Simple unification-based type inference for GADTs
Technical Appendix

University of Pennsylvania Technical Report MS-CIS-05-22

(in preparation)

Simon Peyton Jones — Microsoft Research

Dimitrios Vytiniotis, Stephanie Weirich — University of Pennsylvania

May 1, 2006

1 Introduction

This is a re-written version of the report accompanying the paper "Simple unification-based type inference for GADTs" [PVWW05]. We do not further explain the type system presented in the main paper but proceed with the metatheory of the languages of the main paper.

In the introductory Section 2 we elaborate on the notion of a fresh most general unifier, and describe its relation to most general unifiers. Additionally we present an algorithmic implementation of *fmgu*, which is the one that the type inference engine may use.

In Section 3 we present a core language to which GADT programs get translated during type inference. We give static and operational semantics and show that this core language is type safe.

In Section 4 we present a source language that includes non-nested patterns but patterns that include lexical type variables and we show how this language translates to the target language.

In Section 5 we show that the source language is a conservative extension of a language with patterns (non-nested) and conventional (but perhaps existential) datatypes based on the Hindley-Milner type system.

Section 6 gives a type inference algorithm in the form of inference rules and shows the connection between the type system derivations and derivations in the algorithmic presentation. Finally we give the language that supports fully nested patterns and the corresponding modifications in the algorithm in Section 7. We do not attempt to repeat the proofs for this language.

An older draft of this technical report, which describes a language that does not use *fmgu*s, but rather the *same* most general unifier procedure both in the type system and the specification, and additionally includes nested patterns, and allows binding of "wobbly" type variables can be downloaded from:

`www.cis.upenn.edu/~dimitriv/wobbly/wobbly-techreport-old.pdf`

$$\boxed{\texttt{fmgu}(\tau_1 \doteq \tau_2) = \theta}$$

$$\texttt{fmgu}(\tau_1 \doteq \tau_2) = \texttt{fmgu}^*(\{\tau_1 \doteq \tau_2\}, \mathit{ftv}(\tau_2))\,|_{\mathit{ftv}(\tau_1,\tau_2)}$$

$$\boxed{\texttt{fmgu}^*(\mathcal{E}, \mathcal{B}) = \theta}$$

$$\mathcal{E} = \cdot \mid (\tau_1 \doteq \tau_2) \cup \mathcal{E}$$

$$
\begin{array}{lll}
1. & \texttt{fmgu}^*(\emptyset, \mathcal{B}) & = \varepsilon \\
2. & \texttt{fmgu}^*(a \doteq a \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*(\mathcal{E}, \mathcal{B}) \\
3. & \texttt{fmgu}^*(a \doteq b \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*([a \mapsto b]\mathcal{E}, \mathcal{B}) \cdot [a \mapsto b] \quad a \in \mathcal{B}, b \notin \mathcal{B} \\
4. & \texttt{fmgu}^*(b \doteq a \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*([a \mapsto b]\mathcal{E}, \mathcal{B}) \cdot [a \mapsto b] \quad a \in \mathcal{B}, b \notin \mathcal{B} \\
5. & \texttt{fmgu}^*(a \doteq b \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*([a \mapsto c, b \mapsto c]\mathcal{E}, \mathcal{B}) \cdot [a \mapsto c, b \mapsto c] \\
 & & \qquad \text{where } (a, b \notin \mathcal{B} \vee a, b \in \mathcal{B}) \text{ and } c \text{ fresh} \\
6. & \texttt{fmgu}^*(a \doteq \tau \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*([a \mapsto \tau]\mathcal{E}, \mathcal{B}) \cdot [a \mapsto \tau] \quad \text{where } a \notin \mathit{ftv}(\tau) \text{ and } \tau \neq b \\
7. & \texttt{fmgu}^*(\tau \doteq a \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*([a \mapsto \tau]\mathcal{E}, \mathcal{B}) \cdot [a \mapsto \tau] \quad \text{where } a \notin \mathit{ftv}(\tau) \text{ and } \tau \neq b \\
8. & \texttt{fmgu}^*((T\,\overline{\tau}_1 \doteq T\,\overline{\tau}_2) \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*(\overline{\tau}_1 \doteq \overline{\tau}_2 \cup \mathcal{E}, \mathcal{B}) \\
9. & \texttt{fmgu}^*((\tau_1 \to \tau_2 \doteq \tau_3 \to \tau_4) \cup \mathcal{E}, \mathcal{B}) & = \texttt{fmgu}^*(\{\tau_1 \doteq \tau_2, \tau_3 \doteq \tau_4\} \cup \mathcal{E}, \mathcal{B})
\end{array}
$$

Figure 1: An implementation of *fmgu*

2 Fresh most general unifiers

Here we describe the notion of a fresh most general unifier. We give some preliminary definitions, which will be useful throughout the report first.

Definition 2.1 (Permutations). *A permutation (or renaming), denoted with r, is a mapping from a finite set of variables $dom(r)$ onto itself.*

It can be proved that a substitution is invertible iff it is a permutation. If r is a permutation, we denote its inverse with r^{-1}, such that $r \cdot r^{-1} = \varepsilon$. We use ε to denote the identity-everywhere substitution.

Next, recall the definition of *fmgu*. A fresh most general unifier is an idempotent substitution that can yield a most general unifier but has no directionality preferrence between variables in τ_1 and τ_2. For example, if the mapping $[a \in \mathit{ftv}(\tau_1) \mapsto b \in \mathit{ftv}(\tau_1)]$ exists in a most general unifier of τ_1 and τ_2, then $[a \mapsto c]$ and $[b \mapsto c]$—where c is fresh—in a fresh most general unifier of τ_1 and τ_2.

Definition 2.2 (*fmgu*). *An idempotent substitution θ is a fresh most general unifier of τ_1 and τ_2, written $\theta \in \mathit{fmgu}(\tau_1 \doteq \tau_2)$, iff*

(i) *θ is a unifier of τ_1 and τ_2, that is, $\theta(\tau_1) = \theta(\tau_2)$.*

(ii) *For any idempotent unifier ϕ of τ_1 and τ_2 there exists a substitution ψ such that $\phi(a) = \psi(\theta(a))$ for all $a \in \mathit{ftv}(\tau_1, \tau_2)$.*

(iii) *For every $a, b \in \mathit{ftv}(\tau_1)$, with $a \neq b$, it is $\theta(a) \neq b$. For every $a, b \in \mathit{ftv}(\tau_2)$, with $a \neq b$, it is $\theta(a) \neq b$.*

(iv) *$dom(\theta) \subseteq \mathit{ftv}(\tau_1, \tau_2)$ and all type variables in $range(\theta)$ are either in $\mathit{ftv}(\tau_1, \tau_2)$ or are fresh (disjoint from variables introduced by the typing judgment that uses θ).*

We formalize the freshness requirement at the use sites of refinements in the type system. Recall the implementation of *fmgu*s from the main paper in Figure 1. Notice that we are being implicit about freshness in this figure. This is good for presentation purposes but not appropriate for formalization and proofs.

$$\boxed{\mathtt{fmgu}(\tau_1 \doteq \tau_2\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1}$$

$$\begin{array}{lcl}
\mathtt{fmgu}(\tau_1 \doteq \tau_2\,;\mathcal{A}_0) & = & \theta|_{ftv(\tau_1,\tau_2)}\,;\mathcal{A}_1 \\
 & & \text{where } \mathtt{fmgu}^*(\{\tau_1 \doteq \tau_2\}, ftv(\tau_2)\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1
\end{array}$$

$$\boxed{\mathtt{fmgu}^*(\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1}$$

$$\mathcal{E} = \cdot \mid (\tau_1 \doteq \tau_2) \cup \mathcal{E}$$

$$\begin{array}{llcl}
1. & \mathtt{fmgu}^*(\emptyset,\mathcal{B}\,;\mathcal{A}_0) & = & \varepsilon \\
2. & \mathtt{fmgu}^*(a \doteq a \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \mathtt{fmgu}^*(\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) \\
3. & \mathtt{fmgu}^*(a \doteq b \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \theta \cdot [a \mapsto b]\,;\mathcal{A}_1 \\
 & & & \text{where } \mathtt{fmgu}^*([a \mapsto b]\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1 \quad a \in \mathcal{B}, b \notin \mathcal{B} \\
4. & \mathtt{fmgu}^*(b \doteq a \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \theta \cdot [a \mapsto b]\,;\mathcal{A}_1 \\
 & & & \text{where } \mathtt{fmgu}^*([a \mapsto b]\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1 \quad a \in \mathcal{B}, b \notin \mathcal{B} \\
5. & \mathtt{fmgu}^*(a \doteq b \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \theta \cdot [a \mapsto c, b \mapsto c]\,;\mathcal{A}_1 \\
 & & & \text{where } \mathcal{A}_0 = \mathcal{A}_0' \cup \{c\} \quad \mathtt{fmgu}^*([a \mapsto c, b \mapsto c]\mathcal{E},\mathcal{B}\,;\mathcal{A}_0') = \theta\,;\mathcal{A}_1 \\
 & & & \text{and } (a, b \notin \mathcal{B} \lor a, b \in \mathcal{B}) \\
6. & \mathtt{fmgu}^*(a \doteq \tau \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \theta \cdot [a \mapsto \tau]\,;\mathcal{A}_1 \\
 & & & \text{where } \mathtt{fmgu}^*([a \mapsto \tau]\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1 \\
 & & & \text{and } a \notin ftv(\tau) \text{ and } \tau \neq b \\
7. & \mathtt{fmgu}^*(\tau \doteq a \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \theta \cdot [a \mapsto \tau]\,;\mathcal{A}_1 \\
 & & & \text{where } \mathtt{fmgu}^*([a \mapsto \tau]\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1 \\
 & & & \text{and } a \notin ftv(\tau) \text{ and } \tau \neq b \\
8. & \mathtt{fmgu}^*((T\,\overline{\tau}_1 \doteq T\,\overline{\tau}_2) \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \mathtt{fmgu}^*(\overline{\tau}_1 \doteq \overline{\tau}_2 \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) \\
9. & \mathtt{fmgu}^*((\tau_1 \to \tau_2 \doteq \tau_3 \to \tau_4) \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0) & = & \mathtt{fmgu}^*(\{\tau_1 \doteq \tau_2, \tau_3 \doteq \tau_4\} \cup \mathcal{E},\mathcal{B}\,;\mathcal{A}_0)
\end{array}$$

Figure 2: An implementation of *fmgu* with explicit fresh symbol tapes

To be explicit and formally correct about the freshness condition we consider *tapes* of fresh symbols, denoted with $\mathcal{A}$, passed in to the algorithm, which becomes:

$$\mathtt{fmgu}(\mathcal{E},\mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1$$

The procedure takes an extra argument $\mathcal{A}_0$, the initial fresh symbol tape, off the top of which we remove fresh symbols to use in the algorithm, and $\mathcal{A}_1$ returned with θ is the remaining tape. Formally the modified algorithmic presentation is given in Figure 2. We work with this figure for the rest of this paper.

To provide a better understanding of the relationship between *fmgu*s and *mgu*s, as well as the properties of the `fmgu` procedure, we present some technical results on fresh most general unifiers.

Lemma 2.3 (Connection of *fmgu* and *mgu*). *If* $fmgu(\tau_1 \doteq \tau_2) = \theta$ *then there exists an injective mapping* m*, with* $dom(m) \subseteq range(\theta)$*,* $dom(m) \# ftv(\tau_1,\tau_2)$ *and* $range(m) \subseteq dom(\theta)$*, such that* $m \cdot \theta\,|_{ftv(\tau_1,\tau_2)} \in mgu(\tau_1 \doteq \tau_2)$*.*

Proof. Consider the shape of θ. Divide the domain of θ in a finite number of sets as follows. Let the *type-mapped* variables be the set

$$T = \{a \mid a \in dom(\theta) \land \theta(a) = \tau \land \tau \neq b\}$$

Let the *var-mapped* variables be the set

$$V = \{a \mid a \in dom(\theta) \land \theta(a) = b \land b \in ftv(\tau_1,\tau_2)\}$$

Finally, for every distinct f_i fresh variable for which there exists an $a \in dom(\theta)$ with $\theta(a) = f_i$, create the set

$$F_{f_i} = \{a \mid a \in dom(\theta) \land \theta(a) = f_1\}$$

This partition of $dom(\theta)$ is complete, that is,

$$dom(\theta) = T \cup V \cup (\bigcup F_{f_i})$$

Moreover, consider an (arbitrary) ordering on variables (e.g. lexicographic) and for every set F_{f_i}, pick a representative element, let us call it a_{f_i}. Consider the mapping $m = [\overline{f_i \mapsto a_{f_i}}]$, that is, map every f_i (where there exists an $a \in dom(\theta)$ with $\theta(a) = f_i$) to the representative element of F_{f_i}.

Since we choose the distinct f_i's, this is a valid substitution. Moreover, $dom(m)\#ftv(\tau_1, \tau_2)$, and additionally $range(m) \subseteq dom(\theta)$. Now, it must also be that this map is injective, since all a_{f_i}'s belong in the domain of θ, and therefore are distinct.

The next claim is that $(m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)}$ is a most general unifier. First, it is clear that it is a unifier, since by the definition of *fmgu* θ is a unifier of τ_1 and τ_2. It is also easy to confirm that by construction is idempotent.

To establish that it is indeed a most general unifier we have to show that for every other idempotent substitution ϕ, such that $\phi(\tau_1) = \phi(\tau_2)$ there exists a ψ, where $\phi = \psi \cdot ((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)})$. From the definition of *fmgu* we know that there exists a ψ' such that $\phi(a) = \psi'(\theta(a))$ when $a \in ftv(\tau_1, \tau_2)$. Consider $m^T = [\overline{a_{f_i} \mapsto f_i}]$. Now, by a similar argument as before, m^T is a well-defined substitution (and in fact injective). Let us name the following substitution:

$$\psi = \phi \cdot ((\psi' \cdot m^T) \mid_{ftv(\tau_1,\tau_2)}))$$

We will be done if we can show that

$$\phi = \psi \cdot ((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)})$$

Take an arbitrary variable a. If $a \notin ftv(\tau_1, \tau_2)$ then it must be that $\psi \cdot ((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)}) = \phi(a)$ and we are done. If $a \in ftv(\tau_1, \tau_2)$, then we have two cases to consider. If $a \in dom(\theta)$ then let $\tau = \theta(a)$. It must be that $ftv(\tau)\#\overline{a_{f_i}}$ since the latter are in the domain of θ, and θ is idempotent. Therefore

$$\begin{aligned} & ((\psi' \cdot m^T) \mid_{ftv(\tau_1,\tau_2)})) \cdot ((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)})(a) \\ = \; & \psi'(\theta(a)) \\ = \; & \phi(a) \end{aligned}$$

It follows that

$$\begin{aligned} \psi \cdot ((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)}) &= \phi(\phi(a)) \\ &= \phi(a) \end{aligned}$$

since ϕ is idempotent. On the other hand, if $a \notin dom(\theta)$ then it is $\phi(a) = \psi'(a)$ using the definition of *fmgu*. Then $((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)})(a) = a$ and $((\psi' \cdot m^T) \mid_{ftv(\tau_1,\tau_2)}))(a) = \psi'(a)$ because $a \notin \overline{a_{f_i}}$ (otherwise it would have to be that $a \in dom(\theta)$. Hence we get that

$$\begin{aligned} \psi \cdot ((m \cdot \theta) \mid_{ftv(\tau_1,\tau_2)})(a) &= \phi(\psi'(a)) \\ &= \phi(\phi(a)) = \phi(a) \end{aligned}$$

as required. $\square$

Corollary 2.4. *If $\theta \in fmgu(\tau_1 \doteq \tau_2)$ then there exists an (idempotent) mgu of τ_1 and τ_2, μ and an injective mapping m such that $dom(m)$ consists of fresh variables, and $range(m) \subseteq dom(\theta)$ and $m \cdot \theta = \mu \cdot m$.*

Proof. Just take the construction of μ and m indicated by the previous lemma—the rest is an easy check. $\square$

Next, let us prove that the fmgu procedure actually produces *fmgu*s.

Lemma 2.5. *If* $\mathtt{fmgu}^*(\mathcal{E}, \mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1$ *and* $\mathcal{A}_0\#\mathcal{E},\mathcal{B}$ *then* θ *is an idempotent unifier of* $\mathcal{E}$.

Proof. By induction on the number of steps performed. Case 1 is trivial. Case 2 follows by induction hypothesis. For Case 3, by induction hypothesis we get that $\theta'\,;\mathcal{A}_1 = \mathtt{fmgu}^*([a \mapsto b]\mathcal{E}, \mathcal{B}\,;\mathcal{A}_0)$ with θ' a unifier of $[a \mapsto b]\mathcal{E}$, therefore $\theta' \cdot [a \mapsto b]$ is a unifier of $a \doteq b \cup \mathcal{E}$. For idempotency, by induction hypothesis we know that $dom(\theta') \cap range(\theta') = \emptyset$. We need to show that $a\#range(\theta'), b$. It is clear that $a \neq b$ and a cannot be in the range of θ since a does not appear in the variables of $[a \mapsto b]\mathcal{E}$ (needs an easy separate induction). Additionally, we need to show that $\theta'(b)\#dom(\theta'), a$. It cannot be $a \in \theta'(b)$, as a does not appear in the range of θ'. Moreover $\theta'(\theta'(b)) = \theta'(b)$ by idempotency of θ'. Cases 4,5,6 and 7 are similar. Cases 8 and 9 follow directly by induction hypothesis. $\square$

Corollary 2.6. *If* $\mathtt{fmgu}(\tau_1 \doteq \tau_2\,;\mathcal{A}_0) = \theta$ *with* $\mathcal{A}_0\#ftv(\tau_1,\tau_2)$ *then* $\theta(\tau_1) = \theta(\tau_2)$ *and* θ *is idempotent. Moreovoer* $dom(\theta) \subseteq ftv(\tau_1,\tau_2)$.

Proof. Follows by Lemma 2.5 and the definition of fmgu. $\square$

In what follows we often use $\mathcal{F}$ to describe variables coming from the fresh symbol tape.

Lemma 2.7. *If* $\mathtt{fmgu}^*(\mathcal{E}, \mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1$ *and* $\mathcal{A}_0\#\mathcal{E},\mathcal{B}$*, and* $\mathcal{F} = \mathcal{A}_0 - \mathcal{A}_1$*, then the variable-to-variable bindings of* θ *are of one of the following forms:*

1. $[f \in \mathcal{F} \mapsto f \in \mathcal{F}]$.
2. $[a \in \mathcal{B} \mapsto b \notin \mathcal{B}]$.
3. $[a \notin \mathcal{B} \cup \mathcal{F} \mapsto f \in \mathcal{F}]$.

Proof. By induction on the steps performed. Case 1 is trivial and Case 2 follows by induction hypothesis. For Case 3 we have to show that the new binding satisfies the required properties. To show this it must be the case that b doesn't recursively get bound to $c \in \mathcal{B}$. But by induction hypothesis b will be mapped only to some $f \in \mathcal{F}$, if mapped at all so we are done. Case 4 is similar. For Case 5 we have to take cases. If $a, b \in \mathcal{B}$ then we must show that recursively c does not get mapped to $d \in \mathcal{B}$. But $\mathcal{F} = \mathcal{A}'_0 c - \mathcal{A}_1$ and $\mathcal{F}' = \mathcal{A}'_0 - \mathcal{A}_1$ therefore $c \notin \mathcal{B} \cup \mathcal{F}'$. Then by induction hypothesis, if c is mapped at all, it is mapped to a variable in $\mathcal{F}'$, so we are done. On the other hand, if $a, b \notin \mathcal{B}$ it is also that $a, b \notin \mathcal{A}_0 - \mathcal{A}_1$ and we just have to show that if c is mapped to a variable it is mapped to a fresh one, which holds with a similar argument as before. Cases 6,7,8 and 9 are straightforward. $\square$

Corollary 2.8. *If* $\mathtt{fmgu}(\tau_1 \doteq \tau_2;\mathcal{A}_0) = \theta;\mathcal{A}_1$ *with* $\mathcal{A}_1\#ftv(\tau_1,\tau_2)$*, and* $ftv(\tau_1)\#ftv(\tau_2)$ *then no two variables from* τ_1 *are equated directly through* θ*, and no two variables from* τ_2 *are equated directly through* θ.

Proof. By Lemma 2.7, and taking account of the definition of fmgu the possible variable-to-variable bindings of θ are

1. $[a \in ftv(\tau_2) \mapsto b \notin ftv(\tau_2)]$.
2. $[a \notin ftv(\tau_2) \cup \mathcal{F} \mapsto f \in \mathcal{F}]$.

where $\mathcal{F} = \mathcal{A}_0 - \mathcal{A}_1$. Consequently, if $a \in ftv(\tau_1)$ then the only possible variable-to-variable mappings of a are those that send a to a variable $f \in \mathcal{F}$. $\square$

We assume that the $\texttt{fmgu}^*$ procedure terminates when no step is any longer applicable. Since the cases it has cover all possible forms of $\mathcal{E}$, if a step is no longer applicable, if $\mathcal{E} = \emptyset$ then we say that the algorithm succeeds, otherwise we say that the algorithm terminates with a fail message. We can consequently show that the $\texttt{fmgu}^*$ procedure always terminates.

Lemma 2.9. *The procedure* $\texttt{fmgu}^*$ *terminates.*

Proof. Application of Case 1 terminates immediately. Application of Case 2 reduces the total number of occurrences of variables. Application of Case 8 and 9 reduces the total number of occurrences of constructors in equations. Cases 6 and 7 can only be applied a finite number of times before another case must be applied. Hence, after a finite number of Cases 1,2,6,7,8,9, either the algorithm will terminate, or Cases 3,4, or 5 will be applicable. But these steps reduce the number of variables in the equation set, thus can be applied only a finite number of times. Therefore the algorithm terminates for every input. ☐

Corollary 2.10 (Termination of fmgu). $\texttt{fmgu}$ *terminates.*

Finally, to establish that our procedure indeed returns *fmgu*s we need to show that any unifier can be the result of composing some substitution with the result of the algorithm—restricted to the free type variables of the equation set we are unifying.

In order to state and prove this precisely we introduce the following notation.

Definition 2.11 (Except-equality of substitutions). *We say that two substitutions* θ_1 *and* θ_2 *agree everywhere* exept perhaps *on the set* X, *and we write* $\theta_1 = \theta_2\backslash_X$ *iff for every* $a \notin X$ *it is* $\theta_1(a) = \theta_2(a)$.

Definition 2.12 (Unifiers of equation sets). *We write* $\vdash \phi(\mathcal{E})$ *iff for every* $\{\tau_1 \doteq \tau_2\} \in \mathcal{E}$ *it is* $\phi(\tau_1) = \phi(\tau_2)$.

Lemma 2.13 (Completeness of fmgu*). *If* $\vdash \phi(\mathcal{E})$ *and* $\mathcal{A}_0\#\mathcal{E}, \phi, \mathcal{B}$ *then* $\texttt{fmgu}^*(\mathcal{E}, \mathcal{B}\,; \mathcal{A}_0)$ *terminates and returns a* θ *and a* $\mathcal{A}_1$ *such that* $\phi = \psi \cdot \theta\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.

Proof. Since $\texttt{fmgu}^*$ always terminates, we perform induction on the number of steps of the algorithm. We proceed by case analysis on the form of the equation set $\mathcal{E}$.

- If $\mathcal{E} = \emptyset$ then $\theta = \varepsilon$ and taking $\psi = \theta$ finishes the case.
- Assume $\mathcal{E} = (a \doteq a \cup \mathcal{E}')$. Then it is also $\vdash \phi(\mathcal{E}')$ and $\theta\,; \mathcal{A}_1 = \texttt{fmgu}^*(\mathcal{E}', \mathcal{B}\,; \mathcal{A}_0)$ since Case 2 is applicable. By induction hypothesis then there exists a ψ such that $\phi = \psi \cdot \theta\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.
- Assume $\mathcal{E} = (a \doteq b \cup \mathcal{E}')$. There are several subcases:
  - $a \in \mathcal{B}, b \notin \mathcal{B}$. In this case we have that $\vdash \phi(a \doteq b \cup \mathcal{E}')$, consequently
    $$\vdash \phi([a \mapsto b]\mathcal{E}' \tag{1}$$
    Moreover Case 3 of the algorithm applies and a recursive call to $\texttt{fmgu}^*([a \mapsto b]\mathcal{E}', \mathcal{B}; \mathcal{A}_0)$ is made. Using (1) and induction hypothesis we get $\texttt{fmgu}^*([a \mapsto b]\mathcal{E}', \mathcal{B}\,; \mathcal{A}_0) = \theta\,; \mathcal{A}_1$ and there exists a ψ with
    $$\phi = \psi \cdot \theta\backslash_{\mathcal{A}_0 - \mathcal{A}_1} \tag{2}$$
    To finish the case we need to show that $\phi = \psi \cdot (\theta \cdot [a \mapsto b])\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$. Assume a $c \notin \mathcal{A}_0 - \mathcal{A}_1$; if $c \neq a$ then we are done by (2). If on the other hand $c = a$ then we need to show that $\phi(a) = \psi(\theta(b))$. But by assumptions $\phi(a) = \phi(b)$ and by (2) $\phi(b) = \psi(\theta(b))$.
  - $a \notin \mathcal{B}, b \in \mathcal{B}$. Similar to the previous case (but Case 4 of the algorithm applies).
  - $a, b \in \mathcal{B} \vee a, b \notin \mathcal{B}$. In this case assume that $\mathcal{A}_0 = \mathcal{A}_0' \cup \{c\}$. Moreover it is $\vdash \phi(a \doteq b \cup \mathcal{E}')$ therefore it must also be that $\vdash \phi([a \mapsto c, b \mapsto c]\mathcal{E}')$. Moreover Case 5 of the algorithm applies

and a recursive call to $\texttt{fmgu}^*([a \mapsto c, b \mapsto c]\mathcal{E}', \mathcal{B}\,; \mathcal{A}_0')$ is made. Hence, by induction hypothesis we get $\texttt{fmgu}^*([a \mapsto c, b \mapsto c]\mathcal{E}', \mathcal{B}\,; \mathcal{A}_0') = \theta\,; \mathcal{A}_1$ and there exists a ψ with

$$\phi = \psi \cdot \theta \backslash_{\mathcal{A}_0' - \mathcal{A}_1} \tag{3}$$

To finish the case we need to show that

$$\phi = ([c \mapsto \phi(a)] \cdot \psi) \cdot (\theta \cdot [a \mapsto c, b \mapsto c]) \backslash_{\mathcal{A}_0' c - \mathcal{A}_1} \tag{4}$$

Take a $d \notin \mathcal{A}_0' c - \mathcal{A}_1$. Then also $d \notin \mathcal{A}_0' - \mathcal{A}_1$. If $d \neq a, b$ then we have to show that $\phi(d) = [c \mapsto \phi(a)]\phi(d)$ using equation (3). But $c \notin \mathit{vars}(\phi)$ by assumptions so this equality holds. On the other hand if $d = a$ (similarly if $d = b$) then we need to show that $\phi(a) = [c \mapsto \phi(a)](\psi(\theta(c)))$. But we have that $\psi(\theta(c)) = \phi(c) = c$, therefore $[c \mapsto \phi(a)](\psi(\theta(c))) = \phi(a)$ as required.

- Assume $\mathcal{E} = (a \doteq \tau \cup \mathcal{E}')$. Similar to the first subcase of the previous case with Case 6 of the algorithm being applicable.
- Assume $\mathcal{E} = (a \doteq \tau \cup \mathcal{E}')$. Similar to the previous case, with Case 7 of the algorithm being applicable.
- Assume $\mathcal{E} = (T\,\overline{\tau}_1 \doteq T\,\overline{\tau}_2 \cup \mathcal{E}')$. Follows by induction hypothesis for Case 8 of the algorithm.
- Assume $\mathcal{E} = (\tau_1 \to \tau_2 \doteq \tau_3 \to \tau_4 \cup \mathcal{E}')$. Follows by induction hypothesis for Case 9 of the algorithm.

□

Corollary 2.14. *If there exists a unifier ϕ of τ_1 and τ_2, and $\mathcal{A}_0 \# \phi, \mathit{ftv}(\tau_1, \tau_2)$ then $\texttt{fmgu}(\tau_1 \doteq \tau_2\,; \mathcal{A}_0)$ terminates and returns a θ such that there exists a ψ with $\phi(a) = \psi \cdot \theta(a)$ for $a \in \mathit{ftv}(\tau_1, \tau_2)$.*

It follows that, under the assumption that we can generate as fresh symbols as we want, the procedure `fmgu` indeed calculates fresh most general unifiers.

Next we want to describe a systematic way to turn an arbitrary *fmgu* to one that—modulo choice of fresh variables—is returned by the algorithm. The first observation is that the algorithmic implementation has two extra properties: bias and minimality. Bias gives preferrence to a specific set of variables in the domain of the unifier, and minimality assures that no "unnecessary" freshening happens.

Definition 2.15 (Bias and Minimality). *Assume that $\theta \in \mathit{fmgu}(\tau_1 \doteq \tau_2)$. This θ is* biased and minimal *if:*

Bias *All the variable-to-variable mappings in θ are in one of the following forms:*

1. $[a \in \mathit{ftv}(\tau_2) \mapsto b \notin \mathit{ftv}(\tau_2)]$.
2. $[a \in \mathit{ftv}(\tau_1) \mapsto f \in \mathcal{F}]$.

Minimality *If $\theta(a) = f$ then there exists another $b \neq a$, such that $\theta(b) = f$ and $a, b \in \mathit{ftv}(\tau_2)$ or $a, b \notin \mathit{ftv}(\tau_2)$.*

Here is the roadmap: We fist show that two biased and minimal *fmgu*s of the same types have the same domain. We subsequently use this fact to show that they actually differ only by a renaming of fresh variables. Then we give a systematic procedure that transforms an arbitrary *fmgu* to a biased and minimal one. Finally we show that the algorithm indeed computes a biased and minimal *fmgu*. As a consequence we will present the precise relation between an arbitrary *fmgu* and `fmgu`.

Lemma 2.16. *If $\theta_1, \theta_2 \in \mathit{fmgu}(\tau_1 \doteq \tau_2)$ and are biased and minimal then $\mathit{dom}(\theta_1) = \mathit{dom}(\theta_2)$.*

Proof. Assume by contradiction that $a \in ftv(\tau_1, \tau_2)$ and $a \in dom(\theta_1), a \notin dom(\theta_2)$. Moreover, since an *fmgu* is a unifier we know that

$$\begin{array}{rcl} \theta_2 & = & \psi_1 \cdot \theta_1|_{ftv(\tau_1,\tau_2)} \\ \theta_1 & = & \psi_2 \cdot \theta_2|_{ftv(\tau_1,\tau_2)} \end{array}$$

for some substitutions ψ_1 and ψ_2, for which without loss of generality we can assume that $dom(\psi_i)\#dom(\theta_i)$. Then we have that $\theta_1(a) = \tau$ with $\tau \neq a$, therefore $\psi_1\theta_1(a) = \psi_1(\tau) = \theta_2(a) = a$, therefore

$$\psi_1(\tau) = a \tag{1}$$

Similarly, $\theta_2(a) = a$, hence $\psi_2\theta_2(a) = \psi_2(a)$ and $\theta_1(a) = \psi_2(a)$. Consequently

$$\psi_2(a) = \tau \tag{2}$$

Equation (1) implies that $\tau = b \in ftv(\tau_1, \tau_2)$, or $\tau = f \in \mathcal{F}$. In the first case it must be that $[b \mapsto a] \in \psi_1$, hence $\theta_2(b) = a$ and $\theta_1(a) = b$. But this is impossible since the biasing condition must be violated for either θ_1 or θ_2 or both. On the other hand, assume that $\tau = f \in \mathcal{F}$. In this case $[a \mapsto f] \in \theta_1$, $\psi_2(a) = f$, $\psi_1(f) = a$. By the minimality condition it must also be that there exists a $b \in dom(\theta_1)$ with $b \neq a$, such that $[b \mapsto f] \in \theta_1$ and moreover $a, b \in ftv(\tau_2)$ or $a, b \in ftv(\tau_1)$. We claim that $[b \mapsto a] \in \theta_2$. The reason is because: $\theta_2(b) = \psi_1\theta_1(b) = \psi_1(f) = a$. But this also violates the biasing restriction for θ_2, hence we arrive at a contradiction.

We conclude that $dom(\theta_1) = dom(\theta_2)$. $\square$

Lemma 2.17. *If $\theta_1, \theta_2 \in fmgu(\tau_1 \doteq \tau_2)$ and are biased and minimal then they differ only by substitutions that are variable-to-variable mappings.*

Proof. Recall the construction of *mgu* from an *fmgu* of Lemma 2.3. Then it is: $(m_1 \cdot \theta_1)|_{\tau_1,\tau_2} = r \cdot (m_2 \cdot \theta_2)|_{\tau_1,\tau_2}$ for some injective mappings m_1 and m_2 and an arbitrary renaming r. Then we know that, since an *fmgu* is a unifier, it is:

$$\begin{array}{rcl} \theta_2 & = & \psi_1 \cdot \theta_1|_{ftv(\tau_1,\tau_2)} \\ \theta_1 & = & \psi_2 \cdot \theta_2|_{ftv(\tau_1,\tau_2)} \end{array}$$

This means that

$$(m_1 \cdot \theta_1)|_{\tau_1,\tau_2} = r \cdot (m_2 \cdot (\psi_1 \cdot \theta_1)|_{\tau_1,\tau_2})|_{\tau_1,\tau_2} \tag{1}$$

which implies that all the types in the range of ψ_1 must indeed be variables.

As a remark notice that in fact by Lemma 2.16 $dom(\theta_1) = dom(\theta_2)$ and since $dom(m_i)$ are among fresh variables, it must be that $r = \varepsilon$. Hence:

$$(m_1 \cdot \theta_1)|_{\tau_1,\tau_2} = (m_2 \cdot (\psi_1 \cdot \theta_1)|_{\tau_1,\tau_2})|_{\tau_1,\tau_2} \tag{2}$$

$\square$

Lemma 2.18. *If $\theta_1, \theta_2 \in fmgu(\tau_1 \doteq \tau_2)$ and are biased and minimal then they differ only by renamings of fresh variables, i.e. $\theta_1 = \psi_2 \cdot \theta_2|_{ftv(\tau_1,\tau_2)}$ and $\theta_2 = \psi_1 \cdot \theta_1|_{ftv(\tau_1,\tau_2)}$ with $\psi_1 \cdot \psi_2 = \varepsilon|_{ftv(\theta_2(\tau_1,\tau_2))}$ and $\psi_2 \cdot \psi_1 = \varepsilon|_{ftv(\theta_1(\tau_1,\tau_2))}$.*

Proof. We know that, since an *fmgu* is a unifier, it is:

$$\begin{array}{rcl} \theta_2 & = & \psi_1 \cdot \theta_1|_{ftv(\tau_1,\tau_2)} \\ \theta_1 & = & \psi_2 \cdot \theta_2|_{ftv(\tau_1,\tau_2)} \end{array}$$

for some ψ_1 and ψ_2. Assume also without loss of generality that $dom(\psi_i)\#dom(\theta_i)$. Moreover without loss of generality, because of Lemma 2.16, assume that $dom(\psi_1) \subseteq range(\theta_1)$ and $dom(\psi_2) \subseteq range(\theta_2)$. Assume that $a \in ftv(\tau_1, \tau_2)$ then $\theta_1(a) = \psi_2\psi_1\theta_1(a)$ and similarly we can derive $\theta_2(a) = \psi_1\psi_2\theta_2(a)$ therefore we can derive $\psi_1 \cdot \psi_2 = \varepsilon|_{ftv(\theta_2(\tau_1,\tau_2))}$ and $\psi_2 \cdot \psi_1 = \varepsilon|_{ftv(\theta_1(\tau_1,\tau_2))}$.

Additionally ψ_1 and ψ_2 must be variable-to-variable mappings, as shown by Lemma 2.17.

Moreover it must be that the form of the bindings of ψ_i is $[f \in \mathcal{F} \mapsto a]$ where a is an arbitrary variable, because if the domain of ψ_i contains non-fresh variables, then $dom(\theta_1) \neq dom(\theta_2)$ which contradicts Lemma 2.16. We will additionally show that all the variables in the range of ψ_i are also fresh.

Assume by contradiction that $[f \mapsto a \in ftv(\tau_1, \tau_2)] \in \psi_1$. Then it must be that $f \in range(\theta_1)$, i.e. there is a $b \in ftv(\tau_1, \tau_2)$ such that $f \in \theta_1(b)$, let us refer to this binding as $[b \mapsto \tau]$. Then it is $[b \mapsto \tau] \in \theta_1$. But we have that: $\theta_2(b) = \psi_1\theta_1(b)$ therefore $\theta_2(b) = \psi_1(\tau)$. Then $\theta_1(b) = \psi_2\theta_2(b)$, therefore $\tau = \psi_2\psi_1(\tau)$, hence there must be a binding $[a \mapsto f] \in \psi_2$ which means that $dom(\theta_1) \neq dom(\theta_2)$, a contradiction.

Hence, all the variables of ψ_i are fresh. ☐

Lemma 2.19. *If $\theta \in fmgu(\tau_1 \doteq \tau_2)$ then there exists a mapping of variables m such that $m \cdot \theta|_{ftv(\tau_1,\tau_2)}$ is a minimal and biased fmgu of τ_1 and τ_2. Moreover $vars(m) \subseteq vars(\theta) \cup \mathcal{F}$, and $dom(m) \subseteq ftv(\tau_2) \cup \mathcal{F}$.*

Proof. Assume that we have an *fmgu* θ. We show how to create a minimal and biased *fmgu* in a sequence of steps.

- In this step take all pairs of variables $a, b \in dom(\theta)$ such that $a \in ftv(\tau_1)$, $b \in ftv(\tau_2)$ and $\theta(a) = \theta(b) = f \in \mathcal{F}$ and there is no other $c \in dom(\theta)$ such that $\theta(c) = f$. Create the mapping $m = [f \mapsto a]$ and compose it with θ. Repeat until no such bindings exist in θ.
- Next, consider any maximal set $\overline{a} \subseteq dom(\theta)$ such that $\overline{a} \in ftv(\tau_1)$ such that $[\overline{a} \mapsto b \in ftv(\tau_2)] \in \theta$, i.e. all $\overline{a}$ map to a variable in τ_2. If $\overline{a} = \{a\}$ is a singleton set then create the mapping $m = [b \mapsto a]$ and compose it with the rest of the substitution. If $\overline{a}$ is not a singleton, create a fresh symbol f and the mapping $m = [b \mapsto f]$ and compose it with θ. Repeat this step as many times as is required to eliminate all such bindings in θ.
- Finally consider all bindings of the form $[a \in ftv(\tau_1) \mapsto f \in \mathcal{F}]$ or $[b \in ftv(\tau_2) \mapsto f \in \mathcal{F}]$ and there is no other $c \in dom(\theta)$ such that $\theta(c) = f$. For these singleton bindings, create the mapping $m = [f \mapsto a]$ (or $m = [f \mapsto b]$) and compose it with θ. Repeat until no such singleton bindings exist.

It is easy to confirm that this procedure terminates and gives back an *fmgu* which satisfies both the minimality and bias conditions. Moreover, by construction, the mapping contains in its domain either completely fresh variables, or variables from $ftv(\tau_2)$. ☐

Lemma 2.20. *If $\mathcal{A}_0 \# \mathcal{E}, \mathcal{B}$ and $\texttt{fmgu}^*(\mathcal{E}, \mathcal{B}\,;\mathcal{A}_0) = \theta\,;\mathcal{A}_1$ then for every $f \in \mathcal{A}_0 - \mathcal{A}_1$ with $f \in range(\theta)$, there exist $a, b \in \mathcal{B}$ or $a, b \in ftv(\mathcal{E}) - \mathcal{B}$ such that $\theta(a) = \theta(b) = f$.*

Proof. By induction on the number of recursive calls of the algorithm. We proceed with case analysis. Case 1 is trivial since then $\theta = \varepsilon$. For Case 2 the result follows by induction hypothesis. For Case 6 we have that:

$$\begin{array}{lcl} \texttt{fmgu}^*(a \doteq \tau \cup \mathcal{E}, \mathcal{B}\,;\mathcal{A}_0) & = & \theta \cdot [a \mapsto \tau]\,;\mathcal{A}_1 \\ \text{where } \texttt{fmgu}^*([a \mapsto \tau]\mathcal{E}, \mathcal{B}\,;\mathcal{A}_0) & = & \theta\,;\mathcal{A}_1 \\ \text{and } a \notin ftv(\tau) \text{ and } \tau \neq b & & \end{array}$$

By induction hypothesis the property is true for θ, i.e. for every $f \in \mathcal{A}_0 - \mathcal{A}_1$ (therefore $f \notin ftv(\tau) \cup \{a\}$) with $f \in range(\theta)$ there exist $a', b \in \mathcal{B}$ or $a', b \in ftv([a \mapsto \tau]\mathcal{E}) - \mathcal{B}$ such that $\theta(a') = \theta(b) = f$. Hence the property is also true for $\theta \cdot [a \mapsto \tau]$. Cases 3,4, and 7 are similar to Case 6. Cases 8 and 9 follow directly by induction hypothesis. This leaves us with Case 5.

- Case 5. In this case we have that

$$\begin{array}{lcl}
\texttt{fmgu}^*(a \doteq b \cup \mathcal{E}, \mathcal{B}\,; \mathcal{A}_0) & = & \theta \cdot [a, b \mapsto c]\,; \mathcal{A}_1 \\
\text{where } \mathcal{A}_0 = \mathcal{A}_0' \cup \{c\} & & \\
\texttt{fmgu}^*([a, b \mapsto c]\mathcal{E}, \mathcal{B}\,; \mathcal{A}_0') & = & \theta\,; \mathcal{A}_1 \\
\text{and } (a, b \notin \mathcal{B} \vee a, b \in \mathcal{B}) & &
\end{array}$$

  By induction hypothesis the property is true for θ, i.e. for every $f \in \mathcal{A}_0' - \mathcal{A}_1$ with $f \in \mathit{range}(\theta)$ there exist $a', b' \in \mathcal{B}$ or $a', b' \in \mathit{ftv}([a \mapsto c, b \mapsto c]\mathcal{E}) - \mathcal{B}$, such that $\theta(a') = \theta(b') = f$. We need to show that for every $f \in \mathcal{A}_0' \cup \{c\} - \mathcal{A}_1$ with $f \in \mathit{range}(\theta \cdot [a \mapsto c, b \mapsto c])$ there exist two variables both in $\mathcal{B}$ or both in $\mathit{ftv}(a \doteq b \cup \mathcal{E}) - \mathcal{B}$ such that they both map to f. We have two cases to consider.

  - If $c \notin \mathit{dom}(\theta)$ then we are trivially ok since we can take the same a and b.
  - If $c \in \mathit{dom}(\theta)$, then $c \notin \mathit{range}(\theta \cdot [a \mapsto c, b \mapsto c])$, therefore $\mathit{range}(\theta \cdot [a \mapsto c, b \mapsto c]) = \mathit{range}(\theta)$ and we get the result by induction hypothesis.

□

Corollary 2.21. *If* $\texttt{fmgu}(\tau_1 \doteq \tau_2\,; \mathcal{A}_0) = \theta\,; \mathcal{A}_1$ *then* θ *is biased and minimal.*

Proof. By Lemma 2.7 and the definition of `fmgu` we get that all the variable-to-variable mappings in θ are of the form

1. $[a \in \mathit{ftv}(\tau_2) \mapsto b \notin \mathit{ftv}(\tau_2)]$.
2. $[a \in \mathit{ftv}(\tau_1) \mapsto f \in \mathcal{F}]$.

where $\mathcal{F} = \mathcal{A}_0 - \mathcal{A}_1$. Hence the unifier satisfies the bias condition. For minimality, from Lemma 2.20, assuming that $[a \mapsto f \in \mathcal{F}]$ it must also be that there exist two other variables a', b' such that $a', b' \in \mathit{ftv}(\tau_2)$ or $a', b' \in \mathit{ftv}(\tau_1)$ and $\theta(a') = \theta(b') = f$. Also, since $a \in \mathit{dom}(\theta)$ we have that $a \in \mathit{ftv}(\tau_2)$ or $a \in \mathit{ftv}(\tau_1)$. Therefore, in every case there exists another variable c (which is either a' or b') such that $\theta(a) = \theta(c) = f$ and both $a, c \in \mathit{ftv}(\tau_2)$ or $a, c \notin \mathit{ftv}(\tau_2)$ $(a, b \in \mathit{ftv}(\tau_1))$. Therefore the minimality condition is also satisfied. □

Finally, given an arbitrary most general unifier, we can construct an *fmgu* with a simple iterative procedure.
Lemma 2.22. *If* $\mu \in \mathit{mgu}(\tau_1 \doteq \tau_2)$ *then there is an injective mapping* m *such that* $m \cdot \mu \in \mathit{fmgu}(\tau_1 \doteq \tau_2)$, $\mathit{dom}(m) \subseteq \mathit{ftv}(\tau_1, \tau_2)$ *and* $\mathit{range}(m) \in \mathcal{F}$.

Proof. Just consider any binding of the form $[a \in \mathit{ftv}(\tau_i) \mapsto b \in \mathit{ftv}(\tau_i)] \in \mu$ and create the mapping: $[b \mapsto f \in \mathcal{F}]$ where f is a fresh symbol. Compose the mapping with the current substitution (initially μ) and repeat for the new substitution. In every step, the number of pairs from τ_1 or τ_2 that are directly equated is reduced, so this procedure terminates. Moreover the mapping has only fresh symbols in its domain, and at each step they are chosen fresh so injectivity holds; and it is easy to confirm that the final substitution returned satisfies the rest of the properties of *fmgu* (by showing that these properties are preserved in every step of the construction). □

—Term Syntax—			
Terms	t, u	::=	$\nu \mid \lambda x_\sigma . t \mid t\ u$
		$\mid$	$\texttt{let}\ x\ \texttt{=}\ u\ \texttt{in}\ t$
		$\mid$	$\texttt{case}\ t[\sigma]\ \texttt{of}\ \overline{p\ \texttt{->}\ t}$
		$\mid$	$\Lambda \overline{a} . t \mid t\ [\overline{\sigma}]$
		$\mid$	$\texttt{fail}$
Atoms	ν	::=	$x \mid C$
Patterns	p	::=	$x_\sigma \mid C\ \overline{a}\ \overline{p}$
Values	v	::=	$\lambda p . t \mid \Lambda a . t \mid C[\overline{\sigma}]\ \overline{t}$
—Type Syntax—			
	σ	::=	$\forall \overline{a} . \tau$
	τ	::=	$\tau \to \tau \mid a \mid T\ \overline{\tau}$
—Contexts—			
	Γ, Δ	::=	$\cdot \mid \Gamma, \nu : \sigma$
Local Pattern Triples	K, L	::=	$\{\overline{a}, \Delta, \theta\}$

Figure 3: Syntax of target language

3 The Target Language

The target language syntax and semantics are given in Figures 3-5. We use letters S, T to denote term substitutions (contrast this to the usage of ϕ,ψ, or θ for type substitutions).

The language which resembles System-F contains annotated abstraction, application, `let`-binding (which does no type generalization, but serves only as a binder for term variables), case expressions, type abstractions and type applications, and an extra `fail` element to model a *pattern-match failure.* An interesting property of this language is that refinements also include a $\perp$ value, to denote failure of refinement. Presence of $\perp$ refinement value merely indicates an inaccessible branch. Instead of rejecting such programs (as the source language of the main paper does), our core language accepts programs that contain inaccessible branches. However the definition of composition of two refinements slightly changes:

$$\psi \cdot \theta = \begin{cases} \perp & \psi = \perp \text{ or } \theta = \perp \\ \psi' \text{s.t. } \forall \sigma . \psi'(\sigma) = \psi(\theta(\sigma)) & \text{otherwise} \end{cases}$$

Sometimes we write $\theta_\perp$ to denote a refinement that may be $\perp$.

Additionally, the target language uses a unification-based specification of refinement. However, in some cases no unification is necessary for a branch of a pattern match to type check (if for example the refinement is not used at all). Therefore our target language uses the notion of a *preunifier* which is a substitution "on the way" to the most general unifier.

Definition 3.1 (Preunifier). *We say that the idempotent substitution ψ is a preunifier of τ_1 and τ_2, and we write $\psi \in preunif(\tau_1 \doteq \tau_2)$ iff:*

1. $(\nexists \phi . \phi(\tau_1) = \phi(\tau_2)) \Leftrightarrow \psi = \perp$
2. *For every idempotent unifier ϕ such that $\phi(\tau_1) = \phi(\tau_2)$ there exists a substitution ψ_0 such that $\phi = \psi_0 \cdot \psi$.*

Notice that a most general unifier (if it exists) is also a preunifier, but it is not the case that any unifier is also a preunifier, because we cannot create any other unifier from it by composing it with some substitution.

First an auxiliary lemma about pre-unifiers.

Pattern matching $match(p, t) = (\phi\,; S)_\perp$

$$\frac{t \neq \texttt{fail}}{match(x_\sigma, t) = (\cdot\,; [x \mapsto t])}\ \textsc{MVar} \qquad \frac{}{match(p, \texttt{fail}) = \perp}\ \textsc{MFail}$$

$$\frac{match(\overline{p}, \overline{t}) = (\phi\,; S)_\perp}{match(C\ \overline{a}\ \overline{p}, C\ [\overline{\tau}]\ \overline{t}) = (\phi \cdot [\overline{a \mapsto \tau}]\,; S)_\perp}\ \textsc{MCon} \qquad \frac{}{match(C\ \overline{a}\ \overline{p}, D\ [\overline{\tau}]\ \overline{t}) = \perp}\ \textsc{MBot}$$

Pattern matching sequences $match(\overline{p}, \overline{t}) = S_\perp$

$$\frac{}{match(\cdot, \cdot) = (\emptyset\,; \emptyset)}\ \textsc{MMany-Empty} \qquad \frac{match(p, t) = (\phi_1\,; S_1) \quad match(\overline{p}, \overline{t}) = (\phi_2\,; S_2)_\perp}{match((p, \overline{p}), (t, \overline{t})) = (\phi_2 \cdot \phi_1\,; S_2 \cdot S_1)_\perp}\ \textsc{MMany-Cons}$$

$$\frac{match(p, t) = \perp}{match((p, \overline{p}), (t, \overline{t})) = \perp}\ \textsc{MMany-ConsF}$$

Pattern guarded evaluation $t \xrightarrow[p]{} t'$

$$\frac{t \rightarrow t'}{t \xrightarrow[(C\ \overline{a}\ \overline{p})]{} t'}\ \textsc{PGEval} \qquad \frac{t_i \xrightarrow[p_i]{} t_i'}{C\ [\overline{\tau}]\ t_1 \ldots t_i \ldots t_n \xrightarrow[(C\ \overline{a}\ \overline{p})]{} C\ [\overline{\tau}]\ t_1 \ldots t_i' \ldots t_n}\ \textsc{PGCon}$$

Evaluation relation $t \rightarrow t'$

$$\frac{}{\texttt{let}\ x = u\ \texttt{in}\ t \rightarrow [x \mapsto u]t}\ \textsc{ELet}$$

$$\frac{t \rightarrow t'}{t\ u \rightarrow t'\ u}\ \textsc{EApp1} \qquad \frac{}{(\lambda x_\sigma \,.\, t_0)\ u \rightarrow [\overline{x \mapsto u}]t_0}\ \textsc{EApp2} \qquad \frac{}{\texttt{fail}\ u \rightarrow \texttt{fail}}\ \textsc{EAppF}$$

$$\frac{t \rightarrow t'}{t\ [\overline{\sigma}] \rightarrow t'\ [\overline{\sigma}]}\ \textsc{ETApp1} \qquad \frac{}{(\Lambda \overline{a} \,.\, t_0)\ [\overline{\sigma}] \rightarrow [\overline{a \mapsto \sigma}]t_0}\ \textsc{ETApp2} \qquad \frac{}{\texttt{fail}\ [\overline{\sigma}] \rightarrow \texttt{fail}}\ \textsc{ETAppF}$$

$$\frac{u \xrightarrow[p]{} u'}{\begin{array}{c}\texttt{case}[\sigma]\ u\ \texttt{of}\ (p \texttt{->} t, \overline{p \texttt{->} t}) \rightarrow \\ \texttt{case}[\sigma]\ u'\ \texttt{of}\ (p \texttt{->} t, \overline{p \texttt{->} t})\end{array}}\ \textsc{ECase1} \qquad \frac{match(p, u) = (\phi\,; S)}{\texttt{case}[\sigma]\ u\ \texttt{of}\ (p \texttt{->} t, \overline{p \texttt{->} t}) \rightarrow S(\phi(t))}\ \textsc{ECase2}$$

$$\frac{\begin{array}{c} match(p, u) = \perp \\ \texttt{case}[\sigma]\ u\ \texttt{of}\ \overline{p \texttt{->} t} \rightarrow t' \end{array}}{\texttt{case}[\sigma]\ u\ \texttt{of}\ (p \texttt{->} t, \overline{p \texttt{->} t}) \rightarrow t'}\ \textsc{ECase3} \qquad \frac{}{\texttt{case}[\sigma]\ u\ \texttt{of}\ \cdot \rightarrow \texttt{fail}}\ \textsc{ECaseF}$$

Figure 4: Operational semantics of target language

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

$$\frac{\nu{:}\sigma \in \Gamma}{\Gamma \vdash \nu : \sigma}\ \text{VAR} \qquad \frac{\Gamma \vdash t : \sigma_1 \to \sigma_2 \quad \Gamma \vdash u : \sigma_1}{\Gamma \vdash t\ u : \sigma_2}\ \text{APP} \qquad \frac{\Gamma, x{:}\sigma_1 \vdash t : \sigma_2}{\Gamma \vdash \lambda x_{\sigma_1}.t : \sigma_1 \to \sigma_2}\ \text{ABS}$$

$$\frac{\Gamma \vdash t : \forall\overline{a}.\sigma_1}{\Gamma \vdash t\ [\overline{\sigma}] : [\overline{a \mapsto \sigma}]\sigma_1}\ \text{TAPP} \qquad \frac{\Gamma \vdash t : \sigma \quad \overline{a} \sharp \mathit{ftv}(\Gamma)}{\Gamma \vdash \Lambda\overline{a}.t : \forall\overline{a}.\sigma}\ \text{TABS} \qquad \frac{}{\Gamma \vdash \mathtt{fail} : \sigma}\ \text{FAIL}$$

$$\frac{\Gamma \vdash u : \sigma \quad \Gamma, x{:}\sigma \vdash t : \sigma_1}{\Gamma \vdash (\mathtt{let}\ x\mathtt{=}u\ \mathtt{in}\ t) : \sigma_1}\ \text{LET} \qquad \frac{\Gamma \vdash u : \sigma_1 \quad \Gamma \vdash p_i \to t_i : \sigma_1 \to \sigma_2}{\Gamma \vdash \mathtt{case}[\sigma_2]\ u\ \mathtt{of}\ \overline{p\mathtt{->}t} : \sigma_2}\ \text{CASE}$$

$$\boxed{\text{Pattern alternative } \Gamma \vdash p \to t : \sigma_p \to \sigma_t}$$

$$\frac{\Gamma, \{\emptyset, \cdot, \emptyset\} \vdash p : \sigma_p \blacktriangleright \{\overline{a}, \Delta, \bot\}}{\Gamma \vdash p \to t : \sigma_p \to \sigma_t}\ \text{PAT-INACCESS} \qquad \frac{\Gamma, \{\emptyset, \cdot, \emptyset\} \vdash p : \sigma_p \blacktriangleright \{\overline{a}, \Delta, \theta\} \quad \theta(\Gamma \cup \Delta) \vdash \theta(t) : \theta(\sigma_t) \quad \mathit{ftv}(\Gamma, \sigma_p, \sigma_t) \sharp \overline{a}}{\Gamma \vdash p \to t : \sigma_p \to \sigma_t}\ \text{PAT-REFINE}$$

$$\boxed{\Gamma, K_1 \vdash p : \sigma \blacktriangleright K_2}$$

$$\frac{}{\Gamma, \{\overline{a}, \Delta, \bot\} \vdash p : \sigma \blacktriangleright \{\overline{a}, \Delta, \bot\}}\ \text{PBOT}$$

$$\frac{x \notin \mathit{dom}(\Delta) \quad \theta(\sigma_1) = \theta(\sigma_2)}{\Gamma, \{\overline{a}, \Delta, \theta\} \vdash x_{\sigma_1} : \sigma_2 \blacktriangleright \{\overline{a}, \Delta, x{:}\sigma_1, \theta\}}\ \text{PVAR} \qquad \frac{C : \forall\overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b} \sharp \overline{a} \quad \theta(\sigma) = T\ \overline{\tau}_3 \quad \psi \in \mathit{preunif}(\overline{\tau}_3, \overline{\tau}_2) \quad \Gamma, \{\overline{a}\overline{b}, \Delta, \psi \cdot \theta\} \vdash^{\mathit{fold}} \overline{p : \tau_1} \blacktriangleright K}{\Gamma, \{\overline{a}, \Delta, \theta\} \vdash C\ \overline{b}\ \overline{p} : \sigma \blacktriangleright K}\ \text{PCON}$$

$$\boxed{\Gamma, K_1 \vdash^{\mathit{fold}} \overline{p : \tau} \blacktriangleright K_2}$$

$$\frac{}{\Gamma, K \vdash^{\mathit{fold}} \cdot \blacktriangleright K}\ \text{F-BASE} \qquad \frac{\Gamma, K_1 \vdash p : \tau \blacktriangleright K_2 \quad \Gamma, K_2 \vdash^{\mathit{fold}} \overline{p : \tau} \blacktriangleright K_3}{\Gamma, K_1 \vdash^{\mathit{fold}} (p : \tau), \overline{p : \tau} \blacktriangleright K_3}\ \text{F-REC}$$

Figure 5: Target language typing rules

Lemma 3.2. *If $\psi \in \mathit{preunif}(\tau_1, \tau_2)$ then there exists a $\psi_\phi \in \mathit{preunif}(\phi(\tau_1), \phi(\tau_2))$ and a ψ_r such that $\psi_r \cdot \psi = \psi_\phi \cdot \phi$.*

Proof. First, if $\phi(\tau_1)$ and $\phi(\tau_2)$ are not unifiable then simply take $\psi_\phi =\bot$ and $\psi_r =\bot$. If they are unifiable, then it must also be that τ_1, τ_2 are unifiable. In this case by the definition of pre-unifiers we know that for every unifier δ, such that $\delta(\tau_1) = \delta(\tau_2)$, it is the case that there exists a ψ_r with $\delta = \psi_r \cdot \psi$. Take ψ_ϕ to be a most general unifier of $\phi(\tau_1)$ and $\phi(\tau_2)$—hence ψ_ϕ is also a preunifier. Then $\psi_\phi \cdot \phi$ is a unifier of τ_1 and τ_2 and therefore there exists a ψ_r such that $\psi_r \cdot \psi = \psi_\phi \cdot \phi$, as required. ☐

The next lemma shows that without loss of generality we can assume that the variables bound by a pattern

are as fresh as we may require.
Lemma 3.3.

1. *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \sigma \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\}$ *and* r *is a permutation of* $\overline{a}_2$ *to a fresh set of variables* $\overline{b}_2$ *and* $\overline{b}_1 = r(\overline{a}_1)$ *then*
$$\Gamma, \{\overline{b}_1, r\Delta_1, \theta_1 r^{-1}\} \vdash r(p) : r(\sigma) \blacktriangleright \{\overline{b}_2, r\Delta_2, \theta_2 r^{-1}\}$$

2. *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{\mathit{fold}} \overline{p : \sigma} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\}$ *and* r *is a permutation of* $\overline{a}_n$ *to a fresh set of variables* $\overline{b}_n$ *and* $\overline{b}_1 = r(\overline{a}_1)$ *then*
$$\Gamma, \{\overline{b}_1, r\Delta_1, \theta_1 r^{-1}\} \vdash^{\mathit{fold}} \overline{r(p) : r(\sigma)} \blacktriangleright \{\overline{b}_n, r\Delta_n, \theta_n r^{-1}\}$$

Proof. We prove the two claims simultaneously by induction on the height of the derivations. For the first part, we proceed with case analysis on the last rule used.

- Case PBOT. In this case the result follows trivially by rule PBOT again.
- Case PVAR. In this case we have that
$$\Gamma, \{\overline{a}, \Delta, \theta\} \vdash x_{\sigma_1} : \sigma_2 \blacktriangleright \{\overline{a}, \Delta, x : \sigma_1, \theta\} \tag{1}$$
given that
$$x \notin \mathit{dom}(\Delta) \quad \theta(\sigma_1) = \theta(\sigma_2) \tag{2}$$
Assume an appropriate renaming r such that $r(\overline{a}) = \overline{b}$. To show that
$$\begin{aligned}\Gamma, \{\overline{b}, r\Delta, \theta r^{-1}\} &\vdash x_{r(\sigma_1)} : r(\sigma_2)\\ &\blacktriangleright \{\overline{b}, r\Delta, x : r(\sigma_1), \theta r^{-1}\}\end{aligned}$$
using rule PVAR we have to show that $\theta r^{-1}(r(\sigma_1)) = \theta r^{-1}(r(\sigma_2))$. But the latter is true because of (2).
- Case PCON. In this case we have that
$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash C\ \overline{b}\ \overline{p} : \sigma \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{3}$$
given that
$$\begin{array}{r}C : \forall \overline{b} . \overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b} \sharp \overline{a}_1\\ \theta_1(\sigma) = T\ \overline{\tau}_3 \quad \psi \in \mathit{preunif}(\overline{\tau}_3, \overline{\tau}_2)\\ \Gamma, \{\overline{a}_1 \overline{b}, \Delta_1, \psi \cdot \theta_1\} \vdash^{\mathit{fold}} \overline{p : \tau_1} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\}\end{array}$$
Assume a renaming of $\overline{a}_n$ to completely fresh variables. Notice that this includes the $\overline{b}$. We have that $\theta_1 r^{-1}(r(\sigma)) = T\ \overline{\tau}_3$. Then, by induction hypothesis (second part) we get
$$\begin{aligned}\Gamma, \{r(\overline{a}_1 \overline{b}), r\Delta_1, \psi \cdot \theta_1 r^{-1}\} &\vdash^{\mathit{fold}} \overline{p : r(\tau_1)}\\ &\blacktriangleright \{r(\overline{a}_n), r\Delta_n, \theta_n r^{-1}\}\end{aligned}$$
Finally, applying rule PCON finishes the case.

For the second part, the case of F-BASE is trivial. The case of F-REC follows by induction hypothesis for the first and the second part, and application of rule F-REC again. $\square$

Lemma 3.4 (Substitution for pattern checking). *The following are true:*

1. *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \sigma \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\}$ *and* θ_1' *is one such* $\exists \phi_1$ *with* $\phi_1 \cdot \theta_1 = \theta_1' \cdot \phi$ *then* $\phi\Gamma, \{\overline{a}_1, \phi\Delta_1, \theta_1'\} \vdash \phi(p) : \phi(\sigma) \blacktriangleright \{\overline{a}_2, \phi\Delta_2, \theta_2'\}$ *and* $\exists \phi_2$ *such that* $\phi_2 \cdot \theta_2 = \theta_2' \cdot \phi$.

2. *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\}$ *and* θ_1' *is one such* $\exists \phi_1$ *with* $\phi_1 \cdot \theta_1 = \theta_1' \cdot \phi$ *then* $\phi\Gamma, \{\overline{a}_1, \phi\Delta_1, \theta_1'\} \vdash \overline{\phi(p) : \phi(\tau)} \blacktriangleright \{\overline{a}_n, \phi\Delta_n, \theta_n'\}$ *and* $\exists \phi_n$ *such that* $\phi_n \cdot \theta_n = \theta_n' \cdot \phi$.

Proof. We prove the two parts simultaneously by induction on the height of the derivations. For each, we assume that both hold for derivations of smaller height. For the first part we have to consider the following cases:

- Case PBOT. Trivially follows by PBOT again, since in this case $\theta_i = \perp$, $\theta_i' = \perp$.

- Case PVAR. In this case we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash x_{\sigma_1} : \sigma_2 \blacktriangleright \{\overline{a}_1, \Delta, x : \sigma_1, \theta_1\} \tag{1}$$

given that

$$\theta_1(\sigma_1) = \theta_1(\sigma_2) \tag{2}$$

To show that

$$\phi\Gamma, \{\overline{a}_1, \phi\Delta_1, \theta_1'\} \vdash x_{\phi\sigma_1} : \phi\sigma_2 \blacktriangleright \{\overline{a}_1, \phi\Delta_1, x : \phi\sigma_1, \theta_1'\} \tag{3}$$

using PVAR it is enough to show that $\theta_1'\phi(\sigma_1) = \theta_1'\phi(\sigma_1)$, but since $\exists \phi_1$ with $\phi_1 \cdot \theta_1 = \theta_1' \cdot \phi$ it is enough to show that $\phi_1\theta_1(\sigma_1) = \phi_1\theta_1(\sigma_2)$ but this follows from (2). Taking the same ϕ_1 finishes the case.

- Case PCON. In this case we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash C\ \overline{b}\ \overline{p} : \sigma \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{4}$$

given that

$$C : \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b} \sharp \overline{a}_1 \tag{5}$$

$$\theta_1(\sigma) = T\ \overline{\tau}_3 \tag{6}$$

$$\psi \in \mathit{preunif}(\overline{\tau}_3, \overline{\tau}_2) \tag{7}$$

$$\Gamma, \{\overline{a}_1 \overline{b}, \Delta_1, \psi \cdot \theta_1\} \vdash^{\mathit{fold}} \overline{p : \tau_1} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{8}$$

We also know that for some ϕ_1, $\phi_1 \cdot \theta_1 = \theta_1' \cdot \phi$. From (5), since constuctors have closed types, it is also the case that

$$C : \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \phi\Gamma \quad \overline{b} \sharp \overline{a}_1 \tag{9}$$

From (6) we have that $\phi_1\theta_1(\sigma) = T\ \phi_1\overline{\tau}_3$ or equivalently

$$\theta_1'(\phi(\sigma)) = T\ \phi_1\overline{\tau}_3 \tag{10}$$

We can assume without loss of generality (see Lemma 3.3) that $\overline{b}$ are completely fresh and in this case let

$$\psi' \in \mathit{preunif}(\phi_1\overline{\tau}_3, \overline{\tau}_2) = \mathit{preunif}(\phi_1\overline{\tau}_3, \phi_1(\overline{\tau}_2)) \tag{11}$$

By Lemma 3.2 we get that such a ψ' exists and moreover there exists a ψ_r such that $\psi_r \cdot \psi = \psi' \cdot \phi_1$, which implies $\psi_r \cdot \psi \cdot \theta_1 = \psi' \cdot \phi_1 \cdot \theta_1$, which in turn means

$$\psi_r \cdot (\psi \cdot \theta_1) = (\psi' \cdot \theta_1') \cdot \phi \tag{12}$$

This in turn is enough to apply the induction hypothesis on (8) to get

$$\phi\Gamma, \{\overline{a}_1 \overline{b}, \phi\Delta_1, \psi' \cdot \theta_1'\} \vdash^{\mathit{fold}} \overline{\phi(p) : \phi(\tau_1)} \blacktriangleright \{\overline{a}_n, \phi\Delta_n, \theta_n'\} \tag{13}$$

such that there exists a subsitution ψ_r' with $\psi_r' \cdot \theta_n = \theta_n' \cdot \phi$. Then we can apply rule PCON using equations (9), (10), (11), and (13) to get the result.

For the second part, the case F-BASE follows trivially by our assumptions. For the case of F-REC we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{fold} (p : \tau), \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{14}$$

given that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \tau \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\} \tag{15}$$

$$\Gamma, \{\overline{a}_2, \Delta_2, \theta_2\} \vdash^{fold} \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{16}$$

We also have that for some ψ, $\psi \cdot \theta_1 = \theta_1' \cdot \phi$. By induction hypothesis we get that

$$\phi\Gamma, \{\overline{a}_1, \phi\Delta_1, \theta_1'\} \vdash \phi(p) : \phi(\tau) \blacktriangleright \{\overline{a}_2, \phi\Delta_2, \theta_2'\} \tag{17}$$

and exists some ψ such that $\psi \cdot \theta_2 = \theta_2' \cdot \phi$. By induction then for (16), equation (17), and application of F-REC again we are done.

□

Corollary 3.5 (Substitution for pattern checking). *If* $\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash p : \sigma \blacktriangleright \{\overline{a}, \Delta, \theta\}$ *then* $\phi\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash \phi(p) : \phi(\sigma) \blacktriangleright \{\overline{a}, \phi\Delta, \theta'\}$ *such that there exists a* ψ_r *with* $\psi_r \cdot \theta = \theta' \cdot \phi$.

Proof. Assume without loss of generality that $\overline{a}\#vars(\phi)$. Then by Lemma 3.4 with $\theta_1 = \theta_1' = \varepsilon$ and $\phi_1 = \phi$. □

Lemma 3.6 (Type substitution). *The following are true:*

1. *If* $\Gamma \vdash p \to t : \sigma_1 \to \sigma_2$ *then* $\phi(\Gamma) \vdash \phi(p) \to \phi(t) : \phi(\sigma_1) \to \phi(\sigma_2)$.
2. *If* $\Gamma \vdash t : \sigma$ *then* $\phi\Gamma \vdash t : \phi\sigma$.

Proof. We prove the two claims simultaneously by induction on the height of the derivations.

For the first part, in both the cases of PAT-INACCESS and PAT-REFINE we have that $\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash p : \sigma_1 \blacktriangleright \{\overline{a}, \Delta, \theta_\perp\}$. Then, by Corollary 3.5 we have that $\phi\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash \phi(p) : \phi(\sigma_1) \blacktriangleright \{\overline{a}, \phi\Delta, \theta_\perp'\}$ such that there exists a ψ_r with $\psi_r \cdot \theta_\perp = \theta_\perp' \cdot \phi$. Then, if $\theta_\perp = \perp$ it must be that $\theta_\perp' = \perp$ as well and rule PAT-INACCESS yields $\phi(\Gamma) \vdash \phi(p) \to \phi(t) : \phi(\sigma_1) \to \phi(\sigma_2)$. If on the other hand $\theta_\perp = \theta$, then, if $\theta_\perp' = \perp$ again we can apply PAT-INACCESS. If however $\theta_\perp' = \theta'$ then we know by the premises of PAT-REFINE that

$$\theta(\Gamma, \Delta) \vdash \theta(t) : \theta(\sigma_2) \tag{1}$$

By induction hypothesis then $\psi_r\theta(\Gamma, \Delta) \vdash \psi_r\theta(t) : \psi_r\theta(\sigma_2)$, or equivalently $\theta'(\phi\Gamma, \phi\Delta) \vdash \theta'\phi(t) : \theta'\phi(\sigma_2)$ and applying PAT-REFINE yields $\phi(\Gamma) \vdash \phi(p) \to \phi(t) : \phi(\sigma_1) \to \phi(\sigma_2)$ as required.

For the first part, the cases of VAR, APP, ABS, FAIL, TAPP, and LET are straightforward. The case of CASE follows by induction hypothesis for the first part and rule CASE again. For the case TABS we have that $\Gamma \vdash \Lambda\overline{a}.t : \forall\overline{a}.\sigma$ given that $\Gamma \vdash t : \sigma$ and $\overline{a}\#ftv(\Gamma)$. Consider the substitution $\phi \cdot [\overline{a \mapsto b}]$, where $\overline{b}$ are completely fresh. Then by induction hypothesis $\phi\Gamma \vdash \phi[\overline{a \mapsto b}]t : \phi[\overline{a \mapsto b}]\sigma$ and by rule TABS $\phi\Gamma \vdash \Lambda\overline{b}.\phi[\overline{a \mapsto b}]t : \forall\overline{b}.\phi[\overline{a \mapsto b}]\sigma$, or equivalently $\phi\Gamma \vdash \phi(\Lambda\overline{a}.t) : \phi(\forall\overline{a}.\sigma)$, as required. □

Lemma 3.7 (Canonical forms). *If* $\vdash v : \sigma_1 \to \sigma_2$ *then* $v = \lambda x_\sigma.t$. *If* $\vdash v : T\ \overline{\tau}$ *then* $v = C\ [\overline{\tau}]\ \overline{t}$. *If* $\vdash v : \forall\overline{a}.\sigma$ *then* $v = \Lambda\overline{a}.t$.

Proof. Straightforward. □

In the following we will use the letter S to denote *term* substitutions.

Definition 3.8. *We say that a term substitution S is* compatible *with a context Δ, and denote this with $\Delta \Vdash S$, iff $\forall x \in dom(S)$ it is the case that $x : \sigma \in \Delta$ and $\vdash S(x) : \sigma$.*
Lemma 3.9 (Term Substitution). *If $\Delta \Vdash S$ and $\Delta, \Gamma \vdash t : \sigma$ then $\Gamma \vdash S(t) : \sigma$.*

Proof. Straightforward. □

Corollary 3.10 (Term Substitution). *If $\Delta \Vdash S$ and $\Delta \vdash t : \sigma$ then $\vdash S(t) : \sigma$.*

Proof. Directly follows by Lemma 3.9. □

The following important shows that the type substitution induced by the operational semantics extends the refinement that the static semantics yield, but having the nice property that its domain is a subset of the freshly introduced pattern type variables.
Theorem 3.11 (Matching yields compatible substitutions). *The following are true:*

1. *If*
   - $\vdash t : \phi_1(\sigma)$ *(A0)*
   - $dom(\phi_1) \subseteq \overline{a}_1$ *(A1)*
   - $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \sigma \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\}$ *(A2)*
   - $\theta_1 \neq \perp$ *(A3)*
   - $match(p, t) = (\phi_2\,; S)$ *(A4)*
   - $\phi_1 = \psi_r^1 \cdot \theta_1$ *(A5)*

   then
   - $dom(\phi_2) = \overline{a}_2 - \overline{a}_1$ *and* $\theta_2 \neq \perp$ *(B1)*
   - *there exists* ψ_r^2 *s.t.* $\phi_2 \cdot \phi_1 = \psi_r^2 \cdot \theta_2$ *(B2)*
   - $\phi_2\phi_1(\Delta) \Vdash S$ *(B3)*
2. *If*
   - $\vdash \overline{t} : \phi_1(\overline{\sigma})$ *(C0)*
   - $dom(\phi_1) \subseteq \overline{a}_1$ *(C1)*
   - $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{fold} \overline{p : \sigma} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\}$ *(C2)*
   - $\theta_1 \neq \perp$ *(C3)*
   - $match(\overline{p}, \overline{t}) = (\phi_n\,; S)$ *(C4)*
   - $\phi_1 = \psi_r^1 \cdot \theta_1$ *(C5)*

   then
   - $dom(\phi_n) = \overline{a}_n - \overline{a}_1$ *and* $\theta_n \neq \perp$ *(D1)*
   - *there exists* ψ_r^n *s.t.* $\phi_n \cdot \phi_1 = \psi_r^n \cdot \theta_n$ *(D2)*
   - $\phi_n\phi_1(\Delta) \Vdash S$ *(D3)*

Proof. We prove the two claims simultaneously by induction on the matching derivation and the sequence matching derivation. For each part the induction hypothesis asserts that both parts hold for derivations of smaller height. We proceed with case analysis on the last rule used. For the first part, because of (A5) MBOT and MFAIL cannot happen and the theorem is vacuously true. We need to consider the rest of the cases.

- Case MCON. Here we have by our assumptions that

$$match(C\ \overline{b}\ \overline{p}, C\ \overline{\tau}, \overline{t}) = (\phi_n \cdot [\overline{b \mapsto \tau}]\ ;S) \tag{1}$$
$$match(\overline{p}, \overline{t}) = (\phi_n\ ;S) \tag{2}$$
$$\vdash C\ \overline{\tau}\ \overline{t} : \phi_1(\sigma) \quad dom(\phi_1) \subseteq \overline{a}_1 \tag{3}$$
$$\phi_1 = \psi_r^1 \cdot \theta_1 \tag{4}$$
$$\{\overline{a}_1, \Delta_1, \theta_1\} \vdash C\ \overline{b}\ \overline{p} : \sigma \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\} \tag{5}$$
$$\theta_2 \neq \perp \tag{6}$$

From (5) and inversion on typing we get

$$C : \forall \overline{b}.\overline{\tau}_1 \rightarrow T\ \overline{\tau}_2 \quad \overline{b} \sharp \overline{a}_1 \tag{7}$$
$$\theta_1(\sigma) = T\ \overline{\tau}_3 \quad \psi \in preunif(\overline{\tau}_3, \overline{\tau}_2) \tag{8}$$
$$\{\overline{a}_1\overline{b}, \Delta_1, \psi \cdot \theta_1\} \vdash^{fold} \overline{p : \tau_1} \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\} \tag{9}$$

From (3), (4), and (8) we get that $\vdash C\ \overline{\tau}\ \overline{t} : \psi_r^1(T\ \overline{\tau}_3)$ which by inversion on typing gives:

$$\vdash C\ \overline{\tau}\ \overline{t} : T\ \psi_r^1(\overline{\tau}_3) \tag{10}$$
$$\psi_r^1(\overline{\tau}_3) = [\overline{b \mapsto \tau}]\overline{\tau}_2 \tag{11}$$

We wish to apply the induction hypothesis for (9). Then, by inversion on typing we get (B0), since $[\overline{b \mapsto \tau}] \cdot \phi(\overline{\tau}_1) = [\overline{b \mapsto \tau}]\overline{\tau}_1$. By (10) $\vdash \overline{t} : [\overline{b \mapsto \tau}]\phi_1(\overline{\tau}_1)$ and $dom([\overline{b \mapsto \overline{\tau}}] \cdot \phi_1) \subseteq \overline{a}_1\overline{b}$ so we get (B1). By (2) we get (B4).

Now, to satisfy (B5) we need to find ${\psi_r^1}'$ such that $[\overline{b \mapsto \tau}]\phi_1 = {\psi_r^1}' \cdot \psi \cdot \theta$. But by equation (4) we know that

$$[\overline{b \mapsto \tau}] \cdot \phi_1 = [\overline{b \mapsto \tau}] \cdot \psi_r^1 \cdot \theta_1 \tag{12}$$

But, from equation (11) $[\overline{b \mapsto \tau}]\psi_r^1$ is a unifier (easy check) of $\overline{\tau}_3, \overline{\tau}_2$, and by (8) ψ is a pre-unifier and $\psi \neq \perp$, that is there exists a ${\psi_r^1}'$ such that

$$[\overline{b \mapsto \tau}]\psi_r^1 = {\psi_r^1}' \cdot \psi \tag{13}$$

Using (12) and (13) we get $[\overline{b \mapsto \tau}]\phi_1 = {\psi_r^1}' \cdot \psi \cdot \theta$ as required for (B5). For (B3) we just saw that $\psi \neq \perp$, therefore $\psi \cdot \theta_1 \neq \perp$. Then, by induction hypothesis (second part) we get that $dom(\phi_n) = \overline{a}_2 - \overline{a}_1\overline{b}$ or $dom(\phi_n \cdot [\overline{b \mapsto \tau}]) \subseteq \overline{a}_2 - \overline{a}_1$, and $\theta_n \neq \perp$. Moreover there exists a ψ_r^n such that $\phi_n \cdot [\overline{b \mapsto \tau}]\phi_1 = \psi_r^n \cdot \theta_n$ and $\phi_n[\overline{b \mapsto \tau}]\phi_1(\Delta_2 - \Delta_1) \Vdash S$ as required.

- Case MVAR. We have that

$$\vdash t : \phi_1(\sigma_2) \quad dom(\phi_1) \subseteq \overline{a}_1 \tag{14}$$
$$\{\overline{a}_1, \Delta_1, \theta_1\} \vdash x_{\sigma_1} : \sigma_2 \blacktriangleright \{\overline{a}_1, \Delta_1, x{:}\sigma_1, \theta_1\} \tag{15}$$
$$\theta_1(\sigma_1) = \theta_1(\sigma_2) \tag{16}$$
$$match(x, t) = (\varepsilon\ ; [x \mapsto t]) \tag{17}$$
$$\phi_1 = \psi_r^1 \cdot \theta_1 \tag{18}$$

Then $\phi_2 = \varepsilon$ with $dom(\phi_2) = \emptyset$. Moreover take $\psi_r^2 = \psi_r^1$. Certainly now $\varepsilon \cdot \phi_1(x{:}\sigma_1) \Vdash [x \mapsto t]$, since $\phi_1(\sigma_1) = \psi_r^1\theta_1(\sigma_2) = \psi_r^1\theta_1(\sigma_1) = \phi_1(\sigma_2)$.

For the second part we have two cases to consider (the case of MMANY-CONSF cannot happen).

- Case MMANY-EMPTY. In this case take $\psi_r^2 = \psi_r^1$ and $\phi_2 = \varepsilon$, S is empty.

- Case MMANY-CONS. Here we have that

$$match((p,\overline{p}),(t,\overline{t})) = (\phi_0' \cdot \phi_0 \,;S_2 \cdot S_1) \tag{19}$$

given that

$$match(p,t) = (\phi_0 \,; S_1) \tag{20}$$
$$match(\overline{p},\overline{t}) = (\phi_0' \,; S_2) \tag{21}$$

We also have by assumptions that

$$\vdash t : \phi_1(\sigma) \quad \vdash \overline{t} : \phi_1(\overline{\sigma}) \quad dom(\phi_1) \subseteq \overline{a}_1 \tag{22}$$
$$\{\overline{a}_1, \Delta_2, \theta_1\} \vdash p : \sigma \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\} \tag{23}$$
$$\{\overline{a}_2, \Delta_2, \theta_2\} \vdash \overline{p : \sigma} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{24}$$
$$\theta_1 \neq \perp \quad \phi_1 = \psi_r^1 \cdot \theta_1 \tag{25}$$

By induction hypothesis (first part) we get that $dom(\phi_0) = \overline{a}_2 = \overline{a}_1$ and there exists a ψ_r such that

$$\phi_0 \cdot \phi_1 = \psi_r \cdot \theta_2 \tag{26}$$

and $\phi_0 \cdot \phi_1(\Delta_2 - \Delta_1) \Vdash S_1$. Additionally $\theta_2 \neq \perp$. Then, by the type substitution lemma we get that $\vdash \overline{t} : \phi_0\phi_1(\overline{\sigma})$. By (26) we get condition (B5). Then by induction hypothesis $\phi_0' \cdot \phi_0 \cdot \phi_1 = \psi_r^n \cdot \theta_n$ for some ψ_r^n and moreover $\phi_0' \cdot \phi_0 \cdot \phi_1(\Delta_n - \Delta_2) \Vdash S_2$, therefore (and using the type substitution lemma as well) $\phi_0' \cdot \phi_0 \cdot \phi_1(\Delta_n - \Delta_1) \Vdash S_2 \cdot S_1$ as required, and additonally $\theta_n \neq \perp$.

□

The following is a more intuitive form of the theorem,
Corollary 3.12 (Matching yields compatible substitutions). *If $\vdash t : \sigma$, $\{\emptyset, \cdot, \varepsilon\} \vdash p : \sigma \blacktriangleright \{\overline{a}, \Delta, \theta\}$, and $match(p,t) = (\phi\,; S)$ then $dom(\phi) \subseteq \overline{a}$, $\theta \neq \perp$, and there exists ψ_r such that $\phi = \psi_r \cdot \theta$, and $\phi(\Delta) \Vdash S$.*

Proof. Direct consequence of Theorem 3.11. □

Theorem 3.13 (Subject reduction). *The following are true:*

1. *If $\vdash t : \sigma$ and $t \xrightarrow[p]{} t'$ then $\vdash t' : \sigma$.*
2. *If $\vdash t : \sigma$ and $t \xrightarrow[t]{}{}'$ then $\vdash t' : \sigma$.*

Proof. We prove the two parts simultaneously by induction on the size of t. For each part the induction hypothesis asserts both claims for terms of smaller size. For the first case, the case of PGEVAL follows by induction (second part) hypothesis. The case of PGCON follows by inversion on typing, induction hypothesis (first part), and consecutive applications of TAPP and APP rules.

For the second part, the proof is straightforward, appealing to the substitution lemma in the case of rule LET and APP, and to the type substitution lemma in the case of TAPP. The case of CASE is more interesting. In this case we have

$$\Gamma \vdash \texttt{case}[\sigma_2]\ u\ \texttt{of}\ \overline{p\texttt{->}t} : \sigma_2 \tag{1}$$

given that

$$\Gamma \vdash u : \sigma_1 \tag{2}$$
$$\forall i\,.\,\Gamma \vdash p_i \to t_i : \sigma_1 \to \sigma_2 \tag{3}$$

We consider two cases depending on the size of $\overline{p\texttt{->}t}$.

- $\overline{p\text{->}t} = \cdot$. Then by ECASEF the term steps to `fail`, and $\vdash \texttt{fail} : \sigma_2$ using FAIL.
- $\overline{p\text{->}t} = (p\text{->}t), \overline{p'\text{->}t'}$. By (3) we have that $\vdash p \to t : \sigma_1 \to \sigma_2$, or by inversion $\{\emptyset, \cdot, \varepsilon\} \vdash p : \sigma_1 \blacktriangleright \{\overline{a}, \Delta, \theta\}$. we have three cases to consider for evaluation:
  - Subcase ECASE1. In this case $u \xrightarrow[p]{} u'$ and by induction hypothesis $\vdash u' : \sigma_1$ and by applying rule CASE we are done.
  - Subcase ECASE2, which implies $match(p, u) = (\phi; S)$. In this case we have that by induction, subject reduction holds for all terms of size smaller or equal than the size of u. Then by Corollary 3.12 we get that
  $$\phi(\Delta) \Vdash S \tag{4}$$
  and there exists a ψ_r such that $\phi = \psi_r \cdot \theta$ with $dom(\phi) = \overline{a}$. By inversion on typing we then have that $\theta(\Delta) \vdash \theta(t) : \theta(\sigma_2)$ and by the substitution lemma, $\psi_r\theta(\Delta) \vdash \psi_r\theta(t) : \psi_r\theta(\sigma_2)$, or equivalently $\phi(\Delta) \vdash \phi(\tau) : \phi(\sigma_2)$. But also by inversion on typing it must be that $\overline{a}\#ftv(\sigma_1, \sigma_2)$ and since $dom(\phi) = \overline{a}$ the last equation can be rewritten as
  $$\phi(\Delta) \vdash \phi(\tau) : \sigma_2 \tag{5}$$
  From (4) and (5) and the (term) substitution lemma we get $\vdash S\phi(t) : \sigma_2$ as required.
  - Subcase ECASE3. The result follows by induction hypothesis, since $\texttt{case}[\sigma_2]\ u\ \texttt{of}\ \overline{p'\text{->}t'}$ is of smaller size and well typed using rule CASE.

□

Theorem 3.14 (Progress for pattern matching). *The following are true:*

1. *If*
   - $\vdash t : \phi_1(\sigma)$ *and progress holds for all terms smaller than or equal to the size of* t *(A0)*
   - $dom(\phi_1) \subseteq \overline{a}_1$ *(A1)*
   - $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \sigma \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\}$ *(A2)*
   - $\theta_1 \neq \perp$ *(A3)*
   - $\phi_1 = \psi_r^1 \cdot \theta_1$ *(A4)*

   then
   - $match(p, t) = \perp$, **or** *(B1)*
   - $match(p, t) = (\phi_2\,; S)$ *s.t.* $\phi_2 \cdot \phi_1 = \psi_r^2 \cdot \theta_2$ *and* $dom(\phi_2) \subseteq (\overline{a}_2 - \overline{a}_1)$, **or** *(B2)*
   - $t \xrightarrow[p]{} t'$ *(B3)*
2. *If*
   - $\vdash \overline{t} : \phi_1(\overline{\sigma})$ *and progress holds for all terms smaller than or equal to the size of* $max(\overline{t})$ *(C0)*
   - $dom(\phi_1) \subseteq \overline{a}_1$ *(C1)*
   - $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{fold} \overline{p : \sigma} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\}$ *(C2)*
   - $\theta_1 \neq \perp$ *(C3)*
   - $\phi_1 = \psi_r^1 \cdot \theta_1$ *(C4)*

   then
   - $match(\overline{p}, \overline{t}) = \perp$, **or** *(D1)*

- $match(\overline{p}, \overline{t}) = (\phi_n\,; S)$ *s.t.* $\phi_n \cdot \phi_1 = \psi_r^2 \cdot \theta_2$ *and* $dom(\phi_n) \subseteq (\overline{a}_n - \overline{a}_1)$, **or** *(D2)*
- $\exists t_i \in \overline{t}\,.\, t_i \xrightarrow[p_i]{} t_i'$ *(D3)*

Proof. We prove the two claims simultaneously by induction on the matching typing derivation and the sequence matching typing derivation. For each part the induction hypothesis asserts that both parts hold for derivations of smaller height. We proceed with case analysis on the last rule used. For the first part we have two cases to consider.

- Case PVAR. In this case we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash x_{\sigma_1} : \sigma_2 \blacktriangleright \{\overline{a}_1, \Delta_1, x : \sigma_1, \theta_1\} \tag{1}$$

  If $t = \texttt{fail}$ then we are done by rule MFAIL. If $t \neq \texttt{fail}$ we are done by MVAR.

- Case PCON. By assumptions we have that

$$\vdash t : \phi_1(\sigma) \tag{2}$$
$$dom(\phi_1) \subseteq \overline{a}_1 \tag{3}$$
$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash C\ \overline{b}\ \overline{p} : \sigma \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{4}$$
$$\theta_1 \neq \perp \tag{5}$$
$$\phi_1 = \psi_r^1 \cdot \theta_1 \tag{6}$$

  Also assume that progress holds for all terms smaller than or equal to the size of t. Moreover by (4) we have

$$C : \forall \overline{b}\,.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b} \,\sharp\, \overline{a}_1 \tag{7}$$
$$\theta_1(\sigma) = T\ \overline{\tau}_3 \quad \psi \in preunif(\overline{\tau}_3, \overline{\tau}_2) \tag{8}$$
$$\Gamma, \{\overline{a}_1 \overline{b}, \Delta_1, \psi \cdot \theta_1\} \vdash^{fold} \overline{p : \tau_1} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{9}$$

  By (2) we have three cases to consider:

  - $t = \texttt{fail}$. Then we are done by MFAIL.
  - $t \to t'$. Then we are done using PGEVAL.
  - $t = v$. Then we have that $\vdash v : \phi_1(\sigma) = T\ \psi_r^1(\overline{\tau}_3)$. By canonical forms lemma, it must be that $v = D\ [\overline{\tau}]\ \overline{t}$ for some D If $v \neq C\ [\overline{\tau}]\ \overline{t}$ (i.e. $D \neq C$ or the arities don't agree) then we are done by MBOT. If on the other hand $v = C\ [\overline{\tau}]\ \overline{t}$ then it must be that $\psi_r^1(\overline{\tau}_3) = [\overline{b \mapsto \tau}]\overline{\tau}_2$, therefore it is easy to see that $[\overline{b \mapsto \tau}] \cdot \psi_r^1$ is a unifier of $\overline{\tau}_3$ and $\overline{\tau}_2$, therefore $\psi \neq \perp$ and by the definition of preunifiers $[\overline{b \mapsto \tau}] \cdot \psi_r^1 = \psi_r' \cdot \psi$. Moreover $[\overline{b \mapsto \tau}]\overline{\tau}_1 = [\overline{b \mapsto \tau}]\phi_1(\overline{\tau}_1)$ and $\psi_r' \cdot \psi \cdot \theta = [\overline{b \mapsto \tau}] \cdot \psi_r^1 \cdot \theta_1 = [\overline{b \mapsto \tau}] \cdot \phi_1$. Then we can apply the induction hypothesis where we have three cases:
    - $match(\overline{p}, \overline{t}) = \perp$. In this case we are done by applying rule MCON.
    - $match(\overline{p}, \overline{t}) = (\phi_2\,; S)$ with $\phi_2[\overline{b \mapsto \tau}]\phi_1 = \psi_r^n \cdot \theta_n$. In this case we are done again by MCON.
    - $\exists t_i \in \overline{t}$ such that $t_i \xrightarrow[p_i]{} t_i'$. In this case we are done by PGCON.

For the second part, the case of F-BASE follows by MMANY-EMPTY. For the case of F-REC we have the following:

$$\vdash \overline{t} : \phi_1(\overline{\sigma}) \tag{10}$$
$$dom(\phi_1) \subseteq \overline{a}_1 \tag{11}$$
$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{fold} (p_1 : \tau_1), \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{12}$$
$$\theta_1 \neq \perp \tag{13}$$
$$\phi_1 = \psi_r^1 \cdot \theta_1 \tag{14}$$

Additionally assume that progress holds for all terms smaller than or equal to the size of $max(\overline{t})$. Also

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p_1 : \tau_1 \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\} \tag{15}$$
$$\Gamma, \{\overline{a}_2, \Delta_2, \theta_2\} \vdash^{fold} \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{16}$$

By induction hypothesis for (15) we have the following possibilities.

- $match(p_1, t_1) = \perp$ in which case we are done by rule MMANY-CONSF.
- $t_1 \xrightarrow[p_1]{} t_1'$. This is one of our conclusions so we are done in this case.
- $match(p_1, t_1) = (\phi_1 ; S_1)$ such that $\phi_2\phi_1 = \psi_r^2\theta_2$. Then, by the substitution lemma it must also be that $\vdash \overline{t} : \phi_2\phi_1(\overline{\sigma})$. Moreover $dom(\phi_2 \cdot \phi_1) \subseteq \overline{a}_2$. Finally by Theorem 3.11 $\theta_2 \neq \perp$. Then we can apply the induction hypothesis for (16) to get the following possibilities:
  - $match(\overline{p}, \overline{t}) = (\phi_n ; S_n)$ with $\phi_1 \cdot \phi_2 \cdot \phi_n = \psi_r^n \cdot \theta_n$ and we are done by MMANY-CONS.
  - $match(\overline{p}, \overline{t}) = \perp$ and again rule MMANY-CONS applies.
  - $\exists t_i \in (\overline{t} - t_1)$ such that $t_i \xrightarrow[p_i]{} t_i'$. Again we are done because this is one of our conclusions.

□

Corollary 3.15 (Progress for pattern matching). *If $\vdash t : \sigma$ and progress holds for all terms with size smaller than or equal to the size of t, and $\{\emptyset, \cdot, \varepsilon\} \vdash p : \sigma \blacktriangleright \{\overline{a}, \Delta, \theta\}$ then either $match(p, t) = \perp$, or $match(p, t) = (\phi ; S)$, or $t \xrightarrow[p]{} t'$.*

Proof. Follows directly from Lemma 3.15. □

Theorem 3.16 (Progress). *If $\vdash t : \sigma$ then $t = \texttt{fail}$, or $t = v$, or $t \to t'$ for some t'.*

Proof. By induction on the size of t. We proceed by case analysis on the last rule used in the typing derivation. The case of VAR, ABS, TABS, and FAIL are straightforward. The case of LET follows directly by rule ELET. The cases of APP, and TAPP are easy checks, appealing to the canonical forms lemma. The interesting case is the one for CASE. We have that

$$\Gamma \vdash \texttt{case}[\sigma_2]\ u\ \texttt{of}\ \overline{p\texttt{->}t} : \sigma_2 \tag{1}$$

given that

$$\Gamma \vdash u : \sigma_1 \tag{2}$$
$$\Gamma \vdash p_i \to t_i : \sigma_1 \to \sigma_2 \tag{3}$$

If $\overline{p\texttt{->}t} = \cdot$ then we are done by rule ECASEF. If not, let $\overline{p\texttt{->}t} = (p\texttt{->}t), \overline{p'\texttt{->}t'}$, and then by Corollary 3.15 and using the induction hypothesis we see that either ECASE1, ECASE2, or ECASE3 is applicable and the proof is finished. □

Theorem 3.16 and Theorem 3.13 give us type safety.

3.1 Auxiliary lemmas

We present here some auxiliary properties of the core language, that are mainly used to establish a sound translation of source programs to target programs.

Lemma 3.17 (Pulling out substitutions in core language). *The following are true:*

1. *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash \psi(p) : \psi(\tau) \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\}$ *then* $\Gamma, \{\overline{a}_1, \Delta'_1, \theta_1 \cdot \psi\} \vdash p : \tau \blacktriangleright \{\overline{a}_2, \Delta'_2, \theta'_2\}$ *with* $\theta'_2 = \theta_2 \cdot \psi$ *and* $\psi(\Delta'_2 - \Delta'_1) = \Delta_2 - \Delta_1$.
2. *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{fold} \overline{\psi(p) : \psi(\tau)} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\}$ *then* $\Gamma, \{\overline{a}_1, \Delta'_1, \theta_1 \cdot \psi\} \vdash^{fold} \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta'_n, \theta'_n\}$ *with* $\theta'_n = \theta_n \cdot \psi$ *and* $\psi(\Delta'_n - \Delta'_1) = \Delta_n - \Delta_1$.

Proof. We prove the two claims simultaneously by induction on the height of the derivations. For the first part we have to consider the following cases.

- Case PBOT. Follows trivially.
- Case PVAR. In this case we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash x_{\psi(\sigma_1)} : \psi(\sigma_2) \blacktriangleright \{\overline{a}_1, \Delta_1, x{:}\psi(\sigma_1), \theta_1\} \tag{1}$$

  given that

$$\theta_1\psi(\sigma_1) = \theta_1\psi(\sigma_2) \tag{2}$$

  We wish to show that

$$\Gamma, \{\overline{a}_1, \Delta'_1, \theta_1 \cdot \psi\} \vdash x_{\sigma_1} : \sigma_2 \blacktriangleright \{\overline{a}_1, \Delta'_1, x{:}\sigma_1, \theta_1 \cdot \psi\} \tag{3}$$

  This will be derivable only if $\theta_1\psi(\sigma_1) = \theta_1\psi(\sigma_2)$ using PVAR but we have this from (2).

- Case PCON. In this case—and assume without loss of generality that $\overline{b}\#vars(\psi)$—we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash C\ \overline{b}\ \overline{\psi(p)} : \psi(\sigma) \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{4}$$

  given that

$$C : \forall \overline{b}. \overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b} \sharp \overline{a} \tag{5}$$
$$\theta\psi(\sigma) = T\ \overline{\tau}_3 \tag{6}$$
$$\psi_p \in preunif(\overline{\tau}_3, \overline{\tau}_2) \tag{7}$$
$$\Gamma, \{\overline{a}_1\overline{b}, \Delta_1, \psi_p \cdot \theta_1\} \vdash^{fold} \overline{\psi(p) : \tau_1} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{8}$$

  By (8) and induction hypothesis for the second part we get that

$$\Gamma, \{\overline{a}_1\overline{b}, \Delta'_1, \psi_p \cdot \theta_1 \cdot \psi\} \vdash^{fold} \overline{p : \tau_1} \blacktriangleright \{\overline{a}_n, \Delta'_n, \theta'_n\} \tag{9}$$

  such that $\theta'_n = \theta_n \cdot \psi$ and $\psi(\Delta'_2 - \Delta'_1) = \Delta_2 - \Delta_1$. From this, (5),(6), and (7) we can apply rule PCON to finish the case.

For the second part, the case of F-BASE is straightforward. For the case of F-REC we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash^{fold} (\psi(p) : \psi(\tau)), \overline{\psi(p) : \psi(\tau)} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{10}$$

given that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash \psi(p) : \psi(\tau) \blacktriangleright \{\overline{a}_2, \Delta_2, \theta_2\} \tag{11}$$
$$\Gamma, \{\overline{a}_2, \Delta_2, \theta_2\} \vdash^{fold} \overline{\psi(p) : \psi(\tau)} \blacktriangleright \{\overline{a}_n, \Delta_n, \theta_n\} \tag{12}$$

From (11) and induction hypothesis for the first part we get that $\Gamma, \{\overline{a}_1, \Delta'_1, \theta_1 \cdot \psi\} \vdash p : \tau \blacktriangleright \{\overline{a}_2, \Delta'_2, \theta'_2\}$ such that $\theta'_2 = \theta_2 \cdot \psi$ and $\psi(\Delta'_2 - \Delta'_1) = \Delta_2 - \Delta_1$. Then, by induction hypothesis for (12) we get $\Gamma, \{\overline{a}_2, \Delta'_2, \theta_2 \cdot \psi\} \vdash^{fold} \overline{p : \tau} \blacktriangleright \{\overline{a}_n, \Delta'_n, \theta'_n\}$ such that $\theta'_n = \theta_n \cdot \psi$ and $\psi(\Delta'_n - \Delta'_2) = \Delta_n - \Delta_2$ or $\psi(\Delta'_n - \Delta'_1) = \Delta_n - \Delta_1$. Applying rule F-BASE finishes the case. □

Lemma 3.18 (Equivalence of target pattern checking modulo refinement). *If* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \tau \blacktriangleright K$ *and* $\theta_1(\tau) = \theta_1(\tau')$ *then* $\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash p : \tau' \blacktriangleright K$.

Proof. The proof is by induction on the pattern checking derivation. The case of PBOT follows by directly applying rule PBOT. For the case of PVAR we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash x_{\tau_1} : \tau_2 \blacktriangleright \{\overline{a}_1, \Delta_1, x{:}\sigma_1, \theta_1\} \tag{1}$$

given that $\theta_1(\tau_1) = \theta(\tau_2)$. By assumptions, for any τ'_2 with $\theta_1(\tau'_2) = \theta_1(\tau_2) = \theta_1(\tau_1)$ we apply rule PVAR and we are done. In the case of PCON we have that

$$\Gamma, \{\overline{a}_1, \Delta_1, \theta_1\} \vdash C\ \overline{b}\ \overline{p} : \tau \blacktriangleright K \tag{2}$$

given that

$$C : \forall \overline{b}. \overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b} \sharp \overline{a}_1 \tag{3}$$

$$\theta_1(\tau) = T\ \overline{\tau}_3 \tag{4}$$

$$\psi \in \mathit{preunif}(\overline{\tau}_3, \overline{\tau}_2) \tag{5}$$

$$\Gamma, \{\overline{a}_1 \overline{b}, \Delta_1, \psi \cdot \theta_1\} \vdash^{fold} \overline{p : \tau_1} \blacktriangleright K \tag{6}$$

We have that $\theta_1(\tau) = \theta_1(\tau')$ so (4) is satisfied if τ is replaced for τ' and the result follows by applying rule PCON again. □

—Source language syntax—

$$
\begin{array}{rlcl}
\text{Atoms} & v & ::= & x \mid C \\
\text{Terms} & t, u & ::= & v \mid \lambda x\,.\,t \mid t\ u \\
 & & \mid & \texttt{let}\ x\ \texttt{=}\ u\ \texttt{in}\ t \\
 & & \mid & \texttt{let}\ x\texttt{::sig}\ \texttt{=}\ u\ \texttt{in}\ t \\
 & & \mid & \texttt{case}\ t\ \texttt{of}\ \overline{p\ \texttt{->}\ t} \\
\text{Patterns} & p & ::= & x \mid C\ \overline{p} \mid p\texttt{::tau} \\
\text{Type annotations} & \texttt{sig} & ::= & \texttt{forall}\ \overline{\texttt{a}}\texttt{.tau} \\
 & \texttt{tau} & ::= & \texttt{tau} \rightarrow \texttt{tau} \mid \texttt{a} \mid T\ \overline{\texttt{tau}} \\
\text{Polytypes} & \sigma & ::= & \forall \overline{a}.\tau \\
\text{Monotypes} & \tau, \upsilon & ::= & \tau \rightarrow \tau \mid a \mid T\ \overline{\tau}
\end{array}
$$

—Meta language syntax—

$$
\begin{array}{rlcl}
\text{Environments} & \Gamma, \Delta & ::= & \cdot \mid \Gamma, v{:}^m\, \sigma \mid \Gamma, \texttt{a} \hookrightarrow \tau \\
\text{Modifiers} & m, n & ::= & w \mid r \\
\text{Refinements} & \theta, \psi & ::= & [\overline{a \mapsto \tau}] \\
\text{Triples} & K, L & ::= & \{\overline{a}, \Delta, \theta\}
\end{array}
$$

—Annotation translation—

$$
\begin{array}{lcll}
[\![\texttt{a}]\!]_\Gamma & = & \Gamma(\texttt{a}) & \\
[\![\texttt{tau}_1 \rightarrow \texttt{tau}_2]\!]_\Gamma & = & [\![\texttt{tau}_1]\!]_\Gamma \rightarrow [\![\texttt{tau}_2]\!]_\Gamma & \\
[\![T\ \overline{\texttt{tau}}]\!]_\Gamma & = & T\ \overline{[\![\texttt{tau}]\!]_\Gamma} & \\
[\![\texttt{forall}\ \overline{\texttt{a}}\texttt{.tau}]\!]_\Gamma & = & \forall \overline{a}\,.\,[\![\texttt{tau}]\!]_{\Gamma, \overline{\texttt{a} \hookrightarrow a}} & \overline{a}\ \text{fresh}
\end{array}
$$

—Refinement application—

$$
\begin{array}{lcl}
\theta^r(\sigma) & = & \theta(\sigma) \\
\theta^w(\sigma) & = & \sigma \\
 & & \\
\theta(\cdot) & = & \cdot \\
\theta(\Gamma, x{:}^m\, \sigma) & = & \theta(\Gamma), x{:}^m\, \theta^m(\sigma) \\
\theta(\Gamma, \texttt{a} \hookrightarrow \tau) & = & \theta(\Gamma), \texttt{a} \hookrightarrow \theta(\tau)
\end{array}
$$

—Bindings of lexical variables—

$$
\begin{array}{lcl}
\mathit{bindings}(\cdot) & = & \emptyset \\
\mathit{bindings}(\Delta, \texttt{a} \hookrightarrow a) & = & \{a\} \cup \mathit{bindings}(\Delta) \\
\mathit{bindings}(\Delta, x :^m \sigma) & = & \mathit{bindings}(\Delta)
\end{array}
$$

Figure 6: Syntax of source language and types

4 Source language

The syntax of the source language is given in Figure 6.

We present the source type system in Figure 7 and Figure 8. For pattern matching we first study a syntactic restriction of patterns given by the following syntax:

$$
\begin{array}{lcl}
p & ::= & q \mid C\ \overline{q} \\
q & ::= & x \mid (q\texttt{::tau})
\end{array}
$$

Notice that this restriction still allows open type annotations under constructor bindings—nevertheless does not allow nesting of constructor based patterns. We will start with studying the properties of this simplification, the difference with the nested implementation is that refinement is threaded through.

Figure 9 presents the "shallow" pattern matching judgement. We have made the freshness condition explicit by passing an "avoidance" set X down to the main pattern checking judgement. This set contains variables

$$\boxed{\Gamma \vdash t :^m \tau \rightsquigarrow t'}$$

$$\frac{\nu :^n \forall \overline{a}.\tau \in \Gamma}{\Gamma \vdash \nu :^m [\overline{a \mapsto \upsilon}]\tau \rightsquigarrow v\ \overline{\upsilon}}\ \text{VAR}$$

$$\frac{\begin{array}{c}\Gamma \vdash t :^w \tau_1 \to \tau_2 \rightsquigarrow t' \\ \Gamma \vdash u :^w \tau_1 \rightsquigarrow u'\end{array}}{\Gamma \vdash t\ u :^m \tau_2 \rightsquigarrow t'\ u'}\ \text{APP} \qquad \frac{\Gamma, x :^m \tau_1 \vdash t :^m \tau_2 \rightsquigarrow t'}{\Gamma \vdash \backslash x\,.\,t :^m \tau_1 \to \tau_2 \rightsquigarrow \lambda x_{\tau_1}\,.\,t'}\ \text{ABS}$$

$$\frac{\begin{array}{c}\Gamma \vdash^{poly} u : \sigma \upharpoonright^{m_x} \rightsquigarrow u' \\ \Gamma, x :^{m_x} \sigma \vdash t :^m \tau \rightsquigarrow t'\end{array}}{\Gamma \vdash (\texttt{let}\ x\texttt{=}u\ \texttt{in}\ t) :^m \tau \rightsquigarrow (\texttt{let}\ x_\sigma\texttt{=}u'\ \texttt{in}\ t')}\ \text{LET-W}$$

$$\frac{\begin{array}{c}[\![\texttt{forall}\ \overline{\texttt{a}}\texttt{.tau}]\!]_\Gamma = \forall \overline{a}.\tau \quad \overline{a} \# \Gamma \\ \Gamma, x :^r \forall \overline{a}.\tau, \overline{\texttt{a} \hookrightarrow a} \vdash u :^r \tau \rightsquigarrow u' \qquad \Gamma, x :^r \forall \overline{a}.\tau \vdash t :^m \tau \rightsquigarrow t'\end{array}}{\Gamma \vdash (\texttt{let}\ x\texttt{::(forall}\ \overline{\texttt{a}}\texttt{.tau) =}\ u\ \texttt{in}\ t) :^m \tau \rightsquigarrow (\texttt{let}\ x_{\forall \overline{a}.\tau}\texttt{=}u'\ \texttt{in}\ t')}\ \text{LET-R}$$

$$\frac{\begin{array}{c}\Gamma \vdash u : \tau_p \upharpoonright^{m_p} \rightsquigarrow u' \\ \Gamma \vdash \overline{p \to t} :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t \rightsquigarrow \overline{p'_i \to t'_i}\end{array}}{\Gamma \vdash (\texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t}) :^{m_t} \tau_t \rightsquigarrow (\texttt{case}\ u'\ \texttt{of}\ \overline{p'\texttt{->}t'})}\ \text{CASE}$$

$$\frac{\begin{array}{c}[\![\texttt{forall}\ \overline{\texttt{a}}\texttt{.tau}]\!]_\Gamma = \forall \overline{a}.\tau \quad \overline{a} \# \Gamma \\ \Gamma, \overline{\texttt{a} \hookrightarrow a} \vdash t :^r \tau \rightsquigarrow t'\end{array}}{\Gamma \vdash (t\texttt{::forall}\ \overline{\texttt{a}}\texttt{.tau}) :^m [\overline{a \mapsto \upsilon}]\tau \rightsquigarrow (\Lambda \overline{a}\,.\,t')\ \overline{\upsilon}}\ \text{SIG}$$

Figure 7: Typing rules (main paper)

that the *fmgus* must avoid, and includes the type of the pattern, the variables in the environment and the return type of the branch. This is necessary to formally guarrantee freshness. In turn, rule PCON-R makes sure that the "fresh" variables of the *fmgu* ψ are indeed fresh from X.

Other than that there are no significant differences between this figure and the figure of the main paper—the reader may observe some superfluous threading of triples, but we keep this threading to show explicitly the similarity with the main paper; a complete re-organization of our relations would require more explanatory material to re-establish the connection with nested patterns where the latter were introduced.

We start by showing that the language equipped with shallow pattern matching can be translated to the core language of the previous section.

Lemma 4.1 (Shallow q-pattern translation). *If*

- $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X q :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \rightsquigarrow p'$ *(A1)*
- $\theta, \Gamma, \Delta_1, \tau \# \overline{c}$ *and* $bindings(\Delta_2) \subseteq X$ *(A2)*
- $\theta_c \cdot r = r \cdot \theta \cdot \psi$ *s.t.* $dom(r) \# \overline{e} \times \overline{c}, X, \Delta_1, \Gamma, \tau$ *(A3)*
- $dom(\psi) \subseteq \overline{c}$ *(A4)*

$$\boxed{\Gamma \vdash t : \tau \rceil^{m} \rightsquigarrow t'}$$

$$\frac{\begin{array}{c} \nu :^r \forall \overline{a}.\overline{\tau} \to \tau \in \Gamma \\ \Gamma \vdash u_i : [\overline{a \mapsto \tau_a}]\tau_i \rceil^{m_i} \rightsquigarrow u'_i \\ \overline{a}_r = \{a \in \overline{a} \mid \exists \tau_i \,.\, a \in ftv(\tau_i) \wedge m_i = r\} \\ m = \begin{cases} r & ftv(\tau) \subseteq \overline{a}_r \\ w & \text{otherwise} \end{cases} \end{array}}{\Gamma \vdash \nu\ \overline{u} : [\overline{a \mapsto \tau_a}]\tau \rceil^{m} \rightsquigarrow \nu\ [\overline{\tau_a}]\ \overline{u'}}\ \text{SCR-APP}
\qquad
\frac{[\![\texttt{tau}]\!]_\Gamma = \tau \quad \Gamma \vdash t :^r \tau \rightsquigarrow t'}{\Gamma \vdash (t\,\texttt{::}\,\texttt{tau}) : \tau \rceil^{r} \rightsquigarrow t'}\ \text{SCR-TSIG}$$

$$\frac{\begin{array}{c} t \neq \nu\ \overline{u} \text{ with } \nu \text{ bound rigidly} \quad t \neq (t_0\,\texttt{::}\,\texttt{tau}) \\ \Gamma \vdash t :^w \tau \rightsquigarrow t' \end{array}}{\Gamma \vdash t : \tau \rceil^{w} \rightsquigarrow t'}\ \text{SCR-OTHER}$$

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

$$\frac{\begin{array}{c} [\![\texttt{forall}\ \overline{\texttt{a}}\texttt{.tau}]\!]_\Gamma = \forall \overline{a}.\tau \quad \overline{a}\#\Gamma \\ \Gamma, \overline{\texttt{a} \hookrightarrow a} \vdash t :^r \tau \rightsquigarrow t' \end{array}}{\Gamma \vdash^{poly} (t\,\texttt{::}\,\texttt{forall}\ \overline{\texttt{a}}\texttt{.tau}) : \forall \overline{a}.\tau \rceil^{r} \rightsquigarrow \Lambda \overline{a}.t'}\ \text{SCR-SSIG}
\qquad
\frac{\begin{array}{c} t \neq (t_0\,\texttt{::}\,\texttt{sig}) \\ \Gamma \vdash t : \tau \rceil^{m_1} \rightsquigarrow t' \\ \overline{a} = ftv(\tau) - ftv(\Gamma) \end{array}}{\Gamma \vdash^{poly} t : \forall \overline{a}.\tau \rceil^{m_1} \rightsquigarrow \Lambda \overline{a}.t'}\ \text{SCR-GEN}$$

Figure 8: Scrutinee modifier inference (main paper)

then $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta_c\} \vdash p' : \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta_c\}$ *in the core language and* $dom(r)\#\Delta_2$*,* $\Delta_2\#\overline{c}$*.*

If

- $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash^{fold}_X \overline{q :^m \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\} \rightsquigarrow \overline{p'}$ *(C1)*
- $\theta, \Gamma, \Delta_1, \overline{\tau}\#\overline{c}$ *and* $bindings(\Delta_n) \subseteq X$ *(C2)*
- $\theta_c \cdot r = r \cdot \theta \cdot \psi$ *s.t.* $dom(r)\#\overline{e} \times \overline{c}, X, \Delta_1, \Gamma, \overline{\tau}$ *(C3)*
- $dom(\psi) \subseteq \overline{c}$ *(C4)*

then $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta_c\} \vdash^{fold} \overline{p' : \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta_c\}$ *in the core language and* $dom(r)\#\Delta_n$*,* $\Delta_n\#\overline{c}$*.*

Proof. We prove the two claims simultaneously by induction on the height of the derivations. For the second part the case of F-BASE is trivial and the case of F-REC follows by induction hypothesis for the first part and the second part. The cases for the first part are more interesting.

- Case PVAR. We have that

  $$\Gamma, \{\overline{e} \times \overline{c}, \Delta, \theta\} \vdash_X x :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, x :^m \tau, \theta\} \rightsquigarrow x_\tau$$

  Moreover from (A2)-(A4) we get

  $$\theta, \Gamma, \Delta, \tau\#\overline{c} \quad bindings(\Delta, x :^m \tau) \subseteq X \tag{1}$$
  $$\theta_c \cdot r = r \cdot \theta \cdot \psi \quad dom(r)\#\overline{e} \times \overline{c}, X, \Delta, \Gamma, \tau \tag{2}$$
  $$dom(\psi) \subseteq \overline{c} \tag{3}$$

To apply the target rule PVAR we need to show that $\theta_c(\tau) = \theta_c(\tau)$ which is trivially true. Applying the core PVAR rule we get

$$\Gamma, \{\overline{e} \times \overline{c}, \Delta, \theta\} \vdash x_\tau :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, x{:}\tau, \theta\}$$

Moreover, from (1) it is also $\Delta, x{:}\tau \# \overline{c}$ and from (2) it is $dom(r)\#\Delta, x{:}\tau$ as well.

- Case PANN. In this case we have that

$$\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X (q\texttt{::tau}) :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \rightsquigarrow p'$$

given that

$$\overline{\mathsf{b}} = ftv(\texttt{tau}) - ftv(\Gamma, \Delta) \tag{4}$$
$$[\![\texttt{tau}]\!]_{\Gamma, \Delta_1, \overline{\mathsf{b} \hookrightarrow b}} = \tau_s \tag{5}$$
$$\overline{b} \text{ distinct} \quad \overline{b}\#dom(\theta) \quad \overline{b}\#bindings(\Delta_1) \tag{6}$$
$$\theta(\tau_s) = \theta^m(\tau) \tag{7}$$
$$\Gamma, \{\overline{e} \times \overline{c}, (\Delta_1, \overline{\mathsf{b} \hookrightarrow b}), \theta\} \vdash_X q :^r \tau_s \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \rightsquigarrow p' \tag{8}$$

As a first step we will try to apply the induction hypothesis to (). In order to do this we need to show (A2)-(A4). For (A2) we need to show that $ftv(\tau_s)\#\overline{c}$. But $ftv(\tau_s) \subseteq \overline{b} \cup dom(\Gamma, \Delta_1)$, therefore we only need to show that $\overline{b}\#\overline{c}$. But $\overline{b} \subseteq ftv(\theta(\tau_s))$ since $\overline{b}\#dom(\theta)$, therefore $\overline{b} \in \theta^m(\tau)$. But both $\theta, \tau\#\overline{c}$, therefore $\overline{b}\#\overline{c}$. (A3) follows directly. For (A4) we need to additionally show that $dom(r)\#\tau_s$, which with a similar argument amounts to showing that $\overline{b}\#\tau_s$. But we know that $bindings(\Delta_2) \subseteq X$, therefore $\overline{b} \subseteq X$ and $X\#dom(r)$. Hence we can apply the induction hypothesis which give us that

$$\Gamma, \{\overline{e} \times \overline{c}, (\Delta_1, \overline{\mathsf{b} \hookrightarrow b}), \theta_c\} \vdash p' :^r \tau_s \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta_c\}$$

Moreover $dom(r)\#\Delta_2$ and $\Delta_2\#\overline{c}$. Now, $\theta_c(\tau_s) = \theta_c \cdot r(\tau_s) = r\theta\psi(\tau_s) = r\theta(\tau_s)$ since $dom(\psi) \subseteq \overline{c}$ and $ftv(\tau_s)\#\overline{c}$. This in turn gives $\theta_c(\tau_s) = r\theta^m(\tau)$. If $m = w$ then we get $\theta_c(\tau_s) = \tau$ and assuming that θ_c is idempotent $\theta_c(\tau_s) = \theta_c(\tau)$. If on the other hand $m = r$ then $r\theta^m(\tau) = r\theta(\tau) = r\theta\psi(\tau) = \theta_c(\tau)$. So, in both cases $\theta_c(\tau_s) = \theta_c(\tau)$. Lemma 3.18 finishes the case.

□

Lemma 4.2 (Shallow p-pattern translation). *If*

- $\Gamma \vdash_X p :^r \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \rightsquigarrow p'$ *(A1)*
- $\Gamma, \tau\#\overline{e} \times \overline{c}$ *and* $bindings(\Delta) \subseteq X, \#\overline{c}$ *(A2)*

then $\Gamma \vdash p' : \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta_c\}$ *and there exists a mapping* r *such that* $r \cdot \theta = \theta_c \cdot r$ *with* $dom(r)\#\Gamma, \tau, \Delta, X, \overline{e} \times \overline{c}$.

If

- $\Gamma \vdash_X p :^w \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \varepsilon\} \rightsquigarrow p'$ *(C1)*
- $\Gamma, \tau\#\overline{e} \times \overline{c}$ *and* $bindings(\Delta) \subseteq X, \#\overline{c}$ *(C2)*

then $\Gamma \vdash p' : \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \psi_c\}$ *such that* $\psi_c[\Delta_c] = \Delta$ *and* $dom(\psi) \subseteq \overline{c}$.

Proof. We show the two claims simultaneously by induction on the height of the derivations. We have to consider the following cases.

- Case PCON-W. In this case we have that

$$\Gamma \vdash_X C\ \overline{q} :^w T\ \overline{\tau}_3 \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\} \rightsquigarrow C\ \overline{b}\ \overline{p_i'}$$

given that

$$C :^r \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad (1)$$

$$\overline{b}_c = \overline{b} \cap \mathit{ftv}(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \quad (2)$$

$$\psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \quad (3)$$

$$m_i = \begin{cases} r & \mathit{ftv}(\tau_{1i})\#\overline{b}_c \\ w & \text{otherwise} \end{cases} \quad (4)$$

$$\Gamma, \{\overline{b}_e \times \overline{b}_c, \cdot, \varepsilon\} \vdash_X^{\mathit{fold}} \overline{q_i :^{m_i} \psi(\tau_{1i})} \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \varepsilon\} \rightsquigarrow \overline{p_i'} \quad (5)$$

Moreover, by assumptions we have that $\Gamma, \tau\#\overline{b}_e \times \overline{b}_c$, and $\mathit{bindings}(\Delta_n) \subseteq X, \#\overline{b}_c$. It is also the case that $\psi \in \mathit{preunif}(\overline{\tau}_3, \overline{\tau}_2)$, since ψ must be a most general unifier of $\overline{\tau}_3$ and $\overline{\tau}_3$. To be able then to apply rule PCON of the core language we will first apply Lemma 4.1 to equation (5). We need to show requirements (C2)-(C4) of that lemma. For (C2) we need to show additionally that $\mathit{ftv}(\psi(\overline{\tau}_1))\#\overline{b}_c$. But, if τ_{1i} is a purely existential argument then it only contains variables from $\overline{b}_e$. On the other hand, if not, assume by contradiction that there is a variabls $b \in \mathit{dom}(\psi)$ such that $b \in \tau_{1i}$. Then $\psi(b) \in \mathit{ftv}(\overline{\tau}_3)$ by the construction of ψ which is a contradiction. For (C3) and (C4) let us take $\theta = \varepsilon$, $r = \varepsilon$, $\theta_c = \psi$; then $\mathit{dom}(\psi) \subseteq \overline{b}_c$. Applying Lemma 4.1 then give us

$$\Gamma, \{\overline{b}_e \times \overline{b}_c, \cdot, \varepsilon\} \vdash^{\mathit{fold}} \overline{p_i' :^{m_i} \psi(\tau_{1i})} \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \varepsilon\}$$

Applying Lemma 3.17 gives

$$\Gamma, \{\overline{b}_e \times \overline{b}_c, \cdot, \psi\} \vdash^{\mathit{fold}} \overline{p_i' :^{m_i} \tau_{1i}} \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_{nc}, \psi\}$$

such that $\psi[\Delta_{nc}] = \Delta_n$ as required.

- Case PCON-R. In this case we have

$$\Gamma \vdash_X C\ \overline{q} :^r \tau \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\} \rightsquigarrow C\ \overline{b}\ \overline{p_i'}$$

given that

$$C :^r \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad (6)$$

$$\tau = T\ \overline{\tau}_3 \quad \psi \in \mathit{fmgu}(\overline{\tau}_3 \doteq \overline{\tau}_2) \quad (7)$$

$$(\mathit{ftv}(\psi) - \mathit{ftv}(\overline{\tau}_3, \overline{\tau}_2))\#\overline{b}, X, \Gamma \quad (8)$$

$$\Gamma, \{\overline{b} \times \emptyset, \cdot, \psi\} \vdash_X^{\mathit{fold}} \overline{q_i :^r \tau_{1i}} \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\} \rightsquigarrow \overline{p_i'} \quad (9)$$

Moreover by assumptions $\Gamma, \tau\#\overline{b} \times \emptyset$ and $\mathit{bindings}(\Delta_n) \subseteq X$. From Corollary 2.4 we know that there exists a mapping m and an mgu, μ such that such that $\mu \cdot m = m \cdot \psi$ such that $\mathit{dom}(m)$ is fresh from $\overline{\tau}_3, \overline{\tau}_2$. From (8) it must be that $\mathit{dom}(m)\#\overline{b}, \Gamma, X, \overline{\tau}_3, \overline{\tau}_2$. Consider now the preconditions of Lemma 4.1. (C1) is satisfied by (9). (C2) is satisfied by assumptions, and $\overline{c} = \emptyset$. For (C3) and (C4) take $\psi = \varepsilon$ and $r = m$. Then from Lemma 4.1 we get that

$$\Gamma, \{\overline{b} \times \emptyset, \cdot, \psi\} \vdash^{\mathit{fold}} \overline{p_i' :^r \tau_{1i}} \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\}$$

Applying rule PCON in the core language finishes the case.

- Case PQ. In this case

$$\Gamma \vdash_X q :^m \tau \blacktriangleright \{\emptyset, \Delta, \emptyset\} \rightsquigarrow p'$$

given that

$$\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash_X q :^m \tau \blacktriangleright \{\emptyset, \Delta, \emptyset\} \rightsquigarrow p'$$

and the case is finished by applying Lemma 4.1 and taking $r = \varepsilon$ when $m = r$, or $\psi_c = \varepsilon$ when $m = w$.

$\square$

The following lemmas talk about the variables in the returned environment and refinement.

Lemma 4.3 (Vars of q-returned triples).

1. *Assume* $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X q :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\}$ *and* $vars(\theta)\#\overline{c}$, $\tau\#\overline{c}$, *and* $bindings(\Delta_1) \cup ftv(\Gamma)\#\overline{c}$. *Then also* $bindings(\Delta_2)\#\overline{c}$.
2. *Assume* $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X^{fold} \overline{q :^m \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\}$ *and* $vars(\theta)\#\overline{c}$. $\overline{\tau}\#\overline{c}$ *and* $bindings(\Delta_1) \cup ftv(\Gamma)\#\overline{c}$. *Then also* $bindings(\Delta_2)\#\overline{c}$.

Proof. The two claims are proved simultaneously by induction over the height of the derivations. The second part is straightforward, so we only examine the first part. We have the following cases:

- Rule PVAR. Straightforward sinced the bindings of the returned environment do not change.
- Rule PANN. In this case we have that

$$\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X (q\texttt{::tau}) :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\}$$

given that

$$\overline{\mathtt{b}} = ftv(\texttt{tau}) - ftv(\Gamma, \Delta) \tag{1}$$

$$[\![\texttt{tau}]\!]_{\Gamma, \Delta_1, \overline{\mathtt{b} \hookrightarrow b}} = \tau_s \tag{2}$$

$$\overline{b} \text{ distinct} \quad \overline{b}\#dom(\theta) \quad \overline{b}\#bindings(\Delta_1) \tag{3}$$

$$\theta(\tau_s) = \theta^m(\tau) \tag{4}$$

$$\begin{aligned} \Gamma, \{\overline{e} \times \overline{c}, (\Delta_1, \overline{\mathtt{b} \hookrightarrow b}), \theta\} &\vdash_X q :^r \tau_s \\ &\blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \end{aligned} \tag{5}$$

We know that $\tau\#\overline{c}$ and $\theta\#\overline{c}$. Moreover $ftv(\tau_s) \subseteq ftv(\Gamma) \cup bindings(\Delta_1) \cup \overline{b}$ and we know that $\Gamma, bindings(\Delta_1)\#\overline{c}$. To show that $\overline{b}\#\overline{c}$ just observe that since $\overline{b} \notin dom(\theta)$ it must be that $\overline{b} \in \theta^m(\tau)\#\overline{c}$. Therefore by induction hypothesis $bindings(\Delta_2)\#\overline{c}$ as well.

$\square$

Lemma 4.4 (Vars of p-returned triples). *If* $\Gamma \vdash_X p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\}$ *and* $\tau\#\overline{e} \times \overline{c}$, $\Gamma\#\overline{e} \times \overline{c}$, *then* $vars(\theta)\#\overline{c}$ *and* $bindings(\Delta)\#\overline{c}$.

Proof. The derivation ends with one of rules PQ, PCON-R or PCON-W. If it ends with PQ we get the result by Lemma 4.3. If the last rule used is PCON-W we have that:

$$\Gamma \vdash_X C\ \overline{q} :^w T\ \overline{\tau}_3 \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\} \rightsquigarrow C\ \overline{b}\ \overline{p_i'}$$

given that

$$C :^r \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b}_c = \overline{b} \cap \mathit{ftv}(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \tag{1}$$

$$\psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \quad m_i = \begin{cases} r & \mathit{ftv}(\tau_{1i}) \# \overline{b}_c \\ w & \text{otherwise} \end{cases} \tag{2}$$

$$\Gamma, \{\overline{b}_e \times \overline{b}_c, \cdot, \emptyset\} \vdash^{\mathit{fold}}_X \overline{q_i :^{m_i} \psi(\tau_{1i})} \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\} \rightsquigarrow \overline{p'_i} \tag{3}$$

Since $\theta = \varepsilon$ we have that $\mathit{vars}(\theta)\#\overline{c}$. The second claim is satisfied by Lemma 4.3, since for every i it is that $\overline{b}_c \# \psi(\tau_{1i})$—otherwise it would have to be that $\overline{b}_c \cap \mathit{ftv}(\overline{\tau}_3) \neq \emptyset$.

If the last rule used is PCON-R we have that:

$$\Gamma \vdash_X C\ \overline{q} :^r \tau \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\} \rightsquigarrow C\ \overline{b}\ \overline{p'_i}$$

given that

$$C :^r \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \tau = T\ \overline{\tau}_3 \tag{4}$$

$$\psi \in \mathit{fmgu}(\overline{\tau}_3 \doteq \overline{\tau}_2) \quad (\mathit{ftv}(\psi) - \mathit{ftv}(\overline{\tau}_3, \overline{\tau}_2)) \# \overline{b}, X, \Gamma \tag{5}$$

$$\Gamma, \{\overline{b} \times \emptyset, \cdot, \psi\} \vdash^{\mathit{fold}}_X \overline{q_i :^r \tau_{1i}} \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\} \rightsquigarrow \overline{p'_i} \tag{6}$$

$$\tag{7}$$

In this case we know that $\mathit{vars}(\psi)\#\emptyset$. Moreover $\overline{\tau}_1\#\emptyset$ and since $\overline{\tau}_1$ have all r modifiers, Lemma 4.3 finishes the case. □

Theorem 4.5 (Translation). *Assume that the core language uses* idempotent *refinements.*

1. *If* $\Gamma \vdash t :^m \tau \rightsquigarrow t'$ *then* $\Gamma \vdash t' : \tau$.
2. *If* $\Gamma \vdash t : \tau \upharpoonright^m \rightsquigarrow t'$ *then* $\Gamma \vdash t' : \tau$.
3. *If* $\Gamma \vdash^{\mathit{poly}} t : \sigma \upharpoonright^m \rightsquigarrow t'$ *then* $\Gamma \vdash t' : \sigma$.
4. *If* $\Gamma \vdash p \to t :^{\langle m_1, m_2 \rangle} \tau_1 \to \tau_2 \rightsquigarrow p' \to t'$ *then* $\Gamma \vdash p' \to t' : \tau_1 \to \tau_2$.

Proof. The claims are proved simultaneously by induction on the height of the derivation. For each part the induction hypothesis asserts all parts for derivations of smaller height. The only interesting part is the last. The rule used is PAT and we have

$$\Gamma \vdash p \to t :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t \rightsquigarrow p' \to t'$$

given that

$$\Gamma \vdash_{(\mathit{ftv}(\Gamma, \tau_t, \tau_p), \overline{e} \times \overline{c})} p :^{m_p} \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \rightsquigarrow p' \tag{1}$$

$$\mathit{ftv}(\Gamma, \tau_p, \tau_t) \# \overline{e} \times \overline{c} \quad \mathit{bindings}(\Delta) \subseteq \overline{e} \times \overline{c} \tag{2}$$

$$\theta(\Gamma \cup \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t) \rightsquigarrow t' \tag{3}$$

We consider two cases depending on m_p.

- $m_p = w$. In this case we have that $\theta = \varepsilon$ and we wish to apply Lemma 4.2. (C1) is easy to show; for (C2) we also need to show that $\mathit{bindings}(\Delta)\#\overline{c}$. But this follows from Lemma 4.4. Therefore we get that in the core language

$$\Gamma \vdash p' : \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_c, \psi_c\}$$

  with $\mathit{dom}(\psi_c) \subseteq \overline{c}$. and $\psi_c[\Delta_c] = \Delta$. Moreover we know that $\Gamma, \Delta \vdash t' : \tau_t$ by induction hypothesis and since $\Gamma, \tau_t \# \overline{c}$ we have $\psi_c(\Gamma, \Delta_c) \vdash t' : \psi_c(\tau_t)$. Applying rule PAT-REFINE in the core language finishes the case.

- $m_p = r$. In this case similarly by Lemma 4.2 and Lemma 4.4 we get that

$$\Gamma \vdash p' : \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta_c\}$$

in the core language and there exists a mapping m such that $m \cdot \theta = \theta_c \cdot m$ with $dom(m)$ fresh variables. Then by induction hypothesis we have that $\theta(\Gamma, \Delta) \vdash t' : \theta(\tau_t)$ (selective application of θ to the rigid parts of Γ, Δ). By the substitution lemma for the core language, and since $dom(m)$ does not appear in Δ, $m\theta(\Gamma, \Delta) \vdash t' : m\theta(\tau_t)$, therefore $\theta_c(\Gamma, \Delta) \vdash t' : \theta_c(\tau_s)$. Moreover θ_c is an idempotent most general unifier, therefore $\theta_c[\Gamma, \Delta] \vdash \theta_c(t') : \theta_c(\tau_s)$. Applying rule PAT-REFINE finishes the case.

□

4.1 Target to source translation

The target language presented in this report is based on System F, therefore it allows pattern variables to have polymorphic types. Additionally, it uses *preunifiers*—and the main paper exhibits that there may be programs typeable with a pre-unifier refinement but not with a most general unifier refinement (and an *fmgu* refinement).

If (i) the core language used fresh most general unifiers, (ii) did not allow branches that were statically detected to be inaccessible (as our source language does), and (ii) only allowed second-class polymorphism then we could give a systematic translation from target programs to source programs where we put as many annotations as is necessary to ensure *rigidity* in every possible pattern match. The translation is a straightforward type directed translation and we ommit it from this report. The fact that our rigidity analysis is rather weak implies that the translated programs would probably be heavily annotated—but this is irrelevant to demonstrate that the source language is as expressive as the core language.

$$\boxed{\Gamma \vdash p \to t :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t \rightsquigarrow p' \to t'}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{(ftv(\Gamma,\tau_t,\tau_p)),\overline{e}\times\overline{c})} p :^{m_p} \tau_p \blacktriangleright \{\overline{e}\times\overline{c},\Delta,\theta\} \rightsquigarrow p' \\ ftv(\Gamma,\tau_p,\tau_t)\#\overline{e}\times\overline{c} \quad bindings(\Delta) \subseteq \overline{e}\times\overline{c} \\ \theta(\Gamma\cup\Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t) \rightsquigarrow t'\end{array}}{\Gamma \vdash p \to t :^{\langle m_p, m_t\rangle} \tau_p \to \tau_t \rightsquigarrow p' \to t'}\ \textsc{PAT}$$

$$\boxed{\Gamma, \{\overline{e}\times\overline{c}, \Delta_1, \theta\} \vdash_X q :^m \tau \blacktriangleright \{\overline{e}\times\overline{c},\Delta_2,\theta\}}$$

$$\frac{x \notin dom(\Delta)}{\Gamma,\{\overline{e}\times\overline{c},\Delta,\theta\} \vdash_X x :^m \tau \blacktriangleright \{\overline{e}\times\overline{c},\Delta, x :^m \tau,\theta\} \rightsquigarrow x_\tau}\ \textsc{PVAR}$$

$$\frac{\begin{array}{c}\overline{\mathtt{b}} = ftv(\mathtt{tau}) - ftv(\Gamma,\Delta) \\ [\![\mathtt{tau}]\!]_{\Gamma,\Delta_1,\overline{\mathtt{b}\hookrightarrow b}} = \tau_s \\ \overline{b}\ \text{distinct} \quad \overline{b}\#dom(\theta) \quad \overline{b}\#bindings(\Delta_1) \\ \theta(\tau_s) = \theta^m(\tau) \\ \Gamma,\{\overline{e}\times\overline{c},(\Delta_1,\overline{\mathtt{b}\hookrightarrow b}),\theta\} \vdash_X q :^r \tau_s \blacktriangleright \{\overline{e}\times\overline{c},\Delta_2,\theta\} \rightsquigarrow p'\end{array}}{\Gamma,\{\overline{e}\times\overline{c},\Delta_1,\theta\} \vdash_X (q :: \mathtt{tau}) :^m \tau \blacktriangleright \{\overline{e}\times\overline{c},\Delta_2,\theta\} \rightsquigarrow p'}\ \textsc{PANN}$$

$$\boxed{\Gamma,\{\overline{e}\times\overline{c},\Delta_1,\theta\} \vdash_X^{fold} \overline{q_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e}\times\overline{c},\Delta_n,\theta\} \rightsquigarrow \overline{p_i'}}$$

$$\frac{}{\Gamma,\{\overline{e}\times\overline{c},\Delta,\theta\} \vdash_X^{fold} \cdot \blacktriangleright \{\overline{e}\times\overline{c},\Delta,\theta\} \rightsquigarrow \cdot}\ \textsc{F-BASE}$$

$$\frac{\begin{array}{c}\Gamma,\{\overline{e}\times\overline{c},\Delta_1,\theta\} \vdash_X q :^m \tau \blacktriangleright \{\overline{e}\times\overline{c},\Delta_2,\theta\} \rightsquigarrow p' \\ \Gamma,\{\overline{e}\times\overline{c},\Delta_2,\theta\} \vdash_X^{fold} \overline{q_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e}\times\overline{c},\Delta_n,\theta\} \rightsquigarrow \overline{p_i'}\end{array}}{\Gamma,\{\overline{e}\times\overline{c},\Delta_1,\theta\} \vdash_X^{fold} (q :^m \tau), \overline{q_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e}\times\overline{c},\Delta_n,\theta\} \rightsquigarrow p', \overline{p_i'}}\ \textsc{F-REC}$$

$$\boxed{\Gamma \vdash_X p :^m \tau \blacktriangleright K}$$

$$\frac{\begin{array}{c}C :^r \forall\overline{b}.\overline{\tau}_1 \to T\,\overline{\tau}_2 \in \Gamma \quad \overline{b}_c = \overline{b} \cap ftv(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \\ \psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \quad m_i = \begin{cases} r & ftv(\tau_{1i})\#\overline{b}_c \\ w & \text{otherwise}\end{cases} \\ \Gamma,\{\overline{b}_e\times\overline{b}_c,\cdot,\emptyset\} \vdash_X^{fold} \overline{q_i :^{m_i} \psi(\tau_{1i})} \blacktriangleright \{\overline{b}_e\times\overline{b}_c,\Delta_n,\emptyset\} \rightsquigarrow \overline{p_i'}\end{array}}{\Gamma \vdash_X C\,\overline{q} :^w T\,\overline{\tau}_3 \blacktriangleright \{\overline{b}_e\times\overline{b}_c,\Delta_n,\emptyset\} \rightsquigarrow C\,\overline{b}\,\overline{p_i'}}\ \textsc{PCON-W}$$

$$\frac{\begin{array}{c}C :^r \forall\overline{b}.\overline{\tau}_1 \to T\,\overline{\tau}_2 \in \Gamma \quad \tau = T\,\overline{\tau}_3 \\ \psi \in fmgu(\overline{\tau}_3 \doteq \overline{\tau}_2) \quad (ftv(\psi) - ftv(\overline{\tau}_3,\overline{\tau}_2))\#\overline{b}, X, \Gamma \\ \Gamma,\{\overline{b}\times\emptyset,\cdot,\psi\} \vdash_X^{fold} \overline{q_i :^r \tau_{1i}} \blacktriangleright \{\overline{b}\times\emptyset,\Delta_n,\psi\} \rightsquigarrow \overline{p_i'}\end{array}}{\Gamma \vdash_X C\,\overline{q} :^r \tau \blacktriangleright \{\overline{b}\times\emptyset,\Delta_n,\psi\} \rightsquigarrow C\,\overline{b}\,\overline{p_i'}}\ \textsc{PCON-R}$$

$$\frac{\Gamma,\{\emptyset,\cdot,\varepsilon\} \vdash_X q :^m \tau \blacktriangleright \{\emptyset,\Delta,\emptyset\} \rightsquigarrow p'}{\Gamma \vdash_X q :^m \tau \blacktriangleright \{\emptyset,\Delta,\emptyset\} \rightsquigarrow p'}\ \textsc{PQ}$$

Figure 9: Source language pattern typing (shallow)

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

$$\frac{\nu : \sigma \in \Gamma \quad \vdash^{inst} \sigma \leq \tau}{\Gamma \vdash \nu : \tau}\ \text{HMVAR} \qquad \frac{\Gamma \vdash t : \tau_1 \to \tau_2 \quad \Gamma \vdash u : \tau_1}{\Gamma \vdash t\ u : \tau_2}\ \text{HMAPP}$$

$$\frac{\begin{array}{c}\Gamma \vdash u : \tau_x \quad \overline{a} = ftv(\tau) - ftv(\Gamma) \\ \Gamma, x{:}\forall\overline{a}.\tau_x \vdash t : \tau\end{array}}{\Gamma \vdash (\texttt{let}\ x\texttt{=}u\ \texttt{in}\ t) : \tau}\ \text{HMLET} \qquad \frac{\begin{array}{c}[\![\texttt{sig}]\!]_\Gamma = \forall\overline{a}.\tau_s \\ \Gamma \vdash t : \tau_s \quad \overline{a}\#ftv(\Gamma) \quad \vdash^{inst} \forall\overline{a}.\tau_s \leq \tau\end{array}}{\Gamma \vdash (t :: \texttt{sig}) : \tau}\ \text{HMSIG}$$

$$\frac{\Gamma \vdash x \to t : \tau_1 \to \tau_2}{\Gamma \vdash \backslash x.t : \tau_1 \to \tau_2}\ \text{HMABS} \qquad \frac{\Gamma \vdash u : \tau_1 \quad \Gamma \vdash p_i \to t_i : \tau_1 \to \tau_2}{\Gamma \vdash (\texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t}) : \tau_2}\ \text{HMCASE}$$

$$\boxed{\Gamma \vdash^{HM} p \to t : \tau_1 \to \tau_2}$$

$$\frac{\begin{array}{c}\Gamma \vdash p : \tau_1 \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta\} \\ \Gamma, \Delta \vdash t : \tau_2 \quad ftv(\Gamma, \tau_1, \tau_2)\#\overline{e} \times \overline{c}\end{array}}{\Gamma \vdash p \to t : \tau_1 \to \tau_2}\ \text{HMPAT}$$

$$\boxed{\Gamma, \{\overline{e} \times \overline{c},\ \Delta_1\} \vdash^{HM} q : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_2\}}$$

$$\frac{x \notin dom(\Delta)}{\Gamma, \{\overline{e} \times \overline{c},\ \Delta\} \vdash x : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta, x{:}\tau\}}\ \text{HMPVAR}$$

$$\boxed{\Gamma \vdash^{HM} p : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta\}}$$

$$\frac{\begin{array}{c}C : \forall\overline{b}.\overline{\tau_1} \to T\ \overline{b}_c \in \Gamma \quad \psi = [\overline{b_c \mapsto \tau_3}] \\ \Gamma, \{(\overline{b} - \overline{b}_c) \times \overline{b}_c,\ \cdot\} \vdash^{fold} \overline{x : \psi(\tau_1)} \blacktriangleright \{(\overline{b} - \overline{b}_c) \times \overline{b}_c,\ \Delta_n\}\end{array}}{\Gamma \vdash C\ \overline{x} : T\ \overline{\tau}_3 \blacktriangleright \{(\overline{b} - \overline{b}_c) \times \overline{b}_c,\ \Delta_n\}}\ \text{HMPCON} \qquad \frac{\Gamma, \{\emptyset,\ \cdot\} \vdash x : \tau \blacktriangleright \{\emptyset,\ \Delta\}}{\Gamma \vdash x : \tau \blacktriangleright \{\emptyset,\ \Delta\}}\ \text{HMPQ}$$

$$\boxed{\Gamma, \{\overline{e} \times \overline{c},\ \Delta_1\} \vdash^{fold} \overline{q : \tau} \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_n\}}$$

$$\frac{}{\Gamma, \{\overline{e} \times \overline{c},\ \Delta\} \vdash^{fold} \cdot \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta\}}\ \text{HMF-BASE} \qquad \frac{\begin{array}{c}\Gamma, \{\overline{e} \times \overline{c},\ \Delta_1\} \vdash q : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_2\} \\ \Gamma, \{\overline{e} \times \overline{c},\ \Delta_2\} \vdash^{fold} \overline{q : \tau} \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_n\}\end{array}}{\Gamma, \{\overline{e} \times \overline{c},\ \Delta_1\} \vdash^{fold} (q : \tau), \overline{q : \tau} \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_n\}}\ \text{HMF-REC}$$

Figure 10: HM with (shallow) pattern matching

5 Conservative extension of HM

In Figure 10 we give a vanilla Hindley-Milner system with algebraic datatypes (not generalised, but possibly with existential variables), closed type annotations on terms (but not on patterns), and pattern matching. In this section we prove that our system is a conservative extension of the vanilla Hindley-Milner system given in Figure 10. We note for now that pattern matching is shallow.

We assume in the rest of this section that datatypes are not generalized, i.e. constructors are of the form

$\forall \overline{b}.\overline{\tau}_1 \to T\ \overline{b}_c$ where $\overline{b}_c$ are distinct and $\overline{b} - \overline{b}_c$ are existential variables.

Moreover we assume that there are no type annotations in patterns, and all rest type annotations are closed. These conditions are not of great significance, and we impose them only to make the proofs shorter—the spirit of the proofs and the theorems below does not change if we lift these syntactic restrictions and extend accordingly the HM figure.

Definition 5.1. *For a given environment Δ we use $\lfloor \Delta \rfloor$ to denote the same environment where we ignore the modifiers and the lexical variable bindings.*

5.1 Extension of HM

Lemma 5.2 (q-pattern checking extension). *If $\Gamma, \{\overline{e} \times \overline{c},\ \lfloor \theta(\Delta_1) \rfloor\} \vdash^{HM} q : \theta^m(\tau) \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_2\}$ and $dom(\theta) \subseteq \overline{e}$ then $\Gamma, \{\overline{e}' \times \overline{c}', \Delta_1, \theta\} \vdash q :^m \tau \blacktriangleright \{\overline{e}' \times \overline{c}', \Delta_2', \theta\}$ such that $\lfloor \theta(\Delta_2') \rfloor = \Delta_2$.*

If $\Gamma, \{\overline{e} \times \overline{c},\ \lfloor \theta(\Delta_1) \rfloor\} \vdash^{fold} \overline{q : \theta^m(\tau)} \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta_n\}$ and $dom(\theta) \subseteq \overline{e}$ then $\Gamma, \{\overline{e}' \times \overline{c}', \Delta_1, \theta\} \vdash^{fold} \overline{q :^m \tau} \blacktriangleright \{\overline{e}' \times \overline{c}', \Delta_n', \theta\}$ such that $\lfloor \theta(\Delta_n') \rfloor = \Delta_n$.

Proof. The two claims are proved simultaneously by induction on the derivations. For the second part the case of HMF-BASE is trivial, and the case of HMF-REC follows by induction hypothesis for both parts and rule F-REC. For the first part we have to consider only rule HMPVAR and the result follows easily using PVAR. ☐

Lemma 5.3 (p-pattern checking extension). *If $\Gamma \vdash^{HM} p : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta\}$ then $\Gamma \vdash p :^w \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta', \varepsilon\}$ such that $\lfloor \Delta' \rfloor = \Delta$. Additionally $\Gamma \vdash p :^r \tau \blacktriangleright \{\overline{ec} \times \emptyset, \Delta', \theta\}$ such that $\lfloor \theta(\Delta') \rfloor = \Delta$ and $dom(\theta) \subseteq \overline{c}$.*

Proof. By induction on the HM pattern checking derivation. We have to consider two cases.

- Case HMPCON. In this case we have that

$$\Gamma \vdash C\ \overline{x} : T\ \overline{\tau}_3 \blacktriangleright \{(\overline{b} - \overline{b}_c) \times \overline{b}_c,\ \Delta_n\}$$

given that

$$C : \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{b}_c \in \Gamma \quad \psi = [\overline{b_c \mapsto \tau_3}] \tag{1}$$

$$\Gamma, \{(\overline{b} - \overline{b}_c) \times \overline{b}_c,\ \cdot\} \vdash^{fold} \overline{x : \psi(\tau_1)} \blacktriangleright \{(\overline{b} - \overline{b}_c) \times \overline{b}_c,\ \Delta_n\} \tag{2}$$

For the first part we wish to apply rule PCON-W. From Lemma 5.2 and taking $\theta = \varepsilon$, $m_i = w$ if $ftv(\tau_{1i}) \cap \overline{b}_c \neq \emptyset$ and $m_i = r$ otherwise we get from equation 2 that

$$\Gamma, \{(\overline{b} - \overline{b}_c) \times \overline{b}_c, \cdot, \varepsilon\} \vdash^{fold} \overline{x :^m \psi(\tau_1)} \blacktriangleright \{(\overline{b} - \overline{b}_c) \times \overline{b}_c, \Delta_n', \varepsilon\} \tag{3}$$

such that $\lfloor \Delta_n' \rfloor = \Delta_n$. Applying rule PCON-W finishes the case.

For the second part we wish to apply rule PCON-R. From 1 we know that $\psi \in fmgu(\overline{b}_c \doteq \overline{\tau}_3)$. Taking $\theta = \psi$, $m_i = r$ we can apply Lemma 5.2 on equation 2 to get that

$$\Gamma, \{\overline{b} \times \emptyset, \cdot, \psi\} \vdash^{fold} \overline{x :^r \tau_1} \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n', \varepsilon\} \tag{4}$$

and $\lfloor \psi(\Delta_n') \rfloor = \Delta_n$ as required. Applying rule PCON-R finishes the case.

- Case HMPQ. In this case we have that $\Gamma \vdash x : \tau \blacktriangleright \{\emptyset,\ \Delta\}$ given that $\Gamma, \{\emptyset,\ \cdot\} \vdash x : \tau \blacktriangleright \{\emptyset,\ \Delta\}$. By Lemma 5.2 and taking $\theta = \varepsilon$ we get that $\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash x :^m \tau \blacktriangleright \{\emptyset,\ \Delta'\}\varepsilon$ such that $\lfloor \Delta' \rfloor = \Delta$ for either $m = w$ or r.

□

Theorem 5.4 (Extension of HM). *If* $\lfloor \Gamma \rfloor \vdash^{HM} t : \tau$ *then*

- *for every* m *it is* $\Gamma \vdash t :^m \tau$*, and*
- *there exists an* m *such that* $\Gamma \vdash t : \tau \rceil^m$*.*

Proof. We prove the two claims simultaneously by induction on the height of the *HM* derivation. All cases are straightforward except for the HMCASE case. In this case we have that

$$\lfloor \Gamma \rfloor \vdash (\texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t}) : \tau_2$$

given that

$$\lfloor \Gamma \rfloor \vdash u : \tau_1 \tag{1}$$
$$\lfloor \Gamma \rfloor \vdash p_i \to t_i : \tau_1 \to \tau_2 \tag{2}$$

By induction hypothesis $\Gamma \vdash u : \tau \rceil^{m_p}$ for some m_p. We wish to show that

$$\Gamma \vdash (\texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t}) :^{m_t} \tau_2$$

for any m_t. Take any $p \to t \in \overline{p\texttt{->}t}$. By inversion (rule HMPAT) it must be that

$$\lfloor \Gamma \rfloor \vdash p \to t : \tau_1 \to \tau_2 \tag{3}$$
$$\lfloor \Gamma \rfloor \vdash p : \tau_1 \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta\} \tag{4}$$
$$\lfloor \Gamma \rfloor, \Delta \vdash t : \tau_2 \quad \mathit{ftv}(\Gamma, \tau_1, \tau_2) \# \overline{e} \times \overline{c} \tag{5}$$

Let us consider cases on m_p.

- $m_p = w$. In this case, by Lemma 5.3 we get that

  $$\Gamma \vdash p :^w \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta', \varepsilon\}$$

  such that $\lfloor \Delta' \rfloor = \Delta$. By induction hypothesis then we also have that $\Gamma, \Delta' \vdash t :^{m_t} \tau_2$ or $\varepsilon(\Gamma, \Delta') \vdash t :^{m_t} \varepsilon^{m_t}\tau_2$. Applying rule PAT finishes the case.

- $m_p = r$. In this case, by Lemma 5.3 we get that

  $$\Gamma \vdash p :^r \tau \blacktriangleright \{\overline{ec} \times \emptyset, \Delta', \theta\}$$

  such that $\lfloor \theta(\Delta') \rfloor = \Delta$. By induction hypothesis then $\Gamma, \theta(\Delta') \vdash t :^{m_t} \tau_2$ but since $\mathit{dom}(\theta) \subseteq \overline{c}$ this can be rewritten as $\theta(\Gamma, \Delta') \vdash t :^{m_t} \theta^{m_t}(\tau_2)$. Applying rule PAT finishes the case.

Hence for any m_t it is

$$\Gamma \vdash (\texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t}) :^{m_t} \tau_2$$

and therefore by SCR-OTHER we get

$$\Gamma \vdash (\texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t}) : \tau_2 \rceil^w$$

□

5.2 Conservativity over HM

Lemma 5.5 (Vars of returned HM environments). *If* $\Gamma \vdash^{HM} p : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \Delta\}$ *and* $ftv(\tau)\#\overline{c}$ *then* $ftv(\Delta)\#\overline{c}$.

Proof. Straightforward induction, observing that all the $\overline{c}$ variables are substituted away. □

Definition 5.6. *We write* $X = Y \cup\neg\ Z$ *to denote that if* $a \in X$ *then* $a \in Y$ *or* $a \notin Z$.
Lemma 5.7 (q-pattern conservativity).

1. *If* $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash^{fold}_{X} \overline{q :^{m} \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\}$ *and* $\overline{e} = \overline{e}_1\overline{c}_1$ *and there exists a mapping* m *with* $dom(m) = \overline{c}_1 \cup\neg\ (X, \Gamma, \Delta_1, \overline{\tau}, \overline{ec})$, *and* $\lfloor m\theta(\Delta_1)\rfloor = \Delta'_1$ *then* $\lfloor\Gamma\rfloor, \{\overline{e}_1 \times \overline{c}_1\overline{c},\ \Delta'_1\} \vdash^{fold} \overline{q : (m\theta)^{m}\tau} \blacktriangleright \{\overline{e} \times \overline{c}_1\overline{c},\ \Delta'_n\}$ *such that* $\lfloor m\theta(\Delta_n)\rfloor = \Delta'_n$. *Moreover* $dom(m) = \overline{c}_1 \cup\neg\ (X, \Gamma, \Delta_n, \overline{\tau}, \overline{ec})$.

2. *If* $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X q :^{m} \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\}$ *and* $\overline{e} = \overline{e}_1\overline{c}_1$ *and there exists a mapping* m *with* $dom(m) = \overline{c}_1 \cup\neg\ (X, \Gamma, \Delta_1, \tau, \overline{ec})$, *and* $\lfloor m\theta(\Delta_1)\rfloor = \Delta'_1$ *then* $\lfloor\Gamma\rfloor, \{\overline{e}_1 \times \overline{c}_1\overline{c},\ \Delta'_1\} \vdash q : (m\theta)^{m}\tau \blacktriangleright \{\overline{e} \times \overline{c}_1\overline{c},\ \Delta'_2\}$ *such that* $\lfloor m\theta(\Delta_2)\rfloor = \Delta'_2$. *Moreover* $dom(m) = \overline{c}_1 \cup\neg\ (X, \Gamma, \Delta_2, \tau, \overline{ec})$

Proof. Easy induction. □

Lemma 5.8 (p-pattern conservativity).

1. *If* $\Gamma \vdash_X p :^{w} \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \varepsilon\}$ *then* $\lfloor\Gamma\rfloor \vdash^{HM} p : \tau \blacktriangleright \{\overline{e} \times \overline{c},\ \lfloor\Delta\rfloor\}$.

2. *If* $\Gamma \vdash_X p :^{r} \tau \blacktriangleright \{\overline{e} \times \emptyset, \Delta, \theta\}$ *then* $\overline{e} = \overline{e}_1 \times \overline{c}_1$ *such that there exists a mapping* m *such that* $dom(m) = \overline{c}_1 \cup\neg\ (X, \Gamma, \Delta_1, \tau, \overline{e})$, *and such that* $\lfloor\Gamma\rfloor \vdash^{HM} p : \tau \blacktriangleright \{\overline{e}_1 \times \overline{c}_1,\ \Delta'\}$ *and* $m\lfloor\theta(\Delta)\rfloor = \Delta'$.

Proof. We have to consider three cases.

- Case PCON-W. We have that

$$\Gamma \vdash_X C\ \overline{q} :^{w} T\ \overline{\tau}_3 \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\}$$

  given that

$$\begin{aligned}
C :^{r} \forall\overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \qquad & (1)\\
\overline{b}_c = \overline{b} \cap ftv(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \qquad & (2)\\
\psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \qquad & (3)\\
m_i = \begin{cases} r & ftv(\tau_{1i})\#\overline{b}_c \\ w & \text{otherwise} \end{cases} \qquad & (4)\\
\Gamma, \{\overline{b}_e \times \overline{b}_c, \cdot, \emptyset\} \vdash^{fold}_{X} \overline{q_i :^{m_i} \psi(\tau_{1i})} & \\
\blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\} \qquad & (5)
\end{aligned}$$

  We wish to apply Lemma 5.7 (first part). Take $\overline{c} = \overline{b}_c$, $\overline{e} = \overline{b}_e$, $\overline{e}_1 = \overline{b}_e$, $\overline{c}_1 = \emptyset$, $m = \varepsilon$. Then we get that

$$\lfloor\Gamma\rfloor, \{\overline{b}_e \times \overline{b}_c,\ \cdot\} \vdash^{fold} \overline{q_i : \psi(\tau_{1i})} \blacktriangleright \{\overline{b}_e \times \overline{b}_c,\ \Delta'_n\}$$

  in the HM system and moreover $\lfloor\Delta_n\rfloor = \Delta'_n$. Applying rule HMPCON finishes the case.

- Case PCON-R. We have that

$$\Gamma \vdash_X C\ \overline{q} :^r \tau \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\}$$

given that

$$C :^r \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \tag{6}$$

$$\tau = T\ \overline{\tau}_3 \quad \psi \in \mathit{fmgu}(\overline{\tau}_3 \doteq \overline{\tau}_2) \tag{7}$$

$$(\mathit{ftv}(\psi) - \mathit{ftv}(\overline{\tau}_3, \overline{\tau}_2))\#\overline{b}, X, \Gamma \tag{8}$$

$$\Gamma, \{\overline{b} \times \emptyset, \cdot, \psi\} \vdash^{\mathit{fold}}_X \overline{q_i :^r \tau_{1i}} \blacktriangleright \{\overline{b} \times \emptyset, \Delta_n, \psi\} \tag{9}$$

In this case we will actually show that $\lfloor m\theta(\Delta)\rfloor = \Delta'$. Because we know that that if $x :^w \tau_x \in \Delta$ then $x : \tau_x \in \Delta'$ and by Lemma 5.5 $\tau_x \# \overline{c}$. It follows that $m\lfloor\theta(\Delta)\rfloor = \Delta'$ as is required by the second part. We know that in this case $\overline{\tau}_2 = \overline{b}_c \subseteq \overline{b}$ and moreover since ψ exists, $\overline{b}_c$ and $\overline{\tau}_3$ are unifiable; and additionally $[\overline{b_c \mapsto \tau_3}]$ is a biased and minimal *fmgu* as well. By Lemma 2.19 there exists a mapping m such that $m \cdot \psi$ restricted to $\overline{b}_c$ and the variables of $\overline{\tau}_3$ is biased and minimal. By Lemma 2.18 we know that such a biased and minimal *fmgu* is unique up to renaming of fresh variables. Since $[\overline{b_c \mapsto \tau_3}]$ is biased and minimal it must be that

$$[\overline{b_c \mapsto \tau_3}] = m \cdot \psi|_{\overline{b}_c, \mathit{ftv}(\overline{\tau}_3)}$$

We wish to apply Lemma 5.7 (first part). Take $\overline{e} = \overline{b}$, $\overline{c}_1 = \overline{b}_c$, $\overline{e}_1 = \overline{b} - \overline{b}_c$, $\theta = \psi$ and the above mapping as m. Equation 9 then gives

$$\lfloor\Gamma\rfloor, \{\overline{b} - \overline{b}_c \times \overline{b}_c,\ \cdot\} \vdash^{\mathit{fold}} \overline{q_i : (m\psi)^r(\tau_{1i})} \blacktriangleright \{\overline{b} - \overline{b}_c \times \overline{b}_c,\ \Delta'_n\} \tag{10}$$

such that $\lfloor m\psi(\Delta_n)\rfloor = \Delta'_n$. Applying rule HMPCON finishes the case, by observing that on Δ_n and on $\overline{\tau}_1$, $[\overline{b_c \mapsto \tau_3}]$ and $m \cdot \psi$ give the same results.

- Case PQ. We have that

$$\Gamma \vdash_X q :^m \tau \blacktriangleright \{\emptyset, \Delta, \emptyset\}$$

given that

$$\Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash_X q :^m \tau \blacktriangleright \{\emptyset, \Delta, \emptyset\}$$

The case immediately follows from the second part of Lemma 5.7.

□

Theorem 5.9 (Conservativity over HM). *The following are true*

- *If* $\Gamma \vdash t :^m \tau$ *then* $\lfloor\Gamma\rfloor \vdash^{HM} t : \tau$.
- *If* $\Gamma \vdash p \to t :^{\langle m_p, m_t\rangle} \tau_p \to \tau_t$ *then* $\lfloor\Gamma\rfloor \vdash^{HM} p \to t : \tau_p \to \tau_t$.
- *If* $\Gamma \vdash t : \tau \upharpoonright^m$ *then* $\lfloor\Gamma\rfloor \vdash^{HM} t : \tau$.

Proof. We prove the three claims simultaneously by induction on the height of the derivations. For each part the induction hypothesis asserts both for derivations of smaller height. The only interesting part is the second. We have that:

$$\Gamma \vdash p \to t :^{\langle m_p, m_t\rangle} \tau_p \to \tau_t$$

given that

$$\Gamma \vdash_{(ftv(\Gamma,\tau_t,\tau_p),\overline{e}\times\overline{c})} p :^{m_p} \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \tag{1}$$

$$ftv(\Gamma, \tau_p, \tau_t)\#\overline{e} \times \overline{c} \quad bindings(\Delta) \subseteq \overline{e} \times \overline{c} \tag{2}$$

$$\theta(\Gamma \cup \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t) \tag{3}$$

First of all, if $m_p = w$ then $\theta = \varepsilon$ as well and by Lemma 5.8 we get that $\lfloor\Gamma\rfloor \vdash p : \tau_p \blacktriangleright \{\overline{e} \times \overline{c},\ \lfloor\Delta\rfloor\}$. Moreover, by induction hypothesis we have that $\lfloor\Gamma, \Delta\rfloor \vdash t : \tau_t$ and applying rule HMPAT finishes this case.

On the other hand, if $m_p = r$ then also $\overline{c} = \emptyset$ and by Lemma 5.8 we get that $\overline{e} = \overline{e}_1 \times \overline{c}_1$ and there exists a mapping m such that if $a \in dom(m)$ then either $a \in \overline{c}_1$ or $a\#\tau_t, \tau_p, \Gamma$. Moreover

$$\lfloor\Gamma\rfloor \vdash^{HM} p : \tau_p \blacktriangleright \{\overline{e}_1 \times \overline{c}_1,\ \Delta'\}$$

suych that $m\lfloor\theta(\Delta)\rfloor = \Delta'$. By induction hypothesis we also get that:

$$\lfloor\theta(\Gamma, \Delta)\rfloor \vdash t : \theta^{m_t}(\tau_t)$$

By the type substitution lemma for HM (we assume it holds—its proof can be found elsewhere) we then get that $m\lfloor\theta(\Gamma, \Delta)\rfloor \vdash t : m(\theta^{m_t}(\tau_t))$ which implies $\lfloor m[\theta(\Gamma)]\rfloor, m\lfloor\theta(\Delta)\rfloor \vdash t : \tau_t$ becasue $m\theta^{m_t}(\tau_t) = \tau_t$. Equivalently

$$\lfloor\Gamma\rfloor, \Delta' \vdash t : \tau_t$$

since also $m[\theta(\Gamma)] = \Gamma$. Applying rule HMPAT finishes the case. □

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t :^m \tau \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\nu :^{m_x} \sigma \in \Gamma \quad (\mathcal{A}_0, \delta_0) \succ \vdash^{inst} \sigma \le \tau \succ (\mathcal{A}_1, \delta_1)}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash \nu :^m \tau \succ (\mathcal{A}_1, \delta_1)}\ \text{AVAR}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t :^w \beta \to \tau_2 \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \Gamma \vdash u :^w \beta \succ (\mathcal{A}_2, \delta_2)\end{array}}{(\mathcal{A}_0 \beta, \delta_0) \succ \Gamma \vdash t\ u :^m \tau_2 \succ (\mathcal{A}_2, \delta_2)}\ \text{AAPP}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash^{poly} u : \sigma \upharpoonleft m_x \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \Gamma, x :^{m_x} \sigma \vdash t :^m \tau \succ (\mathcal{A}_2, \delta_2)\end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash (\texttt{let}\ x\texttt{=}u\ \texttt{in}\ t) :^m \tau \succ (\mathcal{A}_2, \delta_2)}\ \text{ALET-W}$$

$$\frac{\begin{array}{c}\mathcal{A}_0 \succ \Gamma \vdash \texttt{sig} \Rightarrow \sigma ; \Delta \succ \mathcal{A}_1 \\ (\mathcal{A}_0, \delta_0) \succ \Gamma, \Delta \vdash^{poly} u :^r \sigma \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \Gamma, x :^r \sigma \vdash t :^m \tau \succ (\mathcal{A}_2, \delta_2)\end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash (\texttt{let}\ x\texttt{::sig=}u\ \texttt{in}\ t) :^m \tau \succ (\mathcal{A}_2, \delta_2)}\ \text{ALET-R}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \tau \doteq^{\to} \tau_1 \to \tau_2 \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \Gamma \vdash p \to t :^{\langle m, m \rangle} \tau_1 \to \tau_2 \succ (\mathcal{A}_2, \delta_2)\end{array}}{(\mathcal{A}_0 \alpha \beta, \delta_0) \succ \Gamma \vdash \backslash p.t :^m \tau \succ (\mathcal{A}_2, \delta_2)}\ \text{APAT}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash u : \tau_1 \upharpoonleft^{m_1} \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_i, \delta_i) \succ \Gamma \vdash p_i \to t_i :^{\langle m_1, m_2 \rangle} \tau_1 \to \tau_2 \succ (\mathcal{A}_{i+1}, \delta_{i+1})\end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash \texttt{case}\ u\ \texttt{of}\ \overline{p\texttt{->}t} :^{m_2} \tau_2 \succ (\mathcal{A}_n, \delta_n)}\ \text{ACASE}$$

$$\frac{\begin{array}{c}\mathcal{A}_0 \succ \Gamma \vdash \texttt{sig} \Rightarrow \sigma ; \Delta \succ \mathcal{A}_1 \\ (\mathcal{A}_1, \delta_0) \succ \Gamma, \Delta \vdash^{poly} t :^r \sigma \succ (\mathcal{A}_2, \delta_1) \\ (\mathcal{A}_2, \delta_1) \succ \vdash^{inst} \sigma \le \tau \succ (\mathcal{A}_3, \delta_2)\end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash (t :: \texttt{sig}) :^m \tau \succ (\mathcal{A}_3, \delta_2)}\ \text{ASIG}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash^{poly} t :^m \sigma \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t :^m [\overline{a \mapsto b}]\tau \succ (\mathcal{A}_1, \delta_1) \\ \overline{b} \# ftv(\delta_1 \Gamma, \delta_1(\forall \overline{a}.\tau))\end{array}}{(\mathcal{A}_0 \overline{b}, \delta_0) \succ \Gamma \vdash^{poly} t :^m \forall \overline{a}.\tau \succ (\mathcal{A}_1, \delta_1)}\ \text{GEN}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \vdash^{inst} \sigma \le \tau \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\delta_0 \succ [\overline{a \mapsto \beta}]\tau_1 \doteq \tau_2 \succ \delta_1}{(\mathcal{A}_0 \overline{\beta}, \delta_0) \succ \vdash^{inst} \forall \overline{a}.\tau_1 \le \tau_2 \succ (\mathcal{A}_0, \delta_1)}\ \text{INST}$$

Figure 11: Algorithmic typing relation

6 Type inference

We give below some useful for type inference properties of the source language. Introducing some notation, for a given context Γ we let $ftv(\Gamma|_r)$ mean the free variables of the rigid parts of the context only. For a given type we define $ftv(\tau|_r) = ftv(\tau)$ and $ftv(\tau|_w) = \emptyset$.

Lemma 6.1 (Rigid modifier inference). *If $\Gamma \vdash t : \tau \upharpoonleft^r$ then $ftv(\tau) \subseteq ftv(\Gamma)$. Moreover $ftv(\tau) \subseteq ftv(\Gamma|_r)$. If $\Gamma \vdash^{poly} t : \sigma \upharpoonleft^r$ then $ftv(\tau) \subseteq ftv(\Gamma)$.*

Proof. We prove each claim by induction on the height of the corresponding derivation. For the first part,

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \tau \doteq^{\to} \tau_1 \to \tau_2 \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{}{(\mathcal{A}_0, \delta_0) \succ \tau_1 \to \tau_2 \doteq^{\to} \tau_1 \to \tau_2 \succ (\mathcal{A}_0, \delta_0)}\ \text{AU1} \qquad \frac{\delta_0 \succ \tau \doteq \beta \to \gamma \succ \delta_1}{(\mathcal{A}_0 \beta\gamma, \delta_0) \succ \tau \doteq^{\to} \beta \to \gamma \succ (\mathcal{A}_0, \delta_1)}\ \text{AU2}$$

Figure 12: Arrow unification

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t : \tau \upharpoonleft^{m} \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\begin{array}{c} \nu :^{r} \forall \overline{a}\,.\,\overline{\tau} \to \tau \in \Gamma \\ (\mathcal{A}_{i-1}, \delta_{i-1}) \succ \Gamma \vdash u_i : \tau'_i \upharpoonleft^{m_i} \succ (\mathcal{A}_i, \delta_i) \\ \delta_n \succ \overline{\tau'} \doteq [\overline{a \mapsto \beta}]\overline{\tau} \succ \delta'_n \\ \overline{a}_r = \{a \in \overline{a} \mid \exists \tau_i\,.\, a \in \mathit{ftv}(\tau_i) \wedge m_i = r\} \\ m = \begin{cases} r & (\mathit{ftv}(\tau) \subseteq \overline{a}_r) \\ w & \text{otherwise} \end{cases} \end{array}}{(\mathcal{A}_0 \overline{\beta}, \delta_0) \succ \Gamma \vdash \nu\ \overline{u} : \delta'_n([\overline{a \mapsto \beta}]\tau) \upharpoonleft^{m} \succ (\mathcal{A}_n, \delta_m)}\ \text{ASCR-APP}$$

$$\frac{\begin{array}{c} \mathcal{A}_0 \succ \Gamma \vdash \texttt{tau} \Rightarrow \tau\,;\emptyset \succ \mathcal{A}_1 \\ (\mathcal{A}_1, \delta_0) \succ \Gamma \vdash t :^{r} \tau \succ (\mathcal{A}_2, \delta_1) \end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash (t :: \texttt{tau}) : \tau \upharpoonleft^{r} \succ (\mathcal{A}_2, \delta_1)}\ \text{ASCR-TSIG}$$

$$\frac{\begin{array}{c} t \neq \nu\ \overline{u} \text{ with } \nu \text{ bound rigidly} \quad t \neq (t_0 :: \texttt{tau}) \\ (\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t :^{w} \beta \succ (\mathcal{A}_1, \delta_1) \end{array}}{(\mathcal{A}_0 \beta, \delta_0) \succ \Gamma \vdash t : \beta \upharpoonleft^{w} \succ (\mathcal{A}_1, \delta_1)}\ \text{ASCR-OTHER}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash^{\mathit{poly}} t : \sigma \upharpoonleft^{m} \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\begin{array}{c} \mathcal{A}_0 \succ \Gamma \vdash \texttt{sig} \Rightarrow \sigma\,;\Delta \succ \mathcal{A}_1 \\ (\mathcal{A}_1, \delta_0) \succ \Gamma, \Delta \vdash^{\mathit{poly}} t :^{r} \sigma \succ (\mathcal{A}_2, \delta_1) \end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash^{\mathit{poly}} (t :: \texttt{sig}) : \sigma \upharpoonleft^{r} \succ (\mathcal{A}_2, \delta_1)}\ \text{ASCR-SSIG}$$

$$\frac{\begin{array}{c} t \neq (t_0 :: \texttt{sig}) \\ (\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t : \tau \upharpoonleft^{m_1} \succ (\mathcal{A}_1, \delta_1) \\ \overline{\alpha} = \mathit{ftv}(\tau) - \mathit{ftv}(\delta_1 \Gamma) \end{array}}{(\mathcal{A}_0 \overline{a}, \delta_0) \succ \Gamma \vdash^{\mathit{poly}} t : \forall \overline{a}\,.\,[\overline{\alpha \mapsto a}]\tau \upharpoonleft^{m_1} \succ (\mathcal{A}_1, \delta_1)}\ \text{ASCR-GEN}$$

Figure 13: Algorithmic scrutinee modifier inference

$$\boxed{\mathcal{A}_0 \succ \Gamma \vdash \texttt{sig} \Rightarrow \sigma\,;\Delta \succ \mathcal{A}_1}$$

$$\frac{\texttt{a} \hookrightarrow \tau \in \Gamma}{\mathcal{A}_0 \succ \Gamma \vdash \texttt{a} \Rightarrow \tau\,;\emptyset \succ \mathcal{A}_0}\ \text{TT-VAR} \qquad \frac{\mathcal{A}_0 \succ \Gamma \vdash \texttt{tau}_1 \Rightarrow \tau_1\,;\emptyset \succ \mathcal{A}_1 \quad \mathcal{A}_1 \succ \Gamma \vdash \texttt{tau}_2 \Rightarrow \tau_2\,;\emptyset \succ \mathcal{A}_2}{\mathcal{A}_0 \succ \Gamma \vdash \texttt{tau}_1 \to \texttt{tau}_2 \Rightarrow (\tau_1 \to \tau_2)\,;\emptyset \succ \mathcal{A}_2}\ \text{TT-FUN}$$

$$\frac{\mathcal{A}_{i-1} \succ \Gamma \vdash \texttt{tau}_i \Rightarrow \tau_i\,;\emptyset \succ \mathcal{A}_i}{\mathcal{A}_0 \succ \Gamma \vdash T\ \overline{\texttt{tau}} \Rightarrow (T\ \overline{\tau})\,;\emptyset \succ \mathcal{A}_n}\ \text{TT-CON} \qquad \frac{\mathcal{A}_0 \succ \Gamma, \overline{\texttt{a} \hookrightarrow a} \vdash \texttt{tau} \Rightarrow \tau\,;\emptyset \succ \mathcal{A}_1}{\mathcal{A}_0\,\overline{a} \succ \Gamma \vdash \forall\overline{\texttt{a}}.\texttt{tau} \Rightarrow (\forall\overline{a}.\tau)\,;\overline{\texttt{a} \hookrightarrow a} \succ \mathcal{A}_1}\ \text{TT-ALL}$$

Figure 14: Algorithmic type translation

if the rule used was SCR-TSIG then we have that

$$\Gamma \vdash (t :: \texttt{tau}) : \tau \upharpoonleft^r \tag{1}$$

given that

$$[\![\texttt{tau}]\!]_\Gamma = \tau \tag{2}$$
$$\Gamma \vdash t :^r \tau \tag{3}$$

From (2) it must be that $\mathit{ftv}(\tau) \subseteq \mathit{ftv}(\Gamma)$ and we are done. The case for SCR-OTHER cannot happen, therefore we only need to consider the case for rule SCR-APP. In this case we have that

$$\Gamma \vdash \nu\ \overline{u} : [\overline{a \mapsto \tau_a}]\tau \upharpoonleft^m \tag{4}$$

given that

$$\nu{:}^r \forall\overline{a}.\overline{\tau} \to \tau \in \Gamma \tag{5}$$
$$\Gamma \vdash u_i : [\overline{a \mapsto \tau_a}]\tau_i \upharpoonleft^{m_i} \tag{6}$$
$$\overline{a}_r = \{a \in \overline{a} \mid \exists \tau_i . a \in \mathit{ftv}(\tau_i) \wedge m_i = r\} \tag{7}$$
$$m = \begin{cases} r & (\mathit{ftv}(\tau) \subseteq \overline{a}_r) \\ w & \text{otherwise} \end{cases} \tag{8}$$

By induction hypothesis, if $\Gamma \vdash u_i : [\overline{a \mapsto \tau_a}]\tau_i \upharpoonleft^r$ then $\mathit{ftv}([\overline{a \mapsto \tau_a}]\tau_i) \subseteq \mathit{ftv}(\Gamma)$. Then, if $[\overline{a_r \mapsto \tau_a^r}]$ denotes the restriction of $[\overline{a \mapsto \tau_a}]$ on $\overline{a}_r$ it must be that $\mathit{ftv}(\overline{\tau}_a^r) \subseteq \mathit{ftv}(\Gamma)$. But note that when $m = r$ then $\mathit{ftv}([\overline{a \mapsto \tau_a}]\tau) \subseteq \mathit{ftv}([\overline{a_r \mapsto \tau_a^r}]\tau) \subseteq \mathit{ftv}(\Gamma)$. The second part is an easy check. □

Corollary 6.2 (Wobbly modifier inference). *If there exists an $a \in \mathit{ftv}(\tau)$ with $a \notin \mathit{ftv}(\Gamma)$ and $\Gamma \vdash t : \tau \upharpoonleft^m$ then $m = w$.*

Proof. If, by contradiction, $m = r$ then by Lemma 6.1 it must be $\mathit{ftv}(\tau) \subseteq \mathit{ftv}(\Gamma)$. □

Let us recall the standard ordering of HM types, denoted as $\vdash \sigma_1 \leq \sigma_2$. It is well-known that this relation is transitive and is stable under substitution. We extend this ordering in environments as follows:

$$\Gamma_1 \leq \Gamma_2 \quad \text{iff} \quad \begin{cases} \nu{:}^m \sigma_1 \in \Gamma_1 \Leftrightarrow \nu{:}^m \sigma_2 \in \Gamma_2 \text{ and} \\ \quad ((m = r \wedge\ \sigma_1 = \sigma_2) \vee \\ \quad\quad (m = w \wedge \vdash \sigma_1 \leq \sigma_2)) \\ \texttt{a} =^m \tau \in \Gamma_1 \Leftrightarrow \texttt{a} =^m \tau \in \Gamma_2 \end{cases}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p \to t :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p :^{m_p} \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \theta(\Gamma \cup \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t) \succ (\mathcal{A}_2, \delta_2) \\ \mathit{bindings}(\delta_2[\Delta]) \subseteq \overline{e} \times \overline{c} \quad \mathit{ftv}(\delta_2[\Gamma], \delta_2\tau_p, \delta_2\tau_t) \# \overline{e} \times \overline{c}\end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p \to t :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t \succ (\mathcal{A}_2, \delta_2)} \ \text{APAT-REFINE}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash q :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{x \notin \mathit{dom}(\Delta)}{(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta, \theta\} \vdash x :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, x :^m \tau, \theta\} \succ (\mathcal{A}_0, \delta_0)} \ \text{APVAR}$$

$$\frac{\begin{array}{c}\overline{\mathtt{b}} = \mathit{ftv}(\mathtt{tau}) - \mathit{dom}(\Gamma \cup \Delta_1) \\ \mathcal{A}_0 \succ \Gamma \cup \Delta_1, \overline{\mathtt{b} \hookrightarrow \beta} \vdash \mathtt{tau} \Rightarrow \tau_1 \,;\, \emptyset \succ \mathcal{A}_1 \\ \delta_0 \succ \theta(\tau_1) \doteq \theta^m(\tau_2) \succ \delta_1 \quad \delta_1\beta = \overline{c} \quad \overline{c} \# \mathit{dom}(\theta) \quad \overline{c} \text{ distinct} \quad \overline{c} \# \mathit{bindings}(\Delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \overline{\mathtt{b} \hookrightarrow c}, \theta\} \vdash q :^r [\overline{\beta \mapsto c}]\tau_1 \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_2, \delta_2)\end{array}}{(\mathcal{A}_0 \overline{\beta}, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash (q :: \mathtt{tau}) :^m \tau_2 \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_2, \delta_2)} \ \text{APANN}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{\begin{array}{c}C :^r \forall \overline{b}.\overline{\tau}_1 \to T\,\overline{\tau}_2 \in \Gamma \quad \overline{b}_c = \overline{b} \cap \mathit{ftv}(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \\ \psi_c = [\overline{b_c \mapsto \beta}] \quad \psi_e = [\overline{b_e \mapsto c}] \quad \delta_0 \succ \tau \doteq T\,\psi_c(\overline{\tau}_2) \succ \delta_1 \quad m_i = \begin{cases} r & \mathit{ftv}(\tau_{1i}) \# \overline{b}_c \\ w & \mathit{otherwise} \end{cases} \\ (\mathcal{A}_0, \delta_1) \succ \Gamma, \{\overline{c} \times \overline{b}_c, \cdot, \emptyset\} \vdash^{\mathit{fold}} \overline{q_i :^{m_i} \psi_c(\psi_e(\tau_{1i}))} \blacktriangleright \{\overline{c} \times \overline{b}_c, \Delta_n, \emptyset\} \succ (\mathcal{A}_1, \delta_2)\end{array}}{(\mathcal{A}_0 \overline{c}\overline{\beta}, \delta_0) \succ \Gamma \vdash C\,\overline{q} :^w \tau \blacktriangleright \{\overline{c} \times \overline{b}_c, \Delta_n, \emptyset\} \succ (\mathcal{A}_1, \delta_2)} \ \text{APCON-W}$$

$$\frac{\begin{array}{c}C :^r \forall \overline{b}.\overline{\tau}_1 \to T\,\overline{\tau}_2 \in \Gamma \quad \tau = T\,\overline{\tau}_3 \quad \mathcal{A}_0 \succ \psi = \mathtt{fmgu}(\overline{\tau}_3 \doteq [\overline{b \mapsto c}]\overline{\tau}_2) \succ \mathcal{A}_1 \\ (\mathcal{A}_1, \delta_0) \succ \Gamma, \{\overline{c} \times \emptyset, \cdot, \psi\} \vdash^{\mathit{fold}} \overline{q_i :^r [\overline{b \mapsto c}]\tau_{1i}} \blacktriangleright \{\overline{c} \times \emptyset, \Delta_n, \psi\} \succ (\mathcal{A}_2, \delta_1)\end{array}}{(\mathcal{A}_0 \overline{c}, \delta_0) \succ \Gamma \vdash C\,\overline{q} :^r \tau \blacktriangleright \{\overline{c} \times \emptyset, \Delta_n, \psi\} \succ (\mathcal{A}_2, \delta_1)} \ \text{APCON-R}$$

$$\frac{(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\emptyset, \cdot, \varepsilon\} \vdash q :^m \tau \blacktriangleright \{\emptyset, \Delta, \varepsilon\} \succ (\mathcal{A}_1, \delta_1)}{(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash q :^m \tau \blacktriangleright \{\emptyset, \Delta, \varepsilon\} \succ (\mathcal{A}_1, \delta_1)} \ \text{APQ}$$

$$\boxed{(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash^{\mathit{fold}} \overline{q_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\} \succ (\mathcal{A}_1, \delta_1)}$$

$$\frac{}{(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta, \theta\} \vdash^{\mathit{fold}} \cdot \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \succ (\mathcal{A}_0, \delta_0)} \ \text{AF-BASE}$$

$$\frac{\begin{array}{c}(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash q :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_1, \delta_1) \\ (\mathcal{A}_1, \delta_1) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \vdash^{\mathit{fold}} \overline{q_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\} \succ (\mathcal{A}_n, \delta_n)\end{array}}{(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash^{\mathit{fold}} (q :^m \tau), \overline{q_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\} \succ (\mathcal{A}_n, \delta_n)} \ \text{AF-REC}$$

Figure 15: Algorithmic pattern typing (shallow)

Lemma 6.3 (Free variables and subsumption). *If* $\Gamma_1 \leq \Gamma_2$ *then* $ftv(\Gamma_1) \subseteq ftv(\Gamma_2)$.

Proof. Easy check. $\square$

The following lemma is crucial for the completeness of type inference.
Lemma 6.4 (Weakening). *The following are true:*

1. *If* $\Gamma_1 \leq \Gamma_2$ *and* $\Gamma_2 \vdash t :^m \tau$ *then* $\Gamma_1 \vdash t :^m \tau$.
2. *If* $\Gamma_1 \leq \Gamma_2$ *and* $\Gamma_2 \vdash t : \tau \rceil^m$ *then* $\Gamma_1 \vdash t : \tau \rceil^m$.
3. *If* $\Gamma_1 \leq \Gamma_2$ *and* $\Gamma_2 \vdash^{poly} t : \sigma \rceil^m$ *then* $\Gamma_1 \vdash^{poly} t : \sigma \rceil^m$.
4. *If* $\Gamma_1 \leq \Gamma_2$ *and* $\Gamma_2 \vdash p \to t :^{\langle m_1, m_2 \rangle} \tau_1 \to \tau_2$ *then* $\Gamma_1 \vdash p \to t :^{\langle m_1, m_2 \rangle} \tau_1 \to \tau_2$.

Moreover the new derivations have the same height.

Proof. We prove the four claims simultaneously by induction on the height of the derivations. The only interesting part is the last one, the rest are straightforward appealing occasionally to Lemma 6.3. For the last part it is easy to observe that if $\Gamma_2 \vdash_X p :^m \tau \blacktriangleright K$ then also $\Gamma_1 \vdash_X p :^m \tau \blacktriangleright K$. Therefore in PAT we will be done if we show that $\theta(\Gamma_1, \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t)$ and $ftv(\Gamma_1, \tau_p, \tau_t) \# \overline{e} \times \overline{c}$ given that $\theta(\Gamma_2, \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t)$ and $ftv(\Gamma_2, \tau_p, \tau_t) \# \overline{e} \times \overline{c}$. The first follows because subsumption is preserved under substitution and the second follows by Lemma 6.3. $\square$

Lemma 6.5 (q-pattern substitution).

1. *If* $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash^{fold}_X \overline{p :^m \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\}$, $vars(\phi) \subseteq X$ *and* $dom(\phi) \# \Gamma|_r, \Delta_1|_r, \bigcup \tau_i|_{m_i}$ *then*
$$\phi[\Gamma], \{\overline{e} \times \overline{c}, \phi[\Delta_1], \theta\} \vdash^{fold}_X \overline{p :^m \phi[\tau]} \blacktriangleright \{\overline{e} \times \overline{c}, \phi[\Delta_n], \theta\}$$
and $dom(\phi) \# \Delta_n|_r$.
2. *If* $\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\}$, $vars(\phi) \subseteq X$ *and* $dom(\phi) \# \Gamma|_r, \Delta_1|_r, \tau|_m$ *then*
$$\phi[\Gamma], \{\overline{e} \times \overline{c}, \phi[\Delta_1], \theta\} \vdash_X p :^m \phi[\tau] \blacktriangleright \{\overline{e} \times \overline{c}, \phi[\Delta_2], \theta\}$$
and $dom(\phi) \# \Delta_2|_r$.

Proof. Straightforward induction. $\square$

Lemma 6.6 (p-pattern substitution). *If* $\Gamma \vdash_X p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\}$, $vars(\phi) \subseteq X$ *and* $dom(\phi) \# \Gamma|_r, \tau|_m$ *then* $\phi[\Gamma] \vdash_X p :^m \phi\tau \blacktriangleright \{\overline{e} \times \overline{c}, \phi[\Delta], \theta\}$.

Proof. We have three cases to consider depending on the last rule used in the derivation of $\Gamma \vdash_X p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\}$. The case for PQ follows from Lemma 6.5. For the other two cases, for the case of PCON-R we know that ϕ will be disjoint from the type of the pattern and the argument types and appealing to the first part of Lemma 6.5 and rule PCON-R again finishes the case. For the case of PCON-W we have that:

$$\Gamma \vdash_X C\ \overline{q} :^w T\ \overline{\tau}_3 \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\}$$

given that

$$C :^r \forall \overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \tag{1}$$

$$\overline{b}_c = \overline{b} \cap ftv(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \tag{2}$$

$$\psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \tag{3}$$

$$m_i = \begin{cases} r & ftv(\tau_{1i})\#\overline{b}_c \\ w & \text{otherwise} \end{cases} \tag{4}$$

$$\Gamma, \{\overline{b}_e \times \overline{b}_c, \cdot, \emptyset\} \vdash^{fold}_X \overline{q_i :^{m_i} \psi(\tau_{1i})} \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \Delta_n, \emptyset\} \tag{5}$$

Consider $\psi_\phi = [\overline{b_c \mapsto \phi(\upsilon)}]$, then it must be that $\phi[\overline{\tau}_3] = \psi_\phi(\overline{\tau}_2)$. Moreover $\psi_\phi(\overline{\tau}_1) = \phi\psi(\overline{\tau}_1)$. By induction hypothesis we get that

$$\begin{aligned} \phi[\Gamma], \{\overline{b}_e \times \overline{b}_c, \cdot, \emptyset\} \vdash^{fold}_X & \overline{q_i :^{m_i} \phi[\psi(\tau_{1i})]} \\ & \blacktriangleright \{\overline{b}_e \times \overline{b}_c, \phi[\Delta_n], \emptyset\} \end{aligned} \tag{6}$$

and by applying rule PCON-W we get the result. □

The following type substitution theorem is essential for soundness of algorithmic type inference. We call it *weak* substitution theorem because substitution occurs only inside wobbly parts of contexts and types[1].

Lemma 6.7 (Weak substitution).

1. *If* $\Gamma \vdash t :^m \tau$ *and* $dom(\theta)\#ftv(\Gamma|_r, \tau|_m)$ *then* $\theta[\Gamma] \vdash t :^m \theta[\tau]$.
2. *If* $\Gamma \vdash t : \tau \uparrow^m$ *and* $dom(\theta)\#ftv(\Gamma|_r)$ *then* $\theta[\Gamma] \vdash t : \theta[\tau] \uparrow^m$.
3. *If* $\Gamma \vdash^{poly} t : \sigma \uparrow^m$ *and* $dom(\theta)\#ftv(\Gamma|_r)$ *then* $\theta[\Gamma] \vdash t : \theta[\sigma] \uparrow^m$.
4. *If* $\Gamma \vdash p \to t :^{\langle m_1, m_2 \rangle} \tau_1 \to \tau_2$ *and* $dom(\theta)\#ftv(\Gamma|_r, \tau_1|_{m_1}, \tau_2|_{m_2})$ *then* $\theta[\Gamma] \vdash p \to t :^{\langle m_1, m_2 \rangle} \theta[\tau_1 \to \tau_2]$.

Proof. The first four parts are easy checking. In the claims about the modifier inference judgement, Lemma 6.1 is used. The last part is the most interesting. By rule PAT we have that

$$\Gamma \vdash p \to t :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t \tag{1}$$

given that

$$\Gamma \vdash p :^{m_p} \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \tag{2}$$

$$\theta(\Gamma, \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t) \tag{3}$$

$$ftv(\Gamma, \tau_p, \tau_t)\#\overline{e} \times \overline{c} \quad bindings(\Delta) \subseteq \overline{e} \times \overline{c} \tag{4}$$

Assume also without loss of generality that $\overline{e} \times \overline{c}$ are disjoint from the variables of ϕ otherwise we can rename them consistently. Then, by Lemma 6.6 we get that

$$\phi[\Gamma], \{\emptyset, \cdot, \varepsilon\} \vdash p :^{m_p} \phi(\tau_p) \blacktriangleright \{\overline{a}, \phi[\Delta], \theta\} \tag{5}$$

Moreover we know that $vars(\theta)$ appear only in rigid parts of the context and the τ_1 if it is rigid, or are in $\overline{e} \times \overline{c}$. Therefore by induction hypothesis we can get

$$\theta(\phi[\Gamma], \phi[\Delta]) \vdash t :^{m_t} \theta^{m_t}(\phi(\tau_t)) \tag{6}$$

Moreover by our freshness assumption for $\overline{e} \times \overline{c}$, $ftv(\phi[\Gamma], \phi(\tau_t), \phi(\tau_p))\#\overline{e} \times \overline{c}$. Applying rule PAT finishes the case. □

The ordinary version of the theorem that one would expect, where there is no restriction on the domain of the substitution, is not true in general.

[1] The intuition being that only wobbly parts of types and contexts can contain unification variables that may be substituted for other types.

6.1 Algorithmic inference

We now present an algorithm for type inference. The algorithm is given in Figures 11-15. It relies on first order unification. We use letters α, β, γ to denote meta variables and δ,ζ to denote *unifiers*, that is, are idempotent maps from meta variables to monotypes. We write $ftv(\sigma), fmv(\sigma), fov(\sigma)$ for the free variables, free meta variables, and free ordinary variables respectively of σ. We closely follow the development in [VWP05] for some of the results presented in here. We assume a judgement $\delta_0 \succ \tau_1 \doteq \tau_2 \succ \delta_1$ that gives first-order unification. Its definition is straightforward and can be found for example in [VWP05]. For the fresh most general unifier procedure we use the judgement given by $\theta\,;\mathcal{A}_1 = \texttt{fmgu}(\tau_1 \doteq \tau_2\,;\mathcal{A}_0)$. Recall also from previous sections that $\theta_1 = \theta_2\backslash_{\overline{a}}$ mean that the two substitutions agree everywhere *except perhaps for some variables in* $\overline{a}$.

The algorithm presented here can be easily shown terminating (termination indicated with a case not being present). Moreover the judgements presented are deterministic in the "output" variables. We ommit these theorems and take them as true as they are not difficult nor interesting enough. In what follows we are being implicit about the "avoidance" set passed down to the pattern checking judgement—a formal treatment of this set and its relation to the algorithm symbol supply set is not interesting. We therefore sometimes completely ommit writing the X set from the pattern checking judgements, to avoid clutter.

Definition 6.8 (Rigidity for types and contexts). *Rigid types should not contain unification variables. This is captured with the relation* $\Vdash^m \sigma$.

$$\Vdash^m \sigma \quad \textit{iff} \quad m = r \Rightarrow fmv(\sigma) = \emptyset$$

We naturally extend the above definition to contexts:

$$\Vdash \Gamma \quad \textit{iff} \quad \begin{array}{l} \forall (x{:}^m \sigma) \in \Gamma.\ \Vdash^m \sigma \\ \forall (\mathsf{a} \hookrightarrow \tau) \in \Gamma.\ \Vdash^r \tau \end{array}$$

Propositions 6.9-6.14 give known results about unification and arrow unification.

Proposition 6.9 (Idempotency of unifiers). *If δ_0 is idempotent and $\delta_0 \succ\vdash \tau_1 \doteq \tau_2 \succ \delta_1$ then δ_1 is idempotent.*

Proposition 6.10 (Unification soundnesss). *If $\delta_0 \succ\vdash \tau_1 \doteq \tau_2 \succ \delta_1$ then $\delta_1\tau_1 = \delta_1\tau_2$ and there exists a substitution δ_r such that $\delta_1 = \delta_r \cdot \delta$. Moreover $vars(\delta_r) \subseteq ftv(\delta_0\tau_1, \delta_0\tau_2)$.*

Proposition 6.11. *If $\delta_0 \succ\vdash \tau_1 \doteq \tau_2 \succ \delta_1$ then $fov(\delta_1\tau_1, \delta_1\tau_2) \subseteq fov(\delta_0\tau_1, \delta_0\tau_2)$.*

Proposition 6.12 (Unification completeness). *If $\delta\delta_0\tau_1 = \delta\delta_0\tau_2$ then $\delta_0 \succ\vdash \tau_1 \doteq \tau_2 \succ \delta_1$, such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1$ for some δ_r.*

Proposition 6.13 (Arrow unification soundness and completeness).

1. *If for some $\mathcal{A}_0$ such that $\mathcal{A}_0\#ftv(\rho), vars(\delta_0)$ it is $(\mathcal{A}_0, \delta_0) \succ\vdash \tau \doteq^{\rightarrow} \tau_1 \rightarrow \tau_2 \succ (\delta_1, \mathcal{A}_1)$ then $\delta_1\tau = \delta_1\tau_1 \rightarrow \delta_1\tau_2$ and $\exists\delta_r$ such that $\delta_1 = \delta_r \cdot \delta_0$. Moreover $ftv(\tau_1 \rightarrow \tau_2) \subseteq ftv(\tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, and $vars(\delta_r) \subseteq ftv(\delta_0\tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$.*

2. *If $\delta\delta_0\tau = \tau_1 \rightarrow \tau_2$ then for any symbol supply $\mathcal{A}_0$ such that $\mathcal{A}_0\#vars(\delta, \delta_0)$ and $\mathcal{A}_0\#ftv(\tau, \tau_1, \tau_2)$, $(\mathcal{A}_0, \delta_0) \succ\vdash \tau \doteq^{\rightarrow} \tau_1' \rightarrow \tau_2' \succ (\delta_1, \mathcal{A}_1)$ such that $\exists\delta_r$ with $\delta \cdot \delta_0 = \delta_r \cdot \delta_1\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ and $\delta_r\delta_1\tau_1' = \tau_1$ and $\delta_r\delta_1\tau_2' = \tau_2$.*

Proposition 6.14. *If $\mathcal{A}_0\#ftv(\rho), vars(\delta_0)$ and $(\mathcal{A}_0, \delta_0) \succ\vdash \tau \doteq^{\rightarrow} \tau_1 \rightarrow \tau_2 \succ (\delta_1, \mathcal{A}_1)$ then $fov(\delta_1\tau, \delta_1\tau_1, \delta_1\tau_2) \subseteq fov(\delta_0\tau)$.*

Lemma 6.15 (Type translation soundness). *If $\mathcal{A}_0\#ftv(\Gamma)$, $\Vdash \Gamma$, and $\mathcal{A}_0 \succ \Gamma \vdash \texttt{sig} = \sigma, \overline{\mathsf{a} \hookrightarrow a} \succ \mathcal{A}_1$ then $[\![\texttt{sig}]\!]_\Gamma = \sigma$ with $\sigma = \forall\overline{a}.\tau$.*

Proof. Easy induction. □

Lemma 6.16 (Type translation completeness). *If* $[\![\mathtt{sig}]\!]_{\delta[\Gamma]} = \forall\overline{a}.\tau$, $\Vdash \Gamma$, $\mathtt{sig} = \forall\overline{\mathtt{a}}.\mathtt{tau}$ *then* $\mathcal{A}_0\#vars(\delta), ftv(\Gamma)$ $\mathcal{A}_0 \succ \Gamma \vdash \mathtt{sig} = \forall\overline{a}.\tau, \overline{\mathtt{a} \hookrightarrow a} \succ \mathcal{A}_1$.

Proof. Easy induction. ◻

6.2 Soundness properties

Lemma 6.17 (Soundness for typing).

1. *If* $\Vdash \Gamma$ *and* $\Vdash^m \tau$, $\mathcal{A}_0\#ftv(\Gamma,\tau), vars(\delta_0)$, *and* $(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash t :^m \tau \succ (\mathcal{A}_1,\delta_1)$, *then there exists a* δ_r *such that* $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0\Gamma, \delta_0\tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, *and* $\delta_1[\Gamma] \vdash t :^m \delta_1(\tau)$.
2. *If* $\Vdash \Gamma$, $\mathcal{A}_0\#ftv(\Gamma), vars(\delta_0)$, *and* $(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash t : \tau \uparrow^m \succ (\mathcal{A}_1,\delta_1)$, *then there exists* δ_r *such that* $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0\Gamma) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, $ftv(\tau) \subseteq ftv(\Gamma, \delta_0\Gamma) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, *and* $\delta_1[\Gamma] \vdash t : \delta_1\tau \uparrow^m$. *Moreover* $\Vdash^m \tau$.
3. *If* $\Vdash \Gamma$, $\mathcal{A}_0\#ftv(\Gamma), vars(\delta_0)$, *and* $(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash^{poly} t : \sigma \uparrow^m \succ (\mathcal{A}_1,\delta_1)$, *then* $\exists\delta_r$ *such that* $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0\Gamma) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, $ftv(\sigma) \subseteq ftv(\Gamma, \delta_0\Gamma)$, *and* $\delta_1[\Gamma] \vdash^{poly} t : \delta_1\sigma \uparrow^m$. *Moreover* $\Vdash^m \sigma$.
4. *If* $\Vdash \Gamma$, $\Vdash^{m_1} \tau_1$, *and* $\Vdash^{m_2} \tau_2$, $\mathcal{A}_0\#ftv(\Gamma,\tau_1,\tau_2), vars(\delta_0)$, *and* $(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash p \rightarrow t :^{\langle m_1,m_2\rangle} \tau_1 \rightarrow \tau_2 \succ (\mathcal{A}_1,\delta_1)$, *then there exists* δ_r *such that* $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0\Gamma, \delta_0\tau_1, \delta_0\tau_2) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, *and* $\delta_1[\Gamma] \vdash p \rightarrow t :^{\langle m_1,m_2\rangle} \delta_1(\tau_1 \rightarrow \tau_2)$.

Proof. The four claims are proved simultaneously by induction on the height of the derivations. For each part, the induction hypothesis asserts all claims by induction on the height of the derivation. The first three parts are straightforward in the style of the soundness proof given at [VWP05]. We examine the last claim wher rule APAT-REFINE give us that

$$(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash p \rightarrow t :^{\langle m_p,m_t\rangle} \tau_p \rightarrow \tau_t \succ (\mathcal{A}_2,\delta_2)$$

given that

$$(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash p :^{m_p} \tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \succ (\mathcal{A}_1,\delta_1) \tag{7}$$
$$(\mathcal{A}_1,\delta_1) \succ \theta(\Gamma \cup \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t) \succ (\mathcal{A}_2,\delta_2) \tag{8}$$
$$bindings(\delta_2[\Delta]) \subseteq \overline{e} \times \overline{c} \tag{9}$$
$$ftv(\delta_2[\Gamma], \delta_2\tau_p, \delta_2\tau_t)\#\overline{e} \times \overline{c} \tag{10}$$

By Lemma 6.20 we get that

$$\delta_1[\Gamma], \{\emptyset, \cdot, \varepsilon\} \vdash p :^{m_p} \delta_1\tau_p \blacktriangleright \{\overline{a}, \delta_1[\Delta], \theta\} \tag{11}$$

and $\delta_1 = \delta_r \cdot \delta_0$. Then, by induction hypothesis we get

$$\delta_2[\theta(\Gamma,\Delta)] \vdash t :^{m_t} \delta_2\theta^{m_t}(\tau_t) \tag{12}$$

such that $\delta_2 = \delta_r' \cdot \delta_1$. Because the domain of δ_2 contains only unification variables, appearing in wobbly parts of the judgement, we get

$$\theta(\delta_2[\Gamma,\Delta])] \vdash t :^{m_t} \theta^{m_t}\delta_2(\tau_t) \tag{13}$$

By the weak substitution lemma for patterns we have

$$\delta_2[\Gamma], \{\emptyset, \cdot, \varepsilon\} \vdash p :^{m_p} \delta_2\tau_p \blacktriangleright \{\overline{a}, \delta_2[\Delta], \theta\} \tag{14}$$

and we are finally done by applying rule PAT. ◻

Lemma 6.18 (Soundness for instantiation). *If $\mathcal{A}_0 \# ftv(\sigma, \tau)$, $vars(\delta_0)$ and $(\mathcal{A}_0, \delta_0) \succ \vdash \sigma \leq \tau \succ (\mathcal{A}_1, \delta_1)$ then $\exists \delta_r$ such that $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0 \sigma, \delta_0 \tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, and $\vdash \delta_1(\sigma) \leq \delta_1(\tau)$.*

Proof. Using rule INST we get that $\sigma = \forall \overline{a}.\tau_1$ and $(\mathcal{A}_0 \overline{\beta}, \delta_0) \succ \vdash \forall \overline{a}.\tau_1 \leq \tau \succ (\mathcal{A}_0, \delta_1)$, given that $\delta_0 \succ [\overline{a \mapsto \beta}]\tau_1 \doteq \tau \succ \delta_1$. By Lemma 6.10 we get that $\exists \delta_r$ such that $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0 [\overline{a \mapsto \beta}]\tau_1, \delta_0 \tau)$, or $vars(\delta_r) \subseteq ftv(\delta_0(\forall \overline{a}.\tau_1), \delta_0 \tau) \cup \overline{\beta}$. Moreover $\delta_1([\overline{a \mapsto \beta}]\tau_1) = \delta_1 \tau_2$, or $[\overline{b \mapsto \delta_1 \beta}]\delta_1[\overline{a \mapsto b}]\tau_1 = \delta_1 \tau_2$, where $\overline{b} \# vars(\delta_1)$. This gives $\vdash \forall \overline{b}.\delta_1[\overline{a \mapsto b}]\tau_1 \leq \delta_1 \tau$ or $\vdash \delta_1(\forall \overline{a}.\tau_1) \leq \delta_1 \tau$ as required. □

Lemma 6.19 (Soundness for pattern checking). *The following hold:*

1. *If $\mathcal{A}_0 \# ftv(\tau, \Gamma, \Delta_1, \overline{e} \times \overline{c}) \cup vars(\delta_0, \theta)$, $\Vdash \Gamma, \Delta_1$ and $\vdash^m \tau$, and*

$$(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash q :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_1, \delta_1)$$

   then there exists δ_r such that $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0[\Gamma, \Delta_2], \delta_0 \tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, and $\delta_1[\Gamma], \{\overline{e} \times \overline{c}, \delta_1[\Delta_1], \theta\} \vdash q :^m \delta_1(\tau) \blacktriangleright \{\overline{e} \times \overline{c}, \delta_1[\Delta_2], \theta\}$. Moreover $\Vdash \Delta_2$.

2. *If $\mathcal{A}_0 \# ftv(\overline{\tau}, \Gamma, \Delta_1, \overline{e} \times \overline{c}) \cup vars(\delta_0, \theta)$, $\Vdash \Gamma, \Delta_1$, for every i it is $\vdash^{m_i} \tau_i$, and*

$$(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash^{fold} \overline{q :^m \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \theta\} \succ (\mathcal{A}_n, \delta_n)$$

   then there exists δ_r such that $\delta_n = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0[\Gamma, \Delta_n], \delta_0 \tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_n)$, and $\delta_n[\Gamma], \{\overline{e} \times \overline{c}, \delta_n[\Delta_1], \theta\} \vdash^{fold} \overline{p :^m \delta_n(\tau)} \blacktriangleright \{\overline{e} \times \overline{c}, \delta_n[\Delta_n], \theta\}$. Moreover $\Vdash \Delta_n$.

Proof. We prove the two claims simultaneously by induction on the height of the derivations. For the second part, the case for AF-BASE trivially follows by F-BASE. The case for AF-REC is straightforward application of the induction hypothesis for both parts and rule F-REC, appealing to the substitution lemma, Lemma 6.6. For the first part we have the following cases.

- Case APVAR. In this case we have that

$$(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta, \theta\} \vdash x :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, x :^m \tau, \theta\} \succ (\mathcal{A}_0, \delta_0) \tag{15}$$

  Pick δ_r to be the empty substitution. Then we need to show that

$$\delta_0[\Gamma], \{\overline{a}, \delta_0[\Delta], \theta\} \vdash x :^m \delta_0 \tau \blacktriangleright \{\overline{a}, \delta_0[\Delta], x :^m \delta_0 \tau, \theta\} \tag{16}$$

  But this follows directly by rule PVAR. Moreover since $\Vdash^m \tau$ it follows that $\Vdash (x :^m \tau)$ as required.

- Case APANN. In this case we have that

$$(\mathcal{A}_0 \overline{\beta}, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash (q :: \mathtt{tau}) :^m \tau_2 \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_2, \delta_2) \tag{17}$$

  given that

$$\overline{\mathtt{b}} = ftv(\mathtt{tau}) - dom(\Gamma \cup \Delta_1) \tag{18}$$

$$\mathcal{A}_0 \succ \Gamma \cup \Delta_1, \overline{\mathtt{b} \hookrightarrow \beta} \vdash \mathtt{tau} \Rightarrow \tau_1 \,;\, \emptyset \succ \mathcal{A}_1 \tag{19}$$

$$\delta_0 \succ \theta(\tau_1) \doteq \theta^m(\tau_2) \succ \delta_1 \quad \delta_1 \beta = \overline{c} \quad \overline{c} \# dom(\theta) \quad \overline{c} \text{ distinct} \quad \overline{c} \# bindings(\Delta_1) \tag{20}$$

$$(\mathcal{A}_1, \delta_1) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \overline{\mathtt{b} \hookrightarrow c}, \theta\} \vdash q :^r [\overline{\beta \mapsto c}]\tau_1 \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \succ (\mathcal{A}_2, \delta_2) \tag{21}$$

  By Lemma 6.15 we get that $[\![\mathtt{tau}]\!]_{\Gamma, \Delta_1, \overline{\mathtt{b} \hookrightarrow \beta}} = \tau_1$, therefore also $[\![\mathtt{tau}]\!]_{\delta_2[\Gamma, \Delta, \overline{\mathtt{b} \hookrightarrow c}]} = [\overline{\beta \mapsto c}]\tau_1$. We need to show that $\theta([\overline{\beta \hookrightarrow c}]\tau_1) = \delta_2(\theta^m(\tau_2))$. By unification soundness we know that $\delta_1(\theta(\tau_1)) = \delta_1(\theta^m(\tau_2))$, and $\delta_1 = \delta_r \cdot \delta_0$. therefore

$$\theta([\overline{\beta \hookrightarrow c}]\tau_1) = \delta_1(\theta^m(\tau_2))$$

Moreover, by induction hypothesis

$$\delta_2[\Gamma], \{\overline{e} \times \overline{c}, \delta_2[\Delta_1, \overline{b \hookrightarrow c}], \theta\} \vdash q :^r \delta_2([\overline{\beta \mapsto c}]\tau_1) \blacktriangleright \{\overline{e} \times \overline{c}, \delta_2[\Delta_2], \theta\}$$

and $\delta_2 = \delta_r' \delta_1 = \delta_r' \delta_r \cdot \delta_0$. Hence it must also be that

$$\delta_r' \theta([\overline{\beta \mapsto c}]\tau_1) = \delta_2(\theta^m(\tau_2))$$

but we know that $\Vdash \tau_2$; therefore we get $\theta([\overline{\beta \mapsto c}]\tau_1) = \theta^m(\delta_2 \tau_2)$ as required. Applying rule PANN finishes the case.

$\square$

Lemma 6.20 (Soundness for pattern checking). *If $\mathcal{A}_0 \# ftv(\tau, \Gamma), vars(\delta_0), \Vdash \Gamma$ and $\vdash^m \tau$, and $(\mathcal{A}_0, \delta_0) \succ \Gamma, \vdash p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \succ (\mathcal{A}_1, \delta_1)$ then there exists a δ_r such that $\delta_1 = \delta_r \cdot \delta_0$, $vars(\delta_r) \subseteq ftv(\delta_0[\Gamma, \Delta], \delta_0 \tau) \cup fmv(\mathcal{A}_0 - \mathcal{A}_1)$, and $\delta_1[\Gamma], \{\emptyset, \cdot, \varepsilon\} \vdash_X p :^m \delta_1(\tau) \blacktriangleright \{\overline{e} \times \overline{c}, \delta_1[\Delta], \theta\}$ for any X disjoint from $\mathcal{A}_0 - \mathcal{A}_1$. Moreover $\Vdash \Delta$.*

Proof. Straightforward, appealing to Lemma 6.19 $\square$

6.3 Completeness properties

For completeness we first adopt the notation $\overline{\Gamma}(\tau)$ from [VWP05] to mean the generalization of τ over its free variables that do not appear in Γ. Assume in what follows that $\vdash \cdot \leq \cdot$ is the standard subsumption relation.

Lemma 6.21. $\vdash \delta\overline{\Gamma}(\tau) \leq \overline{\delta\Gamma}(\delta\rho)$.

Lemma 6.22 (Completeness for instantiation). *If $\vdash \delta\delta_0\sigma \leq \delta\delta_0\tau$ then for any $\mathcal{A}_0 \# vars(\delta, \delta_0), ftv(\tau, \sigma)$ $(\mathcal{A}_0, \delta_0) \succ \vdash \sigma \leq \tau \succ (\mathcal{A}_1, \delta_1)$ and $\exists \delta_r$ such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$*

Proof. It is the case that $\vdash \delta\delta_0\sigma \leq \delta\delta_0\tau$ and assume $\sigma = \forall \overline{a}.\tau_1$. Consider a supply $\mathcal{A}_0\overline{\beta}$ that satisfies the freshness condition. Then by inversion it must be that $\vdash [\overline{\beta \mapsto \tau}]\delta\delta_0([\overline{a \mapsto \beta}]\tau_1) = \delta\delta_0\tau$. But because of the freshness of the symbol supply this can be rewritten as $\vdash [\overline{\beta \mapsto \tau}]\delta\delta_0([\overline{a \mapsto \beta}]\tau_1) = [\overline{\beta \mapsto \tau}]\delta\delta_0\tau$. By unification completeness, Lemma 6.12 we have that $\delta_0 \succ \vdash [\overline{a \mapsto \beta}]\tau_1 \doteq \tau \succ \delta_1$ and moreover $[\overline{\beta \mapsto \tau}] \cdot \delta \cdot \delta_0 = \delta_r \cdot \delta_1$ for some δ_r, which gives $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\overline{\beta}}$. By applying rule INST we get the result. $\square$

Lemma 6.23 (Completeness for typing). *The following are true:*

1. *If $\vdash \delta\delta_0[\Gamma] \vdash t :^m \delta\delta_0\tau$, $\Vdash \Gamma$ and $\Vdash^m \tau$, and $\mathcal{A}_0$ an sufficiently fresh symbol sypply then $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t :^m \tau \succ (\mathcal{A}_1, \delta_1)$, $\exists \delta_r$ such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.*

2. *If $\vdash \delta\delta_0[\Gamma] \vdash t : \tau \upharpoonleft^m$, $\Vdash \Gamma$, and $\mathcal{A}_0$ an sufficiently fresh symbol supply then $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t : \tau' \upharpoonleft^m \succ (\mathcal{A}_1, \delta_1)$, $\exists \delta_r$ such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$, and $\tau = \tau'$ when $m = r$; otherwise $\tau = \delta_r \delta_1 \tau'$.*

3. *If $\vdash \delta\delta_0[\Gamma] \vdash^{poly} t : \sigma \upharpoonleft^m$, $\Vdash \Gamma$, and $\mathcal{A}_0$ an sufficiently fresh symbol supply then $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash^{poly} t : \sigma' \upharpoonleft^m \succ (\mathcal{A}_1, \delta_1)$, $\exists \delta_r$ such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$, and $\sigma = \sigma'$ when $m = r$; otherwise $\vdash \delta_r \delta_1 \tau' \leq \sigma$.*

4. *If $\vdash \delta\delta_0[\Gamma] \vdash p \to t :^{\langle m_1, m_2 \rangle}: \delta\delta_0(\tau_1 \to \tau_2)$ with $\Vdash^{m_1} \tau_1$, $\Vdash^{m_2} \tau_2$, $\Vdash \Gamma$, and $\mathcal{A}_0$ an sufficiently fresh symbol supply then $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p \to t :^{\langle m_1, m_2 \rangle} \tau_1 \to \tau_2 \succ (\mathcal{A}_1, \delta_1)$ and $\exists \delta_r$ such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.*

Proof. We prove the claims simultaneously by induction on the height of the derivations. We proceed by case analysis on the last rule used.

The first part is standard. Indicatively we give the cases for rule VAR and LET-W.

- Case VAR. In this case we have that $\delta\delta_0\Gamma \vdash \nu :^m \delta\delta_0\tau$, given that $\nu :^{m_x} \delta\delta_0\sigma \in \delta\delta_0[\Gamma]$ and $\vdash^{inst} \delta\delta_0\sigma \leq \delta\delta_0\tau$. By instantiation completeness, Lemma 6.22 we have that $(\mathcal{A}_0, \delta_0) \succ \vdash^{inst} \sigma \leq \tau \succ (\mathcal{A}_1, \delta_1)$ and there exists a δ_r such that $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{(\mathcal{A}_0 - \mathcal{A}_1)}$. Applying rule AVAR finishes the case.

- Case LET-W. In this case we have that

$$\delta\delta_0\Gamma \vdash (\texttt{let } x\texttt{=}u \texttt{ in } t) :^m \delta\delta_0\tau \tag{22}$$

given that

$$\delta\delta_0\Gamma \vdash^{poly} u : \sigma \rceil^{m_x} \tag{23}$$
$$\delta\delta_0\Gamma, x :^{m_x} \sigma \vdash t :^m \delta\delta_0\tau \tag{24}$$

Now, by induction hypothesis we know that $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash u : \sigma' \rceil^{m_x} \succ (\mathcal{A}_1, \delta_1)$ such that $\delta\delta_0 = \delta_r^1\delta_1\backslash_{(\mathcal{A}_0 - \mathcal{A}_1)}$. Moreover if $m_x = r$ then $\sigma' = \sigma$ otherwise $\delta_r^1\delta_1\sigma' \leq \sigma$. This means that $\Vdash^{m_x} \sigma'$. By the weakening lemma then (and it is easy to see that the new derivation has the same height) we get

$$\delta_r^1\delta_1\Gamma, x :^{m_x} \delta_r^1\delta_1\sigma' \vdash t :^m \delta_r^1\delta_1\tau \tag{25}$$

By induction hypothesis then

$$(\mathcal{A}_1, \delta_1) \succ \Gamma, x :^{m_x} \sigma' \vdash t :^m \tau \succ (\mathcal{A}_2, \delta_2) \tag{26}$$

and moreover $\delta_r^1\delta_1 = \delta_r^2\delta_2\backslash_{(\mathcal{A}_1 - \mathcal{A}_2)}$ for some δ_r^2, which gives $\delta\delta_0 = \delta_r^2\delta_2\backslash_{(\mathcal{A}_0 - \mathcal{A}_2)}$ and we are done by an application of ALET-W.

For the second part we have the following cases.

- Case SCR-TSIG. In this case we have that

$$\delta\delta_0[\Gamma] \vdash (t\texttt{::tau}) : \tau \rceil^m \tag{27}$$

given that

$$[\![\texttt{tau}]\!]_{\delta\delta_0[\Gamma]} = \tau \tag{28}$$
$$\delta\delta_0[\Gamma] \vdash t :^m \tau \tag{29}$$

By type translation completeness we get that $\mathcal{A}_0 \succ \Gamma \vdash \tau = \tau', \cdot \succ \mathcal{A}_1$ such that $\delta\delta_0\tau' = \tau$. Moreover it is easy to confirm that $\Vdash^m \tau'$. Then equation (29) can be rewritten as $\delta\delta_0[\Gamma] \vdash t :^m \delta\delta_0\tau'$ and by induction hypothesis $(\mathcal{A}_1, \delta_0) \succ \Gamma \vdash t :^m \tau' \succ (\mathcal{A}_2, \delta_1)$. Moreover there exists a δ_r such that $\delta\delta_0 = \delta_r\delta_1\backslash_{(\mathcal{A}_1 - \mathcal{A}_2)}$ which gives $\delta\delta_0 = \delta_r\delta_1\backslash_{(\mathcal{A}_0 - \mathcal{A}_2)}$ as well. Moreover, since $\Vdash^m \tau'$ if $m = r$ then $\tau = \tau'$ since τ' will not contain any meta variables. Otherwise $\delta_r\delta_1\tau' = \tau$ as required.

- Case SCR-OTHER. We have that $\delta\delta_0[\Gamma] \vdash t : \tau \rceil^w$ given that $\delta\delta_0[\Gamma] \vdash t :^w \tau$. Our assumption can be rewritten as $[\beta \mapsto \tau]\delta\delta_0 \vdash t :^w [\beta \mapsto \tau]\delta\delta_0\beta$ where β is a symbol of the supply $\mathcal{A}_0\beta$. Then, it certainly is that $\Vdash^w \beta$ and by induction hypothesis $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash t :^w \beta \succ (\mathcal{A}_1, \delta_1)$. Additionally there exists δ_r such that $[\beta \mapsto \tau]\delta\delta_0 = \delta_r\delta_1\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ or $\delta\delta_0 = \delta_r\delta_1\backslash_{(\mathcal{A}_0\beta - \mathcal{A}_1)}$ as required. Applying rule ASCR-OTHER finishes the case.

- Case SCR-APP. Straightforward, using the same trick of extending δ with an instantiation of new meta-variables with the types $\overline{\tau}_a$, as in the INST case of the instantiation completeness lemma, and observing that rigid bindings in well-formed contexts are not affected by applications of unifiers, since they do not contain unification variables.

The proof of the third part is similar to the proof of the second part, appealing to Lemma 6.21 as well.

For the last part we get by rule PAT that

$$\delta\delta_0[\Gamma] \vdash p \rightarrow t :^{\langle m_p, m_t \rangle} \delta\delta_0(\tau_p \rightarrow \tau_t) \tag{30}$$

given that

$$\delta\delta_0[\Gamma] \vdash p :^{m_p} \delta\delta_0\tau_p \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \theta\} \tag{31}$$

$$\theta(\delta\delta_0[\Gamma], \Delta) \vdash t :^{m_t} \theta^{m_t}(\delta\delta_0\tau_t) \tag{32}$$

$$ftv(\delta\delta_0[\Gamma], \delta\delta_0[\tau_p], \delta\delta_0[\tau_t]) \# \overline{e} \times \overline{c} \quad bindings(\Delta) \subseteq \overline{e} \times \overline{c} \tag{33}$$

Assume without loss of generality that $\overline{e} \times \overline{c}$ are picked in the same way as the algorithm would pick them. Then by pattern checking completeness, Lemma 6.27 and Lemma 6.26 we get that

$$(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_a, \theta_a\} \succ (\mathcal{A}_1, \delta_1)$$

such that $\exists \delta_r^1$ with $\delta \cdot \delta_0 = \delta_r^1 \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ and $m\theta(\delta\delta_0\Delta) = \theta_a(\delta_r^1\delta_1\Delta_a)$ for a mapping m with only fresh variables or variables from $\overline{e}$ in its domain. Then the other premise becomes

$$\theta(\delta_r^1\delta_1[\Gamma], \delta_r^1\delta_1\Delta') \vdash t :^{m_t} \theta^{m_t}(\delta\delta_0\tau_t) \tag{34}$$

and by extending m to be a renaming r_m and applying it consistently to the judgement we get:

$$r_m[\theta(\delta_r^1\delta_1[\Gamma], \delta_r^1\delta_1\Delta')] \vdash t :^{m_t} r_m[\theta^{m_t}(\delta\delta_0\tau_t)] \tag{35}$$

with the same height (we ommit the renaming substitution proof). Moreover θ and $\delta_r^1\delta_1$ commute (since θ is partially applied) and by induction hypothesis we get

$$(\mathcal{A}_1, \delta_1) \succ r_m\theta(\Gamma, \Delta') \vdash t :^{m_t} (r_m\theta)^{m_t}\tau_t \succ (\mathcal{A}_2, \delta_2) \tag{36}$$

Applying rule APAT gives the result and moreover from the induction hypothesis we also get that $\delta_r^1\delta_1 = \delta_r^2\delta_2\backslash_{(\mathcal{A}_1 - \mathcal{A}_2)}$ for some δ_r^2, consequently $\delta\delta_0 = \delta_r^2\delta_2\backslash_{(\mathcal{A}_0 - \mathcal{A}_2)}$. ☐

Lemma 6.24 (wobbly q-pattern checking completeness).

- *If* $\delta\delta_0[\Gamma], \{\overline{e} \times \overline{c}, \delta\delta_0[\Delta_1], \varepsilon\} \vdash q :^m \delta\delta_0\tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \varepsilon\}$, $\Vdash^m \tau$, $\Vdash \Gamma, \Delta_1$, *and* $\mathcal{A}_0$ *an sufficiently fresh symbol supply, then* $(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \varepsilon\} \vdash q :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2', \varepsilon\} \succ (\mathcal{A}_1, \delta_1)$ *such that* $\exists \delta_r$ *with* $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ *and* $\Delta_2 = \delta_r\delta_1\Delta_2'$, $\Vdash \Delta_2$.
- *If* $\delta\delta_0[\Gamma], \{\overline{e} \times \overline{c}, \delta\delta_0[\Delta_1], \varepsilon\} \vdash^{fold} \overline{q :^m \delta\delta_0\tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n, \varepsilon\}$, $\Vdash^{m_i} \tau_i$, $\Vdash \Gamma, \Delta_1$, *and* $\mathcal{A}_0$ *an sufficiently fresh symbol supply, then* $(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \varepsilon\} \vdash^{fold} \overline{q :^m \tau} \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_n', \varepsilon\} \succ (\mathcal{A}_n, \delta_n)$ *such that* $\exists \delta_r$ *with* $\delta \cdot \delta_0 = \delta_r \cdot \delta_n \backslash_{\mathcal{A}_0 - \mathcal{A}_n}$ *and* $\Delta_n = \delta_r\delta_1\Delta_n'$, $\Vdash \Delta_n$.

Proof. Straightforward induction. ☐

Lemma 6.25 (rigid q-pattern checking completeness).

- *If* $\delta\delta_0[\Gamma], \{\overline{e} \times \emptyset, \Delta_1, \theta\} \vdash q :^r \tau \blacktriangleright \{\overline{e} \times \emptyset, \Delta_2, \theta\}$, $\tau, \Delta_1, \Delta_{1a}$ *are completely rigid,* $\Vdash \Gamma$, *and* $\mathcal{A}_0$ *an sufficiently fresh symbol supply, and* θ *is an fmgu, and* m *connects* θ *with a biased and minimal fmgu of the same equation set,* θ_a, *and* $m\theta(\Delta_1) = \theta_a(\Delta_{1a})$ *and* $m\theta(\tau) = \theta_a(\tau_a)$ *and for every binding of* Δ_1, $\mathtt{b} \hookrightarrow b$, $\mathtt{b} \hookrightarrow mb \in \Delta_{1a}$, *and* $bindings(\Delta_2) \subseteq \overline{e}$ *then* $(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \emptyset, \Delta_{1a}, \theta_a\} \vdash q :^r \tau_a \blacktriangleright \{\overline{e} \times \emptyset, \Delta_{2a}, \theta_a\} \succ (\mathcal{A}_1, \delta_1)$ *such that* Δ_2, Δ_{2a} *are rigid,* $m\theta(\Delta_2) = \theta_a(\Delta_{2a})$ *and* $\delta_0 = \delta_1\backslash_{\mathcal{A}_0 - \mathcal{A}_1}$.

- *If* $\delta\delta_0[\Gamma], \{\overline{e} \times \emptyset, \Delta_1, \theta\} \vdash^{fold} \overline{q :^r \tau} \blacktriangleright \{\overline{e} \times \emptyset, \Delta_n, \theta\}$*,* $\overline{\tau}, \Delta_1, \Delta_{1a}$ *are completely rigid,* $\Vdash \Gamma$*, and* $\mathcal{A}_0$ *an sufficiently fresh symbol supply, and* θ *is an fmgu, and there* $\mathcal{A}_0$ *an sufficiently fresh symbol supply, and* θ *is an fmgu, and* m *connects* θ *with a biased and minimal fmgu of the same equation set,* θ_a*, and* $m\theta(\Delta_1) = \theta_a(\Delta_{1a})$ *and* $m\theta(\tau_i) = \theta_a(\tau_{ai})$ *and for every binding of* Δ_1*,* $\mathtt{b} \hookrightarrow b$*,* $\mathtt{b} \hookrightarrow mb \in \Delta_{1a}$*, and* $bindings(\Delta_n) \subseteq \overline{e}$ *then* $(\mathcal{A}_0, \delta_0) \succ \Gamma, \{\overline{e} \times \emptyset, \Delta_{1a}, \theta_a\} \vdash^{fold} \overline{q :^r \tau_a} \blacktriangleright \{\overline{e} \times \emptyset, \Delta_{na}, \theta_a\} \succ (\mathcal{A}_n, \delta_n)$ *such that* Δ_n, Δ_{na} *are rigid,* $m\theta(\Delta_n) = \theta_a(\Delta_{na})$ *and* $\delta_0 = \delta_n \backslash_{\mathcal{A}_0 - \mathcal{A}_n}$*.*

Proof. We show the two claims simultaneously by induction on the height of the derivations. The second part follows easily by induction hypothesis for the first and the second part. For the first part the case of PVAR is straightforward, we now show the case of PANN. In this case we have that

$$\Gamma, \{\overline{e} \times \overline{c}, \Delta_1, \theta\} \vdash_X (q :: \mathtt{tau}) :^m \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\}$$

given that

$$\overline{\mathtt{b}} = ftv(\mathtt{tau}) - ftv(\Gamma, \Delta) \tag{1}$$

$$[\![\mathtt{tau}]\!]_{\Gamma, \Delta_1, \overline{\mathtt{b} \hookrightarrow b}} = \tau_s \tag{2}$$

$$\overline{b} \text{ distinct} \quad \overline{b} \# dom(\theta) \quad \overline{b} \# bindings(\Delta_1) \tag{3}$$

$$\theta(\tau_s) = \theta^m(\tau) \tag{4}$$

$$\Gamma, \{\overline{e} \times \overline{c}, (\Delta_1, \overline{\mathtt{b} \hookrightarrow b}), \theta\} \vdash_X q :^r \tau_s \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_2, \theta\} \tag{5}$$

Moreover by assumptions we have that $\tau, \Delta_1, \Delta_{1a}$ are completely rigid, $\Vdash \Gamma$, and $\mathcal{A}_0$ an sufficiently fresh symbol supply, and θ is an *fmgu*, and m connects θ with a biased and minimal *fmgu* of the same equation set, θ_a, and

$$m\theta(\Delta_1) = \theta_a(\Delta_{1a}) \quad m\theta(\tau) = \theta_a(\tau_a) \tag{6}$$

and moreover

$$\mathtt{b} \hookrightarrow b \in \Delta_1 \Longrightarrow \mathtt{b} \hookrightarrow mb \in \Delta_{1a} \tag{7}$$

and

$$bindings(\Delta_2) \subseteq \overline{e} \tag{8}$$

From 2 and type translation soundness we get that

$$\mathcal{A}_0 \succ \Gamma \cup \Delta_{1a}, \overline{\mathtt{b} \hookrightarrow \beta} \vdash \mathtt{tau} \Rightarrow \tau_1 ; \emptyset \succ \mathcal{A}_1$$

such that $[\overline{\beta \mapsto mb}]\tau_1 = m\tau_s$ using also 7. Moreover we know that $m\theta(m(\tau_s)) = m\theta(\tau)$, since if for a variable $a \in \tau_s$ it is $a \in dom(m)$ then $m\theta m(a) = ma = m\theta(a)$. If $a \notin dom(m)$ we get the above equation by 4. Therefore

$$\delta_0 \succ \theta_a(\tau_1) \doteq \theta_a(\tau_2) \succ \delta_1$$

and it must be that $\delta_1(\overline{\beta}) = \overline{mb}$ since $\overline{\beta}$ are the only unification variables in $m\theta(\tau_1)$. We need to show that $\overline{mb}$ are distinct. But this is true, since m is an injective map and does never unify two variables from $\overline{e}$ by its construction—and $\overline{b} \subseteq \overline{e}$. Next, we need to show that $\overline{mb} \# dom(m\theta)$. Clearly, they are not in the domain of m. If on the other hand we have that $a \in \overline{mb}$ is also in $dom(\theta)$ then there is a $[b \mapsto c]$ binding in m where $[c \mapsto b]$ is a binding in θ. therefore $m\theta(c) = c$ and $c \notin dom(m\theta)$ as required. Finally from (7) we get that $mb \notin bindings(\Delta_{1a})$, otherwise it would have to be two bindings $b_1, b_2 \in bindings(\Delta_1)$ such that $mb_1 = mb_2$. This is impossible if they are both in the domain of m, and is impossible if none is in the domain of m. If one of them is, since both $b_1, b_2 \in \overline{e}$ this would mean that m equates two variables from $\overline{e}$, a contradiction to the construction of m. By induction hypothesis moreover we get that

$$(\mathcal{A}_1, \delta_1) \succ \Gamma, \{\overline{e} \times \overline{c}, \Delta_{1a}, \overline{\mathtt{b} \hookrightarrow mb}, \theta_a\} \vdash q :^r [\overline{\beta \mapsto mb}]\tau_1 \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_{2a}, \theta_a\} \succ (\mathcal{A}_2, \delta_2)$$

and applying rule PANN finishes the case. ☐

Let us assume that the variables bound by constructors are chosen in the type-system specification with the same mechanism as they are chosen in the algorithm. Otherwise we can apply renaming on the type system derivation to use these variables.

Lemma 6.26 (wobbly p-pattern checking completeness). *If* $\delta\delta_0[\Gamma] \vdash p :^w \delta\delta_0\tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta, \varepsilon\}$*,* $\Vdash^m \tau$*,* $\Vdash \Gamma$*, and* $\mathcal{A}_0$ *an sufficiently fresh symbol supply containing* $\overline{e} \times \overline{c}$*, then* $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p :^w \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta', \varepsilon\} \succ (\mathcal{A}_1, \delta_1)$ *such that* $\exists \delta_r$ *with* $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ *and* $\Delta = \delta_r \delta_1 \Delta'$*,* $\Vdash \Delta'$*.*

Proof. Easy induction appealing to Lemma 6.24. □

Lemma 6.27 (rigid p-pattern checking completeness). *If* $\delta\delta_0[\Gamma] \vdash p :^r \tau \blacktriangleright \{\overline{e} \times \emptyset, \Delta, \theta\}$*,* $\Vdash \Gamma$*, and* $\mathcal{A}_0$ *an sufficiently fresh symbol supply containing* $\overline{e} \times \emptyset$*, then* $(\mathcal{A}_0, \delta_0) \succ \Gamma \vdash p :^r \tau \blacktriangleright \{\overline{e} \times \overline{c}, \Delta_a, \theta_a\} \succ (\mathcal{A}_1, \delta_1)$ *such that* $\exists \delta_r$ *with* $\delta \cdot \delta_0 = \delta_r \cdot \delta_1 \backslash_{\mathcal{A}_0 - \mathcal{A}_1}$ *and* $m\theta(\Delta) = \theta_a(\Delta_a)$ *for a mapping* m *with only fresh variables or variables from* $\overline{e}$ *in its domain. Moreover* Δ, Δ_a *are completely rigid.*

Proof. Easy induction appealining to Lemma 6.25 and Lemma 2.19 for the construction of the appropriate m that connects the algorithm *fmgu* with the *fmgu* of the type system specification. □

$$\boxed{\Gamma \vdash p \rightarrow t :^{\langle m_p, m_t \rangle} \tau_p \rightarrow \tau_t}$$

$$\frac{\begin{array}{c}\Gamma, \{\emptyset, \cdot, \emptyset\} \vdash p :^{m_p} \tau_p \blacktriangleright \{\overline{a}, \Delta, \theta\} \\ \mathit{ftv}(\Gamma, \tau_p, \tau_t) \# \overline{a} \quad \mathit{bindings}(\Delta) \subseteq \overline{a} \\ \theta(\Gamma, \Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t)\end{array}}{\Gamma \vdash p \rightarrow t :^{\langle m_p, m_t \rangle} \tau_p \rightarrow \tau_t} \text{ PAT}$$

$$\boxed{\Gamma, K_1 \vdash p :^m \tau \blacktriangleright K_2}$$

$$\frac{x \notin \mathit{dom}(\Delta)}{\Gamma, \{\overline{a}, \Delta, \theta\} \vdash x :^m \tau \blacktriangleright \{\overline{a}, \Delta, x :^m \tau, \theta\}} \text{ PVAR}$$

$$\frac{\begin{array}{c}C :^r \forall \overline{b}. \overline{\tau}_1 \rightarrow T\, \overline{\tau}_2 \in \Gamma \quad \overline{b} \# \overline{a} \\ \overline{b}_c = \overline{b} \cap \mathit{ftv}(\overline{\tau}_2) \quad \psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \\ m_i = \begin{cases} r & \mathit{ftv}(\tau_{1i}) \# \overline{b}_c \\ w & \mathit{otherwise} \end{cases} \\ \Gamma, \{\overline{a}\overline{b}, \Delta, \theta\} \vdash^{\mathit{fold}} \overline{p_i :^{m_i} \psi(\tau_{1i})} \blacktriangleright K\end{array}}{\Gamma, \{\overline{a}, \Delta, \theta\} \vdash C\, \overline{p} :^w T\, \overline{\tau}_3 \blacktriangleright K} \text{ PCON-W}$$

$$\frac{\begin{array}{c}C :^r \forall \overline{b}. \overline{\tau}_1 \rightarrow T\, \overline{\tau}_2 \in \Gamma \quad \overline{b} \# \overline{a} \\ \theta(\tau) = T\, \overline{\tau}_3 \quad \psi \in \mathit{fmgu}(\overline{\tau}_3 \doteq \overline{\tau}_2) \\ \Gamma, \{\overline{a}\overline{b}, \Delta, \psi \cdot \theta\} \vdash^{\mathit{fold}} \overline{p_i :^r \tau_{1i}} \blacktriangleright K\end{array}}{\Gamma, \{\overline{a}, \Delta, \theta\} \vdash C\, \overline{p} :^r \tau \blacktriangleright K} \text{ PCON-R}$$

$$\frac{\begin{array}{c}\overline{\mathtt{b}} = \mathit{ftv}(\mathtt{tau}) - \mathit{dom}(\Gamma, \Delta) \\ \overline{b} \text{ distinct} \quad \overline{b} \# \mathit{dom}(\theta) \quad \overline{b} \# \mathit{bindings}(\Delta) \\ [\![\mathtt{tau}]\!]_{\Gamma, \Delta, \overline{\mathtt{b} \hookrightarrow b}} = \tau_s \quad \theta(\tau_s) = \theta^m(\tau) \\ \Gamma, \{\overline{a}, (\Delta, \overline{\mathtt{b} \hookrightarrow b}), \theta\} \vdash p :^r \tau_s \blacktriangleright K\end{array}}{\Gamma, \{\overline{a}, \Delta, \theta\} \vdash (p :: \mathtt{tau}) :^m \tau \blacktriangleright K} \text{ PANN}$$

$$\boxed{\Gamma, K_1 \vdash^{\mathit{fold}} \overline{p_i :^{m_i} \tau_i} \blacktriangleright K_2}$$

$$\frac{}{\Gamma, K \vdash^{\mathit{fold}} \cdot \blacktriangleright K} \text{ F-BASE} \qquad \frac{\begin{array}{c}\Gamma, K_1 \vdash p :^m \tau \blacktriangleright K_2 \\ \Gamma, K_2 \vdash^{\mathit{fold}} \overline{p_i :^{m_i} \tau_i} \blacktriangleright K_3\end{array}}{\Gamma, K_1 \vdash^{\mathit{fold}} (p :^m \tau), \overline{p_i :^{m_i} \tau_i} \blacktriangleright K_3} \text{ F-REC}$$

Figure 16: Source language pattern typing (simplified version)

7 Fully nested matching

In this section we present the language that contains fully nested patterns. The main typing and scrutinee inference judgements remain the same. The nested pattern matching typing judgements are given in Figure 16. This figure is in accordance to the judgement presented in the main paper. However, for the technical development it is slightly more convenient to work with the judgement presented in Figure 17 which *separates* the variables that take part in refinement (to think of as rigid type variables) and the varibles that

$$\boxed{\Gamma \vdash p \to t :^{\langle m_p, m_t \rangle} \tau_p \to \tau_t}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{(ftv(\Gamma,\tau_t,\tau_p)), \overline{e}\times\overline{c})} p :^{m_p} \tau_p \blacktriangleright \{\overline{e}\times\overline{c}, \Delta, \theta\} \\ ftv(\Gamma,\tau_p,\tau_t)\#\overline{e}\times\overline{c} \quad bindings(\Delta) \subseteq \overline{e}\times\overline{c} \\ \theta(\Gamma\cup\Delta) \vdash t :^{m_t} \theta^{m_t}(\tau_t)\end{array}}{\Gamma \vdash p \to t :^{\langle m_p, m_t\rangle} \tau_p \to \tau_t}\ \text{PAT}$$

$$\boxed{\Gamma, K_1 \vdash_X p :^m \tau \blacktriangleright K_2}$$

$$\frac{x \notin dom(\Delta)}{\Gamma, \{\overline{e}\times\overline{c}, \Delta, \theta\} \vdash_X x :^m \tau \blacktriangleright \{\overline{e}\times\overline{c}, \Delta, x :^m \tau, \theta\}}\ \text{PVAR}$$

$$\frac{\begin{array}{c}C :^r \forall\overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b}\#\overline{e}_1\times\overline{c}_1 \\ \overline{b}_c = \overline{b}\cap ftv(\overline{\tau}_2) \quad \overline{b}_e = \overline{b} - \overline{b}_c \quad \psi = [\overline{b_c \mapsto \upsilon}] \quad \overline{\tau}_3 = \psi(\overline{\tau}_2) \\ m_i = \begin{cases} r & ftv(\tau_{1i})\#\overline{b}_c \\ w & \text{otherwise}\end{cases} \\ \Gamma, \{\overline{e}_1\overline{b}_e \times \overline{c}_1\overline{b}_c, \Delta_1, \theta_1\} \vdash^{fold}_X \overline{p_i :^{m_i} \psi(\tau_{1i})} \blacktriangleright \{\overline{e}_n\times\overline{c}_n, \Delta_n, \theta_n\}\end{array}}{\Gamma, \{\overline{e}_1\times\overline{c}_1, \Delta_1, \theta_1\} \vdash_X C\ \overline{p} :^w T\ \overline{\tau}_3 \blacktriangleright \{\overline{e}_n\times\overline{c}_n, \Delta_n, \theta_n\}}\ \text{PCON-W}$$

$$\frac{\begin{array}{c}C :^r \forall\overline{b}.\overline{\tau}_1 \to T\ \overline{\tau}_2 \in \Gamma \quad \overline{b}\#\overline{e}_1\times\overline{c}_1 \\ \theta_1(\tau) = T\ \overline{\tau}_3 \\ \psi \in fmgu(\overline{\tau}_3 \doteq \overline{\tau}_2) \quad Y = (vars(\psi) - ftv(\overline{\tau}_3, \overline{\tau}_2))\#\overline{e}_1\overline{c}_1\overline{b}, \theta_1, \Delta_1, \Gamma, X \\ \Gamma, \{\overline{e}_1\overline{b}\times\overline{c}_1, \Delta_1, \psi\cdot\theta_1\} \vdash^{fold}_{X\cup Y} \overline{p_i :^r \tau_{1i}} \blacktriangleright \{\overline{e}_n\times\overline{c}_n, \Delta_n, \theta_n\}\end{array}}{\Gamma, \{\overline{e}_1\times\overline{c}_1, \Delta_1, \theta_1\} \vdash_X C\ \overline{p} :^r \tau \blacktriangleright \{\overline{e}_n\times\overline{c}_n, \Delta_n, \theta_n\}}\ \text{PCON-R}$$

$$\frac{\begin{array}{c}\overline{\mathtt{b}} = ftv(\mathtt{tau}) - dom(\Gamma,\Delta) \\ \overline{b}\ \text{distinct} \quad \overline{b}\#dom(\theta_1) \quad \overline{b}\#bindings(\Delta_1) \\ [\![\mathtt{tau}]\!]_{\Gamma,\Delta_1,\overline{\mathtt{b}\hookrightarrow b}} = \tau_s \quad \theta_1(\tau_s) = \theta_1^m(\tau) \\ \Gamma, \{\overline{e}_1\times\overline{c}_1, (\Delta_1, \overline{\mathtt{b}\hookrightarrow b}), \theta_1\} \vdash p :^r \tau_s \blacktriangleright \{\overline{e}_2\times\overline{c}_2, \Delta_2, \theta_2\}\end{array}}{\Gamma, \{\overline{e}_1\times\overline{c}_1, \Delta_1, \theta_1\} \vdash (p :: \mathtt{tau}) :^m \tau \blacktriangleright \{\overline{e}_2\times\overline{c}_2, \Delta_2, \theta_2\}}\ \text{PANN}$$

$$\boxed{\Gamma, K_1 \vdash^{fold}_X \overline{p_i :^{m_i} \tau_i} \blacktriangleright K_2}$$

$$\frac{}{\Gamma, \{\overline{e}\times\overline{c}, \Delta, \theta\} \vdash^{fold}_X \cdot \blacktriangleright \{\overline{e}\times\overline{c}, \Delta, \theta\}}\ \text{F-BASE}$$

$$\frac{\begin{array}{c}\Gamma, \{\overline{e}_1\times\overline{c}_1, \Delta_1, \theta_1\} \vdash_X p :^m \tau \blacktriangleright \{\overline{e}_2\times\overline{c}_2, \Delta_2, \theta_2\} \\ \Gamma, \{\overline{e}_2\times\overline{c}_2, \Delta_2, \theta_2\} \vdash^{fold}_X \overline{p_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e}_n\times\overline{c}_n, \Delta_n, \theta_n\}\end{array}}{\Gamma, \{\overline{e}_1\times\overline{c}_1, \Delta_1, \theta_1\} \vdash^{fold}_X (p :^m \tau), \overline{p_i :^{m_i} \tau_i} \blacktriangleright \{\overline{e}_n\times\overline{c}_n, \Delta_n, \theta_n\}}\ \text{F-REC}$$

Figure 17: Source language pattern typing

take part in unification (to think of as unification variables). The syntax of the triples is slightly different as well. Whenever we have in the main paper a $\{\overline{a}, \Delta, \theta\}$ we now have a $\{\overline{e} \times \overline{c}, \Delta, \theta\}$, with $\overline{a} = \overline{e} \times \overline{c}$. Moreover notice that the freshness of *fmgu*s is made explicit in rule PCON-R in Figure 17. It is straightforward to see that (modulo the implicit freshness assumption) the two relations type the same patterns.

We do not present the proofs corresponding to this nested pattern language, as the threading of substitutions involves tedious generalizations of the statements of most theorems, but the general idea of the system (and the most important problems for completeness and soundness) appear already in the non-nested language presented above. The only part where complications may be expected is the treatment of annotated patterns and introduction of scoped type variables. As far as inference is concerned we present the corresponding modifications to the algorithm in Figure 18.

$$\boxed{(\mathcal{A}_0,\delta_0) \succ \Gamma \vdash p \to t :^{\langle m_p, m_t\rangle} \tau_p \to \tau_t \succ (\mathcal{A}_1,\delta_1)}$$

$$\frac{\begin{array}{c} (\mathcal{A}_0,\delta_0)\succ\Gamma,\{\emptyset,\cdot,\emptyset\}\vdash p :^{m_p} \tau_p \blacktriangleright \{\overline{a},\Delta,\theta\}\succ(\mathcal{A}_1,\delta_1)\\ (\mathcal{A}_1,\delta_1)\succ\theta(\Gamma\cup\Delta)\vdash t :^{m_t} \theta^{m_t}(\tau_t)\succ(\mathcal{A}_2,\delta_2)\\ \mathit{bindings}(\Delta)\subseteq\overline{a} \quad \mathit{ftv}(\delta_2\Gamma,\delta_2\tau_p,\delta_2\tau_t)\#\overline{a}\end{array}}{(\mathcal{A}_0,\delta_0)\succ\Gamma\vdash p\to t :^{\langle m_p,m_t\rangle}\tau_p\to\tau_t\succ(\mathcal{A}_2,\delta_2)}\ \textsc{APat-Refine}$$

$$\boxed{(\mathcal{A}_0,\delta_0) \succ \Gamma, K_1 \vdash p :^m \tau \blacktriangleright K_2 \succ (\mathcal{A}_1,\delta_1)}$$

$$\frac{x \notin \mathit{dom}(\Delta)}{(\mathcal{A}_0,\delta_0)\succ\Gamma,\{\overline{a},\Delta,\theta\}\vdash x :^m \tau \blacktriangleright \{\overline{a},\Delta,x:^m\tau,\theta\}\succ(\mathcal{A}_0,\delta_0)}\ \textsc{APVar}$$

$$\frac{\begin{array}{c} \overline{\mathtt{b}} = \mathit{ftv}(\mathtt{tau}) - \mathit{dom}(\Gamma\cup\Delta)\\ \mathcal{A}_0 \succ \Gamma\cup\Delta, \overline{\mathtt{b}\hookrightarrow\beta} \vdash \mathtt{tau} \Rightarrow \tau_1 \,;\emptyset \succ \mathcal{A}_1\\ \delta_0\succ\theta(\tau_1)\doteq\theta^m(\tau_2)\succ\delta_1\\ \delta_1\overline{\beta}=\overline{c} \quad \overline{c}\#\mathit{dom}(\theta) \quad \overline{c}\text{ distinct} \quad \overline{c}\#\mathit{bindings}(\Delta)\\ (\mathcal{A}_1,\delta_1)\succ\Gamma,\{\overline{a},\Delta,\overline{\mathtt{b}\hookrightarrow c},\theta\}\vdash p :^r [\overline{\beta\mapsto c}]\tau_1 \blacktriangleright K \succ(\mathcal{A}_2,\delta_2)\end{array}}{(\mathcal{A}_0\overline{\beta},\delta_0)\succ\Gamma,\{\overline{a},\Delta,\theta\}\vdash (p\,\mathtt{::}\,\mathtt{tau}) :^m \tau_2 \blacktriangleright K\succ(\mathcal{A}_2,\delta_2)}\ \textsc{APAnn}$$

$$\frac{\begin{array}{c} C :^r \forall\overline{b}.\overline{\tau_1}\to T\,\overline{\tau_2}\in\Gamma \quad \overline{b_c}=\overline{b}\cap \mathit{ftv}(\overline{\tau_2}) \quad \overline{b_e}=\overline{b} - \overline{b_c}\\ \psi_c = [\overline{b_c\mapsto\beta}] \quad \psi_e=[\overline{b_e\mapsto c}] \quad \delta_0\succ\tau\doteq T\,\psi_c(\overline{\tau_2})\succ\delta_1\\ m_i = \begin{cases} r & \mathit{ftv}(\tau_{1i})\#\overline{b}_c\\ w & \mathit{otherwise}\end{cases}\\ (\mathcal{A}_0,\delta_1)\succ\Gamma,\{\overline{ac},\Delta,\theta\}\vdash^{\mathit{fold}} \overline{p_i :^{m_i} \psi_c(\psi_e(\tau_{1i}))} \blacktriangleright K\succ(\mathcal{A}_1,\delta_2)\end{array}}{(\mathcal{A}_0\overline{c}\overline{\beta},\delta_0)\succ\Gamma,\{\overline{a},\Delta,\theta\}\vdash C\,\overline{p} :^w \tau\blacktriangleright K\succ(\mathcal{A}_1,\delta_2)}\ \textsc{APCon-W}$$

$$\frac{\begin{array}{c} C :^r \forall\overline{b}.\overline{\tau_1}\to T\,\overline{\tau_2}\in\Gamma \quad \theta(\tau)=T\,\overline{\tau_3} \quad \mathcal{A}_0\succ\psi=\mathtt{fmgu}(\overline{\tau_3}\doteq[\overline{b\mapsto c}]\overline{\tau_2})\succ\mathcal{A}_1\\ (\mathcal{A}_1,\delta_0)\succ\Gamma,\{\overline{ac},\Delta,\psi\cdot\theta\}\vdash^{\mathit{fold}}\overline{p_i :^r [\overline{b\mapsto c}]\tau_{1i}}\blacktriangleright K\succ(\mathcal{A}_2,\delta_1)\end{array}}{(\mathcal{A}_0\overline{c},\delta_0)\succ\Gamma,\{\overline{a},\Delta,\theta\}\vdash C\,\overline{p} :^r \tau\blacktriangleright K\succ(\mathcal{A}_2,\delta_1)}\ \textsc{APCon-R}$$

$$\boxed{(\mathcal{A}_0,\delta_0) \succ \Gamma, K_1 \vdash^{\mathit{fold}} \overline{p_i :^{m_i} \tau_i} \blacktriangleright K_2 \succ (\mathcal{A}_1,\delta_1)}$$

$$\frac{}{(\mathcal{A}_0,\delta_0)\succ\Gamma,K\vdash^{\mathit{fold}}\cdot\blacktriangleright K\succ(\mathcal{A}_0,\delta_0)}\ \textsc{AF-Base}$$

$$\frac{\begin{array}{c} (\mathcal{A}_0,\delta_0)\succ\Gamma,K_1\vdash p :^m \tau\blacktriangleright K_2\succ(\mathcal{A}_1,\delta_1)\\ (\mathcal{A}_1,\delta_1)\succ\Gamma,K_2\vdash^{\mathit{fold}}\overline{p_i :^{m_i}\tau_i}\blacktriangleright K_3\succ(\mathcal{A}_n,\delta_n)\end{array}}{(\mathcal{A}_0,\delta_0)\succ\Gamma,K_1\vdash^{\mathit{fold}}(p :^m \tau),\overline{p_i :^{m_i}\tau_i}\blacktriangleright K_3\succ(\mathcal{A}_n,\delta_n)}\ \textsc{AF-Rec}$$

Figure 18: Algorithmic pattern typing

References

[PVWW05] Simon Peyton Jones, Dimitrios Vytiniotis, Stephanie Weirich, and Geoffrey Washburn. Simple unification-based type inference for GADTs. April 2005.

[VWP05] Dimitrios Vytiniotis, Stephanie Weirich, and Simon Peyton Jones. Practical type inference for arbitrary-rank types, Technical Appendix. Technical Report MS-CIS-05-14, University of Pennsylvania, July 2005.