Constructing the Galois Adjoint

by the Eindhoven Tuesday Afternoon Club [†]

Quite a number of mathematical theorems express the existence of some object g satisfying some condition $P.g$. A viable way of proving such a theorem is to embark on a proof of $P.g$ and , during that process, collect requirements to be met by the object g . The proof is then completed, if the requirements thus collected can be satisfied . This note deals with a mini-example of such a performance .

The example is one of the first theorems one encounters in the (very nice) theory of so-called Galois Connexions . It reads —notions and notations explained later— :

(0) For any function f ,

$$\begin{array}{l} f \text{ distributes over } \uparrow \\ \Rightarrow \\ \langle \exists g :: \langle \forall x,y :: f.x \le y \equiv x \le g.y \rangle \rangle \end{array}$$

(The reverse implication holds as well, but does not concern is here.) All proofs of this theorem that we know of invariably begin with something miraculous like

" Let $g.y = \langle \uparrow z : f.z \le y : z \rangle$ " ,

and then proceed by showing that the function g

† A. Bijlsma , R.W. Bulterman, W.H.J. Feijen, A.J.M. van Gasteren, C.S. Scholten, and —student members— F.W. van der Sommen and A.I.J. Wijnands

thus defined is a witness for the existence demanded in (0).

One of the purposes of this note is to show how such a witness can be constructed rather than postulated in advance. Furthermore, we wish to show how this can be done by purely uninterpreted manipulation of the formulae, guided by their shapes rather than by their meanings.

* * *

Nomenclature

We consider a universe with a complete partial order, denoted $\leq$ —"at most"—. Then, for any predicate R and for any function f of the appropriate type, we are entitled to write down expression $\langle \uparrow z : R.z : f.z \rangle$, which yields an element of the universe as well. Such expressions are related to $\leq$ by

$$(1)\qquad \begin{aligned} &\langle \uparrow z : R.z : f.z \rangle \leq y \\ &\equiv \\ &\langle \forall z : R.z : f.z \leq y \rangle \qquad\qquad \text{(for all } y\text{)} \end{aligned}$$

(In standard literature, symbol $\uparrow$ —"up"— is often denoted by $\sqcup$, or $\bigvee$, or $\exists$, or lub, or sup.)
The only theorem from $\uparrow$-theory that we will use in what follows is the rule of instantiation —proof omitted— :

(2) $\quad R.x \quad \Rightarrow \quad x \leq \langle \uparrow z : R.z : z \rangle \quad$ (for all x) .

Our target theorem (0) also mentions the notion "f distributes over $\uparrow$". By definition, this equivales

$$f.\langle \uparrow z : R.z : z \rangle = \langle \uparrow z : R.z : f.z \rangle$$
(for all R) .

Without proof, we mention the theorem

(3) $\quad f$ distributes over $\uparrow$
$\Rightarrow$
f is monotonic with respect to $\leq$.

(<u>End</u> of Nomenclature.)

* * *

Now we are ready to prove (0). As announced earlier we embark on a proof of

$$\langle \forall x, y :: \quad f.x \leq y \quad \equiv \quad x \leq g.y \rangle \ ,$$

which we will carry out by proving, for arbitrary x and y,

$$f.x \leq y \quad \equiv \quad x \leq g.y \quad .$$

In order to emphasize that the object $g.y$ is the unknown that has to be resolved, we replace it by a completely meaningless symbol such as $\heartsuit$ —"heart"— . In order to remember that $\heartsuit$ may depend on y, we explicitly state

$$\heartsuit \;=\; g.y \qquad .$$

Now the challenge is to prove

$$\text{A} \qquad \boxed{\begin{array}{ll} & f.x \leq y \\ \equiv & \\ & x \leq \heartsuit \end{array}}$$

for a function f that distributes over $\uparrow$

Careful inspection of the proof obligation depicted in A shows that in the transition from the first to the last line, f disappears. from the left side of the $\leq$-symbol. However, we know of only two simple mechanisms to remove a function application. The one is

Leibniz' Rule: $\quad f.x = f.y \;\Leftarrow\; x = y \quad ,$

and the other is

Monotonicity: $\quad f.x \leq f.y \;\Leftarrow\; x \leq y \quad .$

But no matter which of the two we use they both confront us with a strengthening rather than an equivalence preserving step. Therefore, there is little choice but to replace A by the two independent

$$\underset{\text{B0}}{\boxed{\begin{array}{ll} & f.x \leq y \\ \Leftarrow & \\ & x \leq \heartsuit \end{array}}} \qquad \text{and} \qquad \underset{\text{B1}}{\boxed{\begin{array}{ll} & x \leq \heartsuit \\ \Leftarrow & \\ & f.x \leq y \end{array}}}$$

We concentrate on proof obligation B0 first. In view of the occurrence of symbol $\leq$ in the first line, we choose to eliminate f by an appeal to f's monotonicity:

$$
\begin{array}{ll}
 & f.x \leq y \\
\Leftarrow & \{\bullet\ f.\heartsuit \leq y,\ \text{using that} \leq \text{is transitive}\} \\
 & f.x \leq f.\heartsuit \\
\Leftarrow & \{f \text{ is monotonic since it distributes over } \uparrow,\ \text{see (3)}\} \\
 & x \leq \heartsuit \quad .
\end{array}
$$

And this completes our proof of B0, albeit at the expense of a new proof obligation indicated above by $\bullet$:

C $\quad \boxed{f.\heartsuit \leq y}$

We postpone tackling B1, because there is really nothing we can do with expression $x \leq \heartsuit$. Therefore, we investigate C first.

Because f distributes over $\uparrow$ — a given that we have not used yet — it is sweetly reasonable to propose an $\uparrow$-expression for $\heartsuit$. But besides being sweetly reasonable, the proposal is harmless as well, since any expression can be written as an $\uparrow$-expression, for instance by means of the one-point-rule. So we proceed by taking, for an as yet unknown predicate R

$$\heartsuit \;=\; \langle \uparrow z : R.z : z \rangle$$

Now we calculate with C :

$f.\heartsuit \leq y$

$\equiv$ { definition of $\heartsuit$ }

$f.\langle \uparrow z : R.z : z \rangle \leq y$

$\equiv$ { f distributes over $\uparrow$ }

$\langle \uparrow z : R.z : f.z \rangle \leq y$

$\equiv$ { (1) }

$\langle \forall z : R.z : f.z \leq y \rangle$

$\equiv$ { predicate calculus }

$\langle \forall z :: R.z \Rightarrow f.z \leq y \rangle$

$\equiv$ { • D0 below }

true .

Thus, C is satisfied if we insist on

D0 $\quad \boxed{R.z \;\Rightarrow\; f.z \leq y}$ (for all z) ,

and that is what we shall do .

Now we can tackle the pending B1 , which will no longer offer any problems :

$x \leq \heartsuit$

$\equiv$ { definition of $\heartsuit$ }

$x \leq \langle \uparrow z : R.z : z \rangle$

$\Leftarrow$ { instantiation, see (2) }

$R.x$

$\Leftarrow$ { • D1 below, for $z := x$ }

$f.x \leq y$,

where

D1 $\boxed{R.z \;\Leftarrow\; f.z \le y}$ (for all z) .

Finally, in fulfilling D0 and D1 we have no choice but to take for R

$$R.z \;\equiv\; f.z \le y \qquad \text{(for all } z\text{)} .$$

Thus we find for our unknown $\heartsuit$

$$\heartsuit \;=\; \langle \uparrow z : f.z \le y : z \rangle \; ,$$

and so, g defined by

$$g.y \;=\; \heartsuit \qquad \text{(for all } y\text{)}$$

is a witness for theorem (0)

* * *

In the above example the design of the proof is driven by the shape of formulae and by the manipulative opportunities available, rather than by the traditional mode of formulae interpretation. As far as this departure from tradition is concerned the above example is not incidental at all. Many more and much more fancy parts of mathematical reasoning can be tackled successfully by focussing on syntactic manipulation.

This style of doing mathematics is a direct offspring of the modern style of program development, which has become largely calculational by now. The adoption of the calculational style for program design has greatly increased the quality of the design process itself and of the programs designed. Thus, many programs have been constructed that not only correctly and efficiently meet their specification, but in addition are of a nature far beyond what an operationally thinking human mind can conceive. More and more computing scientists have begun to recognize that the transfer of calculational reasoning to traditional mathematics can be equally rewarding, an observation which is confirmed by this special issue of IPL.

Eindhoven,
5 September 1994