Corrigendum: Decidable Bounded Quantification

Giuseppe Castagna* Benjamin C. Pierce †

At POPL 1994, we presented a paper on "Decidable Bounded Quantification" [CP94]. Sadly, subsequent discussion on the *Types* mailing list revealed that, while strictly speaking our paper did not contain any errors, the central idea was fundamentally flawed. We summarize the situation here to avert misunderstanding.

System $F_{\leq}$, the second-order λ-calculus of bounded quantification, was invented by Cardelli and Wegner (see [CP94] for background and references). The standard formulation, due to Curien and Ghelli, includes the following rule for comparing polymorphic types:

$$(\forall\text{-orig})\quad \frac{\Gamma \vdash T_1 \leq S_1 \qquad \Gamma, (X{\leq}T_1) \vdash S_2 \leq T_2}{\Gamma \vdash \forall(X{\leq}S_1)S_2 \leq \forall(X{\leq}T_1)T_2}$$

This rule turns out to be responsible for the loss of numerous desirable syntactic properties, including the decidability of the subtyping relation. In [CP94], we proposed to replace it by the following simpler rule

$$(\forall\text{-top})\quad \frac{\Gamma \vdash T_1 \leq S_1 \qquad \Gamma, (X{\leq}\mathsf{Top}) \vdash S_2 \leq T_2}{\Gamma \vdash \forall(X{\leq}S_1)S_2 \leq \forall(X{\leq}T_1)T_2}$$

and proved that the resulting subtyping relation satisfies many desirable properties. Since difficulties with $F_{\leq}$ are traditionally associated with the subtyping relation, we assumed that the usual, straightforward treatment of the typing relation would carry over. It does not.

The new system lacks the *minimal typing* property, i.e., the set of all types inhabited by a given term may not contain a least element. This can be seen by the following example, due to Ghelli [message to the *Types* list, 28 Jan. 1994]. Let a be the term $\equiv \Lambda X{\leq}Y.\ \lambda x{:}X.\ x$. One can prove that a has both the type $\forall(X{\leq}Y)X{\rightarrow}X$ and the type $\forall(X{\leq}Y)X{\rightarrow}Y$. But these types are incomparable and they have no common lower bound. An immediate consequence of this fact is that the "standard" typing algorithm for the new calculus (the evident modification of the algorithm for $F_{\leq}$) is correct but not complete, since, for example, it will deduce the first type for a, but not the second one. One might also consider redefining the typechecking relation so that only the first type for a is allowed — effectively establishing the completeness of the straightforward typechecking algorithm by fiat. But this loses subject reduction.

The decidability of typechecking for $F_{\leq}$ with $\forall$-top remains an open problem. But even if the system were decidable (for example, by an algorithm computing finite sets of minimal types), it seems unlikely that such an algorithm would be acceptably efficient. Consider the contexts $\Gamma = \ldots, D{\leq}\ C{\rightarrow}C,\ B \leq C,\ A{\leq}\ B$ and $\Gamma' = \ldots, D{\leq}\ C{\rightarrow}C,\ B \leq C,\ A \leq \mathsf{Top}$. The subtyping statement $D{\leq}\ A{\rightarrow}C$ is provable under Γ, but not under Γ'. It follows that an algorithm that computes a complete set of minimal types for a term like $\Lambda A{\leq}\ B.\ \lambda x{:}D.\ x$ by "breaking the link" between A and its bound B must consider promoting D to $C{\rightarrow}C$ and then further promoting the result, even though D appears quite some distance from A in the context and, worse yet, even though the binding of D is *before* A's binding. This example suggests that the "sets of minimal types" of a term may be very large.

At the moment, there seem to be two choices for calculi with bounded quantification: (1) to embrace $F_{\leq}$ and learn to live with its difficulties; or (2) to return to Cardelli and Wegner's original formulation, sometimes called Kernel Fun, which used another, slightly odd-looking but much better behaved restriction of the quantifier rule:

$$(\forall\text{-Fun})\quad \frac{\Gamma, (X{\leq}U) \vdash S_2 \leq T_2}{\Gamma \vdash \forall(X{\leq}U)S_2 \leq \forall(X{\leq}U)T_2}$$

[CP94] Giuseppe Castagna and Benjamin Pierce. Decidable bounded quantification. In *Proceedings of Twenty-First Annual ACM Symposium on Principles of Programming Languages, Portland, Oregon.* ACM, January 1994.

*LIENS(CNRS) 45, rue d'Ulm Paris, France

†LFCS, Univ. of Edinburgh. Edinburgh, U.K.