4.4 Unification with Parameters

The generalization of the algorithm above to account for universal quantifiers and parameters is not completely straightforward. The difficulty is that $\forall x.\ \exists y.\ y \doteq x$ is valid, while $\exists y.\ \forall x.\ y \doteq x$ is not. In unification logic, the fact that the second cannot be derived is due to the parameter restriction.

$$\dfrac{\dfrac{\dfrac{}{\models a \doteq a}\doteq\mathrm{I}}{\models \forall x.\ a \doteq x}\forall\mathrm{I}^a??}{\models \exists y.\ \forall x.\ y \doteq x}\exists\mathrm{I}$$

In this derivation, the application of $\forall\mathrm{I}^a$ is incorrect. However, if we had a way to postpone choosing the instantiation for y, say, by supplying an existential variable instead, then the situation is far less clear.

$$\dfrac{\dfrac{\dfrac{\text{“}a/Y\text{”}??}{\models Y \doteq a}\doteq\mathrm{I}}{\models \forall x.\ Y \doteq x}\forall\mathrm{I}^a??}{\models \exists y.\ \forall x.\ y \doteq x}\exists\mathrm{I}$$

In this derivation, it is the substitution of a for Y which will invalidate the derivation at the $\forall\mathrm{I}^a$ rule application. Up to that point we could not really fail. Written in our transformation notation:

$$\begin{array}{ll} & \exists y.\ \forall x.\ y \doteq x \\ \mapsto & \forall x.\ Y \doteq x \\ \mapsto & Y \doteq a \\ \mapsto^{??} & \cdot \end{array}$$

From this very simple example it seems clear that we need to prohibit final step: Y may not be instantiated with a term that mentions parameter a. There are two approaches to encoding this restriction. More or less standard in theorem proving is *Skolemization* which we pursue in Exercise 4.3. The dual solution notes for each existential variable which parameters may occur in its substitution term. In the example above, Y was introduced at a point where a did not yet occur, so the substitution of a for Y should be rejected.

In order to describe this concisely, we add a *parameter context* Ψ to the judgment which lists distinct parameters.

$$\textit{Parameter Context} \quad \Psi \quad ::= \quad \cdot \mid \Psi, a$$

We annotate each judgment with the parameter context and introduce the new judgment "*t is closed with respect to* Ψ", written as $\Psi \models t\ \textit{term}$. It is defined by the following rules.

$$\dfrac{}{\Psi_1, a, \Psi_2 \vdash a\ \textit{term}}\mathrm{parm} \qquad \dfrac{\Psi \vdash t_1\ \textit{term} \quad \cdots \quad \Psi \vdash t_n\ \textit{term}}{\Psi \vdash f(t_1, \ldots, t_n)\ \textit{term}}\mathrm{root}$$

We modify the validity judgment for unification formulas to guarantee this condition.

$$\frac{\Psi \vdash t\ \mathit{term} \qquad\qquad \Psi \models [t/x]F}{\Psi \models \exists x.\ F}\exists\mathrm{I} \qquad \frac{\Psi, a \models [a/x]F}{\Psi \models \forall x.\ F}\forall\mathrm{I}^a$$

Now the state of the unification algorithm (that is, the current set of constraints) must record the parameter context. We write this as $\Psi \rhd C$. Ψ is simply carried along from left to right in most transformations.

$$\begin{array}{lcll}
(\Psi \rhd F_1 \wedge F_2, C) & \mapsto & (\Psi \rhd F_1, F_2, C) & \\
(\Psi \rhd \top, C) & \mapsto & (\Psi \rhd C) & \\
(\Psi \rhd f(t_1, \ldots, t_n) \doteq f(s_1 \ldots, s_n), C) & \mapsto & (\Psi \rhd t_1 \doteq s_1, \ldots, t_n \doteq s_n, C) & \\
(\Psi \rhd f(t_1, \ldots, t_n) \doteq g(s_1 \ldots, s_n), C) & \mapsto & (\Psi \rhd \#) & \text{where } f \neq g \\
(\Psi \rhd a \doteq a, C) & \mapsto & (\Psi \rhd C) & \\
(\Psi \rhd a \doteq b, C) & \mapsto & (\Psi \rhd \#) & \text{where } a \neq b \\
(\Psi \rhd a \doteq f(t_1, \ldots, t_n)) & \mapsto & (\Psi \rhd \#) & \\
(\Psi \rhd f(t_1, \ldots, t_n) \doteq a) & \mapsto & (\Psi \rhd \#) &
\end{array}$$

The notion of an existential variable must now be generalized to track the set of parameters its substituend may depend on. We write X_Δ for a unification variable X that may depend on all the parameters in Δ, but no others. All occurrences of a variable X must be annotated with the same Δ—we think of Δ as an intrinsic property of X.

$$\begin{array}{lcll}
(\Psi \rhd \forall x.\ F, C) & \mapsto & (\Psi, a \rhd [a/x]F, C) & \text{where } a \text{ not in } \Psi,\ F, \text{ or } C \\
(\Psi \rhd \exists x.\ F, C) & \mapsto & (\Psi \rhd [X_\Psi/x]F, C) & \text{where } X \text{ not free in } F \text{ or } C
\end{array}$$

An equation $X_\Psi \doteq t$ could now be solved immediately, if all parameters of t are contained in Ψ and X does not occur in t. A first attempt at such a rule would be

$$(\Psi \rhd X_\Delta \doteq t, C) \quad \mapsto \quad (\Psi \rhd [t/X]C) \quad \text{where } \Delta \vdash t\ \mathit{term} \text{ and } X \text{ not free in } t$$

However, in general t will not be closed so we cannot prove that $\Delta \vdash t$ *term*. For example, consider the constraint

$$a \rhd X_{\cdot} \doteq f(Y_a) \wedge Y_a \doteq a$$

where X cannot depend on any parameters and Y can depend on a. This should have no solution, since $X_{\cdot}$ would have to be equal to $f(a)$, which is not permissible. On the other hand,

$$a \rhd X_{\cdot} \doteq f(Y_a) \wedge Y_a \doteq c$$

for a constant c has a solution where Y_a is c and $X_{\cdot}$ is $f(c)$. So when we process an equation $X_\Delta = t$ we need to restrict any variable in t so it can depend only on the parameters in Δ. In the example above, we would substitute $Y'_{\cdot}$ for Y_a.

In order to describe this restriction, we introduce a new form of constraints which expresses the judgment $\Delta \vdash t$ *term* in the presence of unification variables.

We write it as $t \mid_\Delta$, thinking of it as the restriction of t to Δ. It is implemented by the following transformations.

$$\begin{array}{llll}
(\Psi \rhd f(t_1, \ldots, t_n) \mid_\Delta, C) & \mapsto & (\Psi \rhd t_1 \mid_\Delta, \ldots, t_n \mid_\Delta, C) & \\
(\Psi \rhd a \mid_\Delta, C) & \mapsto & (\Psi \rhd C) & \text{if } a \in \Delta \\
(\Psi \rhd a \mid_\Delta, C) & \mapsto & (\Psi \rhd \#) & \text{if } a \notin \Delta \\
(\Psi \rhd Y_{\Delta'} \mid_\Delta, C) & \mapsto & (\Psi \rhd [Y_{\Delta' \cap \Delta}/Y]C) &
\end{array}$$

the collection of the above four rules implement a process called *pruning*. Now we can finally write down the correct rule for existential variables.

$$(\Psi \rhd X_\Delta \doteq t, C) \quad \mapsto \quad (\Psi \rhd t \mid_\Delta, [t/X]C) \quad \text{provided } X \text{ not free in } t$$

From an implementation point of view, it makes sense to first solve $t \mid_\Delta$ before substitution t for X. In fact, it is probably beneficial to combine it with the occurs-check to the term t need only be traversed once.

The soundness and completeness theorems from above extend to the problem with parameters, but become more difficult. The principal new notion we need is an *admissible substitution* θ which has the property that for every existential variable X_Δ we have $\Delta \vdash X[\theta]$ *term* (see Exercise 4.4).

The ML implementation takes advantage of the fact that whenever a variable must be restricted, one of the two contexts is a prefix of the other. This is because every equation in a formula F lies beneath a path of possibly alternating quantifiers, a so-called *mixed quantifier prefix*. When we apply the rules above algorithmically, we instantiate each existentially quantified variable with a new free existential variable which depends on all parameters which were introduced for the universally quantified variables to its left. Clearly, then, for any two variables in the same equation, one context is a prefix of the other. Our ML implementation does take advantage of this observation by simplifying the intersection operation.

We can take this optimization a step further and only record with an integer (a kind of time stamp), which parameters an existential variable may depend on. This improves the efficiency of the algorithm even further, since we only need to calculate the minimum of two integers instead of intersecting two contexts during restriction. In the ML code for this class, we did not optimize to this extent.

4.5 Exercises

Exercise 4.1 Give an alternative proof of the inversion properties (Theorem 4.1) which does not use induction, but instead relies on admissibility of cut in the sequent calculus (Theorem 3.11).

Exercise 4.2 Formulate one or several cut rules directly on inversion sequents as presented in Section 4.1 and prove that they are admissible. Does this simplify the development of the completeness result for inversion proofs? Show how admissibility might be used, or illustrate why it is not much help.

Exercise 4.3 An alternative to indexing unification variables with the parameters they may depend on is *Skolemization*. Instead of changing the notion of unification variable, we change the notion of parameter, replacing it by a so-called *Skolem function*. The two quantifier rules become

$$\begin{array}{lcll}
\forall x.\, F, C & \mapsto & [f(X_1, \ldots, X_n)/x]F, C & \text{where } f \text{ not in } F\text{, or } C\text{, and } X_1, \ldots, X_n \\
 & & & \text{are all free unification variables in } F \\
\exists x.\, F, C & \mapsto & [X/x]F, C & \text{where } X \text{ not free in } F \text{ or } C
\end{array}$$

Now, incorrect dependencies are avoided due to the occurs-check. Reconsider our simple example:

$$\begin{array}{ll}
 & \exists y.\, \forall x.\, y \doteq x \\
\mapsto & \forall x.\, Y \doteq x \\
\mapsto & Y \doteq f(Y) \\
\mapsto & \#
\end{array}$$

Skolemization is attractive because it allows us to use a simpler algorithm for unification. Moreover, in some logics such as classical logic it can be applied statically, before we ever attempt to prove the proposition, completely eliminating parameters from consideration. On the other hand, Skolemization is unsound in some higher-order logics. Also, it is more difficult to recover a proof of proposition if we Skolemize during search.

Prove the correctness of the unification algorithm for the full unification logic (including universal quantifiers) which employs Skolemization.

Exercise 4.4 Extend the proofs of termination and preservation of satisfiability from the purely existential case in Section 4.3 to allow for the presence of parameters as sketched in Section 4.4. An important concept will likely be that of *admissible substitution* θ which has the property that for every existential variable X_Δ we have $\Delta \vdash X[\theta]$ *term*. You should be careful to make a precise connection between the constraint $t\ |_\Delta$ and the judgment $\Delta \vdash t$ *term* (where the latter is not defined for unification variables).