Bounded Quantification for Gradual Typing

Harley Eades III and Michael Townsend

Computer and Information Sciences
Augusta University
Augusta, USA
heades@augusta.edu

Abstract. In an earlier paper we introduce a new categorical model based on retracts that combines static and dynamic typing. We then showed that our model gave rise to a new and simple type system which combines static and dynamic typing. In this paper, we extend this type system with bounded quantification and lists, and then develop a gradually typed surface language that uses our new type system as a core casting calculus. Finally, we prove the gradual guarantee as put forth by Siek et al.

1 Introduction

In a previous paper the authors [**?**] have shown that static and dynamic typing can be combined in a very simple and intuitive way by combining the work of Scott [3] and Lambek [2] on categorical models of the untyped and typed λ-calculus, respectively. First, add a new type ? read "the type of untyped programs" – also sometimes called the unknown type – and then add four new programs $\mathsf{split} : ? \to (? \to ?)$, $\mathsf{squash} : (? \to ?) \to ?$, $\mathsf{box}_C : C \to ?$, and $\mathsf{unbox}_C : ? \to C$, such that, $\mathsf{squash};\mathsf{split} = \mathsf{id}_{? \to ?}$ and $\mathsf{box}_C;\mathsf{unbox}_C = \mathsf{id}_C$. Categorically, split and squash, and box and unbox form two retracts. Then extending the simply typed λ-calculus with these two retracts results in a new core casting calculus, called Simply Typed Grady, for Siek and Taha's gradual functional type system [5]. Furthermore, the authors show that Siek and Taha's system can be given a categorical model in cartesian closed categories with the two retracts.

In this paper we extend Grady with bounded quantification and lists. We chose bounded quantification so that the bounds can be used to control which types are castable and which should not be. Currently, we will not allow polymorphic types to be cast to the unknown type, because we do not have a good model nor practical examples showing this is interesting and useful. We do this by adding a new bounds, $\mathbb{S}$, whose subtypes are all non-polymorphic types – referred to hence forth as simple types. Then we give box and unbox the following types:

$$\frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{box} : \forall (X <: \mathbb{S}).(X \to ?)}\ \mathsf{box} \quad \text{and} \quad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{unbox} : \forall (X <: \mathbb{S}).(? \to X)}\ \mathsf{unbox}$$

This differs from our previous work where we limited box and unbox to only atomic types, but then we showed that they could be extended to any type by combining box and unbox with split and squash. In this paper we take these extended versions as primitive.

Grady now consists of two languages: a surface language – called Surface Grady – and a core language – called Core Grady. The difference between the surface and the core is that the former is gradually typed while the latter is statically typed. Gradual typing is the combination of static and dynamic typing in such a way that one can program in dynamic style. That is, the programmer should not have to introduce explicit casts. The first functional gradually typed language is due to Siek and Taha [4] where they extend the typed λ-calculus with the unknown type ? and a new relation on types, called the type consistency relation, that indicated when types should be considered as being castable or not. For example, consider the function application rule of Surface Grady:

$$\frac{\begin{array}{l}\Gamma \vdash_{\mathsf{SG}} t_1 : C \\ \Gamma \vdash_{\mathsf{SG}} t_2 : A_2 \quad \Gamma \vdash A_2 \sim A_1 \quad \mathsf{fun}(C) = A_1 \to B_1\end{array}}{\Gamma \vdash_{\mathsf{SG}} t_1\, t_2 : B_1} \to_e$$

This rule depends on the relation $\Gamma \vdash A_2 \sim A_1$ which is the type consistency relation. It is reflexive and symmetric, but not transitive, or one could prove that any type is consistent with any other type. Thus, type consistency indicates exactly where explicit casts need to be inserted. This rule also depends on the partial function:

$$\begin{array}{l}\mathsf{fun}(?) = ? \to ? \\ \mathsf{fun}(A_1 \to B_1) = A_1 \to B_1\end{array}$$

As we will see below $\Gamma \vdash ? \sim A$ for any type A. As an example, suppose $\Gamma \vdash_{\mathsf{SG}} t_1 : ?$ and $\Gamma \vdash_{\mathsf{SG}} t_2 : \mathsf{Nat}$. Then based on the rule above, we know that $\Gamma \vdash ? \sim \mathsf{Nat}$, and $\mathsf{fun}(?) = ? \to ?$, thus, $\Gamma \vdash_{\mathsf{SG}} t_1\, t_2 : ?$ is typable. Notice there are no explicit casts. Using split and $\mathsf{box}_{\mathsf{Nat}}$ we can translate this application into Core Grady by adding the casts: $\Gamma \vdash_{\mathsf{CG}} (\mathsf{split}_{(? \to ?)}\, t_1)\,(\mathsf{box}_{\mathsf{Nat}}\, t_2) : ?$.

Subtyping in Core Grady is standard subtyping for bounded system F extended with the new bounds for simple types. One important point is that in Core Grady the unknown type is not a top type, and in fact, is only related to itself and $\mathbb{S}$. However, subtyping in the surface language is substantially different. Subtyping in Surface Grady is the combination of subtyping and type consistency called consistent subtyping due to Garcia [1]. We denote consistent subtyping by $\Gamma \vdash A \lesssim B$. Unlike Core Grady we have $\Gamma \vdash ? \lesssim A$ and $\Gamma \vdash A \lesssim ?$ for any type A. This gives us some flexibility when instantiating polymorphic functions. For example, suppose $\Gamma \vdash_{\mathsf{SG}} t : \forall(X <: \mathsf{Nat}).(X \to X)$. Then, $\Gamma \vdash_{\mathsf{SG}} [?]t : ? \to ?$ is typable, as well as, $\Gamma \vdash_{\mathsf{SG}} [\mathsf{Nat}]t : \mathsf{Nat} \to \mathsf{Nat}$ by subsumption. Similarly, if $\Gamma \vdash_{\mathsf{SG}} t : \forall(X <: ?).(X \to X)$, then we can instantiate t with any type at all. This seems very flexible, but it turns out that it does nothing more than what Core Grady allows when adding explicit casts.

One feature Surface Grady has that cannot be found in previous systems is allowing explicit casts in the surface language. Adding this feature actually increases the expressivity of the language. For example, consider the following Surface Grady program – here we use the concrete syntax from Grady's implementation, but it is very similar to Haskell and not far from the mathematical syntax:

2 Grady: A Categorically Inspired Gradual Type System

2.1 Surface Grady: A Gradual Type System

$$\frac{\Gamma \vdash A : \star}{\Gamma \vdash A \lesssim A}\,\text{refl} \qquad \frac{\Gamma \vdash A : \star}{\Gamma \vdash A \lesssim \top}\,S\_Top \qquad \frac{X <: A' \in \Gamma \quad \Gamma \vdash A' \sim A}{\Gamma \vdash X \lesssim A}\,\text{var}$$

$$\frac{\Gamma \vdash A \lesssim \mathbb{S}}{\Gamma \vdash A \lesssim ?}\,\mathsf{box} \qquad \frac{\Gamma \vdash A \lesssim \mathbb{S}}{\Gamma \vdash ? \lesssim A}\,\mathsf{unbox} \qquad \frac{\Gamma\ \text{Ok}}{\Gamma \vdash ? \lesssim \mathbb{S}}\,S\_USL$$

$$\frac{\Gamma\ \text{Ok}}{\Gamma \vdash \mathsf{Nat} \lesssim \mathbb{S}}\,S\_NatSL \qquad \frac{\Gamma\ \text{Ok}}{\Gamma \vdash \mathsf{Unit} \lesssim \mathbb{S}}\,S\_UnitSL \qquad \frac{\Gamma \vdash A \lesssim \mathbb{S}}{\Gamma \vdash \mathsf{List}\,A \lesssim \mathbb{S}}\,S\_ListSL$$

$$\frac{\Gamma \vdash A \lesssim \mathbb{S} \quad \Gamma \vdash B \lesssim \mathbb{S}}{\Gamma \vdash A \times B \lesssim \mathbb{S}}\,S\_ProdSL \qquad \frac{\Gamma \vdash A \lesssim \mathbb{S} \quad \Gamma \vdash B \lesssim \mathbb{S}}{\Gamma \vdash A \to B \lesssim \mathbb{S}}\,S\_ArrowSL$$

$$\frac{\Gamma \vdash A \lesssim B}{\Gamma \vdash (\mathsf{List}\,A) \lesssim (\mathsf{List}\,B)}\,\mathsf{List} \qquad \frac{\Gamma \vdash A_1 \lesssim A_2 \quad \Gamma \vdash B_1 \lesssim B_2}{\Gamma \vdash (A_1 \times B_1) \lesssim (A_2 \times B_2)}\,\times$$

$$\frac{\Gamma \vdash A_2 \lesssim A_1 \quad \Gamma \vdash B_1 \lesssim B_2}{\Gamma \vdash (A_1 \to B_1) \lesssim (A_2 \to B_2)}\,\to \qquad \frac{\Gamma, X <: A \vdash B_1 \lesssim B_2}{\Gamma \vdash (\forall (X <: A).B_1) \lesssim (\forall (X <: A).B_2)}\,\forall$$

Fig. 1. Subtyping for Surface Grady

2.2 Core Grady: The Casting Calculus

3 Analyzing Grady

Lemma 1 (Inclusion of Bounded System F). *Suppose t is fully annotated and does not contain any applications of* box *or* unbox*, and A is static. Then*

i. $\Gamma \vdash_F t : A$ *if and only if* $\Gamma \vdash_{\mathsf{SG}} t : A$, *and*
ii. $t \rightsquigarrow_F^* t'$ *if and only if* $t \rightsquigarrow^* t'$.

Proof. We give proof sketches for both parts. The interesting cases are the right-to-left directions of each part. If we simply remove all rules mentioning the

$$\frac{\Gamma \vdash A : \star}{\Gamma \vdash A \sim A}\,\text{refl} \qquad \frac{\Gamma \vdash A \lesssim \mathbb{S}}{\Gamma \vdash A \sim ?}\,\mathsf{box} \qquad \frac{\Gamma \vdash A \lesssim \mathbb{S}}{\Gamma \vdash ? \sim A}\,\mathsf{unbox}$$

$$\frac{\Gamma \vdash A \sim B}{\Gamma \vdash (\mathsf{List}\,A) \sim (\mathsf{List}\,B)}\,\mathsf{List} \qquad \frac{\Gamma \vdash A_2 \sim A_1 \quad \Gamma \vdash B_1 \sim B_2}{\Gamma \vdash (A_1 \to B_1) \sim (A_2 \to B_2)}\,\to$$

$$\frac{\Gamma \vdash A_1 \sim A_2 \quad \Gamma \vdash B_1 \sim B_2}{\Gamma \vdash (A_1 \times B_1) \sim (A_2 \times B_2)}\,\times \qquad \frac{\Gamma, X <: A \vdash B_1 \sim B_2}{\Gamma \vdash (\forall (X <: A).B_1) \sim (\forall (X <: A).B_2)}\,\forall$$

Fig. 2. Type consistency for Surface Grady

unknown type ? and the type consistency relation, and then remove box, unbox, and ? from the syntax of Surface Grady, then what we are left with is bounded system F. Since t is fully annotated and A is static, then $\Gamma \vdash_{\mathsf{SG}} t : A$ will hold within this fragment.

Moving on to part two, first, we know that t does not contain any occurrence of box or unbox and is fully annotated. This implies that t lives within the bounded system F fragment of Surface Grady. Thus, before evaluation of t Surface Grady will apply the cast insertion algorithm which will at most insert applications of the identity function into t producing a term $\widehat{t}$, but then after potentially more than one step of evaluation within Core Grady, those applications of the identity function will be β-reduced away resulting in $\widehat{t} \rightsquigarrow^* t \rightsquigarrow^* t'$. In addition, since t in Surface Grady is the exact same program as t in bounded system F, then we know $t \rightsquigarrow^*_F t'$ will hold.

Lemma 2 (Inclusion of DTLC). *Suppose t is a closed term of DTLC. Then*

i. $\cdot \vdash_{\mathsf{SG}} \lceil t \rceil : ?$*, and*
ii. $t \rightsquigarrow^*_{DTLC} t'$ *if and only if* $\lceil t \rceil \rightsquigarrow^* \lceil t' \rceil$*.*

Proof. In this case DTLC is embedded into the simply typed fragment of Grady, and hence, this proof is the same result proven by [4], and [5].

Lemma 3 (Left-to-Right Consistent Subtyping). *Suppose* $\Gamma \vdash A \lesssim B$*.*

i. $\Gamma \vdash A \sim A'$ *and* $\Gamma \vdash A' <: B$ *for some* A'*.*
ii. $\Gamma \vdash B' \sim B$ *and* $\Gamma \vdash A <: B'$ *for some* B'*.*

Proof. This is a proof by induction on $\Gamma \vdash A \lesssim B$. See Appendix B.1 for the complete proof.

Corollary 1 (Consistent Subtyping).

i. $\Gamma \vdash A \lesssim B$ *if and only if* $\Gamma \vdash A \sim A'$ *and* $\Gamma \vdash A' <: B$ *for some* A'*.*

$$\frac{x : A \in \Gamma \quad \Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} x : A}\ \mathrm{var} \qquad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{box} : \forall(X <: \mathbb{S}).(X \to ?)}\ \mathsf{box}$$

$$\frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{unbox} : \forall(X <: \mathbb{S}).(? \to X)}\ \mathsf{unbox} \qquad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{triv} : \mathsf{Unit}}\ \mathsf{Unit}$$

$$\frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} 0 : \mathsf{Nat}}\ \mathsf{zero} \qquad \frac{\Gamma \vdash_{\mathsf{SG}} t : A \quad \mathsf{nat}(A) = \mathsf{Nat}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{succ}\, t : \mathsf{Nat}}\ \mathsf{succ}$$

$$\frac{\begin{array}{l}\Gamma \vdash_{\mathsf{SG}} t : C \quad \mathsf{nat}(C) = \mathsf{Nat} \quad \Gamma \vdash A_1 \sim A \\ \Gamma \vdash_{\mathsf{SG}} t_1 : A_1 \quad \Gamma, x : \mathsf{Nat} \vdash_{\mathsf{SG}} t_2 : A_2 \quad \Gamma \vdash A_2 \sim A\end{array}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{case}\, t\, \mathsf{of}\, 0 \to t_1, (\mathsf{succ}\, x) \to t_2 : A}\ \mathsf{Nat}_e$$

$$\frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} [] : \forall(X <: \top).\mathsf{List}\, X}\ \mathrm{empty}$$

$$\frac{\Gamma \vdash_{\mathsf{SG}} t_1 : A_1 \quad \Gamma \vdash_{\mathsf{SG}} t_2 : A_2 \quad \mathsf{list}(A_2) = \mathsf{List}\, A_3 \quad \Gamma \vdash A_1 \sim A_3}{\Gamma \vdash_{\mathsf{SG}} t_1 :: t_2 : \mathsf{List}\, A_3}\ \mathsf{List}_i$$

$$\frac{\begin{array}{l}\Gamma \vdash_{\mathsf{SG}} t : C \quad \mathsf{list}(C) = \mathsf{List}\, A \\ \Gamma \vdash_{\mathsf{SG}} t_1 : B_1 \quad \Gamma, x : A, y : \mathsf{List}\, A \vdash_{\mathsf{SG}} t_2 : B_2 \quad \Gamma \vdash B_1 \sim B \quad \Gamma \vdash B_2 \sim B\end{array}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{case}\, t\, \mathsf{of}\, [] \to t_1, (x :: y) \to t_2 : B}\ \mathsf{List}_e$$

$$\frac{\Gamma \vdash_{\mathsf{SG}} t_1 : A_1 \quad \Gamma \vdash_{\mathsf{SG}} t_2 : A_2}{\Gamma \vdash_{\mathsf{SG}} (t_1, t_2) : A_1 \times A_2}\ \times_i \qquad \frac{\Gamma \vdash_{\mathsf{SG}} t : B \quad \mathsf{prod}(B) = A_1 \times A_2}{\Gamma \vdash_{\mathsf{SG}} \mathsf{fst}\, t : A_1}\ \times_{e_1}$$

$$\frac{\Gamma \vdash_{\mathsf{SG}} t : B \quad \mathsf{prod}(B) = A_1 \times A_2}{\Gamma \vdash_{\mathsf{SG}} \mathsf{snd}\, t : A_2}\ \times_{e_2} \qquad \frac{\Gamma, x : A \vdash_{\mathsf{SG}} t : B}{\Gamma \vdash_{\mathsf{SG}} \lambda(x : A).t : A \to B}\ \to_i$$

$$\frac{\begin{array}{l}\Gamma \vdash_{\mathsf{SG}} t_1 : C \\ \Gamma \vdash_{\mathsf{SG}} t_2 : A_2 \quad \Gamma \vdash A_2 \sim A_1 \quad \mathsf{fun}(C) = A_1 \to B_1\end{array}}{\Gamma \vdash_{\mathsf{SG}} t_1\, t_2 : B_1}\ \to_e$$

$$\frac{\Gamma, X <: A \vdash_{\mathsf{SG}} t : B}{\Gamma \vdash_{\mathsf{SG}} \Lambda(X <: A).t : \forall(X <: A).B}\ \forall_i$$

$$\frac{\Gamma \vdash_{\mathsf{SG}} t : \forall(X <: B).C \quad \Gamma \vdash A \lesssim B}{\Gamma \vdash_{\mathsf{SG}} [A]t : [A/X]C}\ \forall_e \qquad \frac{\Gamma \vdash_{\mathsf{SG}} t : A \quad \Gamma \vdash A \lesssim B}{\Gamma \vdash_{\mathsf{SG}} t : B}\ \mathrm{sub}$$

Fig. 3. Typing rules for Surface Grady

$$\frac{x : A \in \Gamma \quad \Gamma\,\mathrm{Ok}}{\Gamma \vdash x \Rightarrow x : A} \qquad \frac{\Gamma\,\mathrm{Ok}}{\Gamma \vdash \mathsf{box} \Rightarrow \mathsf{box} : \forall(X <: \mathbb{S}).(X \to ?)}$$

$$\frac{\Gamma\,\mathrm{Ok}}{\Gamma \vdash \mathsf{unbox} \Rightarrow \mathsf{unbox} : \forall(X <: \mathbb{S}).(? \to X)} \qquad \frac{\Gamma\,\mathrm{Ok}}{\Gamma \vdash 0 \Rightarrow 0 : \mathsf{Nat}}$$

$$\frac{\Gamma\,\mathrm{Ok}}{\Gamma \vdash \mathsf{triv} \Rightarrow \mathsf{triv} : \mathsf{Unit}} \qquad \frac{\Gamma \vdash t_1 \Rightarrow t_2 : ?}{\Gamma \vdash \mathsf{succ}\, t_1 \Rightarrow \mathsf{succ}\,(\mathsf{unbox}_{\mathsf{Nat}}\, t_2) : \mathsf{Nat}}$$

$$\frac{\Gamma \vdash t_1 \Rightarrow t_2 : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}\, t_1 \Rightarrow \mathsf{succ}\, t_2 : \mathsf{Nat}}$$

$$\frac{\begin{array}{l}\Gamma \vdash t \Rightarrow t' : ? \quad \Gamma \vdash A_1 \sim A \quad \mathsf{caster}(A_1, A) = c_1 \\ \Gamma \vdash t_1 \Rightarrow t_1' : A_1 \quad \Gamma, x : \mathsf{Nat} \vdash t_2 \Rightarrow t_2' : A_2 \quad \Gamma \vdash A_2 \sim A \quad \mathsf{caster}(A_2, A) = c_2\end{array}}{\Gamma \vdash (\mathsf{case}\, t\, \mathsf{of}\, 0 \to t_1, (\mathsf{succ}\, x) \to t_2) \Rightarrow (\mathsf{case}\,(\mathsf{unbox}_{\mathsf{Nat}}\, t')\, \mathsf{of}\, 0 \to (c_1\, t_1'), (\mathsf{succ}\, x) \to (c_2\, t_2')) : A}$$

$$\frac{\begin{array}{l}\Gamma \vdash t \Rightarrow t' : \mathsf{Nat} \quad \Gamma \vdash A_1 \sim A \quad \mathsf{caster}(A_1, A) = c_1 \\ \Gamma \vdash t_1 \Rightarrow t_1' : A_1 \quad \Gamma, x : \mathsf{Nat} \vdash t_2 \Rightarrow t_2' : A_2 \quad \Gamma \vdash A_2 \sim A \quad \mathsf{caster}(A_2, A) = c_2\end{array}}{\Gamma \vdash (\mathsf{case}\, t\, \mathsf{of}\, 0 \to t_1, (\mathsf{succ}\, x) \to t_2) \Rightarrow (\mathsf{case}\, t'\, \mathsf{of}\, 0 \to t_1', (\mathsf{succ}\, x) \to t_2') : A}$$

$$\frac{\Gamma \vdash t_1 \Rightarrow t_3 : A_1 \quad \Gamma \vdash t_2 \Rightarrow t_4 : A_2}{\Gamma \vdash (t_1, t_2) \Rightarrow (t_3, t_4) : A_1 \times A_2} \qquad \frac{\Gamma \vdash t_1 \Rightarrow t_2 : ?}{\Gamma \vdash \mathsf{fst}\, t_1 \Rightarrow \mathsf{fst}\,(\mathsf{split}_{(? \times ?)}\, t_2) : ?}$$

$$\frac{\Gamma \vdash t_1 \Rightarrow t_2 : A_1 \times A_2}{\Gamma \vdash \mathsf{fst}\, t_1 \Rightarrow \mathsf{fst}\, t_2 : A_1} \qquad \frac{\Gamma \vdash t_1 \Rightarrow t_2 : ?}{\Gamma \vdash \mathsf{snd}\, t_1 \Rightarrow \mathsf{snd}\,(\mathsf{split}_{(? \times ?)}\, t_2) : ?}$$

$$\frac{\Gamma \vdash t_1 \Rightarrow t_2 : A \times B}{\Gamma \vdash \mathsf{snd}\, t_1 \Rightarrow \mathsf{snd}\, t_2 : B} \qquad \frac{\Gamma\,\mathrm{Ok}}{\Gamma \vdash [] \Rightarrow [] : \forall(X <: \top).\mathsf{List}\, X}$$

$$\frac{\Gamma \vdash t_1 \Rightarrow t_1' : A_1 \quad \Gamma \vdash t_2 \Rightarrow t_2' : \mathsf{List}\, A_2 \quad \Gamma \vdash A_1 \lesssim A_2 \quad \mathsf{caster}(A_1, A_2) = c}{\Gamma \vdash (t_1 :: t_2) \Rightarrow ((c\, t_1') :: t_2') : \mathsf{List}\, A_2}$$

$$\frac{\begin{array}{l}\Gamma \vdash t \Rightarrow t' : ? \quad \mathsf{caster}(B_1, B) = c_1 \quad \mathsf{caster}(B_2, B) = c_2 \\ \Gamma \vdash t_1 \Rightarrow t_1' : B_1 \quad \Gamma, x : ?, y : \mathsf{List}\,? \vdash t_2 \Rightarrow t_2' : B_2 \quad \Gamma \vdash B_1 \sim B \quad \Gamma \vdash B_2 \sim B\end{array}}{\Gamma \vdash (\mathsf{case}\, t\, \mathsf{of}\, [] \to t_1, (x :: y) \to t_2) \Rightarrow (\mathsf{case}\,(\mathsf{split}_{(\mathsf{List}\,?)}\, t')\, \mathsf{of}\, [] \to (c_1\, t_1'), (x :: y) \to (c_2\, t_2')) : B}$$

$$\frac{\begin{array}{l}\Gamma \vdash t \Rightarrow t : \mathsf{List}\, A \quad \mathsf{caster}(B_1, B) = c_1 \quad \mathsf{caster}(B_2, B) = c_2 \\ \Gamma \vdash t_1 \Rightarrow t_1' : B_1 \quad \Gamma, x : A, y : \mathsf{List}\, A \vdash t_2 \Rightarrow t_2' : B_2 \quad \Gamma \vdash B_1 \sim B \quad \Gamma \vdash B_2 \sim B\end{array}}{\Gamma \vdash (\mathsf{case}\, t\, \mathsf{of}\, [] \to t_1, (x :: y) \to t_2) \Rightarrow (\mathsf{case}\, t'\, \mathsf{of}\, [] \to (c_1\, t_1'), (x :: y) \to (c_2\, t_2')) : B}$$

$$\frac{\Gamma, x : A_1 \vdash t_1 \Rightarrow t_2 : A_2}{\Gamma \vdash \lambda(x : A_1).t_1 \Rightarrow \lambda(x : A_1).t_2 : A_1 \to A_2}$$

$$\frac{\begin{array}{l}\Gamma \vdash t_1 \Rightarrow t_1' : ? \\ \Gamma \vdash t_2 \Rightarrow t_2' : A_2 \quad \mathsf{caster}(A_2, ?) = c\end{array}}{\Gamma \vdash t_1\, t_2 \Rightarrow (\mathsf{split}_{(? \to ?)}\, t_1')\,(c\, t_2') : ?}$$

$$\frac{\begin{array}{l}\Gamma \vdash t_2 \Rightarrow t_2' : A_2 \\ \Gamma \vdash t_1 \Rightarrow t_1' : A_1 \to B \quad \Gamma \vdash A_2 \sim A_1 \quad \mathsf{caster}(A_2, A_1) = c\end{array}}{\Gamma \vdash t_1\, t_2 \Rightarrow t_1'\,(c\, t_2') : B}$$

$$\frac{\Gamma, X <: A \vdash t_1 \Rightarrow t_2 : B}{\Gamma \vdash (\Lambda(X <: A).t_1) \Rightarrow (\Lambda(X <: A).t_2) : \forall(X <: A).B}$$

$$\frac{\Gamma \vdash t_1 \Rightarrow t_2 : \forall(X <: B).C \quad \Gamma \vdash A \sim A' \quad \Gamma \vdash A' <: B}{\Gamma \vdash ([A]t_1) \Rightarrow ([A']t_2) : [A'/X]C}$$

TODO

Fig. 5. Subtyping for Core Grady

$$\frac{x : A \in \Gamma \quad \Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} x : A}\ \mathrm{var} \qquad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{box} : \forall (X <: \mathbb{S}).(X \to ?)}\ \mathsf{box}$$

$$\frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{unbox} : \forall (X <: \mathbb{S}).(? \to X)}\ \mathsf{unbox} \qquad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{squash}_K : K \to ?}\ \mathsf{squash}$$

$$\frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{split}_K : ? \to K}\ \mathsf{split} \qquad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{triv} : \mathsf{Unit}}\ \mathsf{Unit} \qquad \frac{\Gamma\ \mathrm{Ok}}{\Gamma \vdash_{\mathsf{CG}} 0 : \mathsf{Nat}}\ \mathrm{zero}$$

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : \mathsf{Nat}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{succ}\, t : \mathsf{Nat}}\ \mathsf{succ} \qquad \frac{\begin{array}{l}\Gamma \vdash_{\mathsf{CG}} t : \mathsf{Nat}\\ \Gamma \vdash_{\mathsf{CG}} t_1 : A \quad \Gamma, x : \mathsf{Nat} \vdash_{\mathsf{CG}} t_2 : A\end{array}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{case}\, t \colon \mathsf{Nat}\, \mathsf{of}\ 0 \to t_1, (\mathsf{succ}\, x) \to t_2 : A}\ \mathsf{Nat}_e$$

$$\frac{\Gamma\ \mathrm{Ok} \quad \Gamma \vdash A : \star}{\Gamma \vdash_{\mathsf{CG}} [] : \forall (X <: ?).\mathsf{List}\, X}\ \mathrm{empty} \qquad \frac{\Gamma \vdash_{\mathsf{CG}} t_1 : A \quad \Gamma \vdash_{\mathsf{CG}} t_2 : \mathsf{List}\, A}{\Gamma \vdash_{\mathsf{CG}} t_1 :: t_2 : \mathsf{List}\, A}\ \mathsf{List}_i$$

$$\frac{\begin{array}{l}\Gamma \vdash_{\mathsf{CG}} t : \mathsf{List}\, A\\ \Gamma \vdash_{\mathsf{CG}} t_1 : B \quad \Gamma, x : A, y : \mathsf{List}\, A \vdash_{\mathsf{CG}} t_2 : B\end{array}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{case}\, t \colon \mathsf{List}\, A\, \mathsf{of}\ [] \to t_1, (x :: y) \to t_2 : B}\ \mathsf{List}_e$$

$$\frac{\Gamma \vdash_{\mathsf{CG}} t_1 : A_1 \quad \Gamma \vdash_{\mathsf{CG}} t_2 : A_2}{\Gamma \vdash_{\mathsf{CG}} (t_1, t_2) : A_1 \times A_2}\ \times_i \qquad \frac{\Gamma \vdash_{\mathsf{CG}} t : A_1 \times A_2}{\Gamma \vdash_{\mathsf{CG}} \mathsf{fst}\, t : A_1}\ \times_{e_1}$$

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : A_1 \times A_2}{\Gamma \vdash_{\mathsf{CG}} \mathsf{snd}\, t : A_2}\ \times_{e_2} \qquad \frac{\Gamma, x : A \vdash_{\mathsf{CG}} t : B}{\Gamma \vdash_{\mathsf{CG}} \lambda(x : A).t : A \to B}\ \to_i$$

$$\frac{\Gamma \vdash_{\mathsf{CG}} t_1 : A \to B \quad \Gamma \vdash_{\mathsf{CG}} t_2 : A}{\Gamma \vdash_{\mathsf{CG}} t_1\, t_2 : B}\ \to_e \qquad \frac{\Gamma, X <: A \vdash_{\mathsf{CG}} t : B}{\Gamma \vdash_{\mathsf{CG}} \Lambda(X <: A).t : \forall (X <: A).B}\ \forall_i$$

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : \forall (X <: B).C \quad \Gamma \vdash A <: B}{\Gamma \vdash_{\mathsf{CG}} [A]t : [A/X]C}\ \forall_e \qquad \frac{\Gamma \vdash_{\mathsf{CG}} t : A \quad \Gamma \vdash A <: B}{\Gamma \vdash_{\mathsf{CG}} t : B}\ \mathrm{sub}$$

$$\frac{}{\Gamma \vdash_{\mathsf{CG}} \mathsf{error}_A : A}\ \mathrm{error}$$

Fig. 6. Typing rules for Core Grady

TODO

Fig. 7. Reduction rules for Core Grady

$$\frac{\Gamma \vdash A \lesssim \mathbb{S}}{A \sqsubseteq ?}\ ? \qquad \frac{}{A \sqsubseteq A}\ \text{refl} \qquad \frac{A \sqsubseteq C \quad B \sqsubseteq D}{(A \to B) \sqsubseteq (C \to D)} \to$$

$$\frac{A \sqsubseteq C \quad B \sqsubseteq D}{(A \times B) \sqsubseteq (C \times D)} \times \qquad \frac{A \sqsubseteq B}{(\mathsf{List}\ A) \sqsubseteq (\mathsf{List}\ B)}\ \mathsf{List}$$

$$\frac{B_1 \sqsubseteq B_2}{(\forall (X <: A).B_1) \sqsubseteq (\forall (X <: A).B_2)}\ \forall$$

Fig. 8. Type Precision

ii. $\Gamma \vdash A \lesssim B$ *if and only if* $\Gamma \vdash B' \sim B$ *and* $\Gamma \vdash A <: B'$ *for some* B'.

Proof. The left-to-right direction of both cases easily follows from Lemma 3, and the right-to-left direction of both cases follows from induction on the subtyping derivation and Lemma 26.

Lemma 4 (Gradual Guarantee Part One). *If* $\Gamma \vdash_{\mathsf{SG}} t : A$, $t \sqsubseteq t'$, *and* $\Gamma \sqsubseteq \Gamma'$ *then* $\Gamma' \vdash_{\mathsf{SG}} t' : B$ *and* $A \sqsubseteq B$.

Proof. This is a proof by induction on $\Gamma \vdash_{\mathsf{SG}} t : A$; see Appendix B.4 for the complete proof.

Lemma 5 (Type Preservation for Cast Insertion). *If* $\Gamma \vdash_{\mathsf{SG}} t_1 : A$ *and* $\Gamma \vdash t_1 \Rightarrow t_2 : B$, *then* $\Gamma \vdash_{\mathsf{CG}} t_2 : B$ *and* $\Gamma \vdash A \sim B$.

Proof. The cast insertion algorithm is type directed and with respect to every term t_1 it will produce a term t_2 of the core language with the type A – this is straightforward to show by induction on the form of $\Gamma \vdash_{\mathsf{SG}} t_1 : A$ making use of typing for casting morphisms Lemma 30 – except in the case of type application. Please see Appendix B.5 for the complete proof.

Lemma 6 (Type Preservation). *If* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$ *and* $t_1 \rightsquigarrow t_2$, *then* $\Gamma \vdash_{\mathsf{CG}} t_2 : A$.

Proof. This proof holds by induction on $\Gamma \vdash_{\mathsf{CG}} t_1 : A$ with further case analysis on the structure the derivation $t_1 \rightsquigarrow t_2$.

Lemma 7 (Simulation of More Precise Programs). *Suppose* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$, $\Gamma \vdash t_1 \sqsubseteq t_1'$, $\Gamma \vdash_{\mathsf{CG}} t_1' : A'$, *and* $t_1 \rightsquigarrow t_2$. *Then* $t_1' \rightsquigarrow^* t_2'$ *and* $\Gamma \vdash t_2 \sqsubseteq t_2'$ *for some* t_2'.

Proof. This proof holds by induction on $\Gamma \vdash_{\mathsf{CG}} t_1 : A_1$. See Appendix B.6 for the complete proof.

Theorem 1 (Gradual Guarantee).

i. If $\cdot \vdash_{\mathsf{SG}} t : A$ *and* $t \sqsubseteq t'$*, then* $\cdot \vdash_{\mathsf{SG}} t' : B$ *and* $A \sqsubseteq B$*.*
ii. Suppose $\cdot \vdash_{\mathsf{CG}} t : A$ *and* $\cdot \vdash t \sqsubseteq t'$*. Then*
a. if $t \rightsquigarrow^* v$*, then* $t' \rightsquigarrow^* v'$ *and* $\cdot \vdash v \sqsubseteq v'$*,*
b. if $t \uparrow$*, then* $t' \uparrow$*,*
c. if $t' \rightsquigarrow^* v'$*, then* $t \rightsquigarrow^* v$ *where* $\cdot \vdash v \sqsubseteq v'$*, or* $t \rightsquigarrow^* \mathsf{error}_A$*, and*
d. if $t' \uparrow$*, then* $t \uparrow$ *or* $t \rightsquigarrow^* \mathsf{error}_A$*.*

Proof. This result follows from the same proof as [5], and so, we only give a brief summary. Part i. holds by Lemma 4, and Part ii. follows from simulation of more precise programs (Lemma 7).

References

1. Garcia, R., Clark, A.M., Tanter, E.: Abstracting gradual typing. In: Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. pp. 429–442. POPL '16, ACM, New York, NY, USA (2016)
2. Lambek, J.: From lambda calculus to cartesian closed categories. To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism pp. 376–402 (1980)
3. Scott, D.: Relating theories of the lambda-calculus. In: To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism (eds. Hindley and Seldin). pp. 403–450. Academic Press (1980)
4. Siek, J.G., Taha, W.: Gradual typing for functional languages. In: Scheme and Functional Programming Workshop. 1, vol. 6, pp. 81–92 (2006)
5. Siek, J.G., Vitousek, M.M., Cimini, M., Boyland, J.T.: Refined Criteria for Gradual Typing. In: Ball, T., Bodik, R., Krishnamurthi, S., Lerner, B.S., Morrisett, G. (eds.) 1st Summit on Advances in Programming Languages (SNAPL 2015). Leibniz International Proceedings in Informatics (LIPIcs), vol. 32, pp. 274–293. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2015)

A Auxiliary Results with Proofs

Lemma 8 (Kinding).

i. If $\Gamma \vdash A \sim B$*, then* $\Gamma \vdash A : \star$ *and* $\Gamma \vdash B : \star$*.*
ii. If $\Gamma \vdash A \lesssim B$*, then* $\Gamma \vdash A : \star$ *and* $\Gamma \vdash B : \star$*.*
iii. If $\Gamma \vdash_{\mathsf{SG}} t : A$*, then* $\Gamma \vdash A : \star$*.*

Proof. This proof holds by straightforward induction the form of each assumed judgment.

Lemma 9 (Strengthening for Kinding). *If* $\Gamma, x : A \vdash B : \star$*, then* $\Gamma \vdash B : \star$*.*

Proof. This proof holds by straightforward induction on the form of $\Gamma, x : A \vdash B : \star$.

Lemma 10 (Inversion for Type Precision). *Suppose $\Gamma \vdash A : \star$, $\Gamma \vdash B : \star$, and $A \sqsubseteq B$. Then:*

i. if $A = ?$, then $\Gamma \vdash B \lesssim \mathbb{S}$.
ii. if $A = A_1 \to B_1$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = A_2 \to B_2$, $A_1 \sqsubseteq A_2$, and $B_1 \sqsubseteq B_2$.
iii. if $A = A_1 \times B_1$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = A_2 \times B_2$, $A_1 \sqsubseteq A_2$, and $B_1 \sqsubseteq B_2$.
iv. if $A = \mathsf{List}\, A_1$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = \mathsf{List}\, A_2$ and $A_1 \sqsubseteq A_2$.
v. if $A = \forall(X <: A_1).B_1$, then $B = \forall(X <: A_1).B_1$ and $B_1 \sqsubseteq B_2$.

Proof. This proof holds by straightforward induction on the form of $A \sqsubseteq B$.

Lemma 11 (Surface Grady Inversion for Term Precision). *Suppose $t \sqsubseteq t'$. Then:*

i. if $t = \mathsf{succ}\, t_1$, then $t' = \mathsf{succ}\, t_2$ and $t_1 \sqsubseteq t_2$.
ii. if $t = (\mathsf{case}\, t_1\, \mathsf{of}\, 0 \to t_2, (\mathsf{succ}\, x) \to t_3)$, then $t' = (\mathsf{case}\, t'_1\, \mathsf{of}\, 0 \to t'_2, (\mathsf{succ}\, x) \to t'_3)$, $t_1 \sqsubseteq t'_1$, $t_2 \sqsubseteq t'_2$, and $t_3 \sqsubseteq t'_3$.
iii. if $t = (t_1, t_2)$, then $t' = (t'_1, t'_2)$, $t_1 \sqsubseteq t'_1$, and $t_2 \sqsubseteq t'_2$.
iv. if $t = \mathsf{fst}\, t_1$, then $t' = \mathsf{fst}\, t'_1$ and $t_1 \sqsubseteq t'_1$.
v. if $t = \mathsf{snd}\, t_1$, then $t' = \mathsf{snd}\, t'_1$ and $t_1 \sqsubseteq t'_1$.
vi. if $t = t_1 :: t_2$, then $t' = t'_1 :: t'_2$, $t_1 \sqsubseteq t'_1$, and $t_2 \sqsubseteq t'_2$.
vii. if $t = (\mathsf{case}\, t_1\, \mathsf{of}\, [] \to t_2, (x :: y) \to t_3)$, then $t' = (\mathsf{case}\, t'_1\, \mathsf{of}\, [] \to t'_2, (x :: y) \to t'_3)$, $t_1 \sqsubseteq t'_1$, $t_2 \sqsubseteq t'_2$, and $t_3 \sqsubseteq t'_3$.
viii. if $t = \lambda(x : A_1).t_1$, then $t' = \lambda(x : A_1).t'_1$ and $t_1 \sqsubseteq t'_1$.
ix. if $t = (t_1\, t_2)$, then $t' = (t'_1\, t'_2)$, $t_1 \sqsubseteq t'_1$, and $t_2 \sqsubseteq t'_2$.
x. if $t = \Lambda(X <: A_1).t_1$, then $t' = \Lambda(X <: A_1).t'_1$ and $t_1 \sqsubseteq t'_1$.
xi. if $t = [A]t_1$, then $t' = [A]t'_1$ and $t_1 \sqsubseteq t'_1$.

Proof. This proof holds by straightforward induction on the form of $t \sqsubseteq t'$.

Lemma 12 (Inversion for Type Consistency). *Suppose $\Gamma \vdash A \sim B$. Then:*

i. if $A = ?$, then $\Gamma \vdash B \lesssim \mathbb{S}$.
ii. if $A = \mathsf{List}\, A'$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = \mathsf{List}\, B'$ and $\Gamma \vdash A' \sim B'$.
iii. if $A = A_1 \to B_1$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = A_2 \to B_2$, $\Gamma \vdash A_2 \sim A_1$, and $\Gamma \vdash B_1 \sim B_2$.
iv. if $A = A_1 \to B_1$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = A_2 \to B_2$, $\Gamma \vdash A_2 \sim A_1$, and $\Gamma \vdash B_1 \sim B_2$.
v. if $A = A_1 \times B_1$, then $B = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $B = A_2 \times B_2$, $\Gamma \vdash A_1 \sim A_2$, and $\Gamma \vdash B_1 \sim B_2$.
vi. if $A = \forall(X <: A_1).B_1$, then $B = \forall(X <: A_1).B_2$ and $\Gamma, X <: A_1 \vdash B_1 \sim B_2$.

Proof. This proof holds by straightforward induction on the the form of $\Gamma \vdash A \sim B$.

Lemma 13 (Inversion for Consistent Subtyping). *Suppose $\Gamma \vdash A \lesssim B$. Then:*

i. *if $A = ?$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ or $\Gamma \vdash B \lesssim \mathbb{S}$.*
ii. *if $A = X$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, or $X <: B' \in \Gamma$ and $\Gamma \vdash B' \sim B$.*
iii. *if $A = \mathsf{Nat}$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, or $B = \mathbb{S}$.*
iv. *if $A = \mathsf{Unit}$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, or $B = \mathbb{S}$.*
v. *if $A = \mathsf{List}\, A_1$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, $B = \mathbb{S}$ and $\Gamma \vdash A_1 \lesssim \mathbb{S}$, or $B = \mathsf{List}\, A'_1$ and $\Gamma \vdash A_1 \lesssim A'_1$.*
vi. *if $A = A_1 \to B_1$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, $B = \mathbb{S}$, $\Gamma \vdash A_1 \lesssim \mathbb{S}$ and $\Gamma \vdash B_1 \lesssim \mathbb{S}$, or $B = A'_1 \to B'_1$, $\Gamma \vdash A'_1 \lesssim A_1$, and $\Gamma \vdash B_1 \lesssim B'_1$.*
vii. *if $A = A_1 \times B_1$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, $B = \mathbb{S}$, $\Gamma \vdash A_1 \lesssim \mathbb{S}$ and $\Gamma \vdash B_1 \lesssim \mathbb{S}$, or $B = A'_1 \times B'_1$, $\Gamma \vdash A_1 \lesssim A'_1$, and $\Gamma \vdash B_1 \lesssim B'_1$.*
viii. *if $A = \forall(X <: A_1).B_1$, then $B = A$ and $\Gamma \vdash A : \star$, $B = \top$ and $\Gamma \vdash A : \star$, or $B = \forall(X <: A_1).B'_1$ and $\Gamma, X <: A_1 \vdash B_1 \lesssim B'_1$.*

Proof. This proof holds by straightforward induction on the the form of $\Gamma \vdash A \lesssim B$.

Lemma 14 (Symmetry for Type Consistency). *If $\Gamma \vdash A \sim B$, then $\Gamma \vdash B \sim A$.*

Proof. This holds by straightforward induction on the form of $\Gamma \vdash A \sim B$.

Lemma 15. *If $\Gamma \vdash A <: B$, then $\Gamma \vdash A \lesssim B$.*

Proof. This proof holds by straightforward induction on $\Gamma \vdash A <: B$.

Lemma 16. *if $\Gamma \vdash A \sim B$, then $\Gamma \vdash A \lesssim B$.*

Proof. By straightforward induction on $\Gamma \vdash A \sim B$.

Lemma 17 (Type Precision and Consistency). *Suppose $\Gamma \vdash A : \star$ and $\Gamma \vdash B : \star$. Then if $A \sqsubseteq B$, then $\Gamma \vdash A \sim B$.*

Proof. This proof holds by straightforward induction on $A \sqsubseteq B$.

Corollary 2 (Type Precision and Subtyping). *Suppose $\Gamma \vdash A : \star$ and $\Gamma \vdash B : \star$. Then if $A \sqsubseteq B$, then $\Gamma \vdash A \lesssim B$.*

Proof. This easily follows from the previous two lemmas.

Lemma 18. *Suppose $\Gamma \vdash A : \star$, $\Gamma \vdash B : \star$, and $\Gamma \vdash C : \star$. If $A \sqsubseteq B$ and $A \sqsubseteq C$, then $\Gamma \vdash B \sim C$.*

Proof. It must be the case that either $B \sqsubseteq C$ or $C \sqsubseteq B$, but in both cases we know $\Gamma \vdash B \sim C$ by Lemma 17.

Lemma 19 (Transitivity for Type Precision). *If $A \sqsubseteq B$ and $B \sqsubseteq C$, then $A \sqsubseteq C$.*

Proof. This proof holds by straightforward induction on $A \sqsubseteq B$ with a case analysis over $B \sqsubseteq C$.

Lemma 20. *If $\Gamma \vdash A \sim B$, then $A \sqsubseteq B$ or $B \sqsubseteq A$.*

Proof. This proof holds by straightforward induction over $\Gamma \vdash A \sim B$.

Lemma 21. *If $\Gamma \vdash A \lesssim B$ and $A \sqsubseteq A'$, then $B \sqsubseteq A'$ or $A' \sqsubseteq B$.*

Proof. Suppose $\Gamma \vdash A \lesssim B$ and $A \sqsubseteq A'$. The former implies that $A \sqsubseteq B$ or $B \sqsubseteq A$ by Lemma 3 and Lemma 20. At this point the result easily follows.

Lemma 22. *Suppose $A \sqsubseteq B$. Then*

i. If $\mathsf{nat}(A) = \mathsf{Nat}$, *then* $\mathsf{nat}(B) = \mathsf{Nat}$.
ii. If $\mathsf{list}(A) = \mathsf{List}\ C$, *then* $\mathsf{list}(B) = \mathsf{List}\ C'$ *and* $C \sqsubseteq C'$.
iii. If $\mathsf{fun}(A) = A_1 \to A_2$, *then* $\mathsf{fun}(B) = A'_1 \to A'_2$, $A_1 \sqsubseteq A'_1$, *and* $A_2 \sqsubseteq A'_2$.

Proof. This proof holds by straightforward induction on $A \sqsubseteq B$.

Lemma 23. *If $\Gamma \vdash A \sim B$, $\Gamma \vdash C : \star$, and $A \sqsubseteq C$, then $\Gamma \vdash C \sim B$.*

Proof. Suppose $\Gamma \vdash A \sim B$ and $A \sqsubseteq C$. Then we know that $A \sqsubseteq B$ or $B \sqsubseteq A$. If the former, then we know that $\Gamma \vdash C \sim B$. If the latter, then we obtain $B \sqsubseteq C$ by transitivity, and $\Gamma \vdash B \sim C$ which implies that $\Gamma \vdash C \sim B$ by symmetry.

Lemma 24. *If Γ' Ok, $\Gamma \sqsubseteq \Gamma'$ and $\Gamma \vdash A \sim B$, then $\Gamma' \vdash A \sim B$.*

Proof. This proof holds by straightforward induction on $\Gamma \vdash A \sim B$.

Lemma 25 (Subtyping Context Precision). *If $\Gamma \vdash A \lesssim B$ and $\Gamma \sqsubseteq \Gamma'$, then $\Gamma' \vdash A \lesssim B$.*

Proof. Context precision does not manipulate the bounds on type variables, and thus, with respect to subtyping Γ and Γ' are essentially equivalent.

Lemma 26 (Simply Typed Consistent Types are Subtypes of $\mathbb{S}$). *If $\Gamma \vdash A \lesssim \mathbb{S}$ and $\Gamma \vdash A \sim B$, then $\Gamma \vdash B \lesssim \mathbb{S}$.*

Proof. This holds by straightforward induction on the form of $\Gamma \vdash A \lesssim \mathbb{S}$.

Lemma 27 (Type Precision Preserves $\mathbb{S}$).

i. If $\Gamma \vdash B : \star$, $\Gamma \vdash A \lesssim \mathbb{S}$ *and* $A \sqsubseteq B$, *then* $\Gamma \vdash B \lesssim \mathbb{S}$.
ii. If $\Gamma \vdash A : \star$, $\Gamma \vdash B \lesssim \mathbb{S}$ *and* $A \sqsubseteq B$, *then* $\Gamma \vdash A \lesssim \mathbb{S}$.

Proof. Both cases follow by induction on the assumed consistent subtyping derivation.

Lemma 28 (Congruence of Type Consistency Along Type Precision).

i. If $A_1 \sqsubseteq A'_1$ and $\Gamma \vdash A_1 \sim A_2$ then $\Gamma \vdash A'_1 \sim A_2$.
ii. If $A_2 \sqsubseteq A'_2$ and $\Gamma \vdash A_1 \sim A_2$ then $\Gamma \vdash A_1 \sim A'_2$.

Proof. Both parts hold by induction on the assumed type consistency judgment. See Appendix B.2 for the complete proof.

Corollary 3 (Congruence of Type Consistency Along Type Precision Condensed). *If $A_1 \sqsubseteq A'_1$, $A_2 \sqsubseteq A'_2$, and $\Gamma \vdash A_1 \sim A_2$ then $\Gamma \vdash A'_1 \sim A'_2$.*

Lemma 29 (Congruence of Subtyping Along Type Precision). *Suppose $\Gamma \vdash B : \star$ and $A \sqsubseteq B$.*

i. If $\Gamma \vdash A \lesssim C$ then $\Gamma \vdash B \lesssim C$.
ii. If $\Gamma \vdash C \lesssim A$ then $\Gamma \vdash C \lesssim B$.

Proof. This is a proof by induction on the form of $A \sqsubseteq B$; see Appendix B.3 for the complete proof.

Corollary 4 (Congruence of Subtyping Along Type Precision). *If $A_1 \sqsubseteq A_2$, $B_1 \sqsubseteq B_2$, and $\Gamma \vdash A_1 \lesssim B_1$, then $\Gamma \vdash A_2 \lesssim B_2$.*

Lemma 30 (Typing Casting Morphisms). *If $\Gamma \vdash A \sim B$ and $\mathsf{caster}(A, B) = c$, then $\Gamma \vdash_{\mathsf{CG}} c : A \to B$.*

Proof. This proof holds similarly to how we constructed casting morphisms in the categorical model. See Lemma **??**.

Lemma 31 (Substitution for Consistent Subtyping). *If $\Gamma, X <: B_1 \vdash B_2 \lesssim B_3$ and $\Gamma \vdash A_1 \lesssim B_1$, then $\Gamma \vdash [A_1/X]B_2 \lesssim [A_1/X]B_3$.*

Proof. This holds by straightforward induction on the form of $\Gamma, X <: B_1 \vdash B_2 \lesssim B_3$.

Lemma 32 (Substitution for Reflexive Type Consistency). *If $\Gamma, X <: B_1 \vdash B \sim B$, $\Gamma \vdash A_1 \sim A_2$, and $\Gamma \vdash A_2 <: B_1$, then $\Gamma \vdash [A_1/X]B \sim [A_2/X]B$.*

Proof. This holds by straightforward induction on the form of B.

Lemma 33 (Substitution for Type Consistency). *If $\Gamma, X <: B_1 \vdash B_2 \sim B_3$, $\Gamma \vdash A_1 \sim A_2$, and $\Gamma \vdash A_1 <: B_1$, then $\Gamma \vdash [A_1/X]B_2 \sim [A_2/X]B_3$.*

Proof. This holds by straightforward induction on $\Gamma, X <: B_1 \vdash B_2 \sim B_3$ using both substitution for consistent subtyping (Lemma 31) and substitution for reflexive type consistent (Lemma 32).

Lemma 34 (Typing for Type Precision). *If $\Gamma \vdash_{\mathsf{SG}} t_1 : A$, $t_1 \sqsubseteq t_2$, and $\Gamma \sqsubseteq \Gamma'$, then $\Gamma' \vdash_{\mathsf{SG}} t_2 : B$ and $A \sqsubseteq B$.*

Proof. This proof holds by induction on $\Gamma \vdash_{\mathsf{SG}} t_1 : A$ with a case analysis over $t_1 \sqsubseteq t_2$.

Lemma 35 (Substitution for Term Precision).

i. *If $\Gamma, x : A \vdash t_1 \sqsubseteq t_2$ and $\Gamma \vdash t_1' \sqsubseteq t_2'$, then $\Gamma \vdash [t_1'/x]t_1 \sqsubseteq [t_2'/x]t_2$.*
ii. *If $\Gamma, X <: A_2 \vdash t_1 \sqsubseteq t_2$ and $A_1 \sqsubseteq A_1'$, then $\Gamma \vdash [A_1/X]t_1 \sqsubseteq [A_1'/X]t_2$.*

Proof. This proof of part one holds by straightforward induction on $\Gamma, x : A \vdash t_1 \sqsubseteq t_2$, and the proof of part two holds by straightforward induction on $\Gamma, X <: A_2 \vdash t_1 \sqsubseteq t_2$.

Lemma 36 (Typeability Inversion).

i. *If $\Gamma \vdash_{\mathsf{CG}} \mathsf{succ}\, t : A$, then $\Gamma \vdash_{\mathsf{CG}} t : A'$ for some A'.*
ii. *If $\Gamma \vdash_{\mathsf{CG}} \mathsf{case}\, t \colon \mathsf{Nat}\,\mathsf{of}\; 0 \to t_1, (\mathsf{succ}\, x) \to t_2 : A$, then $\Gamma \vdash_{\mathsf{CG}} t : A_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : A_2$, and $\Gamma, x : \mathsf{Nat} \vdash_{\mathsf{CG}} t_2 : A_3$ for types A_1, A_2, A_3.*
iii. *If $\Gamma \vdash_{\mathsf{CG}} (t_1, t_2) : A$, then $\Gamma \vdash_{\mathsf{CG}} t_1 : A_1$ and $\Gamma \vdash_{\mathsf{CG}} t_2 : A_2$ for types A_1 and A_2.*
iv. *If $\Gamma \vdash_{\mathsf{CG}} \Lambda(X <: B).t : A$, then $\Gamma, X <: B \vdash_{\mathsf{CG}} t : A_1$ for some type A_1.*
v. *If $\Gamma \vdash_{\mathsf{CG}} [B]t : A$, then $\Gamma \vdash_{\mathsf{CG}} t : A_1$ for some type A_1.*
vi. *If $\Gamma \vdash_{\mathsf{CG}} \lambda(x : B).t : A$, then $\Gamma, x : B \vdash_{\mathsf{CG}} t : A_1$ for some type A_1.*
vii. *If $\Gamma \vdash_{\mathsf{CG}} t_1\, t_2 : A$, then $\Gamma \vdash_{\mathsf{CG}} t_1 : A_1$ and $\Gamma \vdash_{\mathsf{CG}} t_2 : A_2$ for types A_1 and A_2.*
viii. *If $\Gamma \vdash_{\mathsf{CG}} \mathsf{fst}\, t : A$, then $\Gamma \vdash_{\mathsf{CG}} t : A_1$ for some type A_1.*
ix. *If $\Gamma \vdash_{\mathsf{CG}} \mathsf{snd}\, t : A$, then $\Gamma \vdash_{\mathsf{CG}} t : A_1$ for some type A_1.*
x. *If $\Gamma \vdash_{\mathsf{CG}} t_1 :: t_2 : A$, then $\Gamma \vdash_{\mathsf{CG}} t_1 : A_1$ and $\Gamma \vdash_{\mathsf{CG}} t_2 : A_2$ for some types A_1 and A_2.*
xi. *If $\Gamma \vdash_{\mathsf{CG}} \mathsf{case}\, t \colon \mathsf{List}\, B\, \mathsf{of}\; [] \to t_1, (x :: y) \to t_2 : A$, then $\Gamma \vdash_{\mathsf{CG}} t : A_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : A_2$, and $\Gamma, x : A, y : \mathsf{List}\, A \vdash_{\mathsf{CG}} t_2 : A_3$ for types A_1, A_2, A_3.*

Lemma 37 (Inversion for Term Precision for Core Grady). *Suppose $\Gamma \vdash t_1 \sqsubseteq t_2$.*

i. *If $t_1 = x$, then one of the following is true:*
   a. *$t_2 = x$, $x : A \in \Gamma$, and Γ Ok*
   b. *$t_2 = \mathsf{box}_A\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$*
   c. *$t_2 = \mathsf{squash}_K\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$*
ii. *If $t_1 = \mathsf{split}_{K_1}$, then one of the following is true:*
   a. *$t_2 = \mathsf{split}_{K_2}$ and $K_1 \sqsubseteq K_2$*
   b. *$t_2 = \mathsf{box}_A\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$*
   c. *$t_2 = \mathsf{squash}_K\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$*
iii. *If $t_1 = \mathsf{squash}_{K_1}$, then one of the following is true:*

a. $t_2 = \mathsf{squash}_{K_2}$ *and* $K_1 \sqsubseteq K_2$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

iv. If $t_1 = \mathsf{box}$*, then one of the following is true:*
a. $t_2 = \mathsf{box}$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

v. If $t_1 = \mathsf{unbox}$*, then one of the following is true:*
a. $t_2 = \mathsf{unbox}$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

vi. If $t_1 = 0$*, then one of the following is true:*
a. $t_2 = 0$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

vii. If $t_1 = \mathsf{triv}$*, then one of the following is true:*
a. $t_2 = \mathsf{triv}$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

viii. If $t_1 = []$*, then one of the following is true:*
a. $t_2 = []$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

ix. If $t_1 = \mathsf{succ}\, t'_1$*, then one of the following is true:*
a. $t_2 = \mathsf{succ}\, t'_2$ *and* $\Gamma \vdash t'_1 \sqsubseteq t'_2$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

x. If $t_1 = \mathsf{case}\, t'_1 : \mathsf{Nat}\,\mathsf{of}\, 0 \to t'_2, (\mathsf{succ}\, x) \to t'_3$*, then one of the following is true:*
a. $t_2 = \mathsf{case}\, t'_4 : \mathsf{Nat}\,\mathsf{of}\, 0 \to t'_5, (\mathsf{succ}\, x) \to t'_6$, $\Gamma \vdash t'_1 \sqsubseteq t'_4$, $\Gamma \vdash t'_2 \sqsubseteq t'_5$, *and* $\Gamma, x : \mathsf{Nat} \vdash t'_3 \sqsubseteq t'_6$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xi. If $t_1 = (t'_1, t'_2)$*, then one of the following is true:*
a. $t_2 = (t'_3, t'_4)$, $\Gamma \vdash t'_1 \sqsubseteq t'_3$, *and* $\Gamma \vdash t'_2 \sqsubseteq t'_4$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xii. If $t_1 = \mathsf{fst}\, t'_1$*, then one of the following is true:*
a. $t_2 = \mathsf{fst}\, t'_2$ *and* $\Gamma \vdash t'_1 \sqsubseteq t'_2$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xiii. If $t_1 = \mathsf{snd}\, t'_1$*, then one of the following is true:*
a. $t_2 = \mathsf{snd}\, t'_2$ *and* $\Gamma \vdash t'_1 \sqsubseteq t'_2$
b. $t_2 = \mathsf{box}_A\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\, t_1$ *and* $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xiv. If $t_1 = t'_1 :: t'_2$*, then one of the following is true:*

a. $t_2 = t'_3 :: t'_4$, $\Gamma \vdash t'_1 \sqsubseteq t'_3$, and $\Gamma \vdash t'_2 \sqsubseteq t'_4$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xv. *If $t_1 = \mathsf{case}\ t'_1 : \mathsf{List}\ A_1\ \mathsf{of}\ [] \to t'_2, (x :: y) \to t'_3$, then one of the following is true:*

a. $t_2 = \mathsf{case}\ t'_4 : \mathsf{List}\ A_2\ \mathsf{of}\ [] \to t'_5, (x :: y) \to t'_6$, $\Gamma \vdash t'_1 \sqsubseteq t'_4$, $\Gamma \vdash t'_2 \sqsubseteq t'_5$, and $\Gamma, x : A_2, y : \mathsf{List}\ A_2 \vdash t'_3 \sqsubseteq t'_6$, and $A_1 \sqsubseteq A_2$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xvi. *If $t_1 = \lambda(x : A_1).t_1$, then one of the following is true:*

a. $t_2 = \lambda(x : A_2).t_2$ and $\Gamma, x : A_2 \vdash t_1 \sqsubseteq t_2$ and $A_1 \sqsubseteq A_2$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xvii. *If $t_1 = t'_1\ t'_2$, then one of the following is true:*

a. $t_2 = t'_3\ t'_4$, $\Gamma \vdash t_3 \sqsubseteq t'_3$, and $\Gamma \vdash t_4 \sqsubseteq t'_4$
b. $t'_1 = \mathsf{unbox}_A$ and $t_2 = t'_2$
c. $t'_1 = \mathsf{split}_K$ and $t_2 = t'_2$
d. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
e. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xviii. *If $t_1 = \mathsf{unbox}_A\ t'_1$, then one of the following is true:*

a. $t_2 = t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : ?$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xix. *If $t_1 = \mathsf{split}_K\ t'_1$, then one of the following is true:*

a. $t_2 = t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : K$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xx. *If $t_1 = \Lambda(X <: A).t'_1$, then one of the following is true:*

a. $t_2 = \Lambda(X <: A).t'_2$ and $\Gamma, X <: A_2 \vdash t'_1 \sqsubseteq t'_2$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xxi. *If $t_1 = [A_1]t'_1$, then one of the following is true:*

a. $t_2 = [A_2]t'_2$, $\Gamma \vdash t'_1 \sqsubseteq t'_2$, and $A_1 \sqsubseteq A_2$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

xxii. *If $t_1 = \mathsf{error}_{A_1}$, then one of the following is true:*

a. $\Gamma \vdash_{\mathsf{CG}} t_2 : A_2$ and $A_1 \sqsubseteq A_2$
b. $t_2 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
c. $t_2 = \mathsf{squash}_K\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K$

Proof. The proof of this result holds by straightforward induction on $\Gamma \vdash t_1 \sqsubseteq t_2$.

B Proofs

B.1 Proof of Left-to-Right Consistent Subtyping (Lemma 3)

This is a proof by induction on $\Gamma \vdash A \lesssim B$. We only show a few of the most interesting cases.

Case.

$$\frac{\Gamma \vdash A \lesssim \mathbb{S}}{\Gamma \vdash A \lesssim ?}\ \mathsf{box}$$

In this case $B = ?$.
Part i. Choose $A' = ?$.
Part ii. Choose $B' = A$.
Case.

$$\frac{\Gamma \vdash B \lesssim \mathbb{S}}{\Gamma \vdash ? \lesssim B}\ \mathsf{unbox}$$

In this case $A = ?$.
Part i. Choose $A' = B$.
Part ii. Choose $B' = ?$.
Case.

$$\frac{\Gamma \vdash A_2 \lesssim A_1 \quad \Gamma \vdash B_1 \lesssim B_2}{\Gamma \vdash (A_1 \to B_1) \lesssim (A_2 \to B_2)} \to$$

In this case $A = A_1 \to B_1$ and $B = A_2 \to B_2$.
Part i. By part two of the induction hypothesis we know that $\Gamma \vdash A_1' \sim A_1$ and $\Gamma \vdash A_2 <: A_1'$, and by part one of the induction hypothesis $\Gamma \vdash B_1 \sim B_1'$ and $\Gamma \vdash B_1' <: B_2$. By symmetry of type consistency we may conclude that $\Gamma \vdash A_1 \sim A_1'$ which along with $\Gamma \vdash B_1 \sim B_1'$ implies that $\Gamma \vdash (A_1 \to B_1) \sim (A_1' \to B_1')$, and by reapplying the rule we may conclude that $\Gamma \vdash (A_1' \to B_1') <: (A_2 \to B_2)$.
Part ii. Similar to part one, except that we first applying part one of the induction hypothesis to the first premise, and then the second part to the second premise.

B.2 Proof of Congruence of Type Consistency Along Type Precision (Lemma 28)

The proofs of both parts are similar, and so we only show a few cases of the first part, but the omitted cases follow similarly.
Proof of part one. This is a proof by induction on the form of $A_1 \sqsubseteq A_1'$.

Case.

$$\frac{\Gamma \vdash A_1 \lesssim \mathbb{S}}{A_1 \sqsubseteq ?}\ ?$$

In this case $A_1' = ?$. Suppose $\Gamma \vdash A_1 \sim A_2$. Then it suffices to show that $\Gamma \vdash ? \sim A_2$, and hence, we must show that $\Gamma \vdash A_2 \lesssim \mathbb{S}$, but this follows by Lemma 26.
Case.

$$\frac{A \sqsubseteq C \quad B \sqsubseteq D}{(A \to B) \sqsubseteq (C \to D)} \to$$

In this case $A_1 = A \to B$ and $A_1' = C \to D$. Suppose $\Gamma \vdash A_1 \sim A_2$. Then by inversion for type consistency it must be the case that either $A_2 = ?$ and $\Gamma \vdash A_1 \lesssim \mathbb{S}$, or $A_2 = A' \to B'$, $\Gamma \vdash A \sim A'$, and $\Gamma \vdash B \sim B'$.
Consider the former. Then it suffices to show that $\Gamma \vdash A_1' \sim ?$, and hence we must show that $\Gamma \vdash A_1' \lesssim \mathbb{S}$, but this follows from Lemma 27.
Consider the case when $A_2 = A' \to B'$, $\Gamma \vdash A \sim A'$, and $\Gamma \vdash B \sim B'$. It suffices to show that $\Gamma \vdash (C \to D) \sim (A' \to B')$ which follows from $\Gamma \vdash A' \sim C$ and $\Gamma \vdash D \sim B'$. Thus, it suffices to show that latter. By assumption we know the following:

$$\begin{array}{l} A \sqsubseteq C \text{ and } \Gamma \vdash A \sim A' \\ B \sqsubseteq D \text{ and } \Gamma \vdash B \sim B' \end{array}$$

Now by two applications of the induction hypothesis we obtain $\Gamma \vdash C \sim A'$ and $\Gamma \vdash D \sim B'$. By symmetry the former implies $\Gamma \vdash A \sim C$ and we obtain our result.

B.3 Proof of Congruence of Subtyping Along Type Precision (Lemma 29)

This is a proof by induction on the form of $A \sqsubseteq B$. The proof of part two follows similarly to part one. We only give the most interesting cases. All others follow similarly.
Proof of part one. We only show the most interesting case, because all others are similar.

Case.

$$\frac{A_1 \sqsubseteq A_2 \quad B_1 \sqsubseteq B_2}{(A_1 \to B_1) \sqsubseteq (A_2 \to B_2)} \to$$

In this case $A = A_1 \to B_1$ and $B = A_2 \to B_2$. Suppose $\Gamma \vdash A \lesssim C$. Thus, by inversion for consistency subtyping it must be the case that $C = \top$ and $\Gamma \vdash A : \star$, $C = ?$ and $\Gamma \vdash A \lesssim \mathbb{S}$, or $C = A'_1 \to B'_1$, $\Gamma \vdash A'_1 \lesssim A_1$, and $\Gamma \vdash B_1 \lesssim B'_1$. The case when $C = \top$ is trivial, and the case when $C = ?$ is similarly to the proof of Lemma 28.

Consider the case when $C = A'_1 \to B'_1$, $\Gamma \vdash A'_1 \lesssim A_1$, and $\Gamma \vdash B_1 \lesssim B'_1$. By assumption we know the following:

$$\begin{array}{l} A_1 \sqsubseteq A_2 \text{ and } \Gamma \vdash A'_1 \lesssim A_1 \\ B_1 \sqsubseteq B_2 \text{ and } \Gamma \vdash B_1 \lesssim B'_1 \end{array}$$

So by part two and one, respectively, of the induction hypothesis we know that $\Gamma \vdash A'_1 \lesssim A_2$ and $\Gamma \vdash B_2 \lesssim B'_1$. Thus, by reapplying the rule above we may now conclude that $\Gamma \vdash (A_2 \to B_2) \lesssim (A'_1 \to B'_2)$ to obtain our result.

B.4 Proof of Gradual Guarantee Part One (Lemma 4)

This is a proof by induction on $\Gamma \vdash_{\mathsf{SG}} t : A$. We only show the most interesting cases, because the others follow similarly.

Case.

$$\frac{x : A \in \Gamma \quad \Gamma\,\mathrm{Ok}}{\Gamma \vdash_{\mathsf{SG}} x : A}\ \text{VAR}$$

In this case $t = x$. Suppose $t \sqsubseteq t'$. Then it must be the case that $t' = x$. If $x : A \in \Gamma$, then there is a type A' such that $x : A' \in \Gamma'$ and $A \sqsubseteq A'$. Thus, choose $B = A'$ and the result follows.

Case.

$$\frac{\Gamma \vdash_{\mathsf{SG}} t_1 : A' \quad \mathsf{nat}(A') = \mathsf{Nat}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{succ}\, t_1 : \mathsf{Nat}}\ \mathsf{succ}$$

In this case $A = \mathsf{Nat}$ and $t = \mathsf{succ}\, t_1$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. Then by definition it must be the case that $t' = \mathsf{succ}\, t_2$ where $t_1 \sqsubseteq t_2$. By the induction hypothesis $\Gamma' \vdash_{\mathsf{SG}} t_2 : B'$ where $A' \sqsubseteq B'$. Since $\mathsf{nat}(A') = \mathsf{Nat}$ and $A' \sqsubseteq B'$, then it must be the case that $\mathsf{nat}(B') = \mathsf{Nat}$ by Lemma 22. At this point we obtain our result by choosing $B = \mathsf{Nat}$, and reapplying the rule above.

Case.

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathsf{SG}} t_1 : C \quad \mathsf{nat}(C) = \mathsf{Nat} \quad \Gamma \vdash A_1 \sim A \\ \Gamma \vdash_{\mathsf{SG}} t_2 : A_1 \quad \Gamma, x : \mathsf{Nat} \vdash_{\mathsf{SG}} t_3 : A_2 \quad \Gamma \vdash A_2 \sim A\end{array}}{\Gamma \vdash_{\mathsf{SG}} \mathsf{case}\, t_1\, \mathsf{of}\, 0 \to t_2, (\mathsf{succ}\, x) \to t_3 : A}\ \mathsf{Nat}_e$$

In this case $t = \mathsf{case}\, t_1\, \mathsf{of}\, 0 \to t_2, (\mathsf{succ}\, x) \to t_3$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. This implies that $t' = \mathsf{case}\, t'_1\, \mathsf{of}\, 0 \to t'_2, (\mathsf{succ}\, x) \to t'_3$ such that $t_1 \sqsubseteq t'_1$, $t_2 \sqsubseteq t'_2$, and $t_3 \sqsubseteq t'_3$. Since $\Gamma \sqsubseteq \Gamma'$ then $(\Gamma, x : \mathsf{Nat}) \sqsubseteq (\Gamma', x : \mathsf{Nat})$. By the induction hypothesis we know the following:

$$\begin{array}{l} \Gamma' \vdash_{\mathsf{SG}} t'_1 : C' \text{ for } C \sqsubseteq C' \\ \Gamma' \vdash_{\mathsf{SG}} t_2 : A'_1 \text{ for } A_1 \sqsubseteq A'_1 \\ \Gamma', x : \mathsf{Nat} \vdash_{\mathsf{SG}} t_3 : A'_2 \text{ for } A_2 \sqsubseteq A'_2 \end{array}$$

By assumption we know that $\Gamma \vdash A_1 \sim A$, $\Gamma \vdash A_2 \sim A$, and $\Gamma \sqsubseteq \Gamma'$, hence, by Lemma 24 we know $\Gamma' \vdash A_1 \sim A$ and $\Gamma' \vdash A_2 \sim A$. By the induction hypothesis we know that $A_1 \sqsubseteq A'_1$ and $A_2 \sqsubseteq A'_2$, so by using Lemma 23 we may obtain that $\Gamma' \vdash A'_1 \sim A$ and $\Gamma' \vdash A'_2 \sim A$. At this point choose $B = A$ and we obtain our result by reapplying the rule.
Case.

$$\frac{\Gamma \vdash_{\mathsf{SG}} t_1 : A_1 \quad \Gamma \vdash_{\mathsf{SG}} t_2 : A_2 \quad \mathsf{list}(A_2) = \mathsf{List}\, A_3 \quad \Gamma \vdash A_1 \sim A_3}{\Gamma \vdash_{\mathsf{SG}} t_1 :: t_2 : \mathsf{List}\, A_3}\ \mathsf{List}_i$$

In this case $A = \mathsf{List}\, A_3$ and $t = t_1 :: t_2$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. Then it must be the case that $t' = t'_1 :: t'_2$ where $t_1 \sqsubseteq t'_1$ and $t_2 \sqsubseteq t'_2$. Then by the induction hypothesis we know the following:

$$\begin{array}{l} \Gamma' \vdash_{\mathsf{SG}} t'_1 : A'_1 \text{ where } A_1 \sqsubseteq A'_1 \\ \Gamma' \vdash_{\mathsf{SG}} t'_2 : A'_2 \text{ where } A_2 \sqsubseteq A'_2 \end{array}$$

By Lemma 22 $\mathsf{list}(A'_2) = \mathsf{List}\, A'_3$ where $A_3 \sqsubseteq A'_3$. Now by Lemma 24 and Lemma 23 we know that $\Gamma' \vdash A'_1 \sim A_3$, and by using the same lemma again, $\Gamma' \vdash A'_1 \sim A'_3$ because $\Gamma' \vdash A_3 \sim A'_1$ holds by symmetry. Choose $B = \mathsf{List}\, A'_3$ and the result follows.
Case.

$$\frac{\Gamma \vdash_{\mathsf{SG}} t_1 : A_1 \quad \Gamma \vdash_{\mathsf{SG}} t_2 : A_2}{\Gamma \vdash_{\mathsf{SG}} (t_1, t_2) : A_1 \times A_2}\ \times_i$$

In this case $A = A_1 \times A_2$ and $t = (t_1, t_2)$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. This implies that $t' = (t'_1, t'_2)$ where $t_1 \sqsubseteq t'_1$ and $t_2 \sqsubseteq t'_2$.
By the induction hypothesis we know:

$$\begin{array}{l} \Gamma' \vdash_{\mathsf{SG}} t'_1 : A'_1 \text{ and } A_1 \sqsubseteq A'_1 \\ \Gamma' \vdash_{\mathsf{SG}} t'_2 : A'_2 \text{ and } A_2 \sqsubseteq A'_2 \end{array}$$

Then choose $B = A'_1 \times A'_2$ and the result follows by reapplying the rule above and the fact that $(A_1 \times A_2) \sqsubseteq (A'_1 \times A'_2)$.
Case.

$$\frac{\Gamma, x : A_1 \vdash_{\mathsf{SG}} t_1 : B_1}{\Gamma \vdash_{\mathsf{SG}} \lambda(x : A_1).t_1 : A_1 \to B_1}\ \to_i$$

In this case $A_1 \rightarrow B_2$ and $t = \lambda(x : A_1).t_1$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. Then it must be the case that $t' = \lambda(x : A_2).t_2$, $t_1 \sqsubseteq t_2$, and $A_1 \sqsubseteq A_2$. Since $\Gamma \sqsubseteq \Gamma'$ and $A_1 \sqsubseteq A_2$, then $(\Gamma, x : A_1) \sqsubseteq (\Gamma', x : A_2)$ by definition. Thus, by the induction hypothesis we know the following:

$$\Gamma', x : A_2 \vdash_{\mathsf{SG}} t'_1 : B_2 \text{ and } B_1 \sqsubseteq B_2$$

Choose $B = A_2 \rightarrow B_2$ and the result follows by reapplying the rule above and the fact that $(A_1 \rightarrow B_1) \sqsubseteq (A_2 \rightarrow B_2)$.
Case.

$$\frac{\Gamma \vdash_{\mathsf{SG}} t_1 : \forall(X <: C_0).C_2 \quad \Gamma \vdash C_1 \lesssim C_0}{\Gamma \vdash_{\mathsf{SG}} [C_1]t_1 : [C_1/X]C_2}\ \forall_e$$

In this case $t = [C_1]t_1$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. Then it must be the case that $t' = [C'_1]t_2$ such that $t_1 \sqsubseteq t_2$ and $C_1 \sqsubseteq C'_1$. By the induction hypothesis:

$$\Gamma' \vdash_{\mathsf{SG}} t_2 : C \text{ where } \forall(X <: C_0).C_2 \sqsubseteq C$$

Thus, it must be the case that $C = \forall(X <: C_0).C'_2$ such that $C_2 \sqsubseteq C'_2$. By assumption we know that $\Gamma \vdash C_1 \lesssim C_0$ and $C_1 \sqsubseteq C'_1$, and thus, by Corollary 4 and Lemma 25 we know $\Gamma' \vdash C'_1 \lesssim C_0$. Thus, choose $B = C$, and the result follows by reapplying the rule above, and the fact that $A \sqsubseteq C$, because $C_2 \sqsubseteq C'_2$.
Case.

$$\frac{\Gamma \vdash_{\mathsf{SG}} t : A' \quad \Gamma \vdash A' \lesssim A}{\Gamma \vdash_{\mathsf{SG}} t : A}\ \textsc{sub}$$

Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. By the induction hypothesis we know that $\Gamma' \vdash_{\mathsf{SG}} t' : A''$ for $A' \sqsubseteq A''$. We know $A'' \sqsubseteq A$ or $A \sqsubseteq A''$, because we know that $\Gamma \vdash A' \lesssim A$ and $A' \sqsubseteq A''$. Suppose $A'' \sqsubseteq A$, then by Corollary 2 $\Gamma' \vdash A'' \lesssim A$, and then by subsumption $\Gamma' \vdash_{\mathsf{SG}} t' : A$, hence, choose $B = A$ and the result follows. If $A \sqsubseteq A''$, then choose $B = A''$ and the result follows.
Case.

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathsf{SG}} t_1 : C \quad \mathsf{fun}(C) = A_1 \rightarrow B_1 \\ \Gamma \vdash_{\mathsf{SG}} t_2 : A_2 \quad \Gamma \vdash A_2 \sim A_1\end{array}}{\Gamma \vdash_{\mathsf{SG}} t_1\, t_2 : B_1}\ \rightarrow_e$$

In this case $A = B_1$ and $t = t_1\, t_2$. Suppose $t \sqsubseteq t'$ and $\Gamma \sqsubseteq \Gamma'$. The former implies that $t' = t'_1\, t'_2$ such that $t_1 \sqsubseteq t'_1$ and $t_2 \sqsubseteq t'_2$. By the induction hypothesis we know the following:

$$\begin{array}{c}\Gamma' \vdash_{\mathsf{SG}} t'_1 : C' \text{ for } C \sqsubseteq C' \\ \Gamma' \vdash_{\mathsf{SG}} t'_2 : A'_2 \text{ for } A_2 \sqsubseteq A'_2\end{array}$$

We know by assumption that $\Gamma \vdash A_2 \sim A_1$ and hence $\Gamma' \vdash A_2 \sim A_1$ because bounds on type variables are left unchanged by context precision. Since $C \sqsubseteq C'$ and $\mathsf{fun}(C) = A_1 \to B_1$, then $\mathsf{fun}(C') = A'_1 \to B'_1$ where $A_1 \sqsubseteq A'_1$ and $B_1 \sqsubseteq B'_1$ by Lemma 22. Furthermore, we know $\Gamma' \vdash A_2 \sim A_1$ and $A_2 \sqsubseteq A'_2$ and $A_1 \sqsubseteq A'_1$, then we know $\Gamma' \vdash A'_2 \sim A'_1$ by Corollary 3. So choose $B = B'_1$. Then reapply the rule above and the result follows, because $B_1 \sqsubseteq B'_1$.

B.5 Proof of Type Preservation for Cast Insertion (Lemma 5)

The cast insertion algorithm is type directed and with respect to every term t_1 it will produce a term t_2 of the core language with the type A – this is straightforward to show by induction on the form of $\Gamma \vdash_{\mathsf{SG}} t_1 : A$ making use of typing for casting morphisms Lemma 30 – except in the case of type application. We only consider this case here.

This is a proof by induction on the form of $\Gamma \vdash_{\mathsf{SG}} t_1 : A$. Suppose the form of $\Gamma \vdash_{\mathsf{SG}} t_1 : A$ is as follows:

$$\frac{\Gamma \vdash_{\mathsf{SG}} t'_1 : \forall(X <: B_1).B_2 \quad \Gamma \vdash A_1 \lesssim B_1}{\Gamma \vdash_{\mathsf{SG}} [A_1]t'_1 : [A_1/X]B_2}\ \forall_e$$

In this case $t_1 = [A_1]t'_1$ and $A = [A_1/X]B_2$. Cast insertion is syntax directed, and hence, inversion for it holds trivially. Thus, it must be the case that the form of $\Gamma \vdash t_1 \Rightarrow t_2 : B$ is as follows:

$$\frac{\Gamma \vdash t'_1 \Rightarrow t'_2 : \forall(X <: B_1).B'_2 \quad \Gamma \vdash A_1 \sim A_2 \quad \Gamma \vdash A_2 <: B_1}{\Gamma \vdash ([A_1]t'_1) \Rightarrow ([A_2]t'_2) : [A_2/X]B'_2}$$

So $t_2 = [A_2]t'_2$ and $B = [A_2/X]B'_2$. Since we know $\Gamma \vdash_{\mathsf{SG}} t'_1 : \forall(X <: B_1).B_2$ and $\Gamma \vdash t'_1 \Rightarrow t'_2 : \forall(X <: B_1).B'_2$ we can apply the induction hypothesis to obtain $\Gamma \vdash_{\mathsf{CG}} t'_2 : \forall(X <: B_1).B'_2$ and $\Gamma \vdash (\forall(X <: B_1).B_2) \sim (\forall(X <: B_1).B'_2)$, and thus, $\Gamma, X <: B_1 \vdash B_2 \sim B'_2$ by inversion for type consistency. If $\Gamma, X <: B_1 \vdash B_2 \sim B'_2$ holds, then $\Gamma \vdash [A_1/X]B_2 \sim [A_2/X]B'_2$ when $\Gamma \vdash A_1 \sim A_2$ by substitution for type consistency (Lemma 33). Since we know $\Gamma \vdash_{\mathsf{CG}} t'_2 : \forall(X <: B_1).B'_2$ by the induction hypothesis and $\Gamma \vdash A_2 <: B_1$ by assumption, then we know $\Gamma \vdash_{\mathsf{CG}} [A_2]t'_2 : [A_2/X]B'_2$ by applying the Core Grady typing rule $\forall_e$.

B.6 Proof of Simulation of More Precise Programs (Lemma 7)

This is a proof by induction on $\Gamma \vdash_{\mathsf{CG}} t_1 : A_1$. We only give the most interesting cases. All others follow similarly. Throughout the proof we implicitly make use of typability inversion (Lemma 36) when applying the induction hypothesis.

Case.

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : \mathsf{Nat}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{succ}\, t : \mathsf{Nat}}\ \mathsf{succ}$$

In this case $t_1 = \mathsf{succ}\, t$ and $A = \mathsf{Nat}$. Suppose $\Gamma \vdash_{\mathsf{CG}} t_1' : A'$. By inversion for term precision we must consider the following cases:
i. $t_1' = \mathsf{succ}\, t'$ and $\Gamma \vdash t \sqsubseteq t'$
ii. $t_1' = \mathsf{box}_{\mathsf{Nat}}\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : \mathsf{Nat}$

Proof of part i. Suppose $t_1' = \mathsf{succ}\, t'$, $\Gamma \vdash t \sqsubseteq t'$, and $t_1 \rightsquigarrow t_2$. Then $t_2 = \mathsf{succ}\, t''$ and $t \rightsquigarrow t''$. Then by the induction hypothesis we know that there is some t''' such that $t' \rightsquigarrow^* t'''$ and $\Gamma \vdash t'' \sqsubseteq t'''$. Choose $t_2' = \mathsf{succ}\, t'''$ and the result follows.

Proof of part ii. Suppose $t_1' = \mathsf{box}_{\mathsf{Nat}}\, t_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : \mathsf{Nat}$, and $t_1 \rightsquigarrow t_2$. Then choose $t_2' = \mathsf{box}_{\mathsf{Nat}}\, t_2$, and the result follows, because we know by type preservation that $\Gamma \vdash_{\mathsf{CG}} t_2 : \mathsf{Nat}$, and hence, $\Gamma \vdash t_2 \sqsubseteq t_2'$.
Case.

$$\frac{\begin{array}{l}\Gamma \vdash_{\mathsf{CG}} t : \mathsf{Nat} \\ \Gamma \vdash_{\mathsf{CG}} t_3 : A \quad \Gamma, x : \mathsf{Nat} \vdash_{\mathsf{CG}} t_4 : A\end{array}}{\Gamma \vdash_{\mathsf{CG}} \mathsf{case}\, t \colon \mathsf{Nat}\, \mathsf{of}\, 0 \to t_3, (\mathsf{succ}\, x) \to t_4 : A}\ \mathsf{Nat}_e$$

In this case $t_1 = \mathsf{case}\, t \colon \mathsf{Nat}\, \mathsf{of}\, 0 \to t_3, (\mathsf{succ}\, x) \to t_4$. Suppose $\Gamma \vdash_{\mathsf{CG}} t_1' : A'$. Then inversion of term precision implies that one of the following must hold:

- $t_1' = \mathsf{case}\, t' \colon \mathsf{Nat}\, \mathsf{of}\, 0 \to t_3', (\mathsf{succ}\, x) \to t_4'$, $\Gamma \vdash t \sqsubseteq t'$, $\Gamma \vdash t_3 \sqsubseteq t_3'$, and $\Gamma, x : \mathsf{Nat} \vdash t_4 \sqsubseteq t_4'$
- $t_1' = \mathsf{box}_A\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
- $t_1' = \mathsf{squash}_K\, t_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : K$, and $A = K$

Proof of part i. Suppose $t_1' = \mathsf{case}\, t' \colon \mathsf{Nat}\, \mathsf{of}\, 0 \to t_3', (\mathsf{succ}\, x) \to t_4'$, $\Gamma \vdash t \sqsubseteq t'$, $\Gamma \vdash t_3 \sqsubseteq t_3'$, and $\Gamma, x : \mathsf{Nat} \vdash t_4 \sqsubseteq t_4'$.

We case split over $t_1 \rightsquigarrow t_2$.

Case. Suppose $t = 0$ and $t_2 = t_3$. Since $\Gamma \vdash t_1 \sqsubseteq t_1'$ we know that it must be the case that $t' = 0$ and $t_1' \rightsquigarrow t_3'$ by inversion for term precision or t_1' would not be typable which is a contradiction. Thus, choose $t_2' = t_3'$ and the result follows.

Case. Suppose $t = \mathsf{succ}\, t''$ and $t_2 = [t''/x]t_4$. Since $\Gamma \vdash t_1 \sqsubseteq t_1'$ we know that $t' = \mathsf{succ}\, t'''$, or t_1' would not be typable, and $\Gamma \vdash t'' \sqsubseteq t'''$ by inversion for term precision. In addition, $t_1' \rightsquigarrow [t'''/x]t_4'$. Choose $t_2 = [t'''/x]t_4'$. Then it suffices to show that $\Gamma \vdash [t''/x]t_4 \sqsubseteq [t'''/x]t_4'$ by substitution for term precision (Lemma 35).

Case. Suppose a congruence rule was used. Then $t_2 = \mathsf{case}\, t'' \colon \mathsf{Nat}\, \mathsf{of}\, 0 \to t_3'', (\mathsf{succ}\, x) \to t_4''$. This case will follow straightforwardly by induction and a case split over which congruence rule was used.

Proof of part ii. Suppose $t_1' = \mathsf{box}_A\, t_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : A$, and $t_1 \rightsquigarrow t_2$. Then choose $t_2' = \mathsf{box}_A\, t_2$, and the result follows, because we know by type preservation that $\Gamma \vdash_{\mathsf{CG}} t_2 : A$, and hence, $\Gamma \vdash t_2 \sqsubseteq t_2'$.

Proof of part iii. Similar to the previous case.
Case.

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : A \times B}{\Gamma \vdash_{\mathsf{CG}} \mathsf{fst}\, t : A} \times_{e_1}$$

In this case $t_1 = \mathsf{fst}\, t$. Suppose $\Gamma \vdash t_1 \sqsubseteq t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : A'$. Then inversion for term precision implies that one of the following must hold:

- $t'_1 = \mathsf{fst}\, t'$ and $\Gamma \vdash t \sqsubseteq t'$
- $t'_1 = \mathsf{box}_A\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
- $t'_1 = \mathsf{squash}_K\, t_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : K$, and $A = K$

We only consider the proof of part i, because the others follow similarly to the previous case. Case split over $t_1 \rightsquigarrow t_2$.

Case. Suppose $t = (t'_3, t''_3)$ and $t_2 = t'_3$. By inversion for term precision it must be the case that $t' = (t'_4, t''_4)$ because $\Gamma \vdash t_1 \sqsubseteq t'_1$ or else t'_1 would not be typable. In addition, this implies that $\Gamma \vdash t'_3 \sqsubseteq t'_4$ and $\Gamma \vdash t''_3 \sqsubseteq t''_4$. Thus, $t'_1 \rightsquigarrow t'_4$. Thus, choose $t'_2 = t'_4$ and the result follows.
Case. Suppose a congruence rule was used. Then $t_2 = \mathsf{fst}\, t''$. This case will follow straightforwardly by induction and a case split over which congruence rule was used.

Case.

$$\frac{\Gamma, x : A_1 \vdash_{\mathsf{CG}} t : A_2}{\Gamma \vdash_{\mathsf{CG}} \lambda(x : A_1).t : A_1 \to A_2} \to_i$$

In this case $t_1 = \lambda(x : A_1).t$ and $A = A_1 \to A_2$. Suppose $\Gamma \vdash t_1 \sqsubseteq t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : A'$. Then inversion of term precision implies that one of the following must hold:

- $t'_1 = \lambda(x : A'_1).t'$
- $t'_1 = \mathsf{box}_A\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
- $t'_1 = \mathsf{squash}_K\, t_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : K$, and $A = K$

We only consider the proof of part i. The reduction relation does not reduce under λ-expressions. Hence, $t_2 = t_1$, and thus, choose $t'_2 = t'_1$, and the case trivially follows.
Case.

$$\frac{\Gamma \vdash_{\mathsf{CG}} t_3 : A_1 \to A_2 \quad \Gamma \vdash_{\mathsf{CG}} t_4 : A_1}{\Gamma \vdash_{\mathsf{CG}} t_3\, t_4 : A_2} \to_e$$

In this case $t_1 = t_3\, t_4$. Suppose $\Gamma \vdash t_1 \sqsubseteq t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : A'$. Then by inversion for term prevision we know one of the following is true:

i. $t'_1 = t'_3\, t'_4$, $\Gamma \vdash t_3 \sqsubseteq t'_3$, and $\Gamma \vdash t_4 \sqsubseteq t'_4$
ii. $t'_1 = \mathsf{box}_{A_2}\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
iii. $t_3 = \mathsf{unbox}_{A_2}$, $t'_1 = t_4$, and $\Gamma \vdash_{\mathsf{CG}} t_4 : ?$

iv. $t_3 = \mathsf{split}_{K_2}$, $t'_1 = t_4$, and $\Gamma \vdash_{\mathsf{CG}} t_4 : ?$
v. $t'_1 = \mathsf{squash}_{K_2}\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : K_2$

Proof of part i. Suppose $t'_1 = t'_3\, t'_4$, $\Gamma \vdash t_3 \sqsubseteq t'_3$, and $\Gamma \vdash t_4 \sqsubseteq t'_4$. We case split on the from of $t_1 \rightsquigarrow t_2$.

Case. Suppose $t_3 = \lambda(x : A_1).t_5$ and $t_2 = [t_4/x]t_5$. Then by inversion for term precision we know that $t'_3 = \lambda(x : A'_1).t'_5$ and $\Gamma, x : A'_2 \vdash t_5 \sqsubseteq t'_5$, because $\Gamma \vdash t_3 \sqsubseteq t'_3$ and the requirement that t'_1 is typable. Choose $t'_2 = [t'_4/x]t'_5$ and it is easy to see that $t'_1 \rightsquigarrow [t'_4/x]t'_4$. We know that $\Gamma, x : A'_2 \vdash t_5 \sqsubseteq t'_5$ and $\Gamma \vdash t_4 \sqsubseteq t'_4$, and hence, by Lemma 35 we know that $\Gamma \vdash [t_4/x]t_5 \sqsubseteq [t'_4/x]t'_5$, and we obtain our result.

Case. Suppose $t_3 = \mathsf{unbox}_A$, $t_4 = \mathsf{box}_A\, t_5$, and $t_2 = t_5$. Then by inversion for term prevision $t'_3 = \mathsf{unbox}_A$, $t'_4 = \mathsf{box}_A\, t'_5$, and $\Gamma \vdash t_5 \sqsubseteq t'_5$. Note that $t'_4 = \mathsf{box}_A\, t'_5$ and $\Gamma \vdash t_5 \sqsubseteq t'_5$ hold even though there are two potential rules that could have been used to construct $\Gamma \vdash t_4 \sqsubseteq t'_4$. Choose $t'_2 = t'_5$ and it is easy to see that $t'_1 \rightsquigarrow t'_5$. Thus, we obtain our result.

Case. Suppose $t_3 = \mathsf{unbox}_A$, $t_4 = \mathsf{box}_B\, t_5$, $A \neq B$, and $t_2 = \mathsf{error}_B$. Then $t'_3 = \mathsf{unbox}_A$ and $t'_4 = \mathsf{box}_B\, t'_5$. Choose $t'_2 = \mathsf{error}_B$ and it is easy to see that $t'_1 \rightsquigarrow t'_5$. Finally, we can see that $\Gamma \vdash t_2 \sqsubseteq t'_2$ by reflexivity.

Case. Suppose $t_3 = \mathsf{split}_U$, $t_4 = \mathsf{squash}_U\, t_5$, and $t_2 = t_5$. Similar to the case for boxing and unboxing.

Case. Suppose $t_3 = \mathsf{split}_{U_1}$, $t_4 = \mathsf{squash}_{U_2}\, t_5$, $U_1 \neq U_2$, and $t_2 = t_5$. Similar to the case for boxing and unboxing.

Case. Suppose a congruence rule was used. Then $t_2 = t'_5\, t'_6$. This case will follow straightforwardly by induction and a case split over which congruence rule was used.

Proof of part ii. We know that $t_1 = t_3\, t_4$. Suppose $t'_1 = \mathsf{box}_{A_2}\, t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$. If $t_1 \rightsquigarrow t_2$, then $t'_1 = (\mathsf{box}_{A_2}\, t_1) \rightsquigarrow (\mathsf{box}_{A_2}\, t_2)$. Thus, choose $t'_2 = \mathsf{box}_{A_2}\, t_2$.

Proof of part iii. We know that $t_1 = t_3\, t_4$. Suppose $t_3 = \mathsf{unbox}_{A_2}$, $t'_1 = t_4$, and $\Gamma \vdash_{\mathsf{CG}} t_4 : ?$. Then $t_1 = \mathsf{unbox}_{A_2}\, t_4$. We case split over $t_1 \rightsquigarrow t_2$. We have three cases to consider.

Suppose $t_4 = \mathsf{box}_{A_2}\, t_5$ and $t_2 = t_5$. Then choose $t'_2 = t_4 = t'_1$, and we obtain our result.

Suppose $t_4 = \mathsf{box}_{A_3}\, t_5$, $A_2 \neq A_3$, and $t_2 = \mathsf{error}_{A_2}$. Then choose $t'_2 = t_4 = t'_1$, and we obtain our result.

Suppose a congruence rule was used. Then $t_2 = t_3\, t'_4$. This case will follow straightforwardly by induction.

Proof of part iv. Similar to part iii.

Proof of part v. Similar to part ii.

Case.

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : \forall(X <: A_2).A_3 \quad \Gamma \vdash A_1 <: A_2}{\Gamma \vdash_{\mathsf{CG}} [A_1]t : [A_1/X]A_3}\ \forall_e$$

In this case $t_1 = [A_1]t$ and $A = [A_1/X]A_3$. Suppose $\Gamma \vdash t_1 \sqsubseteq t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : A'$.

- $t'_1 = [A'_1]t'$, $\Gamma \vdash t \sqsubseteq t'$, and $A_1 \sqsubseteq A'_1$
- $t'_1 = \mathsf{box}_A\ t_1$ and $\Gamma \vdash_{\mathsf{CG}} t_1 : A$
- $t'_1 = \mathsf{squash}_K\ t_1$, $\Gamma \vdash_{\mathsf{CG}} t_1 : K$, and $A = K$

We only consider the proof of part i. We case split over the form of $t_1 \rightsquigarrow t_2$.

Case. Suppose $t = \Lambda(X <: A_2).t_3$ and $t_2 = [A_1/X]t_3$. Then inversion for term precision on $\Gamma \vdash t \sqsubseteq t'$ and the fact that $\Gamma \vdash_{\mathsf{CG}} t : \forall(X <: A_2).A_3$ and $t'_1 = [A'_1]t'$ then it can only be the case that $t' = \Lambda(X <: A_2).t'_3$ and $\Gamma, X <: A_2 \vdash t_3 \sqsubseteq t'_3$, or t'_1 would not be typable which is a contradiction. Then by substitution for term precision we know that $\Gamma \vdash [A_1/X]t_3 \sqsubseteq [A'_1/X]t'_3$ by substitution for term precision (Lemma 35), because we know that $A_1 \sqsubseteq A'_1$. Choose $t'_2 = [A'_1/X]t'_3$ and the result follows, because $t'_1 \rightsquigarrow t'_2$.

Case. Suppose a congruence rule was used. Then $t_2 = [A_1]t''$. This case will follow straightforwardly by induction and a case split over which congruence rule was used.

Case.

$$\frac{\Gamma \vdash_{\mathsf{CG}} t : A_1 \quad \Gamma \vdash A_1 <: A_2}{\Gamma \vdash_{\mathsf{CG}} t : A_2}\ \text{SUB}$$

In this case $t_1 = t$ and $A = A_2$. Suppose $\Gamma \vdash t_1 \sqsubseteq t'_1$ and $\Gamma \vdash_{\mathsf{CG}} t'_1 : A'$. Assume $t_1 \rightsquigarrow t_2$. Then by the induction hypothesis there is a t'_2 such that $t'_1 \rightsquigarrow^* t'_2$ and $\Gamma \vdash t_2 \sqsubseteq t'_2$, thus, we obtain our result.