A surprising (?) new heuristic

While working an exercise for Rob Hoogerwoord's thus-far excellent class in Programming by Calculation, I found myself surprised by what I think is a new heuristic for designing calculational proofs in Wim Feijen's proof format. In this JAW , I will present the heuristic, then my solution to the exercise, and finally some related observations.

In the following, P , Q , and p are of type **boolean scalar**.

* * *

Calculations don't always work the way we'd like them to. The way this all started was a failed effort to prove

(0) $\quad P \Rightarrow Q$

by designing a weakening chain from boolean scalar P to boolean scalar Q . I wound up getting as far as p ; that is, I proved:

(1) $\quad P \Rightarrow p \quad .$

That wasn't enough, but I saw how p could help me get from P to Q , and so — without proving this was a valid technique— I designed a new chain from P to Q , using p in the hints. That is, I proved:

(2) $\quad p \Rightarrow (P \Rightarrow Q) \quad .$

It never dawned on me that I couldn't do this; I had intuitively assumed it was correct.

In fact it is. We can show the equivalence of (0) and (2) under assumption of (1) , as follows:

$$\begin{array}{ll}
 & p \Rightarrow (P \Rightarrow Q) \\
\equiv & \quad \{ \text{ shunting } \} \\
 & p \wedge P \Rightarrow Q \\
\equiv & \quad \{ \ (1) \text{ , ie } p \wedge P \equiv P \ \} \\
 & P \Rightarrow Q \qquad .
\end{array}$$

This yields a not very surprising

Heuristic 0. If $P \Rightarrow p$, we can use p in a proof of $P \Rightarrow Q$. Phrased alternatively: we can use the consequences of P when weakening it. (End of Heuristic 0 .)

A trivial consequence of Heuristic 0 is:

Heuristic 1. If $P \Leftarrow p$, we can use $\neg p$ in a proof of $P \Leftarrow Q$. (End of Heuristic 1 .)

Heuristic 1 follows from Heuristic 0 under the substitution $P, Q, p := \neg P, \neg Q, \neg p$, using the law of the contrapositive twice.

Let us as designers of mathematical arguments try to understand what Heuristic 1 says. Suppose we are attempting to prove $P \Leftarrow Q$ by designing a strengthening chain from P to Q . Further suppose we get only as far as p , that is, we show only $P \Leftarrow p$. Then in our next effort to get from P to Q , we can use $\neg p$!

Is this surprising? Yes, I think so. Is this trivial? Yes, I think so. It is to me as surprising and trivial as the validity of the step:

$$\begin{array}{ll} & a \leq b \quad \vee \quad c \leq d \\ \Leftarrow & \{ \text{ monotonicity, transitivity } \} \\ & a + c \ \leq \ b + d \qquad . \end{array}$$

As surprising, as trivial... and as useful!

* * *

From Rob Hoogerwoord comes the following exercise: Show that no function l exists satisfying $\langle \forall x, y :: \ l.(x.y) = x \ \rangle$. In other words, we wish to show for all l :

(3) $\quad \langle \exists x, y :: \ l.(x.y) \neq x \ \rangle \qquad .$

We like this formulation because of the so-called Anti-Singleton axiom, which states:

(4) $\quad \langle \exists x :: \ E \neq x \ \rangle$

for any expression E in which x does not occur.

The context of this problem is showing that $x.y$ is not a suitable choice for the pair of x and y , denoted in Hoogerwoord's text by $(P.x).y$. The crucial difference between these expressions with respect to x and y is that in the former, x is applied as a function, while in the latter, x is the argument of a function. So somewhere in our proof we would like to exploit the fact that in (3) , x is applied as a function. I propose:

$$\begin{array}{ll} & \langle \exists x, y :: \ l.(x.y) \neq x \ \rangle \\ \Leftarrow & \{ \text{ instantiation } \ x := I \ , \text{ where } \ I \ \text{ is the left-identity of } \ . \ \} \\ & \langle \exists y :: \ l.(I.y) \neq I \ \rangle \\ \equiv & \{ \ I \ \text{ is the left-identity of } \ . \ \} \\ & \langle \exists y :: \ l.y \neq I \ \rangle \qquad . \end{array}$$

It is not clear how to continue this chain. Our options are to use Leibniz, or to instantiate y . Using Leibniz to introduce new arguments would yield $\langle \exists y, x :: \ (l.y).x \neq x \ \rangle$, which is unprovable in general, since l could be C , the right-projection function. Using Leibniz to introduce new functions forces us to consider I as an argument, whereas I is only useful as a function. And instantiating y shifts the focus to l , about which nothing is known.

So we are stuck; however, by Heuristic 1 we know that the negation of this last line is a new tool for our toolkit:

(5) $\quad \langle \forall y :: \ l.y = I \ \rangle \quad .$

Well well! We now have a definition of l at our fingertips! Emboldened, we try again:

$\quad \langle \exists x, y :: \ l.(x.y) \neq x \ \rangle$

$\equiv \quad$ { (5) with $y := x.y$ }

$\quad \langle \exists x, y :: \ I \neq x \ \rangle$

$\equiv \quad$ { vacuous quantification over y }

$\quad \langle \exists x :: \ I \neq x \ \rangle$

$\equiv \quad$ { (4) , ie Anti-Singleton axiom, with $E := I$ }

$\quad$ **true** $\quad .$

What a lovely design!

* * *

In this last section, I would like to mention why I have restricted my discussion to boolean scalars, and not boolean structures (ie predicates) more generally. Simply put, it doesn't work.

To see this, let X , Y , and x be of type predicate, and suppose —analogously to the above— that we have proved $[X \Rightarrow x]$ on the road to proving $[X \Rightarrow Y]$. If we now investigate the analogue of (2) , we find:

$\quad [x] \ \Rightarrow \ [X \Rightarrow Y]$

$\equiv \quad$ { “ $bs \Rightarrow$ ” distributes over $[\,]$ }

$\quad [\ [x] \ \Rightarrow \ (X \ \Rightarrow \ Y)\ ]$

$\equiv \quad$ { shunting }

$\quad [\ [x] \wedge X \ \Rightarrow \ Y\ ]$

$\equiv \quad$ { assuming $[\ X \Rightarrow [x]\ ]$ }

$\quad [\ X \ \Rightarrow \ Y\ ] \quad .$

So we need $[\ X \Rightarrow [x]\ ]$, while we only have $[\ X \Rightarrow x\ ]$. If X or x is scalar, these are equivalent; otherwise we need to know more.

Alternatively, we could use x punctually. The step:

$$\begin{array}{ll} & S \\ \Rightarrow & \{ \text{ hint why } [\ x \ \Rightarrow\ (S \Rightarrow T)\ ] \ \} \\ & T \qquad , \end{array}$$

rather than

$$\begin{array}{ll} & S \\ \Rightarrow & \{ \text{ hint why } [\, x\,] \ \Rightarrow\ [\ S \Rightarrow T\ ] \ \} \\ & T \qquad , \end{array}$$

suffices to salvage our heuristics, as the readers can check for themselves.

* * *

I should have mentioned that of particular interest is the case $p := P$. Our heuristics then tell us that in a proof of $P \Rightarrow Q$ we may use P , and that in a proof of $P \Leftarrow Q$ we may use $\neg P$. The latter observation connects this note to EWD1099 .

* * *

This technical note has the distinction of being my first since arriving in Eindhoven. Tremendous thanks go to Wim Feijen, Tom Verhoeff, and Rob Hoogerwoord for stimulating my mind so much in only 4 days!

Eindhoven, 8 September 2005

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