A joint effort in the predicate calculus

For the last few weeks, Dan Grundy visited us in Eindhoven. During that time, we had some lovely discussions about thought and mathematics, and also had occasion to solve a few mathematics problems. One of our designs was so lovely, I feel it should be preserved for posterity in a JAW . The problem comes from EWD1266a : Let predicates f, g satisfy

(0) $f.x \Rightarrow g.x$ for all x ;

let equation $x : f.x$ have at least 1 solution, ie

(1) $\langle \exists x :: f.x \rangle$;

let equation $x : g.x$ have at most 1 solution, ie

(2) $g.x \wedge g.y \Rightarrow x = y$ for all x, y .

Then we may conclude the converse of (0) , ie

(3) $g.y \Rightarrow f.y$ for all y .

(The mysterious dummy y in (3) —as opposed to x in (0) — is presumably there so as to avoid variable clash with the dummy x in (1) . Still, it was mysterious enough that we were a little disappointed with Dijkstra's phrasing.)

In tackling this problem, we forbade ourselves from making any unjustified design decisions. So the first concern, stressed by Apurva, was to investigate proof shapes. Should we manipulate the entire expression in (3) ? Or should we manipulate one side of it into the other? We could not see a way to decide: monotonicity would allow us to use (0) in any of these proof shapes, and we had no idea how to exploit either (1) or (2) . I was particularly bothered by (2) , because we knew nothing about the type of x and y ; hence I felt the only way for $x = y$ to enter the picture was via the one-point rule. Our concerns having been voiced, we sat and stared at the board in silence for a long time, pondering.

During this period of silence, we found no new answers, so eventually we decided to investigate to what extent the use of the one-point rule could dictate our proof. For simplicity's sake, we put aside the proof shape which begins with the entire expression in (3) —it gave us too much manipulative freedom— , and focussed instead on either weakening $g.y$ to $f.y$, or strengthening $f.y$ to $g.y$. Consider the weakening chain: Once we introduce the expression $x = y$ into the range of a quantification via the one-point rule, we then have to strengthen that expression via (2) . Thus, since we are constructing a weakening chain, we had better use the one-point rule to introduce $\forall$, since $\forall$ is antimonotonic in its range. Similarly, in the strengthening chain, we should use the one-point rule to introduce $\exists$.

Again we sat for awhile pondering, though in retrospect I'm not sure why: the $\exists$ in (1) makes the strengthening chain the more attractive option. Finally, we calculated, for arbitrary y :

$f.y$

$\equiv$ { one-point rule, as mentioned above }

$\langle \exists x : x = y : f.x \rangle$

$\Leftarrow$ { (2) , monotonicity of $\exists$ }

$\langle \exists x : g.x \wedge g.y : f.x \rangle$

$\equiv$ { having formed the goal $g.y$, we extract it using $\wedge$ over $\exists$ }

$g.y \wedge \langle \exists x : g.x : f.x \rangle$

$\equiv$ { absorption, via (0) , preparing for (1) }

$g.y \wedge \langle \exists x :: f.x \rangle$

$\equiv$ { (1) }

$g.y$.

This is lovely. And as Dan pointed out, note that this is a highly asymmetric proof: If we were to invert the order of the steps, the first step from $g.y$ to $g.y \wedge \langle \exists x :: f.x \rangle$ would be a considerable rabbit; the next step would be even worse. Also observe that the weakening chain discussed above is not a viable option, as the following calculation shows:

$g.y$

$\equiv$ { one-point rule }

$\langle \forall x : x = y : g.x \rangle$

$\Rightarrow$ { (2) , antimonotonicity of $\forall$ in the range }

$\langle \forall x : g.x \wedge g.y : g.x \rangle$

$\equiv$ { predicate calculus }

true .

* *
*

Above I mentioned that the inversion of our proof would be full of rabbits. Despite this, it is still possible to design a weakening chain from $g.y$ to $f.y$ that is more-or-less rabbit-free, and this is what Dijkstra does in EWD1266a . His proof is as follows (the heuristics are mine):

$g.y$

$\equiv$ { In this chain, we have to introduce f and eliminate g . We exploit (1) immediately in order to achieve the former goal. }

$\langle \exists x :: f.x \rangle \ \wedge \ g.y$

$\equiv$ { Aiming to exploit (2) to eliminate g , we first need $g.x$ to enter the picture. By (0) , we have $f.x \equiv f.x \wedge g.x$, so we use this to rewrite our manipulandum equivalently. }

$\langle \exists x :: f.x \wedge g.x \rangle \ \wedge \ g.y$

$\equiv$ { Predicate calculus, to form $g.x \wedge g.y$. }

$\langle \exists x :: f.x \wedge g.x \wedge g.y \rangle$

$\Rightarrow$ { (2) , as planned }

$\langle \exists x :: f.x \ \wedge \ x = y \rangle$

$\equiv$ { one-point rule }

$f.y$.

Please note that Dijkstra's proof is not simply the inversion of ours!

* *
*

I would like to conclude with two small comments on Dijkstra's proof.

My first comment is that in trying to fill in the heuristics for Dijkstra's proof, I had to make a slight change to the proof shape, namely, interchanging the second and third steps. Dijkstra distributes $g.y$ into the existential quantification straight away; however, without the expression $g.x$ in the picture, there doesn't seem to be much reason for this move. The point is a small one, but in a note about heuristics, it should be made.

My second comment is that in EWD1266a , Dijkstra makes an error: he precedes his proof with "for arbitrary x " instead of "for arbitrary y " . Again, this is a relatively harmless mistake, but clearly it stems from the aforementioned "mysterious" change of dummy from x to y , which just makes that change all the more lamentable.

en route to Schiphol, 26 April 2006

Jeremy Weissmann
11260 Overland Ave. #21A
Culver City, CA 90230
USA
jeremy@mathmeth.com