Boxy type inference for higher-rank types and impredicativity
Technical Appendix

University of Pennsylvania Technical Report MS-CIS-05-23

Dimitrios Vytiniotis Stephanie Weirich
University of Pennsylvania
{dimitriv,sweirich}@cis.upenn.edu

Simon Peyton-Jones
Microsoft Research
simonpj@microsoft.com

June 26, 2006

Contents

1 Introduction **2**

2 Definitions and conventions **5**

3 Subsumption and Boxy matching **6**
3.1 Boxy matching . . . 6
3.2 Subsumption . . . 6
3.3 Boxing and unboxing of types . . . 7
3.4 Strange boxing relation . . . 18

4 Translation of System-F **22**

5 Translation to System-F and type safety **29**

6 Weakening lemmas **30**

7 Extension of theorems for full type system **32**
7.1 Type safety of full system . . . 33
7.2 Embedding of System-F . . . 33
7.3 Weakening and substitution lemmas for the full system . . . 34
7.4 Conservative extension of Hindley-Milner . . . 35
7.4.1 Base system conservatively extends HM . . . 35
7.4.2 Full system conservatively extends HM . . . 37

8 A type inference algorithm **38**
8.1 Definitions . . . 38
8.2 Summary and generalised results . . . 39
8.3 Detailed properties . . . 40
8.3.1 Algorithmic version of type system . . . 40
8.3.2 Unification . . . 44
8.3.3 Boxy matching and equivalence . . . 46
8.3.4 Subsumption . . . 48
8.3.5 Main algorithm . . . 50

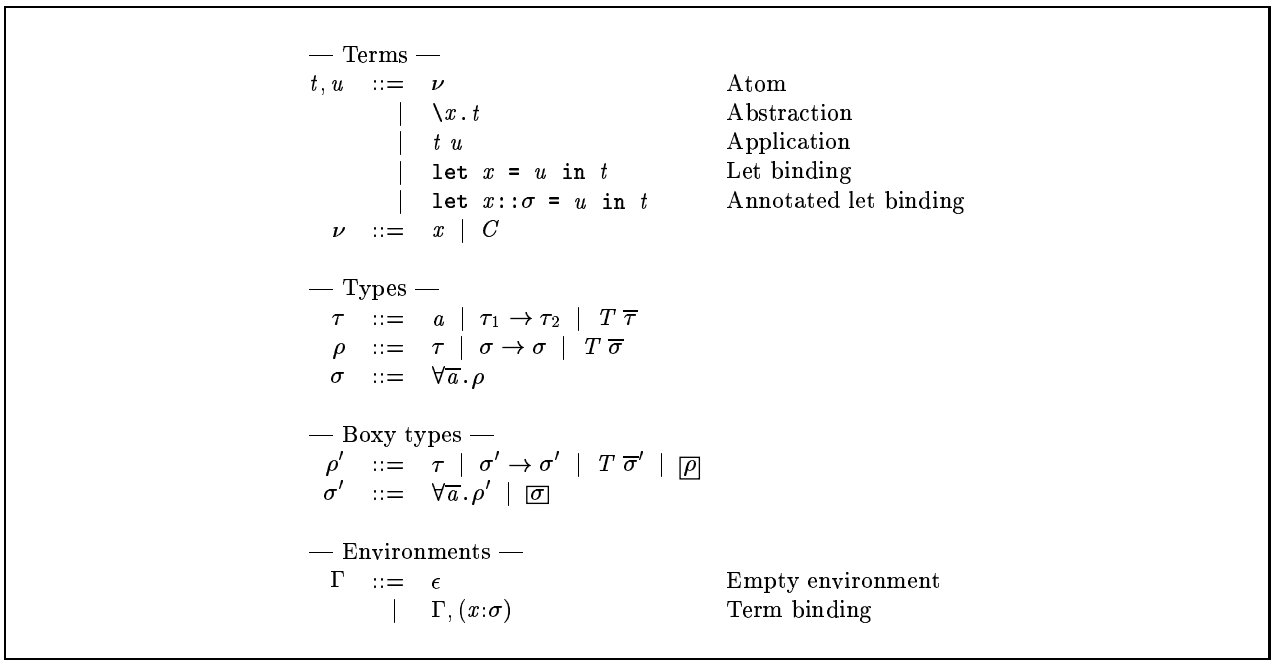

Figure 1: Syntax of the source language and types

1 Introduction

This paper accompanies the main paper "Boxy types: type inference for higher-rank types and impredicativity" [1]. For self-containment we repeat the figures in this document. The basic system syntax is given in Figure 1.

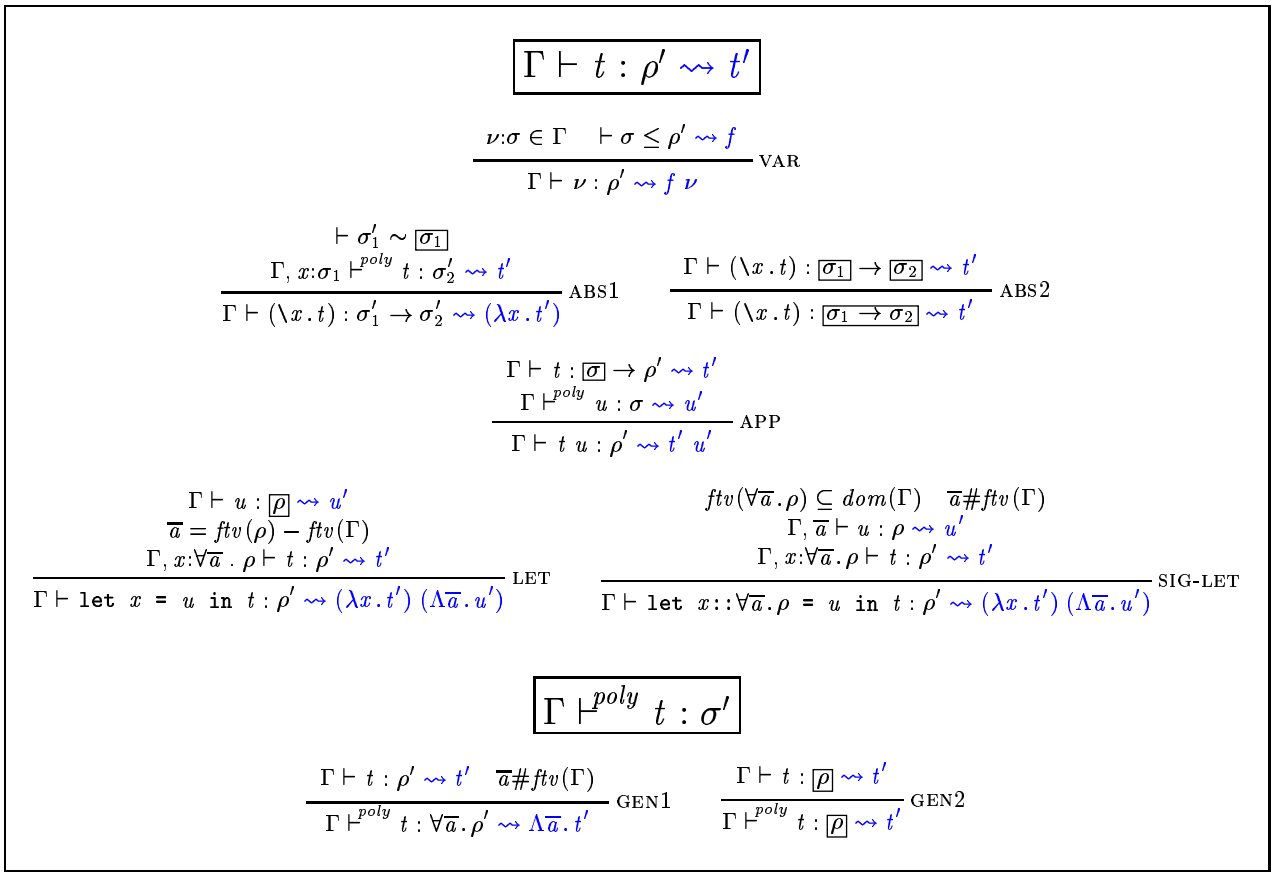

Figure 2: Type system specification

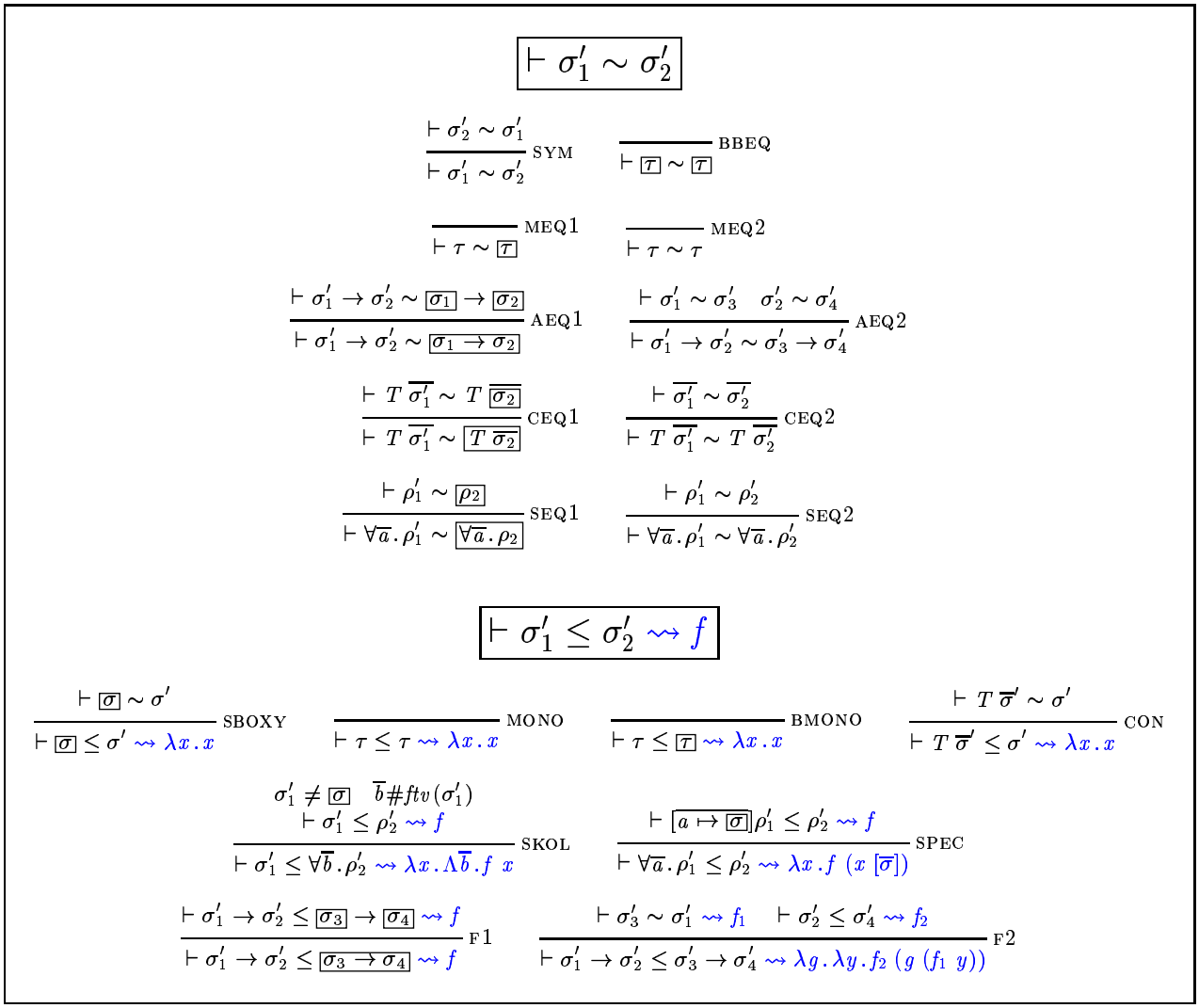

Figure 3: Subsumption and boxy matching

2 Definitions and conventions

Substitutions, denoted with S, T, U, V are, as usual, idempotent maps from (internal) variables to *monotypes*. We use $dom(S)$ and $range(S)$ to denote the domain and the range of a substitution S respectively. Substitutions are naturally extended to be total as follows: We define $S(a) = a$ whenever $a \notin dom(S)$. Sometimes we write $\mathcal{X}_1, \mathcal{X}_2$ to denote the *union* of the two variable sets $\mathcal{X}_1$ and $\mathcal{X}_2$. Composition of substitutions, $S \cdot V$, is defined as usual: $S \cdot V(\sigma) = S(V(\sigma))$.

3 Subsumption and Boxy matching

Subsumption and boxy matching are given in Figure 3.

3.1 Boxy matching

Boxy matching is the relation that is responsible for filling in the holes in the two types that are compared. In an unknown-meets-unknown situation it forces both boxes to be monotypes, otherwise the order of filling in the boxes would be important.

Lemma 3.1 (Substitution for boxy matching). *If* $\vdash \sigma_1' \sim \sigma_2'$ *then* $\vdash S\sigma_1' \sim S\sigma_2'$ *and the new derivation has the same height.*

Proof. Straightforward induction on the height of the derivation $\vdash \sigma_1' \sim \sigma_2'$. The cases where the last rule was SYM, AEQ1, AEQ2, CEQ1, or CEQ2 follow directly by application of the induction hypothesis and application of the same rule. The cases of BBEQ,MEQ1, and MEQ2 follow by the fact that substitutions range over monotypes. For the rest of the cases we have:

- Case SEQ1. We have that $\vdash \forall \overline{a}.\rho_1' \sim \boxed{\forall \overline{a}.\rho_2}$ given that $\vdash \rho_1' \sim \boxed{\rho_2}$. Consider a substitution $S \cdot [\overline{a \mapsto b}]$ where $\overline{b} \# vars(S)$. Then by induction hypothesis $\vdash S[\overline{a \mapsto b}]\rho_1' \sim \boxed{S[\overline{a \mapsto b}]\rho_2}$ and by rule SEQ1 we have that $\vdash \forall \overline{b}.S[\overline{a \mapsto b}]\rho_1' \sim \boxed{\forall \overline{b}.S[\overline{a \mapsto b}]\rho_2}$, or equivalently $\vdash S(\forall \overline{b}.[\overline{a \mapsto b}]\rho_1') \sim S(\boxed{\forall \overline{b}.[\overline{a \mapsto b}]\rho_2})$, or equivalently $\vdash S(\forall \overline{a}.\rho_1') \sim S(\boxed{\forall \overline{a}.\rho_2})$ as required.
- Case SEQ2. Similar to the case for SEQ1.

□

Remark 3.2 (Boxy matching not reflexive). *It is not the case that* $\vdash \boxed{\sigma} \sim \boxed{\sigma}$ *for arbitrary* σ.

Proof. Immediate. □

Remark 3.3 (Boxy matching not transitive). *If* $\vdash \sigma_1' \sim \sigma_2'$ *and* $\vdash \sigma_2' \sim \sigma_3'$ *then it is not necessarily the case that* $\vdash \sigma_1' \sim \sigma_3'$.

Proof. To see why take $\sigma_1' = \boxed{\sigma}$, $\sigma_2' = \sigma$ and $\sigma_3' = \boxed{\sigma}$ where σ is not a monotype. □

3.2 Subsumption

This is a variation of Odersky-Läufer subsumption that is able to handle boxes and uses invariance for function arguments instead of contravariance.

Lemma 3.4 (Substitution for subsumption). *If* $\vdash \sigma_1' \leq \sigma_2'$ *then* $\vdash S\sigma_1' \leq S\sigma_2'$ *and the new derivation has the same height.*

Proof. By induction on the height of the derivation $\vdash \sigma_1' \leq \sigma_2'$. We proceed by case analysis on the last rule used. The case of SBOXY follows from Lemma 3.1. The cases for MONO and BMON follow from the fact that substitutions range over monotypes. The case for CON follows from Lemma 3.1. The case for F1 follows by induction hypothesis and application of F1, and F2 follows by induction hypothesis, Lemma 3.1, and application of F2. For the rest of the cases we have:

- Case SKOL. In this case $\sigma_1' \leq \forall \overline{b}.\rho_2'$ given that

$$\overline{b} \# ftv(\sigma_1') \tag{1}$$
$$\vdash \sigma_1' \leq \rho_2' \tag{2}$$

  Consider $S \cdot [\overline{b \mapsto c}]$ where $\overline{c} \# vars(S), ftv(\sigma_1')$. Then, by induction hypothesis for (2) we get $\vdash S[\overline{b \mapsto c}]\sigma_1' \leq S[\overline{b \mapsto c}]\rho_2'$. Using (1) this is equivalent to $\vdash S\sigma_1' \leq S[\overline{b \mapsto c}]\rho_2'$. By applying then SKOL we are done.
- Case SPEC. We have that $\vdash \forall \overline{a}.\rho_1' \leq \rho_2'$ given that $\vdash \overline{[a \mapsto \boxed{\sigma}]}\rho_1' \leq \rho_2'$. Assume that $\overline{a} \# vars(S)$ otherwise we can apply an α-renaming to $\forall \overline{a}.\rho_1'$. Then we have by induction hypothesis that $\vdash \overline{[a \mapsto \boxed{S\sigma}]}S\rho_1' \leq S\rho_2'$. By applying SPEC we get $\vdash \forall \overline{a}.S\rho_1' \leq S\rho_2'$ and since $\overline{a} \# vars(S)$ this is equivalent to $\vdash S(\forall \overline{a}.\rho_1') \leq S\rho_2'$.

□

Remark 3.5 (Subsumption not reflexive). *It is not the case that* $\vdash \boxed{\sigma} \leq \boxed{\sigma}$ *for arbitrary* σ.

Proof. By contradiction assume that it is derivable; then the only rule applicable is SBOXY but $(\sim)$ is not reflexive by Remark 3.2. □

Remark 3.6 (Subsumption not transitive). *If* $\vdash \sigma'_1 \leq \sigma'_2$ *and* $\vdash \sigma'_2 \leq \sigma'_3$ *then it is not necessarily the case that* $\vdash \sigma'_1 \leq \sigma'_3$.

Proof. Take for example $\sigma'_1 = \boxed{\forall a\,.\,a \to a}$, $\sigma'_2 = \forall a\,.\,a \to a$ and $\sigma'_3 = a \to a$. It is not the case that $\vdash \boxed{\forall a\,.\,a \to a} \leq a \to a$. □

3.3 Boxing and unboxing of types

Controlled boxing around monotype information is given in Figure 4. Arbitrary boxing of types is given in Figure 5, where essentially the only difference is that rule UBBOX allows for boxing arbitrary types, not only monotypes. Finally a restriction of the subsumption relation that is valid on some syntactic categories of types and performs boxing perhaps on negative parts of types and skolemisation on positive parts is given in Figure 6. An auxilliary definition first. Let us define the function $strip(\cdot)$ as follows:

$$\begin{array}{lcl} strip(\forall \overline{a}\,.\,\rho') & = & \forall \overline{a}\,.\,strip(\rho') \\ strip(\sigma'_1 \to \sigma'_2) & = & strip(\sigma'_1) \to strip(\sigma'_2) \\ strip(\boxed{\sigma}) & = & \sigma \\ strip(T\ \overline{\sigma'}) & = & T\ \overline{strip(\sigma')} \\ strip(\tau) & = & \tau \end{array}$$

Strip merely removes the boxes off a boxy type (to avoid confusion we stress that it only has mathematical meaning, it is not used anywhere by the inference system).

Lemma 3.7. *If* $\vdash \tau \triangleright \sigma'$ *then* $\vdash \tau \sim \sigma'$ *and* $\vdash \boxed{\tau} \sim \sigma'$.

Proof. By induction on $\vdash \tau \triangleright \sigma'$. We have the following cases to consider.

- Case CBREFL. Here $\sigma' = \tau$ and the first result follows by MEQ2, the second from MEQ1 and SYM.
- Case CBBOX. Here $\sigma' = \boxed{\tau}$ and the first result follows by MEQ1, the second from BBEQ.
- Case CBFUN. Here $\tau = \tau_1 \to \tau_2$ and $\sigma' = \sigma'_1 \to \sigma'_2$ such that $\vdash \tau_1 \triangleright \sigma'_1$ and $\vdash \tau_2 \triangleright \sigma'_2$. By induction hypothesis $\vdash \tau_1 \sim \sigma'_1$ and $\vdash \tau_2 \sim \sigma'_2$, $\vdash \boxed{\tau_1} \sim \sigma'_1$ and $\vdash \boxed{\tau_2} \sim \sigma'_2$ and by AEQ2 we get the first result, using also AEQ1 we get the second.
- Case CBCON. Similar to the case of CBFUN.
- Case CBCONBOX. Straightforward, as this is the case where the type constructor has no argument types.

□

Lemma 3.8. *If* $\vdash \sigma' \blacktriangleright \tau$ *then* $\sigma' = \tau$.

Proof. By induction on the derivation $\vdash \sigma' \blacktriangleright \tau$. We have the following cases to consider.

- Case UBREFL. Trivial.
- Case UBFUN. In this case $\sigma' = \sigma'_1 \to \sigma'_2$ and $\tau = \tau_1 \to \tau_2$ such that $\vdash \sigma'_1 \blacktriangleright \tau_1$ and $\vdash \sigma'_2 \blacktriangleright \tau_2$. By induction hypothesis $\sigma'_1 = \tau_1$ and $\sigma'_2 = \tau_2$.
- Case UBCON. Similar to the case of UBFUN.
- Cases UBALL, UBBOX, UFUNBOX, UBCONBOX cannot happen (the case of UBALL can happen but it would be a trivial application).

□

Lemma 3.9. *The following are true of the* $(\sim)$ *relation:*

1. *If* $\vdash \sigma'_1 \sim \sigma'_2$ *and* $\vdash \sigma'_1 \triangleright \sigma'_3$ *then* $\vdash \sigma'_3 \sim \sigma'_2$.
2. *If* $\vdash \sigma'_1 \sim \sigma'_2$ *and* $\vdash \sigma'_2 \triangleright \sigma'_4$ *then* $\vdash \sigma'_1 \sim \sigma'_4$.

Proof. We prove the two claims simultaneously by induction on the height of the derivation $\vdash \sigma'_1 \sim \sigma'_2$. For each claim, the induction hypothesis asserts both claims for derivations of smaller height.

Part 1: For the first part we have the following cases to consider.

- Case SYM. The result follows from the induction hypothesis for the second claim.

- Case AEQ1. In this case we have that $\vdash \sigma_1' \to \sigma_2' \sim \boxed{\sigma_1 \to \sigma_2}$ given that $\vdash \sigma_1' \to \sigma_2' \sim \boxed{\sigma_1} \to \boxed{\sigma_2}$ and with an extra inversion

$$\vdash \sigma_1' \sim \boxed{\sigma_1} \tag{1}$$
$$\vdash \sigma_2' \sim \boxed{\sigma_2} \tag{2}$$

  By inversion on the $(\triangleright)$ relation we have the following cases to consider for $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma'$.

  - Case CBREFL. Trivial.
  - Case CBFUN. We have that $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma_1'' \to \sigma_2''$ where

$$\vdash \sigma_1' \triangleright \sigma_1'' \tag{3}$$
$$\vdash \sigma_2' \triangleright \sigma_2'' \tag{4}$$

    From (1) and (3) and induction hypothesis $\vdash \sigma_1'' \sim \boxed{\sigma_1}$, and from (2) and (4) and induction hypothesis we get $\vdash \sigma_2'' \sim \boxed{\sigma_2}$. By rule AEQ2 and AEQ1 we get that $\vdash \sigma_1'' \to \sigma_2'' \sim \boxed{\sigma_1 \to \sigma_2}$.
  - Case CBFUNBOX. In this case $\sigma_1' = \boxed{\sigma_1}$ and $\sigma_2' = \boxed{\sigma_2}$ and then by (1) and (2) it must be the case that $\sigma_1 = \tau_1$ and $\sigma_2 = \tau_2$. Then we need to show that $\vdash \boxed{\tau_1 \to \tau_2} \sim \boxed{\tau_1 \to \tau_2}$ but this follows from rule BBEQ.
  - Case CBBOX. In this case $\sigma_1' \to \sigma_2' = \tau_1 \to \tau_2$ and by (1) and (2) and an easy inversion on $(\sim)$ it must be that $\sigma_1 = \tau_1$ and $\sigma_2 = \tau_2$. Then we need to show that $\vdash \boxed{\tau_1 \to \tau_2} \sim \boxed{\tau_1 \to \tau_2}$ and this follows by rule BBEQ.

- Case AEQ2. In this case we have that $\vdash \sigma_1' \to \sigma_2' \sim \sigma_3' \to \sigma_4'$ given that

$$\vdash \sigma_1' \sim \sigma_3' \tag{5}$$
$$\vdash \sigma_2' \sim \sigma_4' \tag{6}$$

  By inversion on $(\triangleright)$ we have the following cases to consider for $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma'$.

  - Case CBREFL. Trivial.
  - Case CBFUN. In this case we have that $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma_1'' \to \sigma_2''$ where

$$\vdash \sigma_1' \triangleright \sigma_1'' \tag{7}$$
$$\vdash \sigma_2' \triangleright \sigma_2'' \tag{8}$$

    From (5) and (7) and induction hypothesis we get that $\vdash \sigma_1'' \sim \sigma_3'$, and using (6) and (8) we get $\vdash \sigma_2'' \sim \sigma_4'$. Then by applying rule AEQ2 we get $\vdash \sigma_1'' \to \sigma_2'' \sim \sigma_3' \to \sigma_4'$ as required.
  - Case CBFUNBOX. In this case we have that $\vdash \sigma_1' = \boxed{\sigma_1}$ and $\vdash \sigma_2' = \boxed{\sigma_2}$ and $\vdash \sigma_1' \to \sigma_2' \triangleright \boxed{\sigma_1 \to \sigma_2}$. We then need to show that $\vdash \boxed{\sigma_1 \to \sigma_2} \sim \sigma_3' \to \sigma_4'$. But this follows from SYM, equations (5) and (6) and rules AEQ1, AEQ2.
  - Case CBBOX. Similar to the case for CBFUNBOX.

- Case CEQ1. Similar to the case for AEQ1.
- Case CEQ2. Similar to the case for AEQ2.
- Case BBEQ. Trivial, since it can only be that $\vdash \boxed{\tau} \triangleright \boxed{\tau}$.
- Case MEQ1. Here we have that $\vdash \boxed{\tau} \sim \tau$ and we have the following cases for $\vdash \tau \triangleright \sigma'$.
  - Case CBREFL. Trivial.
  - Case CBBOX. The result follows then by rule BBEQ.
  - Case CBFUN. Then $\tau = \tau_1 \to \tau_2$ and we have that

$$\vdash \tau_1 \triangleright \sigma_1' \tag{9}$$
$$\vdash \tau_2 \triangleright \sigma_2' \tag{10}$$

    By Lemma 3.7 we know that $\vdash \boxed{\tau_1} \sim \sigma_1'$ and $\vdash \boxed{\tau_2} \sim \sigma_2'$. We need to show that $\vdash \sigma_1' \to \sigma_2' \sim \boxed{\tau_1 \to \tau_2}$. But this follows by applying AEQ1,AEQ2.
  - Case CBON. Similar to the case for CBFUN.

- Case CBCONBOX. In this case the constructor takes no argument types and the result follows by rule CEQ1.

- Case MEQ2. In this case we have that $\vdash \tau \sim \tau$. We have that $\vdash \tau \triangleright \sigma'$, therefore by Lemma 3.7 $\vdash \tau \sim \sigma'$ and by applying SYM we are done.
- Case SEQ1. Let us assume in this case that we have no useless quantifiers otherwise the case is a degenerate use of the rule. We have that $\vdash \forall \overline{a}.\, \rho_1' \sim \boxed{\forall \overline{a}.\, \rho_2}$ given that $\vdash \rho_1' \sim \boxed{\rho_2}$. There are two cases for $\vdash \forall \overline{a}.\, \rho_1' \triangleright \sigma'$.
  - Case CBREFL. Trivial.
  - Case CBALL. In this case $\vdash \forall \overline{a}.\, \rho_1' \triangleright \forall \overline{a}.\, \rho_1''$ given that $\vdash \rho_1' \triangleright \rho_1''$. By induction hypothesis then $\vdash \rho_1'' \sim \boxed{\rho_2}$ and by applying rule SEQ1 $\vdash \forall \overline{a}.\, \rho_1'' \sim \boxed{\forall \overline{a}.\, \rho_2}$ as required.
- Case SEQ2. Similar to the case for SEQ1.

Part 2: For the second part we have the following cases.

- Case SYM. The result follows from the induction hypothesis for the first claim.
- Case AEQ1. Trivial, since it can only be that $\vdash \boxed{\sigma_1 \to \sigma_2} \triangleright \boxed{\sigma_1 \to \sigma_2}$.
- Case AEQ2. In this case we have that $\vdash \sigma_1' \to \sigma_2' \sim \sigma_3' \to \sigma_4'$ given that

$$\vdash \sigma_1' \sim \sigma_3' \tag{11}$$
$$\vdash \sigma_2' \sim \sigma_4' \tag{12}$$

  By inversion on $(\triangleright)$ we have the following cases to consider for $\vdash \sigma_3' \to \sigma_4' \triangleright \sigma'$.
  - Case CBREFL. Trivial.
  - Case CBFUN. In this case we have that $\vdash \sigma_3' \to \sigma_4' \triangleright \sigma_3'' \to \sigma_4''$ where

$$\vdash \sigma_3' \triangleright \sigma_3'' \tag{13}$$
$$\vdash \sigma_4' \triangleright \sigma_4'' \tag{14}$$

    From (11) and (13) and induction hypothesis we get that $\vdash \sigma_1' \sim \sigma_3''$, and using (12) and (14) we get $\vdash \sigma_2' \sim \sigma_4''$. Then by applying rule AEQ2 we get $\vdash \sigma_1' \to \sigma_2' \sim \sigma_3'' \to \sigma_4''$ as required.
  - Case CBFUNBOX. In this case we have that $\vdash \sigma_3' = \boxed{\sigma_3}$ and $\vdash \sigma_4' = \boxed{\sigma_4}$ and $\vdash \sigma_3' \to \sigma_4' \triangleright \boxed{\sigma_3 \to \sigma_4}$. We then need to show that $\vdash \sigma_1' \to \sigma_2' \sim \boxed{\sigma_3 \to \sigma_4}$. But this follows from equations (11) and (12) and rules AEQ1, AEQ2.
  - Case CBBOX. Similar to the case for CBFUNBOX.
- Case CEQ1. Similar to the case for AEQ1.
- Case CEQ2. Similar to the case for AEQ2.
- Case BBEQ. Similar to the case for AEQ1.
- Case MEQ1. Trivial, since it can only be that $\vdash \boxed{\tau} \triangleright \boxed{\tau}$.
- Case MEQ2. In this case $\vdash \tau \sim \tau$ and by Lemma 3.7 we have that $\vdash \tau \triangleright \sigma'$ implies $\vdash \tau \sim \sigma'$ as required.
- Case SEQ1. Similar to the case for AEQ1.
- Case SEQ2. Assume we have no useless quantifiers. Then the case is similar to the case of SEQ2 of the first part.

□

Corollary 3.10 (Controlled boxing for matching). *If* $\vdash \sigma_1' \sim \sigma_2'$ *and* $\vdash \sigma_1' \triangleright \sigma_3'$ *and* $\vdash \sigma_2' \triangleright \sigma_4'$ *then* $\vdash \sigma_3' \sim \sigma_4'$.

Proof. Directly follows by Lemma 3.9 and the fact that $(\triangleright)$ is reflexive. □

Lemma 3.11. *The following are true of the* $(\sim)$ *relation:*

1. *If* $\vdash \sigma_1' \sim \sigma_2'$ *and* $\vdash \sigma_3' \blacktriangleright \sigma_1'$ *then* $\vdash \sigma_3' \sim \sigma_2'$.
2. *If* $\vdash \sigma_1' \sim \sigma_2'$ *and* $\vdash \sigma_4' \blacktriangleright \sigma_2'$ *then* $\vdash \sigma_1' \sim \sigma_4'$.

Proof. We prove the two claims simultaneously by induction on the height of the derivation $\vdash \sigma'_1 \sim \sigma'_2$. For each part the induction hypothesis asserts both claims for derivations of smaller height. For each part we proceed with case analysis on the last rule used in the derivation.

Part 1: For the first part we consider the following cases.

- Case SYM. Follows by induction hypothesis for the second part.
- Case AEQ1. Here we have that $\vdash \sigma'_1 \to \sigma'_2 \sim \boxed{\sigma_1 \to \sigma_2}$ given that $\vdash \sigma'_1 \to \sigma'_2 \sim \boxed{\sigma_1} \to \boxed{\sigma_2}$ and with an extra inversion we get

$$\vdash \sigma'_1 \sim \boxed{\sigma_1} \tag{15}$$
$$\vdash \sigma'_2 \sim \boxed{\sigma_2} \tag{16}$$

  The cases for $\vdash \sigma' \blacktriangleright \sigma'_1 \to \sigma'_2$ are the following:

  - Case UBREFL. Trivial.
  - Case UBFUN. In this case we have that $\vdash \sigma''_1 \to \sigma''_2 \blacktriangleright \sigma'_1 \to \sigma'_2$, given that

$$\vdash \sigma''_1 \blacktriangleright \sigma'_1 \tag{17}$$
$$\vdash \sigma''_2 \blacktriangleright \sigma'_2 \tag{18}$$

    Then by induction hypothesis from (15) and (17) we get $\vdash \sigma''_1 \sim \boxed{\sigma_1}$. Similarly from (16) and (18) we get $\vdash \sigma''_2 \sim \boxed{\sigma_2}$. Then the result follows by rules AEQ1 and AEQ2.

  .

- Case AEQ2. Similar to the case for AEQ1.
- Case CEQ1. Similar to the case for AEQ1.
- Case CEQ2. Similar to the case for AEQ2.
- Case BBEQ. Here $\vdash \boxed{\tau} \sim \boxed{\tau}$. We have the following cases for $\vdash \sigma' \blacktriangleright \boxed{\tau}$.
  - Case UBREFL. Trivial.
  - Case UBBOX. In this case the result follows by rule MEQ1.
  - Case UBFUNBOX. In this case we have that $\tau = \tau_1 \to \tau_2$ and we need to show that $\vdash \boxed{\tau_1} \to \boxed{\tau_2} \sim \boxed{\tau_1 \to \tau_2}$. But this follows by rules AEQ1, AEQ2, and BBEQ.
  - Case UBCONBOX. Similar to the case of UFUNBOX.
- Case MEQ1. Can't happen, except for the reflexive case which is trivial.
- Case MEQ2. In this case we have $\vdash \tau \sim \tau$. We have that $\vdash \sigma' \blacktriangleright \tau$, therefore by Lemma 3.8 we get $\sigma' = \tau$ as required and MEQ2 finishes the case.
- Case SEQ1. We have that $\vdash \forall \overline{a}.\rho'_1 \sim \boxed{\forall \overline{a}.\rho_2}$ given that $\vdash \rho'_1 \sim \boxed{\rho_2}$. Then we need to consider cases for $\vdash \sigma' \blacktriangleright \forall \overline{a}.\rho'_1$. Assume as well that the quantifiers are not trivial otherwise we could just ommit the rule application in the original derivation.
  - Case UBREFL. Trivial.
  - Case UBALL. We have that $\vdash \forall \overline{a}.\rho''_1 \blacktriangleright \forall \overline{a}.\rho'_1$ given that $\vdash \rho''_1 \blacktriangleright \rho'_1$. We can apply the induction hypothesis then and rule SEQ1 to get the result.
- Case SEQ2. Similar to the case of SEQ1.

Part 2: For the second part we consider the following cases.

- Case SYM. Follows by induction hypothesis for the first part.
- Case AEQ1. We have that $\vdash \sigma'_1 \to \sigma'_2 \sim \boxed{\sigma_1 \to \sigma_2}$ given that

$$\vdash \sigma'_1 \to \sigma'_2 \sim \boxed{\sigma_1} \to \boxed{\sigma_2} \tag{19}$$
$$\vdash \sigma'_1 \sim \boxed{\sigma_1} \tag{20}$$
$$\vdash \sigma'_2 \sim \boxed{\sigma_2} \tag{21}$$

  Then we consider cases for $\vdash \sigma' \blacktriangleright \boxed{\sigma_1 \to \sigma_2}$.

  - Case UBREFL. Trivial.

- Case UBBOX. In this case $\sigma' = \sigma_1 \to \sigma_2$. By induction hypothesis on (20) we get $\vdash \sigma_1' \sim \sigma_1$ and by induction hypothesis on (21) we get $\vdash \sigma_2' \sim \sigma_2$. Then by AEQ2 we get $\vdash \sigma_1' \to \sigma_2' \sim \sigma_1 \to \sigma_2$ as required.
- Case UFUNBOX. In this case $\sigma' = \boxed{\sigma_1} \to \boxed{\sigma_2}$ and we already have the result from (19).

- Case AEQ2. Here we have that $\vdash \sigma_1' \to \sigma_2' \sim \sigma_3' \to \sigma_4'$ given that

$$\vdash \sigma_1' \sim \sigma_3' \tag{22}$$
$$\vdash \sigma_2' \sim \sigma_4' \tag{23}$$

  The cases for $\vdash \sigma' \leq \sigma_3' \to \sigma_4'$ are the following.

  - Case UBREFL. Trivial.
  - Case UBFUN. In this case we have that $\sigma' = \sigma_3'' \to \sigma_4''$ such that

$$\vdash \sigma_3'' \blacktriangleright \sigma_3' \tag{24}$$
$$\vdash \sigma_4'' \blacktriangleright \sigma_4' \tag{25}$$

    From (22) and (24) and induction hypothesis we get $\vdash \sigma_1' \sim \sigma_3''$ and from (23) and (25) we get $\vdash \sigma_2' \sim \sigma_4''$. Then by rule AEQ2 we get $\vdash \sigma_1' \to \sigma_2' \sim \sigma_3'' \to \sigma_4''$ as required.

- Case CEQ1. Similar to the case for AEQ1.
- Case CEQ2. Similar to the case for AEQ2.
- Case BBEQ. In this case we have $\vdash \boxed{\tau} \sim \boxed{\tau}$. We have the following case for $\vdash \sigma' \sim \boxed{\tau}$
  - Case UBREFL. Trivial.
  - Case UBBOX. It must be that $\sigma' = \tau$ and the result follows by MEQ1 and SYM.
  - Case UBFUNBOX. In this case $\tau = \tau_1 \to \tau_2$ and $\sigma' = \boxed{\tau_1} \to \boxed{\tau_2}$. We must show that $\vdash \boxed{\tau_1 \to \tau_2} \sim \boxed{\tau_1} \to \boxed{\tau_2}$ but this follows from rules SYM, AEQ1, AEQ2, and BBEQ.
  - Case UBCONBOX. Similar to the case for UBFUNBOX.
- Case MEQ1. Straightforward case analysis.
- Case MEQ2. Similar to the case for MEQ1.
- Case SEQ1. In this case we have that $\vdash \forall \overline{a}.\, \rho_1' \sim \boxed{\forall \overline{a}.\, \rho_2}$ given that $\vdash \rho_1' \sim \boxed{\rho_2}$. Then we have to consider the following cases for $\vdash \sigma' \blacktriangleright \boxed{\forall \overline{a}.\, \rho_2}$.
  - Case UBREFL. Trivial.
  - Case UBBOX. We have that $\sigma' = \forall \overline{a}.\, \rho_2$. It is then enough to show that $\vdash \rho_1' \sim \rho_2$ but this follows from induction hypothesis, since $\vdash \rho_2 \blacktriangleright \boxed{\rho_2}$.
- Case SEQ2. Again, assume no useless quantifiers. We have that $\vdash \forall \overline{a}.\, \rho_1' \sim \forall \overline{a}.\, \rho_2'$ given that $\vdash \rho_1' \sim \rho_2'$. The only cases for $\vdash \sigma' \blacktriangleright \forall \overline{a}.\, \rho_2'$ are the following.
  - Case UBREFL. Trivial.
  - Case UBALL. In this case $\sigma' = \forall \overline{a}.\, \rho_2''$ such that $\vdash \rho_2'' \blacktriangleright \rho_2$. By induction hypothesis then $\vdash \rho_1 \sim \rho_2''$ and the result follows from SEQ2 again.

□

Corollary 3.12 (Uncontrolled unboxing for matching). *If* $\vdash \sigma_1' \sim \sigma_2'$ *and* $\vdash \sigma_3' \blacktriangleright \sigma_1'$ *and* $\vdash \sigma_4' \blacktriangleright \sigma_2'$ *then* $\vdash \sigma_3' \sim \sigma_4'$.

Proof. Directly follows by Lemma 3.9 and the fact that $(\blacktriangleright)$ is reflexive. □

Lemma 3.13 (Reflexivity on box-free types for matching). $\vdash \sigma \sim \sigma$.

Proof. By induction on the structure of σ.

- Case $\sigma = \forall \overline{a}.\, \rho$. By induction hypothesis and rule SEQ2.
- Case $\sigma = \sigma_1 \to \sigma_2$. By induction hypothesis and rule AEQ2.
- Case $\sigma = T\ \overline{\sigma}$. By induction hypothesis and rule CEQ2.

- Case $\sigma = \tau$. Follows by MEQ2.

□

Lemma 3.14 (One-sided boxy matching). $\vdash \sigma \sim \boxed{\sigma}$.

Proof. By induction on the structure of σ.

- Case $\sigma = \forall \overline{a}.\rho$. By induction hypothesis and rule SEQ1.
- Case $\sigma = \sigma_1 \to \sigma_2$. By induction hypothesis and rule AEQ1.
- Case $\sigma = T\ \overline{\sigma}$. By induction hypothesis and rule CEQ1.
- Case $\sigma = \tau$. By MEQ1.

□

Corollary 3.15. $\vdash \sigma' \sim \mathit{strip}(\sigma')$.

Proof. By induction on the structure of σ'.

- Case $\sigma' = \forall \overline{a}.\rho'$. By induction hypothesis and rule SEQ2.
- Case $\sigma' = \sigma'_1 \to \sigma'_2$. By induction hypothesis and rule AEQ2.
- Case $\sigma' = T\ \overline{\sigma}'$. By induction hypothesis and rule CEQ2.
- Case $\sigma' = \boxed{\sigma}$. By Lemma 3.14.
- Case $\sigma' = \tau$. Follows by MEQ2.

□

Lemma 3.16. *If* $\vdash \tau \triangleright \sigma'$ *then*

1. $\vdash \tau \leq \sigma'$.
2. $\vdash \sigma' \leq \tau$.
3. $\vdash \boxed{\tau} \leq \sigma'$.
4. $\vdash \sigma' \leq \boxed{\tau}$.

Proof. By induction on the derivation $\vdash \tau \triangleright \sigma'$. We have to consider the following cases.

- Case CBREFL. The first two claims follow by MONO, the third by SBOXY, and the last by BMONO.
- Case CBFUN. In this case $\tau = \tau_1 \to \tau_2$ and $\sigma' = \sigma'_1 \to \sigma'_2$ such that $\vdash \tau_1 \triangleright \sigma'_1$ and $\vdash \tau_2 \triangleright \sigma'_2$. By Lemma 3.7 and by Corollary 3.10 we get

$$\vdash \tau_1 \sim \sigma'_1 \tag{1}$$
$$\vdash \boxed{\tau_1} \sim \sigma'_1 \tag{2}$$
$$\vdash \tau_2 \sim \sigma'_2 \tag{3}$$
$$\vdash \boxed{\tau_2} \sim \sigma'_2 \tag{4}$$

  By induction hypothesis we have

$$\vdash \tau_2 \leq \sigma'_2 \tag{5}$$
$$\vdash \sigma'_2 \leq \tau_2 \tag{6}$$
$$\vdash \boxed{\tau_2} \leq \sigma'_2 \tag{7}$$
$$\vdash \sigma'_2 \leq \boxed{\tau_2} \tag{8}$$

  By (1) and (5) and AEQ2 we get that $\vdash \tau \leq \sigma'$. By (1) and (6) and AEQ2 we get that $\vdash \sigma' \leq \tau$. By (2) and (4) and SBOXY we get that $\vdash \boxed{\tau} \leq \sigma'$. By (1) and (8) and rules AEQ1, AEQ2 we get $\vdash \sigma' \leq \boxed{\tau}$.
- Case CBCON. Similar to the case for CBFUN.

□

Lemma 3.17. *If* $\vdash \sigma'_1 \sim \sigma'_2$ *then* $\mathit{ftv}(\sigma'_1) = \mathit{ftv}(\sigma'_2)$.

Proof. Easy induction. □

We next introdude the notion of a *protected* polytype. Protected polytypes are those whose polymorphism is hidden under a box. They are just a subset of the domain of σ', defined as follows.

$$\varpi \quad ::= \quad \tau \mid \boxed{\sigma} \mid \varpi \to \varpi \mid \; T\ \sigma'$$

A feature of the controlled (but not of the uncontrolled) boxing is that whenever the right-hand type is a box, the left-hand type is already protected.

Lemma 3.18. *If* $\vdash \sigma' \triangleright \boxed{\sigma}$ *then* $\sigma' = \varpi$.

Proof. By induction on the derivation $\vdash \sigma' \triangleright \boxed{\sigma}$.

- Case CBREFL. We have that $\boxed{\sigma} \in dom(\varpi)$.
- Case CBBOX. Monotypes are in $dom(\varpi)$.
- Case CBFUNBOX. Types of the form $\boxed{\sigma_1} \to \boxed{\sigma_2}$ are in $dom(\varpi)$.
- Case CBCONBOX. Types of the form $T\ \overline{\boxed{\sigma}}$ are in $dom(\varpi)$.
- The rest of the cases cannot happen.

□

Next we assert that no skolemisation can happen for protected types in subsumption.

Lemma 3.19. *If* $\vdash \varpi \leq \sigma'$ *then* $ftv(\sigma') \subseteq ftv(\varpi)$.

Proof. By induction on the derivation $\vdash \varpi \leq \sigma'$.

- Cases SBOXY and CON. Follow directly by Lemma 3.17
- Cases MONO and BMONO. Immediate.
- Case SKOL. In this case we know that $\vdash \varpi \leq \forall \overline{b}.\rho_2'$ given that $\vdash \varpi \leq \rho_2'$ and $\overline{b} \# ftv(\varpi)$. By induction hypothesis $ftv(\rho_2') \subseteq ftv(\varpi)$ and therefore $ftv(\forall \overline{b}.\rho_2') \subseteq ftv(\varpi)$.
- Case SPEC. Can't happen.
- Case F2. Here we have that $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3' \to \sigma_4'$ given that $\vdash \sigma_3' \sim \sigma_1'$ and $\vdash \sigma_2' \leq \sigma_4'$. But it must be that $\sigma_1' = \varpi_1$ and $\sigma_2' = \varpi_2$ and the result follows by Lemma 3.17 and induction hypothesis.
- Case F1. Similar to the case of F2 with an extra inversion step.

□

Lemma 3.20. *If* $\vdash \tau \leq \sigma'$ *then* $\vdash \sigma' \sim \tau$.

Proof. By induction on the derivation $\vdash \tau \leq \sigma'$. The case for SBOXY cannot happen. The cases for MONO and BMONO are easy. Case CON follows directly by the premise and case SPEC can't happen. For the rest of the cases we have:

- Case SKOL. Assume without loss of generality that this is a non-trivial application of the rule. then τ is a protected type and by Lemma 3.19 this case cannot happed because otherwise it must be that the skolem constants are in $ftv(\tau)$.
- Case F1. Directly follows by induction hypothesis and rule AEQ1.
- Case F2. In this case we have that $\vdash \tau_1 \to \tau_2 \leq \sigma_3' \to \sigma_4'$ given that $\vdash \sigma_3' \sim \tau_1$ and $\vdash \tau_2 \leq \sigma_4'$. By induction hypothesis $\vdash \sigma_4' \sim \tau_2$ and by applying AEQ2 we are done.

□

Lemma 3.21. *If* $\overline{a}$ *do not occur inside boxes in* ρ_1', ρ_1'' *and* $\vdash \rho_1' \triangleright \rho_1''$ *then* $[\overline{a \mapsto \sigma'}]\rho_1' \triangleright [\overline{a \mapsto \sigma'}]\rho_1''$

Proof. Easy induction on the derivation $\vdash \rho_1' \triangleright \rho_1''$. Let us assume without loss of generality that $\overline{a} \in ftv(\rho_1', \rho_1'')$. Then the case for REF follows again by rule REFL. Case CBBOX cannot happen. Cases CBALL, CBFUN and CBCON all follow by induction hypothesis (for CBALL we use the non syntactic invariant that all well-formed types cannot contain quantified variables free inside a box). Finally cases CBFUNBOX and CBCONBOX cannot happen. □

Lemma 3.22. *If* $\overline{a}$ *do not occur inside boxes in* ρ_1', ρ_1'' *and* $\vdash \rho_1' \blacktriangleright \rho_1''$ *then* $[\overline{a \mapsto \sigma'}]\rho_1' \blacktriangleright [\overline{a \mapsto \sigma'}]\rho_1''$

Proof. Similar to the proof of Lemma 3.21. □

Lemma 3.23. *Boxing (controlled or not) does not affect free type variables.*

1. *If* $\vdash \sigma_1' \triangleright \sigma_2'$ *then* $ftv(\sigma_1') = ftv(\sigma_2')$.
2. *If* $\vdash \sigma_1' \blacktriangleright \sigma_2'$ *then* $ftv(\sigma_1') = ftv(\sigma_2')$.

Proof. Easy induction. □

Lemma 3.24. *If* $\vdash \sigma_1' \leq \sigma_2'$ *and* $\vdash \sigma_1' \triangleright \sigma_3'$ *then* $\vdash \sigma_3' \leq \sigma_2'$.

Proof. The proof is by induction on the derivation $\vdash \sigma_1' \leq \sigma_2'$. We proceed with case analysis on the last rule used in the derivation.

- Case SBOXY. In this case we have that $\vdash \boxed{\sigma} \leq \sigma'$ given that $\vdash \boxed{\sigma} \sim \sigma'$. Then it can only be that $\vdash \boxed{\sigma} \triangleright \boxed{\sigma}$ and the result follows trivially.
- Case MONO. In this case we have $\vdash \tau \leq \tau$ and we have $\vdash \tau \triangleright \sigma'$. Then by Lemma 3.16 we get that $\vdash \sigma' \leq \tau$ as required.
- Case BMONO. Here $\vdash \tau \leq \boxed{\tau}$ and $\vdash \tau \triangleright \sigma'$. Then by Lemma 3.16 we get $\vdash \sigma' \leq \boxed{\tau}$.
- Case CON. Follows by Corollary 3.10.
- Case SKOL. We have that $\vdash \sigma_1' \leq \forall \overline{b}.\rho_2'$ given that

$$\sigma_1' \neq \boxed{\sigma} \tag{1}$$
$$\overline{b} \# ftv(\sigma_1') \tag{2}$$
$$\vdash \sigma_1' \leq \rho_2' \tag{3}$$

  Assume that $\vdash \sigma_1' \triangleright \sigma_1''$. Then by Lemma (3.23) $ftv(\sigma_1') = ftv(\sigma_1'')$. Assume additionally that the application of the rule is a non trivial one, otherwise we get the result directly from the induction hypothesis, that is assume $\overline{b} \neq \emptyset$ and $\overline{b} \subseteq ftv(\rho_2')$. By induction hypothesis we get

$$\vdash \sigma_1'' \leq \rho_2' \tag{4}$$

  To be able to apply rule SKOL again we need to show that $\sigma_1'' \neq \boxed{\sigma}$. Assume by contradiction that $\sigma_1'' = \boxed{\sigma}$. Then by Lemma 3.18 it must be that σ_1' is protected. Then by Lemma 3.19 and equation (3) it must be that $ftv(\rho_2') \subseteq ftv(\sigma_1')$ therefore $\overline{b} \subseteq ftv(\sigma_1')$, a contradiction. Therefore $\sigma_1'' \neq \boxed{\sigma}$, and we can apply rule SKOL to get the result.
- Case SPEC. Here $\vdash \forall \overline{a}.\rho_1' \leq \rho_2'$ given that $\vdash [\overline{a \mapsto \boxed{\sigma}}]\rho_1' \leq \rho_2'$. By inversion on $(\triangleright)$ we get either that CBREFL was used and we are trivially done, or that $\vdash \forall \overline{a}.\rho_1' \triangleright \forall \overline{a}.\rho_1''$ such that $\vdash \rho_1' \triangleright \rho_1''$ and moreover $\overline{a}$ don't appear free inside boxes in ρ_1', ρ_1'', so that the types are well-formed. Then by Lemma 3.21 we get that $\vdash [\overline{a \mapsto \boxed{\sigma}}]\rho_1' \triangleright [\overline{a \mapsto \boxed{\sigma}}]\rho_1''$ and by induction hypothesis $\vdash [\overline{a \mapsto \boxed{\sigma}}]\rho_1'' \leq \rho_2'$. Applying rule SPEC finishes the case.
- Case F1. We have that $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3 \to \sigma_4}$ given that $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3} \to \boxed{\sigma_4}$, or with an extra inversion step:

$$\vdash \boxed{\sigma_3} \sim \sigma_1' \tag{5}$$
$$\vdash \sigma_2' \leq \boxed{\sigma_4} \tag{6}$$

  We proceed by taking cases for $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma'$:

  - Case CBREFL. Trivial.
  - Case CBFUN. Direct application of induction hypothesis and rule F1.
  - Case CBBOX. We have that $\sigma_1' \to \sigma_2' = \tau_1 \to \tau_2$ and $\sigma' = \boxed{\tau_1 \to \tau_2}$. We want to show that $\vdash \boxed{\tau_1 \to \tau_2} \leq \boxed{\sigma_3} \to \boxed{\sigma_4}$, or equivalently $\vdash \boxed{\tau_1 \to \tau_2} \sim \boxed{\sigma_3} \to \boxed{\sigma_4}$, or equivalently $\vdash \boxed{\tau_1} \sim \boxed{\sigma_3}$ and $\vdash \boxed{\tau_2} \sim \boxed{\sigma_4}$. But we get the first equation from equation (5) and the latter from equation (6) and Lemma 3.20, and Corollary 3.10.
  - Case CBFUBOX. Similar to the case for CBBOX.
  - The rest of the cases cannot happen.
- Case F2. In this case $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3' \to \sigma_4'$ given that $\vdash \sigma_3' \sim \sigma_1'$ and $\vdash \sigma_2' \leq \sigma_4'$. We consider cases for $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma'$:

  - Case CBREFL. Trivial.

– Case CBBOX. In this case $\sigma_1' \to \sigma_2' = \tau_1 \to \tau_2$ and we know that $\vdash \sigma_3' \sim \tau_1$, therefore by Corollary 3.10 $\vdash \sigma_3' \sim \boxed{\tau_1}$. Similarly we know that

$$\vdash \tau_2 \leq \sigma_4' \tag{7}$$

By Lemma 3.20 we get that $\vdash \tau_2 \sim \sigma_4'$ or by Corollary 3.10

$$\vdash \boxed{\tau_2} \sim \sigma_4' \tag{8}$$

Then we need to show that $\vdash \boxed{\tau_1 \to \tau_2} \leq \sigma_3' \to \sigma_4'$ or using SBOXY $\vdash \boxed{\tau_1 \to \tau_2} \leq \sigma_3' \to \sigma_4'$. But this is derivable from equations (7) and (8), rule SYM, and rules AEQ1 and AEQ2.

– Case CBFUN. Easily follows by induction hypothesis and Corollary 3.10.

– Case CBFUNBOX. In this case $\sigma_1' = \boxed{\sigma_1}$ and $\sigma_2' = \boxed{\sigma_2}$ and $\sigma' = \boxed{\sigma_1 \to \sigma_2}$. Then we have that $\vdash \boxed{\sigma_1} \sim \sigma_3'$ and $\vdash \boxed{\sigma_2} \leq \sigma_4'$, which with an inversion gives $\vdash \boxed{\sigma_2} \sim \sigma_4'$ Then we need to show that $\vdash \boxed{\sigma_1 \to \sigma_2} \leq \sigma_3' \to \sigma_4'$ but this follows by SBOXY, and rules SYM, AEQ1, and AEQ2.

– The rest of the cases cannot happen.

□

Lemma 3.25. *If* $\vdash \sigma_1' \leq \sigma_2'$ *and* $\vdash \sigma_2' \triangleright \sigma_4'$ *then* $\vdash \sigma_1' \leq \sigma_4'$.

Proof. By induction on the derivation of $\vdash \sigma_1' \leq \sigma_2'$. We proceed with case analysis on the last rule used.

- Case SBOXY. In this case $\vdash \boxed{\sigma} \leq \sigma'$ given that $\vdash \boxed{\sigma} \sim \sigma'$. Suppose that $\vdash \sigma' \triangleright \sigma''$. Then by Corollary 3.10 we get $\vdash \boxed{\sigma} \sim \sigma''$ and by applying SBOXY again we are done.
- Case MONO. In this case $\vdash \tau \leq \tau$ and $\vdash \tau \triangleright \sigma'$. By Lemma 3.16 we get $\vdash \tau \leq \sigma'$ as required.
- Case BMONO. Here $\vdash \tau \leq \boxed{\tau}$ and only CBREFL can be applied to $\boxed{\tau}$, so the result follows trivially.
- Case CON. Similar to the case for SBOXY appealing to Corollary 3.10.
- Case SPEC. Directly follows by induction hypothesis.
- Case SKOL. We have that $\vdash \sigma_1' \leq \forall \overline{b}.\rho_2'$ given that $\sigma_1' \neq \boxed{\sigma}$, $\overline{b} \# ftv(\sigma_1')$ and $\vdash \sigma_1' \leq \rho_2'$. Assume as well that this is a non-trivial application of the rule, otherwise we get the result directly by induction hypothesis. Then it must be that either that CBREFL was used and we are trivially done, or that $\vdash \forall \overline{b}.\rho_2' \triangleright \forall \overline{b}.\rho_2''$ such that $\vdash \rho_2' \triangleright \rho_2''$. By induction hypothesis $\vdash \sigma_1' \leq \rho_2''$ and applying rule SKOL finishes the case.
- Case F1. Similar to the case of BMONO.
- Case F2. Here we have that $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3' \to \sigma_4'$ given that

$$\vdash \sigma_3' \sim \sigma_1' \tag{9}$$
$$\vdash \sigma_2' \leq \sigma_4' \tag{10}$$

We consider cases for $\vdash \sigma_3' \to \sigma_4' \triangleright \sigma'$:

– Case CBREFL. Trivial.

– Case CBBOX. Here $\sigma_3' \to \sigma_4' = \tau_3 \to \tau_4$ and $\sigma' = \boxed{\tau_3 \to \tau_4}$. Then we need to show that $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\tau_3 \to \tau_4}$ or by using rules F1 and F2 it is enough to show that $\vdash \sigma_1' \sim \boxed{\tau_3}$ and $\vdash \sigma_2' \leq \boxed{\tau_4}$. The first follows by (9) and Corollary 3.10. The secon follows by (10) and induction hypothesis, since $\vdash \tau_4 \triangleright \boxed{\tau_4}$.

– Case CBFUN. Here we have that $\vdash \sigma_3' \to \sigma_4' \triangleright \sigma_3'' \to \sigma_4''$ given that

$$\vdash \sigma_3' \triangleright \sigma_3'' \tag{11}$$
$$\vdash \sigma_4' \triangleright \sigma_4'' \tag{12}$$

From (9), (11) and Corollary 3.10 we get that $\vdash \sigma_3'' \sim \sigma_1'$. From (10), (12) and induction hypothesis $\vdash \sigma_2' \leq \sigma_4''$. The result follows by applying rule F2 again.

– Case CBFUNBOX. Here $\sigma_3' = \boxed{\sigma_3}$ and $\sigma_4' = \boxed{\sigma_4}$. Equation (9) gives then $\vdash \boxed{\sigma_3} \sim \sigma_1'$ and (10) gives $\vdash \sigma_2' \leq \boxed{\sigma_4}$. We need to show that $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3 \to \sigma_4}$, or by F1 $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3} \to \boxed{\sigma_4}$, or by F2 $\vdash \boxed{\sigma_3} \sim \sigma_1'$ and $\vdash \sigma_2' \leq \boxed{\sigma_4}$ which we already have.

– The rest cases cannot happen.

□

Corollary 3.26 (Controlled boxing for subsumption). *If* $\vdash \sigma_1' \leq \sigma_2'$ *and* $\vdash \sigma_1' \triangleright \sigma_3'$ *and* $\vdash \sigma_2' \triangleright \sigma_4'$ *then* $\vdash \sigma_3' \leq \sigma_4'$.

Proof. Follows by Lemma 3.24 and Lemma 3.25 and the fact that $(\triangleright)$ is reflexive. □

Lemma 3.27. *If* $\vdash \sigma_1' \leq \sigma_2'$ *and* $\vdash \sigma_3' \blacktriangleright \sigma_1'$ *then* $\vdash \sigma_3' \leq \sigma_2'$.

Proof. By induction on the derivation $\vdash \sigma_1' \leq \sigma_2'$. We proceed by case analysis on the last rule used.

- Case SBOXY. Follows by Corollary 3.12 and rule SBOXY.
- Case MONO. We have that $\vdash \tau \leq \tau$ and by Lemma 3.8 it can only be that $\vdash \tau \blacktriangleright \tau$. The result follows trivially.
- Case BMONO. Similar to the case for MONO.
- Case CON. Follows by Corollary 3.12 and rule COR.
- Case SKOL. In this case we have that $\vdash \sigma_1' \leq \forall \overline{b}.\rho_2'$ given that $\sigma_1' \neq \boxed{\sigma}$, $\overline{b} \# ftv(\sigma_1')$, and $\vdash \sigma_1' \leq \rho_2'$. Assume that $\vdash \sigma_1'' \blacktriangleright \sigma_1'$. Then it cannot be that $\sigma_1'' = \boxed{\sigma}$ because it would have to be as well that $\sigma_1' = \boxed{\sigma}$. Also by Lemma 3.23 $ftv(\sigma_1'') = ftv(\sigma_1')$ and we can apply rule SKOL to get the result.
- Case SPEC. We have in this case that $\vdash \forall \overline{a}.\rho_1' \leq \rho_2'$ given that $\vdash \overline{[a \mapsto \boxed{\sigma}]}\rho_1' \leq \rho_2'$. Then the only case for $\vdash \sigma' \blacktriangleright \forall \overline{a}.\rho_1'$ is either using UBREFL, in which case the result follows trivially, or using UBALL. In the latter case we have $\vdash \forall \overline{a}.\rho_1'' \blacktriangleright \forall \overline{a}.\rho_1'$ where $\vdash \rho_1'' \blacktriangleright \rho_1'$. By Lemma 3.22 we get that $\vdash [a \mapsto \boxed{\sigma}]\rho_1'' \blacktriangleright [a \mapsto \boxed{\sigma}]\rho_1'$ and by induction hypothesis $\vdash \overline{[a \mapsto \boxed{\sigma}]}\rho_1'' \leq \rho_2'$. Applying rule SPEC finishes the case.
- Case F1. In this case we have $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3 \to \sigma_4}$ given that

$$\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3} \to \boxed{\sigma_4} \tag{1}$$

  We have only two cases for $\vdash \sigma' \blacktriangleright \sigma_1' \to \sigma_2'$. If rule UBREFL was used then we are trivially done. If rule UBFUN was used we are done by an application of the induction hypothesis for (1) and rule F1.
- Case F2. Here $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3' \to \sigma_4'$ given that

$$\vdash \sigma_3' \sim \sigma_1' \tag{2}$$
$$\vdash \sigma_2' \leq \sigma_4' \tag{3}$$

  We consider cases for $\vdash \sigma' \blacktriangleright \sigma_1' \to \sigma_2'$.

  - Case UBREFL. Trivial.
  - Case UBFUN. We have that $\sigma' = \sigma_1'' \to \sigma_2''$ such that

$$\vdash \sigma_1'' \blacktriangleright \sigma_1' \tag{4}$$
$$\vdash \sigma_2'' \blacktriangleright \sigma_2' \tag{5}$$

    From (2) and (4) and Corollary 3.12 $\vdash \sigma_3 \sim \sigma_1''$, and using (3) and (5) and induction hypothesis we get $\vdash \sigma_2'' \leq \sigma_4'$. The result then follows by applying F2.

□

Lemma 3.28. *If* $\vdash \sigma_1' \leq \sigma_2'$ *and* $\vdash \sigma_4' \blacktriangleright \sigma_2'$ *then* $\vdash \sigma_1' \leq \sigma_4'$.

Proof. By induction on the derivation $\vdash \sigma_1' \leq \sigma_2'$. We proceed by case analysis on the last rule used in the derivation.

- Case SBOXY. Follows by Corollary 3.12 and application of rule SBOXY.
- Case MONO. We have that $\vdash \tau \leq \tau$ and by Lemma 3.8 it can only be that $\vdash \tau \blacktriangleright \tau$. The result follows by MONO.
- Case BMONO. In this case we have $\vdash \tau \leq \boxed{\tau}$. We consider cases for $\vdash \sigma' \blacktriangleright \boxed{\tau}$:
  - Case UBREFL. Trivial.
  - Case UBBOX. In this case we have $\sigma' = \tau$ and we need to show that $\vdash \tau \leq \tau$ but this follows by rule MONO.
  - Case UBFUNBOX. In this case we have that $\tau = \tau_1 \to \tau_2$, $\sigma' = \boxed{\tau_1} \to \boxed{\tau_2}$. We need to show that $\vdash \tau_1 \to \tau_2 \leq \boxed{\tau_1} \to \boxed{\tau_2}$, and both $\vdash \tau_1 \sim \boxed{\tau_1}$ and $\vdash \tau_2 \leq \boxed{\tau_2}$ are derivable.
  - Case UBCONBOX. Similar to the case for UBFUNBOX.
- Case CON. Easily follows by Corollary 3.12 and rule CON.

- Case SKOL. We have that $\vdash \sigma_1' \leq \forall \overline{b}.\rho_2'$ given that $\sigma_1' \neq \boxed{\sigma}$, $\overline{b} \# ftv(\sigma_1')$ and

$$\vdash \sigma_1' \leq \rho_2' \tag{1}$$

  Consider cases for $\vdash \sigma' \blacktriangleright \forall \overline{b}.\rho_2$. If UBREFL was used then the result follows trivially. Otherwise UBALL must have been used and $\sigma' = \forall \overline{b}.\rho_2''$ such that $\vdash \rho_2'' \blacktriangleright \rho_2'$. By induction hypothesis for (1) we get then that $\vdash \sigma_1 \leq \rho_2''$ and we are done by applying SKOL again.

- Case SPEC. Directly follows by induction hypothesis and application of rule SPEC.

- Case F1. We have that $\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3 \to \sigma_4}$ given that

$$\vdash \sigma_1' \to \sigma_2' \leq \boxed{\sigma_3} \to \boxed{\sigma_4} \tag{2}$$
$$\vdash \boxed{\sigma_3} \sim \sigma_1' \tag{3}$$
$$\vdash \sigma_2' \leq \boxed{\sigma_4} \tag{4}$$

  We have the following cases to consider for $\vdash \sigma' \blacktriangleright \boxed{\sigma_3 \to \sigma_4}$:

  - Case UBREFL. Trivial.
  - Case UBBOX. Here we have that $\sigma' = \sigma_3 \to \sigma_4$ and we need to show that $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3 \to \sigma_4$, or using F2 that $\vdash \sigma_3 \sim \sigma_1'$ and $\vdash \sigma_2' \leq \sigma_4$. The former follows by (3) and Corollary 3.12. The latter follows from (4) and induction hypothesis, since $\vdash \sigma_4 \blacktriangleright \boxed{\sigma_4}$.
  - Case UBFUNBOX. In this case we have that $\sigma' = \boxed{\sigma_3} \to \boxed{\sigma_4}$ and we have the result directly from equation (2).

- Case F2. In this case $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3' \to \sigma_4'$ given that

$$\vdash \sigma_3' \sim \sigma_1' \tag{5}$$
$$\vdash \sigma_2' \leq \sigma_4' \tag{6}$$

  We have the following cases for $\vdash \sigma' \blacktriangleright \sigma_3' \to \sigma_4'$:

  - Case UBREFL. Trivial.
  - Case UBFUN. Here $\sigma' = \sigma_3'' \to \sigma_4''$ with

$$\vdash \sigma_3'' \blacktriangleright \sigma_3' \tag{7}$$
$$\vdash \sigma_4'' \blacktriangleright \sigma_4' \tag{8}$$

    From (5) and (7), and Corollary 3.12 $\vdash \sigma_3'' \sim \sigma_1'$ and from (6) and (8) and induction hypothesis $\vdash \sigma_2' \leq \sigma_4'$. Applying rule F2 finishes the case.

□

Corollary 3.29 (Uncontrolled unboxing for subsumption). *If* $\vdash \sigma_1' \leq \sigma_2'$ *and* $\vdash \sigma_3' \blacktriangleright \sigma_1'$ *and* $\vdash \sigma_4' \blacktriangleright \sigma_2'$ *then* $\vdash \sigma_3' \leq \sigma_4'$.

Proof. Follows by Lemma 3.27 and Lemma 3.28 and the fact that $(\blacktriangleright)$ is reflexive. □

The next theorem asserts that subsumption behaves the same for classes of $(\triangleright)$-equivalent types.

Theorem 3.30. *If* $\vdash \sigma_1' \leq \sigma_2'$ *then* $\vdash [\sigma_1']_{\triangleright} \leq [\sigma_2]_{\triangleright}$.

Proof. Direct consequence of Corollary 3.26 and Corollary 3.29 and the observation that $(\triangleright) \subseteq (\blacktriangleright)$. □

Lemma 3.31. $\vdash \sigma' \leq strip(\sigma')$.

Proof. By induction on the structure of σ'. We proceed by case analysis on σ'.

- Case $\sigma' = \forall \overline{a}.\rho'$. We need to show that $\vdash \forall \overline{a}.\rho' \leq \forall \overline{a}.strip(\rho')$. We have by induction hypothesis that $\vdash \rho' \leq strip(\rho')$ and by Corollary 3.26 we get $\vdash [\overline{a \mapsto \boxed{a}}]\rho' \leq strip(\rho')$. Then by SPEC we get $\vdash \forall \overline{a}.\rho' \leq strip(\rho')$ and the result follows by applying rule SKOL.
- Case $\sigma' = \boxed{\sigma}$. We need to show that $\vdash \boxed{\sigma} \leq \sigma$, or by SBOXY, that $\vdash \boxed{\sigma} \sim \sigma$, which holds by Lemma 3.15.

- Case $\sigma' = \sigma'_1 \to \sigma'_2$. We need to show that $\vdash \sigma'_1 \to \sigma'_2 \leq \mathit{strip}(\sigma'_1) \to \mathit{strip}(\sigma'_2)$ or equivalently that $\vdash \mathit{strip}(\sigma'_1) \sim \sigma'_1$ and $\vdash \sigma'_2 \leq \mathit{strip}(\sigma'_2)$. The former follows by rule SYM and Lemma 3.15 and the latter by induction hypothesis.
- Case $\sigma' = T\ \overline{\sigma}$. Follows by Lemma 3.15 and rule CON.
- Case $\sigma' = \tau$. Follows by rule MONO.

□

Corollary 3.32 (Reflexivity on box-free types for subsumption). $\vdash \sigma \leq \sigma$.

Proof. Just observe that $\mathit{strip}(\sigma) = \sigma$ and the result follows from Lemma 3.31. □

3.4 Strange boxing relation

This relation will be useful when we prove that we can embedd System-F in our language, provided we supply extra annotations on type lambdas and type applications. Here are its properties.

Lemma 3.33. *If* $\vdash \tau \succ \sigma'$ *then* $\vdash \tau \leq \sigma'$.

Proof. By induction on the derivation $\vdash \tau \succ \sigma'$. The case for SB1 is trivial. The case for SB2 cannot happen. For SB3 we have that $\tau = \tau_1 \to \tau_2$ and $\vdash \tau_1 \to \tau_2 \succ \boxed{\tau_1} \to \sigma'_2$ where $\vdash \tau_2 \succ \sigma'_2$. By induction $\vdash \tau_2 \leq \sigma'_2$ and $\vdash \boxed{\tau_1} \sim \tau_1$. Applying rule F2 gives the result. The case for SB4 is similar. □

Lemma 3.34 (Strange boxing preserves subsumption). *If* $\vdash \sigma_1 \leq \sigma'_2$ *end* $\vdash \sigma'_2 \succ \sigma''_2$ *then* $\vdash \sigma_1 \leq \sigma''_2$.

Proof. By induction on the derivation $\vdash \sigma \leq \rho$. We proceed with case analysis on the last rule used.

- Case SBOXY. Cannot happen as σ_1 is box-free type.
- Case MONO. In this case we have that $\vdash \tau \leq \tau$ and $\vdash \tau \succ \sigma'$ therefore by Lemma 3.33 $\vdash \tau \leq \sigma'$.
- Case BMONO. The only applicable rule would be SB1 and the result follows trivially.
- Case CON. We have that $\vdash T\ \overline{\sigma} \leq \sigma_2$ when $\vdash T\ \overline{\sigma} \sim \sigma_2$, therefore it must be that $\sigma_2 = T\ \overline{\sigma}_2$ by an easy inversion. Then the only rule applicable is SB1 and the result follows trivially.
- Case SKOL. If SB1 was used the result follows trivially; otherwise SB2 must have been used in which case the result follows from the premises of SKOL.
- Case SPEC. Directly follows by induction hypothesis and rule SPEC.
- Case F1. The only applicable rule would be SB1 and the result follows trivially.
- Case F2. We have that $\vdash \sigma_1 \to \sigma_2 \leq \sigma_3 \to \sigma_4$ given that

$$\vdash \sigma_3 \sim \sigma_1 \tag{9}$$
$$\vdash \sigma_2 \leq \sigma_4 \tag{10}$$

  We have the following cases for $\vdash \sigma_3 \to \sigma_4 \succ \sigma'$.

  - Case SB1. Trivial.
  - Case SB3. In this case we have that $\vdash \sigma_4 \succ \sigma'_4$ and by induction applied to (10) we get $\vdash \sigma_2 \leq \sigma'_4$. We then need to show that $\vdash \boxed{\sigma_3} \sim \sigma_1$. But (9) implies that $\sigma_3 = \sigma_1$ and this follows by Lemma 3.15. Then we can apply F2 to get the result.

□

Lemma 3.35 (Controlled boxing for typing). *The following are true of the typing relation:*

1. *If* $\Gamma \vdash^{poly} t : \sigma'$ *and* $\vdash \sigma' \triangleright \sigma''$ *then* $\Gamma \vdash^{poly} t : \sigma''$.
2. *If* $\Gamma \vdash t : \rho'$ *and* $\vdash \rho' \triangleright \rho''$ *then* $\Gamma \vdash t : \rho''$.

Proof. We prove the two claims simultaneously by induction on the height of the derivations. For each part the induction hypothesis asserts both claims for derivations of smaller height.

Part 1: For the first part we have two cases to consider.

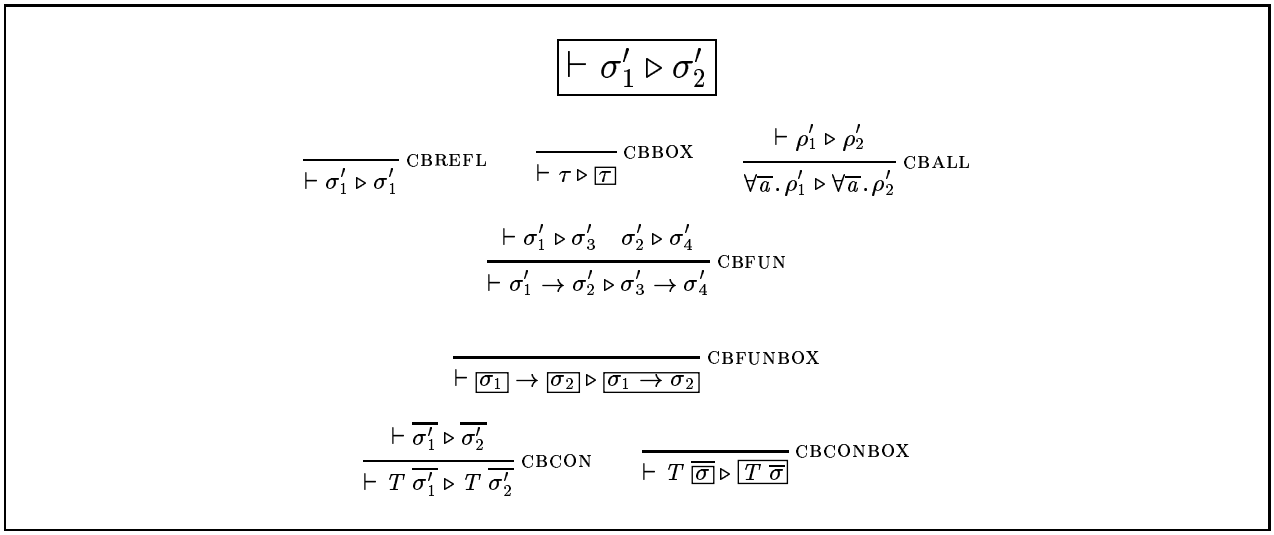

Figure 4: Controlled boxing of types

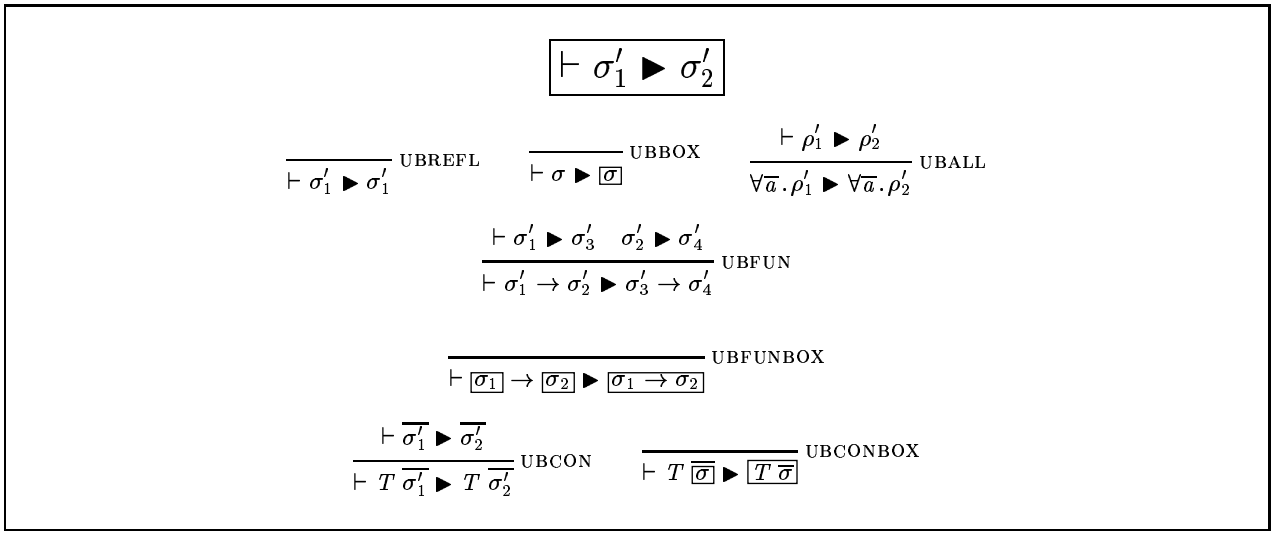

Figure 5: Uncontrolled boxing of types

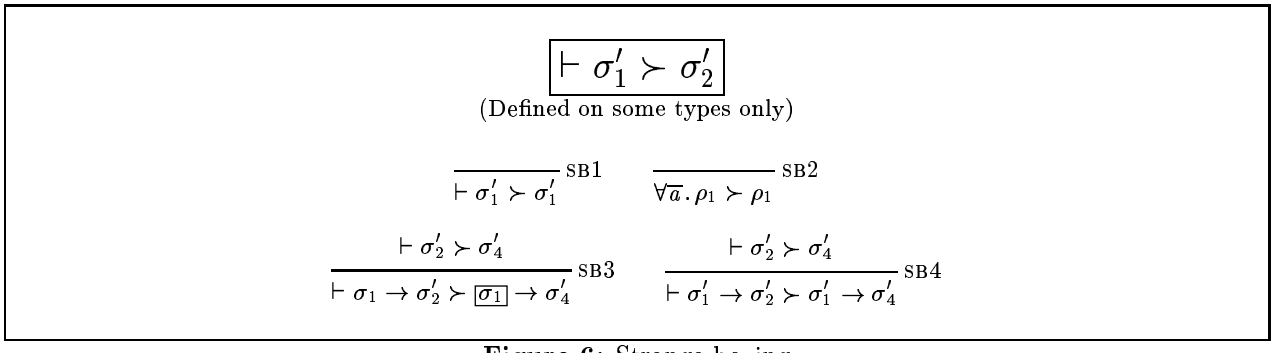

Figure 6: Strange boxing

- Case GEN1. We have that $\Gamma \vdash^{poly} t : \forall \overline{a}.\rho'$ given that $\Gamma \vdash t : \rho'$ and $\overline{a} \# ftv(\Gamma)$. We consider cases for $\vdash \forall \overline{a}.\rho' \triangleright \sigma'$. Either the rule CBREFL was used in which case we get the result trivially, or CBALL was used. In the latter case we have $\vdash \forall \overline{a}.\rho' \triangleright \forall \overline{a}.\rho''$ given that $\vdash \rho' \triangleright \rho''$. By induction hypothesis $\Gamma \vdash t : \rho''$ and by applying rule GEN1 we get the result again.
- Case GEN2. In this case we have that $\Gamma \vdash^{poly} t : \boxed{\rho}$ and it can only be that $\vdash \boxed{\rho} \triangleright \boxed{\rho}$ therefore we are trivially done.

Part 2: For the second part we have the following cases to consider.

- Case VAR. Follows directly be Corollary 3.26.
- Case ABS1. We have that $\Gamma \vdash \backslash x.t : \sigma_1' \to \sigma_2'$ given that

$$\vdash \sigma_1' \sim \boxed{\sigma_1} \tag{1}$$

$$\Gamma, x{:}\sigma_1 \vdash^{poly} t : \sigma_2' \tag{2}$$

  We consider cases for $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma'$:

  - Case CBREFL. Trivial.
  - Case CBBOX. In this case $\sigma_1' \to \sigma_2' = \tau_1 \to \tau_2$. We need to show that $\Gamma \vdash \backslash x.t : \boxed{\tau_1 \to \tau_2}$ or by applying ABS2, that $\Gamma \vdash \backslash x.t : \boxed{\tau_1} \to \boxed{\tau_2}$. By Corollary 3.26 and (1) we get $\vdash \boxed{\tau_1} \sim \boxed{\sigma_1}$ and by (2) and induction hypothesis we get $\Gamma, x{:}\sigma_1 \vdash^{poly} t : \boxed{\tau_2}$. Applying rule ABS1 finishes the case.
  - Case CBFUN. In this case we have that $\vdash \sigma_1' \to \sigma_2' \triangleright \sigma_1'' \to \sigma_2''$ given that $\vdash \sigma_1' \triangleright \sigma_1''$ and $\vdash \sigma_2' \triangleright \sigma_2''$. By Corollary 3.26 and (1) then we get that $\vdash \sigma_1'' \sim \boxed{\sigma_1}$. By (2) and induction hypothesis $\Gamma, x{:}\sigma_1 \vdash^{poly} t : \sigma_2''$. Applying rule ABS1 finishes the case.
  - Case CBFUNBOX. In this case $\sigma_1' = \boxed{\sigma_1}$ and $\sigma_2' = \boxed{\sigma_2}$ and $\sigma' = \boxed{\sigma_1 \to \sigma_2}$. We then need to show that $\Gamma \vdash \backslash x.t : \boxed{\sigma_1 \to \sigma_2}$ or using ABS2 that $\Gamma \vdash \backslash x.t : \boxed{\sigma_1} \to \boxed{\sigma_2}$ which we already have.
- Case ABS2. Only the rule CBREFL can be used so the result follows trivially.
- Case APP. We have that $\Gamma \vdash t\ u : \rho'$ given that

$$\Gamma \vdash t : \boxed{\sigma} \to \rho' \tag{3}$$

$$\Gamma \vdash^{poly} u : \sigma \tag{4}$$

  Take $\vdash \rho' \triangleright \rho''$, then $\vdash \boxed{\sigma} \to \rho' \triangleright \boxed{\sigma} \to \rho''$ and by induction hypothesis for (3) we get $\Gamma \vdash t : \boxed{\sigma} \to \rho''$. From this, (4) and rule APP again we get the result.
- Case LET. Easy unfolding and application of the induction hypothesis and rule LET.
- Case SIG-LET. Similar to the case for LET.

□

Lemma 3.36 (Uncontrolled unboxing for typing). *The following are true of the typing relation:*

1. *If* $\Gamma \vdash^{poly} t : \sigma'$ *and* $\vdash \sigma'' \blacktriangleright \sigma'$ *then* $\Gamma \vdash^{poly} t : \sigma''$.
2. *If* $\Gamma \vdash t : \rho'$ *and* $\vdash \rho'' \blacktriangleright \rho'$ *then* $\Gamma \vdash t : \rho''$.

Proof. We prove the two claims simultaneously by induction on the height of the derivations. For each part the induction hypothesis asserts both claims for derivations of smaller height. We proceed with case analysis on the last rule used.

Part 1: For the first part we consider two cases.

- Case GEN1. We have that $\Gamma \vdash^{poly} t : \forall \overline{a}.\rho'$ given that $\Gamma \vdash t : \rho'$ and $\overline{a} \# ftv(\Gamma)$. We consider cases for $\vdash \sigma' \blacktriangleright \forall \overline{a}.\rho'$. Either the rule UBREFL was used in which case we get the result trivially, or UBALL was used. In the latter case we have $\vdash \forall \overline{a}.\rho'' \triangleright \forall \overline{a}.\rho'$ given that $\vdash \rho'' \triangleright \rho'$. By induction hypothesis $\Gamma \vdash t : \rho''$ and by applying rule GEN1 we get the result again.
- Case GEN2. We have that $\Gamma \vdash^{poly} t : \boxed{\rho}$ given that $\Gamma \vdash t : \boxed{\rho}$. By induction hypothesis, if $\vdash \rho' \blacktriangleright \boxed{\rho}$ then $\Gamma \vdash t : \rho'$ and by rule GEN1 with no generalised variables we get the result.

Part 2: For the second part we consider the following cases.

- Case VAR. Follows by Corollary 3.29.

- Case ABS1. We have that $\Gamma \vdash \backslash x\,.\,t : \sigma_1' \to \sigma_2'$ given that

$$\vdash \sigma_1' \sim \boxed{\sigma_1} \tag{1}$$

$$\Gamma, x{:}\sigma_1 \vdash^{poly} t : \sigma_2' \tag{2}$$

  We consider cases for $\vdash \sigma' \blacktriangleright \sigma_1' \to \sigma_2'$.

  - Case UBREFL. Trivial.
  - Case UBFUN. In this case we have that $\vdash \sigma_1'' \to \sigma_2'' \blacktriangleright \sigma_1' \to \sigma_2'$ such that $\vdash \sigma_1'' \blacktriangleright \sigma_1'$ and $\vdash \sigma_2'' \blacktriangleright \sigma_2'$. By Corollary 3.12 $\vdash \sigma_1'' \sim \boxed{\sigma_1}$ and by induction hypothesis $\Gamma, x{:}\sigma_1 \vdash^{poly} t : \sigma_2''$. Applying rule ABS1 finishes the case.

- Case ABS2. In this case we have that $\Gamma \vdash \backslash x\,.\,t : \boxed{\sigma_1 \to \sigma_2}$ given that

$$\Gamma \vdash \backslash x\,.\,t : \boxed{\sigma_1} \to \boxed{\sigma_2} \tag{3}$$

  We have the following cases for $\vdash \sigma' \blacktriangleright \boxed{\sigma_1 \to \sigma_2}$:

  - Case UBREFL. Trivial.
  - Case UBFUNBOX. Directly follows by (3).
  - Case UBBOX. In this case we need to show that $\Gamma \vdash \backslash x\,.\,t : \sigma_1 \to \sigma_2$. But it is easy to confirm that $\vdash \sigma_1 \to \sigma_2 \blacktriangleright \boxed{\sigma_1} \to \boxed{\sigma_2}$ and the result follows by this and induction hypothesis for (3).

- Case APP. In this case we have that $\Gamma \vdash t\ u : \rho'$ given that

$$\Gamma \vdash t : \boxed{\sigma} \to \rho' \tag{4}$$

$$\Gamma \vdash^{poly} u : \sigma \tag{5}$$

  Take $\vdash \rho'' \blacktriangleright \rho'$, then $\vdash \boxed{\sigma} \to \rho'' \blacktriangleright \boxed{\sigma} \to \rho'$ and by induction hypothesis for (4) we get that $\Gamma \vdash t : \boxed{\sigma} \to \rho''$. From this, (5) and APP we get the result.

- Case LET. Easy unfolding and application of induction hypothesis and rule LET.
- Case SIG-LET. Similar to the case for LET.

□

Theorem 3.37 (Boxing-unboxing for typing). *If* $\Gamma \vdash t : \rho'$ *then* $\Gamma \vdash t : [\rho']_{\triangleright}$. *If* $\Gamma \vdash^{poly} t : \sigma'$ *then* $\Gamma \vdash^{poly} t : [\sigma']_{\triangleright}$.

Proof. Follows by Lemma 3.35 and Lemma 3.36 and the fact that $(\triangleright) \subseteq (\blacktriangleright)$. □

$$\boxed{\Gamma \vdash^F t : \sigma}$$

$$\frac{\nu{:}\sigma \in \Gamma}{\Gamma \vdash^F \nu : \sigma}\ \text{FVAR}$$

$$\frac{\Gamma, x{:}\sigma_1 \vdash^F t : \sigma_2 \quad \Gamma \vdash \sigma_1}{\Gamma \vdash^F \lambda x\,.\,t : \sigma_1 \to \sigma_2}\ \text{ABS} \qquad \frac{\Gamma \vdash^F t_1 : \sigma_1 \to \sigma_2 \quad \Gamma \vdash^F t_2 : \sigma_1}{\Gamma \vdash^F t_1\ t_2 : \sigma_2}\ \text{APP}$$

$$\frac{\Gamma, \overline{a} \vdash^F t : \sigma \quad \overline{a} \notin dom(\Gamma)}{\Gamma \vdash^F \Lambda\overline{a}\,.\,t : \forall\overline{a}\,.\,\rho}\ \text{TABS} \qquad \frac{\Gamma \vdash^F t : \forall\overline{a}\,.\,\rho \quad \Gamma \vdash \overline{\sigma}_1}{\Gamma \vdash^F t\ \overline{\sigma}_1 : [\overline{a \mapsto \sigma_1}]\rho}\ \text{TAPP}$$

Figure 7: System-F (à la Curry)

$$\boxed{\text{System-F translation } [\![t^F]\!]_\Gamma = t}$$

$$\begin{array}{lcll}
[\![x]\!]_\Gamma & = & x & \\
[\![\lambda x\,.\,t]\!]_\Gamma & = & (\backslash x\,.\,[\![t]\!]_{\Gamma,x:\tau_1}) & \text{where} \quad \Gamma, x{:}\tau_1 \vdash^F t : \sigma_2 \\
[\![\lambda x\,.\,t]\!]_\Gamma & = & (\backslash x\,.\,[\![t]\!]_{\Gamma,x:\sigma_1}) :: (\sigma_1 \to \sigma_2) & \text{where} \quad \Gamma, x{:}\sigma_1 \vdash^F t : \sigma_2 \\
[\![t_1\ t_2]\!]_\Gamma & = & [\![t_1]\!]_\Gamma\ [\![t_2]\!]_\Gamma & \\
[\![\Lambda\overline{a}\,.\,t]\!]_\Gamma & = & [\![t]\!]_\Gamma & \text{where } \overline{a}\#ftv(t) \\
[\![\Lambda\overline{a}\,.\,t]\!]_\Gamma & = & [\![t]\!]_{\Gamma,\overline{a}} :: \sigma & \text{where} \quad \Gamma \vdash^F \Lambda\overline{a}\,.\,t : \sigma \\
[\![t\ \overline{\tau}]\!]_\Gamma & = & [\![t]\!]_\Gamma & \\
[\![t\ \overline{\sigma}_1]\!]_\Gamma & = & [\![t]\!]_\Gamma :: \sigma & \text{where} \quad \Gamma \vdash^F t\ \overline{\sigma}_1 : \sigma
\end{array}$$

Figure 8: Translation of System-F

4 Translation of System-F

The System-F figure is given in Figure 7. Notice that this version of System-F is as expressive as the more familiar one where type abstractions and type applications occur one at a time. The results in this section can be made to work with the simpler version of System-F, but for convenience we present them for the multi-abstraction, multi-application version. Consider the following syntax for terms, denoted with r, s.

$$\begin{array}{lcl}
s, r & ::= & \nu \mid \lambda^\flat \mid \lambda^\sharp \mid s\ r \mid s :: \sigma \\
\lambda^\flat & ::= & \backslash x :: \tau\,.\,s \\
\lambda^\sharp & ::= & (\backslash x\,.\,s) :: (\sigma_1 \to \sigma_2)
\end{array}$$

Consider also the following translation of System-F, called *pre-translation*:

$$\begin{array}{lcl}
[\![x]\!]^\dagger_\Gamma & = & x \\
[\![\lambda x\,.\,t]\!]^\dagger_\Gamma & = & (\backslash x :: \tau_1\,.\,[\![t]\!]^\dagger_{\Gamma,x:\tau_1}) \\
 & & \text{where} \quad \Gamma, x{:}\tau_1 \vdash^F t : \sigma_2 \\
[\![\lambda x\,.\,t]\!]^\dagger_\Gamma & = & (\backslash x\,.\,[\![t]\!]^\dagger_{\Gamma,x:\sigma_1}) :: (\sigma_1 \to \sigma_2) \\
 & & \text{where} \quad \Gamma, x{:}\sigma_1 \vdash^F t : \sigma_2 \\
[\![t_1\ t_2]\!]^\dagger_\Gamma & = & [\![t_1]\!]^\dagger_\Gamma\ [\![t_2]\!]^\dagger_\Gamma \\
[\![\Lambda\overline{a}\,.\,t]\!]^\dagger_\Gamma & = & [\![t]\!]^\dagger_\Gamma \\
 & & \text{where } \overline{a}\#ftv(t) \\
[\![\Lambda\overline{a}\,.\,t]\!]^\dagger_\Gamma & = & [\![t]\!]^\dagger_{\Gamma,\overline{a}} :: \sigma \\
 & & \text{where} \quad \Gamma \vdash^F \Lambda\overline{a}\,.\,t : \sigma \\
[\![t\ \overline{\tau}]\!]^\dagger_\Gamma & = & [\![t]\!]^\dagger_\Gamma \\
[\![t\ \overline{\sigma}_1]\!]^\dagger_\Gamma & = & [\![t]\!]^\dagger_\Gamma :: \sigma \\
 & & \text{where} \quad \Gamma \vdash^F t\ \overline{\sigma}_1 : \sigma
\end{array}$$

It is an easy observation that if t is a System-F term, then $[\![t]\!]^{\dagger}_{\Gamma}$ yields an s-term.
To type s-terms we use VAR, and APP for variables and applications. We use the following rule for $s::\sigma$:

$$\frac{ftv(\forall\overline{a}.\rho) \subseteq dom(\Gamma) \quad \overline{a}\#ftv(\Gamma) \quad \Gamma, \overline{a} \vdash t : \rho \quad \vdash \forall\overline{a}.\rho \leq \rho'}{\Gamma \vdash (t::\forall\overline{a}.\rho) : \rho'}\ \text{SIG}$$

For $\lambda^{\sharp}$ we have the following rule:

$$\frac{ftv(\sigma_1 \to \sigma_2) \subseteq dom(\Gamma) \quad \Gamma, x{:}\sigma_1 \vdash^{poly} r : \sigma_2 \quad \vdash \sigma_1 \to \sigma_2 \leq \rho' \quad ftv(\sigma_1, \sigma_2) \subseteq dom(\Gamma)}{\Gamma \vdash (\backslash x.r)::(\sigma_1 \to \sigma_2) : \rho'}\ \text{ABS}^*$$

Finally, for $\lambda^{\flat}$ we use the following pair of rules:

$$\frac{ftv(\tau) \subseteq dom(\Gamma) \quad \vdash \sigma_1' \sim \boxed{\tau} \quad \Gamma, x{:}\tau \vdash^{poly} t : \sigma_2'}{\Gamma \vdash (\backslash x::\tau.t) : \sigma_1' \to \sigma_2'}\ \text{ABS1}^*$$

$$\frac{\Gamma \vdash (\backslash x::\tau.t) : \boxed{\sigma_1} \to \boxed{\sigma_2}}{\Gamma \vdash (\backslash x::\tau.t) : \boxed{\sigma_1 \to \sigma_2}}\ \text{ABS2}^*$$

The following lemma is then true.

Lemma 4.1. *If* $\vdash \Gamma \vdash s : \rho'$ *and* $\vdash \rho' \succ \rho''$ *then* $\vdash \Gamma \vdash s : \rho''$. *If* $\vdash \Gamma \vdash^{poly} s : \sigma'$ *and* $\vdash \sigma' \succ \sigma''$ *then* $\vdash \Gamma \vdash^{poly} s : \sigma''$.

Proof. By induction on the structure of s. We inline uses of the second part in the first, so we can apply induction hypothesis for both. We have the following cases.

- Case $s = \nu$. Directly follows by rule VAR and Lemma 3.34.
- Case $s = r_1\ r_2$. In this case we have that $\Gamma \vdash r_1\ r_2 : \rho'$ given that

$$\Gamma \vdash r_1 : \boxed{\sigma} \to \rho' \tag{1}$$
$$\Gamma \vdash^{poly} r_2 : \sigma \tag{2}$$

  By induction hypothesis for (1) $\Gamma \vdash r_1 : \boxed{\sigma} \to \rho''$ and using this and (2) and rule APP finishes the case.
- Case $s = r::\sigma$. In this case assume $\sigma = \forall\overline{a}.\rho$ and
  we have that $\overline{a}\#ftv(\Gamma)$, $ftv(\forall\overline{a}.\rho) \subseteq dom(\Gamma)$, $\Gamma, \overline{a} \vdash r : \rho$, and $\vdash \forall\overline{a}.\rho \leq \rho'$. Using Lemma 3.34 we get $\vdash \forall\overline{a}.\rho \leq \rho''$ and we can use the derived rule SIG to get the result.
- Case $s = (\backslash x.s)::(\sigma_1 \to \sigma_2)$. Similar to the case for the annotated term.
- Case $s = (\backslash x::\tau.s)$. In this case we have two subcases.
  - Rule ABS1* was used, in which case we have that

$$\Gamma \vdash (\backslash x::\tau.t) : \sigma_1' \to \sigma_2' \tag{3}$$

    given that

$$\vdash \sigma_1' \sim \boxed{\tau} \tag{4}$$
$$\Gamma, x{:}\tau \vdash^{poly} t : \sigma_2' \tag{5}$$

    Let us consider cases for $\vdash \sigma_1' \to \sigma_2' \succ \rho'$.
    - Case SB1. Trivial.
    - Case SB2. Cannot happen.
    - Case SB3. Here $\sigma_1' \to \sigma_2' = \sigma_1 \to \sigma_2'$ and $\rho' = \boxed{\sigma_1} \to \sigma_2'$. From (4) it must also be that $\vdash \boxed{\sigma_1} \sim \boxed{\tau}$ and by applying rule ABS1* we are done.

* Case SB4. Follows by induction hypothesis for (5) and rule ABS1*.

– Rule ABS2* was used. In this case after two steps of inversion a similar analysis as above applies.

The second part is straightforward. □

Theorem 4.2 (Arbitrary instantiation). $\vdash \forall \overline{a}.\rho \leq [\overline{a \mapsto \sigma}]\rho$.

Proof. By SPEC it is enough to show that $\vdash [\overline{a \mapsto \boxed{\sigma}}]\rho \leq [\overline{a \mapsto \sigma}]\rho$. But this follows from Lemma 3.31. □

Notice that in the theorem above our types are *sans-box*. A definition we need is the *weak prenex form* conversion.

$$
\begin{array}{lcll}
wpr(\forall \overline{a}.\rho) & = & \forall \overline{a}\overline{b}.\rho_1 & \text{where } wpr(\rho) = \forall \overline{b}.\rho_1 \\
 & & & \text{and } \overline{b} \# \overline{a} \\
wpr(\sigma_1 \to \sigma_2) & = & \sigma_1 \to \rho_2 & \text{where } wpr(\sigma_2) = \forall \overline{a}.\rho_2 \\
wpr(T\ \overline{\sigma}) & = & T\ \overline{\sigma} & \\
wpr(\tau) & = & \tau &
\end{array}
$$

Lemma 4.3 (Subsumption infers weak prenex form). *If* $wpr(\sigma) = \forall \overline{a}.\rho$ *then* $\vdash \sigma \leq \boxed{\rho}$.

Proof. By induction on the structure of σ.

- Case $\sigma = \forall \overline{a}.\rho_1$. Assume $wpr(\rho_1) = \forall \overline{b}.\rho_2$ such that without loss of generality $\overline{b} \# \overline{a}$. We then need to show that $\vdash \forall \overline{a}.\rho_1 \leq \boxed{\rho_2}$. But by induction hypothesis we have that $\vdash \rho_1 \leq \boxed{\rho_2}$ and by Corollary 3.26 we have $\vdash [\overline{a \mapsto \boxed{a}}]\rho_1 \leq \boxed{\rho_2}$. Then we can apply rule SPEC to finish the case.
- Case $\sigma = \sigma_1 \to \sigma_2$. Assume that $wpr(\sigma_2) = \forall \overline{b}.\rho_2$ and without loss of generality $\overline{b} \# ftv(\sigma)$. Then we need to show that $\vdash \sigma_1 \to \sigma_2 \leq \boxed{\sigma_1 \to \rho_2}$ or by rule F1 and rule F2 $\vdash \boxed{\sigma_1} \sim \sigma_1$ and $\vdash \sigma_2 \leq \boxed{\rho_2}$. The former follows by Lemma 3.15 and the latter by induction hypothesis.
- Case $\sigma = T\ \overline{\sigma}$. Here $wpr(T\ \overline{\sigma}) = T\ \overline{\sigma}$ and we need to show that $\vdash T\ \overline{\sigma} \sim \boxed{T\ \overline{\sigma}}$ which follows by CEQ1, CEQ2 and Lemma 3.15.
- Case $\sigma = \tau$. Trivially follows by BMONO since $wpr(\tau) = \tau$.

□

Lemma 4.4 (Polytype substitution for matching). *If* $\overline{a}$ *have unboxed occurrences in* σ_1', σ_2', *and* $\vdash \sigma_1' \sim \sigma_2'$ *then* $\vdash [\overline{a \mapsto \sigma}]\sigma_1' \sim [\overline{a \mapsto \sigma}]\sigma_2'$.

Proof. The proof is by induction on the height of the derivation $\vdash \sigma_1' \sim \sigma_2'$. We proceed by case analysis on the last rule used.

- Case SYM. By induction hypothesis and SYM again.
- Case AEQ1. By induction hypothesis and AEQ1 again.
- Case AEQ2. By induction hypothesis and AEQ2 again.
- Case CEQ1. By induction hypothesis and CEQ1 again.
- Case CEQ2. By induction hypothesis and CEQ2 again.
- Case BBEQ. Trivially follows as all occurrences of $\overline{a}$ are guarded under a box.
- Case MEQ1. Follows by Lemma 3.15 and SYM.
- Case MEQ2. Follows by reflexivity for box-free types, Lemma 3.13.
- Case SEQ2. For SEQ2 we have that $\vdash \forall \overline{c}.\rho_1' \sim \forall \overline{c}.\rho_2'$ given that $\vdash \rho_1' \sim \rho_2'$. By the (monotype) substitution lemma for matching, Lemma 3.1 we get that $\vdash [\overline{c \mapsto b}]\rho_1' \sim [\overline{c \mapsto b}]\rho_2'$ with the same height, where $\overline{b} \# vars(\overline{\sigma}, \overline{a})$ Then by induction hypothesis $\vdash [\overline{a \mapsto \sigma}][\overline{c \mapsto b}]\rho_1' \sim [\overline{a \mapsto \sigma}][\overline{c \mapsto b}]\rho_2'$ and by SEQ2 $\vdash \forall \overline{b}.[\overline{a \mapsto \sigma}][\overline{c \mapsto b}]\rho_1' \sim \forall \overline{b}.[\overline{a \mapsto \sigma}][\overline{c \mapsto b}]\rho_2'$, or equivalently $\vdash [\overline{a \mapsto \sigma}]\forall \overline{b}.[\overline{c \mapsto b}]\rho_1' \sim [\overline{a \mapsto \sigma}]\forall \overline{b}.[\overline{c \mapsto b}]\rho_2'$, or $\vdash [\overline{a \mapsto \sigma}]\forall \overline{c}.\rho_1' \sim [\overline{a \mapsto \sigma}]\forall \overline{c}.\rho_2'$ as required.
- Case SEQ1. Similar to the case for SEQ2.

□

Lemma 4.5 (Polytype substitution for subsumption). *Assume that the following three conditions hold:*

1. $\overline{a}$ *have unboxed occurrences in* σ_1', σ_2'.
2. *All occurrences of* $\overline{a}$ *in* σ_2' *are unboxed.*

3. $\vdash \sigma_1' \leq \sigma_2'$.

Then $\vdash [\overline{a \mapsto \sigma}]\sigma_1' \leq [\overline{a \mapsto \sigma}]\sigma_2'$.

Proof. By induction on the height of the derivation $\vdash \sigma_1' \leq \sigma_2'$. We proceed by case analysis on the last rule used.

- Case SBOXY. Follows by Lemma 4.4.
- Case MONO. Follows by reflexivity for box-free types, Corollary 3.32.
- Case BMONO. Trivially follows since all occurrences of $\overline{a}$ in the right-hand side type are guarded under a box.
- Case CON. Follows by Lemma 4.4.
- Case SKOL. In this case we have that $\vdash \sigma_1' \leq \forall \overline{b}\,.\,\rho_2'$, given that

$$\sigma_1' \neq \boxed{\sigma} \quad \overline{b} \# ftv(\sigma_1') \tag{1}$$
$$\vdash \sigma_1' \leq \rho_2' \tag{2}$$

  Consider $\overline{c} \# ftv(\sigma_1'), vars(\overline{\sigma}), \overline{a}$. Then by induction hypothesis for (2) we get $[\overline{a \mapsto \sigma}][\overline{b \mapsto c}]\sigma_1' \leq [\overline{a \mapsto \sigma}][\overline{b \mapsto c}]\rho_2'$, or equivalently using (1) $[\overline{a \mapsto \sigma}]\sigma_1' \leq [\overline{a \mapsto \sigma}][\overline{b \mapsto c}]\rho_2'$ and by applying rule SKOL $[\overline{a \mapsto \sigma}]\sigma_1' \leq \forall \overline{c}\,.\,[\overline{a \mapsto \sigma}][\overline{b \mapsto c}]\rho_2'$; equivalently $[\overline{a \mapsto \sigma}]\sigma_1' \leq [\overline{a \mapsto \sigma}]\forall \overline{b}\,.\,\rho_2'$ as required.
- Case SPEC. Let us name the substitution $\psi = [\overline{a \mapsto \sigma}]$. In this case we have that $\vdash \forall \overline{c}\,.\,\rho_1' \leq \rho_2'$ given that $\vdash [\overline{c \mapsto \boxed{\sigma}}]\rho_1' \leq \rho_2'$. Equivalently

$$\vdash [\overline{b \mapsto \boxed{\sigma}}][\overline{c \mapsto b}]\rho_1' \leq \rho_2' \tag{3}$$

  where we assume $\overline{b} \# vars(\psi), ftv(\overline{c}, \overline{\boxed{\sigma}})$, that is $\overline{b}$ completely fresh. By induction hypothesis we get $\vdash \psi[\overline{b \mapsto \boxed{\sigma}}][\overline{c \mapsto b}]\rho_1' \leq \psi(\rho_2')$ or by the freshness of $\overline{b}$ $\vdash [\overline{b \mapsto \boxed{\psi(\sigma)}}]\psi[\overline{c \mapsto b}]\rho_1' \leq \psi(\rho_2')$. By applying rule SPEC $\vdash \forall \overline{b}\,.\,\psi[\overline{c \mapsto b}]\rho_1' \leq \psi(\rho_2')$ or equivalently $\vdash \psi(\forall \overline{c}\,.\,\rho_1') \leq \psi(\rho_2')$ as required.
- Case F1. Trivially follows sinc $\overline{a}$ cannot be inside boxes in the right-hand side type.
- Case F2. Follows by induction hypothesis, Lemma 4.4 and application of rule F2 again.

□

Lemma 4.6 (Substitution for typing). *Assume without loss of generality that scoped variables can α-vary, so as to satisfy the condition that $dom(\Gamma) \# vars(S)$. The following are true for the typing relation.*

1. *If* $\Gamma \vdash t : \rho'$ *then* $S\Gamma \vdash t : S\rho'$.
2. *If* $\Gamma \vdash^{poly} t : \sigma'$ *then* $S\Gamma \vdash^{poly} t : S\sigma'$

Moreover in each case the new derivation has the same height.

Proof. The two claims are proved simultanously by induction on the height of the derivations.

For the first part, all cases but the LET case follow by by application of the induction hypothesis appealing to Lemma 3.4, Lemma 3.1. The only interesting case is the one for LET. Here we have that $\Gamma \vdash \texttt{let}\ x\ \texttt{=}\ u\ \texttt{in}\ t : \rho'$ given that

$$\Gamma \vdash u : \boxed{\rho} \tag{1}$$
$$\overline{a} = ftv(\rho) - ftv(\Gamma) \tag{2}$$
$$\Gamma, x{:}\forall \overline{a}\,.\,\rho \vdash t : \rho' \tag{3}$$

Consider $\overline{b} \# ftv(\Gamma, \rho), vars(S)$. Then by induction hypothesis $S[\overline{a \mapsto b}]\Gamma \vdash u : \boxed{S[\overline{a \mapsto b}]\rho}$, or $S\Gamma \vdash u : \boxed{S[\overline{a \mapsto b}]\rho}$. Then we claim that $\overline{b} = ftv(S[\overline{a \mapsto b}]\rho) - ftv(S\Gamma)$. For one direction assume that there exists a $b \in \overline{b}$ such that $b \notin ftv(S[\overline{a \mapsto b}]\rho) - ftv(S\Gamma)$. Then it must be that $b \in ftv(S\Gamma)$, a contradiction. Conversely assume that $g \in ftv(S[\overline{a \mapsto b}]\rho) - ftv(S\Gamma)$ but $g \notin \overline{b}$. Then it must be that $\overline{g} \notin \overline{a}$ otherwise $g \in \overline{b}$. But then there must exists $c \in ftv(\rho)$ such that $g \in ftv(Sc)$. Also $c \notin ftv(\Gamma)$ because otherwise $g \in ftv(S\Gamma)$. Then $c \in \overline{a}$, a contradiction.

For the second part the case for GEN2 follows by induction hypothesis and the case for GEN1 follows by induction hypothesis with an extra renaming substitution as in the case for rule SKOL in the the substitution lemma for subsumption. □

Lemma 4.7 (Polytype substitution for restricted typing). *If all positive occurrences of $\overline{a}$ in ρ' are unboxed and $\overline{a} \# ftv(\Gamma)$, and $\psi = [\overline{a \mapsto \sigma}]$ then:*

1. *If* $\Gamma \vdash r : \rho'$ *then* $\Gamma \vdash^{poly} r : \psi(\rho')$.

2. *If* $\Gamma \vdash^{poly} r : \sigma'$ *then* $\Gamma \vdash^{poly} r : \psi(\sigma')$.

(Notice that this is the syntactically restricted system.)

Proof. We prove the two claims simultaneously by induction on the structure of the term r. We inline the second part for which we assume that the first part always holds (notice that the usage of the second part always is for smaller terms inside the first part). For the first part we have the following cases.

- Case $r = \nu$. Directly follows by rule VAR and Lemma 4.5.
- Case $r = r_1\ r_2$. In this case we have that $\Gamma \vdash r_1\ r_2 : \rho'$ given that

$$\Gamma \vdash r_1 : \boxed{\sigma} \to \rho' \tag{1}$$
$$\Gamma \vdash^{poly} r_2 : \sigma \tag{2}$$

  By Lemma 3.36 and (1) we get $\Gamma \vdash r_1 : \sigma \to \rho'$. By induction hypothesis (r_1) is a sub-term of r. $\Gamma \vdash r_1 : \psi(\sigma) \to \psi(\rho')$ and by Lemma 4.1 $\Gamma \vdash r_1 : \boxed{\psi(\sigma)} \to \psi(\rho')$. Then by induction hypothesis for (2) we get $\Gamma \vdash^{poly} r_2 : \psi(\sigma)$ and by applying rule APP we are done.

- Case $r = s::\sigma$. In this case assume $\sigma = \forall \overline{a}\,.\,\rho$ and we have that $\overline{a} \# ftv(\Gamma)$, $ftv(\forall \overline{a}\,.\,\rho) \subseteq dom(\Gamma)$, $\Gamma, \overline{a} \vdash r : \rho$, and $\vdash \forall \overline{a}\,.\,\rho \leq \rho'$. Using Lemma 4.5 we get $\vdash \forall \overline{a}\,.\,\rho \leq \psi(\rho')$ and we can use the derived rule SIG to get the result.
- Case $r = (\backslash x\,.\,s)::(\sigma_1 \to \sigma_2)$. Similar to the case for the annotated term.
- Case $r = (\backslash x::\tau\,.\,s)$. Here we have two cases to consider.
  - Case ABS1*. In this case we have that

$$\Gamma \vdash (\backslash x::\tau\,.\,t) : \sigma_1' \to \sigma_2' \tag{3}$$

    given that

$$\vdash \sigma_1' \sim \boxed{\tau} \tag{4}$$
$$\Gamma, x{:}\tau \vdash^{poly} t : \sigma_2' \tag{5}$$
$$\tag{6}$$

    Moreover by (4) if must be that $ftv(\sigma_1') = ftv(\tau) \subseteq ftv(\Gamma)$, since they are scoped variables. Therefore, also all positive occurences of $\overline{a}$ in $\sigma_1' \to \sigma_2'$ are unboxed implies that all positive occurences in σ_2' are unboxed and by induction hypothesis for equation (5) and application of rule ABS1* again we are done.
  - Case ABS2*. Vacuously true.

For the second part we assume that the first always holds. We proceed with case analysis on the rule used.

- Case GEN1. We have that $\Gamma \vdash^{poly} r : \forall \overline{a}\,.\,\rho'$ given that $\Gamma \vdash r : \rho'$ and $\overline{a} \# ftv(\Gamma)$. Consider then a renaming substitution $[\overline{a \mapsto b}]$, with fresh $\overline{b}$, composed with our substitution and by the first part $\Gamma \vdash r : \psi[\overline{a \mapsto b}]\rho'$. Then $\overline{b} \# ftv(\Gamma)$ and by rule GEN1 again we are done.
- Case GEN2. Directly follows by the first part.

□

Theorem 4.8 (Translation of System-F to restricted system). *If* $\Gamma \vdash^F t : \sigma$ *and* $wpr(\sigma) = \forall \overline{a}\,.\,\rho$ *then*

1. $\Gamma \vdash^{poly} [\![t]\!]^{\dagger}_{\Gamma} : \sigma$ *in the restricted system.*
2. $\Gamma \vdash^{poly} [\![t]\!]^{\dagger}_{\Gamma} : \boxed{\rho}$ *in the restricted system.*

Proof. The proof is by induction on the derivation $\Gamma \vdash^F t : \sigma$ and we proceed with case analysis on the last rule used.

- Case VAR. Assume that $\sigma = \forall \overline{b}\,.\,\rho_0$. Then we have $\Gamma \vdash^F \nu : \sigma$ given that $\nu{:}\sigma \in \Gamma$. We need to show that $\Gamma \vdash^{poly} \nu : \sigma$ or by inversion that $\Gamma \vdash \nu : \rho_0$, or $\vdash \sigma \leq \rho_0$ but this follows because subsumption is reflexive for box-free types, that is $\vdash \sigma \leq \sigma$ and the result follows by inversion on rule SKOL. For the second part the result follows from Lemma 4.3.

- Case ABS. In this case we have that $\Gamma \vdash^F \lambda x\,.\,t : \sigma_1 \to \sigma_2$ given that

$$\Gamma, x{:}\sigma_1 \vdash^F t : \sigma_2 \tag{1}$$
$$\Gamma \vdash \sigma_1 \quad \text{that is } ftv(\sigma_1) \subseteq ftv(\Gamma) \tag{2}$$

  To see which translation rule applies we have the following cases:

  - $\sigma_1 \neq \tau_1$. By induction hypothesis for (1) we get

$$\Gamma, x{:}\sigma_1 \vdash [\![t]\!]^{\dagger}_{\Gamma, x:\sigma_1} : \sigma_2 \tag{3}$$

    Moreover $\vdash \sigma_1 \sim \boxed{\sigma_1}$ and from this we need to show the following two equations:

$$\Gamma \vdash ((\backslash x\,.\,[\![t]\!]^{\dagger}_{\Gamma}) :: (\sigma_1 \to \sigma_2)) : \sigma_1 \to \sigma_2 \tag{4}$$
$$\Gamma \vdash ((\backslash x\,.\,[\![t]\!]^{\dagger}_{\Gamma}) :: (\sigma_1 \to \sigma_2)) : \boxed{\sigma_1 \to \rho_2} \tag{5}$$

    where $wpr(\sigma_2) = \forall \overline{a}\,.\,\rho_2$. It is then enough using rule ABS* to show that

$$\vdash \sigma_1 \to \sigma_2 \leq \sigma_1 \to \sigma_2 \tag{6}$$
$$\vdash \sigma_1 \to \sigma_2 \leq \boxed{\sigma_1 \to \rho_2} \tag{7}$$

    But the first follows from Corollary 3.32 and the second from Lemma 4.3.

  - $\sigma_1 = \tau_1$. By induction hypothesis in this case we get that

$$\Gamma, x{:}\tau_1 \vdash [\![t]\!]^{\dagger}_{\Gamma, x:\tau_1} : \sigma_2 \tag{8}$$
$$\Gamma, x{:}\tau_1 \vdash [\![t]\!]^{\dagger}_{\Gamma, x:\tau_1} : \boxed{\rho_2} \tag{9}$$

    Obviously $\vdash \tau_1 \sim \boxed{\tau_1}$, and $\vdash \boxed{\tau_1} \sim \boxed{\tau_1}$, therefore by rule ABS1* we get that

$$\Gamma \vdash \backslash x :: \tau\,.\,[\![t]\!]^{\dagger}_{\Gamma, x:\tau_1} : \tau_1 \to \sigma_2 \tag{10}$$
$$\Gamma \vdash \backslash x :: \tau\,.\,[\![t]\!]^{\dagger}_{\Gamma, x:\tau_1} : \boxed{\tau_1} \to \boxed{\rho_2} \tag{11}$$

    Equation (10) gives the first part, and equation (11) with an application of ABS2* gives the second.

- Case APP. We have that $\Gamma \vdash^F t_1\ t_2 : \sigma_2$ given that

$$\Gamma \vdash^F t_1 : \sigma_1 \to \sigma_2 \tag{12}$$
$$\Gamma \vdash^F t_2 : \sigma_1 \tag{13}$$

  By induction hypothesis for (12) and (13) we get

$$\Gamma \vdash [\![t_1]\!]^{\dagger}_{\Gamma} : \sigma_1 \to \sigma_2 \tag{14}$$
$$\Gamma \vdash [\![t_2]\!]^{\dagger}_{\Gamma} : \sigma_1 \tag{15}$$
$$\Gamma \vdash [\![t_1]\!]^{\dagger}_{\Gamma} : \boxed{\sigma_1 \to \rho_2} \tag{16}$$

  where $wpr(\sigma_2) = \forall \overline{a}\,.\,\rho_2$. Then by Lemma 3.36 and (16) we get $\Gamma \vdash [\![t_1]\!]^{\dagger}_{\Gamma} : \boxed{\sigma_1} \to \boxed{\rho_2}$ and from this, equation (13) and rule APP we get the second part. Moreover assume that $\sigma_2 = \forall \overline{b}\,.\,\rho_2^0$. Then $\vdash \sigma_1 \to \sigma_2 \succ \boxed{\sigma_1} \to \rho_2^0$ and by Lemma 4.1 on (14) we get $\Gamma \vdash [\![t_1]\!]^{\dagger}_{\Gamma} : \boxed{\sigma_1} \to \rho_2^0$. Then, we can apply rule APP to get that $\Gamma \vdash [\![t_1\ t_2]\!]^{\dagger}_{\Gamma} : \rho_2^0$ and by applying rule GEN1 we are done.

- Case TABS. In this case we have that $\Gamma \vdash^F \Lambda\overline{a}\,.\,t : \forall\overline{a}\,.\,\rho$ given that $\Gamma, \overline{a} \vdash t : \rho$, $\overline{a} \# ftv(\Gamma)$. Assume that $\overline{a} \in ftv(t)$. We only show the first claim, the second is by essentialy a similar argument. By induction we get that $\Gamma, \overline{a} \vdash^{poly} [\![t]\!]^{\dagger}_{\Gamma,\overline{a}} : \rho$ or by using GEN1 and the derived rule SIG $\Gamma \vdash [\![t]\!]^{\dagger}_{\Gamma,\overline{a}} :: (\forall\overline{a}\,.\,\rho) : \rho$ and by using GEN1 again we are done. If on the other hand $\overline{a} \notin ftv(t)$ the result follows by System-F weakening and the fact that it preserves heights of derivations and application of induction hypothesis.

- Case TAPP. In this case we have that $\Gamma \vdash^F t\ \overline{\sigma} : [\overline{a \mapsto \sigma}]\rho$ given that

$$\Gamma \vdash^F t : \forall\overline{a}\,.\,\rho \tag{17}$$

  and $\Gamma \vdash \overline{\sigma}$. We have two cases to consider.

– $\overline{\sigma} \neq \overline{\tau}$. By induction hypothesis $\Gamma \vdash^{poly} [\![t]\!]^{\dagger}_{\Gamma} : \forall \overline{a}.\rho$ and assuming that $\overline{a} \# ftv(\Gamma)$ (which we can get with one more step of inversion on (17)) we get $\Gamma \vdash [\![t]\!]^{\dagger}_{\Gamma} : \rho$. Then by Lemma 4.7 we get that

$$\Gamma \vdash [\![t]\!]^{\dagger}_{\Gamma} : [\overline{a \mapsto \sigma}]\rho \tag{18}$$

Then we need to show that $\Gamma \vdash^{poly} [\![t]\!]^{\dagger}_{\Gamma} :: ([\![[\overline{a \mapsto \sigma}]\rho]\!]^{\dagger}_{\Gamma}) : [\overline{a \mapsto \sigma}]\rho$ and this easily follows from (18) and the derived rule SIG since subsumption is reflexive for box-free types. Similarly the second part follows from equation (18), an extra skolemisation step, and Lemma 4.3.

– $\sigma = \overline{\tau}$. Similar to the analysis above we get $\Gamma \vdash [\![t]\!]^{\dagger}_{\Gamma} : \rho$ and by an extension of the substitution lemma for the restricted system we have $\Gamma \vdash [\![t]\!]^{\dagger}_{\Gamma} : [\overline{a \mapsto \tau}]\rho$. Moreover by induction hypothesis we geta $\Gamma \vdash [\![t]\!]^{\dagger}_{\Gamma} : \boxed{\rho_0}$ where $wpr(\rho) = \forall \overline{ca}.\rho_0$ (notice that we did not rename the $\overline{a}$ as they are considered fresh already) and by the type substitution lemma we are done.

□

So far we have established that every well typed System-F term translates to a well typed s-term. Consider now the following translation of s-terms to normal terms, which merely removes the τ annotations from abstractions:

$$\begin{array}{lcl}
\lceil \nu \rceil & = & \nu \\
\lceil s\ r \rceil & = & \lceil s \rceil\ \lceil r \rceil \\
\lceil \backslash x :: \tau\ .\ s \rceil & = & \backslash x\,.\lceil s \rceil \\
\lceil \backslash x\ .\ s :: (\sigma_1 \to \sigma_2) \rceil & = & \backslash x\,.\lceil s \rceil :: (\sigma_1 \to \sigma_2) \\
\lceil s :: \sigma \rceil = \lceil s \rceil :: \sigma &&
\end{array}$$

It is easy to see that this translation, composed with weak translation yields the original translation of System-F.

Corollary 4.9. $[\![\cdot]\!]_{\Gamma} = \lceil [\![\cdot]\!]^{\dagger}_{\Gamma} \rceil$.

Proof. Directly follows from the definitions. □

Moreover, if a restricted term was typable in the restricted system, its stripping-off the monotype annotations will be in the **original system** (where, of course, we treat user type annotations as derived forms, using as well the derived typing rule SIG).

Lemma 4.10. *The following are true:*

1. *If* $\Gamma \vdash r : \rho'$ *then* $\Gamma \vdash \lceil r \rceil : \rho'$.
2. *If* $\Gamma \vdash^{poly} r : \sigma'$ *then* $\Gamma \vdash \lceil r \rceil : \sigma'$.

Proof. Easy induction on the height of the derivations of the restricted system. □

Theorem 4.11 (Translation of System-F). *If* $\Gamma \vdash^{F} t : \sigma$ *and* $wpr(\sigma) = \forall \overline{a}.\rho$ *then*

1. $\Gamma \vdash^{poly} [\![t]\!]_{\Gamma} : \sigma$.
2. $\Gamma \vdash^{poly} [\![t]\!]_{\Gamma} : \boxed{\rho}$.

Proof. Follows directly from Theorem 4.8, Corollary 4.9, and Lemma 4.10. □

5 Translation to System-F and type safety

In this section we give a type-preserving translation to System-F. We define the operational semantics of a translated System-F term to be the operational semantics of the source language, using therefore System-F as the "core" execution language associated with our "surface" language. Type safety of the source language follows then by progress and subject reduction for all well-typed System-F terms.

First of all it is easy to see that if $\vdash \sigma'_1 \sim \sigma'_2$ then $\mathit{strip}(\sigma'_1) = \mathit{strip}(\sigma'_2)$. Therefore there is no retyping induced by boxy matching. Subsumption caused by boxy matching merely yields identity retyping functions.

Lemma 5.1 (Retyping functions). *If* $\vdash \sigma'_1 \leq \sigma'_2 \rightsquigarrow f$ *then* $\vdash^F f : \mathit{strip}(\sigma'_1 \to \sigma'_2)$.

Proof. Easy induction. □

Lemma 5.2 (Term translation). *If* $\Gamma \vdash t : \rho' \rightsquigarrow t'$ *then* $\Gamma \vdash^F t' : \mathit{strip}(\rho')$. *If* $\Gamma \vdash^{poly} t : \sigma' \rightsquigarrow t'$ *then* $\Gamma \vdash^F t' : \mathit{strip}(\sigma')$.

Proof. Easy induction appealing to Lemma 5.1. □

6 Weakening lemmas

We define the ML "shallow subsumption" relation with the following rule.

$$\frac{\overline{b} \# ftv(\forall \overline{a}.\rho)}{\vdash \forall \overline{a}.\rho \leq_{sh} \forall \overline{b}.[\overline{a \mapsto \tau}]\rho}\ \textsc{shsubs}$$

Lemma 6.1. *If* $\vdash \sigma_1 \leq_{sh} \sigma_2$ *then* $\vdash \sigma_1 \leq \sigma_2$.

Proof. Let $\sigma_1 = \forall \overline{a}.\rho$ and $\sigma_2 = \forall \overline{b}.[\overline{a \mapsto \tau}]\rho$. Then $\vdash \sigma_1 \leq_{sh} \sigma_2$ given that $\overline{b} \# ftv(\sigma_1)$. If we apply rule SKOL it is enough to show that $\vdash \forall \overline{a}.\rho \leq [\overline{a \to \tau}]\rho$. By SPEC it is enough to show that $\vdash [\overline{a \mapsto \boxed{\tau}}]\rho \leq [\overline{a \mapsto \tau}]\rho$. But this follows from Lemma 3.31 since $strip([\overline{a \mapsto \boxed{\tau}}]\rho) = [\overline{a \mapsto \tau}]\rho$. □

Lemma 6.2. *If* $\forall \overline{a}.\tau \leq \tau_1$ *then* $\forall \overline{a}.\tau \leq_{sh} \tau_1$.

Proof. By inversion it must be that $[\overline{a \mapsto \boxed{\sigma}}]\tau \leq \tau_1$. By Theorem 3.30 it must also be that $\boxed{[\overline{a \mapsto \sigma}]\tau} \leq \tau_1$ which means that $\boxed{[\overline{a \mapsto \sigma}]\tau} \sim \tau_1$, consequently it must be that $\overline{\sigma} = \overline{\tau}$. Then we are done by applying Theorem 3.30 and rule SHSUBS. □

Lemma 6.3. *If* $\vdash \sigma_1 \leq_{sh} \sigma_2$ *then* $ftv(\sigma_1) \subseteq ftv(\sigma_2)$.

Proof. Immediate. □

Lemma 6.4. *The following are true of* $(\sim)$.

1. *If* $\vdash \sigma' \sim \tau$ *then* $\vdash \tau \triangleright^* \sigma'$.
2. *If* $\vdash \tau \sim \sigma'$ *then* $\vdash \tau \triangleright^* \sigma'$.

Where $(\triangleright^*)$ *is the transitive closure of* $(\triangleright)$.

Proof. We prove the two claims simultaneously by induction on the derivations. For each claim the induction hypothesis asserts both claims for derivations of smaller height. For the first part we have to consider the following cases:

- Case SYM. Follows by induction hypothesis for the second claim.
- Case AEQ2. Follows by induction hypothesis and rule CBFUN.
- Case CEQ2. Similar to the case for AEQ2.
- Case MEQ1. Can't happen.
- Case MEQ2. Follows by rule CBREFL.
- The rest of the cases cannot happen.

For the second part we have the following cases:

- Case SYM. Follows by induction hypothesis for the first claim.
- Case AEQ1. We have that $\vdash \tau_1 \to \tau_2 \leq \boxed{\sigma_1 \to \sigma_2}$ given that $\vdash \tau_1 \to \tau_2 \leq \boxed{\sigma_1} \to \boxed{\sigma_2}$. By induction we know that $\vdash \tau_1 \to \tau_2 \triangleright^* \boxed{\sigma_1} \to \boxed{\sigma_2}$ and by rule CBFUNBOX we have $\vdash \boxed{\sigma_1} \to \boxed{\sigma_2} \triangleright \boxed{\sigma_1 \to \sigma_2}$, therefore $\vdash \tau_1 \to \tau_2 \triangleright^* \boxed{\sigma_1 \to \sigma_2}$ as required.
- Case AEQ2. Follows by induction hypothesis and rule CBFUN again.
- Case CEQ1. Similar to the case for AEQ1.
- Case CEQ2. Similar to the case for rule AEQ2.
- Case MEQ1. Here $\sigma' = \boxed{\tau}$ and the result follows by rule CBBOX.
- Case MEQ2. Follows by rule CBREFL.
- The rest of the cases cannot happen.

□

Theorem 6.5 (Weakening for subsumption). *If* $\vdash \sigma_1 \leq \sigma_2'$ *and* $\vdash \sigma_0 \leq_{sh} \sigma_1$ *then* $\vdash \sigma_0 \leq \sigma_2'$[1].

[1] Notice that in the theorem statement σ_1 and σ_2 are box-free while σ_2' is an arbitrary polytype.

Proof. The proof is by induction on the derivation $\vdash \sigma_1 \leq \sigma_2'$. We proceed with case analysis on the last rule used in the derivation.

- Case SBOXY. Cannot happen.
- Case MONO. In this case we have that $\vdash \tau \leq \tau$ and $\vdash \sigma_0 \leq_{sh} \tau$. Applying Lemma 6.1 gives $\vdash \sigma_0 \leq \tau$ as required.
- Case BMONO. Here we have that $\vdash \tau \leq \boxed{\tau}$ and $\vdash \sigma_0 \leq_{sh} \tau$. Applying Lemma 6.1 gives $\vdash \sigma_0 \leq \tau$ and with Corollary 3.26 we get $\vdash \sigma_0 \leq \boxed{\tau}$ as required.
- Case SKOL. We have that $\vdash \sigma_1 \leq \forall \overline{b}\,.\,\rho_2'$ such that $\overline{b} \# ftv(\sigma_1)$, $\vdash \sigma_1 \leq rho_2'$. By induction hypothesis $\vdash \sigma_0 \leq \rho_2'$ and by Lemma 6.3 $ftv(\sigma_0) \subseteq ftv(\sigma_1)$ therefore $\overline{b} \# ftv(\sigma_0)$. Applying rule SKOL finishes the case.
- Case SPEC. In this case $\vdash \forall \overline{a}\,.\,\rho_1 \leq \rho_2'$ given that

$$\vdash [\overline{a \mapsto \boxed{\sigma}}]\rho_1 \leq \rho_2' \tag{1}$$

  Assume that $\sigma_0 = \forall \overline{c}\,.\,\rho_0$. Then it is the case that $\rho_1 = [\overline{c \mapsto \tau}]\rho_0$, $\overline{a} \# ftv(\sigma_0)$, and (1) becomes

$$\vdash [\overline{a \mapsto \boxed{\sigma}}][\overline{c \mapsto \tau}]\rho_0 \leq \rho_2' \tag{2}$$

  Let us give names to the mappings above:

$$\theta = [\overline{a \mapsto \boxed{\sigma}}] \quad \theta^{strip} = [\overline{a \mapsto \sigma}] \quad \psi = [\overline{c \mapsto \tau}]$$

  Consider the mapping $\phi = [\overline{c \mapsto \boxed{\sigma_c}}]$ defined as follows:

$$\phi(c) = \boxed{\sigma_c} \Leftrightarrow \sigma_c = \theta^{strip}(\psi(c))$$

  Then from (2) and Corollary 3.26 we get

$$\vdash [\overline{c \mapsto \boxed{\sigma_c}}]\rho_0 \leq \rho_2' \tag{3}$$

  Finally we can apply rule SPEC to get the result.
- Case F1. We have that $\vdash \sigma_1 \to \sigma_2 \leq \boxed{\sigma_3 \to \sigma_4}$ given that $\vdash \sigma_1 \to \sigma_2 \leq \boxed{\sigma_3} \to \boxed{\sigma_4}$. Assume that $\sigma_0 = \forall \overline{a}\,.\,\rho_0$. Then it is the case that $\sigma_1 \to \sigma_2 = [\overline{a \mapsto \tau}]\rho$, so we have that

$$\vdash [\overline{a \mapsto \tau}]\rho \leq \boxed{\sigma_3 \to \sigma_4} \tag{4}$$

  By Corollary 3.26 we get

$$\vdash [\overline{a \mapsto \boxed{\tau}}]\rho \leq \boxed{\sigma_3 \to \sigma_4} \tag{5}$$

  and by application of rule SPEC we are done.
- Case F2. In this case we have $\vdash \sigma_1' \to \sigma_2' \leq \sigma_3' \to \sigma_4'$. Assume that $\sigma_0 = \forall \overline{a}\,.\,\rho_0$ such that $[\overline{a \mapsto \tau}]\rho_0 = \sigma_1' \to \sigma_2'$. Then we have that $\vdash [\overline{a \mapsto \tau}]\rho_0 \leq \sigma_3' \to \sigma_4'$, and by Corollary 3.26 $\vdash [\overline{a \mapsto \boxed{\tau}}]\rho_0 \leq \sigma_3' \to \sigma_4'$. Then we can just apply the rule SPEC and to finish the case.
- Case CON. Similar to the case for F2.

□

Theorem 6.6 (Weakening for typing). *The following are true:*

1. *If* $\Gamma_1 \vdash t : \rho'$ *and* $\vdash \Gamma_2 \leq_{sh} \Gamma_1$ *then* $\Gamma_2 \vdash t : \rho'$.
2. *If* $\Gamma_1 \vdash^{poly} t : \sigma'$ *and* $\vdash \Gamma_2 \leq_{sh} \Gamma_1$ *then* $\Gamma_2 \vdash^{poly} t : \sigma'$.

Proof. The two claims are proved simultaneously by induction on the height of the derivations. The only interesting case is the case for VAR where we appeal to Theorem 6.5. For the second part and the SIG-LET case we also appeal to Lemma 6.3. □

— Extended Environments —

$$\begin{array}{lcll} \Gamma & ::= & \epsilon & \text{Empty environment} \\ & | & \Gamma, (x \overset{s}{:} \sigma) & \text{Rigid term binding} \\ & | & \Gamma, (x \overset{i}{:} \sigma) & \text{Inferred term binding} \end{array}$$

Figure 9: Environment syntax for full system

$$\boxed{\Gamma \vdash t : \rho'}$$

$$\frac{x \overset{i}{:} \sigma \in \Gamma \quad \vdash \sigma \leq \rho' \rightsquigarrow f}{\Gamma \vdash x : \rho' \rightsquigarrow f\ x}\ \text{VAR-INF} \qquad \frac{\begin{array}{c}\nu \overset{s}{:} \forall \overline{a}.\overline{\sigma} \to \sigma \in \Gamma \\ \overline{a}_c = \overline{a} \cap ftv(\sigma) \quad \overline{a}_e = \overline{a} - \overline{a}_c \\ \vdash [a_c \mapsto \boxed{\overline{\sigma_c}}]\sigma \leq \rho' \rightsquigarrow f \\ \Gamma \vdash^{poly} u_i : [a_e \mapsto \boxed{\overline{\sigma_e}}, \overline{a_c \mapsto \sigma_c}]\sigma_i \rightsquigarrow u_i'\end{array}}{\Gamma \vdash \nu\ \overline{u} : \rho' \rightsquigarrow f\ (\nu\ [\overline{\sigma}_c \overline{\sigma}_e]\ \overline{u'})}\ \text{VAR-SIG}$$

$$\frac{\begin{array}{c}\vdash \sigma_1' \sim \boxed{\sigma_1} \\ \Gamma, x \overset{s}{:} \sigma_1 \vdash^{poly} t : \sigma_2' \rightsquigarrow t'\end{array}}{\Gamma \vdash (\backslash x\,.\,t) : \sigma_1' \to \sigma_2' \rightsquigarrow (\lambda x\,.\,t')}\ \text{ABS1} \qquad \frac{\Gamma \vdash (\backslash x\,.\,t) : \boxed{\sigma_1} \to \boxed{\sigma_2} \rightsquigarrow t'}{\Gamma \vdash (\backslash x\,.\,t) : \boxed{\sigma_1 \to \sigma_2} \rightsquigarrow t'}\ \text{ABS2}$$

$$\frac{\begin{array}{c}\Gamma \vdash t : \boxed{\sigma} \to \rho' \rightsquigarrow t' \\ \Gamma \vdash^{poly} u : \sigma \rightsquigarrow u'\end{array}}{\Gamma \vdash t\ u : \rho' \rightsquigarrow t'\ u'}\ \text{APP}$$

$$\frac{\begin{array}{c}\Gamma \vdash u : \boxed{\rho} \rightsquigarrow u' \\ \overline{a} = ftv(\rho) - ftv(\Gamma) \\ \Gamma, x \overset{i}{:} \forall \overline{a}\ .\ \rho \vdash t : \rho' \rightsquigarrow t'\end{array}}{\Gamma \vdash \texttt{let}\ x\ \texttt{=}\ u\ \texttt{in}\ t : \rho' \rightsquigarrow (\lambda x\,.\,t')\ (\Lambda \overline{a}\,.\,u')}\ \text{LET} \qquad \frac{\begin{array}{c}ftv(\forall \overline{a}\,.\,\rho) \subseteq dom(\Gamma) \quad \overline{a} \# ftv(\Gamma) \\ \Gamma, \overline{a} \vdash u : \rho \rightsquigarrow u' \\ \Gamma, x \overset{s}{:} \forall \overline{a}.\rho \vdash t : \rho' \rightsquigarrow t'\end{array}}{\Gamma \vdash \texttt{let}\ x\texttt{::}\forall \overline{a}.\rho\ \texttt{=}\ u\ \texttt{in}\ t : \rho' \rightsquigarrow (\lambda x\,.\,t')\ (\Lambda \overline{a}\,.\,u')}\ \text{SIG-LET}$$

$$\boxed{\Gamma \vdash^{poly} t : \sigma'}$$

$$\frac{\Gamma \vdash t : \rho' \rightsquigarrow t' \quad \overline{a} \# ftv(\Gamma)}{\Gamma \vdash^{poly} t : \forall \overline{a}.\rho' \rightsquigarrow \Lambda \overline{a}\,.\,t'}\ \text{GEN1} \qquad \frac{\Gamma \vdash t : \boxed{\rho} \rightsquigarrow t'}{\Gamma \vdash^{poly} t : \boxed{\rho} \rightsquigarrow t'}\ \text{GEN2}$$

Figure 10: Full type system specification

7 Extension of theorems for full type system

Here we extend the properties of the base system for the one given in Figure 10. Note that the contexts used contain modifiers indicating whether a given type entered the context as a result of a user type annotation or abstraction, or by a `let`-binding. The extended environments are given in Figure 9. The only significant change between this system and the previous one is the presence of the rule VAR-INF which is a useful heuristic for reducing the number of annotations that are required in type applications. Unfortunately, in order to still have an easy, unification-based algorithm that is complete for this system, VAR-INF may be used only when the head's type is fully known because of some user annotation. Hence the need for keeping track in the environment of whether the type associated with a binder originated in a type annotation or was inferred.

As an aside, it is straightforward to add annotated abstractions and this has no complications for the type inference algorithm, nor the specification—but we ommit it because it does not add to the expressiveness of the system. Notice as well that annotated abstractions would provide for a slightly different translation of System-F but

again we will not bother introducing them here.

In what follows, whenever we use $\Gamma \vdash t : \rho'$ or $\Gamma \vdash^{poly} t : \sigma'$ we refer to derivations of the **full system** of Figure 10 unless explicitly stated otherwise. We need to generalise the theorems proved in previous sections first.

7.1 Type safety of full system

We naturally extend the type directed translation given for the shorter version of the language to the full system. The next theorem is an easy check.

Lemma 7.1 (Term translation (extends Theorem 5.2)). *If* $\Gamma \vdash t : \rho' \rightsquigarrow t'$ *then* $\Gamma \vdash^F t' : strip(\rho')$. *If* $\Gamma \vdash^{poly} t : \sigma' \rightsquigarrow t'$ *then* $\Gamma \vdash^F t' : strip(\sigma')$.

Proof. Easy induction appealing to Lemma 5.1. □

7.2 Embedding of System-F

The new system is still capable of capturing System-F. To see this we first observe that the new system is an extension of the base system. As an abbreviation assume that for a given full system context Γ we let $\Gamma^\dagger$ to be the same context where we erased all its modifiers. Formally:

$$\begin{array}{lcl} .^\dagger & = & . \\ (\Gamma, a)^\dagger & = & \Gamma^\dagger, a \\ (\Gamma, x \overset{i}{:} \sigma)^\dagger & = & \Gamma^\dagger, x{:}\sigma \\ (\Gamma, x \overset{s}{:} \sigma)^\dagger & = & \Gamma^\dagger, x{:}\sigma \end{array}$$

Lemma 7.2 (Full system extends base). *The following are true:*

1. *If* $\Gamma^\dagger \vdash t_1 : \rho'$ *in base then* $\Gamma \vdash t_1 : \rho'$ *in the full system.*
2. *If* $\Gamma^\dagger \vdash^{poly} t_1 : \sigma'$ *in base then* $\Gamma \vdash^{poly} t_1 : \sigma'$ *in the full system.*

Proof. We proceed by induction on the height of the typing derivation. We inline usages of the second part in usages of the first part, so we can apply the induction hypothesis in all cases. We proceed by case analysis on the structure of t_1.

- Case $t_1 = \nu$ and $\nu \overset{i}{:} \sigma \in \Gamma$. The only rule that could have been used was VAR and we are done by observing that VAR-INF applies in the full system.
- Case $t_1 = \nu\ \overline{u}$ and $\nu \overset{s}{:} \sigma \in \Gamma$. We prove this part by inner induction on the size of $\overline{u}$. If $\overline{u}$ is empty then, in the core language we had $\Gamma^\dagger \vdash \nu : \rho'$ given that $\nu \overset{s}{:} \forall \overline{a}.\rho \in \Gamma$, and $\vdash \forall \overline{a}.\rho \leq \rho'$, or by inversion $\vdash [\overline{a \mapsto \boxed{\sigma}}]\rho \leq \rho'$ for some $\overline{\boxed{\sigma}}$. Using this we can apply the rule VAR-SIG to get the result. Suppose now for the inductive step that

$$\Gamma^\dagger \vdash (\nu\ \overline{u})\ u : \rho' \tag{1}$$
$$\Gamma^\dagger \vdash (\nu\ \overline{u}) : \boxed{\sigma} \to \rho' \tag{2}$$
$$\Gamma^\dagger \vdash^{poly} u : \sigma \tag{3}$$

  Then induction hypothesis tell us that in the full system:

$$\Gamma \vdash \nu\ \overline{u} : \boxed{\sigma} \to \rho' \tag{4}$$
$$\Gamma^\dagger \vdash^{poly} u : \sigma \tag{5}$$

  By applying rule APP we are then done[2].
- Case $t_1 = t\ u$ not in all other cases above. In the base system this could only be derivable using rule APP and in the target system we can apply induction hypothesis to the premises of the rule and by the full system APP rule we are done.
- Case $t_1 = \backslash x.t$. Then either ABS1 or ABS2 was used in the base system and in each case the result follows by application of the induction hypothesis and rules ABS1 and ABS2 accordingly.

[2]Notice that we did not assume that the rule APP was only applicable when rule VAR-SIG is not applicable; on the contrary there is an overlap which is going to introduce some problems when we will discuss the completeness of an algorithm implementing the full type system.

- Case $t_1 = \texttt{let}\ x\ \texttt{=}\ u\ \texttt{in}\ t$. Then LET must have been used and the result follows by the induction hypothesis and rule LET in the full system again.
- Case $t_1 = \texttt{let}\ x :: \forall \overline{a}.\rho\ \texttt{=}\ u\ \texttt{in}\ t$. Rule SIG-LET must have been used and the result follows by induction hypothesis and rule SIG-LET in the full system.

The second part is straightforward. □

Theorem 7.3 (Translation of System-F (full)). *If $\Gamma \vdash^F t : \sigma$ and $wpr(\sigma) = \forall \overline{a}.\rho$ then*

1. $\Gamma \vdash^{poly} [\![t]\!]_\Gamma : \sigma$.
2. $\Gamma \vdash^{poly} [\![t]\!]_\Gamma : \boxed{\rho}$.

Proof. Directly follows by Theorem 4.11 and Lemma 7.2. □

7.3 Weakening and substitution lemmas for the full system

Lemma 7.4 (Substitution for typing). *Assume without loss of generality that scoped variables can α-vary, so as to satisfy $dom(\Gamma)\#vars(S)$. The following are true for the typing relation.*

1. *If $\Gamma \vdash t : \rho'$ then $S\Gamma \vdash t : S\rho'$.*
2. *If $\Gamma \vdash^{poly} t : \sigma'$ then $S\Gamma \vdash^{poly} t : S\sigma'$*

Proof. Easy induction in the style of the proof of the substitution for the base system. □

For weakening we have to be a little more careful. We define an ordering on contexts as follows.

$$\Gamma_1 \preceq \Gamma_2 \quad \text{iff} \quad \begin{array}{l} dom(\Gamma_1) = dom(\Gamma_2) \\ \forall x \overset{s}{:} \sigma \in \Gamma_1 . x \overset{s}{:} \sigma \in \Gamma_2 \\ \forall x \overset{i}{:} \sigma_1 \in \Gamma_1 . \exists \sigma_2 . x \overset{i}{:} \sigma_2 \in \Gamma_2 \wedge (\vdash \sigma_1 \leq_{sh} \sigma_2) \end{array}$$

Now weakening holds with respect to this ordering. The intuition behind this ordering of contexts is that the algorithm itself will put more polymorphic types in the contexts in general. However it will put exactly the same types for rigid modifiers (modulo unification information, of course) because these rigid bindings originate somewhere in a type annotation (if of course they contain polymorphism).

Theorem 7.5 (Weakening for typing (extends Theorem 6.6)). *The following are true:*

1. *If $\Gamma_1 \vdash t : \rho'$ and $\vdash \Gamma_2 \preceq \Gamma_1$ then $\Gamma_2 \vdash t : \rho'$.*
2. *If $\Gamma_1 \vdash^{poly} t : \sigma'$ and $\vdash \Gamma_2 \preceq \Gamma_1$ then $\Gamma_2 \vdash^{poly} t : \sigma'$.*

Proof. Easy induction. The case of VAR-INF goes through without any problem since the heads are bound with exactly the same type in the two contexts. □

Finally, boxing and unboxing behaviour is the same as was in the base system.

Lemma 7.6 (Uncontrolled unboxing for typing (full)). *The following are true of the typing relation:*

1. *If $\Gamma \vdash^{poly} t : \sigma'$ and $\vdash \sigma'' \blacktriangleright \sigma'$ then $\Gamma \vdash^{poly} t : \sigma''$.*
2. *If $\Gamma \vdash t : \rho'$ and $\vdash \rho'' \blacktriangleright \rho'$ then $\Gamma \vdash t : \rho''$.*

Proof. Similar to the proof of Lemma 3.36 appealing to Corollary 3.29 in the VAR-SIG case. □

Lemma 7.7 (Controlled boxing for typing (full)). *The following are true of the typing relation in the full system:*

1. *If $\Gamma \vdash^{poly} t : \sigma'$ and $\vdash \sigma' \triangleright \sigma''$ then $\Gamma \vdash^{poly} t : \sigma''$.*
2. *If $\Gamma \vdash t : \rho'$ and $\vdash \rho' \triangleright \rho''$ then $\Gamma \vdash t : \rho''$.*

Proof. Similar to the proof of Lemma 3.35 appealing to Corollary 3.26 in the VAR-SIG case. □

$$\boxed{\Gamma \vdash^{HM} t : \tau}$$

$$\frac{\nu{:}\sigma \in \Gamma \quad \vdash \sigma \leq_{sh} \tau}{\Gamma \vdash \nu : \tau}\ \textsc{hmvar}$$

$$\frac{\Gamma, x{:}\tau_1 \vdash t : \tau_2}{\Gamma \vdash \backslash x\,.\,t : \tau_1 \to \tau_2}\ \textsc{hmabs} \qquad \frac{\Gamma \vdash t : \tau_1 \to \tau_2 \quad \Gamma \vdash u : \tau_1}{\Gamma \vdash t\ u : \tau_2}\ \textsc{hmapp}$$

$$\frac{\Gamma \vdash u : \tau_x \quad \overline{a} = ftv(\tau) - ftv(\Gamma) \quad \Gamma, x{:}\forall \overline{a}\,.\,\tau_x \vdash t : \tau}{\Gamma \vdash (\texttt{let}\ x\texttt{=}u\ \texttt{in}\ t) : \tau}\ \textsc{hmlet}$$

Figure 11: Vanilla Hindley-Milner type system

7.4 Conservative extension of Hindley-Milner

In this section we show that the full system conservatively extends the vanilla Hindley-Milner system, given in Figure 11. Throughout this section *we assume that the term and type syntax adheres with the restrictions of HM*, i.e. no type annotations on terms, no higher-rank types in contexts.

7.4.1 Base system conservatively extends HM

We first show in this section that the base system is a conservative extension of Hindley-Milner type system.

Theorem 7.8 (Extension of HM (base)). *If* $\Gamma \vdash^{HM} t : \tau$ *then* $\Gamma \vdash t : \tau$.

Proof. The proof is by induction on the height of the typing derivation. We proceed with case analysis on the last rule used. We just show the first part; the second follows directly by the first and Theorem 3.37.

- Case HMVAR. We have that $\Gamma \vdash^{HM} \nu : \tau$ given that $\nu{:}\sigma \in \Gamma$ and $\vdash \sigma \leq_{sh} \tau$. By Lemma 6.1 we get that $\vdash \sigma \leq \tau$ and by applying VAR we are done.
- Case HMAPP. In this case we have that $\Gamma \vdash^{HM} t\ u : \tau_2$, given that

$$\Gamma \vdash^{HM} t : \tau_1 \to \tau_2 \tag{1}$$
$$\Gamma \vdash^{HM} u : \tau_1 \tag{2}$$

  By induction hypothesis we have that $\Gamma \vdash t : \tau_1 \to \tau_2$, and using Theorem 3.37 $\Gamma \vdash t : \boxed{\tau_1} \to \tau_2$. By induction we also have $\Gamma \vdash u : \tau_1$ and we are done by applying rule APP.
- Case HMABS. Here we have $\Gamma \vdash^{HM} \backslash x\,.\,t : \tau_1 \to \tau_2$ given that

$$\Gamma, x{:}\tau_1 \vdash^{HM} t : \tau_2 \tag{3}$$

  By induction $\Gamma, x{:}\tau_1 \vdash t : \tau_2$ and by observing that $\vdash \tau_1 \sim \boxed{\tau_1}$ and rule ABS1 we get $\Gamma \vdash t : \tau_1 \to \tau_2$ as required.
- Case HMLET. Here $\Gamma \vdash^{HM} (\texttt{let}\ x\texttt{=}u\ \texttt{in}\ t) : \tau$ given that

$$\Gamma \vdash^{HM} u : \tau_x \quad \overline{a} = ftv(\tau) - ftv(\Gamma) \tag{4}$$
$$\Gamma, x{:}\forall \overline{a}\,.\,\tau_x \vdash^{HM} t : \tau \tag{5}$$

  By induction hypothesis and Theorem 3.37 we get $\Gamma \vdash u : \boxed{\tau_x}$ and by induction hypothesis we have $\Gamma, x{:}\forall \overline{a}\,.\,\tau_x \vdash t : \tau$. The result follows then by applying rule LET.

□

For conservativity we first need an auxiliary lemma. Assume a stricter version of the protected types given below:

$$\varpi \quad ::= \quad \tau \mid \boxed{\sigma} \mid \varpi \to \varpi \mid T\ \varpi$$

Lemma 7.9 (Inference mode for Hindley-Milner (base)). *Assume $\Gamma \vdash t : \varpi$ and that the environment does not contain higher-rank types, and t does not contain type annotations. Then the boxes of ϖ can only contain monotypes.*

Proof. The proof is by induction on the height of the derivations. We proceed with case analysis on the rule that was used in the derivation.

- Case VAR. We have in this case that $\Gamma \vdash \nu : \varpi$ given that $\nu{:}\sigma \in \Gamma$, and $\vdash \sigma \leq \varpi$. By our restrictions, it is the case that $\sigma = \forall \overline{a}.\tau$. Then by inversion it is the case that $\vdash \boxed{[\overline{a \mapsto \boxed{\sigma}}]\tau} \leq \varpi$ and by Theorem 3.30 we can expand the box of ϖ, getting a $\boxed{\sigma}$ and it will still be that $\vdash \boxed{[\overline{a \mapsto \sigma}]\tau} \leq \boxed{\sigma}$ or it must be that $\overline{\sigma}$ are monotypes and σ is a monotype, yielding that ϖ must only contain monotypes inside its boxes.
- Case ABS1. Here we have that $\Gamma \vdash (\backslash x\,.\,t) : \varpi_1' \to \varpi_2'$ given that

$$\vdash \varpi_1' \sim \boxed{\sigma_1} \tag{1}$$

$$\Gamma, x{:}\sigma_1 \vdash^{poly} t : \varpi_2' \tag{2}$$

  From (1) and Corollary 3.10 we can expand the boxes of ϖ_1' to get $\boxed{\sigma}$. But then it must be that $\sigma_1 = \sigma = \tau$ by BBEQ; it follows that all the boxes of ϖ_1' contain monotypes. By induction hypothesis we get the same for the boxes of ϖ_2'.
- Case ABS2. Follows directly by induction hypothesis.
- Case APP. Easy application of the induction hypothesis.
- Case LET. Follows by induction hypothesis.
- Case SIG-LET. Cannot happed because of the syntactic restrictions.

□

Corollary 7.10 (Pure inference for Hindley-Milner). *If $\Gamma \vdash t : \boxed{\sigma}$, Γ does not contain higher-rank types, and t does not contain type annotations then $\sigma = \tau$ for some τ.*

Proof. Direct consequence of Lemma 7.9. □

Theorem 7.11 (Conservativity over HM (base)). *If $\Gamma \vdash t : \tau$ then $\Gamma \vdash^{HM} t : \tau$.*

Proof. The proof is by induction on the structure of t. We proceed by case analysis on the last rule used in the derivation.

- Case VAR. Follows by Lemma 6.2 and rule HMABS.
- Case ABS1. Easy application of the induction hypothesis and rule ABS.
- Case ABS2. Cannot happen.
- Case APP. Here we have that $\Gamma \vdash t\ u : \tau_2$ given that

$$\Gamma \vdash t : \boxed{\sigma_1} \to \tau_2 \tag{1}$$

$$\Gamma \vdash^{poly} u : \sigma_1 \tag{2}$$

  By Lemma 7.9 it must be that $\sigma_1 = \tau_1$ and we also have by Theorem 3.37 that $\Gamma \vdash t : \tau_1 \to \tau_2$, therefore by induction hypothesis $\Gamma \vdash^{HM} t : \tau_1 \to \tau_2$. Then by induction hypothesis as well we have $\Gamma \vdash^{HM} u : \tau_1$ and by rule HMAPP we are done.
- Case LET. Similar to the case of APP above to unbox the inferred type.
- Case SIG-LET. Cannot happen because of the syntactic restrictions.

□

7.4.2 Full system conservatively extends HM

The extension theorem is straightforward.

Theorem 7.12 (Extension of HM (full)). *If* $\Gamma^\dagger \vdash^{HM} t : \tau$ *then* $\Gamma \vdash t : \tau$ *and* $\Gamma \vdash t : \boxed{\tau}$ *in the full system.*

Proof. Follows by Theorem 7.8 and the fact that the full system extends base. □

In order to show conservativity, just as in the proof of Theorem 7.14, we need the versions of Lemma 7.9 and Theorem 3.37 for the full system.

Lemma 7.13 (Inference mode for Hindley-Milner (full)). *Assume* $\Gamma \vdash t : \varpi$ *and that the environment does not contain higher-rank types, and* t *does not contain type annotations. Then the boxes of* ϖ *can only contain monotypes.*

Proof. Similar to the proof of Lemma 7.9. □

Theorem 7.14 (Conservativity over HM (full)). *If* $\Gamma \vdash t : \tau$ *then* $\Gamma^\dagger \vdash^{HM} t : \tau$.

Proof. Similar to the proof of Theorem 7.11. We just show here the case for VAR-SIG. In this case we have that:

$$\Gamma \vdash \nu\, \overline{u} : \tau \tag{1}$$

given that

$$\nu \overset{s}{:} \forall \overline{a}.\overline{\tau} \to \tau_r \in \Gamma \tag{2}$$

$$\overline{a}_c = \overline{a} \cap ftv(\tau_r) \quad \overline{a}_e = \overline{a} - \overline{a}_c \tag{3}$$

$$\vdash \overline{[a_c \mapsto \boxed{\sigma_c}]}\tau_r \leq \tau \tag{4}$$

$$\Gamma \vdash^{poly} u_i : \overline{[a_e \mapsto \boxed{\sigma_e}]}, \overline{a_c \mapsto \sigma_c}]\tau_i \tag{5}$$

By equation (4) and by expanding the boxes using Lemma 3.26 we get that

$$\vdash \boxed{\overline{[a_c \mapsto \sigma_c]}\tau_r} \leq \boxed{\tau} \tag{6}$$

which means that $\overline{\sigma}_c = \overline{\tau}_c$ and $\overline{[a_c \mapsto \tau_c]}\tau_r = \tau$. Furthermore we know that for each u_i we have $\Gamma \vdash^{poly} u_i : \overline{[a_e \mapsto \boxed{\sigma_e}]}, \overline{a_c \mapsto \tau_c}]\tau_i$ and by Lemma 7.13 $\overline{\sigma}_e = \overline{\tau}_e$. Then by the unboxing lemma, Lemma 7.6, it is also the case that $\Gamma \vdash^{poly} u_i : \overline{[a_e \mapsto \tau_e}, \overline{a_c \mapsto \tau_c}]\tau_i$. Then the result follows by an application of the VAR rule and a sequence of applications of APP. □

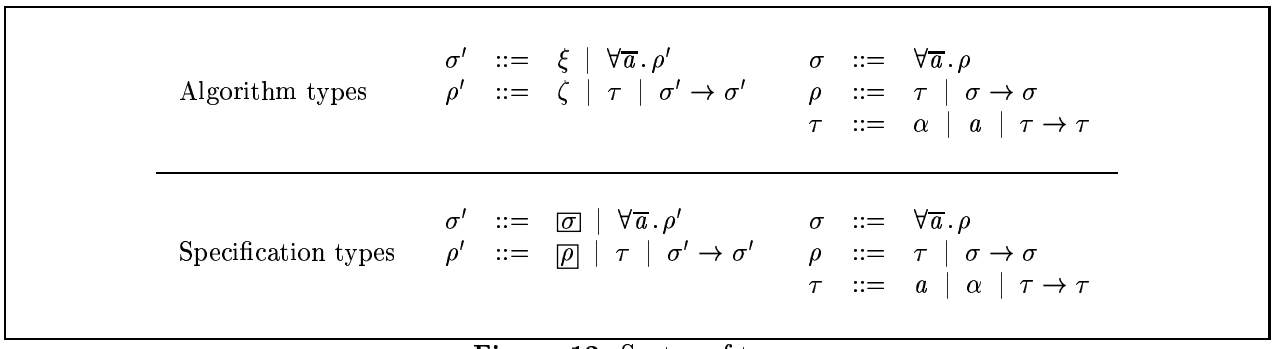

Figure 12: Syntax of types

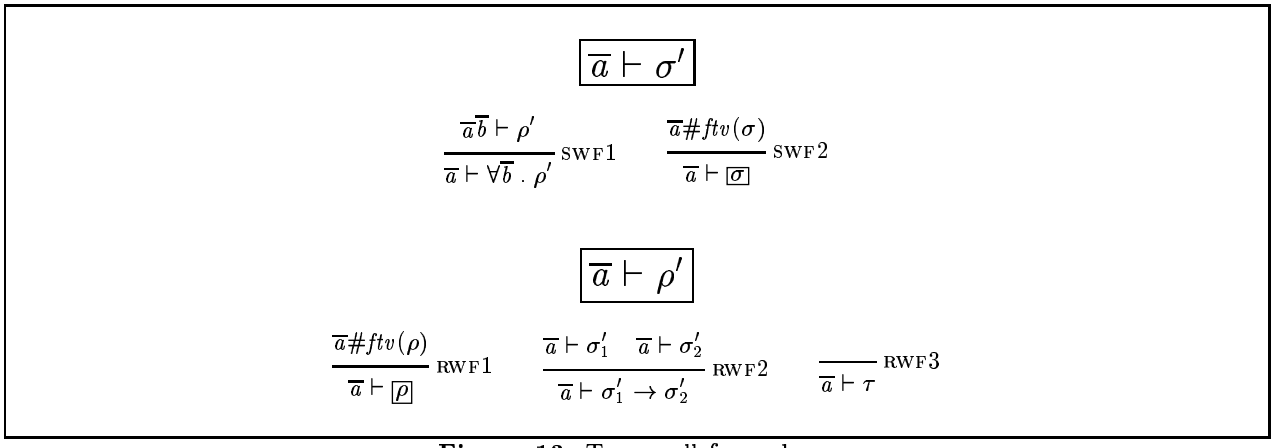

Figure 13: Type well-formedness

8 A type inference algorithm

In this section we give a concrete type inference algorithm for the full system of Figure 10 and prove that this algorithm is sound and complete with respect to the specification of the type system.

8.1 Definitions

Figure 12 gives the syntactic systems we use. The types that appear in the algorithm include normal unification variables denoted with α, β, γ and boxy σ- and ρ- variables denoted with ξ, ζ. Normal unification variables will be used to force a type to be monotype, since they can only be mapped to monotypes. There is no restriction on the other hand on boxy σ- or ρ-variables.

In Figure 13 we formalize the notion of well-formedness of types. The specification types have to satisfy a non-syntactic restriction: quantified variables should not appear inside boxes. It is easy to verify that our specification maintains this invariant.

Next, we define the notion of *well-formed unifiers*. A unifier, denoted with letters S, R, is going to be a (total) map from unification or boxy variables to algorithm types. Moreover we want to formalize the fact that monotype variables cannot be mapped to ρ or σ types. We incorporate this in the notion of a well-formed unifier, denoted by $\vdash S$.

$$\vdash S \quad \Leftrightarrow \quad \begin{array}{ll} \forall \sigma'.\, SS(\sigma') = S\sigma' & (P_1) \\ \alpha \in dom(S) \Rightarrow S\alpha \text{ is a } \tau \text{ type.} & (P_2) \\ \zeta \in dom(S) \Rightarrow \nexists \zeta' \in S\zeta & (P_3) \end{array}$$

Given a set of variables $\mathcal{X}$, we define the *excluded-$\mathcal{X}$ equivalence* relation on unifiers as:

$$S_1 = S_2 \backslash_{\mathcal{X}} \quad \Leftrightarrow \quad \forall \alpha, \zeta \# \mathcal{X}\ S_1(\alpha, \zeta) = S_2(\alpha, \zeta)$$

Intuitively, two substitutions are excluded-$\mathcal{X}$ equivalent if they agree everywhere except perhaps for some variables in $\mathcal{X}$.

For a given unifier S, we have a translation operation $[\![\cdot]\!]_S$ whose definition is given below:

$$\begin{array}{lcll} [\![\forall\overline{a}\,.\,\rho']\!]_S & = & \forall\overline{a}\,.\,[\![\rho']\!]_S & \overline{a}\#range(S) \\ [\![\xi]\!]_S & = & \boxed{S\xi} & \\ [\![\sigma_1' \to \sigma_2']\!]_S & = & [\![\sigma_1']\!]_S \to [\![\sigma_2']\!]_S & \\ [\![\tau]\!]_S & = & S\tau & \end{array}$$

We can also define a reverse translation $[\![\cdot]\!]^{-1}$ that replaces every box with an appropriate variable, giving back a unifier.

$$\begin{array}{lcl} [\![\forall\overline{a}\,.\,\rho']\!]^{-1} & = & (\forall\overline{a}\,.\,\rho_t', S) \quad [\![\rho']\!]^{-1} = (\rho_t', S) \\ [\![\boxed{\sigma}]\!]^{-1} & = & (\xi, [\xi \mapsto \sigma]) \quad \xi \text{ fresh} \\ [\![\sigma_1' \to \sigma_2']\!]^{-1} & = & (\sigma_{t1}' \to \sigma_{t2}', S_2 \cdot S_1) \\ & & \quad [\![\sigma_1']\!]^{-1} = (\sigma_{t1}', S_1) \\ & & \quad [\![\sigma_2']\!]^{-1} = (\sigma_{t2}', S_2) \\ [\![\boxed{\rho}]\!]^{-1} & = & (\zeta, [\zeta \mapsto \rho]) \quad \zeta \text{ fresh} \\ [\![\tau]\!]^{-1} & = & (\tau, \emptyset) \end{array}$$

Then the obvious property holds: If we translate a specification type to an algorithm type and translate it back to a specification type, we get the same type back.

Corollary 8.1. *If* $(\sigma_t', S) = [\![\sigma']\!]^{-1}$ *then* $[\![\sigma_t']\!]_S = \sigma'$.

Proof. Directly follows by the definitions above. □

Moreover is is easy to see the following:

Corollary 8.2. *If* $[\![\sigma']\!]^{-1} = (\sigma_t', S)$ *then* $\vdash S$.

Proof. Directly follows by the definitions above. □

We use the following syntax for *generalization* over the variables of an environment.

$$\overline{\Gamma}(\rho) \quad = \quad \forall\overline{a}\,.\,\rho \text{ where } \overline{a} = ftv(\rho) - ftv(\Gamma)$$

Lemma 8.3. $\vdash S\overline{\Gamma}(\rho) \leq_{sh} \overline{S\Gamma}(S\rho)$.

Proof. Let $\overline{\Gamma}(\rho) = \forall\overline{a}\,.\,\rho$ where $\overline{a} = ftv(\rho) - ftv(\Gamma)$. Let $\overline{g}$ be a new set of variables, such that $\overline{g} \notin ftv(\Gamma), vars(S), ftv(\rho)$. Then $S\overline{\Gamma}(\rho) = \forall\overline{g}\,.\,S([\overline{a \mapsto g}]\rho)$. Now, let $\overline{S\Gamma}(S\rho) = \forall\overline{b}\,.\,S\rho$, where $\overline{b} \in ftv(S\rho) - ftv(S\Gamma)$. We want to prove that $\vdash \forall\overline{g}\,.\,S([\overline{a \mapsto g}]\rho) \leq_{sh} \forall\overline{b}\,.\,S\rho$. First we need to show that $\overline{b} \notin ftv(S\overline{\Gamma}(\rho))$. By contradiction, assume that there exists a $b \in \overline{b}$ such that $b \in ftv(S\overline{\Gamma}(\rho))$. Therefore there exists $d \in ftv(\overline{\Gamma}(\rho))$ such that $b \in Sd$. From this we get that $d \in ftv(\rho)$ and $d \in ftv(\Gamma)$. Then, since $b \in Sd$, $b \in S\Gamma$, which is a contradiction to the fact that $b \in ftv(S\rho) - ftv(S\Gamma)$. Therefore, it only remains to be shown that for some types $\overline{\tau}$ it is the case that $[\overline{g \mapsto \tau}]S([\overline{a \mapsto g}]\rho) = S\rho$. Pick $\overline{\tau} = \overline{Sa}$. □

We use $ftv(\cdot)$ do denote the free variables (ordinary, meta, and boxy) of the argument. We use $fov(\cdot), fmv(\cdot), fbv(\cdot)$ to denote the ordinary, meta, and boxy variables of the argument respectively. We use $fuv(\cdot)$ to denote free meta, and boxy variables.

8.2 Summary and generalised results

We first give in this section a summary of the most important results along with their proofs. In the following section we give the statements of the theorems upon which the results of this section are based.

Corollary 8.4 (First Completeness Corollary). *If* $\vdash S$, $[\![\Gamma]\!]_S \vdash t : [\![\rho']\!]_S$, *and* $\mathcal{A}_0\#ftv(\Gamma, \rho'), vars(S)$ *then* $(\emptyset, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S_0, \mathcal{A}_1)$ *such that* $\exists R\,.\,S = R \cdot S_0 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.

Proof. Follows directly from Theorem 8.38 with empty initial unifier. □

The corollary implies that if we replace the boxes in a context and in a type with arbitrary fresh variables and run the algorithm with the same term and the new context and type, the algorithm is going to succeed and return back a most general unifier.

We can simplify even more the completeness corollary using as well the fact that fresh boxy variables get filled in in every step of the algorithm:

Corollary 8.5 (Second Completeness Corollary). *if* $\vdash t : \boxed{\rho}$ *and* $\mathcal{A}_0\zeta\#ftv(\rho)$ *then* $(\emptyset, \mathcal{A}_0) \succ \vdash t : \zeta \succ (S_0, \mathcal{A}_1)$ *such that* $S_0\zeta = \rho_0$ *and* $\exists R\,.\,R\rho_0 = \rho$.

Proof. Consider corollary 8.4 and take $S = [\zeta \mapsto \rho]$. Then it follows that $(\emptyset, \mathcal{A}_0) \succ \vdash t : \zeta \succ (S_0, \mathcal{A}_1)$. Now, by Lemma 8.36 it must be that $\zeta \in dom(S_0)$, therefore $S_0\zeta = \rho_0$ for some ρ_0. But then, since $S = R \cdot S_0\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ for some R and $\zeta\#\mathcal{A}_0$ we have $S\zeta = RS_0\zeta$, which implies $\rho = R\rho_0$. □

Intuitively this says that if the specification assigns a completely boxy type $\boxed{\rho}$ to a term then the algorithm is always going to be able to infer a type ρ_0 such that ρ is the result of a substitution applied to ρ_0.

The soundness corollary is the following.

Corollary 8.6 (Soundness Corollary). *If* $\mathcal{A}_0\#ftv(\Gamma, \rho')$, $\vdash \Gamma$, *and* $(\emptyset, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S, \mathcal{A}_1)$ *then* $[\![\Gamma]\!]_S \vdash t : [\![\rho']\!]_S$.

Proof. It follows directly from Theorem 8.37 taking an empty initial unifier. □

Completeness and soundness of the algorithm combined with determinacy give a property that asserts that there is a "best" type that can be given to a term; modulo checked information.

Definition 8.7 (Spine equivalence). *The spine equivalence relation is defined as a subset of the set of pairs of (specification) types as follows:*

$$\begin{array}{lcll} \forall\overline{a}\,.\,\rho_1' & \simeq & \forall\overline{a}\,.\,\rho_2' & \text{iff } \rho_1' \simeq \rho_2' \\ \boxed{\sigma_1} & \simeq & \boxed{\sigma_2} & \\ \sigma_1' \to \sigma_2' & \simeq & \sigma_3' \to \sigma_4' & \text{iff } \sigma_1' \simeq \sigma_3' \text{ and } \sigma_2' \simeq \sigma_4' \\ \tau & \simeq & \tau & \end{array}$$

It is easy to confirm that this relation is indeed an equivalence relation. Then combining soundness and completeness we get the following theorem.

Theorem 8.8 (Principal Types). *Suppose* $\vdash t : \rho'$. *Then there exists a* ρ_0' *such that,* $\forall\rho''$ *with* $\vdash t : \rho''$ *and* $\rho'' \in [\rho']_\simeq$ *we have that* $\vdash t : \rho_0'$ *and* $\rho'' = R\rho_0'$ *for some substitution* R.

Proof. Consider $(\rho_a', S) = [\![\rho']\!]^{-1}$. It is easy to see that $\forall\rho'' \in [\rho']_\simeq\,.\,(\rho_a', S') = [\![\rho'']\!]^{-1}$ provided we choose the fresh variables in the same way, that is, all spine-equivalent types are the results of substitutions to a common algorithm type. Then, by Corollary 8.4, for each of those S' we have that $(\emptyset, \mathcal{A}_0) \succ \vdash t : \rho_a' \succ (S_0, \mathcal{A}_1)$ and $S' = R \cdot S_0\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ for some R. However we know that $[\![\rho_a']\!]_{S'} = \rho''$ therefore $R[\![\rho_a']\!]_{S_0} = \rho''$ since also by Lemma 8.36 all the boxy variables of ρ_a' are in $dom(S_0)$. Taking $\rho_0' = [\![\rho_a']\!]_{S_0}$ finishes the proof. □

This last theorem says that for all types that belong in the same "spine" equivelence class, that is, **they have the same amount of checkable information** , there exists a best type that can be assigned to the term, such that all others are substitution instances of that type.

8.3 Detailed properties

We now give in more detail the theorems that are true of the algorithmic relations. Here is a small roadmap. We are going to prove soundness directly by induction on the algorithmic derivations. However for completeness it is slightly more convenient and economical in notation if we first convert the specification derivations into an algorithmic form, where we have eliminated overlapping. Then we show that if a judgement is derivable in the specification, it is derivable in the algorithmic version of it. Finally we show, roughly speaking, that everything derivable in the algorithmic version of the specification is derivable in the concrete algorithm.

8.3.1 Algorithmic version of type system

Figures 14, and 15 give the algorithmic presentation of the judgements[3].

Lemma 8.9 (Algorithmic matching symmetry). *If* $\vdash\!\blacktriangleright \sigma_1' \sim \sigma_2'$ *then* $\vdash\!\blacktriangleright \sigma_2' \sim \sigma_1'$.

[3]Notice that it still does not correspond 1-1 to concrete algorithm derivations because of unification. Take for example the case of VAR-SIG and APP and their concrete algorithm versions. Immediately we see that the correspondence may break because of unification; however we will show that if VAR-SIG was used in the specification, we can recover this situation by (potential) use of AVAR-SIG and rule AAAP in the algorithm.

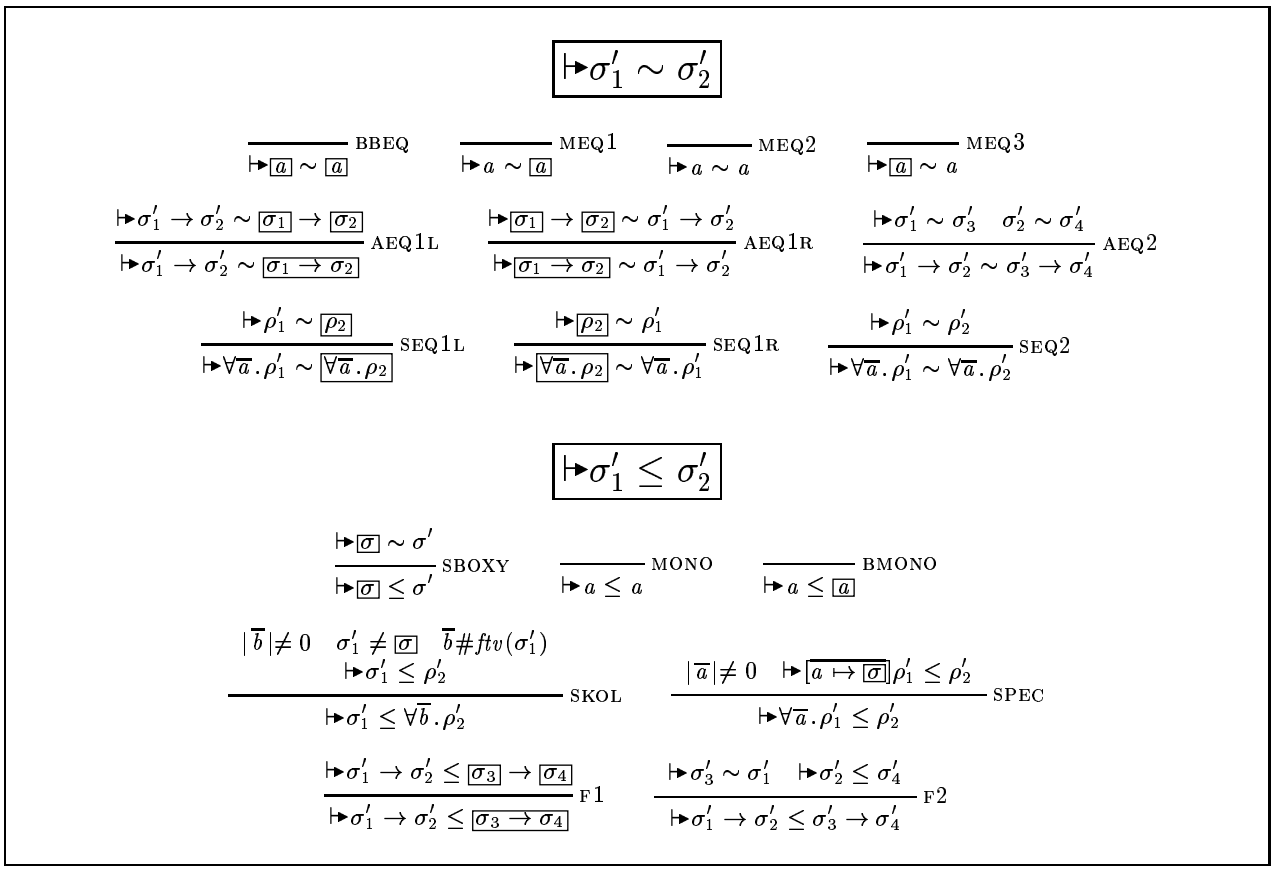

Figure 14: Subsumption and boxy matching—Algorithmic presentation

Proof. Straightforward induction on the height of the derivations. □

Lemma 8.10 (Algorithmic monotype matching). *For any τ it is the case that $\vdash\!\blacktriangleright \tau \sim \tau$, $\vdash\!\blacktriangleright \boxed{\tau} \sim \tau$ and $\vdash\!\blacktriangleright \boxed{\tau} \sim \boxed{\tau}$.*

Proof. Straightforward induction on the structure of τ. □

Theorem 8.11 (Algorithmic boxy matching). *If $\Gamma \vdash \sigma'_1 \sim \sigma'_2$ then $\Gamma \vdash\!\blacktriangleright \sigma'_1 \sim \sigma'_2$.*

Proof. By induction on the height of the derivation $\Gamma \vdash \sigma'_1 \sim \sigma'_2$. The case for SYM follows by Lemma 8.9, the cases of rule BBEQ, MEQ1, and MEQ2 follow directly by Lemma 8.10, and the rest cases are straightforward applications of the induction hypotheses and the corresponding rules in the algorithmic version. □

Theorem 8.12 (Algorithmic boxy matching (inverse)). *If $\vdash\!\blacktriangleright \sigma'_1 \sim \sigma'_2$ then $\vdash \sigma'_1 \sim \sigma'_2$.*

Proof. Straightforward induction. □

Lemma 8.13 (Algorithmic subsumption for monotypes). *For all τ, $\vdash\!\blacktriangleright \tau \leq \tau$ and $\vdash\!\blacktriangleright \tau \leq \boxed{\tau}$.*

Proof. By induction on the structure of τ. If $\tau = a$ then the result follows by rules MONO and BMONO. If on the other hand $\tau = \tau_1 \to \tau_2$, for the first part using rule F2 we must show that $\vdash\!\blacktriangleright \tau_1 \sim \tau_1$ and $\vdash\!\blacktriangleright \tau_2 \leq \tau_2$. The first follows by Lemma 8.10 and the second by induction hypothesis. For the second part, using rule F1 and F2, we must show that $\vdash\!\blacktriangleright \boxed{\tau_1} \sim \tau_1$ and $\vdash\!\blacktriangleright \tau_2 \leq \boxed{\tau_2}$. The first follows by Lemma 8.10, and the second by induction hypothesis. □

Theorem 8.14 (Algorithmic subsumption). *If $\Gamma \vdash \sigma'_1 \leq \sigma'_2$ then $\Gamma \vdash\!\blacktriangleright \sigma'_1 \leq \sigma'_2$.*

Proof. By induction on the height of the derivation $\Gamma \vdash \sigma'_1 \leq \sigma'_2$. The case of SBOXY follows by Theorem 8.11 and rule SBOXY. The case of MONO and BMONO follow by Lemma 8.13. The cases of SKOL, SPEC, and F1 follow by induction hypotheses and the corresponding rule and the case of F2 follows by induction hypothesis, Theorem 8.11 and application of F2. □

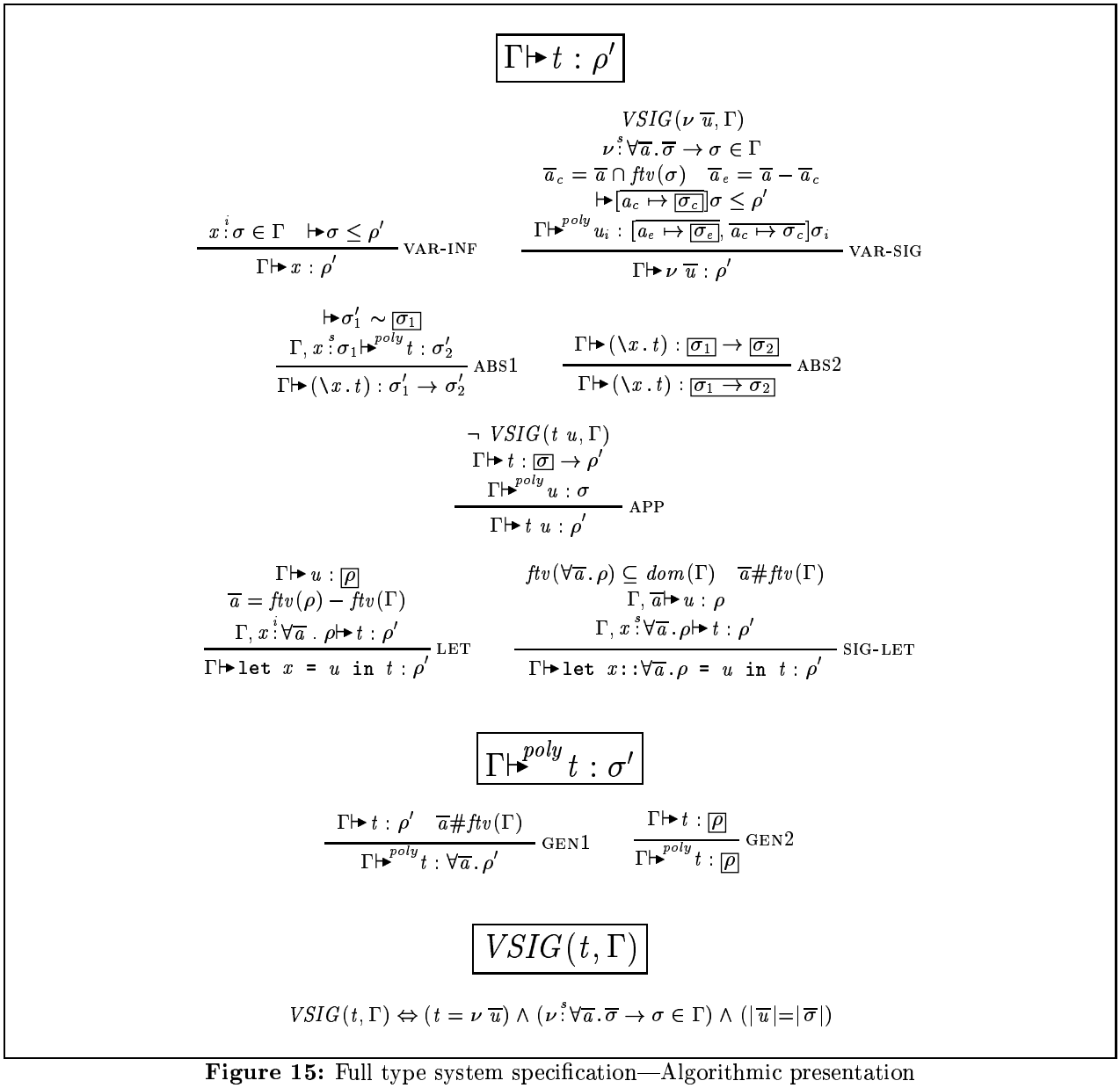

Figure 15: Full type system specification—Algorithmic presentation

Theorem 8.15 (Algorithmic subsumption (inverse)). *If* $\vdash\!\blacktriangleright \sigma_1' \leq \sigma_2'$ *then* $\vdash \sigma_1' \leq \sigma_2'$.

Proof. Straightforward induction. □

Theorem 8.16.

1. *If* $\Gamma \vdash t : \rho'$ *then* $\Gamma \vdash\!\blacktriangleright t : \rho'$.
2. *If* $\Gamma \vdash^{poly} t : \sigma'$ *then* $\Gamma \vdash\!\blacktriangleright^{poly} t : \sigma'$.

Proof. We prove the two claims simultaneously by induction on the height of the derivations. It is straightforward to modify the controlled boxing and uncontrolled unboxing proofs for the algorithmic system and we take them to be true in what follows. All cases except for the application case are straightforward appealing to Theorem 8.11 and Theorem 8.14. We consider now the case of APP. In this case we have that

$$\Gamma \vdash t\ u : \rho' \tag{1}$$

given that

$$\Gamma \vdash t : \boxed{\sigma_a} \to \rho' \tag{2}$$

$$\Gamma \vdash^{poly} u : \sigma_a \tag{3}$$

We have two cases for term $t\ u$:

- Case $\neg\ VSIG(t\ u, \Gamma)$. In this case we are easily done by the induction hypothesis and rule APP.
- Case $VSIG(t\ u, \Gamma)$. In other words it is the case that $(t\ u = (\nu\ \overline{u}^n\ u)$, $\nu \overset{s}{:} \forall \overline{a}.\overline{\sigma}^{n+1} \to \sigma \in \Gamma)$, and $n + 1 = |\overline{\sigma}|$, where $\overline{u}^n$ simply states that $|\overline{u}| = n$. By induction hypothesis it is the case that $\Gamma \vdash\!\blacktriangleright t : \boxed{\sigma} \to \rho'$. Moreover it must also be that $VSIG(t, \Gamma)$, which means that it must have been the case that:

$$\nu \overset{s}{:} \forall \overline{a}.\overline{\sigma}^n \to \sigma_{n+1} \to \sigma \in \Gamma \tag{4}$$

$$\overline{a}_c = \overline{a} \cap ftv(\sigma_{n+1} \to \sigma) \quad \overline{a}_e = \overline{a} - \overline{a}_c \tag{5}$$

$$\vdash\!\blacktriangleright [\overline{a_c \mapsto \boxed{\sigma_c}}](\sigma_{n+1} \to \sigma) \leq \boxed{\sigma_a} \to \rho' \tag{6}$$

$$\Gamma \vdash\!\blacktriangleright^{poly} u_i : [\overline{a_e \mapsto \boxed{\sigma_e}}, \overline{a_c \mapsto \sigma_c}]\sigma_i \tag{7}$$

Now consider the following split of $\overline{a}_c = \overline{a}_{c1} \cup \overline{a}_{c2}$ such that $\overline{a}_{c1} = \overline{a} \cap ftv(\sigma)$, and $\overline{a}_{c2} = \overline{a}_c - \overline{a}_{c1}$. Moreover let $\overline{a}_{e1} = \overline{a} - \overline{a}_{c1} = \overline{a}_e \cup \overline{a}_{c2}$. Then equation 6 with inversion gives:

$$\vdash\!\blacktriangleright [\overline{a_{c1} \mapsto \boxed{\sigma_{c1}}}]\sigma \leq \rho' \tag{8}$$

$$\vdash\!\blacktriangleright [\overline{a_{c2} \mapsto \boxed{\sigma_{c2}}}, \overline{a_{c1} \mapsto \boxed{\sigma_{c1}}}]\sigma_{n+1) \sim \boxed{\sigma_a}} \tag{9}$$

But this implies as well that $\overline{\sigma_{c2}}$ are all monotypes $\overline{\tau_{c2}}$. Equation 7 becomes

$$\Gamma \vdash\!\blacktriangleright^{poly} u_i : [\overline{a_e \mapsto \boxed{\sigma_e}}, \overline{a_{c2} \mapsto \tau_{c2}}, \overline{a_{c1} \mapsto \sigma_{c1}}]\sigma_i \tag{10}$$

And using the monotype boxing (for $\vdash\!\blacktriangleright$) we get

$$\Gamma \vdash\!\blacktriangleright^{poly} u_i : [\overline{a_e \mapsto \boxed{\sigma_e}}, \overline{a_{c2} \mapsto \boxed{\tau_{c2}}}, \overline{a_{c1} \mapsto \sigma_{c1}}]\sigma_i \tag{11}$$

To finish the case we just have to show that

$$\Gamma \vdash\!\blacktriangleright^{poly} u : [\overline{a_e \mapsto \boxed{\sigma_e}}, \overline{a_{c2} \mapsto \boxed{\tau_{c2}}}, \overline{a_{c1} \mapsto \sigma_{c1}}]\sigma_{n+1} \tag{12}$$

but $\overline{a}_e \# ftv(\sigma_{n+1})$ therefore we just have to show that

$$\Gamma \vdash\!\blacktriangleright^{poly} u : [\overline{a_{c2} \mapsto \boxed{\tau_{c2}}}, \overline{a_{c1} \mapsto \sigma_{c1}}]\sigma_{n+1} \tag{13}$$

and by the controlled boxing lemma (for $\vdash\!\blacktriangleright$) we just need to show that

$$\Gamma \vdash\!\blacktriangleright^{poly} u : [\overline{a_{c2} \mapsto \tau_{c2}}, \overline{a_{c1} \mapsto \sigma_{c1}}]\sigma_{n+1} \tag{14}$$

But from equation 9, those $\overline{a}_{c1}$ appearing in σ_{n+1} must be mapped to monotypes, therefore, and by monotype unboxing for subsumption we get that actually

$$[\overline{a_{c2} \mapsto \tau_{c2}}, \overline{a_{c1} \mapsto \sigma_{c1}}]\sigma_{n+1} = \sigma_a \tag{15}$$

And we are done by observing that we already have (3).

□

Theorem 8.17.

1. *If* $\Gamma \vdash\!\blacktriangleright t : \rho'$ *then* $\Gamma \vdash t : \rho'$.
2. *If* $\Gamma \vdash\!\blacktriangleright^{poly} t : \sigma'$ *then* $\Gamma \vdash^{poly} t : \sigma'$.

Proof. Straightforward induction (all rules are admissible). □

8.3.2 Unification

So far we have established that there exists an algorithmic version of our type system, which will also guide the implementation. From this section onwards we describe such an implementation, starting by presenting first-order unification. Although the results on unification are well known, for self-containment of this document we present them here as well.

Lemma 8.18 (Unification soundness). *If* $S_0 \succ \tau_1 \doteq \tau_2 \succ S_1$ *and* $\vdash S_0$ *then* $S_1\tau_1 = S_1\tau_2$, $\vdash S_1$, *and* $\exists R.\, S_1 = R \cdot S_0$ *with* $vars(R) \subseteq ftv(S_0\tau_1, S_0\tau_2)$.

Proof. By induction on the height of the derivation. We proceed with case analysis on the last rule used in the derivation.

- Case UREFL. Trivial, just take $R = \emptyset$.
- Case BVAR1 (case BVAR2 is similar). Here
$$S_0 \succ \vdash \alpha \doteq \tau \succ S_1 \tag{1}$$
given that
$$\alpha \in dom(S_0) \quad \tau \neq \alpha \tag{2}$$
$$S_0 \succ \vdash S_0\alpha \doteq \tau \succ S_1 \tag{3}$$
By induction hypothesis $\vdash S_1$ and $\exists R.\, S_1 = R \cdot S_0$ with $vars(R) \subseteq ftv(S_0S_0\alpha, S_0\tau_2)$, or since $\vdash S_0$ $vars(R) \subseteq ftv(S_0\alpha, S_0\tau_2)$ as required. Moreover $S_1S_0\alpha = S_1\tau$ or $RS_0S_0\alpha = S_1\tau$ or $RS_0\alpha = S_1\tau$, or $S_1\alpha = S_1\tau$.
- Case UVAR1 (case UVAR2 is similar). We have
$$S_0 \succ \vdash \alpha \doteq \tau \succ [\alpha \mapsto S_0\tau] \cdot S_0 \tag{4}$$
given that
$$\alpha \notin dom(S_0) \quad \tau \neq \alpha \tag{5}$$
$$\alpha \notin ftv(S_0\tau) \tag{6}$$
Then it is the case that $[\alpha \mapsto S_0\tau]S_0\alpha = [\alpha \mapsto S_0\tau]\tau$ since $\alpha \notin ftv(\tau)$ (otherwise $\alpha \in ftv(S_0\tau)$ as well). Moreover take $R = [\alpha \mapsto S_0\tau]$; then $vars(R) \subseteq ftv(S_0\alpha, S_0\tau)$. Finally to show that $\vdash (R \cdot S_0)$ it is enough to show that
$$dom([\alpha \mapsto S_0\tau] \cdot S_0) \# range([\alpha \mapsto S_0\tau] \cdot S_0) \tag{7}$$
or
$$\alpha, dom(S_0) \# ftv(S_0\tau) \cup (range(S_0) - \{\alpha\}) \tag{8}$$
which is an easy check, given that $dom(S_0) \# range(S_0)$ as well.
- Case UFUN. In this case we have that
$$S_0 \succ \vdash \tau_1 \to \tau_2 \doteq \tau_3 \to \tau_4 \succ S_2 \tag{9}$$
given that
$$S_0 \succ \vdash \tau_1 \doteq \tau_3 \succ S_1 \tag{10}$$
$$S_1 \succ \vdash \tau_2 \doteq \tau_4 \succ S_2 \tag{11}$$
By induction for (10) we get that $\vdash S_1$, and $\exists R_1.\, S_1 = R_1 \cdot S_0$ with $vars(R_1) \subseteq ftv(S_0\tau_1, S_0\tau_3)$. Then, by induction for (11) we have $\vdash S_2$ and $\exists R_2.\, S_2 = R_2 \cdot S_1$, with $vars(R_2) \subseteq ftv(S_1\tau_2, S_1\tau_4)$. Take $R = R_2 \cdot R_1$. Then $vars(R) \subseteq ftv(S_1\tau_2, S_1\tau_4) \cup ftv(S_0\tau_1, S_0\tau_3)$. But $ftv(S_1\tau_2, S_1\tau_4) \subseteq vars(R) \cup ftv(S_0\tau_2, S_0\tau_4)$ which implies $ftv(S_1\tau_2, S_1\tau_4) \subseteq ftv(S_0\tau_1, S_0\tau_3) \cup ftv(S_0\tau_2, S_0\tau_4)$. Moreover by induction we get $S_1\tau_1 = S_1\tau_3$ and $S_2\tau_2 = S_2\tau_4$, and then $R_2S_1\tau_1 = R_2S_1\tau_3$ or $S_2\tau_1 = S_2\tau_3$ as well.

□

Lemma 8.19 (Unification determinacy). *If* $S_0 \succ \vdash \tau_1 \doteq \tau_2 \succ S_{1a}$ *and* $S_0 \succ \vdash \tau_1 \doteq \tau_2 \succ S_{1b}$ *then* $S_{1a} = S_{1b}$.

Proof. By induction on the first derivation. Just observe that for each rule used there is only one corresponding rule that can be used in the second derivation and the result follows in each case either directly, or by the inductive hypothesis. □

Definition 8.20. *Consider the following lexicographic pair to be a metric for a given unifier S_0, types τ_1 and τ_2.*

$$\mu \quad = \quad \langle |\, range(S_0) \cup ftv(\tau_1, \tau_2)\,|, size(\tau_1) + size(\tau_2) \rangle$$

Lemma 8.21. *If $\vdash S_0$ and $S_0 \succ \vdash \alpha \doteq \tau \succ S_1$, then $\tau = \alpha$ or $\alpha \notin ftv(\tau)$. Similarly if $S_0 \succ \vdash \tau \doteq \alpha \succ S_1$.*

Proof. By induction on the height of the derivation. The case UREFL is trivial and the case for UVAR1 (and UVAR2) cannot happen as $\alpha \notin dom(S_0)$, which would imply $\alpha \in ftv(S_0\tau)$. The case UFUN cannot happen. In the case for BVAR1 (and similarly BVAR2), if it were the case that $\alpha \in ftv(\tau)$ we would have by unification soundness that $S_1\alpha = S_1\tau$ which cannot happen as the two types would not have the same size. □

Lemma 8.22 (Unification termination). *Assuming that $\vdash S_0$, unification defines an algorithm that always terminates on inputs S_0, τ_1, and τ_2.*

Proof. We show that in any instance of the rules, the metric of the conclusion is strictly greater than the metric of any of the premises. Cases UREFL, UVAR1, UVAR2, are trivial. For the rest of the cases we have

- Case BVAR1 (case BVAR2 is similar). Here we have that $S_0 \succ \vdash \alpha \doteq \tau \succ S_1$ given that $\alpha \in dom(S_0)$ and $S_0 \succ \vdash S_0\alpha \doteq \tau \succ S_1$, $\tau \neq \alpha$. The metric of the conclusion is

  $$\mu_c \quad = \quad \langle |\, range(S_0) \cup ftv(\alpha, \tau)\,|, 1 + size(\tau) \rangle$$

  But by we know that $\alpha \in dom(S_0)$ therefore $\alpha \notin range(S_0)$, since $\vdash S_0$. Moreover by Lemma 8.21 $\alpha \notin ftv(\tau)$. Then the metric of the premise is

  $$\mu_p \quad = \quad \langle |\, range(S_0) \cup ftv(S_0\alpha, \tau)\,|, size(S_0\alpha) + size(\tau) \rangle$$

  Since $ftv(S_0\alpha) \subseteq range(S_0)$ and $\alpha \notin range(S_0)$, it follows that $\mu_p \lneq \mu_c$ according to the lexicographic ordering.
- Case UFUN. Here

  $$S_0 \succ \vdash \tau_1 \to \tau_2 \doteq \tau_3 \to \tau_4 \succ S_2$$

  given that

  $$S_0 \succ \vdash \tau_1 \doteq \tau_3 \succ S_1$$
  $$S_1 \succ \vdash \tau_2 \doteq \tau_4 \succ S_2$$

  The metrics of the conclusion and the premises are:

  $$\begin{aligned} \mu_c \quad &= \quad \langle |\, range(S_0) \cup ftv(\tau_1, \tau_2, \tau_3, \tau_4)\,|, \\ &\qquad size(\tau_1 \to \tau_2) + size(\tau_3 \to \tau_4) \rangle \\ \mu_{p1} \quad &= \quad \langle |\, range(S_0) \cup ftv(\tau_1, \tau_3)\,|, size(\tau_1) + size(\tau_3) \rangle \\ \mu_{p2} \quad &= \quad \langle |\, range(S_1) \cup ftv(\tau_2, \tau_4)\,|, size(\tau_1) + size(\tau_3) \rangle \end{aligned}$$

  It is obvious that $\mu_{p1} \lneq \mu_c$. To see that $\mu_{p2} \lneq \mu_c$ just observe that by unification soundness, Lemma 8.18, $S_1 = R \cdot S_0$ with $vars(R) \subseteq ftv(S_0\tau_1, S_0\tau_2)$, or $range(S_1) \subseteq range(S_0) \cup ftv(\tau_1, \tau_2)$.

□

Lemma 8.23 (Unification completeness). *If $\vdash S_0$ and $SS_0\tau_1 = SS_0\tau_2$ then $S_0 \succ \vdash \tau_1 \doteq \tau_2 \succ S_1$ and $\exists R.\, S \cdot S_0 = R \cdot S_1$.*

Proof. By induction on the metric $\mu(S_0, \tau_1, \tau_2)$. We proceed by case analysis on the structure of τ_1 and τ_2. Consider the following logical splitting of cases:

- Case $\tau_1 = \alpha$, $\tau_2 \neq \alpha$. Here we have two subcases.
  - If $\alpha \notin dom(S_0)$ then it is easy to see that $\alpha \notin ftv(S_0\tau_2)$, otherwise it could not possibly be that $SS_0\alpha = SS_0\tau_2$. Then by applying rule UVAR1 we get that $S_0 \succ \vdash \alpha = \tau_2 \succ [\alpha \mapsto S_0\tau_2] \cdot S_0$. Now it must be that $\alpha \in dom(S)$ and we claim that $S \cdot S_0$ can be written as $R \cdot [\alpha \mapsto S_0\tau_2]S_0$. Just take R to be the restriction of S where $\alpha \notin dom(R)$. Then it is the case that $R \cdot [\alpha \mapsto S_0\tau_2]S_0(\alpha) = RS_0\tau_2 = SS_0\tau_2$ since $\alpha \notin ftv(S_0\tau_2)$. On the other hand, $R \cdot [\alpha \mapsto S_0\tau_2]S_0(\alpha') = SS_0(\alpha')$ as well.
  - If $\alpha \in dom(S_0)$ then we can apply rule BVAR1 and by induction hypothesis (see proof of Lemma 8.22 to see why the metric reduces) we get the result.

- Case $\tau_1 = \alpha$, $\tau_2 = \alpha$. We can use UREFL and taking $R = S$ we are done.
- Case $\tau_1 = a, \tau_{11} \to \tau_{12}$, $\tau_2 = \alpha$. Similarly to the first case either UVAR2 or BVAR2 is applicable.
- Case $\tau_1 = a_1$, $\tau_2 = a_2$. In this case it must also be that $a_1 = a_2 = a$. By UREFL and taking $R = S$ we are done.
- Case $\tau_1 = \tau_{11} \to \tau_{12}$, $\tau_2 = \tau_{21} \to \tau_{22}$. Applying rule UFUN and noticing (see proof of Lemma 8.22) by induction that $S \cdot S_0 = R_1 \cdot S_1$ for some R_1 and that $R_1 \cdot S_1 = R_2 \cdot S_2$ again by induction we finally get that $S \cdot S_0 = R_2 \cdot S_2$ as required.
- Case $\tau_1 = a_1$, $\tau_2 = \tau_{21} \to \tau_{22}$. Cannot happen.
- Case $\tau_1 = \tau_{11} \to \tau_{12}$, $\tau_2 = a_2$. Cannot happen.

□

Lemma 8.24 (Arrow unification soundness and completeness). *The following are true:*

1. *If* $\mathcal{A}_0 \# ftv(\rho'), vars(S_0)$, $\vdash S_0$, *and* $(S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho' \doteq \sigma'_1 \to \sigma'_2 \succ (S_1, \mathcal{A}_1)$ *then* $\vdash S_1$, $S_1\sigma'_1 \to S_1\sigma'_2 = S_1\rho'$, $\exists R . S_1 = R \cdot S_0$, *and* $vars(R) \subseteq ftv(S_0\rho') \cup (\mathcal{A}_0 - \mathcal{A}_1)$
2. *If* $\vdash S, S_0$, $\rho' \neq \zeta$, *and* $SS_0\rho' = \sigma'_1 \to \sigma'_2$, *and* $\mathcal{A}_0 \# vars(S, S_0), ftv(\rho')$ *then* $(S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho' \doteq \sigma'_3 \to \sigma'_4 \succ (S_1, \mathcal{A}_1)$, $\exists R . S \cdot S_0 = R \cdot S_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$, *and* $RS_1(\sigma'_3 \to \sigma'_4) = \sigma'_1 \to \sigma'_2$.

Proof. Easy induction appealing to unification soundness and completeness, Lemmas 8.18 and 8.23 respectively. □

Lemma 8.25 (Arrow unification termination and determinacy). *Arrow unification defines a deterministic, terminating algorithm on inputs* S_0, $\mathcal{A}_0$, *and* ρ' *when* $\vdash S_0$, *and* $\mathcal{A}_0 \# ftv(\rho'), vars(S_0)$.

Proof. Easy check appealing to Lemma 8.19 and Lemma 8.22. □

8.3.3 Boxy matching and equivalence

Lemma 8.26 (Boxy matching fills holes). *If* $(S_0, \mathcal{A}_0) \succ \vdash \sigma'_1 \sim \sigma'_2 \succ (S_1, \mathcal{A}_1)$ *then* $fbv(\sigma'_1, \sigma'_2) \subseteq dom(S_1)$.

Proof. Straightforward induction. □

Lemma 8.27 (Boxy matching and filling soundness). *The following are true:*

1. *If* $\mathcal{A}_0 \# ftv(\sigma'_1, \sigma'_2), vars(S_0)$, $\vdash S_0$, $(S_0, \mathcal{A}_0) \succ \vdash \sigma'_1 \sim \sigma'_2 \succ (S_1, \mathcal{A}_1)$ *then* $\vdash [\![\sigma'_1]\!]_{S_1} \sim [\![\sigma'_2]\!]_{S_1}$, $\vdash S_1$, *and* $\exists R . S_1 = R \cdot S_0$ *with* $vars(R) \subseteq ftv(S_0\sigma'_1, S_0\sigma'_2) \cup (\mathcal{A}_0 - \mathcal{A}_1)$.
2. *If If* $\mathcal{A}_0 \# ftv(\zeta, \sigma), vars(S_0)$, $\vdash S_0$, $(S_0, \mathcal{A}_0) \succ \vdash \zeta \leftarrow \sigma \succ (S_1, \mathcal{A}_1)$ *then* $\vdash \boxed{S_1\zeta} \sim S_1\sigma$, $\vdash S_1$, *and* $\exists R . S_1 = R \cdot S_0$ *with* $vars(R) \subseteq ftv(S_0\zeta, S_0\sigma) \cup (\mathcal{A}_0 - \mathcal{A}_1)$.

Proof. The proof is by straightforward induction on the height of the derivations. The two claims are proved simultaneously. We only give two interesting cases of the first part; the rest cases are in the same style.

- Case AEQ2. We have in this case that

$$\mathcal{A}_0 \# ftv(\sigma'_1, \sigma'_2, \rho'), vars(S_0) \tag{1}$$
$$\vdash S_0 \tag{2}$$
$$(S_0, \mathcal{A}_0) \succ \vdash \sigma'_1 \to \sigma'_2 \sim \rho' \succ (S_1, \mathcal{A}_1) \tag{3}$$

given that

$$(S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho' \doteq \sigma'_3 \to \sigma'_4 \succ (S_1, \mathcal{A}_1) \tag{4}$$
$$(S_1, \mathcal{A}_1) \succ \vdash \sigma'_1 \sim \sigma'_3 \succ (S_2, \mathcal{A}_2) \tag{5}$$
$$(S_2, \mathcal{A}_2) \succ \vdash \sigma'_2 \sim \sigma'_4 \succ (S_3, \mathcal{A}_3) \tag{6}$$

By (1) and (2) and arrow unification soundness, Lemma 8.24, we get that

$$\vdash S_1 \tag{7}$$
$$S_1\rho' = S_1\sigma'_3 \to S_1\sigma'_4 \tag{8}$$

Moreover $S_1 = R_1 S_0$ with $vars(R_1) \subseteq ftv(S_0\rho') \cup (\mathcal{A}_0 - \mathcal{A}_1)$. Then it is easy to confirm as well that $\mathcal{A}_1 \# ftv(\sigma'_1, \sigma'_3), vars(S_1)$ and by induction hypothesis for (5) we get that

$$\vdash [\![\sigma'_1]\!]_{S_2} \sim [\![\sigma'_3]\!]_{S_2} \tag{9}$$

Moreover, $\vdash S_2$ and $S_2 = R_2 S_1$ such that $vars(R_2) \subseteq ftv(S_1\sigma_1', S_1\sigma_3') \cup (\mathcal{A}_1 - \mathcal{A}_2)$. Then it will also be that $\mathcal{A}_2 \# ftv(\sigma_2', \sigma_4')$, $vars(S_2)$ and by induction for (6) we get

$$\vdash [\![\sigma_2']\!]_{S_3} \sim [\![\sigma_4']\!]_{S_3} \tag{10}$$

Additionally $S_3 = R_3 S_2$ with $vars(R_3) \subseteq ftv(S_2\sigma_2', S_2\sigma_4') \cup (\mathcal{A}_2 - \mathcal{A}_3)$. From (9), and the subsumption substitution lemma with the fact that all $fbv(\sigma_1', \sigma_3') \subseteq dom(S_2)$ by Lemma 8.26 it will also be that

$$\vdash [\![\sigma_1']\!]_{S_3} \sim [\![\sigma_3']\!]_{S_3} \tag{11}$$

From (10) and (11) and rule AEQ2 we get that

$$\vdash [\![\sigma_1' \to \sigma_3']\!]_{S_3} \sim [\![\sigma_3' \to \sigma_4']\!]_{S_3} \tag{12}$$

But observe that, using (8) it must also be that $[\![\sigma_3' \to \sigma_4']\!]_{S_3} = [\![\rho']\!]_{S_3}$. To finish the case just pick $R = R_3 \cdot R_2 \cdot R_1$ and the rest is easy check.

- Case ASEQ1L. In this case we have that

$$\mathcal{A}_0 \# ftv(\xi, \forall \overline{a}.\rho'), vars(S_0) \tag{13}$$
$$\vdash S_0 \tag{14}$$
$$(S_0, \mathcal{A}_0 \overline{b} \zeta_1) \succ \vdash \xi \sim \forall \overline{a}.\rho' \succ (S_2, \mathcal{A}_2) \tag{15}$$

given that

$$|\overline{a}| \neq 0 \tag{16}$$
$$(S_0, \mathcal{A}_0) \succ \vdash \zeta_1 \sim [\overline{a \mapsto b}]\rho' \succ (S_1, \mathcal{A}_1) \tag{17}$$
$$\overline{b} \# ftv(S_1(\forall \overline{a}.\rho')) \tag{18}$$
$$(S_1, \mathcal{A}_1) \succ \vdash \xi \leftarrow \forall \overline{b}.S_1\zeta_1 \succ (S_2, \mathcal{A}_2) \tag{19}$$

By induction hypothesis for (17) we get that

$$\vdash \boxed{S_1\zeta_1} \sim [\![\overline{[a \mapsto b]}\rho']\!]_{S_1} \tag{20}$$
$$\vdash S_1 \quad S_1 = R_1 S_0 \tag{21}$$
$$vars(R_1) \subseteq ftv(S_0\zeta_1, S_0(\overline{[a \mapsto b]}\rho')) \cup (\mathcal{A}_0 - \mathcal{A}_1) \tag{22}$$

By induction hypothesis for (19) (second part) we get that

$$\vdash \boxed{S_2\xi} \sim S_2(\forall \overline{b}.S_1\zeta_1) \tag{23}$$
$$\vdash S_2 \quad S_2 = R_2 S_1 \tag{24}$$
$$vars(R_2) \subseteq ftv(S_1\xi, S_1(\forall \overline{b}.S_1\zeta_1)) \cup (\mathcal{A}_1 - \mathcal{A}_2) \tag{25}$$

Applying the substitution lemma to (20) we get that

$$\vdash \boxed{S_2\zeta_1} \sim [\![\overline{[a \mapsto b]}\rho']\!]_{S_2} \tag{26}$$

and applying symmetry and SEQ1 we are done noticing that condition (18) allows us to α-convert $[\![\forall \overline{a}.\rho']\!]_{S_2}$ to $[\![\forall \overline{b}.\overline{[a \mapsto b]}\rho']\!]_{S_2}$. Taking $R = R_2 \cdot R_1$ finishes the case and the rest is again easy check.

□

Lemma 8.28 (Boxy matching and filling completeness).

1. *If* $\mapsto [\![\sigma_1']\!]_{SS_0} \sim [\![\sigma_2']\!]_{SS_0}$, $\vdash S_0$, *and* $\mathcal{A}_0 \# vars(S, S_0), ftv(\sigma_1', \sigma_2')$ *then* $(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \sim \sigma_2' \succ (S_1, \mathcal{A}_1)$, *and* $\exists R.S \cdot S_0 = R \cdot S_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1} \cdot$
2. *If* $\mapsto [\![\xi]\!]_{SS_0} \sim [\![\sigma]\!]_{SS_0}$, $\vdash S_0$, *and* $\mathcal{A}_0 \# vars(S, S_0), ftv(\xi, \sigma)$ *then* $(S_0, \mathcal{A}_0) \succ \vdash \alpha \leftarrow \sigma \succ (S_1, \mathcal{A}_1)$, *and* $\exists R.S \cdot S_0 = R \cdot S_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1} \cdot$

Proof. We just give a sketch. We first prove the first claim in the special case where σ_1', σ_2' are box-free, by induction on the height of the derivations. Then, this is used to prove the second claim. Finally the full version of the first claim is proved by induction on the height of the derivations again. The case analysis is on the last rule used in the specification derivation. We give one interesting case, the case of BBEQ. In this case we have that $[\![\sigma_1']\!]_{SS_0} = [\![\sigma_2']\!]_{SS_0} = \boxed{a}$. This in turn implies that $\sigma_1' = \zeta_1$, and $\sigma_2' = \zeta_2$. Now it must also be (by MEQ3) that also

$$\vdash [\![\zeta_1]\!]_{SS_0} \sim a \tag{1}$$
$$\vdash [\![\zeta_2]\!]_{SS_0} \sim a \tag{2}$$

But then, the sums of sizes of types appearing in (1) and (2) is smaller. Now take a $S' = [\alpha \mapsto a] \cdot S$ and equations (1) and (2) can be rewritten as

$$\vdash [\![\zeta_1]\!]_{S'S_0} \sim S'S_0\alpha \tag{3}$$
$$\vdash [\![\zeta_2]\!]_{S'S_0} \sim S'S_0\alpha \tag{4}$$

Then from the second part for (3) we get that

$$(S_0, \mathcal{A}_0) \succ \zeta_1 \leftarrow \alpha \succ (S_1, \mathcal{A}_1) \tag{5}$$

and moreover $S'S_0 = R_1 \cdot S_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$. We can rewrite equation (4) as

$$\vdash [\![\zeta_2]\!]_{R_1 S_1} \sim R_1 S_1 \alpha \tag{6}$$

and it is easy to confirm the preconditions for the induction hypothesis. Hence we get

$$(S_1, \mathcal{A}_1) \succ \zeta_2 \leftarrow \alpha \succ (S_2, \mathcal{A}_2) \tag{7}$$

as required, and moreover $R_1 S_1 = R_2 S_2 \backslash_{\mathcal{A}_1 - \mathcal{A}_2}$, giving finally $S'S_0 = R_2 S_2 \backslash_{\mathcal{A}_0 - \mathcal{A}_2}$ or $SS_0 = R_2 S_2 \backslash_{\mathcal{A}_0 \alpha - \mathcal{A}_2}$ as required. Applying rule ABBEQ finishes the case. □

Lemma 8.29 (Boxy matching terminates for box-free types). *The judgement* $(S_0, \mathcal{A}_0) \succ \vdash \sigma_1 \sim \sigma_2 \succ (S_1, \mathcal{A}_1)$, *when* $\vdash S_0$ *and* $\mathcal{A}_0 \# ftv(\sigma_1, \sigma_2), vars(S_0)$ *defines a terminating algorithm.*

Proof. Straightforward induction on the sum of sizes of the two types. If the rule applicable is AMEQ2 then this follows from unification termination. In the cases of AEQ2L and AEQ2R the metric becomes smaller, as is also the case for ASEQ2. The rest of the cases cannot happen. □

Lemma 8.30 (Filling termination). *The judgement* $(S_0, \mathcal{A}_0) \succ \vdash \alpha \leftarrow \sigma \succ (S_1, \mathcal{A}_1)$, *when* $\vdash S_0$ *and* $\mathcal{A}_0 \# ftv(\sigma, \alpha), vars(S_0)$ *defines a terminating algorithm.*

Proof. Follows by Lemma 8.29. □

Lemma 8.31 (Matching termination and determinacy). *The judgement* $(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \sim \sigma_2' \succ (S_1, \mathcal{A}_1)$ *defines a deterministic, terminating algorithm on inputs* S_0,$\mathcal{A}_0$,σ_1', *and* σ_2', *when* $\vdash S_0$ *and* $\mathcal{A}_0 \# ftv(\sigma_1', \sigma_2'), vars(S_0)$.

Proof. It is straightforward to check determinacy. For termination it is also straightforward to see that either after at most two steps the metric $size(\sigma_1') + size(\sigma_2')$ reduces, or the algorithm terminates immediately, appealing to Lemma 8.22 or Lemma 8.30. □

8.3.4 Subsumption

Lemma 8.32 (Subsumption fills holes). *If* $(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \leq \sigma_2' \succ (S_1, \mathcal{A}_1)$ *then* $fbv(\sigma_1', \sigma_2') \subseteq dom(S_1)$.

Proof. Straightforward induction. □

Lemma 8.33 (Subsumption soundness). *If* $\mathcal{A}_0 \# ftv(\sigma_1', \sigma_2'), vars(S_0), \vdash S_0$, *and* $(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \leq \sigma_2' \succ (S_1, \mathcal{A}_1)$ *then* $\vdash [\![\sigma_1']\!]_{S_1} \leq [\![\sigma_2']\!]_{S_1}$, $\vdash S_1$, *and* $\exists R . S_1 = R \cdot S_0$ *with* $vars(R) \subseteq ftv(S_0\sigma_1', S_0\sigma_2') \cup (\mathcal{A}_0 - \mathcal{A}_1)$.

Proof. Easy induction on the height of the algorithm derivations. We only show one case, the case for ASPEC. Here we have that

$$\mathcal{A}_0\overline{\xi}\#ftv(\forall\overline{a}\,.\,\sigma_1', \rho_2'), vars(S_0) \tag{1}$$
$$\vdash S_0 \tag{2}$$
$$(S_0, \mathcal{A}_0)\succ\vdash \forall\overline{a}\,.\,\rho_1' \leq \rho_2'\succ(S_1, \mathcal{A}_1) \tag{3}$$

given that

$$|\,\overline{a}\,|\neq 0 \tag{4}$$
$$(S_0, \mathcal{A}_0)\succ\vdash \overline{[a \mapsto \xi]}\rho_1' \;\leq\; \rho_2'\succ(S_1, \mathcal{A}_1) \tag{5}$$

By induction hypothesis we get that

$$\vdash [\![\overline{[a \mapsto \xi]}\rho_1']\!]_{S_1} \leq [\![\rho_2']\!]_{S_1} \tag{6}$$

which can be rewritten as

$$\vdash \overline{[a \mapsto \boxed{S_1\xi}]}[\![\rho_1']\!]_{S_1} \leq [\![\rho_2']\!]_{S_2} \tag{7}$$

Moreover $\exists R\,.\,S_1 = R\cdot S_0$ with $vars(R) \subseteq ftv(S_0[\overline{a \mapsto \xi}]\rho_1', S_0(\rho_2'))\cup(\mathcal{A}_0 - \mathcal{A}_1)$ or $vars(R) \subseteq ftv(S_0(\forall\overline{a}\,.\,\rho_1'), S_0(\rho_2'))\cup(\mathcal{A}_0\overline{\xi} - \mathcal{A}_1)$ as required. By applying rule SPEC we are done. ☐

Lemma 8.34 (Subsumption completeness).

If $\mapsto[\![\sigma_1']\!]_{SS_0} \leq [\![\sigma_2']\!]_{SS_0}$, $\vdash S_0$, *and* $\mathcal{A}_0\#vars(S, S_0), ftv(\sigma_1', \sigma_2')$ *then* $(S_0, \mathcal{A}_0)\succ\vdash \sigma_1' \leq \sigma_2'\succ(S_1, \mathcal{A}_1)$ *and* $\exists R\,.\,S\cdot S_0 = R\cdot S_1\backslash_{\mathcal{A}_0-\mathcal{A}_1}$.

Proof. Straightforward induction on the height of the derivations. We only show one interesting case, the case of SPEC.

- Case SPEC. Here it must be that σ_1' is a polytype and assume that $\sigma_1' = \forall\overline{a}\,.\,\rho_1'$ and $\sigma_2' = \rho_2'$. Assume also without loss of generality that $\overline{a}\notin vars(S, S_0)$. Then we have that

  $$\mapsto\forall\overline{a}\,.\,[\![\rho_1']\!]_{SS_0} \leq \rho_2' \tag{1}$$
  $$\vdash S_0 \tag{2}$$
  $$\mathcal{A}_0\overline{\xi}\#vars(S, S_0), ftv(\forall\overline{a}\,.\,\rho_1', \rho_2') \tag{3}$$

  given that

  $$|\,\overline{a}\,|\neq 0 \quad \mapsto\overline{[a \mapsto \boxed{\sigma}]}[\![\rho_1']\!]_{SS_0} \leq [\![\rho_2']\!]_{SS_0} \tag{4}$$

  Now consider $S' = [\overline{\xi \mapsto \sigma}]\cdot S$. Then equation 4 can be rewritten as

  $$\mapsto[\![\overline{[a \mapsto \xi]}\rho_1']\!]_{S'S_0} \leq [\![\rho_2']\!]_{S'S_0} \tag{5}$$

  By induction hypothesis (by (4) $ftv(\sigma)\subseteq vars(S, S_0), ftv(\rho_2')$, therefore $ftv(\sigma)\#\mathcal{A}_0$) we get:

  $$(S_0, \mathcal{A}_0)\succ\vdash \overline{[a \mapsto \xi]}\rho_1' \;\leq\; \rho_2'\succ(S_1, \mathcal{A}_1) \tag{6}$$

  with $S'S_0 = RS_1\backslash_{\mathcal{A}_0-\mathcal{A}_1}$, or $SS_0 = RS_1\backslash_{\mathcal{A}_0\overline{\xi}-\mathcal{A}_1}$. Applying rule ASPEC finishes the case.
- Case SKOL. For this case we have that $\mapsto[\![\sigma_1']\!]_{SS_0} \leq [\![\sigma_2']\!]_{SS_0}$, given that

  $$[\![\sigma_2']\!]_{SS_0} = \forall\overline{b}\,.\,\rho_2' \tag{7}$$
  $$\overline{b}\notin ftv([\![\sigma_1']\!]_{SS_0}) \tag{8}$$
  $$|\,\overline{b}\,|\neq 0 \tag{9}$$
  $$[\![\sigma_1']\!]_{SS_0} \neq \boxed{\sigma} \tag{10}$$
  $$\mapsto[\![\sigma_1']\!]_{SS_0} \leq \rho_2' \tag{11}$$

  Take an appropriate supply $\mathcal{A}_0\overline{c}$ such that

  $$\mathcal{A}_0\overline{c}\#vars(S, S_0)\cup ftv(\sigma_1', \sigma_2') \tag{12}$$

Let as well $\sigma_2' = \forall \overline{a}\,.\,\rho_{2a}'$. Then

$$[\![\forall \overline{c}\,.\,[\overline{a \mapsto c}]\rho_{2a}']\!]_{SS_0} = \forall \overline{c}\,.\,[\![[\overline{a \mapsto c}]\rho_{2a}']\!]_{SS_0} \tag{13}$$

and it must be that

$$[\![[\overline{a \mapsto c}]\rho_{2a}']\!]_{SS_0} = [\overline{b \mapsto c}]\rho_2' \tag{14}$$

It is easy to see that the substitution lemma holds for $\mapsto$ subsumption and moreover renaming substitutions also preserve the height of the derivation (easy induction). Then from (11) we also get $\mapsto [\![\sigma_1']\!]_{SS_0} \leq [\overline{b \mapsto c}]\rho_2'$ with the same height, and by induction hypothesis we have:

$$(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \leq [\overline{a \mapsto c}]\rho_{2a}' \succ (S_1, \mathcal{A}_1) \tag{15}$$

Moreover $SS_0 = RS_1 \setminus_{\mathcal{A}_0 - \mathcal{A}_1}$. Then it must also be that $\overline{c} \# ftv(S_1\sigma_1', S_1(\forall \overline{a}\,.\,\rho_{2a}'))$, otherwise there would be a $c \in \overline{c}$ also in $ftv(S_1\sigma_1', S_1(\forall \overline{a}\,.\,\rho_{2a}'))$, or $ftv(RS_1\sigma_1', RS_1(\forall \overline{a}\,.\,\rho_{2a}'))$, or $ftv(SS_0\sigma_1', SS_0(\forall \overline{a}\,.\,\rho_{2a}'))$, a contradiction to (12). Applying then rule ASKOL finishes the case.

□

Lemma 8.35 (Subsumption termination and determinacy). *The judgement* $(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \leq \sigma_2' \succ (S_1, \mathcal{A}_1)$ *defines a deterministic, terminating algorithm on inputs* S_0,$\mathcal{A}_0$,σ_1', *and* σ_2', *when* $\vdash S_0$ *and* $\mathcal{A}_0 \# ftv(\sigma_1', \sigma_2'), vars(S_0)$.

Proof. Determinacy is an easy check. Moreover the sum of sizes of two types becomes smaller after at most two steps, or the algorithm terminates because of the auxiliary judgements termination (by soundness lemmas, all unifiers created are well-formed, and therefore termination for boxy matching and filling holds). □

8.3.5 Main algorithm

Lemma 8.36 (Main algorithm fills holes).

1. *If* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S_1, \mathcal{A}_1)$ *then* $fbv(\rho') \subseteq dom(S_1)$.
2. *If* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash^{poly} t : \sigma' \succ (S_1, \mathcal{A}_1)$ *then* $fbv(\sigma') \subseteq dom(S_1)$.

Proof. Straightforward induction. □

Theorem 8.37 (Soundness). *The following are true:*

1. *If* $\mathcal{A}_0 \# ftv(\Gamma, \rho'), vars(S_0), \vdash S_0$, *and* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S_1, \mathcal{A}_1)$ *then* $[\![\Gamma]\!]_{S_1} \vdash t : [\![\rho']\!]_{S_1}, \vdash S_1$, *and* $\exists R\,.\,S_1 = R \cdot S_0$ *with* $vars(R) \subseteq ftv(S_0\rho', S_0\Gamma) \cup (\mathcal{A}_0 - \mathcal{A}_1)$.
2. *If* $\mathcal{A}_0 \# ftv(\Gamma, \sigma'), vars(S_0), \vdash S_0$, *and* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash^{poly} t : \sigma' \succ (S_1, \mathcal{A}_1)$ *then* $[\![\Gamma]\!]_{S_1} \vdash^{poly} t : [\![\sigma']\!]_{S_1}, \vdash S_1$, *and* $\exists R\,.\,S_1 = R \cdot S_0$ *with* $vars(R) \subseteq ftv(S_0\sigma', S_0\Gamma) \cup (\mathcal{A}_0 - \mathcal{A}_1)$.

Proof. The two claims are proved simultaneously by induction on the height of the derivations, appealing to the substitution property of typing. We only show the case for LET. We have that

$$\mathcal{A}_0 \overline{a} \zeta \# ftv(\Gamma, \rho') \tag{1}$$

$$\vdash S_0 \tag{2}$$

$$(S_0, \mathcal{A}_0 \overline{a} \zeta) \succ \Gamma \vdash \texttt{let } x \texttt{ = } u \texttt{ in } t : \rho' \succ (S_2, \mathcal{A}_2) \tag{3}$$

given that

$$(S_0, \mathcal{A}_0) \succ \Gamma \vdash u : \zeta \succ (S_1, \mathcal{A}_1) \tag{4}$$

$$\overline{\alpha} = ftv(S_1\zeta) - ftv(S_1\Gamma) \tag{5}$$

$$(S_1, \mathcal{A}_1) \succ \Gamma, x \overset{i}{:} \forall \overline{a}\,.\,[\overline{\alpha \mapsto a}]S_1\zeta \vdash t : \rho' \succ (S_2, \mathcal{A}_2) \tag{6}$$

By induction hypothesis for (4) we get that

$$[\![\Gamma]\!]_{S_1} \vdash u : \boxed{S_1\zeta} \tag{7}$$

and $\vdash S_1$ and $\exists R_1\,.\,S_1 = R_1 \cdot S_0$ with $vars(R_1) \subseteq ftv(S_0\Gamma, \zeta) \cup (\mathcal{A}_0 - \mathcal{A}_1)$. It is easy to confirm that the induction hypothesis applies then to (5) and we get

$$[\![\Gamma, x \overset{i}{:} \forall \overline{a}\,.\,[\overline{\alpha \mapsto a}]S_1\zeta]\!]_{S_2} \vdash t : [\![\rho']\!]_{S_2} \tag{8}$$

with $\vdash S_2$, and $\exists R_2 \,.\, S_2 = R_2 \cdot S_1$ with

$$vars(R_2) \subseteq ftv(S_1\Gamma, S_1(\forall \overline{a} \,.\, [\overline{\alpha \mapsto a}]S_1\zeta), S_1\rho') \cup (\mathcal{A}_1 - \mathcal{A}_2) \tag{9}$$

which implies that $vars(R_2)\#\overline{a}$ as well. By (7) and the substitution lemma we get

$$[\![\Gamma]\!]_{S_2} \vdash u : \boxed{R_2[\overline{\alpha \mapsto a}]S_1\zeta} \tag{10}$$

Now it is not hard to confirm that $ftv(R_2[\overline{\alpha \mapsto a}]S_1\zeta) - ftv(S_2\Gamma) = \overline{a}$ and moreover $[\![\forall \overline{a} \,.\, [\overline{\alpha \mapsto a}]S_1\zeta]\!]_{S_2} = [\![\forall \overline{a} \,.\, [\overline{\alpha \mapsto a}]S_1\zeta]\!]_{R_2}$ and by applying rule LET we get the result. Taking $R = R_1 \cdot R_2$ finishes the case. The rest is easy check. □

Theorem 8.38 (Completeness). *The following are true:*

1. *If* $[\![\Gamma]\!]_{SS_0} \mapsto t : [\![\rho']\!]_{SS_0}$, $\vdash S_0$, *and* $\mathcal{A}_0\#vars(S, S_0), ftv(\Gamma, \rho')$ *then* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S_1, \mathcal{A}_1)$, *and* $\exists R \,.\, S \cdot S_0 = R \cdot S_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.
2. *If* $[\![\Gamma]\!]_{SS_0} \mapsto^{poly} t : [\![\sigma']\!]_{SS_0}$, $\vdash S_0$, *and* $\mathcal{A}_0\#vars(S, S_0), ftv(\Gamma, \sigma')$ *then* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash^{poly} t : \sigma' \succ (S_1, \mathcal{A}_1)$, *and* $\exists R \,.\, S \cdot S_0 = R \cdot S_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.

Proof. The two claims are proved simultaneously by induction on the size of the term t. We inline uses of the second part in the first so for the second part we can assume that the first always holds. For the second part the case for AGEN2 follows by the first part and the case for AGEN1 is similar to the case of SKOL in the completeness subsumption proof. For the first part the only interesting cases are the one for LET and the one for VAR-SIG.

- Case LET. In this case we have that

$$\mathcal{A}_0\overline{a}\zeta\#vars(S, S_0), ftv(\Gamma, \rho') \tag{1}$$
$$\vdash S_0 \tag{2}$$
$$[\![\Gamma]\!]_{SS_0} \mapsto \texttt{let } x \texttt{ = } u \texttt{ in } t : [\![\rho']\!]_{SS_0} \tag{3}$$

  given that

$$[\![\Gamma]\!]_{SS_0} \mapsto u : \boxed{\rho} \tag{4}$$
$$\overline{a} = ftv(\rho) - ftv([\![\Gamma]\!]_{SS_0}) \tag{5}$$
$$[\![\Gamma]\!]_{SS_0}, x \overset{i}{:} \forall \overline{a} \,.\, \rho \mapsto t : [\![\rho']\!]_{SS_0} \tag{6}$$

  Consider $S' = [\zeta \mapsto \rho] \cdot S$. Then we have by induction hypothesis for (4) (assume without loss of generality that the supply variables are disjoint from the variables of ρ) that

$$(S_0, \mathcal{A}_0) \succ \Gamma \vdash u : \zeta \succ (S_1, \mathcal{A}_1) \tag{7}$$

  and $S'S_0 = R_1S_1\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$. Moreover by soundness we have that $\vdash S_1$ and it is also the case that $\forall \overline{a} \,.\, \rho = \overline{SS_0\Gamma}(\rho)$. But notice that using Lemma 8.3 we have

$$\begin{array}{rcl} R_1S_1\overline{S_1\Gamma}(S_1\zeta) & \leq_{sh} & \overline{R_1S_1S_1\Gamma}(R_1S_1S_1\zeta) \\ & = & \overline{R_1S_1\Gamma}(R_1S_1\zeta) \\ & = & \overline{SS_0\Gamma}(R_1S_1\zeta) = \overline{SS_0\Gamma}(\rho) \end{array}$$

  By the weakening lemma then and equation (6) we get that

$$[\![\Gamma]\!]_{R_1S_1}, [\![x \overset{i}{:} \overline{S_1\Gamma}S_1\zeta]\!]_{R_1S_1} \mapsto t : [\![\rho']\!]_{R_1S_1} \tag{8}$$

  And by induction hypothesis we get that

$$(S_1, \mathcal{A}_1) \succ \Gamma, x \overset{i}{:} \overline{S_1\Gamma}S_1\zeta \vdash t : \rho' \succ (S_2, \mathcal{A}_2) \tag{9}$$

  with $R_1S_1 = R_2S_2\backslash_{\mathcal{A}_1 - \mathcal{A}_2}$ for some R_2. This gives us then that $SS_0 = R_2S_2\backslash_{\mathcal{A}_0\overline{a}\zeta - \mathcal{A}_2}$ as required and application of rule ALET finishes the case.

- Case VAR-SIG. We just give a sketch in this case. We have that

$$[\![\Gamma]\!]_{SS_0} \mapsto \nu\ \overline{u} : [\![\rho']\!]_{SS_0} \tag{10}$$

with $\vdash S_0$ and assuming a fresh enough $\mathcal{A}_0$, given that

$$VSIG(\nu\ \overline{u}, [\![\Gamma]\!]_{SS_0}) \tag{11}$$

$$\nu : \overset{s}{\forall} \overline{a}\,.\,\overline{\sigma}^n \to \sigma \in [\![\Gamma]\!]_{SS_0} \tag{12}$$

$$\overline{a}_c = \overline{a} \cap ftv(\sigma) \quad \overline{a}_e = \overline{a} - \overline{a}_c \tag{13}$$

$$\mapsto [\overline{a_c \mapsto \boxed{\sigma_c}}]\sigma \leq [\![\rho']\!]_{SS_0} \tag{14}$$

$$\Gamma \mapsto^{poly} u_i : [\overline{a_e \mapsto \boxed{\sigma_e}}, \overline{a_c \mapsto \sigma_c}]\sigma_i \tag{15}$$

Algorithmically we have to consider two cases; if $VSIG(\nu; \overline{u}, \Gamma)$ then we notice that the split of variables must be exactly the same (unification variables never get mapped to bound variables) and the result follows by repeated applications of the induction hypothesis and rule AVAR-SIG again. If on the other hand $\neg\ VSIG(\nu\ \overline{u}, \Gamma)$ we pick the maximum m, $0 \leq m \lneq n$ such that it holds that $VSIG(\nu\ \overline{u}^{1..m}, \Gamma)$ and we show that we can type this term using AVAR-SIG. By induction on $n - m$ we then show that we can type the rest of the application using AAPP.

□

Lemma 8.39 (Termination and determinacy). *The judgement* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S_1, \mathcal{A}_1)$ *defines a deterministic, terminating algorithm on inputs* S_0,$\mathcal{A}_0$,Γ,t, *and* ρ', *when* $\vdash S_0$ *and* $\mathcal{A}_0 \# ftv(\Gamma, \rho'), vars(S_0)$. *Similarly for the judgement* $(S_0, \mathcal{A}_0) \succ \Gamma \vdash^{poly} t : \sigma' \succ (S_1, \mathcal{A}_1)$.

Proof. We just give a sketch. Determinacy is an easy check. Moreover notice that the size of the term becomes smaller in each step and all auxilliary judgements terminate (by the series of soundness lemmas we have that the corresponding unifiers are well-formed and so the termination lemmas are applicable). □

$$\boxed{(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \leq \sigma_2' \succ (S_1, \mathcal{A}_1)}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash \xi \sim \sigma' \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \vdash \xi \leq \sigma' \succ (S_1, \mathcal{A}_1)}\ \textsc{asboxy} \qquad \frac{\tau_i = a, \alpha \quad S_0 \succ \vdash \tau_1 \doteq \tau_2 \succ S_1}{(S_0, \mathcal{A}_0) \succ \vdash \tau_1 \leq \tau_2 \succ (S_1, \mathcal{A}_0)}\ \textsc{amono}$$

$$\frac{\tau = a, \alpha \quad (S_0, \mathcal{A}_0) \succ \vdash \zeta \leftarrow \tau \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \vdash \tau \leq \zeta \succ (S_1, \mathcal{A}_1)}\ \textsc{abmono}$$

$$\frac{|\overline{a}| \neq 0 \quad (S_0, \mathcal{A}_0) \succ \vdash [\overline{a \mapsto \xi}]\rho_1' \ \leq\ \rho_2' \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0\overline{\xi}) \succ \vdash \forall \overline{a}\,.\,\rho_1' \ \leq\ \rho_2' \succ (S_1, \mathcal{A}_1)}\ \textsc{aspec} \qquad \frac{\sigma_1' \neq \zeta \quad |\overline{b}| \neq 0 \quad (S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \leq [\overline{b \mapsto c}]\rho_2' \succ (S_1, \mathcal{A}_1) \quad \overline{c} \# ftv(S_1(\sigma_1'), S_1(\forall \overline{b}\,.\,\rho_2'))}{(S_0, \mathcal{A}_0\overline{c}) \succ \vdash \sigma_1' \leq \forall \overline{b}\,.\,\rho_2' \succ (S_1, \mathcal{A}_1)}\ \textsc{askol}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \to \sigma_2' \leq \xi_3 \to \xi_4 \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \zeta \leftarrow (S_1\xi_3 \to S_1\xi_4) \succ (S_2, \mathcal{A}_2)}{(S_0, \mathcal{A}_0\xi_3\xi_4) \succ \vdash \sigma_1' \to \sigma_2' \leq \zeta \succ (S_3, \mathcal{A}_3)}\ \textsc{af1}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash^{\to} \alpha \doteq \sigma_1' \to \sigma_2' \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \sigma_3' \sim \sigma_1' \succ (S_2, \mathcal{A}_2) \quad (S_2, \mathcal{A}_2) \succ \vdash \sigma_2' \leq \sigma_4' \succ (S_3, \mathcal{A}_3)}{(S_0, \mathcal{A}_0) \succ \vdash \alpha \leq \sigma_3' \to \sigma_4' \succ (S_3, \mathcal{A}_3)}\ \textsc{af2l}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho_2' \doteq \sigma_3' \to \sigma_4' \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \sigma_3' \sim \sigma_1' \succ (S_2, \mathcal{A}_2) \quad (S_2, \mathcal{A}_2) \succ \vdash \sigma_2' \leq \sigma_4' \succ (S_3, \mathcal{A}_3)}{(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \to \sigma_2' \leq \rho_2' \succ (S_3, \mathcal{A}_3)}\ \textsc{af2r}$$

Figure 16: Algorithmic subsumption

$$\boxed{(S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \rho' \succ (S_1, \mathcal{A}_1)}$$

$$\frac{\nu \overset{i}{:} \sigma \in \Gamma \quad (S_0, \mathcal{A}_0) \succ \vdash \sigma \leq \rho' \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \Gamma \vdash \nu : \rho' \succ (S_1, \mathcal{A}_1)} \text{AVAR-INF}$$

$$\frac{\begin{array}{c} VSIG(\nu\ \overline{u}, \Gamma) \\ \nu \overset{s}{:} \forall \overline{a} . \overline{\sigma} \to \sigma \in \Gamma \\ \overline{a}_c = \overline{a} \cap ftv(\sigma) \quad \overline{a}_e = \overline{a} - \overline{a}_c \\ (S_0, \mathcal{A}_0) \succ \vdash [\overline{a_c \mapsto \xi_c}]\sigma \leq \rho' \succ (S_1, \mathcal{A}_1) \quad \overline{\sigma}_c = S_1 \overline{\xi_c} \\ (S_i, \mathcal{A}_i) \succ \Gamma \vdash^{poly} u_i : [\overline{a_e \mapsto \xi_e}, \overline{\overline{a_c \mapsto \sigma_c}}]\sigma_i \succ (S_{i+1}, \mathcal{A}_{i+1}) \end{array}}{(S_0, \mathcal{A}_0 \overline{\xi}_c \overline{\xi}_e) \succ \Gamma \vdash \nu\ \overline{u} : \rho' \succ (S_n, \mathcal{A}_n)} \text{AVAR-SIG}$$

$$\frac{\begin{array}{c} (S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho' \doteq \sigma_1' \to \sigma_2' \succ (S_1, \mathcal{A}_1) \\ (S_1, \mathcal{A}_1) \succ \vdash \zeta \sim \sigma_1' \succ (S_2, \mathcal{A}_2) \\ (S_2, \mathcal{A}_2) \succ \Gamma, x \overset{s}{:} S_2\zeta \vdash^{poly} t : \sigma_2' \succ (S_3, \mathcal{A}_3) \end{array}}{(S_0, \mathcal{A}_0\zeta) \succ \Gamma \vdash (\backslash x\, .\, t) : \rho' \succ (S_3, \mathcal{A}_3)} \text{AABS1}$$

$$\frac{\begin{array}{c} (S_0, \mathcal{A}_0) \succ \Gamma \vdash (\backslash x\ .\ t) : \zeta_1 \to \zeta_2 \succ (S_1, \mathcal{A}_1) \\ (S_1, \mathcal{A}_1) \succ \vdash \zeta \leftarrow (S_1\zeta_1 \to S_1\zeta_2) \succ (S_2, \mathcal{A}_2) \end{array}}{(S_0, \mathcal{A}_0\zeta_1\zeta_2) \succ \Gamma \vdash (\backslash x\ .\ t) : \zeta \succ (S_2, \mathcal{A}_2)} \text{AABS2}$$

$$\frac{\begin{array}{c} \neg\ VSIG(t\ u, \Gamma) \\ (S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \xi \to \rho' \succ (S_1, \mathcal{A}_1) \\ (S_1, \mathcal{A}_1) \succ \Gamma \vdash^{poly} u : S_1\xi \succ (S_2, \mathcal{A}_2) \end{array}}{(S_0, \mathcal{A}_0\xi) \succ \Gamma \vdash t\ u : \rho' \succ (S_2, \mathcal{A}_2)} \text{AAPP}$$

$$\frac{\begin{array}{c} (S_0, \mathcal{A}_0) \succ \Gamma \vdash u : \zeta \succ (S_1, \mathcal{A}_1) \\ \overline{\alpha} = ftv(S_1\zeta) - ftv(S_1\Gamma) \\ (S_1, \mathcal{A}_1) \succ \Gamma, x \overset{i}{:} \forall \overline{a} . [\overline{\alpha \mapsto a}]S_1\zeta \vdash t : \rho' \succ (S_2, \mathcal{A}_2) \end{array}}{(S_0, \mathcal{A}_0\overline{a}\zeta) \succ \Gamma \vdash \texttt{let}\ x\ \texttt{=}\ u\ \texttt{in}\ t : \rho' \succ (S_2, \mathcal{A}_2)} \text{ALET}$$

$$\frac{\begin{array}{c} fuv(\forall \overline{a} . \rho) = \emptyset \quad ftv(\forall \overline{a} . \rho) \subseteq dom(\Gamma) \\ (S_0, \mathcal{A}_0) \succ \Gamma \vdash u : [\overline{a \mapsto c}]\rho \succ (S_1, \mathcal{A}_1) \quad \overline{c} \# ftv(S_1\Gamma) \\ (S_1, \mathcal{A}_1) \succ \Gamma, x \overset{s}{:} \forall \overline{a} . \rho \vdash t : \rho' \succ (S_2, \mathcal{A}_2) \end{array}}{(S_0, \mathcal{A}_0\overline{c}) \succ \Gamma \vdash \texttt{let}\ x :: \forall \overline{a} . \rho\ \texttt{=}\ u\ \texttt{in}\ t : \rho' \succ (S_2, \mathcal{A}_2)} \text{ASIG-LET}$$

$$\boxed{(S_0, \mathcal{A}_0) \succ \Gamma \vdash^{poly} t : \sigma' \succ (S_1, \mathcal{A}_1)}$$

$$\frac{\begin{array}{c} (S_0, \mathcal{A}_0) \succ \Gamma \vdash t : [\overline{a \mapsto b}]\rho' \succ (S_1, \mathcal{A}_1) \\ \overline{b} \# ftv(S_1\Gamma, S_1(\forall \overline{a} . \rho')) \end{array}}{(S_0, \mathcal{A}_0\overline{b}) \succ \Gamma \vdash^{poly} t : \forall \overline{a} . \rho' \succ (S_1, \mathcal{A}_1)} \text{AGEN1} \qquad \frac{(S_0, \mathcal{A}_0) \succ \Gamma \vdash t : \xi \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \Gamma \vdash^{poly} t : \xi \succ (S_1, \mathcal{A}_1)} \text{AGEN2}$$

Figure 17: Inference/Checking Algorithm

$$\boxed{(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \sim \sigma_2' \succ (S_1, \mathcal{A}_1)}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \zeta_1 \leftarrow \alpha \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \zeta_2 \leftarrow \alpha \succ (S_2, \mathcal{A}_2)}{(S_0, \mathcal{A}_0\alpha) \succ \vdash \zeta_1 \sim \zeta_2 \succ (S_2, \mathcal{A}_2)}\ \textsc{abbeq} \qquad \frac{\tau_i = a, \alpha \quad S_0 \succ \vdash \tau_1 \doteq \tau_2 \succ S_1}{(S_0, \mathcal{A}_0) \succ \vdash \tau_1 \sim \tau_2 \succ (S_1, \mathcal{A}_0)}\ \textsc{ameq2}$$

$$\frac{\tau = a, \alpha \quad (S_0, \mathcal{A}_0) \succ \zeta \leftarrow \tau \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \vdash \zeta \sim \tau \succ (S_1, \mathcal{A}_1)}\ \textsc{ameq1l} \qquad \frac{\tau = a, \alpha \quad (S_0, \mathcal{A}_0 \succ \zeta \leftarrow \tau \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \vdash \tau \sim \zeta \succ (S_1, \mathcal{A}_1)}\ \textsc{ameq1r}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash \xi_1 \to \xi_2 \sim \sigma_3' \to \sigma_4' \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \zeta \leftarrow (S_1\xi_1 \to S_1\xi_2) \succ (S_2, \mathcal{A}_2)}{(S_0, \mathcal{A}_0\xi_1\xi_2) \succ \vdash \zeta \sim (\sigma_3' \to \sigma_4') \succ (S_2, \mathcal{A}_2)}\ \textsc{aeq1l}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \to \sigma_2' \sim \xi_3 \to \xi_4 \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \zeta \leftarrow (S_1\xi_3 \to S_1\xi_4) \succ (S_2, \mathcal{A}_2)}{(S_0, \mathcal{A}_0\xi_3\xi_4) \succ \vdash (\sigma_1' \to \sigma_2') \sim \zeta \succ (S_2, \mathcal{A}_2)}\ \textsc{aeq1r}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash^{\to} \alpha \doteq \sigma_1' \to \sigma_2' \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \sigma_1' \sim \sigma_3' \succ (S_2, \mathcal{A}_2) \quad (S_2, \mathcal{A}_2) \succ \vdash \sigma_2' \sim \sigma_4' \succ (S_3, \mathcal{A}_3)}{(S_0, \mathcal{A}_0) \succ \vdash \alpha \sim \sigma_3' \to \sigma_4' \succ (S_3, \mathcal{A}_3)}\ \textsc{aeq2l}$$

$$\frac{(S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho' \doteq \sigma_3' \to \sigma_4' \succ (S_1, \mathcal{A}_1) \quad (S_1, \mathcal{A}_1) \succ \vdash \sigma_1' \sim \sigma_3' \succ (S_2, \mathcal{A}_2) \quad (S_2, \mathcal{A}_2) \succ \vdash \sigma_2' \sim \sigma_4' \succ (S_3, \mathcal{A}_3)}{(S_0, \mathcal{A}_0) \succ \vdash \sigma_1' \to \sigma_2' \sim \rho' \succ (S_3, \mathcal{A}_3)}\ \textsc{aeq2r}$$

$$\frac{|\overline{a}| \neq 0 \quad (S_0, \mathcal{A}_0) \succ \vdash \zeta_1 \sim [\overline{a \mapsto b}]\rho' \succ (S_1, \mathcal{A}_1) \quad \overline{b} \# ftv(S_1(\forall \overline{a}.\, \rho')) \quad (S_1, \mathcal{A}_1) \succ \vdash \xi \leftarrow \forall \overline{b}.\, S_1\zeta_1 \succ (S_2, \mathcal{A}_2)}{(S_0, \mathcal{A}_0\overline{b}\zeta_1) \succ \vdash \xi \sim \forall \overline{a}.\, \rho' \succ (S_2, \mathcal{A}_2)}\ \textsc{aseq1l}$$

$$\frac{|\overline{a}| \neq 0 \quad (S_0, \mathcal{A}_0) \succ \vdash \zeta_1 \sim [\overline{a \mapsto b}]\rho' \succ (S_1, \mathcal{A}_1) \quad \overline{b} \# ftv(S_1(\forall \overline{a}.\, \rho')) \quad (S_1, \mathcal{A}_1) \succ \vdash \xi \leftarrow \forall \overline{b}.\, S_1\zeta_1 \succ (S_2, \mathcal{A}_2)}{(S_0, \mathcal{A}_0\overline{b}\zeta_1) \succ \vdash \forall \overline{a}.\, \rho' \sim \xi \succ (S_2, \mathcal{A}_2)}\ \textsc{aseq1r}$$

$$\frac{|\overline{a}| = |\overline{b}| \neq 0 \quad (S_0, \mathcal{A}_0) \succ \vdash [\overline{a \mapsto c}]\rho_1' \sim [\overline{a \mapsto c}]\rho_2' \succ (S_1, \mathcal{A}_1) \quad \overline{c} \# ftv(S_1(\forall \overline{a}.\, \rho_1'), S_1(\forall \overline{b}.\, \rho_2'))}{(S_0, \mathcal{A}_0\overline{c}) \succ \vdash \forall \overline{a}.\, \rho_1' \sim \forall \overline{b}.\, \rho_2' \succ (S_1, \mathcal{A}_1)}\ \textsc{aseq2}$$

$$\boxed{(S_0, \mathcal{A}_0) \succ \vdash^{\to} \rho' \doteq \sigma_1' \to \sigma_2' \succ (S_1, \mathcal{A}_1)} \qquad \boxed{(S_0, \mathcal{A}_0) \succ \vdash \zeta \leftarrow \sigma \succ (S_1, \mathcal{A}_1)}$$

$$\frac{S_0 \succ \vdash \alpha \doteq \alpha_1 \to \alpha_2 \succ S_1}{(S_0, \mathcal{A}_0\alpha_1\alpha_2) \succ \vdash^{\to} \alpha \doteq \alpha_1 \to \alpha_2 \succ (S_1, \mathcal{A}_0)}\ \textsc{auf1} \qquad \frac{\zeta \notin dom(S_0)}{(S_0, \mathcal{A}_0) \succ \vdash \zeta \leftarrow \sigma \succ ([\zeta \mapsto S_0\sigma] \cdot S_0, \mathcal{A}_0)}\ \textsc{fill}$$

$$\frac{}{(S_0, \mathcal{A}_0) \succ \vdash^{\to} \sigma_1' \to \sigma_2' \doteq \sigma_1' \to \sigma_2' \succ (S_0, \mathcal{A}_0)}\ \textsc{auf2} \qquad \frac{\zeta \in dom(S_0) \quad (S_0, \mathcal{A}_0) \succ \vdash S_0\zeta \sim \sigma \succ (S_1, \mathcal{A}_1)}{(S_0, \mathcal{A}_0) \succ \vdash \zeta \leftarrow \sigma \succ (S_1, \mathcal{A}_1)}\ \textsc{unif}$$

Figure 18: Boxy matching and relatives

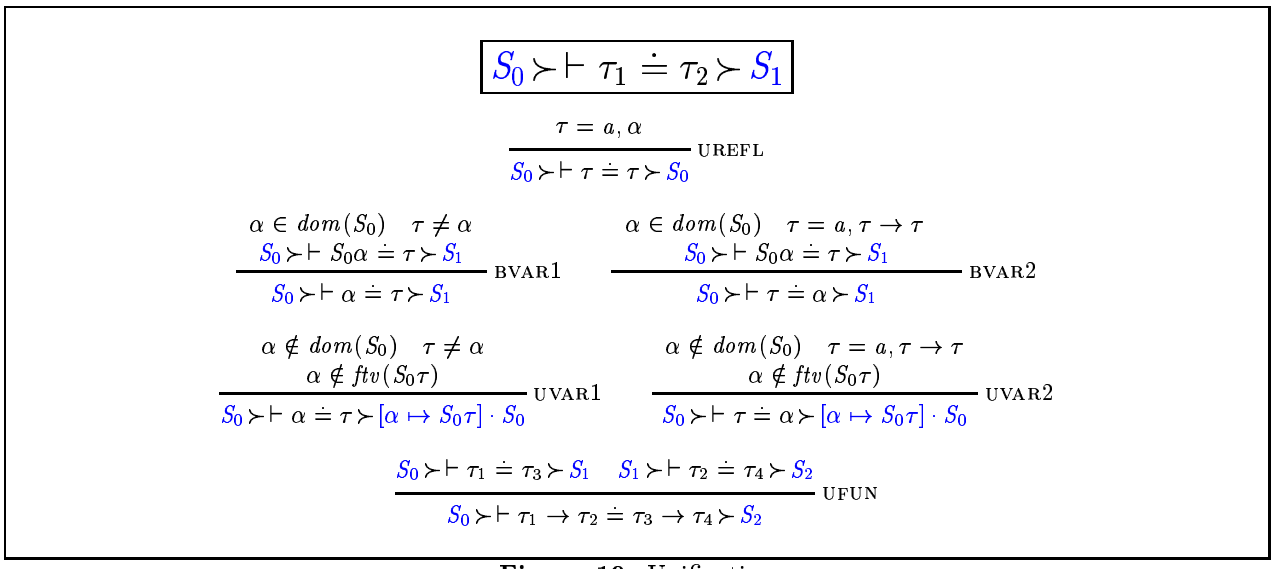

Figure 19: Unification

References

[1] Dimitrios Vytiniotis, Stephanie Weirich, and Simon Peyton Jones. Boxy type inference for higher-rank types and impredicativity. 2006.