An informal guide to picking fresh names

François Pottier

January 3, 2008

1 Introduction

These draft course notes provide the proofs of a few key lemmas concerning Damas and Milner's type system, and attempt to explain, clearly but informally, how to reason "up to α-conversion", that is, where and why it is possible to "pick sufficiently fresh names".

Figure 1 presents Damas and Milner's type system [3].

2 Types, type schemes, type environments

The syntax of types, type schemes, and type environments is as follows:

$$\begin{array}{l} \tau ::= \alpha \mid \tau \to \tau \\ \sigma ::= \forall \bar{\alpha}.\tau \\ \Gamma ::= \emptyset \mid (\Gamma; x : \tau) \end{array}$$

3 Renamings and substitutions

Definition 3.1 *A* renaming ρ *is a total, bijective mapping of type variables to type variables whose domain is finite. The* domain *of* ρ *is the set of the type variables* α *such that* $\rho(\alpha) \neq \alpha$*. The* support *of* ρ *is its domain.* $\diamond$

A renaming is also viewed as a bijective mapping of types to types, of type schemes to type schemes, and of type environments to type environments, as follows:

$$\begin{array}{rl} \rho(\tau_1 \to \tau_2) &= \rho(\tau_1) \to \rho(\tau_2) \\ \rho(\forall \bar{\alpha}.\tau) &= \forall \rho(\bar{\alpha}).\rho(\tau) \\ \rho(\emptyset) &= \emptyset \\ \rho(\Gamma; x : \sigma) &= \rho(\Gamma); x : \rho(\sigma) \end{array}$$

$$\textsc{dm-Var}\ \frac{\Gamma(x) = \sigma}{\Gamma \vdash x : \sigma} \qquad \textsc{dm-Abs}\ \frac{\Gamma; x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x.e : \tau \to \tau'} \qquad \textsc{dm-App}\ \frac{\Gamma \vdash e_1 : \tau \to \tau' \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1\ e_2 : \tau'}$$

$$\textsc{dm-Let}\ \frac{\Gamma \vdash e_1 : \sigma \qquad \Gamma; x : \sigma \vdash e_2 : \tau}{\Gamma \vdash \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 : \tau} \qquad \textsc{dm-Gen}\ \frac{\Gamma \vdash e : \tau \qquad \bar{\alpha} \# \Gamma}{\Gamma \vdash e : \forall \bar{\alpha}.\tau} \qquad \textsc{dm-Inst}\ \frac{\Gamma \vdash e : \forall \bar{\alpha}.\tau}{\Gamma \vdash e : [\bar{\alpha} \mapsto \vec{\tau}]\tau}$$

Figure 1: Damas and Milner's type system

Lemma 3.2 *Type derivations are preserved under renamings. That is, for every derivation of* $\Gamma \vdash e : \sigma$*, there exists an isomorphic derivation of* $\rho(\Gamma) \vdash e : \rho(\sigma)$*.* $\diamond$

Proof. A meta-theoretic argument suffices here. None of the typing rules that define Damas and Milner's type system is sensitive to the choice of type variable names, so the inductive predicate that these rules define must be preserved by renamings. This is what Gabbay and Pitts refer to as *equivariance* [1]. $\square$

Definition 3.3 *A* substitution φ *is a total mapping of type variables to types whose domain is finite. The* domain *of* φ *is the set of the type variables* α *such that* $\varphi(\alpha) \neq \alpha$*. The* codomain *of* φ *is the set of the type variables that appear free in the image of its domain. The* support *of* φ *is the union of its domain and codomain.* $\diamond$

As announced above, a type variable α is considered *fresh for* φ, which I write $\alpha \# \varphi$, if and only if α is not in the support of φ.

A substitution is also viewed as a mapping of types to types, of type schemes to type schemes, and of type environments to type environments, as follows:

$$\begin{aligned} \varphi(\tau_1 \to \tau_2) &= \varphi(\tau_1) \to \varphi(\tau_2) \\ \varphi(\forall \bar{\alpha}.\tau) &= \forall \bar{\alpha}.\varphi(\tau) \qquad \text{if } \bar{\alpha} \# \varphi \\ \varphi(\emptyset) &= \emptyset \\ \varphi(\Gamma; x : \sigma) &= \varphi(\Gamma); x : \varphi(\sigma) \end{aligned}$$

The following lemma states that, if a type variable appears in $\varphi(\alpha)$, then it must appear in φ or α.

Lemma 3.4 $\bar{\alpha} \# \varphi$ *and* $\bar{\alpha} \# \alpha$ *imply* $\bar{\alpha} \# \varphi(\alpha)$*.* $\diamond$

Proof. If α is in the domain of φ, then, by definition, the free type variables of $\varphi(\alpha)$ are in the codomain of φ, so the first hypothesis implies the goal. If, on the other hand, α is not in the domain of φ, then $\varphi(\alpha)$ is α, so the second hypothesis is the goal. $\square$

This lemma can be generalized, by replacing α with a type τ, a type scheme σ, or a type environment Γ. It is a general property that, if a type variable is fresh for a number of objects, then it is fresh for the result of applying an equivariant operation (here, function application) to these objects.

4 The type substitution lemma

Lemma 4.1 *Type derivations are preserved under substitutions. That is, for every derivation of* $\Gamma \vdash e : \sigma$*, there exists an isomorphic derivation of* $\varphi(\Gamma) \vdash e : \varphi(\sigma)$*.* $\diamond$

The proof of this lemma is by structural induction over the derivation of the judgement $\Gamma \vdash e : \tau$. In each case, one *inverts* the rule, thus obtaining access to its premises; then, one *invokes* the induction hypothesis, applying it to one of the premises; finally, one *builds* a new instance of the rule. The fact that one builds a new instance of the *same* rule clearly shows that the lemma is structure-preserving. The proof is very straightforward in all cases, except in cases DM-GEN and DM-INST, where one must be careful to "pick sufficiently fresh names". These two cases (and only these two) are detailed below. In each of these two cases, the reasoning is in two steps. First, one examines the ideal situation where a suitable freshness hypothesis is available. Then, one shows that, in the general situation, a renaming step allows obtaining such a hypothesis. In research papers, the second step is usually omitted – the reader is trusted to convince herself that the freshness hypothesis can indeed be assumed "without loss of generality".

Proof. I consider only the two cases announced above.

∘ *Case* DM-GEN. The hypothesis is $\Gamma \vdash e : \forall \bar{\alpha}.\tau$, and is obtained as the conclusion of an instance of DM-GEN whose premises are $\Gamma \vdash e : \tau$ and $\bar{\alpha} \# \Gamma$. The goal is $\varphi(\Gamma) \vdash e : \varphi(\forall \bar{\alpha}.\tau)$.

Step 1. Let us assume the freshness hypothesis $\bar{\alpha} \# \varphi$. Invoking the induction hypothesis yields $\varphi(\Gamma) \vdash e : \varphi(\tau)$. Besides, the freshness hypothesis $\bar{\alpha} \# \varphi$, together with the premise $\bar{\alpha} \# \Gamma$, imply $\bar{\alpha} \# \varphi(\Gamma)$ (Lemma 3.4). Thus, we can build a new instance of DM-GEN, whose conclusion is $\varphi(\Gamma) \vdash e : \forall\bar{\alpha}.\varphi(\tau)$. Thanks to the freshness hypothesis, this is $\varphi(\Gamma) \vdash e : \varphi(\forall\bar{\alpha}.\tau)$—that is, the goal.

Step 2. Now, what when the freshness hypothesis is not met? Pick a vector of distinct type variables $\bar{\beta}$, whose length is the same as that of $\bar{\alpha}$, such that $\bar{\beta}$ is fresh for the judgement $\Gamma \vdash e : \forall\bar{\alpha}.\tau$ (that is, $\bar{\beta} \# \Gamma$ and $\bar{\beta} \# \forall\bar{\alpha}.\tau$) and $\bar{\beta}$ satisfies the desired freshness hypothesis (that is, $\bar{\beta} \# \varphi$). Let ρ be the renaming $(\bar{\alpha}\,\bar{\beta})$, that is, the bijective mapping that swaps $\bar{\alpha}$ and $\bar{\beta}$ and leaves all other type variables unaffected.

Now, notice that also $\bar{\alpha}$ is fresh for the judgement $\Gamma \vdash e : \forall\bar{\alpha}.\tau$ (that is, both $\bar{\alpha} \# \Gamma$ and $\bar{\alpha} \# \forall\bar{\alpha}.\tau$ hold, the former by hypothesis and the latter by construction). Since both $\bar{\alpha}$ and $\bar{\beta}$ are fresh for this judgement, there follows, by definition of ρ, that ρ preserves this judgement (that is, $\rho(\Gamma) = \Gamma$ and $\rho(\forall\bar{\alpha}.\tau) = \forall\bar{\alpha}.\tau$).

By Lemma 3.2, applying ρ to the derivation of this judgement yields another valid derivation with the same structure, and, by the above remarks, it is a derivation of the same judgement. By construction, this new derivation ends with an instance of DM-GEN whose premises are $\Gamma \vdash e : \rho(\tau)$ and $\bar{\beta} \# \Gamma$. This instance of DM-GEN satisfies the freshness hypothesis that was assumed in step 1.

∘ *Case* DM-INST. The hypothesis is $\Gamma \vdash e : [\bar{\alpha} \mapsto \vec{\tau}]\tau$, and is obtained as the conclusion of an instance of DM-INST whose premise is $\Gamma \vdash e : \forall\bar{\alpha}.\tau$. The goal is $\varphi(\Gamma) \vdash e : \varphi([\bar{\alpha} \mapsto \vec{\tau}]\tau)$.

Step 1. Let us assume the freshness hypothesis $\bar{\alpha} \# \varphi$. Invoking the induction hypothesis yields $\varphi(\Gamma) \vdash e : \varphi(\forall\bar{\alpha}.\tau)$, which, by the freshness hypothesis, can be written $\varphi(\Gamma) \vdash e : \forall\bar{\alpha}.\varphi(\tau)$. Let us build a new instance of DM-INST, whose conclusion is $\varphi(\Gamma) \vdash e : [\bar{\alpha} \mapsto \varphi(\vec{\tau})]\varphi(\tau)$. I claim that this is the goal. To establish this claim, it suffices to check that the substitutions $\varphi_1 = \varphi \circ [\bar{\alpha} \mapsto \vec{\tau}]$ and $\varphi_2 = [\bar{\alpha} \mapsto \varphi(\vec{\tau})] \circ \varphi$ coincide. This is done by applying both substitutions to an arbitrary variable α. I distinguish two sub-cases.

Sub-case $\alpha \in \bar{\alpha}$. For some index i, α is α_i, the i-th element of the vector $\bar{\alpha}$. Then, $\varphi_1(\alpha)$ is $\varphi(\tau_i)$, where τ_i is the i-th element of the vector $\vec{\tau}$. Besides, $\alpha \in \bar{\alpha}$ and $\bar{\alpha} \# \varphi$ imply $\alpha \# \varphi$, so α is not in the domain of φ, so $\varphi(\alpha)$ is α. There follows that $\varphi_2(\alpha)$ is also $\varphi(\tau_i)$.

Sub-case $\alpha \notin \bar{\alpha}$. Then, $\varphi_1(\alpha)$ is $\varphi(\alpha)$. Besides, $\bar{\alpha} \# \varphi$ and $\bar{\alpha} \# \alpha$ imply $\bar{\alpha} \# \varphi(\alpha)$, which implies that $\varphi_2(\alpha)$ is $\varphi(\alpha)$.

Step 2. Now, what when the freshness hypothesis is not met? Pick a vector of distinct type variables $\bar{\beta}$, whose length is the same as that of $\bar{\alpha}$, such that $\bar{\beta}$ is fresh for $\forall\bar{\alpha}.\tau$ and $\bar{\beta}$ satisfies the desired freshness hypothesis (that is, $\bar{\beta} \# \varphi$). Let ρ be the renaming $(\bar{\alpha}\,\bar{\beta})$.

Because both $\bar{\alpha}$ and $\bar{\beta}$ are fresh for the type scheme $\forall\bar{\alpha}.\tau$, ρ is fresh for $\forall\bar{\alpha}.\tau$ as well. This implies $\rho(\forall\bar{\alpha}.\tau) = \forall\bar{\alpha}.\tau$, which can be written $\forall\bar{\beta}.\rho(\tau) = \forall\bar{\alpha}.\tau$. In short, the bound variables of the type scheme can be changed from $\bar{\alpha}$ to $\bar{\beta}$ via an α-conversion step.

Furthermore, I claim that the type $[\bar{\alpha} \mapsto \vec{\tau}]\tau$ can also be written $[\bar{\beta} \mapsto \vec{\tau}]\rho(\tau)$. To prove this, it suffices to check that the substitutions $[\bar{\alpha} \mapsto \vec{\tau}]$ and $[\bar{\beta} \mapsto \vec{\tau}] \circ \rho$ coincide when applied to a type variable α that appears free in τ. As above, I distinguish two sub-cases. If, for some index i, α is α_i, the i-th element of the vector $\bar{\alpha}$, then it is clear that both substitutions map α to τ_i, the i-th element of the vector $\vec{\tau}$. If, on the other hand, α is not in $\bar{\alpha}$, then α appears free in $\forall\bar{\alpha}.\tau$, which, because $\bar{\beta}$ is fresh for $\forall\bar{\alpha}.\tau$, means that α is not in $\bar{\beta}$. It is then clear that both substitutions map α to itself.

By these two remarks, our instance of DM-INST has premise $\Gamma \vdash e : \forall\bar{\beta}.\rho(\tau)$ and conclusion $\Gamma \vdash e : [\bar{\beta} \mapsto \vec{\tau}]\rho(\tau)$. When viewed in this manner, it satisfies the freshness hypothesis that was assumed in step 1. □

It is somewhat discomforting that the renaming arguments (the "step 2"s), when written out in full, like above, are quite complex, yet are usually omitted – only the central arguments (the "step 1"s) are written on paper. Expert readers are usually able to form an intuition for which freshness hypotheses are reasonable and which are not. In case DM-GEN, the intuitive reason why $\bar{\alpha}$ can be picked fresh is that it does not appear free in the conclusion of the rule, so that a renaming of the entire derivation does not affect the conclusion. In case DM-INST, the intuitive reason is that $\bar{\alpha}$ is mute both in the premise (where it is universally bound) and in the conclusion (where it is substituted out), so that a local α-conversion step suffices – no global renaming of the derivation is even required. In practice, it

is useful for researchers to form this kind of intuition. Yet, mistakes remain possible, so that, ideally, reasoning up to freshness hypotheses should be machine-checked. See, for instance, Pitts [2] for a rigorous development of recursion and induction principles up to freshness hypotheses.

References

[1] Murdoch J. Gabbay and Andrew M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13(3–5):341–363, July 2002.

[2] Andrew M. Pitts. Alpha-structural recursion and induction. *Journal of the ACM*, 53:459–506, 2006.

[3] François Pottier and Didier Rémy. The essence of ML type inference. In Benjamin C. Pierce, editor, *Advanced Topics in Types and Programming Languages*, chapter 10, pages 389–489. MIT Press, 2005.