monotypes or sequences. Following the original definition in [17], sequences can only appear in the left-hand side (domain) of the arrow type constructor. Therefore, the shape of a (valid) arrow type is $\tau_1 \wedge \ldots \wedge \tau_n \to \rho$, with $n \geq 1$. The intersection type connective $\wedge$ has higher precedence than the arrow type constructor $\to$, and $\to$ associates to the right. We introduce the following relation: $\tau \in \tau_1 \wedge \ldots \wedge \tau_n$ means that $\tau \equiv \tau_i$ for some $i \in 1..n$. We say a type is static if it contains no *Dyn* type components.

3.1 Syntax

Our language is an explicitly annotated lambda calculus with term constants, i.e. integers and booleans. We include *parallel terms* from [9], which are annotated by sequences, and form one of the key features in our system. Similarly to intersection, the parallel operator is non-commutative and non-idempotent: $M^\tau \mid N^\rho \neq N^\rho \mid M^\tau$ and $M^\tau \mid M^\tau \neq M^\tau$. Let M and N (possibly with subscripts) range over typed terms, x, y and z (possibly with subscripts) range over term variables, k range over term constants, such as integers and booleans, and i, j, m and n range over positive integers. We use Π and Υ (possibly with subscripts) to range over parallel terms $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$, where $n \geq 1$, and call each $M_i^{\tau_i}$ a *component* of Π^σ. We extend the language with built-in addition; the other arithmetic operations can be defined similarly. We define the syntax of *type-annotated terms*, and supporting definitions [9], below:

$$
\begin{array}{rlll}
\text{Monotyped Terms} & M & ::= & k^B \mid c_i^\tau(x) \mid \lambda x : \sigma \,.\, M^\tau \mid \\
& & & M^\tau\, \Pi^\sigma \mid M^\tau + M^\tau \\
\text{Parallel Terms} & \Pi & ::= & (M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}) \quad n \geq 1
\end{array}
$$

Coercions [9], of the form $c_i^\tau(x)$, annotate a term variable with a monotype. Considering the example $\lambda x : ((Int \to Int) \to Int \to Int) \wedge (Int \to Int) \,.\, x\; x$, we have that x is typed by the sequence annotated in the lambda abstraction. However, the type used in the typing derivation for each occurrence of x will be an element of that sequence. Therefore, we annotate the term as follows: $\lambda x : ((Int \to Int) \to Int \to Int) \wedge (Int \to Int) \,.\, c_i^{(Int\to Int)\to Int\to Int}(x)\; c_j^{Int\to Int}(x)$

Definition 3.1 (Coercion). Given a variable x, a *coercion* $c_i^\tau(x)$ assigns type τ and *flow mark* i to x (flow marks are not relevant now, and will be explained in subsection 5.1).

Definition 3.2 (Rank). The *rank* of a type is defined by the following rules:

- rank(τ) = 0, if τ is a simple type i.e. no occurrences of the intersection operator;
- rank($\sigma \to \tau$) = max(1 + rank(σ), rank(τ)), if rank(σ) + rank(τ) > 0;
- rank($\tau_1 \wedge \ldots \wedge \tau_n$) = max(1, rank($\tau_1$), ..., rank($\tau_n$)) for $n \geq 2$.

Given a term M^τ, $fv(M^\tau)$ denotes the set of free variables in M^τ. We say a term is static if it contains only static type annotations. According to the definition of rank restriction [26, 31], a *rank n intersection type* can have no intersection type connective $\wedge$ to the left of n or more arrow type constructors $\to$. We restrict types in our system to be only of up to rank 2, e.g. $((\tau_1 \to \rho_1) \wedge \tau_1 \to \rho_1) \wedge ((\tau_2 \to \rho_2) \wedge \tau_2 \to \rho_2)$ is a valid type; $(((\tau \to \rho) \wedge \tau) \to \rho) \to \tau$ is not. In a λ-abstraction $\lambda x : \sigma \,.\, M^\tau$, type σ is a rank 1 or lower type.

Definition 3.3 (Typing Context). A *typing context* is a finite set, represented by $\{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$, of *type bindings* between type variables and rank 1 σ types. We use Γ (possibly with subscripts) to range over typing contexts, and write $\emptyset$ for an empty context. We write $x : \sigma$ for the context $\{x : \sigma\}$ and abbreviate $x : \sigma \equiv \{x : \sigma\}$; and write Γ_1, Γ_2 for the union of contexts Γ_1 and Γ_2, assuming Γ_1 and Γ_2 are disjoint, and abreviate $\Gamma_1, \Gamma_2 \equiv \Gamma_1 \cup \Gamma_2$.

Definition 3.4 (Joining Typing Contexts). Let Γ_1 and Γ_2 be two typing contexts. $\Gamma_1 \wedge \Gamma_2$ is a typing context, where $x : \sigma \in \Gamma_1 \wedge \Gamma_2$ if and only if σ is defined as follows:

$$
\sigma = \begin{cases}
\sigma_1 \wedge \sigma_2, & if\ x : \sigma_1 \in \Gamma_1\ and\ x : \sigma_2 \in \Gamma_2 \\
\sigma_1, & if\ x : \sigma_1 \in \Gamma_1\ and\ \neg\exists\sigma_2 \,.\, x : \sigma_2 \in \Gamma_2 \\
\sigma_2, & if\ \neg\exists\sigma_1 \,.\, x : \sigma_1 \in \Gamma_1\ and\ x : \sigma_2 \in \Gamma_2
\end{cases}
$$

4 GRADUAL INTERSECTION TYPE SYSTEM

Before defining our gradual intersection type system, we present some auxiliary definitions.

4.1 Consistency and Precision

The consistency relation $\sim$ [15, 38] forms, along with the *Dyn* type, the key cornerstones of gradual typing. It allows the comparison of gradual types, where two types are consistent if they are equal in the parts where they are static. However, we must adapt consistency to support non-idempotent and non-commutative intersection types. Due to our interpretation of intersection types, which consists in assigning various types to an expression, we consider the *Dyn* type incompatible with sequences. Thus, we restrict *Dyn* to be consistent only with rank 0 monotypes τ, and so sequences can only be consistent with other sequences. With this design choice, our system stays simple while still keeping the desired expressive power.

Definition 4.1 (Consistency). Given two types σ and υ, such that $rank(\sigma) = rank(\upsilon)$, the *consistency* relation between σ and υ is defined by the following set of axioms and rules:

$$
\sigma \sim \sigma \qquad Dyn \sim \tau \qquad \tau \sim Dyn \qquad \frac{\sigma_1 \sim \sigma_2 \qquad \tau_1 \sim \tau_2}{\sigma_1 \to \tau_1 \sim \sigma_2 \to \tau_2}
$$

$$
\frac{\tau_1 \sim \rho_1 \ \ldots\ \tau_n \sim \rho_n}{\tau_1 \wedge \ldots \wedge \tau_n \sim \rho_1 \wedge \ldots \wedge \rho_n}
$$

We also require a pattern matching relation that retrieves monotypes from dynamically typed functions in applications, or from dynamically typed arguments in additions.

Definition 4.2 (Pattern Matching). The definition follows:

$$
Dyn \rhd Dyn \to Dyn \qquad \sigma \to \tau \rhd \sigma \to \tau
$$

$$
Dyn \rhd B \qquad B \rhd B
$$

The *precision* relation (definition 4.3) between two types, written as $\sigma \sqsubseteq \upsilon$, holds if type σ is more unknown than υ. Therefore, the *Dyn* type is less precise ($\sqsubseteq$) than any other monotype τ. We lift the precision relation to contexts (definition 4.4) and terms (definition 4.5).

Definition 4.3 (Precision). Given two types σ and υ, such that $rank(\sigma) = rank(\upsilon)$, the precision relation between σ and υ is defined by the following set of axioms and rules:

$$
\sigma \sqsubseteq \sigma \qquad Dyn \sqsubseteq \tau \qquad \frac{\sigma_1 \sqsubseteq \sigma_2 \qquad \tau_1 \sqsubseteq \tau_2}{\sigma_1 \to \tau_1 \sqsubseteq \sigma_2 \to \tau_2}
$$

$$
\frac{\tau_1 \sqsubseteq \rho_1 \ \ldots\ \tau_n \sqsubseteq \rho_n}{\tau_1 \wedge \ldots \wedge \tau_n \sqsubseteq \rho_1 \wedge \ldots \wedge \rho_n}
$$

Definition 4.4 (Precision on Contexts). Precision between two contexts Γ_1 and Γ_2, where both have type bindings for exactly the same variables, is defined as point-wise precision between bound types: $\Gamma_1, x : \sigma \sqsubseteq \Gamma_2, x : \upsilon \iff \Gamma_1 \sqsubseteq \Gamma_2$ and $\sigma \sqsubseteq \upsilon$; and $\emptyset \sqsubseteq \emptyset$.

Definition 4.5 (Precision on Terms). Precision between two terms, $\Pi^\sigma \sqsubseteq \Upsilon^\upsilon$, means that Π^σ has less precise type annotations than Υ^υ:

$$[\text{P-Con}]\ \frac{}{k^B \sqsubseteq k^B} \qquad [\text{P-Var}]\ \frac{\rho \sqsubseteq \tau}{c_i^\rho(x) \sqsubseteq c_i^\tau(x)}$$

$$[\text{P-Abs}]\ \frac{\upsilon \sqsubseteq \sigma \qquad N^\rho \sqsubseteq M^\tau}{\lambda x : \upsilon \,.\, N^\rho \sqsubseteq \lambda x : \sigma \,.\, M^\tau}$$

$$[\text{P-App}]\ \frac{N^\rho \sqsubseteq M^\tau \qquad \Upsilon^\upsilon \sqsubseteq \Pi^\sigma}{N^\rho\ \Upsilon^\upsilon \sqsubseteq M^\tau\ \Pi^\sigma}$$

$$[\text{P-Add}]\ \frac{N_1^{\rho_1} \sqsubseteq M_1^{\tau_1} \qquad N_2^{\rho_2} \sqsubseteq M_2^{\tau_2}}{N_1^{\rho_1} + N_2^{\rho_2} \sqsubseteq M_1^{\tau_1} + M_2^{\tau_2}}$$

$$[\text{P-Par}]\ \frac{N_1^{\rho_1} \sqsubseteq M_1^{\tau_1} \quad \ldots \quad N_n^{\rho_n} \sqsubseteq M_n^{\tau_n}}{N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n} \sqsubseteq M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}}$$

Proposition 4.6 (Monotonicity of $\Gamma_1 \wedge \Gamma_2$ w.r.t. Precision). *If $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_2' \sqsubseteq \Gamma_2$ then $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$.*

4.2 Type System

Components of a parallel term are differently typed versions of the same term, thus equivalent modulo α-conversion. The typed calculus of [9] enforces this restriction by synchronously typing the components of a parallel term. In the parallel application $M_1^{\tau_1}\ \Pi_1^{\sigma_1} \mid M_2^{\tau_2}\ \Pi_2^{\sigma_2}$ both $M_1^{\tau_1}$ and $M_2^{\tau_2}$ are identical terms with different type annotations, and the same is true for $\Pi_1^{\sigma_1}$ and $\Pi_2^{\sigma_2}$. Type checking is simply a matter of checking $M_1^{\tau_1} \mid M_2^{\tau_2}$ and then checking $\Pi_1^{\sigma_1} \mid \Pi_2^{\sigma_2}$, rather than checking individually each component, $M_1^{\tau_1}\ \Pi_1^{\sigma_1}$ and then $M_2^{\tau_2}\ \Pi_2^{\sigma_2}$. With this approach, the generating rules are able to ensure that components of the parallel term are equivalent modulo α-conversion.

This restriction cannot be enforced in our system, because it is not preserved by reduction. In fact, equivalence modulo α-conversion of components must be relaxed because during term reduction some components may gather more run-time checks than others. Our type system provides this necessary flexibility. We present the $\bowtie$ (variant) relation between terms in definition 4.7, and expand it in section 5 to account for run-time checks and errors. In essence, $\Pi^\sigma \bowtie \Upsilon^\upsilon$ (Π^σ is a variant term of Υ^υ) holds if Π^σ and Υ^υ have the same shape of their syntactic trees, while disregarding extra run-time checks and errors. We assume terms are equivalent up to α-reducion, in order to prevent variable capture. For example, $\lambda x \,.\, \lambda y \,.\, x \bowtie \lambda z \,.\, \lambda w \,.\, z$ holds, but $\lambda x \,.\, \lambda y \,.\, x \not\bowtie \lambda z \,.\, \lambda w \,.\, w$.

Definition 4.7 (Variant Terms $\bowtie$). The $\bowtie$ relation is defined by the following rules:

$$[\text{V-Con}]\ \frac{}{k^B \bowtie k^B} \qquad [\text{V-Var}]\ \frac{}{c_i^\tau(x) \bowtie c_i^\rho(x)}$$

$$[\text{V-Abs}]\ \frac{M^\tau \bowtie N^\rho}{\lambda x : \sigma \,.\, M^\tau \bowtie \lambda x : \upsilon \,.\, N^\rho}$$

$$[\text{V-App}]\ \frac{M^\tau \bowtie N^\rho \qquad \Pi^\sigma \bowtie \Upsilon^\upsilon}{M^\tau\ \Pi^\sigma \bowtie N^\rho\ \Upsilon^\upsilon}$$

$$[\text{V-Add}]\ \frac{M_1^{\tau_1} \bowtie N_1^{\rho_1} \qquad M_2^{\tau_2} \bowtie N_2^{\rho_2}}{M_1^{\tau_1} + M_2^{\tau_2} \bowtie N_1^{\rho_1} + N_2^{\rho_2}}$$

$$[\text{V-Par}]\ \frac{M_1^{\tau_1} \bowtie N_1^{\rho_1} \quad \ldots \quad M_n^{\tau_n} \bowtie N_n^{\rho_n}}{M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \bowtie N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n}}$$

Definition 4.8 (Variant Set). We define a variant set as follows:

$$\bowtie (M_1^{\tau_1}, \ldots, M_n^{\tau_n}) \overset{def}{=} \forall i \in 1..n, j \in 1..n \,.\, M_i^{\tau_i} \bowtie M_j^{\tau_j}$$

We define the gradual type system in figure 2, and its counterpart static type system in figure 1. The only difference between both type systems is that in the static type system, due to the lack of the *Dyn* type, the consistency $\sim$ and pattern matching $\triangleright$ relations reduce to equality. This difference manifests only in the formulation of rules [T-App] and [T-Add]. Hence, the remaining rules ([T-Con], [T-Var], [T-AbsI], [T-AbsK] and [T-Par]) are obtained from figure 1.

Although each term is annotated with its type, we may omit type annotations if they are trivially reconstructed, e.g. $\lambda x : \sigma \,.\, M^\tau$ instead of $(\lambda x : \sigma \,.\, M^\tau)^{\sigma \to \tau}$. We impose the following restriction on lambda abstractions. If x occurs free in M^ρ, then the occurrences of x in $\lambda x : \sigma \,.\, M^\rho$ are in a one-to-one correspondence with the elements of σ. Thus, for each element of the abstraction's annotation, there is a single variable in the body that is typed by that element, and vice-versa. Furthermore, the order of variables in the body matches the order of the related elements in the type annotation. Therefore, lambda abstractions, whose bound variable occurs in the body, have the following form: $\lambda x : \tau_1 \wedge \ldots \wedge \tau_n \,.\, \ldots\ c_0^{\tau_1}(x)\ \ldots\ c_0^{\tau_n}(x)\ \ldots$. Also, according to rule [T-App], the condition $\upsilon \sim \sigma$ ensures the order of components in the argument parallel term matches the domain type of the function. Therefore, applications with parallel terms as arguments are of the form: $M^{\tau_1 \wedge \ldots \wedge \tau_n \to \tau}\ (N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n})$, assumming $\upsilon = \rho_1 \wedge \ldots \wedge \rho_n$ and $\sigma = \tau_1 \wedge \ldots \wedge \tau_n$. This restriction ensures the system benefits from important properties, which will be introduced in section 5.

To enforce this restriction, we rely on type system rules and the non-commutativity and non-idempotence of intersection types. Rule [T-Var] inserts into the context the instances assigned to each variable. Then, rules [T-App], [T-Add] and [T-Par] join the contexts, per definition 3.4, such that types bound to the same variable are joined in a sequence ordered w.r.t. the order of ocurrences of the variable. Finally, rule [T-AbsI] ensures the type bound to the variable in the context equals the type annotation in the abstraction, ensuring the one-to-one correspondence. The exception is

$$[\text{T-Con}]\ \frac{\text{k is a constant of base type B}}{\emptyset \vdash_\wedge k^B\ : B} \qquad [\text{T-Var}]\ \frac{}{x : \tau \vdash_\wedge c_i^\tau(x) : \tau} \qquad [\text{T-AbsI}]\ \frac{\Gamma, x : \sigma \vdash_\wedge M^\tau : \tau}{\Gamma \vdash_\wedge \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau}\ x \in fv(M^\tau)$$

$$[\text{T-AbsK}]\ \frac{\Gamma \vdash_\wedge M^\tau : \tau}{\Gamma \vdash_\wedge \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau}\ x \notin fv(M^\tau) \qquad [\text{T-App}]\ \frac{\Gamma_1 \vdash_\wedge M^{\sigma\to\tau} : \sigma \to \tau \quad \Gamma_2 \vdash_\wedge \Pi^\sigma : \sigma}{\Gamma_1 \wedge \Gamma_2 \vdash_\wedge M^{\sigma\to\tau}\ \Pi^\sigma : \tau} \qquad [\text{T-Add}]\ \frac{\Gamma_1 \vdash_\wedge M^{Int} : Int \quad \Gamma_2 \vdash_\wedge N^{Int} : Int}{\Gamma_1 \wedge \Gamma_2 \vdash_\wedge M^{Int} + N^{Int} : Int}$$

$$[\text{T-Par}]\ \frac{\Gamma_1 \vdash_\wedge M_1^{\tau_1} : \tau_1 \ \ldots\ \Gamma_n \vdash_\wedge M_n^{\tau_n} : \tau_n \qquad \bowtie (M_1^{\tau_1}, \ldots, M_n^{\tau_n})}{\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_\wedge M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n}\ \forall i \,.\, rank(\tau_i) = 0$$

Figure 1: Static Intersection Type System ($\Gamma \vdash_\wedge \Pi : \sigma$)

Static Intersection Type System ($\Gamma \vdash_\wedge \Pi : \sigma$) rules and

$$[\text{T-App}]\ \frac{\Gamma_1 \vdash_{\wedge G} M^\rho : \rho \qquad \rho \rhd \sigma \to \tau \qquad \Gamma_2 \vdash_{\wedge G} \Pi^\upsilon : \upsilon \qquad \upsilon \sim \sigma}{\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^\rho\ \Pi^\upsilon : \tau} \qquad [\text{T-Add}]\ \frac{\Gamma_1 \vdash_{\wedge G} M^\tau : \tau \qquad \tau \rhd Int \qquad \Gamma_2 \vdash_{\wedge G} N^\rho : \rho \qquad \rho \rhd Int}{\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^\tau + N^\rho : Int}$$

Figure 2: Gradual Intersection Type System ($\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$)

when the bound variable does not occur in the body of a lambda abstraction, in which case we apply instead rule [T-AbsK].

Proposition 4.9. *If $\Gamma \vdash_{\wedge G} \lambda x : \tau_1 \wedge \ldots \wedge \tau_n \,.\, M^\rho : \tau_1 \wedge \ldots \wedge \tau_n \to \rho$, and $x \in fv(M^\rho)$, then the number of free occurrences of x in M^ρ equals n, and these occurrences are typed with $\tau_1, \ldots, \tau_n$, considering an order from left to right.*

Rule [T-App] uses the standard relations from gradual typing [15], the $\rhd$ and $\sim$ relations. We also introduce a new rule [T-Par] which individually types terms in a parallel term. Note that components of a parallel term must share the same term structure ($\bowtie$) (this replaces the Local Renaming rule from [9]). Since components share the same free variables, they are typed using a unique context Γ.

Example 4.10. We illustrate these concepts in the following example. We set flow marks to 0 since they don't influence type checking. Consider the following expression

$$(\lambda x : Dyn \wedge Dyn \,.\, c_0^{Dyn}(x)\ c_0^{Dyn}(x))$$
$$(\lambda y : Int^2 \,.\, c_0^{Int^2}(y) \mid \lambda z : Int \,.\, c_0^{Int}(z))$$

where we abbreviate as follows: Dyn^2 denotes the type $Dyn \to Dyn$; I^2 denotes the type $Int \to Int$; I^4 denotes the type $(Int \to Int) \to Int \to Int$. We have the following derivations.

Derivation D_1:

[T-Var]	$x : Dyn \vdash_{\wedge G} c_0^{Dyn}(x) : Dyn$
$def.4.2$	$Dyn \rhd Dyn \to Dyn$
$def.4.1$	$Dyn \sim Dyn$
[T-App]	$x : Dyn \wedge Dyn \vdash_{\wedge G} c_0^{Dyn}(x)\ c_0^{Dyn}(x) : Dyn$
[T-AbsI]	$\emptyset \vdash_{\wedge G} \lambda x : Dyn \wedge Dyn \,.$ $c_0^{Dyn}(x)\ c_0^{Dyn}(x) : Dyn \wedge Dyn \to Dyn$

Derivation D_2:

[T-Var]	$y : Int \to Int \vdash_{\wedge G} c_0^{Int\to Int}(y) : Int \to Int$
[T-AbsI]	$\emptyset \vdash_{\wedge G} \lambda y : Int \to Int \,.\, c_0^{Int\to Int}(t) : I^4$

Derivation D_3:

[T-Var]	$z : Int \vdash_{\wedge G} c_0^{Int}(z) : Int$
[T-AbsI]	$\emptyset \vdash_{\wedge G} \lambda z : Int \,.\, c_0^{Int}(z) : Int \to Int$

Final derivation: by D_2 and D_3 and since $\lambda y : Int \to Int \,.\, c_0^{Int\to Int}(y) \bowtie \lambda z : Int \,.\, c_0^{Int}(z)$ holds, and finally by D_1:

[T-Par]	$\emptyset \vdash_{\wedge G} \lambda y : Int \to Int \,.\, c_0^{Int\to Int}(y) \mid$ $\lambda z : Int \,.\, c_0^{Int}(z)) : Int^4 \wedge Int^2$
$def.4.2$	$Dyn \wedge Dyn \to Dyn \rhd Dyn \wedge Dyn \to Dyn$
$def.4.1$	$(Int^4 \wedge (Int \to Int) \sim Dyn \wedge Dyn$
[T-App]	$\emptyset \vdash_{\wedge G} (\lambda x : Dyn \wedge Dyn \,.\, c_0^{Dyn}(x)\ c_0^{Dyn}(x))$ $(\lambda y : Int^2 \,.\, c_0^{Int^2}(y) \mid \lambda z : Int \,.\, c_0^{Int}(z)) : Dyn$

We show the typed calculus has the following properties, including those from [40]:

Proposition 4.11 (Sequence Types and Parallel Terms). *If $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$ and $\sigma \equiv \tau_1 \wedge \ldots \wedge \tau_n$, with $n > 1$, then Π^σ is a parallel term, namely $\Pi^\sigma \equiv M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$ for some $M_1^{\tau_1}, \ldots, M_n^{\tau_n}$.*

Proposition 4.12 (Basic Properties). *If $\Gamma \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then:*

1. *for any $x : \sigma \in \Gamma$ and for any $M_i^{\tau_i}$ ($1 \leq i \leq n$), each occurrence of x in $M_i^{\tau_i}$ is the argument of a coercion of the shape c_j^τ where $\tau \in \sigma$;*

(2) *for any term of the shape* $N_1^{\rho_1} \mid \ldots \mid N_m^{\rho_m}$*, where for all* i ($1 \leq i \leq m$) *there exists* j ($1 \leq j \leq n$) *such that* $N_i^{\rho_i} \equiv M_j^{\tau_j}$*, the judgement* $\Gamma \vdash_{\wedge G} N_1^{\rho_1} \mid \ldots \mid N_m^{\rho_m} : \rho_1 \wedge \ldots \wedge \rho_m$ *is derivable. If we can derive a parallel term, we can also derive a permutation of it, a shorter parallel term and a parallel term with copies of some components.*

Lemma 4.13 (Inversion Lemma).

(1) *Rule [T-Con]. If* $\emptyset \vdash_{\wedge G} k^B : B$ *then* k *is a constant of base type* B.
(2) *Rule [T-Var]. We have that* $x : \tau \vdash_{\wedge G} c_i^\tau(x) : \tau$ *holds.*
(3) *Rule [T-AbsI]. Assuming* $x \in fv(M^\tau)$*, if* $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^\tau : \sigma \rightarrow \tau$ *then* $\Gamma, x : \sigma \vdash_{\wedge G} M^\tau : \tau$.
(4) *Rule [T-AbsK]. Assuming* $x \notin fv(M^\tau)$*, if* $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^\tau : \sigma \rightarrow \tau$ *then* $\Gamma \vdash_{\wedge G} M^\tau : \tau$.
(5) *Rule [T-App]. If* $\Gamma \vdash_{\wedge G} M^\rho\, \Pi^\upsilon : \tau$ *then typing context* Γ *can be divided into* Γ_1 *and* Γ_2 *such that* $\Gamma_1 \wedge \Gamma_2 = \Gamma$ *and* $\Gamma_1 \vdash_{\wedge G} M^\rho : \rho$, $\rho \rhd \sigma \rightarrow \tau$, $\Gamma_2 \vdash_{\wedge G} \Pi^\upsilon : \upsilon$ *and* $\upsilon \sim \sigma$.
(6) *Rule [T-Add]. If* $\Gamma \vdash_{\wedge G} M^\tau + N^\rho : Int$ *then typing context* Γ *can be divided into* Γ_1 *and* Γ_2 *such that* $\Gamma_1 \wedge \Gamma_2 = \Gamma$ *and* $\Gamma_1 \vdash_{\wedge G} M^\tau : \tau$ *and* $\tau \rhd Int$ *and* $\Gamma_2 \vdash_{\wedge G} N^\rho : \rho$ *and* $\rho \rhd Int$.
(7) *Rule [T-Par]. If* $\Gamma \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ *then typing context* Γ *can be divided into* $\Gamma_1, \ldots, \Gamma_n$ *such that* $\Gamma_1 \wedge \ldots \wedge \Gamma_n = \Gamma$ *and* $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ *and* $\ldots$ *and* $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$ *and* $\bowtie (M_1^{\tau_1}, \ldots, M_n^{\tau_n})$.

Proof. By induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$. □

Theorem 4.14 (Conservative Extension of Type System). *If* Π^σ *is static and* σ *is a static type, then* $\Gamma \vdash_\wedge \Pi^\sigma : \sigma \iff \Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$.

Proof. By induction on the length of the derivation tree of $\Gamma \vdash_\wedge \Pi^\sigma : \sigma$ and $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$. □

Theorem 4.15 (Monotonicity w.r.t. Precision). *If* $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$ *and* $\Upsilon^\upsilon \sqsubseteq \Pi^\sigma$ *then* $\exists \Gamma'$ *such that* $\Gamma' \sqsubseteq \Gamma$ *and* $\Gamma' \vdash_{\wedge G} \Upsilon^\upsilon : \upsilon$ *and* $\upsilon \sqsubseteq \sigma$.

Proof. By induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$. □

5 CAST CALCULUS

In gradual typing, type verification is also delayed to run-time, thus our language must be compiled into a calculus that supports run-time verification. This target language is widely known as the *Cast Calculus* [15], compiled from the typed source language by adding run-time type checks called *casts*. We define the syntax of this calculus for our system below and its typing rules in figure 3:

$$\begin{array}{rrcl} \text{Monotyped Terms} & M & ::= & \ldots \mid M^\tau : \tau \Rightarrow \tau \mid wrong^\tau \\ \text{Parallel Terms} & \Pi & ::= & \ldots \mid wrong^\sigma \end{array}$$

Notice that new terms are added to the syntax of section 3. The run-time verification, in the form of the cast $M^\tau : \tau \Rightarrow \rho$, checks if a term M^τ of source type τ can be treated as having target type ρ. The term $wrong^\sigma$ signals a run-time error, being considered either a monotyped term or a parallel term depending on the type annotation. These terms are adapted from [15], and serve the same purpose. Regarding the type system, new rules for application [T-App] and addition [T-Add] are introduced, as well as for casts [T-Cast] and errors [T-Wrong]. Since casts make explicit the consistency and pattern matching checks, these are removed from rules [T-App] and [T-Add]. The remaining rules ([T-Con], [T-Var], [T-AbsI], [T-AbsK] and [T-Par]) are obtained from figure 2. We also expand the definition of $\sqsubseteq$ (precision from definition 4.5) and $\bowtie$ (variant terms from definition 4.7) on terms, to include casts and errors:

Definition 5.1 (Precision on Cast Calculus). We redefine $\sqsubseteq$ on terms with the rules from definition 4.5 and the following rules:

$$[\text{P-Cast}]\ \frac{N^{\rho_1} \sqsubseteq M^{\tau_1} \qquad \rho_1 \sqsubseteq \tau_1 \qquad \rho_2 \sqsubseteq \tau_2}{N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq M^{\tau_1} : \tau_1 \Rightarrow \tau_2}$$

$$[\text{P-Wrong}]\ \frac{\upsilon \sqsubseteq \sigma}{\Upsilon^\upsilon \sqsubseteq wrong^\sigma} \qquad [\text{P-CastL}]\ \frac{\begin{array}{c} N^{\rho_1} \sqsubseteq M^\tau \\ \rho_1 \sqsubseteq \tau \qquad \rho_2 \sqsubseteq \tau \end{array}}{N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq M^\tau}$$

$$[\text{P-CastR}]\ \frac{\begin{array}{c} N^\rho \sqsubseteq M^{\tau_1} \\ \rho \sqsubseteq \tau_1 \qquad \rho \sqsubseteq \tau_2 \end{array}}{N^\rho \sqsubseteq M^{\tau_1} : \tau_1 \Rightarrow \tau_2}$$

Definition 5.2 (Variant Terms on Cast Calculus). We redefine $\bowtie$ on terms with the rules from definition 4.7 and the following rules:

$$[\text{V-Cast}]\ \frac{M^{\tau_1} \bowtie N^{\rho_1}}{M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2}$$

$$[\text{V-WrongL}]\ \frac{\begin{array}{c} \sigma = \tau_1 \wedge \ldots \wedge \tau_n \\ \upsilon = \rho_1 \wedge \ldots \wedge \rho_n \end{array}}{wrong^\sigma \bowtie \Upsilon^\upsilon} \qquad [\text{V-WrongR}]\ \frac{\begin{array}{c} \sigma = \tau_1 \wedge \ldots \wedge \tau_n \\ \upsilon = \rho_1 \wedge \ldots \wedge \rho_n \end{array}}{\Pi^\sigma \bowtie wrong^\upsilon}$$

$$[\text{V-CastL}]\ \frac{M^{\tau_1} \bowtie N^\rho}{M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^\rho}$$

$$[\text{V-CastR}]\ \frac{M^\tau \bowtie N^{\rho_1}}{M^\tau \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2}$$

Casts and run-time errors are not considered syntactic terms of the source language, such as applications or variables. Instead, they denote transformations between types and typed expressions, i.e. their presence in the language comes solely from types and not from terms. So, they play no role in deciding whether an expression is syntactically equivalent to another, and thus are treated as void elements in the above definitions.

5.1 Flow Marking

Before compiling expressions into the cast calculus, we must add annotations that guarantee the correct flow of terms from argument positions to their respective variable occurrences. According to definitions 4.1 and 4.2, when applying a function to an argument, the *Dyn* type is thought of a yet unknown static type. In $\lambda x : Dyn \,.\, c_0^{Dyn}(x) + 1^{Int}$, the *Dyn* type can be thought of as being the *Int* type, but with a run-time type verification. In the presence

Gradual Intersection Type System ($\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$) rules and

$$[\text{T-App}]\ \frac{\Gamma_1 \vdash_{\wedge CC} M^{\sigma \to \tau} : \sigma \to \tau \qquad \Gamma_2 \vdash_{\wedge CC} \Pi^{\sigma} : \sigma}{\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M^{\sigma \to \tau}\ \Pi^{\sigma} : \tau}$$

$$[\text{T-Add}]\ \frac{\Gamma_1 \vdash_{\wedge CC} M^{Int} : Int \qquad \Gamma_2 \vdash_{\wedge CC} N^{Int} : Int}{\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M^{Int} + N^{Int} : Int} \qquad [\text{T-Cast}]\ \frac{\Gamma \vdash_{\wedge CC} M^{\tau} : \tau \qquad \tau \sim \rho}{\Gamma \vdash_{\wedge CC} M^{\tau} : \tau \Rightarrow \rho : \rho} \qquad [\text{T-Wrong}]\ \frac{}{\emptyset \vdash_{\wedge CC} wrong^{\sigma} : \sigma}$$

Figure 3: Gradual Intersection Cast Calculus ($\Gamma \vdash_{\wedge CC} \Pi^{\sigma} : \sigma$)

of non-commutative and non-idempotent intersection types, this meaning of the *Dyn* type differs slightly. We can have expressions with several instances of the *Dyn* type:

$$(\lambda x : Dyn \wedge Dyn \,.\, c_0^{Dyn}(x)\ c_0^{Dyn}(x))$$
$$(\lambda y : Int \to Int \,.\, c_0^{Int \to Int}(y) \mid \lambda z : Int \,.\, c_0^{Int}(z))$$

These can be thought of as different, yet unknown, static types, with a delayed type verification in run-time. The first occurrence, appearing on the left of the $\wedge$ and also on the first coercion, can be thought of as the type $(Int \to Int) \to Int \to Int$. The second occurrence, appearing on the right of the $\wedge$ and also on the second coercion, can be thought of as the type $Int \to Int$. Therefore, since these two *Dyn* occurrences represent two different types, they will correspond to distinct components of the argument parallel term. Operational semantics must distinguish these types, and keep the flow of arguments to their respective occurrences [9] as intended. The first term in the parallel should flow to the first occurrence of x while the second term should flow to the second occurrence. However, since the different occurrences are typed with the same *Dyn* type, it is possible that the first component in the parallel term flows to both of them. This erroneous behaviour originates an expression which is not the intention of the programmer and that leads to a *wrong* error: $(\lambda y : Int \to Int \,.\, c_0^{Int \to Int}(y))\ (\lambda y : Int \to Int \,.\, c_0^{Int \to Int}(y))$.

Our solution is to mark coercions with an index, called *flow mark*, according to the position of its type in the lambda abstraction's type annotation. Having both coercions and parallel term components ordered w.r.t. the order of instances in lambda abstraction annotations facilitates this. So, we effectively link each component in the argument parallel term with its corresponding coercion in the body. We define flow marking in figure 4, and also in definitions 5.3 and 5.4. We overload the type connective $\wedge$ to also accept indices, and use $\bar{i}$ (possibly with subscripts) to range over lists of indices. We then overload the $\wedge$ operator from typing contexts, definition 3.4, to also accept flow contexts, and reuse the definition.

Definition 5.3 (Flow Context). A *flow context* is a finite set, of the form $\{x_1 : \overline{i_1},\ \ldots,\ x_n : \overline{i_n}\}$, of (variable, list of indices) pairs called *flow bindings*, where $\overline{i_1} = i_{11} \wedge \ldots \wedge i_{1j}$ and $\ldots$ and $\overline{i_n} = i_{n1} \wedge \ldots \wedge i_{nm}$. We use Σ (possibly with subscripts) to range over flow contexts, and write $\emptyset$ for an empty context. We write $x : \bar{i}$ for the context $\{x : \bar{i}\}$ and abbreviate $x : \bar{i} \equiv \{x : \bar{i}\}$; and write Σ_1, Σ_2 for the union of contexts Σ_1 and Σ_2, assuming Σ_1 and Σ_2 are disjoint, and abreviate $\Sigma_1, \Sigma_2 \equiv \Sigma_1 \cup \Sigma_2$.

Definition 5.4 (Flow Marking on Contexts). We obtain the corresponding flow context from a typing context by replacing the types with indices: $\Gamma \hookrightarrow \Sigma \iff \Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \hookrightarrow \Sigma, x : 1 \wedge \ldots \wedge n$; and $\emptyset \hookrightarrow \emptyset$. We define the abbreviation $(\Gamma)_{\hookrightarrow}$ as follows: $(\Gamma)_{\hookrightarrow} = \Sigma$, if $\Gamma \hookrightarrow \Sigma$.

Example 5.5. Consider the previous example after flow marking:

$$(\lambda x : Dyn \wedge Dyn \,.\, c_1^{Dyn}(x)\ c_2^{Dyn}(x))$$
$$(\lambda y : Int \to Int \,.\, c_1^{Int \to Int}(y) \mid \lambda z : Int \,.\, c_1^{Int}(z))$$

Notice that the first coercion in the λ-abstraction, with a mark of 1, will be replaced by the first component in the parallel term. Similarly, the second coercion, with mark 2, will be replaced by the second component. Both coercions in the parallel term are marked with 1 since there is only one instance in the annotation. Flow marking is type-preserving and monotonic w.r.t. precision [40]:

Theorem 5.6 (Type Preservation of Flow Marking). *If* $\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$ *then* $\Sigma \vdash_{\wedge G} \Pi^{\sigma} \hookrightarrow \Upsilon^{\sigma}$ *and* $\Gamma \vdash_{\wedge G} \Upsilon^{\sigma} : \sigma$*, where* $\Gamma \hookrightarrow \Sigma$.

Proof. By induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$. □

Theorem 5.7 (Monotonicity of Flow Marking). *If* $\Sigma_1 \vdash_{\wedge G} \Pi_1^{\sigma} \hookrightarrow \Pi_2^{\sigma}$ *and* $\Sigma_2 \vdash_{\wedge G} \Upsilon_1^{v} \hookrightarrow \Upsilon_2^{v}$ *and* $\Upsilon_1^{v} \sqsubseteq \Pi_1^{\sigma}$ *then* $\Upsilon_2^{v} \sqsubseteq \Pi_2^{\sigma}$.

Proof. By induction on the length of the derivation tree of $\Sigma_1 \vdash_{\wedge G} \Pi_1^{\sigma} \hookrightarrow \Pi_2^{\sigma}$. □

5.2 Cast Insertion

After applying the marking operation, the expression can be *compiled* into the cast calculus by the rules defined in figure 5. Most rules are straightforward, recursively inserting casts in the sub-expressions, but rule [C-App] deserves a thorough explanation.

Example 5.8. Going back to our example in subsection 4.2, we insert casts as follows:

$$((\lambda x : Dyn \wedge Dyn \,.\, (c_1^{Dyn}(x) : Dyn \Rightarrow Dyn^2)$$
$$(c_2^{Dyn}(x) : Dyn \Rightarrow Dyn))$$
$$: Dyn \wedge Dyn \to Dyn \Rightarrow Dyn \wedge Dyn \to Dyn)$$
$$((\lambda y : I^2 \,.\, c_1^{I^2}(y)) : I^4 \Rightarrow Dyn \mid (\lambda z : Int \,.\, c_1^{Int}(z)) : I^2 \Rightarrow Dyn)$$

Inserting casts in function terms is simple: make the source type the type of the function, and the target type the result of pattern

$$[\text{M-Con}]\ \frac{}{\emptyset \vdash_{\wedge G} k^B \hookrightarrow k^B} \qquad [\text{M-Var}]\ \frac{}{x : i \vdash_{\wedge G} c_0^\tau(x) \hookrightarrow c_i^\tau(x)} \qquad [\text{M-AbsI}]\ \frac{\Sigma, (x : \sigma)\hookrightarrow \vdash_{\wedge G} M^\tau \hookrightarrow N^\tau}{\Sigma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^\tau \hookrightarrow \lambda x : \sigma \,.\, N^\tau}\ x \in fv(M^\tau)$$

$$[\text{M-AbsK}]\ \frac{\Sigma \vdash_{\wedge G} M^\tau \hookrightarrow N^\tau}{\Sigma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^\tau \hookrightarrow \lambda x : \sigma \,.\, N^\tau}\ x \notin fv(M^\tau) \qquad [\text{M-App}]\ \frac{\Sigma_1 \vdash_{\wedge G} M^\tau \hookrightarrow N^\tau \qquad \Sigma_2 \vdash_{\wedge G} \Pi^\sigma \hookrightarrow \Upsilon^\sigma}{\Sigma_1 \wedge \Sigma_2 \vdash_{\wedge G} M^\tau\ \Pi^\sigma \hookrightarrow N^\tau\ \Upsilon^\sigma}$$

$$[\text{M-Add}]\ \frac{\Sigma_1 \vdash_{\wedge G} M_1^\tau \hookrightarrow N_1^\tau \qquad \Sigma_2 \vdash_{\wedge G} M_2^\rho \hookrightarrow N_2^\rho}{\Sigma_1 \wedge \Sigma_2 \vdash_{\wedge G} M_1^\tau + M_2^\rho \hookrightarrow N_1^\tau + N_2^\rho} \qquad [\text{M-Par}]\ \frac{\Sigma_1 \vdash_{\wedge G} M_1^{\tau_1} \hookrightarrow N_1^{\tau_1}\ \ldots\ \Sigma_n \vdash_{\wedge G} M_n^{\tau_n} \hookrightarrow N_n^{\tau_n}}{\Sigma_1 \wedge \ldots \wedge \Sigma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \hookrightarrow N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}}$$

Figure 4: Flow Marking ($\Sigma \vdash_{\wedge G} \Pi^\sigma \hookrightarrow \Upsilon^\sigma$)

$$[\text{C-Con}]\ \frac{\text{k is a constant of base type B}}{\emptyset \vdash_{\wedge CC} k^B \rightsquigarrow k^B : B} \qquad [\text{C-Var}]\ \frac{}{x : \tau \vdash_{\wedge CC} c_i^\tau(x) \rightsquigarrow c_i^\tau(x) : \tau}$$

$$[\text{C-AbsI}]\ \frac{\Gamma, x : \sigma \vdash_{\wedge CC} M^\tau \rightsquigarrow N^\tau : \tau}{\Gamma \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau \rightsquigarrow \lambda x : \sigma \,.\, N^\tau : \sigma \to \tau}\ x \in fv(M^\tau) \qquad [\text{C-AbsK}]\ \frac{\Gamma \vdash_{\wedge CC} M^\tau \rightsquigarrow N^\tau : \tau}{\Gamma \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau \rightsquigarrow \lambda x : \sigma \,.\, N^\tau : \sigma \to \tau}\ x \notin fv(M^\tau)$$

$$[\text{C-App}]\ \frac{\Gamma_1 \vdash_{\wedge CC} M^\rho \rightsquigarrow N^\rho : \rho \qquad \rho \rhd \sigma \to \tau \qquad \Gamma_2 \vdash_{\wedge CC} \Pi^\upsilon \rightsquigarrow \Upsilon^\upsilon : \upsilon \qquad \upsilon \sim \sigma}{\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M^\rho\ \Pi^\upsilon \rightsquigarrow (N^\rho : \rho \Rightarrow \sigma \to \tau)\ (\Upsilon^\upsilon : \upsilon \Rightarrow_\wedge \sigma) : \tau}$$

$$[\text{C-Add}]\ \frac{\Gamma_1 \vdash_{\wedge CC} M_1^\tau \rightsquigarrow N_1^\tau : \tau \qquad \tau \rhd \mathit{Int} \qquad \Gamma_2 \vdash_{\wedge CC} M_2^\rho \rightsquigarrow N_2^\rho : \rho \qquad \rho \rhd \mathit{Int}}{\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M_1^\tau + M_2^\rho \rightsquigarrow (N_1^\tau : \tau \Rightarrow \mathit{Int}) + (N_2^\rho : \rho \Rightarrow \mathit{Int}) : \mathit{Int}}$$

$$[\text{C-Par}]\ \frac{\Gamma_1 \vdash_{\wedge CC} M_1^{\tau_1} \rightsquigarrow N_1^{\tau_1} : \tau_1\ \ldots\ \Gamma_n \vdash_{\wedge CC} M_n^{\tau_n} \rightsquigarrow N_n^{\tau_n} : \tau_n}{\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge CC} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \rightsquigarrow N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n}\ \forall i \,.\, rank(\tau_i) = 0$$

$$\frac{\Pi^\sigma = M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \qquad \sigma = \tau_1 \wedge \ldots \wedge \tau_n \qquad \upsilon = \rho_1 \wedge \ldots \wedge \rho_n}{\Pi^\sigma : \sigma \Rightarrow_\wedge \upsilon = M_1^{\tau_1} : \tau_1 \Rightarrow \rho_1 \mid \ldots \mid M_n^{\tau_n} : \tau_n \Rightarrow \rho_n}$$

Figure 5: Gradual Intersection Cast Insertion ($\Gamma \vdash_{\wedge CC} \Pi^\sigma \rightsquigarrow \Upsilon^\sigma : \sigma$)

matching. In the example, an *identity cast* arises, since the source and target types are the same. Inserting casts in argument terms is not so simple. When type checking, we compare each element of the domain of the function's type with the appropriate element of the type of the argument: $Dyn \sim (Int \to Int) \to Int \to Int$ and $Dyn \sim (Int \to Int)$. Therefore, we add casts in each component of the parallel term, from its respective type to the type they are compared with using the $\sim$ relation. In a way, we add a cast from one sequence type to another, with their elements split between the components of the parallel term, according to $\Pi^\sigma : \sigma \Rightarrow_\wedge \upsilon$. Cast insertion is type-preserving and monotonic w.r.t. precision [40]:

Theorem 5.9 (Type Preservation of Cast Insertion). *If* $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$ *then* $\Gamma \vdash_{\wedge CC} \Pi^\sigma \rightsquigarrow \Upsilon^\sigma : \sigma$ *and* $\Gamma \vdash_{\wedge CC} \Upsilon^\sigma : \sigma$.

Proof. By induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$. □

Theorem 5.10 (Monotonicity of Cast Insertion). *If* $\Gamma_1 \vdash_{\wedge CC} \Pi_1^\sigma \rightsquigarrow \Pi_2^\sigma : \sigma$ *and* $\Gamma_2 \vdash_{\wedge CC} \Upsilon_1^\upsilon \rightsquigarrow \Upsilon_2^\upsilon : \upsilon$ *and* $\Upsilon_1^\upsilon \sqsubseteq \Pi_1^\sigma$ *then* $\Upsilon_2^\upsilon \sqsubseteq \Pi_2^\sigma$ *and* $\upsilon \sqsubseteq \sigma$.

Proof. By induction on the length of the derivation tree of $\Gamma_1 \vdash_{\wedge CC} \Pi_1^\sigma \rightsquigarrow \Pi_2^\sigma : \sigma$. □

6 OPERATIONAL SEMANTICS

We now introduce our operational semantics, adapted from [16], starting with the definition of normal forms and evaluation contexts:

$$\begin{array}{rrcl}
\text{Ground Types} & G & ::= & B \mid Dyn \to Dyn \\
\text{Values} & v & ::= & k^B \mid \lambda x : \sigma \,.\, M^\tau \mid v^G : G \Rightarrow Dyn \mid \\
 & & & v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \upsilon \to \rho \\
\text{Results} & r & ::= & v^\tau \mid wrong^\tau \\
\text{Parallel Values} & \pi & ::= & (v_1^{\tau_1} \mid \ldots \mid v_n^{\tau_n}) \mid wrong^\sigma \quad n \geq 1 \\
\text{Evaluation Contexts} & E & ::= & \square \mid E\ \Pi^\sigma \mid v^\tau\ E \mid E + M^\tau \mid \\
 & & & v^\tau + E \mid E : \tau \Rightarrow \rho
\end{array}$$

Ground types are used as a bridge when comparing different gradual types, carrying the information of the type constructor. Besides the standard normal forms of the λ-calculus, we also treat casts as values depending on their types. We consider both casts from a ground type to a *Dyn* type, and casts from a function type to a different function type, as values. In our language, $wrong^\tau$ may be a normal form, but its behaviour is different than those of values: it is pushed upwards along the syntactic tree. We distinguish between values and $wrong^\tau$, and consider both as results. Parallel values are either parallel terms composed solely of values, or a $wrong^\sigma$. Therefore, if there's a $wrong^\tau$ in any component, then it is not considered a parallel value, since the $wrong^\tau$ still needs to be pushed upwards. We write $E[\Pi^\sigma]$ for the term obtained by replacing the hole in E by the term Π^σ. We employ the call-by-value reduction strategy, as evidenced by our formulation of evaluation contexts.

Casts must be reduced to their normal form according to the rules of figure 6. Rules [EC-Identity] and [EC-Succeed] correspond to a successful cast reduction, i.e. the run-time check succeeded. Rules [EC-Application], [EC-Ground] and [EC-Expand] propagate casts through the expression. Rule [EC-Application] allows the verification of an application (the definition of $\Rightarrow_\wedge$ is in figure 5), assuming π^v is not a *wrong*. This is done by wrapping function casts around the argument and the whole expression, taking into account contravariance and covariance, respectively. Rules [EC-Ground] and [EC-Expand] reformulate the types within these checks, by passing them through ground types. Finally, the failure of a run-time check is given by rule [EC-Fail].

We also need reduction rules for lambda expressions. We introduce the gradual operational semantics in figure 8. The counterpart static operational semantics, written as $\longrightarrow_\wedge$, is equivalent to $\longrightarrow_{\wedge CC}$, except that casts and run-time errors are not included in the syntax, and both cast handler rules and rules [E-Push] and [E-Wrong] are not defined, as seen in figure 7.

Our calculus' reduction strategy is call-by-value, i.e. no reduction inside the body of a lambda abstraction, so only closed terms are evaluated. Therefore, term variables cannot be swapped, removed or duplicated, ensuring reduction preserves non-idempotent and non-commutative intersection types. The purpose of the flow marks becomes clear in rule [E-Beta]: the contraction of the beta-redex is performed by replacing each coercion with flow mark i, with the parallel term component in the ith position:

Definition 6.1 (Projection on Typed Parallel Values). If $\pi^\sigma = v_1^{\rho_1} \mid \ldots \mid v_n^{\rho_n}$ is a typed parallel value, $\sigma = \rho_1 \wedge \ldots \wedge \rho_n$ and $\rho \in \rho_1 \wedge \ldots \wedge \rho_n$ then: $\langle v_1^{\rho_1} \mid \ldots \mid v_n^{\rho_n} \rangle_i^\rho \overset{def}{=} v_i^{\rho_i} \quad$ if $\rho_i = \rho$

Flow marking, in figure 4, ensures the types of the coercions match the types of the component in the parallel term, and so, the condition $\rho_i = \rho$ always holds.

During reduction, any $wrong^\sigma$ is pushed upwards in the syntactic tree, according to rule [E-Wrong]. However, when reducing a parallel term, components which are not yet a result are simultaneously reduced one step, via rule [E-Par]. This means $wrong^\tau$ can arise in a component, in which case $wrong^\tau$ is pushed out, via rule [E-Push], effectively substituting the parallel term. If $wrong^\tau$ doesn't arise in any component of a parallel term, then that parallel term is considered a value.

We show several important properties, including those from [40], that hold for our operational semantics. We first show our calculus is a conservative extension of its static counterpart. Therefore, when no dynamic types are used, the calculus behaves as a static calculus, i.e. no type checking is delayed until run-time.

Theorem 6.2 (Conservative Extension of Operational Semantics). *If Π^σ is static and σ is a static type, then $\Pi^\sigma \longrightarrow_\wedge \Upsilon^\sigma \iff \Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$.*

Proof. By structural induction on evaluation contexts, for both directions, where the base case is by induction on the length of the reductions using $\longrightarrow_\wedge$ and $\longrightarrow_{\wedge CC}$. □

Another fundamental property is that of type safety, which comprises the two theorems below.

Theorem 6.3 (Type Preservation). *If $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ and $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$ then $\emptyset \vdash_{\wedge CC} \Upsilon^\sigma : \sigma$.*

Proof. By structural induction on evaluation contexts, where the base case is by induction on the length of the reduction using $\longrightarrow_{\wedge CC}$. □

Theorem 6.4 (Progress). *If $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ then either Π^σ is a parallel value or $\exists \Upsilon^\sigma$ such that $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$.*

Proof. By induction on the length of the derivation tree of $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$. □

Gradual Guarantee is a useful property, as it ensures that evolving type annotations, from less precise to more precise types and vice-versa, doesn't cause unexpected behaviour. In particular, taking a well-typed program and making its type annotations less precise, i.e. introducing more dynamic type annotations, doesn't change the behaviour of the program, as it still reduces to a value. On the other hand, making type annotations more precise either causes the program to evaluate the same, or it might cause a run-time type error. The proof of Gradual Guarantee is arguably the most technically challenging proof in this paper, requiring four lemmas that handle specific cases:

Lemma 6.5 (Extra Cast on the Left). *If $\emptyset \vdash_{\wedge CC} v_1^{\tau_1} : \tau_1$, $\emptyset \vdash_{\wedge CC} v_2^{\tau_2} : \tau_2$, $v_2^{\tau_2} \sqsubseteq v_1^{\tau_1}$ and $\tau_2 \sqsubseteq \tau_1$ and $\tau_3 \sqsubseteq \tau_1$ then $v_2^{\tau_2} : \tau_2 \Rightarrow \tau_3 \longrightarrow_{\wedge CC}^* v_3^{\tau_3}$ and $v_3^{\tau_3} \sqsubseteq v_1^{\tau_1}$.*

Proof. By case analysis on τ_2 and τ_3: □

Lemma 6.6 (Catchup to Value on the Right). *If $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} M^\rho : \rho$ and $M^\rho \sqsubseteq v^\tau$ then $M^\rho \longrightarrow_{\wedge CC}^* v'^\rho$ and $v'^\rho \sqsubseteq v^\tau$.*

Proof. By induction on the length of the derivation tree of $M^\rho \sqsubseteq v^\tau$. □

Lemma 6.7 (Simulation of Function Application). *Assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} v'^{\upsilon \to \rho} : \upsilon \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^\upsilon : \upsilon$ and $\upsilon \to \rho \sqsubseteq \sigma \to \tau$. If $v'^{\upsilon \to \rho} \sqsubseteq \lambda x : \sigma \,.\, M^\tau$ and $\pi'^\upsilon \sqsubseteq \pi^\sigma$ then $v'^{\upsilon \to \rho}\ \pi'^\upsilon \longrightarrow_{\wedge CC}^* M'^\rho$, $M'^\rho \sqsubseteq [c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\, M^\tau$ and $\emptyset \vdash_{\wedge CC} M'^\rho : \rho$.*

Proof. By induction on the length of the derivation tree of $v'^{\upsilon \to \rho} \sqsubseteq \lambda x : \sigma \,.\, M^\tau$. □

[EC-Identity]	$v^{\tau} : \tau \Rightarrow \tau$	$\longrightarrow_{\wedge CC}$	v^{τ}	
[EC-Application]	$(v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \upsilon \to \rho)\ \pi^{\upsilon}$	$\longrightarrow_{\wedge CC}$	$(v^{\sigma\to\tau}\ (\pi^{\upsilon} : \upsilon \Rightarrow_{\wedge} \sigma)) : \tau \Rightarrow \rho$	if $\pi^{\upsilon} \neq wrong^{\upsilon}$
[EC-Succeed]	$v^{G} : G \Rightarrow Dyn : Dyn \Rightarrow G$	$\longrightarrow_{\wedge CC}$	v^{G}	
[EC-Fail]	$v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2$	$\longrightarrow_{\wedge CC}$	$wrong^{G_2}$	if $G_1 \neq G_2$
[EC-Ground]	$v^{\tau} : \tau \Rightarrow Dyn$	$\longrightarrow_{\wedge CC}$	$v^{\tau} : \tau \Rightarrow G : G \Rightarrow Dyn$	if $\tau \neq Dyn, \tau \neq G$ and $\tau \sim G$
[EC-Expand]	$v^{Dyn} : Dyn \Rightarrow \tau$	$\longrightarrow_{\wedge CC}$	$v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$	if $\tau \neq Dyn, \tau \neq G$ and $\tau \sim G$

Figure 6: Cast Handler Reduction Rules ($\Pi^{\sigma} \longrightarrow_{\wedge CC} \Upsilon^{\sigma}$)

$$\text{[E-Beta]}\ \frac{for\ all\ c_i^{\rho}(x)\ in\ M^{\tau}}{(\lambda x : \sigma\ .\ M^{\tau})\ \pi^{\sigma} \longrightarrow_{\wedge} [c_i^{\rho}(x) \mapsto \langle \pi^{\sigma} \rangle_i^{\rho}]\ M^{\tau}} \qquad \text{[E-Add]}\ \frac{k_3 \text{ is the sum of } k_1 \text{ and } k_2}{k_1^{Int} + k_2^{Int} \longrightarrow_{\wedge} k_3^{Int}}$$

$$\text{[E-Ctx]}\ \frac{\Pi^{\sigma} \longrightarrow_{\wedge} \Upsilon^{\sigma}}{E[\Pi^{\sigma}] \longrightarrow_{\wedge} E[\Upsilon^{\sigma}]} \qquad \text{[E-Par]}\ \frac{M_1^{\tau_1} \longrightarrow_{\wedge} N_1^{\tau_1}\ \ldots\ M_n^{\tau_n} \longrightarrow_{\wedge} N_n^{\tau_n} \qquad n > 1}{M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}}$$

Figure 7: Static Operational Semantics ($\Pi^{\sigma} \longrightarrow_{\wedge} \Upsilon^{\sigma}$)

$$\text{[E-Beta]}\ \frac{\pi^{\sigma} \neq wrong^{\sigma} \qquad for\ all\ c_i^{\rho}(x)\ in\ M^{\tau}}{(\lambda x : \sigma\ .\ M^{\tau})\ \pi^{\sigma} \longrightarrow_{\wedge CC} [c_i^{\rho}(x) \mapsto \langle \pi^{\sigma} \rangle_i^{\rho}]\ M^{\tau}} \qquad \text{[E-Ctx]}\ \frac{\Pi^{\sigma} \longrightarrow_{\wedge CC} \Upsilon^{\sigma}}{E[\Pi^{\sigma}] \longrightarrow_{\wedge CC} E[\Upsilon^{\sigma}]} \qquad \text{[E-Wrong]}\ \frac{\emptyset \vdash_{\wedge CC} E[wrong^{\sigma}] : \tau}{E[wrong^{\sigma}] \longrightarrow_{\wedge CC} wrong^{\tau}}$$

$$\text{[E-Add]}\ \frac{k_3 \text{ is the sum of } k_1 \text{ and } k_2}{k_1^{Int} + k_2^{Int} \longrightarrow_{\wedge CC} k_3^{Int}} \qquad \text{[E-Push]}\ \frac{\sigma = \tau_1 \wedge \ldots \wedge \tau_n \qquad \exists i\ .\ r_i^{\tau_i} = wrong^{\tau_i}}{r_1^{\tau_1} \mid \ldots \mid r_n^{\tau_n} \longrightarrow_{\wedge CC} wrong^{\sigma}}$$

$$\text{[E-Par]}\ \frac{\forall i\ .\ \text{either } M_i^{\tau_i} \text{ is a result and } M_i^{\tau_i} = N_i^{\tau_i} \text{ or } M_i^{\tau_i} \longrightarrow_{\wedge CC} N_i^{\tau_i} \qquad \exists i\ .\ M_i^{\tau_i} \text{ is not a result} \qquad n > 1}{M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}}$$

Figure 8: Cast Calculus Operational Semantics ($\Pi^{\sigma} \longrightarrow_{\wedge CC} \Upsilon^{\sigma}$)

Lemma 6.8 (Simulation of Unwrapping). *Assume $\emptyset \vdash_{\wedge CC} v^{\sigma\to\tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{\upsilon\to\rho} : \upsilon \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^{\upsilon} : \upsilon$ and $\upsilon \to \rho \sqsubseteq \sigma \to \tau$. If $v'^{\upsilon\to\rho} \sqsubseteq v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ and $\pi'^{\upsilon} \sqsubseteq \pi^{\sigma'}$ then $v'^{\upsilon\to\rho}\ \pi'^{\upsilon} \longrightarrow^*_{\wedge CC} M^{\rho}$ and $M^{\rho} \sqsubseteq v^{\sigma\to\tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma) : \tau \Rightarrow \tau'$.*

Proof. By induction on the length of the derivation tree of $v'^{\upsilon\to\rho} \sqsubseteq v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$. □

Lemma 6.9 (Simulation of More Precise Programs). *For all $\Upsilon_1^{\upsilon} \sqsubseteq \Pi_1^{\sigma}$ such that $\emptyset \vdash_{\wedge CC} \Pi_1^{\sigma} : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon_1^{\upsilon} : \upsilon$, if $\Pi_1^{\sigma} \longrightarrow_{\wedge CC} \Pi_2^{\sigma}$ then $\Upsilon_1^{\upsilon} \longrightarrow^*_{\wedge CC} \Upsilon_2^{\upsilon}$ and $\Upsilon_2^{\upsilon} \sqsubseteq \Pi_2^{\sigma}$.*

Proof. By induction on the length of the derivation tree of $\Upsilon_1^{\upsilon} \sqsubseteq \Pi_1^{\sigma}$, followed by case analysis on $\Pi_1^{\sigma} \longrightarrow_{\wedge CC} \Pi_2^{\sigma}$, and using lemmas 6.5, 6.6, 6.7 and 6.8, and theorems 6.3 and 6.4. □

Theorem 6.10 (Gradual Guarantee). *For all $\Upsilon^{\upsilon} \sqsubseteq \Pi^{\sigma}$ such that $\emptyset \vdash_{\wedge CC} \Pi^{\sigma} : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon^{\upsilon} : \upsilon$, and assuming $\pi_1^{\sigma} \neq wrong^{\sigma}$ and $\pi_2^{\upsilon} \neq wrong^{\upsilon}$:*

(1) *if $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} \pi_1^{\sigma}$ then $\Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} \pi_2^{\upsilon}$ and $\pi_2^{\upsilon} \sqsubseteq \pi_1^{\sigma}$.*
if Π^{σ} diverges then Υ^{υ} diverges.

(2) *if $\Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} \pi_2^{\upsilon}$ then either $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} \pi_1^{\sigma}$ and $\pi_2^{\upsilon} \sqsubseteq \pi_1^{\sigma}$, or $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} wrong^{\sigma}$.*
if Υ^{υ} diverges then Π^{σ} diverges or $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} wrong^{\sigma}$.

Proof. The proof for part 1 follows by induction on the length of the reduction sequence using lemma 6.9; for the diverging case, it follows by simulation (lemma 6.9) on the infinite reduction sequence. Part 2 is a corollary of part 1. □

In [9], the reduction of terms is synchronized between components of parallel terms since they are equivalent modulo α-conversion. In our language, one component may have more casts than another, or be reduced to a $wrong^{\tau}$ while the other proceeds reduction. Therefore, each component is independently reduced, as shown in rule [E-Par]. We show that, after reduction, components are all equivalent to each other, under the variant relation $\bowtie$ (definition 5.2), by showing reduction is confluent modulo $\bowtie$. Similar to the proof of Gradual Guarantee, the main lemma also depends on the following four auxiliary lemmas:

Lemma 6.11 (Extra Cast on the Right (Confluency)). *If $\emptyset \vdash_{\wedge CC} v_1^{\tau_1} : \tau_1$, $\emptyset \vdash_{\wedge CC} r_2^{\tau_2} : \tau_2$, $v_1^{\tau_1} \bowtie r_2^{\tau_2}$ then $r_2^{\tau_2} : \tau_2 \Rightarrow \tau_3 \longrightarrow^*_{\wedge CC} r_3^{\tau_3}$ and $v_1^{\tau_1} \bowtie r_3^{\tau_3}$.*

Proof. We divide this proof into 2 parts: either $r_2^{\tau_2} = wrong^{\tau_2}$; or $r_2^{\tau_2}$ is a value $v_2^{\tau_2}$, in which case we proceed by case analysis on τ_2 and τ_3. □

Lemma 6.12 (Catchup to Value on the Left (Confluency)). *If $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} N^\rho : \rho$ and $v^\tau \bowtie N^\rho$ then $N^\rho \longrightarrow^*_{\wedge CC} r^\rho$ and $v^\tau \bowtie r^\rho$.*

Proof. By induction on the length of the derivation tree of $v^\tau \bowtie N^\rho$. □

Lemma 6.13 (Simulation of Function Application (Confluency)). *Assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} v'^{v \to \rho} : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^v : v$. If $\lambda x : \sigma \,.\, M^\tau \bowtie v'^{v \to \rho}$ and $\pi^\sigma \bowtie \pi'^v$ then $v'^{v \to \rho}\ \pi'^v \longrightarrow^*_{\wedge CC} M'^\rho$ and $[c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\ M^\tau \bowtie M'^\rho$.*

Proof. By induction on the length of the derivation tree of $\lambda x : \sigma \,.\, M^\tau \bowtie v'^{v \to \rho}$. □

Lemma 6.14 (Simulation of Unwrapping (Confluency)). *Assume $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{v \to \rho} : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^v : v$. If $v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v \to \rho}$ and $\pi^{\sigma'} \bowtie \pi'^v$ then $v'^{v \to \rho}\ \pi'^v \longrightarrow^*_{\wedge CC} M^\rho$ and $v^{\sigma \to \tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma) : \tau \Rightarrow \tau' \bowtie M^\rho$.*

Proof. By induction on the length of the derivation tree of $v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v \to \rho}$. □

Lemma 6.15 (Simulation of Variant Programs). *For all $\Pi_1^\sigma \bowtie \Upsilon_1^v$ such that $\emptyset \vdash_{\wedge CC} \Pi_1^\sigma : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon_1^v : v$, if $\Pi_1^\sigma \longrightarrow_{\wedge CC} \Pi_2^\sigma$ then there exists a Υ_2^v such that $\Upsilon_1^v \longrightarrow^*_{\wedge CC} \Upsilon_2^v$ and $\Pi_2^\sigma \bowtie \Upsilon_2^v$.*

Proof. Proof by induction on the length of the derivation tree of $\Pi_1^\sigma \bowtie \Upsilon_1^v$ followed by case analysis on $\Pi_1^\sigma \longrightarrow_{\wedge CC} \Pi_2^\sigma$, and using lemmas 6.11, 6.12, 6.13 and 6.14, and theorems 6.3 and 6.4. □

Theorem 6.16 (Confluency of Operational Semantics). *For all $\Pi^\sigma \bowtie \Upsilon^v$ such that $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon^v : v$, and assuming $\pi_1^\sigma \neq wrong^\sigma$, if $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$ then $\Upsilon^v \longrightarrow^*_{\wedge CC} \pi_2^v$ and $\pi_1^\sigma \bowtie \pi_2^v$.*

Proof. By induction on the length of the reduction sequence using lemma 6.15. □

Example 6.17. Finishing the example presented in subsections 4.2 and 5.2, we start with the compiled expression:

$$\begin{aligned}
&((\lambda x : Dyn \wedge Dyn \,.\, (c_1^{Dyn}(x) : Dyn \Rightarrow Dyn^2)\\
&\qquad\qquad (c_2^{Dyn}(x) : Dyn \Rightarrow Dyn))\\
&\qquad : Dyn \wedge Dyn \to Dyn \Rightarrow Dyn \wedge Dyn \to Dyn)\\
&((\lambda y : I^2 \,.\, c_1^{I^2}(y)) : I^4 \Rightarrow Dyn \mid (\lambda z : Int \,.\, c_1^{Int}(z)) : I^2 \Rightarrow Dyn)
\end{aligned}$$

First, we get rid of the identity casts of the function with rule [EC-Identity], and then we expand the casts of the arguments via rule [EC-Ground].

$$\begin{aligned}
&((\lambda x : Dyn \wedge Dyn \,.\, (c_1^{Dyn}(x) : Dyn \Rightarrow Dyn^2)\\
&\qquad\qquad (c_2^{Dyn}(x) : Dyn \Rightarrow Dyn))\\
&((\lambda y : I^2 \,.\, c_1^{I^2}(y)) : I^4 \Rightarrow Dyn^2 : Dyn^2 \Rightarrow Dyn \mid\\
&(\lambda z : Int \,.\, c_1^{Int}(z)) : I^2 \Rightarrow Dyn^2 : Dyn^2 \Rightarrow Dyn)
\end{aligned}$$

Since both function and arguments are values, we can proceed by β-reduction, [E-Beta]. Placing the arguments into the body of the function leads to new casts, which are then reduced with rules [EC-Succeed] and [EC-Identity].

$$\begin{aligned}
&((\lambda y : I^2 \,.\, c_1^{I^2}(y)) : I^4 \Rightarrow Dyn^2)\\
&((\lambda z : Int \,.\, c_1^{Int}(z)) : I^2 \Rightarrow Dyn^2 : Dyn^2 \Rightarrow Dyn)
\end{aligned}$$

By rule [EC-Application], the cast in the function is wrapped around the argument and the application. This leads to new casts that must be reduced until a value is reached, via [EC-Expand] and [EC-Succeed].

$$\begin{aligned}
&((\lambda y : I^2 \,.\, c_1^{I^2}(y))\\
&((\lambda z : Int \,.\, c_1^{Int}(z)) : I^2 \Rightarrow Dyn^2 : Dyn^2 \Rightarrow I^2)) : I^2 \Rightarrow Dyn
\end{aligned}$$

Finally, we apply the β-reduction rule [E-Beta], and then normalize the casts with rule [EC-Ground].

$$\begin{aligned}
&(\lambda z : Int \,.\, c_1^{Int}(z)) : I^2 \Rightarrow Dyn^2 :\\
&Dyn^2 \Rightarrow I^2 : I^2 \Rightarrow Dyn^2 : Dyn^2 \Rightarrow Dyn
\end{aligned}$$

7 CONCLUSION AND FUTURE WORK

In this paper we present a new gradual intersection typed calculus, where dynamic annotations delay type-checking until the evaluation phase. We are now working on a type inference algorithm to automatically infer the static type information used in our calculus. We plan to accomplish this by drawing inspiration from [26] and our previous work in [5]. We also want to enhance the language with blame tracking [2], a feature we have so far disregarded.

ACKNOWLEDGMENTS

This work was partially financially supported by the portuguese Fundação para a Ciência e a Tecnologia, under the PhD grant number SFRH/BD/145183/2019 and by Base Funding - UIDB/00027/2020 of the Artificial Intelligence and Computer Science Laboratory – LIACC - funded by national funds through the FCT/MCTES (PIDDAC).

REFERENCES

[1] Beniamino Accattoli, Stéphane Graham-Lengrand, and Delia Kesner. 2018. Tight typings and split bounds. *Proc. ACM Program. Lang.* 2, ICFP (2018), 94:1–94:30.

[2] Amal Ahmed, Robert Bruce Findler, Jeremy G. Siek, and Philip Wadler. 2011. Blame for All. In *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (Austin, Texas, USA) *(POPL '11)*. Association for Computing Machinery, New York, NY, USA, 201–214. https://doi.org/10.1145/1926385.1926409

[3] Sandra Alves, Delia Kesner, and Daniel Ventura. 2019. A Quantitative Understanding of Pattern Matching. In *25th International Conference on Types for Proofs and Programs, TYPES 2019 (LIPIcs, Vol. 175)*. 3:1–3:36.

[4] Pedro Ângelo and Mário Florido. 2018. Gradual Intersection Types. In *Ninth Workshop on Intersection Types and Related Systems, ITRS 2018, Oxford, U.K., 8 July 2018*. https://pedroangelo.github.io/gradual-intersection-types.pdf

[5] Pedro Ângelo and Mário Florido. 2020. Type Inference for Rank 2 Gradual Intersection Types. In *Trends in Functional Programming*, William J. Bowman and Ronald Garcia (Eds.). Springer International Publishing, Cham, 84–120. https://doi.org/10.1007/978-3-030-47147-7_5

[6] Steffen van Bakel. 1996. Rank 2 Intersection Type Assignment in Term Rewriting. *Fundam. Inf.* 26, 2 (May 1996), 141–166.

[7] Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. 1983. A filter lambda model and the completeness of type assignment. *Journal of Symbolic Logic* 48, 4 (1983), 931–940. https://doi.org/10.2307/2273659

[8] Lorenzo Bettini, Viviana Bono, Mariangiola Dezani-Ciancaglini, Paola Giannini, and Betti Venneri. 2018. Java & Lambda: a Featherweight Story. *Logical Methods in Computer Science* Volume 14, Issue 3 (Sept. 2018). https://doi.org/10.23638/LMCS-14(3:17)2018

[9] Viviana Bono, Betti Venneri, and Lorenzo Bettini. 2008. A Typed Lambda Calculus with Intersection Types. *Theor. Comput. Sci.* 398, 1–3 (May 2008), 95–113. https://doi.org/10.1016/j.tcs.2008.01.046

[10] Antonio Bucciarelli, Delia Kesner, and Daniel Ventura. 2017. Non-idempotent intersection types for the Lambda-Calculus. *Log. J. IGPL* 25, 4 (2017), 431–464.

[11] Sébastien Carlier and J.B. Wells. 2005. Expansion: the Crucial Mechanism for Type Inference with Intersection Types: A Survey and Explanation. *Electronic Notes in Theoretical Computer Science* 136 (2005), 173 – 202. https://doi.org/10.1016/j.entcs.2005.03.026 Proceedings of the Third International Workshop on Intersection Types and Related Systems (ITRS 2004).

[12] Giuseppe Castagna and Victor Lanvin. 2017. Gradual Typing with Union and Intersection Types. *Proc. ACM Program. Lang.* 1, ICFP, Article 41 (Aug. 2017), 28 pages. https://doi.org/10.1145/3110285

[13] Giuseppe Castagna, Victor Lanvin, Tommaso Petrucciani, and Jeremy G. Siek. 2019. Gradual Typing: A New Perspective. *Proc. ACM Program. Lang.* 3, POPL, Article 16 (Jan. 2019), 32 pages. https://doi.org/10.1145/3290329

[14] Avik Chaudhuri. 2016. Flow: Abstract Interpretation of JavaScript for Type Checking and Beyond. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security* (Vienna, Austria) *(PLAS '16)*. Association for Computing Machinery, New York, NY, USA, 1. https://doi.org/10.1145/2993600.2996280

[15] Matteo Cimini and Jeremy G. Siek. 2016. The Gradualizer: A Methodology and Algorithm for Generating Gradual Type Systems. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (St. Petersburg, FL, USA) *(POPL '16)*. ACM, New York, NY, USA, 443–455. https://doi.org/10.1145/2837614.2837632

[16] Matteo Cimini and Jeremy G. Siek. 2017. Automatically Generating the Dynamic Semantics of Gradually Typed Languages. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages* (Paris, France) *(POPL 2017)*. ACM, New York, NY, USA, 789–803. https://doi.org/10.1145/3009837.3009863

[17] M. Coppo and M. Dezani-Ciancaglini. 1980. An extension of the basic functionality theory for the λ-calculus. *Notre Dame Journal of Formal Logic* 21, 4 (10 1980), 685–693. https://doi.org/10.1305/ndjfl/1093883253

[18] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Patrick Sallé. 1979. Functional Characterization of Some Semantic Equalities inside Lambda-Calculus. In *Automata, Languages and Programming, 6th Colloquium, July 16-20, 1979 (Lecture Notes in Computer Science, Vol. 71)*. Springer, 133–146.

[19] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. 1981. Functional Characters of Solvable Terms. *Mathematical Logic Quarterly* 27, 2-6 (1981), 45–58. https://doi.org/10.1002/malq.19810270205 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/malq.19810270205

[20] Vincent Cremet, François Garillot, Sergueï Lenglet, and Martin Odersky. 2006. A Core Calculus for Scala Type Checking. In *Mathematical Foundations of Computer Science 2006*, Rastislav Královič and Paweł Urzyczyn (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–23.

[21] Ferruccio Damiani. 2003. Rank 2 Intersection Types for Local Definitions and Conditional Expressions. *ACM Trans. Program. Lang. Syst.* 25, 4 (July 2003), 401–451. https://doi.org/10.1145/778559.778560

[22] Mariangiola Dezani-Ciancaglini, Paola Giannini, and Betti Venneri. 2019. *Intersection Types in Java: Back to the Future.* Springer International Publishing, Cham, 68–86. https://doi.org/10.1007/978-3-030-22348-9_6

[23] Ronald Garcia and Matteo Cimini. 2015. Principal Type Schemes for Gradual Programs. In *Proceedings of the 42Nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (Mumbai, India) *(POPL '15)*. ACM, New York, NY, USA, 303–315. https://doi.org/10.1145/2676726.2676992

[24] Ronald Garcia, Alison M. Clark, and Éric Tanter. 2016. Abstracting Gradual Typing. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (St. Petersburg, FL, USA) *(POPL '16)*. ACM, New York, NY, USA, 429–442. https://doi.org/10.1145/2837614.2837670

[25] T. Jim. 1995. *Rank 2 Type Systems and Recursive Definitions.* Technical Report. Cambridge, MA, USA.

[26] Trevor Jim. 1996. What Are Principal Typings and What Are They Good for?. In *Proceedings of the 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (St. Petersburg Beach, Florida, USA) *(POPL '96)*. ACM, New York, NY, USA, 42–53. https://doi.org/10.1145/237721.237728

[27] Matthias Keil and Peter Thiemann. 2015. Blame Assignment for Higher-Order Contracts with Intersection and Union. In *Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming* (Vancouver, BC, Canada) *(ICFP 2015)*. Association for Computing Machinery, New York, NY, USA, 375–386. https://doi.org/10.1145/2784731.2784737

[28] Delia Kesner and Pierre Vial. 2020. Non-idempotent types for classical calculi in natural deduction style. *Log. Methods Comput. Sci.* 16, 1 (2020).

[29] A.J. Kfoury and J.B. Wells. 2004. Principality and type inference for intersection types using expansion variables. *Theoretical Computer Science* 311, 1 (2004), 1 – 70. https://doi.org/10.1016/j.tcs.2003.10.032

[30] A. J. Kfoury and J. B. Wells. 1999. Principality and Decidable Type Inference for Finite-rank Intersection Types. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (San Antonio, Texas, USA) *(POPL '99)*. ACM, New York, NY, USA, 161–174. https://doi.org/10.1145/292540.292556

[31] Daniel Leivant. 1983. Polymorphic Type Inference. In *Proceedings of the 10th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages* (Austin, Texas) *(POPL '83)*. Association for Computing Machinery, New York, NY, USA, 88–98. https://doi.org/10.1145/567067.567077

[32] Luigi Liquori and Simona Ronchi Della Rocca. 2007. Intersection-types à la Church. *Information and Computation* 205, 9 (2007), 1371 – 1386. https://doi.org/10.1016/j.ic.2007.03.005

[33] Luigi Liquori and Claude Stolze. 2019. The Delta-calculus: Syntax and Types. In *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 131)*, Herman Geuvers (Ed.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 28:1–28:20. https://doi.org/10.4230/LIPIcs.FSCD.2019.28

[34] Yuki Nishida and Atsushi Igarashi. 2019. Manifest Contracts with Intersection Types. In *Programming Languages and Systems - 17th Asian Symposium, APLAS 2019, Nusa Dua, Bali, Indonesia, December 1-4, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11893)*, Anthony Widjaja Lin (Ed.). Springer, 33–52. https://doi.org/10.1007/978-3-030-34175-6_3

[35] G. Pottinger. 1980. A Type Assignment for the Strongly Normalizable Lambda-Terms. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. Hindley and J. Seldin (Eds.). Academic Press, 561–577.

[36] John C. Reynolds. 1997. *Design of the Programming Language Forsythe.* Birkhäuser Boston, Boston, MA, 173–233. https://doi.org/10.1007/978-1-4612-4118-8_9

[37] Simona Ronchi Della Rocca. 2003. Intersection Typed λ-calculus. *Electronic Notes in Theoretical Computer Science* 70, 1 (2003), 163 – 181. https://doi.org/10.1016/S1571-0661(04)80496-1 ITRS '02, Intersection Types and Related Systems (FLoC Satellite Event).

[38] Jeremy G Siek and Walid Taha. 2006. Gradual typing for functional languages. In *Scheme and Functional Programming Workshop*, Vol. 6. 81–92.

[39] Jeremy G. Siek and Manish Vachharajani. 2008. Gradual Typing with Unification-based Inference. In *Proceedings of the 2008 Symposium on Dynamic Languages* (Paphos, Cyprus) *(DLS '08)*. ACM, New York, NY, USA, Article 7, 12 pages. https://doi.org/10.1145/1408681.1408688

[40] Jeremy G. Siek, Michael M. Vitousek, Matteo Cimini, and John Tang Boyland. 2015. Refined Criteria for Gradual Typing. In *1st Summit on Advances in Programming Languages (SNAPL 2015) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 32)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 274–293. https://doi.org/10.4230/LIPIcs.SNAPL.2015.274

[41] Stephanus Johannes Van Bakel. 1993. *Intersection type disciplines in lambda calculus and applicative term rewriting systems.* Amsterdam: Mathematisch Centrum.

[42] Panagiotis Vekris, Benjamin Cosman, and Ranjit Jhala. 2016. Refinement Types for TypeScript. In *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation* (Santa Barbara, CA, USA) *(PLDI '16)*. Association for Computing Machinery, New York, NY, USA, 310–325. https://doi.org/10.1145/2908080.2908110

[43] Joe B. Wells and Christian Haack. 2002. Branching Types. In *Proceedings of the 11th European Symposium on Programming Languages and Systems (ESOP '02)*. Springer-Verlag, London, UK, UK, 115–132. http://dl.acm.org/citation.cfm?id=645396.651968

[44] Jack Williams, J. Garrett Morris, and Philip Wadler. 2018. The Root Cause of Blame: Contracts for Intersection and Union Types. *Proc. ACM Program. Lang.* 2, OOPSLA, Article 134 (Oct. 2018), 29 pages. https://doi.org/10.1145/3276504

A PROOFS (TYPE SYSTEM)

In this section we present the full proofs for all the properties in section 4:

- Lemma 4.13 (Inversion Lemma) in A;
- Theorem 4.14 (Conservative Extension of Operational Semantics) in A;
- Theorem 4.15 (Monotonicity w.r.t. Precision) in A.

Proposition 4.6 (Monotonicity of $\Gamma_1 \wedge \Gamma_2$ w.r.t. Precision). *If $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_2' \sqsubseteq \Gamma_2$ then $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$.*

Proof. For all $x : \sigma \in \Gamma_1 \wedge \Gamma_2$, there are 3 possibilities:

- $x : \sigma_1 \in \Gamma_1$ and $x : \sigma_2 \in \Gamma_2$. Since $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_2' \sqsubseteq \Gamma_2$ then by definition 4.4, $x : \upsilon_1 \in \Gamma_1'$ and $\upsilon_1 \sqsubseteq \sigma_1$, and $x : \upsilon_2 \in \Gamma_2'$ and $\upsilon_2 \sqsubseteq \sigma_2$. By definition 4.3, we have that $\upsilon_1 \wedge \upsilon_2 \sqsubseteq \sigma_1 \wedge \sigma_2$. By definition 3.4, we have that $x : \upsilon_1 \wedge \upsilon_2 \in \Gamma_1' \wedge \Gamma_2'$, and $x : \sigma_1 \wedge \sigma_2 \in \Gamma_1 \wedge \Gamma_2$. Therefore, $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$.
- $x : \sigma_1 \in \Gamma_1$ and $\neg\exists \sigma_2 \,.\, x : \sigma_2 \in \Gamma_2$. Since $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_2' \sqsubseteq \Gamma_2$ then by definition 4.4, $x : \upsilon_1 \in \Gamma_1'$ and $\upsilon_1 \sqsubseteq \sigma_1$, and $\neg\exists \upsilon_2 \,.\, x : \upsilon_2 \in \Gamma_2'$. By definition 3.4, we have that $x : \upsilon_1 \in \Gamma_1' \wedge \Gamma_2'$, and $x : \sigma_1 \in \Gamma_1 \wedge \Gamma_2$. Therefore, $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$.
- $\neg\exists \sigma_1 \,.\, x : \sigma_1 \in \Gamma_1$ and $x : \sigma_2 \in \Gamma_2$. Since $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_2' \sqsubseteq \Gamma_2$ then by definition 4.4, $\neg\exists \upsilon_1 \,.\, x : \upsilon_1 \in \Gamma_1'$, and $x : \upsilon_2 \in \Gamma_2'$ and $\upsilon_2 \sqsubseteq \sigma_2$. By definition 3.4, we have that $x : \upsilon_2 \in \Gamma_1' \wedge \Gamma_2'$, and $x : \sigma_2 \in \Gamma_1 \wedge \Gamma_2$. Therefore, $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$.

□

Proposition A.1. *If $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} \Pi^\sigma : \sigma$, and $x \in fv(\Pi^\sigma)$, then the number of free occurrences of x in Π^σ equals n (the number of instances bound to x in $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n$), and these occurrences are typed with $\tau_1, \ldots, \tau_n$ (instances bound to x in $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n$), considering an order from left to right.*

Proof. We proceed by induction on Π^σ.

Base case:

- $\Pi^\sigma = k^B$. According to rule [T-Con], we have $\emptyset \vdash_{\wedge G} k^B : B$, which is vacuously true.
- $\Pi^\sigma = c_0^\tau(x)$. According to rule [T-Var], we have that $x : \tau \vdash_{\wedge G} c_0^\tau(x) : \tau$.

Induction step:

- $\Pi^\sigma = \lambda y : \upsilon \,.\, N^{\rho'}$. If $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} \lambda y : \upsilon \,.\, N^{\rho'} : \upsilon \to \rho'$, then by rule [T-AbsI] (resp. [T-AbsK]), we have that $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n, y : \upsilon \vdash_{\wedge G} N^{\rho'} : \rho'$ (resp. $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} N^{\rho'} : \rho'$). By the induction hypothesis, we have that the number of free occurrences of x in $N^{\rho'}$ equals n, and these occurrences are typed with $\tau_1, \ldots, \tau_n$, considering an order from left to right. Therefore, the same holds for $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} \lambda y : \upsilon \,.\, N^{\rho'} : \upsilon \to \rho'$.
- $\Pi^\sigma = N^{\rho'}\ \Pi^{\upsilon'}$. If $\Gamma_1 \wedge \Gamma_2, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} N^{\rho'}\ \Pi^{\upsilon'} : \rho$, then by rule [T-App], we have that $\Gamma_1' \vdash_{\wedge G} N^{\rho'} : \rho', \rho' \rhd \upsilon \to \rho$, $\Gamma_2' \vdash_{\wedge G} \Pi^{\upsilon'} : \upsilon'$ and $\upsilon' \sim \upsilon$, where $\Gamma_1 \wedge \Gamma_2, x : \tau_1 \wedge \ldots \wedge \tau_n = \Gamma_1' \wedge \Gamma_2'$. Therefore, by the induction hypothesis, and definition 3.4, the number of free occurrences of x in $N^{\rho'}$ (resp. $\Pi^{\upsilon'}$) equals the number of instances bound to x in Γ_1' (resp. Γ_2'), and these occurrences are typed with the instances bound to x in Γ_1' (resp. Γ_2'), considering an order from left to right. By definition 3.4 and rule [T-App], the same property holds for $\Gamma_1 \wedge \Gamma_2, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} N^{\rho'}\ \Pi^{\upsilon'} : \rho$.
- $\Pi^\sigma = N_1^\tau + N_2^\tau$. If $\Gamma_1 \wedge \Gamma_2, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} N_1^\tau + N_2^\rho : Int$, then by rule [T-Add], we have that $\Gamma_1' \vdash_{\wedge G} N^\tau : \tau$, $\tau \rhd Int$, $\Gamma_2' \vdash_{\wedge G} N_2^\rho : \rho$ and $\rho \rhd Int$, where $\Gamma_1 \wedge \Gamma_2, x : \tau_1 \wedge \ldots \wedge \tau_n = \Gamma_1' \wedge \Gamma_2'$. Therefore, by the induction hypothesis, and definition 3.4, the number of free occurrences of x in N_1^τ (resp. N_2^ρ) equals the number of instances bound to x in Γ_1' (resp. Γ_2'), and these occurrences are typed with the instances bound to x in Γ_1' (resp. Γ_2'), considering an order from left to right. By definition 3.4 and rule [T-Add], the same property holds for $\Gamma_1 \wedge \Gamma_2, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} N_1^\tau + N_2^\rho : Int$.
- $\Pi^\sigma = M_1^{\rho_1} \mid \ldots \mid M_n^{\rho_n}$. If $\Gamma_1 \wedge \ldots \wedge \Gamma_n, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} M_1^{\rho_1} \mid \ldots \mid M_n^{\rho_n} : \rho_1 \wedge \ldots \wedge \rho_n$, then by rule [T-Par], we have that $\Gamma_1' \vdash_{\wedge G} M_1^{\rho_1} : \rho_1$ and ... and $\Gamma_n' \vdash_{\wedge G} M_n^{\rho_n} : \rho_n$, where $\Gamma_1 \wedge \ldots \wedge \Gamma_n, x : \tau_1 \wedge \ldots \wedge \tau_n = \Gamma_1' \wedge \ldots \wedge \Gamma_n'$. Therefore, by the induction hypothesis, and definition 3.4, the number of free occurrences of x in $M_1^{\rho_1}$ and ... and $M_n^{\rho_n}$ equals the number of instances bound to x in Γ_1' and ... and Γ_n', and these occurrences are typed with the instances bound to x in Γ_1' and ... and Γ_n', considering an order from left to right. By definition 3.4 and rule [T-Par], the same property holds for $\Gamma_1 \wedge \ldots \wedge \Gamma_n, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} M_1^{\rho_1} \mid \ldots \mid M_n^{\rho_n} : \rho_1 \wedge \ldots \wedge \rho_n$.

□

Proposition 4.9. *If $\Gamma \vdash_{\wedge G} \lambda x : \tau_1 \wedge \ldots \wedge \tau_n \,.\, M^\rho : \tau_1 \wedge \ldots \wedge \tau_n \to \rho$, and $x \in fv(M^\rho)$, then the number of free occurrences of x in M^ρ equals n, and these occurrences are typed with $\tau_1, \ldots, \tau_n$, considering an order from left to right.*

Proof. If $\Gamma \vdash_{\wedge G} \lambda x : \tau_1 \wedge \ldots \wedge \tau_n \,.\, M^\rho : \tau_1 \wedge \ldots \wedge \tau_n \to \rho$, then by rule [T-AbsI], we have that $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} M^\rho : \tau_1 \wedge \ldots \wedge \tau_n \to \rho$. By proposition A.1, we have that for $\Gamma, x : \tau_1 \wedge \ldots \wedge \tau_n \vdash_{\wedge G} M^\rho : \rho$, the property holds. By rule [T-AbsI], the property holds for $\Gamma \vdash_{\wedge G} \lambda x : \tau_1 \wedge \ldots \wedge \tau_n \,.\, M^\rho : \tau_1 \wedge \ldots \wedge \tau_n \to \rho$. □

Lemma 4.13 (Inversion Lemma).

1. *Rule [T-Con]. If $\emptyset \vdash_{\wedge G} k^B : B$ then k is a constant of base type B.*
2. *Rule [T-Var]. We have that $x : \tau \vdash_{\wedge G} c_i^\tau(x) : \tau$ holds.*
3. *Rule [T-AbsI]. Assuming $x \in fv(M^\tau)$, if $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ then $\Gamma, x : \sigma \vdash_{\wedge G} M^\tau : \tau$.*
4. *Rule [T-AbsK]. Assuming $x \notin fv(M^\tau)$, if $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ then $\Gamma \vdash_{\wedge G} M^\tau : \tau$.*
5. *Rule [T-App]. If $\Gamma \vdash_{\wedge G} M^\rho\ \Pi^\upsilon : \tau$ then typing context Γ can be divided into Γ_1 and Γ_2 such that $\Gamma_1 \wedge \Gamma_2 = \Gamma$ and $\Gamma_1 \vdash_{\wedge G} M^\rho : \rho$, $\rho \rhd \sigma \to \tau$, $\Gamma_2 \vdash_{\wedge G} \Pi^\upsilon : \upsilon$ and $\upsilon \sim \sigma$.*
6. *Rule [T-Add]. If $\Gamma \vdash_{\wedge G} M^\tau + N^\rho : Int$ then typing context Γ can be divided into Γ_1 and Γ_2 such that $\Gamma_1 \wedge \Gamma_2 = \Gamma$ and $\Gamma_1 \vdash_{\wedge G} M^\tau : \tau$ and $\tau \rhd Int$ and $\Gamma_2 \vdash_{\wedge G} N^\rho : \rho$ and $\rho \rhd Int$.*
7. *Rule [T-Par]. If $\Gamma \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then typing context Γ can be divided into $\Gamma_1, \ldots, \Gamma_n$ such that $\Gamma_1 \wedge \ldots \wedge \Gamma_n = \Gamma$ and $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$ and $\bowtie (M_1^{\tau_1}, \ldots, M_n^{\tau_n})$.*

Proof. Proof is trivial. □

Theorem 4.14 (Conservative Extension of Type System). *If Π^σ is static and σ is a static type, then* $\Gamma \vdash_\wedge \Pi^\sigma : \sigma \iff \Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$.

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_\wedge \Pi^\sigma : \sigma$ and $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$ for the right and left direction of the implication, respectively.

Base cases:

- Rule [T-Con]:
  - If $\emptyset \vdash_\wedge k^B : B$ then by rule [T-Con] we have that k is a constant of base type B. Therefore, by rule [T-Con], we have that $\emptyset \vdash_{\wedge G} k^B : B$ holds.
  - If $\emptyset \vdash_{\wedge G} k^B : B$ then by rule [T-Con] we have that k is a constant of base type B. Therefore, by rule [T-Con], we have that $\emptyset \vdash_\wedge k^B : B$ holds.
- Rule [T-Var]. Both $x : \tau \vdash_\wedge c_i^\tau(x) : \tau$ and $x : \tau \vdash_{\wedge G} c_i^\tau(x) : \tau$ hold.

Induction step:

- Rule [T-AbsI]:
  - If $\Gamma \vdash_\wedge \lambda x : \sigma . M^\tau : \sigma \to \tau$ then by rule [T-AbsI] we have that $\Gamma, x : \sigma \vdash_\wedge M^\tau : \tau$ and $x \in fv(M^\tau)$ hold. By the induction hypothesis, we have that $\Gamma, x : \sigma \vdash_{\wedge G} M^\tau : \tau$ holds. By rule [T-AbsI], we then have that $\Gamma \vdash_{\wedge G} \lambda x : \sigma . M^\tau : \sigma \to \tau$ holds.
  - If $\Gamma \vdash_{\wedge G} \lambda x : \sigma . M^\tau : \sigma \to \tau$ then by rule [T-AbsI] we have that $\Gamma, x : \sigma \vdash_{\wedge G} M^\tau : \tau$ and $x \in fv(M^\tau)$ hold. By the induction hypothesis, we have that $\Gamma, x : \sigma \vdash_\wedge M^\tau : \tau$ holds. By rule [T-AbsI], we then have that $\Gamma \vdash_\wedge \lambda x : \sigma . M^\tau : \sigma \to \tau$ holds.
- Rule [T-AbsK]:
  - If $\Gamma \vdash_\wedge \lambda x : \sigma . M^\tau : \sigma \to \tau$ then by rule [T-AbsK] we have that $\Gamma \vdash_\wedge M^\tau : \tau$ and $x \notin fv(M^\tau)$ hold. By the induction hypothesis, we have that $\Gamma \vdash_{\wedge G} M^\tau : \tau$ holds. By rule [T-AbsK], we then have that $\Gamma \vdash_{\wedge G} \lambda x : \sigma . M^\tau : \sigma \to \tau$ holds.
  - If $\Gamma \vdash_{\wedge G} \lambda x : \sigma . M^\tau : \sigma \to \tau$ then by rule [T-AbsK] we have that $\Gamma, x : \sigma \vdash_{\wedge G} M^\tau : \tau$ and $x \notin fv(M^\tau)$ hold. By the induction hypothesis, we have that $\Gamma \vdash_\wedge M^\tau : \tau$ holds. By rule [T-AbsK], we then have that $\Gamma \vdash_\wedge \lambda x : \sigma . M^\tau : \sigma \to \tau$ holds.
- Rule [T-App]:
  - If $\Gamma_1 \wedge \Gamma_2 \vdash_\wedge M^{\sigma\to\tau}\, \Pi^\sigma : \tau$ then by rule [T-App] we have that $\Gamma_1 \vdash_\wedge M^{\sigma\to\tau} : \sigma \to \tau$ and $\Gamma_2 \vdash_\wedge \Pi^\sigma : \sigma$ hold. By the induction hypothesis, we have that $\Gamma_1 \vdash_{\wedge G} M^{\sigma\to\tau} : \sigma \to \tau$ and $\Gamma_2 \vdash_{\wedge G} \Pi^\sigma : \sigma$ hold. As $\sigma \to \tau \rhd \sigma \to \tau$ holds, and also as $\sigma \sim \sigma$ holds, then by rule [T-App] we have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^{\sigma\to\tau}\, \Pi^\sigma : \tau$ holds.
  - If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^\rho\, \Pi^\upsilon : \tau$ then by rule [T-App] we have that $\Gamma_1 \vdash_{\wedge G} M^\rho : \rho, \rho \rhd \sigma \to \tau, \Gamma_2 \vdash_{\wedge G} \Pi^\upsilon : \upsilon$ and $\upsilon \sim \sigma$ hold. Since ρ is a static type, then $\rho = \sigma \to \tau$. Also, since both σ and υ are static types, then $\sigma = \upsilon$. By the induction hypothesis, we have that $\Gamma_1 \vdash_\wedge M^{\sigma\to\tau} : \sigma \to \tau$ and $\Gamma_2 \vdash_\wedge \Pi^\sigma : \sigma$ holds. By rule [T-App], we have that $\Gamma_1 \wedge \Gamma_2 \vdash_\wedge M^{\sigma\to\tau}\, \Pi^\sigma : \tau$ holds.
- Rule [T-Add]:
  - If $\Gamma_1 \wedge \Gamma_2 \vdash_\wedge M^{Int} + N^{Int} : Int$ then by rule [T-Add] we have that $\Gamma_1 \vdash_\wedge M^{Int} : Int$ and $\Gamma_2 \vdash_\wedge N^{Int} : Int$ hold. By the induction hypothesis, we have that $\Gamma_1 \vdash_{\wedge G} M^{Int} : Int$ and $\Gamma_2 \vdash_{\wedge G} N^{Int} : Int$ hold. As $Int \rhd Int$ holds, then by rule [T-Add] we have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^{Int} + N^{Int} : Int$ holds.
  - If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^\tau + N^\rho : Int$ then by rule [T-App] we have that $\Gamma_1 \vdash_{\wedge G} M^\tau : \tau, \tau \rhd Int, \Gamma_2 \vdash_{\wedge G} N^\rho : \rho$ and $\rho \rhd Int$ hold. Since both τ and ρ are static types, then $\tau = Int$ and $\rho = Int$. By the induction hypothesis, we have that $\Gamma_1 \vdash_\wedge M^{Int} : Int$ and $\Gamma_2 \vdash_\wedge N^{Int} : Int$ holds. By rule [T-App], we have that $\Gamma_1 \wedge \Gamma_2 \vdash_\wedge M^{Int} + N^{Int} : Int$ holds.
- Rule [T-Par]:
  - If $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_\wedge M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then by rule [T-Par] we have that $\Gamma_1 \vdash_\wedge M_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_\wedge M_n^{\tau_n} : \tau_n$ and $\bowtie (M_1^{\tau_1}, \ldots, M_n^{\tau_n})$. By the induction hypothesis, we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$. Then, by rule [T-Par], we have that $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$.
  - If $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then by rule [T-Par] we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$ and $\bowtie (M_1^{\tau_1}, \ldots, M_n^{\tau_n})$. By the induction hypothesis, we have that $\Gamma_1 \vdash_\wedge M_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_\wedge M_n^{\tau_n} : \tau_n$. Then, by rule [T-Par], we have that $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_\wedge M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$.

□

Theorem 4.15 (Monotonicity w.r.t. Precision). *If* $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$ *and* $\Upsilon^\upsilon \sqsubseteq \Pi^\sigma$ *then* $\exists \Gamma'$ *such that* $\Gamma' \sqsubseteq \Gamma$ *and* $\Gamma' \vdash_{\wedge G} \Upsilon^\upsilon : \upsilon$ *and* $\upsilon \sqsubseteq \sigma$.

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^\sigma : \sigma$.

Base cases:

- Rule [T-Con]. If $\emptyset \vdash_{\wedge G} k^B : B$ and $k^B \sqsubseteq k^B$ then, we have that $\emptyset \vdash_{\wedge G} k^B : B$ and $B \sqsubseteq B$.
- Rule [T-Var]. If $x : \tau \vdash_{\wedge G} c_i^\tau(x) : \tau$ and $c_i^\rho(x) \sqsubseteq c_i^\tau(x)$ then by rule [P-Con], we have that $\rho \sqsubseteq \tau$. By rule [T-Var], we have that $x : \rho \vdash_{\wedge G} c_i^\rho(x) : \rho$ and $\rho \sqsubseteq \tau$.

Induction step:

- Rule [T-AbsI]. If $\Gamma \vdash_{\wedge G} \lambda x : \sigma . M^\tau : \sigma \to \tau$ and $\lambda x : \upsilon . N^\rho \sqsubseteq \lambda x : \sigma . M^\tau$, then by rule [T-AbsI], we have that $\Gamma, x : \sigma \vdash_{\wedge G} M^\tau : \tau$ and by rule [P-Abs], we have that $\upsilon \sqsubseteq \sigma$ and $N^\rho \sqsubseteq M^\tau$. By the induction hypothesis, $\exists \Gamma', x : \upsilon$ such that $\Gamma', x : \upsilon \sqsubseteq \Gamma, x : \sigma$ and $\Gamma', x : \upsilon \vdash_{\wedge G} N^\rho : \rho$ and $\rho \sqsubseteq \tau$. Therefore, by rule [T-AbsI], we have that $\Gamma' \vdash_{\wedge G} \lambda x : \upsilon . N^\rho : \upsilon \to \rho$ and by definition 4.3, we have that $\upsilon \to \rho \sqsubseteq \sigma \to \tau$.
- Rule [T-AbsK]. If $\Gamma \vdash_{\wedge G} \lambda x : \sigma . M^\tau : \sigma \to \tau$ and $\lambda x : \upsilon . N^\rho \sqsubseteq \lambda x : \sigma . M^\tau$, then by rule [T-AbsK], we have that $\Gamma \vdash_{\wedge G} M^\tau : \tau$ and by rule [P-Abs], we have that $\upsilon \sqsubseteq \sigma$ and $N^\rho \sqsubseteq M^\tau$. By the induction hypothesis, $\exists \Gamma'$ such that $\Gamma' \sqsubseteq \Gamma$ and $\Gamma' \vdash_{\wedge G} N^\rho : \rho$ and $\rho \sqsubseteq \tau$. Therefore, by rule [T-AbsK], we have that $\Gamma' \vdash_{\wedge G} \lambda x : \upsilon . N^\rho : \upsilon \to \rho$ and by definition 4.3, we have that $\upsilon \to \rho \sqsubseteq \sigma \to \tau$.
- Rule [T-App]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^\rho\, \Pi^\upsilon : \tau$ and $N^{\rho'}\, \Upsilon^{\upsilon'} \sqsubseteq M^\rho\, \Pi^\upsilon$ then by rule [T-App], we have that $\Gamma_1 \vdash_{\wedge G} M^\rho : \rho, \rho \rhd \sigma \to \tau$,

$\Gamma_2 \vdash_{\wedge G} \Pi^{\upsilon} : \upsilon$ and $\upsilon \sim \sigma$, and by rule [P-App], we have that $N^{\rho'} \sqsubseteq M^{\rho}$ and $\Upsilon^{\upsilon'} \sqsubseteq \Pi^{\upsilon}$. By the induction hypothesis, $\exists \Gamma_1'$ such that $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_1' \vdash_{\wedge G} N^{\rho'} : \rho'$ and $\rho' \sqsubseteq \rho$, and $\exists \Gamma_2'$ such that $\Gamma_2' \sqsubseteq \Gamma_2$ and $\Gamma_2' \vdash_{\wedge G} \Upsilon^{\upsilon'} : \upsilon'$ and $\upsilon' \sqsubseteq \upsilon$. Since $\rho \rhd \sigma \rightarrow \tau$ and $\rho' \sqsubseteq \rho$, then by definition 4.2, we have that $\rho' \rhd \sigma' \rightarrow \tau'$, $\sigma' \sqsubseteq \sigma$ and $\tau' \sqsubseteq \tau$. Since $\sigma \sim \upsilon$, $\sigma' \sqsubseteq \sigma$ and $\upsilon' \sqsubseteq \upsilon$, then by definition 4.1 we have that $\upsilon' \sim \sigma'$. By proposition 4.6, $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$. Therefore, by rule [T-App] we have that $\Gamma_1' \wedge \Gamma_2' \vdash_{\wedge G} N^{\rho'}\ \Upsilon^{\upsilon'} : \tau'$.
- Rule [T-Add]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M_1^{\tau_1} + M_2^{\tau_2} : \mathit{Int}$ and $N_1^{\rho_1} + N_2^{\rho_2} \sqsubseteq M_1^{\tau_1} + M_2^{\tau_2}$ then by rule [T-Add], we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$, $\tau_1 \rhd \mathit{Int}$, $\Gamma_2 \vdash_{\wedge G} M_2^{\tau_2} : \tau_2$ and $\tau_2 \rhd \mathit{Int}$, and by rule [P-Add], we have that $N_1^{\rho_1} \sqsubseteq M_1^{\tau_1}$ and $N_2^{\rho_2} \sqsubseteq M_2^{\tau_2}$. By the induction hypothesis, $\exists \Gamma_1'$ such that $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_1' \vdash_{\wedge G} N^{\rho_1} : \rho_1$ and $\rho_1 \sqsubseteq \tau_1$, and $\exists \Gamma_2'$ such that $\Gamma_2' \sqsubseteq \Gamma_2$ and $\Gamma_2' \vdash_{\wedge G} N^{\rho_2} : \rho_2$ and $\rho_2 \sqsubseteq \tau_2$. By definition 4.2 and 4.3, we have that $\rho_1 \rhd \mathit{Int}$ and $\rho_2 \rhd \mathit{Int}$. By proposition 4.6, $\Gamma_1' \wedge \Gamma_2' \sqsubseteq \Gamma_1 \wedge \Gamma_2$. Therefore, by rule [T-Add] we have that $\Gamma_1' \wedge \Gamma_2' \vdash_{\wedge G} N_1^{\rho_1} + N_2^{\rho_2} : \mathit{Int}$.
- Rule [T-Par]. If $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ and $N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n} \sqsubseteq M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$ then by rule [T-Par] we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ and $\ldots$ and $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$ and by rule [P-Par] we have that $N_1^{\rho_1} \sqsubseteq M_1^{\tau_1}$ and $\ldots$ and $N_n^{\rho_n} \sqsubseteq M_n^{\tau_n}$. By the induction hypothesis, $\exists \Gamma_1'$ such that $\Gamma_1' \sqsubseteq \Gamma_1$ and $\Gamma_1' \vdash_{\wedge G} N_1^{\rho_1} : \rho_1$ and $\rho_1 \sqsubseteq \tau_1$, and $\ldots$ and $\exists \Gamma_n'$ such that $\Gamma_n' \sqsubseteq \Gamma_n$ and $\Gamma_n' \vdash_{\wedge G} N_n^{\rho_n} : \rho_n$ and $\rho_n \sqsubseteq \tau_n$. By proposition 4.6, $\Gamma_1' \wedge \ldots \wedge \Gamma_n' \sqsubseteq \Gamma_1 \wedge \ldots \wedge \Gamma_n$. Then, by rule [T-Par] we have that $\Gamma_1' \wedge \ldots \wedge \Gamma_n' \vdash_{\wedge G} N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n} : \rho_1 \wedge \ldots \wedge \rho_n$, and by definition 4.3 we have that $\rho_1 \wedge \ldots \wedge \rho_n \sqsubseteq \tau_1 \wedge \ldots \wedge \tau_n$.

□

B PROOFS (CAST CALCULUS)

In this section we present the full proofs for all the properties in section 5:

- Theorem 5.6 (Type Preservation of Flow Marking) in B;
- Theorem 5.7 (Monotonicity of Flow Marking) in B;
- Theorem 5.9 (Type Preservation of Cast Insertion) in B;
- Theorem 5.10 (Monotonicity of Cast Insertion) in B.

Theorem 5.6 (Type Preservation of Flow Marking). *If* $\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$ *then* $\Sigma \vdash_{\wedge G} \Pi^{\sigma} \hookrightarrow \Upsilon^{\sigma}$ *and* $\Gamma \vdash_{\wedge G} \Upsilon^{\sigma} : \sigma$*, where* $\Gamma \hookrightarrow \Sigma$.

Proof. This property is easy to verify, since flow marks play no role in type checking, and changing flow marks does not change types. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$.

Base cases:

- Rule [T-Con]. By rule [T-Con], we have that $\emptyset \vdash_{\wedge G} k^B : B$ holds. By rule [M-Con], we have that $\emptyset \vdash_{\wedge G} k^B \hookrightarrow k^B$ holds. By rule [T-Con] we have that $\emptyset \vdash_{\wedge G} k^B : B$ holds.
- Rule [T-Var]. By rule [T-Var], we have that $x : \tau \vdash_{\wedge G} c_0^{\tau}(x) : \tau$ holds. By rule [M-Var], we have that $x : i \vdash_{\wedge G} c_0^{\tau}(x) \rightsquigarrow c_i^{\tau}(x)$ holds. By rule [T-Var], we have that $x : \tau \vdash_{\wedge G} c_i^{\tau}(x) : \tau$ holds.

Induction step:

- Rule [T-AbsI]. If $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} : \sigma \rightarrow \tau$ then by rule [T-AbsI], we have that $\Gamma, x : \sigma \vdash_{\wedge G} M^{\tau} : \tau$ and $x \in fv(M^{\tau})$. By the induction hypothesis, we have that $\Sigma, (x : \sigma)_{\hookrightarrow} \vdash_{\wedge G} M^{\tau} \hookrightarrow N^{\tau}$ and $\Gamma, x : \sigma \vdash_{\wedge G} N^{\tau} : \tau$ hold. By rule [M-AbsI], we have that $\Sigma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} \hookrightarrow \lambda x : \sigma \,.\, N^{\tau}$, and by rule [T-AbsI], we have that $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, N^{\tau} : \sigma \rightarrow \tau$.
- Rule [T-AbsK]. If $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} : \sigma \rightarrow \tau$ then by rule [T-AbsK], we have that $\Gamma \vdash_{\wedge G} M^{\tau} : \tau$ and $x \notin fv(M^{\tau})$. By the induction hypothesis, we have that $\Sigma \vdash_{\wedge G} M^{\tau} \hookrightarrow N^{\tau}$ and $\Gamma \vdash_{\wedge G} N^{\tau} : \tau$ hold. By rule [M-AbsK], we have that $\Sigma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} \hookrightarrow \lambda x : \sigma \,.\, N^{\tau}$, and by rule [T-AbsK], we have that $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, N^{\tau} : \sigma \rightarrow \tau$.
- Rule [T-App]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^{\rho}\ \Pi^{\upsilon} : \tau$ then by rule [T-App] we have that $\Gamma_1 \vdash_{\wedge G} M^{\rho} : \rho$, $\rho \rhd \sigma \rightarrow \tau$, $\Gamma_2 \vdash_{\wedge G} \Pi^{\upsilon} : \upsilon$ and $\upsilon \sim \sigma$ hold. By the induction hypothesis we have that $\Sigma_1 \vdash_{\wedge G} M^{\rho} \hookrightarrow N^{\rho}$ and $\Sigma_2' \vdash_{\wedge G} \Pi^{\upsilon} \hookrightarrow \Upsilon'^{\upsilon}$ hold, and also that $\Gamma_1 \vdash_{\wedge G} N^{\rho} : \rho$ and $\Gamma_2 \vdash_{\wedge G} \Upsilon'^{\upsilon} : \upsilon$ hold.

  According to the induction hypothesis, we have that $\Gamma_1 \hookrightarrow \Sigma_1$ and $\Gamma_2 \hookrightarrow \Sigma_2'$. Therefore, for each variable x in both Γ_1 and Γ_2, we have that $x : 1 \wedge \ldots \wedge n \in \Sigma_1$ and $x : 1 \wedge \ldots \wedge m \in \Sigma_2'$. We can have a flow context Σ_2, where $\Sigma_2 \backslash \{x : \overline{i_1}\} = \Sigma_2' \backslash \{x : \overline{i_2}\}$, for some $\overline{i_1}$ and $\overline{i_2}$, such that $x : n+1 \wedge \ldots \wedge n+m \in \Sigma_2$. Therefore, we have that $\Sigma_2 \vdash_{\wedge G} \Pi^{\upsilon} \hookrightarrow \Upsilon^{\upsilon}$ and $\Gamma_2 \vdash_{\wedge G} \Upsilon^{\upsilon} : \upsilon$ hold.

  By rule [M-App] we then have that $\Sigma_1 \wedge \Sigma_2 \vdash_{\wedge G} M^{\rho}\ \Pi^{\upsilon} \hookrightarrow N^{\rho}\ \Upsilon^{\upsilon}$ holds. By rule [T-App] we then have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} N^{\rho}\ \Upsilon^{\upsilon} : \tau$ holds.
- Rule [T-Add]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M_1^{\tau} + M_2^{\rho} : \mathit{Int}$ then by rule [T-Add] we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau} : \tau$, $\tau \rhd \mathit{Int}$, $\Gamma_2 \vdash_{\wedge G} M_2^{\rho} : \rho$ and $\rho \rhd \mathit{Int}$ hold. By the induction hypothesis, we have that $\Sigma_1 \vdash_{\wedge G} M_1^{\tau} \hookrightarrow N_1^{\tau}$ and $\Sigma_2' \vdash_{\wedge G} M_2^{\rho} \hookrightarrow N_2'^{\rho}$ hold, and also that $\Gamma_1 \vdash_{\wedge G} N_1^{\tau} : \tau$ and $\Gamma_2 \vdash_{\wedge G} N_2'^{\rho} : \rho$ hold.

  According to the induction hypothesis, we have that $\Gamma_1 \hookrightarrow \Sigma_1$ and $\Gamma_2 \hookrightarrow \Sigma_2'$. Therefore, for each variable x in both Γ_1 and Γ_2, we have that $x : 1 \wedge \ldots \wedge n \in \Sigma_1$ and $x : 1 \wedge \ldots \wedge m \in \Sigma_2'$. We can have a flow context Σ_2, where $\Sigma_2 \backslash \{x : \overline{i_1}\} = \Sigma_2' \backslash \{x : \overline{i_2}\}$, for some $\overline{i_1}$ and $\overline{i_2}$, such that $x : n+1 \wedge \ldots \wedge n+m \in \Sigma_2$. Therefore, we have that $\Sigma_2 \vdash_{\wedge G} \Pi^{\upsilon} \hookrightarrow \Upsilon^{\upsilon}$ and $\Gamma_2 \vdash_{\wedge G} \Upsilon^{\upsilon} : \upsilon$ hold.

  By rule [M-Add] we then have that $\Sigma_1 \wedge \Sigma_2 \vdash_{\wedge G} M_1^{\tau} + M_2^{\rho} \hookrightarrow N_1^{\tau} + N_2^{\rho}$ holds. By rule [T-Add] we then have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} N_1^{\tau} + N_2^{\rho}$ holds.
- Rule [T-Par]. If $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then by rule [T-Par] we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ and $\ldots$ and $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$ hold. By the induction hypothesis, we have that $\Sigma_1 \vdash_{\wedge G} M_1^{\tau_1} \hookrightarrow N_1^{\tau_1}$ and $\Gamma_1 \vdash_{\wedge G} N_1^{\tau_1} : \tau_1$ and $\ldots$ and $\Sigma_n' \vdash_{\wedge G} M_n^{\tau_n} \hookrightarrow N_n'^{\tau_n}$ and $\Gamma_n \vdash_{\wedge G} N_n'^{\tau_n} : \tau_n$ hold.

  We now use the same method to obtain Σ_2 from Σ_2' and $\ldots$ and Σ_n from Σ_n', and $N_2^{\tau_2}$ from $N_2'^{\tau_2}$ and $\ldots$ and $N_n^{\tau_n}$

from $N_n'^{\tau_n}$. Therefore, we have that $\Sigma_2 \vdash_{\wedge G} M_2^{\tau_2} \hookrightarrow N_2^{\tau_2}$ and $\Gamma_2 \vdash_{\wedge G} N_2^{\tau_2} : \tau_2$ and ... and $\Sigma_n \vdash_{\wedge G} M_n^{\tau_n} \hookrightarrow N_n^{\tau_n}$ and $\Gamma_n \vdash_{\wedge G} N_n^{\tau_n} : \tau_n$ hold.

By rule [M-Par] we then have that $\Sigma_1 \wedge \ldots \wedge \Sigma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \hookrightarrow N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ holds, and by rule [T-Par] we have that $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge G} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ holds.

□

Theorem 5.7 (Monotonicity of Flow Marking). *If* $\Sigma_1 \vdash_{\wedge G} \Pi_1^{\sigma} \hookrightarrow \Pi_2^{\sigma}$ *and* $\Sigma_2 \vdash_{\wedge G} \Upsilon_1^{\upsilon} \hookrightarrow \Upsilon_2^{\upsilon}$ *and* $\Upsilon_1^{\upsilon} \sqsubseteq \Pi_1^{\sigma}$ *then* $\Upsilon_2^{\upsilon} \sqsubseteq \Pi_2^{\sigma}$.

Proof. This property is easy to verify since we mark coercions in the same position in the term with the same flow marks. We proceed by induction on the length of the derivation tree of $\Sigma_1 \vdash_{\wedge G} \Pi_1^{\sigma} \hookrightarrow \Pi_2^{\sigma}$.

Base cases:

- Rule [M-Con]. If $\emptyset \vdash_{\wedge G} k^B \hookrightarrow k^B$ and $\emptyset \vdash_{\wedge G} k^B \hookrightarrow k^B$ and $k^B \sqsubseteq k^B$ then $k^B \sqsubseteq k^B$.
- Rule [M-Var]. If $c_0^{\rho}(x) \sqsubseteq c_0^{\tau}(x)$, then we have that $c_0^{\rho}(x)$ and $c_0^{\tau}(x)$ are in the same position in the expression. Since flow marking inserts flow marks according to the position in the expression, then $c_0^{\rho}(x)$ and $c_0^{\tau}(x)$ will have the same flow mark. If $x : i \vdash_{\wedge G} c_0^{\tau}(x) \hookrightarrow c_i^{\tau}(x)$ and $x : i \vdash_{\wedge G} c_0^{\rho}(x) \hookrightarrow c_i^{\rho}(x)$ and $c_0^{\rho}(x) \sqsubseteq c_0^{\tau}(x)$ then by rule [P-Var] we have that $\rho \sqsubseteq \tau$. Therefore, we have that $c_i^{\rho}(x) \sqsubseteq c_i^{\tau}(x)$.

Induction step:

- Rule [M-AbsI]. If $\Sigma_1 \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} \hookrightarrow \lambda x : \sigma \,.\, M'^{\tau}$ and $\Sigma_2 \vdash_{\wedge G} \lambda x : \upsilon \,.\, N^{\rho} \hookrightarrow \lambda x : \upsilon \,.\, N'^{\rho}$ and $\lambda x : \upsilon \,.\, N^{\rho} \sqsubseteq \lambda x : \sigma \,.\, M^{\tau}$ then by rule [M-AbsI] we have that $\Sigma_1, (x : \sigma)_{\hookrightarrow} \vdash_{\wedge G} M^{\tau} \hookrightarrow M'^{\tau}$ and $\Sigma_2, (x : \upsilon)_{\hookrightarrow} \vdash_{\wedge G} N^{\rho} \hookrightarrow N'^{\rho}$. By rule [P-Abs], we have that $N^{\rho} \sqsubseteq M^{\tau}$ and $\upsilon \sqsubseteq \sigma$. By the induction hypothesis, we have that $N'^{\rho} \sqsubseteq M'^{\tau}$. Therefore, by rule [P-Abs], we have that $\lambda x : \upsilon \,.\, N'^{\rho} \sqsubseteq \lambda x : \sigma \,.\, M'^{\tau}$.
- Rule [M-AbsK]. If $\Sigma_1 \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} \hookrightarrow \lambda x : \sigma \,.\, M'^{\tau}$ and $\Sigma_2 \vdash_{\wedge G} \lambda x : \upsilon \,.\, N^{\rho} \hookrightarrow \lambda x : \upsilon \,.\, N'^{\rho}$ and $\lambda x : \upsilon \,.\, N^{\rho} \sqsubseteq \lambda x : \sigma \,.\, M^{\tau}$ then by rule [M-AbsK] we have that $\Sigma_1 \vdash_{\wedge G} M^{\tau} \hookrightarrow M'^{\tau}$ and $\Sigma_2 \vdash_{\wedge G} N^{\rho} \hookrightarrow N'^{\rho}$. By rule [P-Abs], we have that $N^{\rho} \sqsubseteq M^{\tau}$ and $\upsilon \sqsubseteq \sigma$. By the induction hypothesis, we have that $N'^{\rho} \sqsubseteq M'^{\tau}$. Therefore, by rule [P-Abs], we have that $\lambda x : \upsilon \,.\, N'^{\rho} \sqsubseteq \lambda x : \sigma \,.\, M'^{\tau}$.
- Rule [M-App]. If $\Sigma_1 \wedge \Sigma_2 \vdash_{\wedge G} M^{\rho}\, \Pi^{\upsilon} \hookrightarrow N^{\rho}\, \Upsilon^{\upsilon}$ and $\Sigma_1' \wedge \Sigma_2' \vdash_{\wedge G} M'^{\rho'}\, \Pi'^{\upsilon'} \hookrightarrow N'^{\rho'}\, \Upsilon'^{\upsilon'}$ and $M'^{\rho'}\, \Pi'^{\upsilon'} \sqsubseteq M^{\rho}\, \Pi^{\upsilon}$ then by rule [M-App] we have that $\Sigma_1 \vdash_{\wedge G} M^{\rho} \hookrightarrow N^{\rho}$ and $\Sigma_2 \vdash_{\wedge G} \Pi^{\upsilon} \hookrightarrow \Upsilon^{\upsilon}$, and $\Sigma_1' \vdash_{\wedge G} M'^{\rho'} \hookrightarrow N'^{\rho'}$ and $\Sigma_2' \vdash_{\wedge G} \Pi'^{\upsilon'} \hookrightarrow \Upsilon'^{\upsilon'}$. By rule [P-App], we have that $M'^{\rho'} \sqsubseteq M^{\rho}$ and $\Pi'^{\upsilon'} \sqsubseteq \Pi^{\upsilon}$. By the induction hypothesis, we have that $N'^{\rho'} \sqsubseteq N^{\rho}$ and $\Upsilon'^{\upsilon'} \sqsubseteq \Upsilon^{\upsilon}$. By rule [P-App], we have that $N'^{\rho'}\, \Upsilon'^{\upsilon'} \sqsubseteq N^{\rho}\, \Upsilon^{\upsilon}$.
- Rule [M-Add]. If $\Sigma_1 \wedge \Sigma_2 \vdash_{\wedge G} M_1^{\tau} + M_2^{\rho} \hookrightarrow N_1^{\tau} + N_2^{\rho}$ and $\Sigma_1' \wedge \Sigma_2' \vdash_{\wedge G} M_1'^{\tau'} + M_2'^{\rho'} \hookrightarrow N_1'^{\tau'} + N_2'^{\rho'}$ and $M_1'^{\tau'} + M_2'^{\rho'} \sqsubseteq M_1^{\tau} + M_2^{\rho}$ then by rule [M-Add] we have that $\Sigma_1 \vdash_{\wedge G} M_1^{\tau} \hookrightarrow N_1^{\tau}$ and $\Sigma_2 \vdash_{\wedge G} M_2^{\rho} \hookrightarrow N_2^{\rho}$, and $\Sigma_1' \vdash_{\wedge G} M_1'^{\tau'} \hookrightarrow N_1'^{\tau'}$ and $\Sigma_2' \vdash_{\wedge G} M_2'^{\rho'} \hookrightarrow N_2'^{\rho'}$. By rule [P-Add], we have that $M_1'^{\tau'} \sqsubseteq M_1^{\tau}$ and $M_2'^{\rho'} \sqsubseteq M_2^{\rho}$. By the induction hypothesis, we have that $N_1'^{\tau'} \sqsubseteq N_1^{\tau}$ and $N_2'^{\rho'} \sqsubseteq N_2^{\rho}$. By rule [P-Add], we have that $N_1'^{\tau'} + N_2'^{\rho'} \sqsubseteq N_1^{\tau} + N_2^{\rho}$.
- Rule [M-Par]. If $\Sigma_1 \wedge \ldots \wedge \Sigma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \hookrightarrow N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ and $\Sigma_1' \wedge \ldots \wedge \Sigma_n' \vdash_{\wedge G} M_1'^{\rho_1} \mid \ldots \mid M_n'^{\rho_n} \hookrightarrow N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n}$ and $M_1'^{\rho_1} \mid \ldots \mid M_n'^{\rho_n} \sqsubseteq M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$ then by rule [M-Par] we have that $\Sigma_1 \vdash_{\wedge G} M_1^{\tau_1} \hookrightarrow N_1^{\tau_1}$ and ... and $\Sigma_n \vdash_{\wedge G} M_n^{\tau_n} \hookrightarrow N_n^{\tau_n}$, and $\Sigma_1' \vdash_{\wedge G} M_1'^{\rho_1} \hookrightarrow N_1'^{\rho_1}$ and ... and $\Sigma_n' \vdash_{\wedge G} M_n'^{\rho_n} \hookrightarrow N_n'^{\rho_n}$. By rules [P-Par], we have that $M_1'^{\rho_1} \sqsubseteq M_1^{\tau_1}$ and ... and $M_n'^{\rho_n} \sqsubseteq M_n^{\tau_n}$. By the induction hypothesis, we have that $N_1'^{\rho_1} \sqsubseteq N_1^{\tau_1}$ and ... and $N_n'^{\rho_n} \sqsubseteq N_n^{\tau_n}$. By rule [P-Par], we have that $N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n} \sqsubseteq N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ and by definition 4.3, we have that $\rho_1 \wedge \ldots \wedge \rho_n \sqsubseteq \tau_1 \wedge \ldots \wedge \tau_n$.

□

Theorem 5.9 (Type Preservation of Cast Insertion). *If* $\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$ *then* $\Gamma \vdash_{\wedge CC} \Pi^{\sigma} \rightsquigarrow \Upsilon^{\sigma} : \sigma$ *and* $\Gamma \vdash_{\wedge CC} \Upsilon^{\sigma} : \sigma$.

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\wedge G} \Pi^{\sigma} : \sigma$.

Base cases:

- Rule [T-Con]. If $\emptyset \vdash_{\wedge G} k^B : B$ then by rule [T-Con] we have that k is a constant of base type B. Then, by rule [C-Con], we have that $\emptyset \vdash_{\wedge CC} k^B \rightsquigarrow k^B : B$ holds and by rule [T-Con] we have that $\emptyset \vdash_{\wedge CC} k^B : B$ holds.
- Rule [T-Var]. By rule [T-Var], we have that $x : \tau \vdash_{\wedge G} c_i^{\tau}(x) : \tau$ holds. By rule [C-Var], we have that $x : \tau \vdash_{\wedge CC} c_i^{\tau}(x) \rightsquigarrow c_i^{\tau}(x) : \tau$ holds. By rule [T-Var], we have that $x : \tau \vdash_{\wedge CC} c_i^{\tau}(x) : \tau$ holds.

Induction step:

- Rule [T-AbsI]. If $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} : \sigma \to \tau$ then by rule [T-AbsI] we have that $\Gamma, x : \sigma \vdash_{\wedge G} M^{\tau} : \tau$ and $x \in fv(M^{\tau})$. By the induction hypothesis, we have that $\Gamma, x : \sigma \vdash_{\wedge CC} M^{\tau} \rightsquigarrow N^{\tau} : \tau$ and $\Gamma, x : \sigma \vdash_{\wedge CC} N^{\tau} : \tau$ hold. By rule [C-AbsI], we then have that $\Gamma \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^{\tau} \rightsquigarrow \lambda x : \sigma \,.\, N^{\tau} : \sigma \to \tau$ holds, and by rule [T-AbsI], we then have that $\Gamma \vdash_{\wedge CC} \lambda x : \sigma \,.\, N^{\tau} : \sigma \to \tau$.
- Rule [T-AbsK]. If $\Gamma \vdash_{\wedge G} \lambda x : \sigma \,.\, M^{\tau} : \sigma \to \tau$ then by rule [T-AbsK] we have that $\Gamma \vdash_{\wedge G} M^{\tau} : \tau$ and $x \notin fv(M^{\tau})$. By the induction hypothesis, we have that $\Gamma \vdash_{\wedge CC} M^{\tau} \rightsquigarrow N^{\tau} : \tau$ and $\Gamma \vdash_{\wedge CC} N^{\tau} : \tau$ hold. By rule [C-AbsK], we then have that $\Gamma \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^{\tau} \rightsquigarrow \lambda x : \sigma \,.\, N^{\tau} : \sigma \to \tau$ holds, and by rule [T-AbsK], we then have that $\Gamma \vdash_{\wedge CC} \lambda x : \sigma \,.\, N^{\tau} : \sigma \to \tau$.
- Rule [T-App]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M^{\rho}\, \Pi^{\upsilon} : \tau$ then by rule [T-App] we have that $\Gamma_1 \vdash_{\wedge G} M^{\rho} : \rho$, $\rho \rhd \sigma \to \tau$, $\Gamma_2 \vdash_{\wedge G} \Pi^{\upsilon} : \upsilon$ and $\upsilon \sim \sigma$ hold. By the induction hypothesis we have that $\Gamma_1 \vdash_{\wedge CC} M^{\rho} \rightsquigarrow N^{\rho} : \rho$ and $\Gamma_2 \vdash_{\wedge CC} \Pi^{\upsilon} \rightsquigarrow \Upsilon^{\upsilon} : \upsilon$ hold, and also that $\Gamma_1 \vdash_{\wedge CC} N^{\rho} : \rho$ and $\Gamma_2 \vdash_{\wedge CC} \Upsilon^{\upsilon} : \upsilon$ hold. By rule [C-App] we then have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M^{\rho}\, \Pi^{\upsilon} \rightsquigarrow (N^{\rho} : \rho \Rightarrow \sigma \to \tau)\ (\Upsilon^{\upsilon} : \upsilon \Rightarrow_{\wedge} \sigma) : \tau$ holds. By rule [T-Cast] we have that $\Gamma_1 \vdash_{\wedge CC} (N^{\rho} : \rho \Rightarrow \sigma \to \tau) : \sigma \to \tau$ holds, and also that $\Gamma_2 \vdash_{\wedge CC} (\Upsilon^{\upsilon} : \upsilon \Rightarrow_{\wedge} \sigma) : \sigma$ holds. By rule [T-App]

we then have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} (N^\rho : \rho \Rightarrow \sigma \to \tau)\ (\Upsilon^\upsilon : \upsilon \Rightarrow_\wedge \sigma) : \tau$ holds.

- Rule [T-Add]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge G} M_1^\tau + M_2^\rho : Int$ then by rule [T-Add] we have that $\Gamma_1 \vdash_{\wedge G} M_1^\tau : \tau, \tau \rhd Int, \Gamma_2 \vdash_{\wedge G} M_2^\rho : \rho$ and $\rho \rhd Int$ hold. By the induction hypothesis, we have that $\Gamma_1 \vdash_{\wedge CC} M_1^\tau \rightsquigarrow N_1^\tau : \tau$ and $\Gamma_2 \vdash_{\wedge CC} M_2^\rho \rightsquigarrow N_2^\rho : \rho$ hold, and also that $\Gamma_1 \vdash_{\wedge CC} N_1^\tau : \tau$ and $\Gamma_2 \vdash_{\wedge CC} N_2^\rho : \rho$ hold. By rule [C-Add] we then have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M_1^\tau + M_2^\rho \rightsquigarrow (N_1^\tau : \tau \Rightarrow Int) + (N_2^\rho : \rho \Rightarrow Int) : Int$ holds. By rule [T-Cast] we have that $\Gamma_1 \vdash_{\wedge CC} (N_1^\tau : \tau \Rightarrow Int) : Int$ holds, and also that $\Gamma_2 \vdash_{\wedge CC} (N_2^\rho : \rho \Rightarrow Int) : Int$ holds. By rule [T-Add] we then have that $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} (N_1^\tau : \tau \Rightarrow Int) + (N_2^\rho : \rho \Rightarrow Int) : Int$ holds.
- Rule [T-Par]. If $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge G} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then by rule [T-Par] we have that $\Gamma_1 \vdash_{\wedge G} M_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_{\wedge G} M_n^{\tau_n} : \tau_n$ hold. By the induction hypothesis, we have that $\Gamma_1 \vdash_{\wedge CC} M_1^{\tau_1} \rightsquigarrow N_1^{\tau_1} : \tau_1$ and $\Gamma_1 \vdash_{\wedge CC} N_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_{\wedge CC} M_n^{\tau_n} \rightsquigarrow N_n^{\tau_n} : \tau_n$ and $\Gamma_n \vdash_{\wedge CC} N_n^{\tau_n} : \tau_n$ hold. By rule [C-Par] we then have that $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge CC} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \rightsquigarrow N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} : \tau_1 \ldots \wedge \tau_n$ holds, and by rule [T-Par] we have that $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ holds.

∎

Theorem 5.10 (Monotonicity of Cast Insertion). *If $\Gamma_1 \vdash_{\wedge CC} \Pi_1^\sigma \rightsquigarrow \Pi_2^\sigma : \sigma$ and $\Gamma_2 \vdash_{\wedge CC} \Upsilon_1^\upsilon \rightsquigarrow \Upsilon_2^\upsilon : \upsilon$ and $\Upsilon_1^\upsilon \sqsubseteq \Pi_1^\sigma$ then $\Upsilon_2^\upsilon \sqsubseteq \Pi_2^\sigma$ and $\upsilon \sqsubseteq \sigma$.*

Proof. We proceed by induction on the length of the derivation tree of $\Gamma_1 \vdash_{\wedge CC} \Pi_1^\sigma \rightsquigarrow \Pi_2^\sigma : \sigma$.

Base cases:

- Rule [C-Con]. If $\emptyset \vdash_{\wedge CC} k^B \rightsquigarrow k^B : B$ and $\emptyset \vdash_{\wedge CC} k^B \rightsquigarrow k^B : B$ and $k^B \sqsubseteq k^B$ then $k^B \sqsubseteq k^B$ and $B \sqsubseteq B$.
- Rule [C-Var]. If $x : \tau \vdash_{\wedge CC} c_i^\tau(x) \rightsquigarrow c_i^\tau(x) : \tau$ and $x : \rho \vdash_{\wedge CC} c_i^\rho(x) \rightsquigarrow c_i^\rho(x) : \rho$ and $c_i^\rho(x) \sqsubseteq c_i^\tau(x)$ then by rule [P-Var] we have that $\rho \sqsubseteq \tau$. Therefore, we have that $c_i^\rho(x) \sqsubseteq c_i^\tau(x)$ and $\rho \sqsubseteq \tau$.

Induction step:

- Rule [C-AbsI]. If $\Gamma_1 \vdash_{\wedge CC} \lambda x : \sigma\ .\ M^\tau \rightsquigarrow \lambda x : \sigma\ .\ M'^\tau : \sigma \to \tau$ and $\Gamma_2 \vdash_{\wedge CC} \lambda x : \upsilon\ .\ N^\rho \rightsquigarrow \lambda x : \upsilon\ .\ N'^\rho : \upsilon \to \rho$ and $\lambda x : \upsilon\ .\ N^\rho \sqsubseteq \lambda x : \sigma\ .\ M^\tau$ then by rule [C-AbsI] we have that $\Gamma_1, x : \sigma \vdash_{\wedge CC} M^\tau \rightsquigarrow M'^\tau : \tau$ and $\Gamma_2, x : \upsilon \vdash_{\wedge CC} N^\rho \rightsquigarrow N'^\rho : \rho$. By rule [P-Abs], we have that $N^\rho \sqsubseteq M^\tau$ and $\upsilon \sqsubseteq \sigma$. By the induction hypothesis, we have that $N'^\rho \sqsubseteq M'^\tau$ and $\rho \sqsubseteq \tau$. Therefore, by rule [P-Abs], we have that $\lambda x : \upsilon\ .\ N'^\rho \sqsubseteq \lambda x : \sigma\ .\ M'^\tau$. By definition 4.3, we have that $\upsilon \to \rho \sqsubseteq \sigma \to \tau$.
- Rule [C-AbsK]. If $\Gamma_1 \vdash_{\wedge CC} \lambda x : \sigma\ .\ M^\tau \rightsquigarrow \lambda x : \sigma\ .\ M'^\tau : \sigma \to \tau$ and $\Gamma_2 \vdash_{\wedge CC} \lambda x : \upsilon\ .\ N^\rho \rightsquigarrow \lambda x : \upsilon\ .\ N'^\rho : \upsilon \to \rho$ and $\lambda x : \upsilon\ .\ N^\rho \sqsubseteq \lambda x : \sigma\ .\ M^\tau$ then by rule [C-AbsK] we have that $\Gamma_1 \vdash_{\wedge CC} M^\tau \rightsquigarrow M'^\tau : \tau$ and $\Gamma_2 \vdash_{\wedge CC} N^\rho \rightsquigarrow N'^\rho : \rho$. By rule [P-Abs], we have that $N^\rho \sqsubseteq M^\tau$ and $\upsilon \sqsubseteq \sigma$. By the induction hypothesis, we have that $N'^\rho \sqsubseteq M'^\tau$ and $\rho \sqsubseteq \tau$. Therefore, by rule [P-Abs], we have that $\lambda x : \upsilon\ .\ N'^\rho \sqsubseteq \lambda x : \sigma\ .\ M'^\tau$. By definition 4.3, we have that $\upsilon \to \rho \sqsubseteq \sigma \to \tau$.
- Rule [C-App]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M^\rho\ \Pi^\upsilon \rightsquigarrow (N^\rho : \rho \Rightarrow \sigma \to \tau)\ (\Upsilon^\upsilon : \upsilon \Rightarrow_\wedge \sigma) : \tau$ and $\Gamma_1' \wedge \Gamma_2' \vdash_{\wedge CC} M'^{\rho'}\ \Pi'^{\upsilon'} \rightsquigarrow (N'^{\rho'} : \rho' \Rightarrow \sigma' \to \tau')\ (\Upsilon'^{\upsilon'} : \upsilon' \Rightarrow_\wedge \sigma') : \tau'$ and $M'^{\rho'}\ \Pi'^{\upsilon'} \sqsubseteq M^\rho\ \Pi^\upsilon$ then by rule [C-App] we have that $\Gamma_1 \vdash_{\wedge CC} M^\rho \rightsquigarrow N^\rho : \rho, \rho \rhd \sigma \to \tau, \Gamma_2 \vdash_{\wedge CC} \Pi^\upsilon \rightsquigarrow \Upsilon^\upsilon : \upsilon$ and $\upsilon \sim \sigma$, and $\Gamma_1' \vdash_{\wedge CC} M'^{\rho'} \rightsquigarrow N'^{\rho'} : \rho', \rho' \rhd \sigma' \to \tau', \Gamma_2' \vdash_{\wedge CC} \Pi'^{\upsilon'} \rightsquigarrow \Upsilon'^{\upsilon'} : \upsilon'$ and $\upsilon' \sim \sigma'$. By rule [P-App], we have that $M'^{\rho'} \sqsubseteq M^\rho$ and $\Pi'^{\upsilon'} \sqsubseteq \Pi^\upsilon$. By the induction hypothesis, we have that $N'^{\rho'} \sqsubseteq N^\rho$ and $\Upsilon'^{\upsilon'} \sqsubseteq \Upsilon^\upsilon$, and that $\rho' \sqsubseteq \rho$ and $\upsilon' \sqsubseteq \upsilon$. By definition 4.3, we have that $\sigma' \to \tau' \sqsubseteq \sigma \to \tau$. Therefore, by rule [P-Cast], we have that $(N'^{\rho'} : \rho' \Rightarrow \sigma' \to \tau') \sqsubseteq (N^\rho : \rho \Rightarrow \sigma \to \tau)$ and $(\Upsilon'^{\upsilon'} : \upsilon' \Rightarrow_\wedge \sigma') \sqsubseteq (\Upsilon^\upsilon : \upsilon \Rightarrow_\wedge \sigma)$. By rule [P-App], we have that $(N'^{\rho'} : \rho' \Rightarrow \sigma' \to \tau')\ (\Upsilon'^{\upsilon'} : \upsilon' \Rightarrow_\wedge \sigma') \sqsubseteq (N^\rho : \rho \Rightarrow \sigma \to \tau)\ (\Upsilon^\upsilon : \upsilon \Rightarrow_\wedge \sigma)$. By definition 4.3, we have that $\tau' \sqsubseteq \tau$.
- Rule [C-Add]. If $\Gamma_1 \wedge \Gamma_2 \vdash_{\wedge CC} M_1^\tau + M_2^\rho \rightsquigarrow (N_1^\tau : \tau \Rightarrow Int) + (N_2^\rho : \rho \Rightarrow Int) : Int$ and $\Gamma_1' \wedge \Gamma_2' \vdash_{\wedge CC} M_1'^{\tau'} + M_2'^{\rho'} \rightsquigarrow (N_1'^{\tau'} : \tau' \Rightarrow Int) + (N_2'^{\rho'} : \rho' \Rightarrow Int) : Int$ and $M_1'^{\tau'} + M_2'^{\rho'} \sqsubseteq M_1^\tau + M_2^\rho$ then by rule [C-Add] we have that $\Gamma_1 \vdash_{\wedge CC} M_1^\tau \rightsquigarrow N_1^\tau : \tau, \tau \rhd Int, \Gamma_2 \vdash_{\wedge CC} M_2^\rho \rightsquigarrow N_2^\rho : \rho$ and $\rho \rhd Int$, and $\Gamma_1' \vdash_{\wedge CC} M_1'^{\tau'} \rightsquigarrow N_1'^{\tau'} : \tau', \tau' \rhd Int, \Gamma_2' \vdash_{\wedge CC} M_2'^{\rho'} \rightsquigarrow N_2'^{\rho'} : \rho'$ and $\rho' \rhd Int$. By rule [P-Add], we have that $M_1'^{\tau'} \sqsubseteq M_1^\tau$ and $M_2'^{\rho'} \sqsubseteq M_2^\rho$. By the induction hypothesis, we have that $N_1'^{\tau'} \sqsubseteq N_1^\tau$ and $N_2'^{\rho'} \sqsubseteq N_2^\rho$, and that $\tau' \sqsubseteq \tau$ and $\rho' \sqsubseteq \rho$. By definition 4.3, we have that $Int \sqsubseteq Int$. Therefore, by rule [P-Cast], we have that $N_1'^{\tau'} : \tau' \Rightarrow Int \sqsubseteq N_1^\tau : \tau \Rightarrow Int$ and $N_2'^{\rho'} : \rho' \Rightarrow Int \sqsubseteq N_2^\rho : \rho \Rightarrow Int$. By rule [P-Add], we have that $(N_1'^{\tau'} : \tau' \Rightarrow Int) + (N_2'^{\rho'} : \rho' \Rightarrow Int) \sqsubseteq (N_1^\tau : \tau \Rightarrow Int) + (N_2^\rho : \rho \Rightarrow Int)$.
- Rule [C-Par]. If $\Gamma_1 \wedge \ldots \wedge \Gamma_n \vdash_{\wedge CC} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \rightsquigarrow N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ and $\Gamma_1' \wedge \ldots \wedge \Gamma_n' \vdash_{\wedge CC} M_1'^{\rho_1} \mid \ldots \mid M_n'^{\rho_n} \rightsquigarrow N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n} : \rho_1 \wedge \ldots \wedge \rho_n$ and $M_1'^{\rho_1} \mid \ldots \mid M_n'^{\rho_n} \sqsubseteq M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$ then by rule [C-Par] we have that $\Gamma_1 \vdash_{\wedge CC} M_1^{\tau_1} \rightsquigarrow N_1^{\tau_1} : \tau_1$ and ... and $\Gamma_n \vdash_{\wedge CC} M_n^{\tau_n} \rightsquigarrow N_n^{\tau_n} : \tau_n$, and $\Gamma_1' \vdash_{\wedge CC} M_1'^{\rho_1} \rightsquigarrow N_1'^{\rho_1} : \rho_1$ and ... and $\Gamma_n' \vdash_{\wedge CC} M_n'^{\rho_n} \rightsquigarrow N_n'^{\rho_n} : \rho_n$. By rules [P-Par], we have that $M_1'^{\rho_1} \sqsubseteq M_1^{\tau_1}$ and ... and $M_n'^{\rho_n} \sqsubseteq M_n^{\tau_n}$. By the induction hypothesis, we have that $N_1'^{\rho_1} \sqsubseteq N_1^{\tau_1}$ and ... and $N_n'^{\rho_n} \sqsubseteq N_n^{\tau_n}$ and $\rho_1 \sqsubseteq \tau_1$ and ... and $\rho_n \sqsubseteq \tau_n$. By rule [P-Par], we have that $N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n} \sqsubseteq N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ and by definition 4.3, we have that $\rho_1 \wedge \ldots \wedge \rho_n \sqsubseteq \tau_1 \wedge \ldots \wedge \tau_n$.

∎

C PROOFS (OPERATIONAL SEMANTICS)

In this section we present the full proofs for all the properties in section 6:

- Theorem 6.2 (Conservative Extension of Operational Semantics) in C;
- Theorem 6.3 (Type Preservation) in C;

- Theorem 6.4 (Progress) in C;
- Lemma 6.9 (Simulation of More Precise Programs) in C;
- Theorem 6.10 (Gradual Guarantee) in C;
- Lemma 6.15 (Simulation of Variant Programs) in C;
- Theorem 6.16 (Confluency of Operational Semantics) in C.

$$\begin{array}{rrcl} \mathit{Values} & v & ::= & k^B \mid \lambda x : \sigma \,.\, M^\tau \\ \mathit{Parallel\ Values} & \pi & ::= & (v_1^{\tau_1} \mid \ldots \mid v_n^{\tau_n}) \quad (\text{with } n \geq 1) \\ \mathit{Evaluation\ Contexts} & E & ::= & \square \mid E\ \Pi^\sigma \mid v^\tau\ E \mid E + M^\tau \mid v^\tau + E \end{array}$$

Lemma C.1 (Conservative Extension of Operational Semantics). *If Π^σ is a static term and σ is a static type, then $\Pi^\sigma \longrightarrow_\wedge \Upsilon^\sigma \iff \Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$.*

Proof. We proceed by induction on the length of the reductions using $\longrightarrow_\wedge$ and $\longrightarrow_{\wedge CC}$ for the right and left direction of the implication, respectively.

Base case:

- Rule [E-Beta]. As $(\lambda x : \sigma \,.\, M^\tau)\ \pi^\sigma \longrightarrow_\wedge [c_i^\rho(x) \mapsto \langle \pi^\sigma \rangle_i^\rho]\ M^\tau$ and $(\lambda x : \sigma \,.\, M^\tau)\ \pi^\sigma \longrightarrow_{\wedge CC} [c_i^\rho(x) \mapsto \langle \pi^\sigma \rangle_i^\rho]\ M^\tau$, it is proven.
- Rule [E-Add]. As $k_1^{Int} + k_2^{Int} \longrightarrow_\wedge k_3^{Int}$ and $k_1^{Int} + k_2^{Int} \longrightarrow_{\wedge CC} k_3^{Int}$, it is proven.

Induction step:

- Rule [E-Par].
  - If $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_\wedge N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ then by rule [E-Par], we have that $M_1^{\tau_1} \longrightarrow_\wedge N_1^{\tau_1}$ and … and $M_n^{\tau_n} \longrightarrow_\wedge N_n^{\tau_n}$. By the induction hypothesis, we have that $M_1^{\tau_1} \longrightarrow_{\wedge CC} N_1^{\tau_1}$ and … and $M_n^{\tau_n} \longrightarrow_{\wedge CC} N_n^{\tau_n}$. Therefore, by rule [E-Par], we have that $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$.
  - If $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ then by rule [E-Par], we have that $\forall i$. either $M_i^{\tau_i}$ is a result and $M_i^{\tau_i} = N_i^{\tau_i}$ or $M_i^{\tau_1} \longrightarrow_{\wedge CC} N_i^{\tau_i}$ and $\exists i$. $M_i^{\tau_i}$ is not a result. Since $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$ is a static term, then each term in the parallel is exactly the same except for type annotations. Therefore, we have that $M_1^{\tau_1} \longrightarrow_{\wedge CC} N_1^{\tau_1}$ and … and $M_n^{\tau_n} \longrightarrow_{\wedge CC} N_n^{\tau_n}$. By the induction hypothesis, we have that $M_1^{\tau_1} \longrightarrow_\wedge N_1^{\tau_1}$ and … and $M_n^{\tau_n} \longrightarrow_\wedge N_n^{\tau_n}$. By rule [E-Par], we have that $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_\wedge N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$.

□

Theorem 6.2 (Conservative Extension of Operational Semantics). *If Π^σ is static and σ is a static type, then $\Pi^\sigma \longrightarrow_\wedge \Upsilon^\sigma \iff \Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$.*

Proof. We proceed by structural induction on evaluation contexts, for both directions of the implication, and using lemma C.1.

Base case: by lemma C.1.
Induction step:

- Context $E\ \Pi^\sigma$.
  - If $E\ \Pi^\sigma \longrightarrow_\wedge E'\ \Pi^\sigma$, then by rule [E-Ctx], we have that $E \longrightarrow_\wedge E'$. By the induction hypothesis, we have that $E \longrightarrow_{\wedge CC} E'$. By rule [E-Ctx], we have that $E\ \Pi^\sigma \longrightarrow_{\wedge CC} E'\ \Pi^\sigma$.
  - If $E\ \Pi^\sigma \longrightarrow_{\wedge CC} E'\ \Pi^\sigma$, then by rule [E-Ctx], we have that $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $E \longrightarrow_{\wedge CC} E'$. By rule [E-Ctx], we have that $E\ \Pi^\sigma \longrightarrow_\wedge E'\ \Pi^\sigma$.
- Context $v^\tau\ E$.
  - If $v^\tau\ E \longrightarrow_\wedge v^\tau\ E'$, then by rule [E-Ctx], we have that $E \longrightarrow_\wedge E'$. By the induction hypothesis, we have that $E \longrightarrow_{\wedge CC} E'$. By rule [E-Ctx], we have that $v^\tau\ E \longrightarrow_{\wedge CC} v^\tau\ E'$.
  - If $v^\tau\ E \longrightarrow_{\wedge CC} v^\tau\ E'$, then by rule [E-Ctx], we have that $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $E \longrightarrow_{\wedge CC} E'$. By rule [E-Ctx], we have that $v^\tau\ E \longrightarrow_\wedge v^\tau\ E'$.
- Context $E + M^\tau$.
  - If $E + M^\tau \longrightarrow_\wedge E' + M^\tau$, then by rule [E-Ctx], we have that $E \longrightarrow_\wedge E'$. By the induction hypothesis, we have that $E \longrightarrow_{\wedge CC} E'$. By rule [E-Ctx], we have that $E + M^\tau \longrightarrow_{\wedge CC} E' + M^\tau$.
  - If $E + M^\tau \longrightarrow_{\wedge CC} E' + M^\tau$, then by rule [E-Ctx], we have that $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $E \longrightarrow_\wedge E'$. By rule [E-Ctx], we have that $E + M^\tau \longrightarrow_\wedge E' + M^\tau$.
- Context $v^\tau + E$.
  - If $v^\tau + E \longrightarrow_\wedge v^\tau + E'$, then by rule [E-Ctx], we have that $E \longrightarrow_\wedge E'$. By the induction hypothesis, we have that $E \longrightarrow_{\wedge CC} E'$. By rule [E-Ctx], we have that $v^\tau + E \longrightarrow_{\wedge CC} v^\tau + E'$.
  - If $v^\tau + E \longrightarrow_{\wedge CC} v^\tau + E'$, then by rule [E-Ctx], we have that $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $E \longrightarrow_\wedge E'$. By rule [E-Ctx], we have that $v^\tau + E \longrightarrow_\wedge v^\tau + E'$.

□

Lemma C.2 (Type Preservation). *If $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ and $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$ then $\emptyset \vdash_{\wedge CC} \Upsilon^\sigma : \sigma$.*

Proof. We proceed by induction on the length of the reduction using $\longrightarrow_{\wedge CC}$.

Base cases:

- Rule [EC-Identity]. If $\emptyset \vdash_{\wedge CC} v^\tau : \tau \Rightarrow \tau : \tau$ and $v^\tau : \tau \Rightarrow \tau \longrightarrow_{\wedge CC} v^\tau$ then by rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} v^\tau : \tau$.
- Rule [EC-Application]. If $\emptyset \vdash_{\wedge CC} (v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow v \to \rho)\ \pi^v : \rho$ and $(v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow v \to \rho)\ \pi^v \longrightarrow_{\wedge CC} (v^{\sigma \to \tau}\ (\pi^v : v \Rightarrow_\wedge \sigma)) : \tau \Rightarrow \rho$, then by rule [T-App], we have that $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow v \to \rho : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi^v : v$. By rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau$. By rule [T-Par] and [T-Cast], we have that $\emptyset \vdash_{\wedge CC} \pi^v : v \Rightarrow_\wedge \sigma : \sigma$. By rule [T-App] we have that $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau}\ (\pi^v : v \Rightarrow_\wedge \sigma) : \tau$. By rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} (v^{\sigma \to \tau}\ (\pi^v : v \Rightarrow_\wedge \sigma)) : \tau \Rightarrow \rho : \rho$.

- Rule [EC-Succeed]. If $\emptyset \vdash_{\wedge CC} v^G : G \Rightarrow Dyn : Dyn \Rightarrow G : G$ and $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \longrightarrow_{\wedge CC} v^G$, then by rule [T-Cast] we have that $\emptyset \vdash_{\wedge CC} v^G : G \Rightarrow Dyn : Dyn$. By rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} v^G : G$.
- Rule [EC-Fail]. If $\emptyset \vdash_{\wedge CC} v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 : G_2$ and $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \longrightarrow_{\wedge CC} wrong^{G_2}$ then by rule [T-Wrong], we have that $\emptyset \vdash_{\wedge CC} wrong^{G_2} : G_2$.
- Rule [EC-Ground]. If $\emptyset \vdash_{\wedge CC} v^\tau : \tau \Rightarrow Dyn : Dyn$ and $v^\tau : \tau \Rightarrow Dyn \longrightarrow_{\wedge CC} v^\tau : \tau \Rightarrow G : G \Rightarrow Dyn$ then we have that $\tau \sim G$ and by rule [T-Cast], $\emptyset \vdash_{\wedge CC} v^\tau : \tau$. By rule [T-Cast] we have $\emptyset \vdash_{\wedge CC} v^\tau : \tau \Rightarrow G : G$. By rule [T-Cast] we have that $\emptyset \vdash_{\wedge CC} v^\tau : \tau \Rightarrow G : G \Rightarrow Dyn : Dyn$.
- Rule [EC-Expand]. If $\emptyset \vdash_{\wedge CC} v^{Dyn} : Dyn \Rightarrow \tau : \tau$ and $v^{Dyn} : Dyn \Rightarrow \tau \longrightarrow_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$ then we have that $\tau \sim G$ and by rule [T-Cast], $\emptyset \vdash_{\wedge CC} v^{Dyn} : Dyn$. By rule [T-Cast] we have that $\emptyset \vdash_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G$. By rule [T-Cast] we have that $\emptyset \vdash_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau : \tau$.
- Rule [E-Beta]. If $\emptyset \vdash_{\wedge CC} (\lambda x : \sigma . M^\tau)\ \pi^\sigma : \tau$ and $(\lambda x : \sigma . M^\tau)\ \pi^\sigma \longrightarrow_{\wedge CC} [c_i^\rho(x) \mapsto \langle \pi^\sigma \rangle_i^\rho]\ M^\tau$ then $[c_i^\rho(x) \mapsto \langle \pi^\sigma \rangle_i^\rho]\ M^\tau$ is formed by replacing coercions of type ρ by terms of type ρ, according to figure 4 and 6.1, in the term M^τ of type τ. Therefore, $\emptyset \vdash_{\wedge CC} [c_i^\rho(x) \mapsto \langle \pi^\sigma \rangle_i^\rho]\ M^\tau : \tau$.
- Rule [E-Add]. If $\emptyset \vdash_{\wedge CC} k_1^{Int} + k_2^{Int} : Int$ and $k_1^{Int} + k_2^{Int} \longrightarrow_{\wedge CC} k_3^{Int}$, by rule [T-Con], we have that $\emptyset \vdash_{\wedge CC} k_3^{Int} : Int$.
- Rule [E-Wrong]. If $\emptyset \vdash_{\wedge CC} E[wrong^\sigma] : \tau$ and $E[wrong^\sigma] \longrightarrow_{\wedge CC} wrong^\tau$ then, by rule [T-Wrong], $\emptyset \vdash_{\wedge CC} wrong^\tau : \tau$.
- Rule [E-Push]. If $\emptyset \vdash_{\wedge CC} r_1^{\tau_1} \mid \ldots \mid r_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ and $r_1^{\tau_1} \mid \ldots \mid r_n^{\tau_n} \longrightarrow_{\wedge CC} wrong^\sigma$ (with $\sigma = \tau_1 \wedge \ldots \wedge \tau_n$) then, by rule [T-Wrong], $\emptyset \vdash_{\wedge CC} wrong^\sigma : \tau_1 \wedge \ldots \wedge \tau_n$.

Induction step:

- Rule [E-Par]. If $\emptyset \vdash_{\wedge CC} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ and $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ then by rule [T-Par] we have that $\emptyset \vdash_{\wedge CC} M_1^{\tau_1} : \tau_1$ and $\ldots$ and $\emptyset \vdash_{\wedge CC} M_n^{\tau_n} : \tau_n$, and by rule [E-Par], we have that $\forall i$. either $M_i^{\tau_i}$ is a result and $M_i^{\tau_i} = N_i^{\tau_i}$ or $M_i^{\tau_1} \longrightarrow_{\wedge CC} N_i^{\tau_i}$ and $\exists i$. $M_i^{\tau_i}$ is not a result. For all i such that $M_i^{\tau_1} \longrightarrow_{\wedge CC} N_i^{\tau_i}$, by the induction hypothesis, we have that $\emptyset \vdash_{\wedge CC} N_i^{\tau_i} : \tau_i$. By rule [T-Par], we have that $\emptyset \vdash_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$.

□

Theorem 6.3 (Type Preservation). *If* $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ *and* $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$ *then* $\emptyset \vdash_{\wedge CC} \Upsilon^\sigma : \sigma$.

Proof. We proceed by structural induction on evaluation contexts, and using lemma C.2.

Base case: by lemma C.2.
Induction step:

- Context $E\ \Pi^\sigma$. If $\emptyset \vdash_{\wedge CC} E\ \Pi^\sigma : \tau$ and $E\ \Pi^\sigma \longrightarrow_{\wedge CC} E'\ \Pi^\sigma$ then by rule [T-App], $\emptyset \vdash_{\wedge CC} E : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$, and by rule [E-Ctx], $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $\emptyset \vdash_{\wedge CC} E' : \sigma \to \tau$. By rule [T-App], we have that $\emptyset \vdash_{\wedge CC} E'\ \Pi^\sigma : \tau$.
- Context $v^\tau\ E$. If $\emptyset \vdash_{\wedge CC} v^\tau\ E : \rho$ and $v^\tau\ E \longrightarrow_{\wedge CC} v^\tau\ E'$ then by rule [T-App], $\emptyset \vdash_{\wedge CC} v^\tau : \tau$, with $\tau = \sigma \to \rho$ and $\emptyset \vdash_{\wedge CC} E : \sigma$, and by rule [E-Ctx], $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $\emptyset \vdash_{\wedge CC} E' : \sigma$. By rule [T-App], we have that $\emptyset \vdash_{\wedge CC} v^\tau\ E' : \rho$.
- Context $E + M^\tau$. If $\emptyset \vdash_{\wedge CC} E + M^{Int} : Int$ and $E + M^{Int} \longrightarrow_{\wedge CC} E' + M^{Int}$ then by rule [T-Add], $\emptyset \vdash_{\wedge CC} E : Int$ and $\emptyset \vdash_{\wedge CC} M^{Int} : Int$, and by rule [E-Ctx], $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $\emptyset \vdash_{\wedge CC} E' : Int$. By rule [T-App], we have that $\emptyset \vdash_{\wedge CC} E' + M^{Int} : Int$.
- Context $v^\tau + E$. If $\emptyset \vdash_{\wedge CC} v^{Int} + E : Int$ and $v^{Int} + E \longrightarrow_{\wedge CC} v^{Int} + E'$ then by rule [T-Add], $\emptyset \vdash_{\wedge CC} v^{Int} : Int$ and $\emptyset \vdash_{\wedge CC} E : Int$, and by rule [E-Ctx], $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $\emptyset \vdash_{\wedge CC} E' : Int$. By rule [T-Add], we have that $\emptyset \vdash_{\wedge CC} v^{Int} + E' : Int$.
- Context $E : \tau \Rightarrow \rho$. If $\emptyset \vdash_{\wedge CC} E : \tau \Rightarrow \rho : \rho$ and $E : \tau \Rightarrow \rho \longrightarrow_{\wedge CC} E' : \tau \Rightarrow \rho$ then by rule [T-Cast], $\emptyset \vdash_{\wedge CC} E : \tau$, and by rule [E-Ctx], we have that $E \longrightarrow_{\wedge CC} E'$. By the induction hypothesis, we have that $\emptyset \vdash_{\wedge CC} E' : \tau$. By rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} E' : \tau \Rightarrow \rho : \rho$.

□

Theorem 6.4 (Progress). *If* $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ *then either* Π^σ *is a parallel value or* $\exists \Upsilon^\sigma$ *such that* $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$.

Proof. We proceed by induction on the length of the derivation tree of $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$.

Base cases:

- Rule [T-Con]. If $\emptyset \vdash_{\wedge CC} k^B : B$ then k^B is a value.
- Rule [T-Wrong]. If $\emptyset \vdash_{\wedge CC} wrong^\sigma : \sigma$ then $wrong^\sigma$ is a parallel value.

Induction step:

- Rule [T-AbsI. If $\emptyset \vdash_{\wedge CC} \lambda x : \sigma . M^\tau : \sigma \to \tau$ then $\lambda x : \sigma . M^\tau$ is a value.
- Rule [T-AbsK]. If $\emptyset \vdash_{\wedge CC} \lambda x : \sigma . M^\tau : \sigma \to \tau$ then $\lambda x : \sigma . M^\tau$ is a value.
- Rule [T-App]. If $\emptyset \vdash_{\wedge CC} M^\tau\ \Pi^\sigma : \rho$ then by rule [T-App], we have that $\emptyset \vdash_{\wedge CC} M^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$. By the induction hypothesis M^τ is either a value or *wrong* or $\exists N^\tau$ such that $M^\tau \longrightarrow_{\wedge CC} N^\tau$, and also by the induction hypothesis Π^σ is either a parallel value or $\exists \Upsilon^\sigma$ such that $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$. There are several possibilities:
  - If M^τ is a value and Π^σ is a parallel value (without any *wrong*), then M^τ must be a λ-abstraction, and we can apply rule [E-Beta], or M^τ is a cast and we can apply rule [EC-Application].
  - If M^τ is a value and Π^σ is a $wrong^\sigma$, by rule [E-Wrong], $M^\tau\ \Pi^\sigma \longrightarrow_{\wedge CC} wrong^\rho$.
  - If M^τ is a value and Π^σ is not a parallel value, then since $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$, by context $v^\tau\ E$, $M^\tau\ \Pi^\sigma \longrightarrow_{\wedge CC} M^\tau\ \Upsilon^\sigma$.
  - If M^τ is a *wrong*, by rule [E-Wrong], $M^\tau\ \Pi^\sigma \longrightarrow_{\wedge CC} wrong^\rho$.
  - If M^τ is not a value or *wrong*, then $M^\tau \longrightarrow_{\wedge CC} N^\tau$, and by context $E\ \Pi^\sigma$, $M^\tau\ \Pi^\sigma \longrightarrow_{\wedge CC} N^\tau\ \Pi^\sigma$.
- Rule [T-Add]. If $\emptyset \vdash_{\wedge CC} M_1^{Int} + M_2^{Int} : Int$ then by rule [T-Add], we have that $\emptyset \vdash_{\wedge CC} M_1^{Int} : Int$ and $\emptyset \vdash_{\wedge CC} M_2^{Int}$:

Int. By the induction hypothesis M_1^{Int} is either a value or $wrong$ or $\exists N_1^{Int}$ such that $M_1^{Int} \longrightarrow_{\wedge CC} N_1^{Int}$, and also by the induction hypothesis M_2^{Int} is either a value or $wrong$ or $\exists N_2^{Int}$ such that $M_2^{Int} \longrightarrow_{\wedge CC} N_2^{Int}$. There are several possibilities:

- If M_1^{Int} is a value and M_2^{Int} is also a value, then M_1^{Int} is a constant k_1^{Int} and M_2^{Int} is a constant k_2^{Int} and therefore, by rule [E-Add], we have that $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} k^{Int}$.
- If M_1^{Int} is a $wrong$, then by rule [E-Wrong], we have that $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} wrong^{Int}$.
- If M_1^{Int} is neither a value or a $wrong$ and M_2^{Int} is not a $wrong$ then $M_1^{Int} \longrightarrow_{\wedge CC} N_1^{Int}$, and by context $E + M_2^{Int}$, $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} N_1^{Int} + M_2^{Int}$.
- If M_1^{Int} is not a $wrong$ and M_2^{Int} is a $wrong$, then by rule [E-Wrong], we have that $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} wrong^{Int}$.
- If M_1^{Int} is a value and M_2^{Int} is neither a value or a $wrong$ then $M_2^{Int} \longrightarrow_{\wedge CC} N_2^{Int}$, and by context $v^{Int} + E$, $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} M_1^{Int} + N_2^{Int}$.

- Rule [T-Par]. If $\emptyset \vdash_{\wedge CC} M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} : \tau_1 \wedge \ldots \wedge \tau_n$ then by rule [T-Par], we have that $\emptyset \vdash_{\wedge CC} M_1^{\tau_1} : \tau_1$ and ... and $\emptyset \vdash_{\wedge CC} M_n^{\tau_n} : \tau_n$. By the induction hypothesis, we have that either $M_1^{\tau_1}$ is a value or $wrong^{\tau_1}$ or $\exists N_1^{\tau_1}$ such that $M_1^{\tau_1} \longrightarrow_{\wedge CC} N_1^{\tau_1}$ and ... and we have that either $M_n^{\tau_n}$ is a value or $wrong^{\tau_n}$ or $\exists N_n^{\tau_n}$ such that $M_n^{\tau_n} \longrightarrow_{\wedge CC} N_n^{\tau_n}$. If $M_1^{\tau_1}$ and ... and $M_n^{\tau_n}$ are all values, than $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n}$ is a parallel value. If $M_1^{\tau_1}$ and ... and $M_n^{\tau_n}$ are all results, and $\exists i \,.\, M_i^{\tau_i} = wrong^{\tau_i}$, by rule [E-Push], $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} wrong^{\tau_1 \wedge \ldots \wedge \tau_n}$. Otherwise, by rule [E-Par], we have that $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$.
- Rule [T-Cast]. If $\emptyset \vdash_{\wedge CC} M^{\tau} : \tau \Rightarrow \rho : \rho$ then by rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} M^{\tau} : \tau$. By the induction hypothesis, M^{τ} is either a value or a $wrong$ or $\exists N^{\tau}$ such that $M^{\tau} \longrightarrow_{\wedge CC} N^{\tau}$. If M^{τ} is a value, and $M^{\tau} : \tau \Rightarrow \rho$ is of the form $M^{\tau} : G \Rightarrow Dyn$, or of the form $M^{\tau} : \sigma_1 \rightarrow \tau_1 \Rightarrow \sigma_2 \rightarrow \tau_2$, then $M^{\tau} : \tau \Rightarrow \rho$ is a value. Otherwise, by rules [EC-Identity], [EC-Succeed], [EC-Fail], [EC-Ground] or [EC-Expand], we have that $M^{\tau} : \tau \Rightarrow \rho \longrightarrow_{\wedge CC} M'^{\rho}$. If M^{τ} is a $wrong$ then by rule [E-Wrong], we have that $M^{\tau} : \tau \Rightarrow \rho \longrightarrow_{\wedge CC} wrong^{\rho}$. If M^{τ} is not a value or a $wrong$, then by context $E : \tau \Rightarrow \rho$, $M^{\tau} : \tau \Rightarrow \rho \longrightarrow_{\wedge CC} N^{\tau} : \tau \Rightarrow \rho$.

□

Lemma 6.5 (Extra Cast on the Left). *If* $\emptyset \vdash_{\wedge CC} v_1^{\tau_1} : \tau_1$, $\emptyset \vdash_{\wedge CC} v_2^{\tau_2} : \tau_2$, $v_2^{\tau_2} \sqsubseteq v_1^{\tau_1}$ *and* $\tau_2 \sqsubseteq \tau_1$ *and* $\tau_3 \sqsubseteq \tau_1$ *then* $v_2^{\tau_2} : \tau_2 \Rightarrow \tau_3 \longrightarrow^*_{\wedge CC} v_3^{\tau_3}$ *and* $v_3^{\tau_3} \sqsubseteq v_1^{\tau_1}$.

Proof. We proceed by case analysis on τ_2 and τ_3:

- Both τ_2 and τ_3 are the same. If $v_2^{\tau_2} \sqsubseteq v_1^{\tau_1}$ and $\tau_2 \sqsubseteq \tau_1$ and $\tau_2 \sqsubseteq \tau_1$ then by rule [EC-Identity], $v_2^{\tau_2} : \tau_2 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} v_2^{\tau_2}$ and $v_2^{\tau_2} \sqsubseteq v_1^{\tau_1}$.
- τ_2 is a base type B and $\tau_3 = Dyn$. If $v_2^B \sqsubseteq v_1^{\tau_1}$ and $B \sqsubseteq \tau_1$ and $Dyn \sqsubseteq \tau_1$ then $v_2^B : B \Rightarrow Dyn$ is a value, so $v_2^B : B \Rightarrow Dyn \longrightarrow^0_{\wedge CC} v_2^B : B \Rightarrow Dyn$ and by rule [P-CastL], $v_2^B : B \Rightarrow Dyn \sqsubseteq v_1^{\tau_1}$.
- $\tau_2 = Dyn$ and τ_3 is a base type B. If $v_2^{Dyn} \sqsubseteq v_1^{\tau_1}$ and $Dyn \sqsubseteq \tau_1$ and $B \sqsubseteq \tau_1$, by definition 4.3, $\tau_1 = B$. If $\tau_1 = B$ and $v_1^{\tau_1}$ is a value, then $v_1^{\tau_1}$ must be a constant k^B, according to the definition of values in section 6. By rule [P-CastL] and [P-Con], we have that $v_2^{Dyn} = v_2'^B : B \rightarrow Dyn$, and $v_2'^B \sqsubseteq r_1^B$. By rule [EC-Succeed], we have that $v_2'^B : B \rightarrow Dyn : Dyn \rightarrow B \longrightarrow_{\wedge CC} v_2'^B$.
- $\tau_2 = \tau_2' \rightarrow \tau_2''$ and $\tau_3 = Dyn$. If $v_2^{\tau_2' \rightarrow \tau_2''} \sqsubseteq v_1^{\tau_1}$ and $\tau_2' \rightarrow \tau_2'' \sqsubseteq \tau_1$ and $Dyn \sqsubseteq \tau_1$ then there are two possibilities:
  - $\tau_2' \rightarrow \tau_2'' = G$. Then $v_2^G : G \Rightarrow Dyn$ is a value and therefore $v_2^G : G \Rightarrow Dyn \longrightarrow^0_{\wedge CC} v_2^G : G \Rightarrow Dyn$ and by rule [P-CastL], $v_2^G : G \Rightarrow Dyn \sqsubseteq v_1^{\tau_1}$.
  - $\tau_2' \rightarrow \tau_2'' \neq G$. Then by rule [EC-Ground], $v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow Dyn \longrightarrow_{\wedge CC} v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow G : G \Rightarrow Dyn$. As $\tau_2' \rightarrow \tau_2'' \sqsubseteq \tau_1$ then $G \sqsubseteq \tau_1$, and by rule [P-CastL], we have that $v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow G \sqsubseteq v_1^{\tau_1}$. By rule [P-CastL], we have that $v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow G : G \Rightarrow Dyn \sqsubseteq v_1^{\tau_1}$.
- $\tau_2 = Dyn$ and $\tau_3 = \tau_3' \rightarrow \tau_3''$. If $v_2^{Dyn} \sqsubseteq v_1^{\tau_1}$ and $Dyn \sqsubseteq \tau_1$ and $\tau_3' \rightarrow \tau_3'' \sqsubseteq \tau_1$ then there are two possibilities:
  - $\tau_3' \rightarrow \tau_3'' = G$. By definition 4.3, we have that τ_1 is an arrow type. By the definition of values in section 6, $v_1^{\tau_1}$ is a λ-abstraction, possibly with several casts. Therefore, since $v_2^{Dyn} \sqsubseteq v_1^{\tau_1}$, v_2^{Dyn} is also a λ-abstraction, possibly with several casts. Then, according to the definition of values in section 6, we have that $v_2^{Dyn} = v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn$. There are three possibilities:
    * By rule [P-Cast], we have that $v_1^{\tau_1} = v_1'^{\tau_1'} : \tau_1' \Rightarrow \tau_1$ such that $v_2'^{\tau_3' \rightarrow \tau_3''} \sqsubseteq v_1'^{\tau_1'}$, where $\tau_3' \rightarrow \tau_3'' \sqsubseteq \tau_1'$ and $\tau_3' \rightarrow \tau_3'' \sqsubseteq \tau_1$. By rule [EC-Succeed], we have that $v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow_{\wedge CC} v_2'^{\tau_3' \rightarrow \tau_3''}$. By rule [P-CastR], we have that $v_2'^{\tau_3' \rightarrow \tau_3''} \sqsubseteq v_1'^{\tau_1'} : \tau_1' \Rightarrow \tau_1$.
    * By rule [P-CastL], $v_2'^{\tau_3' \rightarrow \tau_3''} \sqsubseteq v_1^{\tau_1}$. By rule [EC-Succeed], we have that $v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow_{\wedge CC} v_2'^{\tau_3' \rightarrow \tau_3''}$.
    * By rule [P-CastR], we have that $v_1^{\tau_1} = v_1'^{\tau_1'} : \tau_1' \Rightarrow \tau_1$ such that $v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn \sqsubseteq v_1'^{\tau_1'}$ and $Dyn \sqsubseteq \tau_1'$ and $Dyn \sqsubseteq \tau_1$. Since we have that $\tau_3' \rightarrow \tau_3'' \sqsubseteq \tau_1$, and in order for $v_1'^{\tau_1'} : \tau_1' \Rightarrow \tau_1$ to be a value, we have that $\tau_3' \rightarrow \tau_3'' \sqsubseteq \tau_1'$. By rule [EC-Succeed], we have that $v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow_{\wedge CC} v_2'^{\tau_3' \rightarrow \tau_3''}$. By rule [P-CastR], we have that $v_2'^{\tau_3' \rightarrow \tau_3''} \sqsubseteq v_1'^{\tau_1'} : \tau_1' \Rightarrow \tau_1$.
  - $\tau_3' \rightarrow \tau_3'' \neq G$. Then by rule [EC-Expand], $v_2^{Dyn} : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow_{\wedge CC} v_2^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau_3' \rightarrow \tau_3''$. As $\tau_3' \rightarrow \tau_3'' \sqsubseteq \tau_1$ then $G \sqsubseteq \tau_1$, and by rule [P-CastL], we

have that $v_2^{Dyn} : Dyn \Rightarrow G \sqsubseteq v_1^{\tau_1}$. By rule [P-CastL], we have that $v_2^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau_3' \to \tau_3'' \sqsubseteq v_1^{\tau_1}$.

- $\tau_2 = \tau_2' \to \tau_2''$ and $\tau_3 = \tau_3' \to \tau_3''$. If $v_2^{\tau_2' \to \tau_2''} \sqsubseteq v_1^{\tau_1}$ and $\tau_2' \to \tau_2'' \sqsubseteq \tau_1$ and $\tau_3' \to \tau_3'' \sqsubseteq \tau_1$ then $v_2^{\tau_2' \to \tau_2''} : \tau_2' \to \tau_2'' \Rightarrow \tau_3' \to \tau_3''$ is a value, and therefore $v_2^{\tau_2' \to \tau_2''} : \tau_2' \to \tau_2'' \Rightarrow \tau_3' \to \tau_3'' \longrightarrow^0_{\wedge CC} v_2^{\tau_2' \to \tau_2''} : \tau_2' \to \tau_2'' \Rightarrow \tau_3' \to \tau_3''$. By rule [P-CastL], we have that $v_2^{\tau_2' \to \tau_2''} : \tau_2' \to \tau_2'' \Rightarrow \tau_3' \to \tau_3'' \sqsubseteq v_1^{\tau_1}$.

□

Lemma 6.6 (Catchup to Value on the Right). *If* $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ *and* $\emptyset \vdash_{\wedge CC} M^\rho : \rho$ *and* $M^\rho \sqsubseteq v^\tau$ *then* $M^\rho \longrightarrow^*_{\wedge CC} v'^\rho$ *and* $v'^\rho \sqsubseteq v^\tau$.

Proof. We proceed by induction on the length of the derivation tree of $M^\rho \sqsubseteq v^\tau$.

Base cases:

- Rule [P-Con]. If $\emptyset \vdash_{\wedge CC} k^B : B$ and $\emptyset \vdash_{\wedge CC} k^B : B$ and $k^B \sqsubseteq k^B$ then, since k^B is a value, $k^B \longrightarrow^0_{\wedge CC} k^B$ and $k^B \sqsubseteq k^B$.
- Rule [P-Abs]. If $\emptyset \vdash_{\wedge CC} \lambda x : v \,.\, N^\rho : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ and $\lambda x : \sigma \,.\, M^\tau \sqsubseteq \lambda x : v \,.\, N^\rho$ then, since $\lambda x : \sigma \,.\, M^\tau$ is a value, $\lambda x : \sigma \,.\, M^\tau \longrightarrow^0_{\wedge CC} \lambda x : \sigma \,.\, M^\tau$ and $\lambda x : \sigma \,.\, M^\tau \sqsubseteq \lambda x : v \,.\, N^\rho$.

Induction step:

- Rule [P-Cast]. If $\emptyset \vdash_{\wedge CC} v^{\tau_1} : \tau_1 \Rightarrow \tau_2 : \tau_2$ and $\emptyset \vdash_{\wedge CC} N^{\rho_1} : \rho_1 \Rightarrow \rho_2 : \rho_2$ and $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$ then by rule [P-Cast], we have that $N^{\rho_1} \sqsubseteq v^{\tau_1}$ and $\rho_1 \sqsubseteq \tau_1$ and $\rho_2 \sqsubseteq \tau_2$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} v'^{\rho_1}$ and $v'^{\rho_1} \sqsubseteq v^{\tau_1}$. By rule [E-Ctx] and context $E : \tau \Rightarrow \rho$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [P-Cast], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$. Since $v^{\tau_1} : \tau_1 \Rightarrow \tau_2$ is a value, then either $\tau_1 = G$ and $\tau_2 = Dyn$ or $\tau_1 = \tau_1' \to \tau_1''$ and $\tau_2 = \tau_2' \to \tau_2''$. If $\tau_1 = G$ and $\tau_2 = Dyn$ then there are two possibilities:
  - Both ρ_1 and ρ_2 are Dyn. Then, we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow_{\wedge CC} v'^{\rho_1}$ and by rule [P-CastL], $v'^{\rho_1} \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$.
  - $\rho_1 = G$ and $\rho_2 = Dyn$. Therefore, $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$ is a value.

  If $\tau_1 = \tau_1' \to \tau_1''$ and $\tau_2 = \tau_2' \to \tau_2''$ then there are four possibilities:
  - Both ρ_1 and ρ_2 are the same. Then, we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow_{\wedge CC} v'^{\rho_1}$ and by rule [P-CastL], $v'^{\rho_1} \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$.
  - $\rho_1 = \rho_1' \to \rho_1''$ and $\rho_2 = Dyn$, with $\rho_1' \to \rho_1'' \neq G$. Therefore, by rule [E-Ground], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow G : G \Rightarrow \rho_2$. By rule [P-CastR], we have that $v'^{\rho_1} : \rho_1 \Rightarrow G \sqsubseteq v^{\tau_1}$ and by rule [P-Cast], we have that $v'^{\rho_1} : \rho_1 \Rightarrow G : G \Rightarrow \rho_2 \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$.
  - $\rho_1 = Dyn$ and $\rho_2 = \rho_2' \to \rho_2''$, with $\rho_2' \to \rho_2'' \neq G$. Therefore, by rule [E-Expand], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow G : G \Rightarrow \rho_2$. By rule [P-Cast], we have that $v'^{\rho_1} : \rho_1 \Rightarrow G \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$ and by rule [P-CastL], we have that $v'^{\rho_1} : \rho_1 \Rightarrow G : G \Rightarrow \rho_2 \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$.
  - $\rho_1 = \rho_1' \to \rho_1''$ and $\rho_2 = \rho_2' \to \rho_2''$. Therefore, $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$ is a value.
- Rule [P-CastL]. If $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} N^{\rho_1} : \rho_1 \Rightarrow \rho_2 : \rho_2$ and $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau$ then by rule [P-CastL], we have that $N^{\rho_1} \sqsubseteq v^\tau$ and $\rho_1 \sqsubseteq \tau$ and $\rho_2 \sqsubseteq \tau$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} v'^{\rho_1}$ and $v'^{\rho_1} \sqsubseteq v^\tau$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$, and by rule [P-CastL], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau$. By lemma 6.5, we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v''^{\rho_2}$ and $v''^{\rho_2} \sqsubseteq v^\tau$.
- Rule [P-CastR]. If $\emptyset \vdash_{\wedge CC} v^{\tau_1} : \tau_1 \Rightarrow \tau_2 : \tau_2$ and $\emptyset \vdash_{\wedge CC} N^\rho : \rho$ and $N^\rho \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$ then by rule [P-CastR], we have that $N^\rho \sqsubseteq v^{\tau_1}$ and $\rho \sqsubseteq \tau_1$ and $\rho \sqsubseteq \tau_2$. By the induction hypothesis, we have that $N^\rho \longrightarrow^*_{\wedge CC} v'^\rho$ and $v'^\rho \sqsubseteq v^{\tau_1}$. By rule [P-CastR], we have that $v'^\rho \sqsubseteq v^{\tau_1} : \tau_1 \Rightarrow \tau_2$.

□

Lemma 6.7 (Simulation of Function Application). *Assume* $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ *and* $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} v'^{v \to \rho} : v \to \rho$ *and* $\emptyset \vdash_{\wedge CC} \pi'^v : v$ *and* $v \to \rho \sqsubseteq \sigma \to \tau$. *If* $v'^{v \to \rho} \sqsubseteq \lambda x : \sigma \,.\, M^\tau$ *and* $\pi'^v \sqsubseteq \pi^\sigma$ *then* $v'^{v \to \rho}\ \pi'^v \longrightarrow^*_{\wedge CC} M'^\rho$, $M'^\rho \sqsubseteq [c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\ M^\tau$ *and* $\emptyset \vdash_{\wedge CC} M'^\rho : \rho$.

Proof. We proceed by induction on the length of the derivation tree of $v'^{v \to \rho} \sqsubseteq \lambda x : \sigma \,.\, M^\tau$. [1]

Base cases:

- Rule [P-Abs]. We assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} \lambda x : v \,.\, N^\rho : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^v : v$ and $v \to \rho \sqsubseteq \sigma \to \tau$. If $\lambda x : v \,.\, N^\rho \sqsubseteq \lambda x : \sigma \,.\, M^\tau$ and $\pi'^v \sqsubseteq \pi^\sigma$, then by rule [E-Beta], we have that $(\lambda x : v \,.\, N^\rho)\ \pi'^v \longrightarrow_{\wedge CC} [c_i^{\rho'}(x) \mapsto \langle \pi'^v \rangle_i^{\rho'}]\ N^\rho$, and $[c_i^{\rho'}(x) \mapsto \langle \pi'^v \rangle_i^{\rho'}]\ N^\rho \sqsubseteq [c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\ M^\tau$ and $\emptyset \vdash_{\wedge CC} [c_i^{\rho'}(x) \mapsto \langle \pi'^v \rangle_i^{\rho'}]\ N^\rho : \rho$.

Induction step:

- Rule [P-CastL]. We assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} v'^{v' \to \rho'} : v' \to \rho' \Rightarrow v \to \rho : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^v : v$ and $v \to \rho \sqsubseteq \sigma \to \tau$. If $v'^{v' \to \rho'} : v' \to \rho' \Rightarrow v \to \rho \sqsubseteq \lambda x : \sigma \,.\, M^\tau$ and $\pi'^v \sqsubseteq \pi^\sigma$, then by rule [P-CastL], we have that $v'^{v' \to \rho'} \sqsubseteq \lambda x : \sigma \,.\, M^\tau$ and $v' \to \rho' \sqsubseteq \sigma \to \tau$ and $v \to \rho \sqsubseteq \sigma \to \tau$, and by definition 4.3, we have that $v' \sqsubseteq \sigma$ and $v \sqsubseteq \sigma$ and $\rho' \sqsubseteq \tau$ and $\rho \sqsubseteq \tau$. By rule [EC-Application], we have that $(v'^{v' \to \rho'} : v' \to \rho' \Rightarrow v \to \rho)\ \pi'^v \longrightarrow_{\wedge CC} (v'^{v' \to \rho'}\ (\pi'^v : v \Rightarrow_\wedge v')) : \rho' \Rightarrow \rho$. By rule [P-Par] and rule [P-CastL], we have that $\pi'^v : v \Rightarrow_\wedge v' \sqsubseteq \pi^\sigma$. By the induction hypothesis, we have that $(v'^{v' \to \rho'}\ (\pi'^v : v \Rightarrow_\wedge v')) \longrightarrow^*_{\wedge CC} N^{\rho'}$ and $N^{\rho'} \sqsubseteq [c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\ M^\tau$ and $\emptyset \vdash_{\wedge CC} N^{\rho'} : \rho'$. By rule [E-Ctx] and context $E : \rho' \Rightarrow \rho$, we have that $(v'^{v' \to \rho'}\ (\pi'^v : v \Rightarrow_\wedge v')) : \rho' \Rightarrow \rho \longrightarrow^*_{\wedge CC} N^{\rho'} : \rho' \Rightarrow \rho$. By rule [P-CastL], we have that

[1] This lemma is used in the proof of Lemma 6.9, in rule [T-App], case rule [E-Beta]. According to rule [E-Beta], π^σ is not *wrong*, and since $\pi'^v \sqsubseteq \pi^\sigma$, π'^v is also not *wrong*.

$N^{\rho'} : \rho' \Rightarrow \rho \sqsubseteq [c_i^{\tau'}(x) \mapsto \langle \pi^{\sigma} \rangle_i^{\tau'}]\ M^{\tau}$ and by rule [T-Cast], we have that $\emptyset \vdash_{\wedge CC} N^{\rho'} : \rho' \Rightarrow \rho : \rho$.

$\rho \Rightarrow \rho' \longrightarrow^*_{\wedge CC} M^{\rho} : \rho \Rightarrow \rho'$. By rule [P-CastL], we have that $M^{\rho} : \rho \Rightarrow \rho' \sqsubseteq v^{\sigma \to \tau}\ (\pi^{\sigma} : \sigma' \Rightarrow_{\wedge} \sigma) : \tau \Rightarrow \tau'$.

□

Lemma 6.8 (Simulation of Unwrapping). *Assume* $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau$ *and* $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{\upsilon \to \rho} : \upsilon \to \rho$ *and* $\emptyset \vdash_{\wedge CC} \pi'^{\upsilon} : \upsilon$ *and* $\upsilon \to \rho \sqsubseteq \sigma \to \tau$. *If* $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ *and* $\pi'^{\upsilon} \sqsubseteq \pi^{\sigma'}$ *then* $v'^{\upsilon \to \rho}\ \pi'^{\upsilon} \longrightarrow^*_{\wedge CC} M^{\rho}$ *and* $M^{\rho} \sqsubseteq v^{\sigma \to \tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma) : \tau \Rightarrow \tau'$.

Proof. We proceed by induction on the length of the derivation tree of $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$. [2]

Base cases:

- Rule [P-Cast]. We assume $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{\upsilon \to \rho} : \upsilon \to \rho \Rightarrow \upsilon' \to \rho' : \upsilon' \to \rho'$ and $\emptyset \vdash_{\wedge CC} \pi'^{\upsilon'} : \upsilon'$ and $\upsilon' \to \rho' \sqsubseteq \sigma \to \tau$. If $v'^{\upsilon \to \rho} : \upsilon \to \rho \Rightarrow \upsilon' \to \rho' \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ and $\pi'^{\upsilon'} \sqsubseteq \pi^{\sigma'}$ then by rule [P-Cast], we have that $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau}$ and $\upsilon \to \rho \sqsubseteq \sigma \to \tau$ and $\upsilon' \to \rho' \sqsubseteq \sigma' \to \tau'$. By rule [EC-Application], we have that $(v'^{\upsilon \to \rho} : \upsilon \to \rho \Rightarrow \upsilon' \to \rho')\ \pi'^{\upsilon'} \longrightarrow_{\wedge CC} (v'^{\upsilon \to \rho}\ (\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon)) : \rho \Rightarrow \rho'$. Since $\upsilon' \sqsubseteq \sigma'$ and $\upsilon \sqsubseteq \sigma$, by rules [P-Par] and [P-Cast] we have that $\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon \sqsubseteq \pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma$. Since $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau}$, by rule [P-App], we have that $v'^{\upsilon \to \rho}\ (\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon) \sqsubseteq v^{\sigma \to \tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma)$. Since $\rho \sqsubseteq \tau$ and $\rho' \sqsubseteq \tau'$, by rule [P-Cast], we have that $(v'^{\upsilon \to \rho}\ (\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon)) : \rho \Rightarrow \rho' \sqsubseteq (v^{\sigma \to \tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma)) : \tau \Rightarrow \tau'$.
- Rule [P-CastR]. We assume $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{\upsilon \to \rho} : \upsilon \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^{\upsilon} : \upsilon$ and $\upsilon \to \rho \sqsubseteq \sigma \to \tau$. If $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ and $\pi'^{\upsilon} \sqsubseteq \pi^{\sigma'}$ then by rule [P-CastR], we have that $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau}$ and $\upsilon \to \rho \sqsubseteq \sigma \to \tau$ and $\upsilon \to \rho \sqsubseteq \sigma' \to \tau'$. Since $v'^{\upsilon \to \rho}$ and π'^{υ} are values, we have that $v'^{\upsilon \to \rho}\ \pi'^{\upsilon} \longrightarrow^0_{\wedge CC} v'^{\upsilon \to \rho}\ \pi'^{\upsilon}$. By rule [P-CastR], we have that $\pi'^{\upsilon} \sqsubseteq \pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma$. By rule [P-App], we have that $v'^{\upsilon \to \rho}\ \pi'^{\upsilon} \sqsubseteq v^{\sigma \to \tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma)$. By rule [P-CastR], we have that $v'^{\upsilon \to \rho}\ \pi'^{\upsilon} \sqsubseteq (v^{\sigma \to \tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_{\wedge} \sigma)) : \tau \Rightarrow \tau'$.

Induction step:

- Rule [P-CastL]. We assume $\emptyset \vdash_{\wedge CC} v^{\sigma \to \tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{\upsilon \to \rho} : \upsilon \to \rho \Rightarrow \upsilon' \to \rho' : \upsilon' \to \rho'$ and $\emptyset \vdash_{\wedge CC} \pi'^{\upsilon'} : \upsilon'$ and $\upsilon' \to \rho' \sqsubseteq \sigma \to \tau$. If $v'^{\upsilon \to \rho} : \upsilon \to \rho \Rightarrow \upsilon' \to \rho' \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ and $\pi'^{\upsilon'} \sqsubseteq \pi^{\sigma'}$ then by rule [P-CastL], we have that $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ and $\upsilon \to \rho \sqsubseteq \sigma' \to \tau'$ and $\upsilon' \to \rho' \sqsubseteq \sigma' \to \tau'$. By rule [EC-Application], we have that $(v'^{\upsilon \to \rho} : \upsilon \to \rho \Rightarrow \upsilon' \to \rho')\ \pi'^{\upsilon'} \longrightarrow_{\wedge CC} (v'^{\upsilon \to \rho}\ (\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon)) : \rho \Rightarrow \rho'$. Since $v'^{\upsilon \to \rho} \sqsubseteq v^{\sigma \to \tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau'$ and $\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon \sqsubseteq \pi^{\sigma'}$, by the induction hypothesis, we have that $v'^{\upsilon \to \rho}\ (\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon) \longrightarrow^*_{\wedge CC} M^{\rho}$ and $M^{\rho} \sqsubseteq v^{\sigma \to \tau}\ (\pi^{\sigma} : \sigma' \Rightarrow_{\wedge} \sigma) : \tau \Rightarrow \tau'$. By rule [E-Ctx] and context $E : \rho \Rightarrow \rho'$, we have that $(v'^{\upsilon \to \rho}\ (\pi'^{\upsilon'} : \upsilon' \Rightarrow_{\wedge} \upsilon)) :$

Lemma 6.9 (Simulation of More Precise Programs). *For all* $\Upsilon_1^{\upsilon} \sqsubseteq \Pi_1^{\sigma}$ *such that* $\emptyset \vdash_{\wedge CC} \Pi_1^{\sigma} : \sigma$ *and* $\emptyset \vdash_{\wedge CC} \Upsilon_1^{\upsilon} : \upsilon$, *if* $\Pi_1^{\sigma} \longrightarrow_{\wedge CC} \Pi_2^{\sigma}$ *then* $\Upsilon_1^{\upsilon} \longrightarrow^*_{\wedge CC} \Upsilon_2^{\upsilon}$ *and* $\Upsilon_2^{\upsilon} \sqsubseteq \Pi_2^{\sigma}$.

Proof. We proceed by induction on the length of the derivation tree of $\Upsilon_1^{\upsilon} \sqsubseteq \Pi_1^{\sigma}$, followed by case analysis on $\Pi_1^{\sigma} \longrightarrow_{\wedge CC} \Pi_2^{\sigma}$, and using lemmas 6.5, 6.6, 6.7 and 6.8, and theorems 6.3 and 6.4.

Base cases:

- Rule [P-Con]. If $k^B \sqsubseteq k^B$, and since k^B is a value, then it is proved.
- Rule [P-Wrong]. If $\Pi^{\upsilon} \sqsubseteq wrong^{\sigma}$ and $wrong^{\sigma} \longrightarrow_{\wedge CC} wrong^{\sigma}$, then by rule [P-Wrong], we have that $\upsilon \sqsubseteq \sigma$. By theorems 6.3 and 6.4, any amount of evaluation steps, say $\Pi^{\upsilon} \longrightarrow^*_{\wedge CC} \Upsilon^{\upsilon}$, yields an expression Υ^{υ} with type υ. By rule [P-Wrong], we have that $\Upsilon^{\upsilon} \sqsubseteq wrong^{\sigma}$.

Induction step:

- Rule [P-Abs]. If $\lambda x : \sigma\ .\ M^{\tau} \sqsubseteq \lambda x : \upsilon\ .\ N^{\rho}$, and since both $\lambda x : \sigma\ .\ M^{\tau}$ and $\lambda x : \upsilon\ .\ N^{\rho}$ are values, then it is proved.
- Rule [P-App]. There are six possibilities:
  - Rule [E-Beta]. If $M^{\tau}\ \Pi^{\sigma} \sqsubseteq (\lambda x : \upsilon\ .\ N'^{\rho'})^{\rho}\ \pi^{\upsilon}$ and $(\lambda x : \upsilon\ .\ N'^{\rho'})^{\rho}\ \pi^{\upsilon} \longrightarrow_{\wedge CC} [c_i^{\rho''}(x) \mapsto \langle \pi^{\upsilon} \rangle_i^{\rho''}]\ N'^{\rho'}$, then by rule [P-App], we have that $M^{\tau} \sqsubseteq (\lambda x : \upsilon\ .\ N'^{\rho'})^{\rho}$ and $\Pi^{\sigma} \sqsubseteq \pi^{\upsilon}$. By lemma 6.6, we have that $M^{\tau} \longrightarrow^*_{\wedge CC} v'^{\tau}$ and $v'^{\tau} \sqsubseteq (\lambda x : \upsilon\ .\ N'^{\rho'})^{\rho}$. By applying lemma 6.6 to each component of Π^{σ}, and then by rule [E-Par], we have that $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} \pi'^{\sigma}$ and $\pi'^{\sigma} \sqsubseteq \pi^{\upsilon}$. By applying rule [E-Ctx] with context $E\ \Pi^{\sigma}$ and then with context $v'^{\tau}\ E$, we have that $M^{\tau}\ \Pi^{\sigma} \longrightarrow^*_{\wedge CC} v'^{\tau}\ \Pi^{\sigma}$, and $v'^{\tau}\ \Pi^{\sigma} \longrightarrow^*_{\wedge CC} v'^{\tau}\ \pi'^{\sigma}$. By lemma 6.7, we have that $v'^{\tau}\ \pi'^{\sigma} \longrightarrow^*_{\wedge CC} M'^{\tau'}$ and $M'^{\tau'} \sqsubseteq [c_i^{\rho''}(x) \mapsto \langle \pi^{\upsilon} \rangle_i^{\rho''}]\ N'^{\rho'}$.
  - Rule [E-Ctx] and context $E\ \Upsilon^{\upsilon}$. If $M^{\tau}\ \Pi^{\sigma} \sqsubseteq N^{\rho}\ \Upsilon^{\upsilon}$ and $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow_{\wedge CC} N'^{\rho}\ \Upsilon^{\upsilon}$, then by rule [P-App], we have that $M^{\tau} \sqsubseteq N^{\rho}$ and $\Pi^{\sigma} \sqsubseteq \Upsilon^{\upsilon}$, and by rule [E-Ctx], we have that $N^{\rho} \longrightarrow_{\wedge CC} N'^{\rho}$. By the induction hypothesis, we have that $M^{\tau} \longrightarrow^*_{\wedge CC} M'^{\tau}$ and $M'^{\tau} \sqsubseteq N'^{\rho}$. By rule [E-Ctx], we have that $M^{\tau}\ \Pi^{\sigma} \longrightarrow^*_{\wedge CC} M'^{\tau}\ \Pi^{\sigma}$, and by rule [P-App], we have that $M'^{\tau}\ \Pi^{\sigma} \sqsubseteq N'^{\rho}\ \Upsilon^{\upsilon}$.
  - Rule [E-Ctx] and context $v^{\rho}\ E$. If $M^{\tau}\ \Pi^{\sigma} \sqsubseteq N^{\rho}\ \Upsilon^{\upsilon}$ and $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow_{\wedge CC} N^{\rho}\ \Upsilon'^{\upsilon}$, then by rule [P-App], we have that $M^{\tau} \sqsubseteq N^{\rho}$ and $\Pi^{\sigma} \sqsubseteq \Upsilon^{\upsilon}$ and by rule [E-Ctx], we have that $\Upsilon^{\upsilon} \longrightarrow_{\wedge CC} \Upsilon'^{\upsilon}$. By the induction hypothesis, we have that $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} \Pi'^{\sigma}$ and $\Pi'^{\sigma} \sqsubseteq \Upsilon'^{\upsilon}$. By rule [E-Ctx], we have that $M^{\tau}\ \Pi^{\sigma} \longrightarrow^*_{\wedge CC} M^{\tau}\ \Pi'^{\sigma}$, and by rule [P-App], we have that $M^{\tau}\ \Pi'^{\sigma} \sqsubseteq N^{\rho}\ \Upsilon'^{\upsilon}$.
  - Rule [E-Wrong] and context $E\ \Upsilon^{\upsilon}$ or $v^{\rho}\ E$. If $M^{\tau}\ \Pi^{\sigma} \sqsubseteq N^{\rho}\ \Upsilon^{\upsilon}$ and $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow_{\wedge CC} wrong^{\rho'}$, by rule [P-App], we have that $M^{\tau} \sqsubseteq N^{\rho}$ and $\Pi^{\sigma} \sqsubseteq \Upsilon^{\upsilon}$. By definition 5.1, we have that $\tau \sqsubseteq \rho$, where $\rho = \upsilon \to \rho'$ and $\tau = \sigma \to \tau'$, and therefore $\tau' \sqsubseteq \rho'$. By theorems 6.3 and 6.4, $M^{\tau}\ \Pi^{\sigma} \longrightarrow^*_{\wedge CC} M'^{\tau'}$, and by rule [P-Wrong], $M'^{\tau'} \sqsubseteq wrong^{\rho'}$.

[2]This lemma is used in the proof of Lemma 6.9, in rule [T-App], case rule [EC-Application]. According to rule [EC-Application], $\pi^{\sigma'}$ is not *wrong*, and since $\pi'^{\upsilon} \sqsubseteq \pi^{\sigma'}$, π'^{υ} is also not *wrong*.

  - Rule [EC-Application]. If $M^\tau\ \Pi^\sigma \sqsubseteq (v^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho)\ \pi^v$ and $(v^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho)\ \pi^v \longrightarrow_{\wedge CC} (v^{v'\to\rho'}\ (\pi^v : v \Rightarrow_\wedge v')) : \rho' \Rightarrow \rho$, then by rule [P-App], we have that $M^\tau \sqsubseteq (v^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho)$ and $\Pi^\sigma \sqsubseteq \pi^v$. By lemma 6.6, we have that $M^\tau \longrightarrow^*_{\wedge CC} v'^\tau$ and $v'^\tau \sqsubseteq (v^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho)$. By applying lemma 6.6 to each component of Π^σ, and then by rule [E-Par], we have that $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi'^\sigma$ and $\pi'^\sigma \sqsubseteq \pi^v$. By applying rule [E-Ctx] with context $E\ \Pi^\sigma$ and then with context $v'^\tau\ E$, we have that $M^\tau\ \Pi^\sigma \longrightarrow^*_{\wedge CC} v'^\tau\ \Pi^\sigma$, and $v'^\tau\ \Pi^\sigma \longrightarrow^*_{\wedge CC} v'^\tau\ \pi'^\sigma$. By lemma 6.8, we have that $v'^\tau\ \pi'^\sigma \longrightarrow^*_{\wedge CC} M'^{\tau'}$ and $M'^{\tau'} \sqsubseteq (v^{v'\to\rho'}\ (\pi^v : v \Rightarrow_\wedge v')) : \rho' \Rightarrow \rho$.
- Rule [P-Add]. There are five possibilities:
  - Rule [E-Add]. If $M_1^{Int} + M_2^{Int} \sqsubseteq k_1^{Int} + k_2^{Int}$ and $k_1^{Int} + k_2^{Int} \longrightarrow_{\wedge CC} k_3^{Int}$ then by rule [P-Add], we have that $M_1^{Int} \sqsubseteq k_1^{Int}$ and $M_2^{Int} \sqsubseteq k_2^{Int}$. By lemma 6.6, we have that $M_1^{Int} \longrightarrow^*_{\wedge CC} v_1^{Int}$ and $v_1^{Int} \sqsubseteq k_1^{Int}$ and $M_2^{Int} \longrightarrow^*_{\wedge CC} v_2^{Int}$ and $v_2^{Int} \sqsubseteq k_2^{Int}$. By definitions 4.3 and 5.1, we have that v_1^{Int} is a constant k_4^{Int} and v_2^{Int} is a constant k_5^{Int}. By rule [E-Ctx], and contexts $E + M^\tau$ and $v^\tau + E$, we have that $M_1^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} k_4^{Int} + M_2^{Int}$ and $k_4^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} k_4^{Int} + k_5^{Int}$. By rule [E-Add], we have that $k_4^{Int} + k_5^{Int} \longrightarrow_{\wedge CC} k_3^{Int}$. By rule [P-Con], we have that $k_3^{Int} \sqsubseteq k_3^{Int}$.
  - Rule [E-Ctx] and context $E + M^\tau$. If $M_1^{Int} + M_2^{Int} \sqsubseteq N_1^{Int} + N_2^{Int}$ and $N_1^{Int} + N_2^{Int} \longrightarrow_{\wedge CC} N_1'^{Int} + N_2^{Int}$, then by rule [P-Add], we have that $M_1^{Int} \sqsubseteq N_1^{Int}$ and $M_2^{Int} \sqsubseteq N_2^{Int}$, and by rule [E-Ctx], we have that $N_1^{Int} \longrightarrow_{\wedge CC} N_1'^{Int}$. By the induction hypothesis, we have that $M_1^{Int} \longrightarrow^*_{\wedge CC} M_1'^{Int}$ and $M_1'^{Int} \sqsubseteq N_1'^{Int}$. By rule [E-Ctx], we have that $M_1^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} M_1'^{Int} + M_2^{Int}$ and by rule [P-Add], we have that $M_1'^{Int} + M_2^{Int} \sqsubseteq N_1'^{Int} + N_2^{Int}$.
  - Rule [E-Ctx] and context $v^\tau + E$. If $M_1^{Int} + M_2^{Int} \sqsubseteq N_1^{Int} + N_2^{Int}$ and $N_1^{Int} + N_2^{Int} \longrightarrow_{\wedge CC} N_1^{Int} + N_2'^{Int}$, then by rule [P-Add], we have that $M_1^{Int} \sqsubseteq N_1^{Int}$ and $M_2^{Int} \sqsubseteq N_2^{Int}$, and by rule [E-Ctx], we have that $N_2^{Int} \longrightarrow_{\wedge CC} N_2'^{Int}$. By the induction hypothesis, we have that $M_2^{Int} \longrightarrow^*_{\wedge CC} M_2'^{Int}$ and $M_2'^{Int} \sqsubseteq N_2'^{Int}$. By rule [E-Ctx], we have that $M_1^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} M_1^{Int} + M_2'^{Int}$ and by rule [P-Add], we have that $M_1^{Int} + M_2'^{Int} \sqsubseteq N_1^{Int} + N_2'^{Int}$.
  - Rule [E-Wrong] and context $E + M^\tau$ or $v^\tau + E$. If $M_1^{Int} + M_2^{Int} \sqsubseteq N_1^{Int} + N_2^{Int}$ and $N_1^{Int} + N_2^{Int} \longrightarrow_{\wedge CC} wrong^{Int}$, then by theorems 6.3 and 6.4, $M_1^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} M^{Int}$, and by rule [P-Wrong], $M^{Int} \sqsubseteq wrong^{Int}$.
- Rule [P-Par]. There are two possibilities:
  - Rule [E-Push]. If $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \sqsubseteq r_1^{\rho_1} \mid \ldots \mid r_n^{\rho_n}$ and $r_1^{\rho_1} \mid \ldots \mid r_n^{\rho_n} \longrightarrow_{\wedge CC} wrong^{\rho_1 \wedge \ldots \wedge \rho_n}$, then by definition 4.5, $M_1^{\tau_1} \sqsubseteq r_1^{\rho_1}$ and $\ldots$ and $M_n^{\tau_n} \sqsubseteq r_n^{\rho_n}$, and by definition 5.1, we have that $\tau_1 \sqsubseteq \rho_1$ and $\ldots$ and $\tau_n \sqsubseteq \rho_n$. By definition 4.3, $\tau_1 \wedge \ldots \wedge \tau_n \sqsubseteq \rho_1 \wedge \ldots \wedge \rho_n$. By theorems 6.3 and 6.4, $M_1^{\tau_1} \longrightarrow^*_{\wedge CC} N_1^{\tau_1}$ and $\ldots$ and $M_n^{\tau_n} \longrightarrow^*_{\wedge CC} N_n^{\tau_n}$. By rule [E-Par], we have that $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow^*_{\wedge CC} N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n}$ and by rule [P-Wrong], we have that $N_1^{\tau_1} \mid \ldots \mid N_n^{\tau_n} \sqsubseteq wrong^{\rho_1 \wedge \ldots \wedge \rho_n}$.
  - Rule [E-Par]. If $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \sqsubseteq N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n}$ and $N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n} \longrightarrow_{\wedge CC} N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n}$, then by rule [P-Par], we have that $M_1^{\tau_1} \sqsubseteq N_1^{\rho_1}$ and $\ldots$ and $M_n^{\tau_n} \sqsubseteq N_n^{\rho_n}$ and by rule [E-Par], $\forall i$. either $N_i^{\rho_i}$ is a result and $N_i^{\rho_i} = N_i'^{\rho_i}$ or $N_i^{\rho_i} \longrightarrow_{\wedge CC} N_i'^{\rho_i}$ and $\exists i$. $N_i^{\rho_i}$ is not a result.

    For all i such that $N_i^{\rho_i}$ is a result, then either $N_i^{\rho_i} = v_i^{\rho_i}$ or $N_i^{\rho_i} = wrong^{\rho_i}$. If $N_i^{\rho_i} = v_i^{\rho_i}$, then by lemma 6.6, we have that $M_i^{\tau_i} \longrightarrow^*_{\wedge CC} v_i'^{\tau_i}$ and $v_i'^{\tau_i} \sqsubseteq v_i^{\rho_i}$ and let $M_i'^{\tau_i} = v_i'^{\tau_i}$. Therefore, $M_i'^{\tau_i} \sqsubseteq N_i'^{\rho_i}$. If $N_i^{\rho_i} = wrong^{\rho_i}$, then by theorems 6.3 and 6.4, $M_i^{\tau_i} \longrightarrow^*_{\wedge CC} M_i'^{\tau_i}$ and by definition 5.1, $M_i'^{\tau_i} \sqsubseteq N_i'^{\rho_i}$.

    For all i such that $N_i^{\rho_i} \longrightarrow_{\wedge CC} N_i'^{\rho_i}$, by the induction hypothesis, we have that $M_i^{\tau_i} \longrightarrow^*_{\wedge CC} M_i'^{\tau_i}$ and $M_i'^{\tau_i} \sqsubseteq N_i'^{\rho_i}$.

    By rule [E-Par], $M_1^{\tau_1} \mid \ldots \mid M_n^{\tau_n} \longrightarrow^*_{\wedge CC} M_1'^{\tau_1} \mid \ldots \mid M_n'^{\tau_n}$, and by rule [P-Par], we have that $M_1'^{\tau_1} \mid \ldots \mid M_n'^{\tau_n} \sqsubseteq N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n}$.
- Rule [P-Cast]. There are seven possibilities:
  - Rule [E-Ctx] and context $E : \tau_1 \Rightarrow \tau_2$. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq M^{\tau_1} : \tau_1 \Rightarrow \tau_2$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} M'^{\tau_1} : \tau_1 \Rightarrow \tau_2$ then by rule [P-Cast], we have that $N^{\rho_1} \sqsubseteq M^{\tau_1}$ and $\rho_1 \sqsubseteq \tau_1$ and $\rho_2 \sqsubseteq \tau_2$, and by rule [E-Ctx], we have that $M^{\tau_1} \longrightarrow_{\wedge CC} M'^{\tau_1}$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} N'^{\rho_1}$ and $N'^{\rho_1} \sqsubseteq M'^{\tau_1}$. By rule [E-Ctx], we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} N'^{\rho_1} : \rho_1 \Rightarrow \rho_2$, and by rule [P-Cast], we have that $N'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq M'^{\tau_1} : \tau_1 \Rightarrow \tau_2$.
  - Rule [E-Wrong] and context $E : \tau_1 \Rightarrow \tau_2$. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq M^{\tau_1} : \tau_1 \Rightarrow \tau_2$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} wrong^{\tau_2}$ then by rule [P-Cast], we have that $N^{\rho_1} \sqsubseteq M^{\tau_1}$ and $\rho_1 \sqsubseteq \tau_1$ and $\rho_2 \sqsubseteq \tau_2$. By theorems 6.3 and 6.4, $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} N'^{\rho_2}$, and by rule [P-Wrong], $N'^{\rho_2} \sqsubseteq wrong^{\tau_2}$.
  - Rule [EC-Identity]. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau : \tau \Rightarrow \tau$ and $v^\tau : \tau \Rightarrow \tau \longrightarrow_{\wedge CC} v^\tau$ then by rule [P-Cast], we have that $N^{\rho_1} \sqsubseteq v^\tau$ and $\rho_1 \sqsubseteq \tau$ and $\rho_2 \sqsubseteq \tau$. By rule [P-CastL], we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau$. By lemma 6.6, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v'^{\rho_2}$ and $v'^{\rho_2} \sqsubseteq v^\tau$.
  - Rule [EC-Succeed]. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^G : G \Rightarrow Dyn : Dyn \Rightarrow G$ and $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \longrightarrow_{\wedge CC} v^G$ then by rule [P-Cast], $N^{\rho_1} \sqsubseteq v^G : G \Rightarrow Dyn$ and $\rho_1 \sqsubseteq Dyn$ and $\rho_2 \sqsubseteq G$. Since $\rho_1 \sqsubseteq Dyn$ then $\rho_1 \sqsubseteq G$. By lemma 6.6, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} v'^{\rho_1}$ and $v'^{\rho_1} \sqsubseteq v^G : G \Rightarrow Dyn$. By rule [P-CastR], $v'^{\rho_1} \sqsubseteq v^G$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [P-CastL], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^G$.
  - Rule [EC-Fail]. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2$ and $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \longrightarrow_{\wedge CC} wrong^{G_2}$ then by rule [P-Cast], $N^{\rho_1} \sqsubseteq v^{G_1} : G_1 \Rightarrow Dyn$

and $\rho_1 \sqsubseteq Dyn$ and $\rho_2 \sqsubseteq G_2$. By theorems 6.3 and 6.4, $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} N'^{\rho_2}$, and by rule [P-Wrong], $N'^{\rho_2} \sqsubseteq wrong^{G_2}$.

- Rule [EC-Ground]. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau : \tau \Rightarrow Dyn$ and $v^\tau : \tau \Rightarrow Dyn \longrightarrow_{\wedge CC} v^\tau : \tau \Rightarrow G : G \Rightarrow Dyn$, then by rule [P-Cast], we have that $N^{\rho_1} \sqsubseteq v^\tau$ and $\rho_1 \sqsubseteq \tau$ and $\rho_2 \sqsubseteq Dyn$. By lemma 6.6, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} v'^{\rho_1}$ and $v'^{\rho_1} \sqsubseteq v^\tau$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$. Since $\rho_2 \sqsubseteq Dyn$ then $\rho_2 \sqsubseteq G$. By rule [P-Cast], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau : \tau \Rightarrow G$, and by rule [P-CastR], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^\tau : \tau \Rightarrow G : G \Rightarrow Dyn$.
- Rule [EC-Expand]. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^{Dyn} : Dyn \Rightarrow \tau$ and $v^{Dyn} : Dyn \Rightarrow \tau \longrightarrow_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$, then by rule [P-Cast], we have that $N^{\rho_1} \sqsubseteq v^{Dyn}$ and $\rho_1 \sqsubseteq Dyn$ and $\rho_2 \sqsubseteq \tau$. By lemma 6.6, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} v'^{\rho_1}$ and $v'^{\rho_1} \sqsubseteq v^{Dyn}$. By rule [E-Ctx] and context E : $\rho_1 \Rightarrow \rho_2$, $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} v'^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [P-CastR], we have that $v'^{\rho_1} \sqsubseteq v^{Dyn} : Dyn \Rightarrow G$. Since $\rho_1 \sqsubseteq Dyn$ then $\rho_1 \sqsubseteq G$, and by rule [P-Cast], we have that $v'^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$.

- Rule [P-CastL]. If $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \sqsubseteq M^\tau$ and $M^\tau \longrightarrow_{\wedge CC} M'^\tau$ then by rule [P-CastL], we have that $N^{\rho_1} \sqsubseteq M^\tau$, $\rho_1 \sqsubseteq \tau$ and $\rho_2 \sqsubseteq \tau$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} N'^{\rho_1}$ and $N'^{\rho_1} \sqsubseteq M'^\tau$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow_{\wedge CC} N'^{\rho_1} : \rho_1 \Rightarrow \rho_2$, and by rule [P-CastL], we have that N'^{ρ_1} : $\rho_1 \Rightarrow \rho_2 \sqsubseteq M'^\tau$.
- Rule [P-CastR]. There are seven possibilities:
  - Rule [E-Ctx] and context $E : \tau_1 \Rightarrow \tau_2$. If $N^\rho \sqsubseteq M^{\tau_1}$: $\tau_1 \Rightarrow \tau_2$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} M'^{\tau_1} : \tau_1 \Rightarrow \tau_2$ then by rule [P-CastR], we have that $N^\rho \sqsubseteq M^{\tau_1}$ and $\rho \sqsubseteq \tau_1$ and $\rho \sqsubseteq \tau_2$, and by rule [E-Ctx], we have that $M^{\tau_1} \longrightarrow_{\wedge CC} M'^{\tau_1}$. By the induction hypothesis, we have that $N^\rho \longrightarrow^*_{\wedge CC} N'^\rho$ and $N'^\rho \sqsubseteq M'^{\tau_1}$. By rule [P-CastR], we have that $N'^\rho \sqsubseteq M'^{\tau_1} : \tau_1 \Rightarrow \tau_2$.
  - Rule [E-Wrong] and context $E : \tau_1 \Rightarrow \tau_2$. If $N^\rho \sqsubseteq M^{\tau_1}$: $\tau_1 \Rightarrow \tau_2$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} wrong^{\tau_2}$ then by rule [P-CastR], we have that $N^\rho \sqsubseteq M^{\tau_1}$ and $\rho \sqsubseteq \tau_1$ and $\rho \sqsubseteq \tau_2$. By theorems 6.3 and 6.4, $N^\rho \longrightarrow^*_{\wedge CC} N'^\rho$, and by rule [P-Wrong], $N'^\rho \sqsubseteq wrong^{\tau_2}$.
  - Rule [EC-Identity]. If $N^\rho \sqsubseteq v^\tau : \tau \Rightarrow \tau$ and $v^\tau : \tau \Rightarrow \tau \longrightarrow_{\wedge CC} v^\tau$ then by rule [P-CastR], we have that $N^\rho \sqsubseteq v^\tau$ and $\rho \sqsubseteq \tau$ and $\rho \sqsubseteq \tau$. By lemma 6.6, we have that $N^\rho \longrightarrow^*_{\wedge CC} v'^\rho$ and $v'^\rho \sqsubseteq v^\tau$.
  - Rule [EC-Succeed]. If $N^\rho \sqsubseteq v^G : G \Rightarrow Dyn : Dyn \Rightarrow G$ and $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \longrightarrow_{\wedge CC} v^G$ then by rule [P-CastR], $N^\rho \sqsubseteq v^G : G \Rightarrow Dyn$ and $\rho \sqsubseteq Dyn$ and $\rho \sqsubseteq G$. By rule [P-CastR], $N^\rho \sqsubseteq v^G$ and $\rho \sqsubseteq G$ and $\rho \sqsubseteq Dyn$. By lemma 6.6, we have that $N^\rho \longrightarrow^*_{\wedge CC} v'^\rho$ and $v'^\rho \sqsubseteq v^G$.
  - Rule [EC-Fail]. If $N^\rho \sqsubseteq v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2$ and $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \longrightarrow_{\wedge CC} wrong^{G_2}$ then by rule [P-CastR], $N^\rho \sqsubseteq v^{G_1} : G_1 \Rightarrow Dyn$ and $\rho \sqsubseteq Dyn$ and $\rho \sqsubseteq G_2$. By theorems 6.3 and 6.4, $N^\rho \longrightarrow^*_{\wedge CC} N'^\rho$, and by rule [P-Wrong], $N'^\rho \sqsubseteq wrong^{G_2}$.
  - Rule [EC-Ground]. If $N^\rho \sqsubseteq v^\tau : \tau \Rightarrow Dyn$ and $v^\tau : \tau \Rightarrow Dyn \longrightarrow_{\wedge CC} v^\tau : \tau \Rightarrow G : G \Rightarrow Dyn$, then by rule [P-CastR], we have that $N^\rho \sqsubseteq v^\tau$ and $\rho \sqsubseteq \tau$ and $\rho \sqsubseteq Dyn$. By lemma 6.6, we have that $N^\rho \longrightarrow^*_{\wedge CC} v'^\rho$ and $v'^\rho \sqsubseteq v^\tau$. By rule [P-CastR], we have that $v'^\rho \sqsubseteq v^\tau : \tau \Rightarrow G$, and by rule [P-CastR], we have that $v'^\rho \sqsubseteq v^\tau : \tau \Rightarrow G : G \Rightarrow Dyn$.
  - Rule [EC-Expand]. If $N^\rho \sqsubseteq v^{Dyn} : Dyn \Rightarrow \tau$ and v^{Dyn} : $Dyn \Rightarrow \tau \longrightarrow_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$, then by rule [P-CastR], we have that $N^\rho \sqsubseteq v^{Dyn}$ and $\rho \sqsubseteq Dyn$ and $\rho \sqsubseteq \tau$. By lemma 6.6, we have that $N^\rho \longrightarrow^*_{\wedge CC} v'^\rho$ and $v'^\rho \sqsubseteq v^{Dyn}$. By rule [P-CastR], we have that $v'^\rho \sqsubseteq v^{Dyn} : Dyn \Rightarrow G$, and by rule [P-CastR], we have that $v'^\rho \sqsubseteq v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$.

□

Theorem 6.10 (Gradual Guarantee). *For all $\Upsilon^\upsilon \sqsubseteq \Pi^\sigma$ such that $\emptyset \vdash_{\wedge CC} \Pi^\sigma : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon^\upsilon : \upsilon$, and assuming $\pi_1^\sigma \neq wrong^\sigma$ and $\pi_2^\upsilon \neq wrong^\upsilon$:*

(1) *if $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$ then $\Upsilon^\upsilon \longrightarrow^*_{\wedge CC} \pi_2^\upsilon$ and $\pi_2^\upsilon \sqsubseteq \pi_1^\sigma$.*
if Π^σ diverges then Υ^υ diverges.
(2) *if $\Upsilon^\upsilon \longrightarrow^*_{\wedge CC} \pi_2^\upsilon$ then either $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$ and $\pi_2^\upsilon \sqsubseteq \pi_1^\sigma$, or $\Pi^\sigma \longrightarrow^*_{\wedge CC} wrong^\sigma$.*
if Υ^υ diverges then Π^σ diverges or $\Pi^\sigma \longrightarrow^*_{\wedge CC} wrong^\sigma$.

Proof. Proof for part 1. By lemma 6.9 and induction on the length of the reduction sequence, applying theorem 6.4, we have that $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$, $\Upsilon^\upsilon \longrightarrow^*_{\wedge CC} \Upsilon'^\upsilon$ and $\Upsilon'^\upsilon \sqsubseteq \pi_1^\sigma$. By lemma 6.6 applied to each component, and by rule [E-Par], then $\Upsilon'^\upsilon \longrightarrow^*_{\wedge CC} \pi_2^\upsilon$ and $\pi_2^\upsilon \sqsubseteq \pi_1^\sigma$.

If Π^σ diverges, then we have an infinite reduction chain $\Pi^\sigma \longrightarrow_{\wedge CC} \Pi'^\sigma \longrightarrow_{\wedge CC} \cdots$. By lemma 6.9, we also have an infinite reduction chain $\Upsilon^\upsilon \longrightarrow_{\wedge CC} \Upsilon'^\upsilon \longrightarrow_{\wedge CC} \cdots$. Therefore, Υ^υ diverges.

Proof for part 2. If $\Upsilon^\upsilon \longrightarrow_{\wedge CC} \pi_2^\upsilon$, then, because Π^σ is well-typed, by theorem 6.4, either $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$, $\Pi^\sigma \longrightarrow^*_{\wedge CC} wrong^\sigma$ or Π^σ diverges. If $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$, then by part 1, we have that $\pi_2^\upsilon \sqsubseteq \pi_1^\sigma$. If Π^σ diverges, then by part 2, Υ^υ also diverges, which is a contradiction.

If Υ^υ diverges, let's assume $\Pi^\sigma \longrightarrow^*_{\wedge CC} \pi_1^\sigma$. Then, by part 1, we have that $\Upsilon^\upsilon \longrightarrow^*_{\wedge CC} \pi_2^\upsilon$, which is a contradiction. Therefore, Π^σ diverges or $\Pi^\sigma \longrightarrow^*_{\wedge CC} wrong^\sigma$. □

Lemma 6.11 (Extra Cast on the Right (Confluency)). *If $\emptyset \vdash_{\wedge CC} v_1^{\tau_1} : \tau_1$, $\emptyset \vdash_{\wedge CC} r_2^{\tau_2} : \tau_2$, $v_1^{\tau_1} \bowtie r_2^{\tau_2}$ then $r_2^{\tau_2} : \tau_2 \Rightarrow \tau_3 \longrightarrow^*_{\wedge CC} r_3^{\tau_3}$ and $v_1^{\tau_1} \bowtie r_3^{\tau_3}$.*

Proof. We divide this proof into 2 parts: either $r_2^{\tau_2} = wrong^{\tau_2}$; or $r_2^{\tau_2}$ is a value $v_2^{\tau_2}$, in which case we proceed by case analysis on τ_2 and τ_3.

Proof for $r_2^{\tau_2} = wrong^{\tau_2}$. If $v_1^{\tau_1} \bowtie wrong^{\tau_2}$ then by rule [E-Wrong], $wrong^{\tau_2} : \tau_2 \Rightarrow \tau_3 \longrightarrow_{\wedge CC} wrong^{\tau_3}$ and by rule [V-WrongR], $v_1^{\tau_1} \bowtie wrong^{\tau_3}$.

Proof for $r_2^{\tau_2} = v_2^{\tau_2}$:

- Both τ_2 and τ_3 are the same. If $v_1^{\tau_1} \bowtie v_2^{\tau_2}$ then by rule [EC-Identity], $v_2^{\tau_2} : \tau_2 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} v_2^{\tau_2}$ and $v_1^{\tau_1} \bowtie v_2^{\tau_2}$.
- τ_2 is a base type B and $\tau_3 = Dyn$. If $v_1^{\tau_1} \bowtie v_2^B$ then $v_2^B : B \Rightarrow Dyn$ is a value, so $v_2^B : B \Rightarrow Dyn \longrightarrow^0_{\wedge CC} v_2^B : B \Rightarrow Dyn$ and by rule [V-CastR], $v_1^{\tau_1} \bowtie v_2^B : B \Rightarrow Dyn$.
- $\tau_2 = Dyn$ and τ_3 is a base type B. If $v_1^{\tau_1} \bowtie v_2^{Dyn}$ then there are two possibilities:
  - $v_2^{Dyn} : Dyn \Rightarrow B \longrightarrow^*_{\wedge CC} v_2'^B$, so we have that $v_2^{Dyn} = v_2'^B : B \Rightarrow Dyn$ and by rule [V-CastR], we have that $v_1^{\tau_1} \bowtie v_2'^{\tau_2}$. By rule [EC-Succeed], we have that $v_2'^B : B \Rightarrow Dyn : Dyn \Rightarrow B \longrightarrow_{\wedge CC} v_2'^B$.
  - $v_2^{Dyn} : Dyn \Rightarrow B \longrightarrow^*_{\wedge CC} wrong^B$, so by rule [V-WrongR], $v_1^{\tau_1} \bowtie wrong^B$.
- $\tau_2 = \tau_2' \rightarrow \tau_2''$ and $\tau_3 = Dyn$. If $v_1^{\tau_1} \bowtie v_2^{\tau_2' \rightarrow \tau_2''}$ then there are two possibilities:
  - $\tau_2' \rightarrow \tau_2'' = G$. Then $v_2^G : G \Rightarrow Dyn$ is a value and therefore $v_2^G : G \Rightarrow Dyn \longrightarrow^0_{\wedge CC} v_2^G : G \Rightarrow Dyn$ and by rule [V-CastR], $v_1^{\tau_1} \bowtie v_2^G : G \Rightarrow Dyn$.
  - $\tau_2' \rightarrow \tau_2'' \neq G$. Then by rule [EC-Ground], $v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow Dyn \longrightarrow_{\wedge CC} v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow G : G \Rightarrow Dyn$. By rule [V-CastR], we have that $v_1^{\tau_1} \bowtie v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow G$. By rule [V-CastR], we have that $v_1^{\tau_1} \bowtie v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow G : G \Rightarrow Dyn$.
- $\tau_2 = Dyn$ and $\tau_3 = \tau_3' \rightarrow \tau_3''$. If $v_1^{\tau_1} \bowtie v_2^{Dyn}$ then there are two possibilities:
  - $\tau_3' \rightarrow \tau_3'' = G$. There are two possibilities:
    * $v_2^{Dyn} : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow^*_{\wedge CC} v_2'^{\tau_3' \rightarrow \tau_3''}$, so we have that $v_2^{Dyn} = v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn$. By rule [V-CastR], $v_1^{\tau_1} \bowtie v_2'^{\tau_3' \rightarrow \tau_3''}$. By rule [EC-Succeed], we have that $v_2'^{\tau_3' \rightarrow \tau_3''} : \tau_3' \rightarrow \tau_3'' \Rightarrow Dyn : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow_{\wedge CC} v_2'^{\tau_3' \rightarrow \tau_3''}$.
    * $v_2^{Dyn} : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow^*_{\wedge CC} wrong^{\tau_3' \rightarrow \tau_3''}$, by rule [V-WrongR], we have that $v_1^{\tau_1} \bowtie wrong^{\tau_3' \rightarrow \tau_3''}$.
  - $\tau_3' \rightarrow \tau_3'' \neq G$. Then by rule [EC-Expand], $v_2^{Dyn} : Dyn \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow_{\wedge CC} v_2^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau_3' \rightarrow \tau_3''$. By rule [V-CastR], we have that $v_1^{\tau_1} \bowtie v_2^{Dyn} : Dyn \Rightarrow G$. By rule [V-CastR], we have that $v_1^{\tau_1} \bowtie v_2^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau_3' \rightarrow \tau_3''$.
- $\tau_2 = \tau_2' \rightarrow \tau_2''$ and $\tau_3 = \tau_3' \rightarrow \tau_3''$. If $v_1^{\tau_1} \bowtie v_2^{\tau_2' \rightarrow \tau_2''}$ then $v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow \tau_3' \rightarrow \tau_3''$ is a value, and therefore $v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow \tau_3' \rightarrow \tau_3'' \longrightarrow^0_{\wedge CC} v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow \tau_3' \rightarrow \tau_3''$. By rule [V-CastR], we have that $v_1^{\tau_1} \bowtie v_2^{\tau_2' \rightarrow \tau_2''} : \tau_2' \rightarrow \tau_2'' \Rightarrow \tau_3' \rightarrow \tau_3''$.

□

Lemma 6.12 (Catchup to Value on the Left (Confluency)). *If $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} N^\rho : \rho$ and $v^\tau \bowtie N^\rho$ then $N^\rho \longrightarrow^*_{\wedge CC} r^\rho$ and $v^\tau \bowtie r^\rho$.*

Proof. We proceed by induction on the length of the derivation tree of $v^\tau \bowtie N^\rho$.

Base cases:

- Rule [V-Con]. If $\emptyset \vdash_{\wedge CC} k^B : B$ and $\emptyset \vdash_{\wedge CC} k^B : B$ and $k^B \bowtie k^B$ then, since k^B is a value, $k^B \longrightarrow^0_{\wedge CC} k^B$ and $k^B \bowtie k^B$.
- Rule [V-Abs]. If $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \rightarrow \tau$ and $\emptyset \vdash_{\wedge CC} \lambda x : v \,.\, N^\rho : v \rightarrow \rho$ and $\lambda x : \sigma \,.\, M^\tau \bowtie \lambda x : v \,.\, N^\rho$ then, since $\lambda x : v \,.\, N^\rho$ is a value, $\lambda x : v \,.\, N^\rho \longrightarrow^0_{\wedge CC} \lambda x : v \,.\, N^\rho$ and $\lambda x : \sigma \,.\, M^\tau \bowtie \lambda x : v \,.\, N^\rho$.
- Rule [V-WrongR]. If $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} wrong^\rho : \rho$ and $v^\tau \bowtie wrong^\rho$, then since $wrong^\rho$ is already a result, $wrong^\rho \longrightarrow^0_{\wedge CC} wrong^\rho$ and $v^\tau \bowtie wrong^\rho$.

Induction step:

- Rule [V-Cast]. If $\emptyset \vdash_{\wedge CC} v^{\tau_1} : \tau_1 \Rightarrow \tau_2 : \tau_2$ and $\vdash_{\wedge CC} N^{\rho_1} : \rho_1 \Rightarrow \rho_2 : \rho_2$ and $v^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ then by rule [V-Cast], we have that $v^{\tau_1} \bowtie N^{\rho_1}$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} r^{\rho_1}$ and $v^{\tau_1} \bowtie r^{\rho_1}$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [V-CastL], we have that $v^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie r^{\rho_1}$. By lemma 6.11, $r^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r'^{\rho_2}$ and $v^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie r'^{\rho_2}$.
- Rule [V-CastL]. If $\emptyset \vdash_{\wedge CC} v^{\tau_1} : \tau_1 \Rightarrow \tau_2 : \tau_2$ and $\emptyset \vdash_{\wedge CC} N^\rho : \rho$ and $v^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^\rho$ then by rule [V-CastL], we have that $v^{\tau_1} \bowtie N^\rho$. By the induction hypothesis, we have that $N^\rho \longrightarrow^*_{\wedge CC} r^\rho$ and $v^{\tau_1} \bowtie r^\rho$. By rule [V-CastL], we have that $v^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie r^\rho$.
- Rule [V-CastR]. If $\emptyset \vdash_{\wedge CC} v^\tau : \tau$ and $\emptyset \vdash_{\wedge CC} N^{\rho_1} : \rho_1 \Rightarrow \rho_2 : \rho_2$ and $v^\tau \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ then by rule [V-CastR], we have that $v^\tau \bowtie N^{\rho_1}$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} r^{\rho_1}$ and $v^\tau \bowtie r^{\rho_1}$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By lemma 6.11, we have that $r^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r'^{\rho_2}$ and $v^\tau \bowtie r'^{\rho_2}$.

□

Lemma 6.13 (Simulation of Function Application (Confluency)). *Assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \rightarrow \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} v'^{v \rightarrow \rho} : v \rightarrow \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^v : v$. If $\lambda x : \sigma \,.\, M^\tau \bowtie v'^{v \rightarrow \rho}$ and $\pi^\sigma \bowtie \pi'^v$ then $v'^{v \rightarrow \rho}\ \pi'^v \longrightarrow^*_{\wedge CC} M'^\rho$ and $[c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\ M^\tau \bowtie M'^\rho$.*

Proof. We proceed by induction on the length of the derivation tree of $\lambda x : \sigma \,.\, M^\tau \bowtie v'^{v \rightarrow \rho}$. [3]

Base cases:

- Rule [V-Abs]. We assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma \,.\, M^\tau : \sigma \rightarrow \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} \lambda x : v \,.\, N^\rho : v \rightarrow \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^v : v$. If $\lambda x : \sigma \,.\, M^\tau \bowtie \lambda x : v \,.\, N^\rho$ and $\pi^\sigma \bowtie \pi'^v$, then by rule [E-Beta], we have that $(\lambda x : v \,.\, N^\rho)\ \pi'^v \longrightarrow_{\wedge CC} [c_i^{\rho'}(x) \mapsto \langle \pi'^v \rangle_i^{\rho'}]\ N^\rho$, and $[c_i^{\tau'}(x) \mapsto \langle \pi^\sigma \rangle_i^{\tau'}]\ M^\tau \bowtie [c_i^{\rho'}(x) \mapsto \langle \pi'^v \rangle_i^{\rho'}]\ N^\rho$.

Induction step:

[3] This lemma is used in Lemma 6.15, in rule [V-App], case rule [E-Beta]. According to rule [E-Beta], π^σ is not *wrong*. In the specific case we use the lemma, we assume π'^v is not *wrong*.

- Rule [V-CastR]. We assume $\emptyset \vdash_{\wedge CC} \lambda x : \sigma\,.\, M^\tau : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^\sigma : \sigma$, $\emptyset \vdash_{\wedge CC} v'^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^{v} : v$. If $\lambda x : \sigma\,.\, M^\tau \bowtie v'^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho$ and $\pi^\sigma \bowtie \pi'^v$, then by rule [V-CastR], we have that $\lambda x : \sigma\,.\, M^\tau \bowtie v'^{v'\to\rho'}$. By rule [EC-Application], we have that $(v'^{v'\to\rho'} : v' \to \rho' \Rightarrow v \to \rho)\ \pi'^v \longrightarrow_{\wedge CC} (v'^{v'\to\rho'}\ (\pi'^v : v \Rightarrow_\wedge v')) : \rho' \Rightarrow \rho$. By rule [V-Par] and rule [V-CastR], we have that $\pi^\sigma \bowtie \pi'^v : v \Rightarrow_\wedge v'$. By the induction hypothesis, we have that $(v'^{v'\to\rho'}\ (\pi'^v : v \Rightarrow_\wedge v')) \longrightarrow^*_{\wedge CC} N^{\rho'}$ and $[c_i^{\tau'}(x) \mapsto \langle\pi^\sigma\rangle_i^{\tau'}]\, M^\tau \bowtie N^{\rho'}$. By rule [E-Ctx] and context $E : \rho' \Rightarrow \rho$, we have that $(v'^{v'\to\rho'}\ (\pi'^v : v \Rightarrow_\wedge v')) : \rho' \Rightarrow \rho \longrightarrow^*_{\wedge CC} N^{\rho'} : \rho' \Rightarrow \rho$. By rule [V-CastR], we have that $[c_i^{\tau'}(x) \mapsto \langle\pi^\sigma\rangle_i^{\tau'}]\, M^\tau \bowtie N^{\rho'} : \rho' \Rightarrow \rho$.

□

Lemma 6.14 (Simulation of Unwrapping (Confluency)). *Assume $\emptyset \vdash_{\wedge CC} v^{\sigma\to\tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{v\to\rho} : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^{v} : v$. If $v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v\to\rho}$ and $\pi^{\sigma'} \bowtie \pi'^v$ then $v'^{v\to\rho}\ \pi'^v \longrightarrow^*_{\wedge CC} M^\rho$ and $v^{\sigma\to\tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma) : \tau \Rightarrow \tau' \bowtie M^\rho$.*

Proof. We proceed by induction on the length of the derivation tree of $v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v\to\rho}$. [4]

Base cases:

- Rule [V-Cast]. We assume $\emptyset \vdash_{\wedge CC} v^{\sigma\to\tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{v\to\rho} : v \to \rho \Rightarrow v' \to \rho' : v' \to \rho'$ and $\emptyset \vdash_{\wedge CC} \pi'^{v'} : v'$. If $v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v\to\rho} : v \to \rho \Rightarrow v' \to \rho'$ and $\pi^{\sigma'} \bowtie \pi'^{v'}$ then by rule [V-Cast], we have that $v^{\sigma\to\tau} \bowtie v'^{v\to\rho}$. By rule [EC-Application], we have that $(v'^{v\to\rho} : v \to \rho \Rightarrow v' \to \rho')\ \pi'^{v'} \longrightarrow_{\wedge CC} (v'^{v\to\rho}\ (\pi'^{v'} : v' \Rightarrow_\wedge v)) : \rho \Rightarrow \rho'$. By rules [V-Par] and [V-Cast] we have that $\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma \bowtie \pi'^{v'} : v' \Rightarrow_\wedge v$. By rule [V-App], we have that $v^{\sigma\to\tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma) \bowtie v'^{v\to\rho}\ (\pi'^{v'} : v' \Rightarrow_\wedge v)$. By rule [V-Cast], we have that $(v^{\sigma\to\tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma)) : \tau \Rightarrow \tau' \bowtie (v'^{v\to\rho}\ (\pi'^{v'} : v' \Rightarrow_\wedge v)) : \rho \Rightarrow \rho'$.
- Rule [V-CastL]. We assume $\emptyset \vdash_{\wedge CC} v^{\sigma\to\tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{v\to\rho} : v \to \rho$ and $\emptyset \vdash_{\wedge CC} \pi'^{v} : v$. If $v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v\to\rho}$ and $\pi^{\sigma'} \bowtie \pi'^v$ then by rule [V-CastL], we have that $v^{\sigma\to\tau} \bowtie v'^{v\to\rho}$. Since $v'^{v\to\rho}$ and π'^v are values, we have that $v'^{v\to\rho}\ \pi'^v \longrightarrow^0_{\wedge CC} v'^{v\to\rho}\ \pi'^v$. By rule [V-CastL], we have that $\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma \bowtie \pi'^v$. By rule [V-App], we have that $v^{\sigma\to\tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma) \bowtie v'^{v\to\rho}\ \pi'^v$. By rule [V-CastL], we have that $(v^{\sigma\to\tau}\ (\pi^{\sigma'} : \sigma' \Rightarrow_\wedge \sigma)) : \tau \Rightarrow \tau' \bowtie v'^{v\to\rho}\ \pi'^v$.

Induction step:

- Rule [V-CastR]. We assume $\emptyset \vdash_{\wedge CC} v^{\sigma\to\tau} : \sigma \to \tau$ and $\emptyset \vdash_{\wedge CC} \pi^{\sigma'} : \sigma'$, $\emptyset \vdash_{\wedge CC} v'^{v\to\rho} : v \to \rho \Rightarrow v' \to \rho' : v' \to \rho'$ and $\emptyset \vdash_{\wedge CC} \pi'^{v'} : v'$. If $v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v\to\rho} : v \to \rho \Rightarrow v' \to \rho'$ and $\pi^{\sigma'} \bowtie \pi'^{v'}$ then by rule [V-CastR], we have that $v^{\sigma\to\tau} : \sigma \to \tau \Rightarrow \sigma' \to \tau' \bowtie v'^{v\to\rho}$, and by rule [V-CastR], we have that $\pi^{\sigma'} \bowtie \pi'^{v'} : v' \Rightarrow_\wedge v$. By rule [EC-Application], we have that $(v'^{v\to\rho} : v \to \rho \Rightarrow v' \to \rho')\ \pi'^{v'} \longrightarrow_{\wedge CC} (v'^{v\to\rho}\ (\pi'^{v'} : v' \Rightarrow_\wedge v)) : \rho \Rightarrow \rho'$. By the induction hypothesis, we have that $v'^{v\to\rho}\ (\pi'^{v'} : v' \Rightarrow_\wedge v) \longrightarrow^*_{\wedge CC} M^\rho$ and $v^{\sigma\to\tau}\ (\pi^\sigma : \sigma' \Rightarrow_\wedge \sigma) : \tau \Rightarrow \tau' \bowtie M^\rho$. By rule [E-Ctx] and context $E : \rho \Rightarrow \rho'$, we have that $(v'^{v\to\rho}\ (\pi'^{v'} : v' \Rightarrow_\wedge v)) : \rho \Rightarrow \rho' \longrightarrow^*_{\wedge CC} M^\rho : \rho \Rightarrow \rho'$. By rule [V-CastR], we have that $v^{\sigma\to\tau}\ (\pi^\sigma : \sigma' \Rightarrow_\wedge \sigma) : \tau \Rightarrow \tau' \bowtie M^\rho : \rho \Rightarrow \rho'$.

□

Lemma 6.15 (Simulation of Variant Programs). *For all $\Pi_1^\sigma \bowtie \Upsilon_1^v$ such that $\emptyset \vdash_{\wedge CC} \Pi_1^\sigma : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon_1^v : v$, if $\Pi_1^\sigma \longrightarrow_{\wedge CC} \Pi_2^\sigma$ then there exists a Υ_2^v such that $\Upsilon_1^v \longrightarrow^*_{\wedge CC} \Upsilon_2^v$ and $\Pi_2^\sigma \bowtie \Upsilon_2^v$.*

Proof. We proceed by induction on the length of the derivation tree of $\Pi_1^\sigma \bowtie \Upsilon_1^v$ (definition 5.2) followed by case analysis on $\Pi_1^\sigma \longrightarrow_{\wedge CC} \Pi_2^\sigma$, and using lemmas 6.11, 6.12, 6.13 and 6.14, and theorems 6.3 and 6.4.

Base cases:

- Rule [V-Con]. If $k^B \bowtie k^B$ and since k^B is a value, then it is proved.
- Rule [V-WrongL]. If $wrong^\sigma \bowtie \Pi^v$ and $wrong^\sigma \longrightarrow_{\wedge CC} wrong^\sigma$, then by theorem 6.4, any amount of evaluation steps, say $\Pi^v \longrightarrow^*_{\wedge CC} \Upsilon^v$, yields an expression Υ^v. By rule [V-WrongL], we have that $wrong^\sigma \bowtie \Upsilon^v$.
- Rule [V-WrongR]. If $\Pi^\sigma \bowtie wrong^v$ and $\Pi^\sigma \longrightarrow_{\wedge CC} \Upsilon^\sigma$, then we have that $wrong^v \longrightarrow^0_{\wedge CC} wrong^v$ and by rule [V-WrongR], we have that $\Upsilon^\sigma \bowtie wrong^\sigma$.

Induction Step

- Rule [V-Abs]. If $\lambda x : \sigma\,.\, M^\tau \bowtie \lambda x : v\,.\, N^\rho$, and since both $\lambda x : \sigma\,.\, M^\tau$ and $\lambda x : v\,.\, N^\rho$ are values, then it is proved.
- Rule [V-App]. There are six possibilities:
  - Rule [E-Beta]. If $(\lambda x : \sigma\,.\, M^\tau)\ \pi^\sigma \bowtie N^\rho\ \Upsilon^v$ and $(\lambda x : \sigma\,.\, M^\tau)\ \pi^\sigma \longrightarrow_{\wedge CC} [c_i^{\tau'}(x) \mapsto \langle\pi^\sigma\rangle_i^{\tau'}]\, M^\tau$, then by rule [V-App], we have that $\lambda x : \sigma\,.\, M^\tau \bowtie N^\rho$ and $\pi^\sigma \bowtie \Upsilon^v$. By lemma 6.12, we have that $N^\rho \longrightarrow^*_{\wedge CC} r^\rho$ and $\lambda x : \sigma\,.\, M^\tau \bowtie r^\rho$. By applying lemma 6.12 to each derivation of rule [E-Par], we have that $\Upsilon^v \longrightarrow^*_{\wedge CC} \Upsilon'^v$ and $\pi^\sigma \bowtie \Upsilon'^v$, such that components in Υ'^v are all results. By applying rule [E-Ctx] with context $E\ \Upsilon^v$, we have that $N^\rho\ \Upsilon^v \longrightarrow^*_{\wedge CC} r^\rho\ \Upsilon^v$.

    If $r^\rho = wrong^\rho$, then by rule [E-Wrong], we have that $r^\rho\ \Upsilon^v \longrightarrow_{\wedge CC} wrong^{\rho'}$, and by rule [V-WrongR], $[c_i^{\tau'}(x) \mapsto \langle\pi^\sigma\rangle_i^{\tau'}]\, M^\tau \bowtie wrong^{\rho'}$.

    If $r^\rho \neq wrong^\rho$, then by rule [E-Ctx] with context $v^\rho\ E$, we have that $v^\rho\ \Upsilon^v \longrightarrow_{\wedge CC} v^\rho\ \Upsilon'^v$. If there exists a component of Υ'^v that is *wrong*, then by rule [E-Push], $\Upsilon'^v \longrightarrow_{\wedge CC} wrong^v$. By rule [E-Ctx], we have that $v^\rho\ \Upsilon'^v \longrightarrow_{\wedge CC} v^\rho\ wrong^v$ and by rule [E-Wrong], $v^\rho\ wrong^v \longrightarrow_{\wedge CC} wrong^{\rho'}$, and by rule [V-WrongR], $[c_i^{\tau'}(x) \mapsto \langle\pi^\sigma\rangle_i^{\tau'}]\, M^\tau \bowtie wrong^{\rho'}$.

[4] This lemma is used in Lemma 6.15, in rule [V-App], case rule [EC-Application]. According to rule [EC-Application], π^σ is not *wrong*. In the specific case we use the lemma, we assume π'^v is not *wrong*.

If $\Upsilon'^{\upsilon} = \pi'^{\upsilon}$, then by lemma 6.13, we have that $v^{\rho}\ \pi'^{\upsilon} \longrightarrow^*_{\wedge CC} N'^{\rho'}$ and $[c_i^{\tau'}(x) \mapsto \langle \pi^{\sigma} \rangle_i^{\tau'}]\ M^{\tau} \bowtie N'^{\rho'}$.

- Rule [E-Ctx] and context $E\ \Pi^{\sigma}$. If $M^{\tau}\ \Pi^{\sigma} \bowtie N^{\rho}\ \Upsilon^{\upsilon}$ and $M^{\tau}\ \Pi^{\sigma} \longrightarrow_{\wedge CC} M'^{\tau}\ \Pi^{\sigma}$, then by rule [V-App], we have that $M^{\tau} \bowtie N^{\rho}$ and $\Pi^{\sigma} \bowtie \Upsilon^{\upsilon}$, and by rule [E-Ctx], we have that $M^{\tau} \longrightarrow_{\wedge CC} M'^{\tau}$. By the induction hypothesis there exists a N'^{ρ} such that $N^{\rho} \longrightarrow^*_{\wedge CC} N'^{\rho}$ and $M'^{\tau} \bowtie N'^{\rho}$. By rule [E-Ctx], we have that $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} N'^{\rho}\ \Upsilon^{\upsilon}$, and by rule [V-App], we have that $M'^{\tau}\ \Pi^{\sigma} \bowtie N'^{\rho}\ \Upsilon^{\upsilon}$.
- Rule [E-Ctx] and context $v^{\tau}\ E$. If $M^{\tau}\ \Pi^{\sigma} \bowtie N^{\rho}\ \Upsilon^{\upsilon}$ and $M^{\tau}\ \Pi^{\sigma} \longrightarrow_{\wedge CC} M^{\tau}\ \Pi'^{\sigma}$, then by rule [V-App], we have that $M^{\tau} \bowtie N^{\rho}$ and $\Pi^{\sigma} \bowtie \Upsilon^{\upsilon}$, and by rule [E-Ctx], we have that $\Pi^{\sigma} \longrightarrow_{\wedge CC} \Pi'^{\sigma}$. By the induction hypothesis there exists a Υ'^{υ} such that $\Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} \Upsilon'^{\upsilon}$ and $\Pi'^{\sigma} \bowtie \Upsilon'^{\upsilon}$. By rule [E-Ctx], we have that $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} N^{\rho}\ \Upsilon'^{\upsilon}$, and by rule [V-App], we have that $M^{\tau}\ \Pi'^{\sigma} \bowtie N^{\rho}\ \Upsilon'^{\upsilon}$.
- Rule [E-Wrong] and context $E\ \Upsilon^{\upsilon}$ or $v^{\rho}\ E$. If $M^{\tau}\ \Pi^{\sigma} \bowtie N^{\rho}\ \Upsilon^{\upsilon}$ and $M^{\tau}\ \Pi^{\sigma} \longrightarrow_{\wedge CC} wrong^{\tau'}$, for $\tau = \sigma \to \tau'$ and $\rho = \upsilon \to \rho'$, then by theorems 6.3 and 6.4, $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} N'^{\rho'}$, and by rule [V-WrongL], $wrong^{\tau'} \bowtie N'^{\rho'}$.
- Rule [EC-Application]. If $(v^{\sigma' \to \tau'} : \sigma' \to \tau' \Rightarrow \sigma \to \tau)\ \pi^{\sigma} \bowtie N^{\rho}\ \Upsilon^{\upsilon}$ and $(v^{\sigma' \to \tau'} : \sigma' \to \tau' \Rightarrow \sigma \to \tau)\ \pi^{\sigma} \longrightarrow_{\wedge CC} (v^{\sigma' \to \tau'}\ (\pi^{\sigma} : \sigma \Rightarrow_{\wedge} \sigma')) : \tau' \Rightarrow \tau$, then by rule [V-App], we have that $(v^{\sigma' \to \tau'} : \sigma' \to \tau' \Rightarrow \sigma \to \tau) \bowtie N^{\rho}$ and $\pi^{\sigma} \bowtie \Upsilon^{\upsilon}$. By lemma 6.12, we have that $N^{\rho} \longrightarrow^*_{\wedge CC} r^{\rho}$ and $(v^{\sigma' \to \tau'} : \sigma' \to \tau' \Rightarrow \sigma \to \tau) \bowtie r^{\rho}$. By applying lemma 6.12 to each derivation of rule [E-Par], we have that $\Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} \Upsilon'^{\upsilon}$ and $\pi^{\sigma} \bowtie \Upsilon'^{\upsilon}$, such that components in Υ'^{υ} are all results. By applying rule [E-Ctx] with context $E\ \Upsilon^{\upsilon}$, we have that $N^{\rho}\ \Upsilon^{\upsilon} \longrightarrow^*_{\wedge CC} r^{\rho}\ \Upsilon^{\upsilon}$.

  If $r^{\rho} = wrong^{\rho}$, then by rule [E-Wrong], we have that $r^{\rho}\ \Upsilon^{\upsilon} \longrightarrow_{\wedge CC} wrong^{\rho'}$, and by rule [V-WrongR], $(v^{\sigma' \to \tau'}\ (\pi^{\sigma} : \sigma \Rightarrow_{\wedge} \sigma')) : \tau' \Rightarrow \tau \bowtie wrong^{\rho'}$.

  If $r^{\rho} \neq wrong^{\rho}$, then by rule [E-Ctx] with context $v'^{\rho}\ E$, we have that $v'^{\rho}\ \Upsilon^{\upsilon} \longrightarrow_{\wedge CC} v'^{\rho}\ \Upsilon'^{\upsilon}$. If there exists a component of Υ'^{υ} that is *wrong*, then by rule [E-Push], $\Upsilon'^{\upsilon} \longrightarrow_{\wedge CC} wrong^{\upsilon}$. By rule [E-Ctx], we have that $v'^{\rho}\ \Upsilon'^{\upsilon} \longrightarrow_{\wedge CC} v'^{\rho}\ wrong^{\upsilon}$ and by rule [E-Wrong], $v'^{\rho}\ wrong^{\upsilon} \longrightarrow_{\wedge CC} wrong^{\rho'}$, and by rule [V-WrongR], $(v^{\sigma' \to \tau'}\ (\pi^{\sigma} : \sigma \Rightarrow_{\wedge} \sigma')) : \tau' \Rightarrow \tau \bowtie wrong^{\rho'}$.

  If $\Upsilon'^{\upsilon} = \pi'^{\upsilon}$, then by lemma 6.14, we have that $v'^{\rho}\ \pi'^{\upsilon} \longrightarrow^*_{\wedge CC} N'^{\rho'}$ and $(v^{\sigma' \to \tau'}\ (\pi^{\sigma} : \sigma \Rightarrow_{\wedge} \sigma')) : \tau' \Rightarrow \tau \bowtie N'^{\rho'}$.

- Rule [V-Add]. There are five possibilities:
  - Rule [E-Add]. If $k_1^{Int} + k_2^{Int} \bowtie M_1^{Int} + M_2^{Int}$ and $k_1^{Int} + k_2^{Int} \longrightarrow_{\wedge CC} k_3^{Int}$ then by rule [V-Add], we have that $k_1^{Int} \bowtie M_1^{Int}$ and $k_2^{Int} \bowtie M_2^{Int}$. By lemma 6.12, we have that $M_1^{Int} \longrightarrow^*_{\wedge CC} r_1^{Int}$ and $k_1^{Int} \bowtie r_1^{Int}$ and $M_2^{Int} \longrightarrow^*_{\wedge CC} r_2^{Int}$ and $k_2^{Int} \bowtie r_2^{Int}$.

    If either r_1^{Int} or r_2^{Int} is a *wrong*, then by rule [E-Wrong] and contexts $E + M_2^{Int}$ or $v^{Int} + E$, $M_1^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} wrong^{Int}$ and by rule [V-WrongR], $k_3^{Int} \bowtie wrong^{Int}$.

    Otherwise, we have that r_1^{Int} is a constant k_4^{Int} and r_2^{Int} is a constant k_5^{Int}. By rule [E-Ctx], and contexts $E + M^{\tau}$ and $v^{\tau} + E$, we have that $M_1^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} k_4^{Int} + M_2^{Int}$ and $k_4^{Int} + M_2^{Int} \longrightarrow^*_{\wedge CC} k_4^{Int} + k_5^{Int}$. By rule [E-Add], we have that $k_4^{Int} + k_5^{Int} \longrightarrow_{\wedge CC} k_3^{Int}$. By rule [V-Con], we have that $k_3^{Int} \bowtie k_3^{Int}$.
  - Rule [E-Ctx] and context $E + M^{\tau}$. If $M_1^{Int} + M_2^{Int} \bowtie N_1^{Int} + N_2^{Int}$ and $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} M_1'^{Int} + M_2^{Int}$, then by rule [V-Add], we have that $M_1^{\tau_1} \bowtie N_1^{\rho_1}$ and $M_2^{\tau_2} \bowtie N_2^{\rho_2}$, and by rule [E-Ctx], we have that $M_1^{Int} \longrightarrow_{\wedge CC} M_1'^{Int}$. By the induction hypothesis, we have that $N_1^{Int} \longrightarrow^*_{\wedge CC} N_1'^{Int}$ and $M_1'^{Int} \bowtie N_1'^{Int}$. By rule [E-Ctx], we have that $N_1^{Int} + N_2^{Int} \longrightarrow^*_{\wedge CC} N_1'^{Int} + N_2^{Int}$ and by rule [V-Add], we have that $M_1'^{Int} + M_2^{Int} \bowtie N_1'^{Int} + N_2^{Int}$.
  - Rule [E-Ctx] and context $v^{\tau} + E$. If $M_1^{Int} + M_2^{Int} \bowtie N_1^{Int} + N_2^{Int}$ and $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} M_1^{Int} + M_2'^{Int}$, then by rule [V-Add], we have that $M_1^{Int} \bowtie N_1^{Int}$ and $M_2^{Int} \bowtie N_2^{Int}$, and by rule [E-Ctx], we have that $M_2^{Int} \longrightarrow_{\wedge CC} M_2'^{Int}$. By the induction hypothesis, we have that $N_2^{Int} \longrightarrow^*_{\wedge CC} N_2'^{Int}$ and $M_2'^{Int} \bowtie N_2'^{Int}$. By rule [E-Ctx], we have that $N_1^{Int} + N_2^{Int} \longrightarrow^*_{\wedge CC} N_1^{Int} + N_2'^{Int}$ and by rule [V-Add], we have that $M_1^{Int} + M_2'^{Int} \bowtie N_1^{Int} + N_2'^{Int}$.
  - Rule [E-Wrong] and context $E + M^{\tau}$ or $v^{\tau} + E$. If $M_1^{Int} + M_2^{Int} \bowtie N_1^{Int} + N_2^{Int}$ and $M_1^{Int} + M_2^{Int} \longrightarrow_{\wedge CC} wrong^{Int}$, then by theorems 6.3 and 6.4, $N_1^{Int} + N_2^{Int} \longrightarrow^*_{\wedge CC} N^{Int}$, and by rule [V-WrongL], $wrong^{Int} \bowtie N^{Int}$.
- Rule [V-Par]. There are two possibilities:
  - Rule [E-Push]. If $r_1^{\tau_1} \mid \dots \mid r_n^{\tau_n} \bowtie M_1^{\rho_1} \mid \dots \mid M_n^{\rho_n}$ and $r_1^{\tau_1} \mid \dots \mid r_n^{\tau_n} \longrightarrow_{\wedge CC} wrong^{\tau_1 \wedge \dots \wedge \tau_n}$ then by theorems 6.3 and 6.4, we have that $M_1^{\rho_1} \longrightarrow^*_{\wedge CC} N_1^{\rho_1}$ and ... and $M_n^{\rho_n} \longrightarrow^*_{\wedge CC} N_n^{\rho_n}$. By rule [E-Par], we have that $M_1^{\rho_1} \mid \dots \mid M_n^{\rho_n} \longrightarrow^*_{\wedge CC} N_1^{\rho_1} \mid \dots \mid N_n^{\rho_n}$ and by rule [V-WrongL], we have that $wrong^{\tau_1 \wedge \dots \wedge \tau_n} \bowtie N_1^{\rho_1} \mid \dots \mid N_n^{\rho_n}$.
  - Rule [E-Par]. If $M_1^{\tau_1} \mid \dots \mid M_n^{\tau_n} \bowtie N_1^{\rho_1} \mid \dots \mid N_n^{\rho_n}$ and $M_1^{\tau_1} \mid \dots \mid M_n^{\tau_n} \longrightarrow_{\wedge CC} M_1'^{\tau_1} \mid \dots \mid M_n'^{\tau_n}$, then by rule [V-Par], we have that $M_1^{\tau_1} \bowtie N_1^{\rho_1}$ and ... and $M_n^{\tau_n} \bowtie N_n^{\rho_n}$ and by rule [E-Par], $\forall i$. either $M_i^{\tau_i}$ is a result and $M_i^{\tau_i} = M_i'^{\tau_i}$ or $M_i^{\tau_1} \longrightarrow_{\wedge CC} M_i'^{\tau_i}$ and $\exists i$. $M_i^{\tau_i}$ is not a result.

    For all i such that $M_i^{\tau_i}$ is a result, then either $M_i^{\tau_i} = v_i^{\tau_i}$ or $M_i^{\tau_i} = wrong^{\tau_i}$. If $M_i^{\tau_i} = v_i^{\tau_i}$, then by lemma 6.12, we have that $N_i^{\rho_i} \longrightarrow^*_{\wedge CC} r_i^{\rho_i}$ and $v_i^{\tau_i} \bowtie r_i^{\rho_i}$ and let $N_i'^{\rho_i} = r_i^{\rho_i}$. Therefore, $M_i'^{\tau_i} \bowtie N_i'^{\rho_i}$. If $M_i^{\tau_i} = wrong^{\tau_i}$, then by theorems 6.3 and 6.4, $N_i^{\rho_i} \longrightarrow^*_{\wedge CC} N_i'^{\rho_i}$ and by definition 5.1, $M_i'^{\tau_i} \bowtie N_i'^{\rho_i}$.

    For all i such that $M_i^{\tau_i} \longrightarrow_{\wedge CC} M_i'^{\tau_i}$, by the induction hypothesis, we have that $N_i^{\rho_i} \longrightarrow^*_{\wedge CC} N_i'^{\rho_i}$ and $M_i'^{\tau_i} \bowtie$

$N_i'^{\rho_i}$.

By rule [E-Par], we have that $N_1^{\rho_1} \mid \ldots \mid N_n^{\rho_n} \longrightarrow^*_{\wedge CC} N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n}$ and by rule [V-Par], we have that $M_1'^{\tau_1} \mid \ldots \mid M_n'^{\tau_n} \bowtie N_1'^{\rho_1} \mid \ldots \mid N_n'^{\rho_n}$.

- Rule [V-Cast]. There are seven possibilities:
  - Rule [E-Ctx] and context $E : \tau_1 \Rightarrow \tau_2$. If $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} M'^{\tau_1} : \tau_1 \Rightarrow \tau_2$ then by rule [V-Cast], we have that $M^{\tau_1} \bowtie N^{\rho_1}$, and by rule [E-Ctx], we have that $M^{\tau_1} \longrightarrow_{\wedge CC} M'^{\tau_1}$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} N'^{\rho_1}$ and $M'^{\tau_1} \bowtie N'^{\rho_1}$. By rule [E-Ctx], we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} N'^{\rho_1} : \rho_1 \Rightarrow \rho_2$, and by rule [V-Cast], we have that $M'^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N'^{\rho_1} : \rho_1 \Rightarrow \rho_2$.
  - Rule [E-Wrong] and context $E : \tau_1 \Rightarrow \tau_2$. If $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} wrong^{\tau_2}$ then by theorems 6.3 and 6.4, $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} N'^{\rho_2}$, and by rule [V-WrongL], $wrong^{\tau_2} \bowtie N'^{\rho_2}$.
  - Rule [EC-Identity]. If $v^{\tau} : \tau \Rightarrow \tau \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $v^{\tau} : \tau \Rightarrow \tau \longrightarrow_{\wedge CC} v^{\tau}$ then by rule [V-Cast], we have that $v^{\tau} \bowtie N^{\rho_1}$. By rule [V-CastR], we have that $v^{\tau} \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By lemma 6.12, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r^{\rho_2}$ and $v^{\tau} \bowtie r^{\rho_2}$.
  - Rule [EC-Succeed]. If $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \longrightarrow_{\wedge CC} v^G$ then by rule [V-Cast], $v^G : G \Rightarrow Dyn \bowtie N^{\rho_1}$. By lemma 6.12, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} r^{\rho_1}$ and $v^G : G \Rightarrow Dyn \bowtie r^{\rho_1}$. By rule [V-CastL], $v^G \bowtie r^{\rho_1}$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By lemma 6.11, $r^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r'^{\rho_2}$ and $v^G \bowtie r'^{\rho_2}$.
  - Rule [EC-Fail]. If $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \longrightarrow_{\wedge CC} wrong^{G_2}$ then by theorems 6.3 and 6.4, $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} N'^{\rho_2}$, and by rule [V-WrongL], $wrong^{G_2} \bowtie N'^{\rho_2}$.
  - Rule [EC-Ground]. If $v^{\tau} : \tau \Rightarrow Dyn \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $v^{\tau} : \tau \Rightarrow Dyn \longrightarrow_{\wedge CC} v^{\tau} : \tau \Rightarrow G : G \Rightarrow Dyn$, then by rule [V-Cast], we have that $v^{\tau} \bowtie N^{\rho_1}$. By lemma 6.12, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} r^{\rho_1}$ and $v^{\tau} \bowtie r^{\rho_1}$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [V-Cast], we have that $v^{\tau} : \tau \Rightarrow G \bowtie r^{\rho_1} : \rho_1 \Rightarrow \rho_2$, and by rule [V-CastL], we have that $v^{\tau} : \tau \Rightarrow G : G \Rightarrow Dyn \bowtie r^{\rho_1} : \rho_1 \Rightarrow \rho_2$.
  - Rule [EC-Expand]. If $v^{Dyn} : Dyn \Rightarrow \tau \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $v^{Dyn} : Dyn \Rightarrow \tau \longrightarrow_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$, then by rule [V-Cast], we have that $v^{Dyn} \bowtie N^{\rho_1}$. By lemma 6.12, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} r^{\rho_1}$ and $v^{Dyn} \bowtie r^{\rho_1}$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow^*_{\wedge CC} r^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [V-Cast], we have that $v^{Dyn} : Dyn \Rightarrow G \bowtie r^{\rho_1} : \rho_1 \Rightarrow \rho_2$. By rule [V-CastL], we have that $v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau \bowtie r^{\rho_1} : \rho_1 \Rightarrow \rho_2$.
- Rule [V-CastL]. There are seven possibilities:
  - Rule [E-Ctx] and context $E : \tau_1 \Rightarrow \tau_2$. If $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^{\rho}$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} M'^{\tau_1} : \tau_1 \Rightarrow \tau_2$ then by rule [V-CastL], we have that $M^{\tau_1} \bowtie N^{\rho}$ and by rule [E-Ctx], we have that $M^{\tau_1} \longrightarrow_{\wedge CC} M'^{\tau_1}$. By the induction hypothesis, we have that $N^{\rho} \longrightarrow^*_{\wedge CC} N'^{\rho}$ and $M'^{\tau_1} \bowtie N'^{\rho}$. By rule [V-CastL], we have that $M'^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N'^{\rho}$.
  - Rule [E-Wrong] and context $E : \tau_1 \Rightarrow \tau_2$. If $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \bowtie N^{\rho}$ and $M^{\tau_1} : \tau_1 \Rightarrow \tau_2 \longrightarrow_{\wedge CC} wrong^{\tau_2}$ then by theorems 6.3 and 6.4, $N^{\rho} \longrightarrow^*_{\wedge CC} N'^{\rho}$, and by rule [V-WrongL], $wrong^{\tau_2} \bowtie N'^{\rho}$.
  - Rule [EC-Identity]. If $v^{\tau} : \tau \Rightarrow \tau \bowtie N^{\rho}$ and $v^{\tau} : \tau \Rightarrow \tau \longrightarrow_{\wedge CC} v^{\tau}$ then by rule [V-CastL], we have that $v^{\tau} \bowtie N^{\rho}$. By lemma 6.12, we have that $N^{\rho} \longrightarrow^*_{\wedge CC} r^{\rho}$ and $v^{\tau} \bowtie r^{\rho}$.
  - Rule [EC-Succeed]. If $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \bowtie N^{\rho}$ and $v^G : G \Rightarrow Dyn : Dyn \Rightarrow G \longrightarrow_{\wedge CC} v^G$ then by rule [V-CastL], $v^G : G \Rightarrow Dyn \bowtie N^{\rho}$. By rule [V-CastL], $v^G \bowtie N^{\rho}$. By lemma 6.12, we have that $N^{\rho} \longrightarrow^*_{\wedge CC} r^{\rho}$ and $v^G \bowtie r^{\rho}$.
  - Rule [EC-Fail]. If $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \bowtie N^{\rho}$ and $v^{G_1} : G_1 \Rightarrow Dyn : Dyn \Rightarrow G_2 \longrightarrow_{\wedge CC} wrong^{G_2}$ then by theorems 6.3 and 6.4, $N^{\rho} \longrightarrow^*_{\wedge CC} N'^{\rho}$, and by rule [V-WrongL], $wrong^{G_2} \bowtie N'^{\rho}$.
  - Rule [EC-Ground]. If $v^{\tau} : \tau \Rightarrow Dyn \bowtie N^{\rho}$ and $v^{\tau} : \tau \Rightarrow Dyn \longrightarrow_{\wedge CC} v^{\tau} : \tau \Rightarrow G : G \Rightarrow Dyn$, then by rule [V-CastL], we have that $v^{\tau} \bowtie N^{\rho}$. By lemma 6.12, we have that $N^{\rho} \longrightarrow^*_{\wedge CC} r^{\rho}$ and $v^{\tau} \bowtie r^{\rho}$. By rule [V-CastL], we have that $v^{\tau} : \tau \Rightarrow G \bowtie r^{\rho}$, and by rule [V-CastL], we have that $v^{\tau} : \tau \Rightarrow G : G \Rightarrow Dyn \bowtie r^{\rho}$.
  - Rule [EC-Expand]. If $v^{Dyn} : Dyn \Rightarrow \tau \bowtie N^{\rho}$ and $v^{Dyn} : Dyn \Rightarrow \tau \longrightarrow_{\wedge CC} v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau$, then by rule [V-CastL], we have that $v^{Dyn} \bowtie N^{\rho}$. By lemma 6.12, we have that $N^{\rho} \longrightarrow^*_{\wedge CC} r^{\rho}$ and $v^{Dyn} \bowtie r^{\rho}$. By rule [V-CastL], we have that $v^{Dyn} : Dyn \Rightarrow G \bowtie r^{\rho}$, and by rule [V-CastL], we have that $v^{Dyn} : Dyn \Rightarrow G : G \Rightarrow \tau \bowtie r^{\rho}$.
- Rule [V-CastR]. If $M^{\tau} \bowtie N^{\rho_1} : \rho_1 \Rightarrow \rho_2$ and $M^{\tau} \longrightarrow_{\wedge CC} M'^{\tau}$ then by rule [V-CastR], we have that $M^{\tau} \bowtie N^{\rho_1}$. By the induction hypothesis, we have that $N^{\rho_1} \longrightarrow^*_{\wedge CC} N'^{\rho_1}$ and $M'^{\tau} \bowtie N'^{\rho_1}$. By rule [E-Ctx] and context $E : \rho_1 \Rightarrow \rho_2$, we have that $N^{\rho_1} : \rho_1 \Rightarrow \rho_2 \longrightarrow_{\wedge CC} N'^{\rho_1} : \rho_1 \Rightarrow \rho_2$, and by rule [V-CastR], we have that $M'^{\tau} \bowtie N'^{\rho_1} : \tau_1 \Rightarrow \tau_2$.

□

Theorem 6.16 (Confluency of Operational Semantics). *For all $\Pi^{\sigma} \bowtie \Upsilon^{v}$ such that $\emptyset \vdash_{\wedge CC} \Pi^{\sigma} : \sigma$ and $\emptyset \vdash_{\wedge CC} \Upsilon^{v} : v$, and assuming $\pi_1^{\sigma} \neq wrong^{\sigma}$, if $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} \pi_1^{\sigma}$ then $\Upsilon^{v} \longrightarrow^*_{\wedge CC} \pi_2^{v}$ and $\pi_1^{\sigma} \bowtie \pi_2^{v}$.*

Proof. By lemma 6.15 and induction on the length of the reduction sequence, applying theorem 6.4, we have that $\Pi^{\sigma} \longrightarrow^*_{\wedge CC} \pi_1^{\sigma}$ and $\Upsilon^{v} \longrightarrow^*_{\wedge CC} \Upsilon'^{v}$ and $\pi_1^{\sigma} \bowtie \Upsilon'^{v}$. By lemma 6.12 applied to each component, and by rule [E-Par], either $\Upsilon'^{v} \longrightarrow^*_{\wedge CC} \pi_2^{v}$ and $\pi_1^{\sigma} \bowtie \pi_2^{v}$, or $\Upsilon'^{v} \longrightarrow^*_{\wedge CC} \Upsilon''^{v}$ and by rule [E-Push], $\Upsilon''^{v} \longrightarrow_{\wedge CC} wrong^{v}$ and $\pi_1^{\sigma} \bowtie wrong^{v}$. □