How subtypes should enter the picture

In June 1992, I completed EWD1123 "The unification of three calculi", which was written for educational purposes. Its use in the classroom, however, revealed that its elliptic introduction of the traditional boolean domain as subtype of the boolean structures was at least utterly confusing. The main purpose of this note is to remedy that situation; the fact that its title does not refer to boolean scalars and structures reflects my belief that the remedy is more generally applicable.

A proper treatment would consist of the following ingredients.

- the traditional boolean domain {true, false}, together with its operators $\equiv \Rightarrow \Leftarrow \wedge \vee \neg$, is assumed to be known; we call it the domain of the "boolean scalars" in order to distinguish it from the boolean structures to be introduced shortly.
- the "everywhere operator", applied by surrounding the argument by a pair of square brackets, is a function from the boolean structures to the boolean scalars
- on boolean structures the infix operator

$\dot{\equiv}$ is introduced with the properties

(0) $[x \mathrel{\dot{\equiv}} y] \equiv x = y$

$\dot{\equiv}$ is associative and symmetric; note that the symmetry of $\dot{\equiv}$

$$(x \mathrel{\dot{\equiv}} y) = (y \mathrel{\dot{\equiv}} x)$$

can be expressed —thanks to (0) and the associativity of $\dot{\equiv}$ — as

$$[x \mathrel{\dot{\equiv}} y \mathrel{\dot{\equiv}} y \mathrel{\dot{\equiv}} x] \quad ,$$

from which it follows that $y \mathrel{\dot{\equiv}} y$ is (for any y) the neutral element of $\dot{\equiv}$; we denote it by $\mathit{tr\dot{u}e}$:

$$[x \mathrel{\dot{\equiv}} \mathit{tr\dot{u}e} \mathrel{\dot{\equiv}} x]$$

- $\dot{\vee}$ is postulated to be symmetric, associative, and idempotent and to distribute over $\dot{\equiv}$

- analogously to EWD1123, $\dot{\Rightarrow}$, $\dot{\Leftarrow}$, $\dot{\neg}$ and the constant $\mathit{f\dot{a}lse}$ are introduced, and the properties of the everywhere operator are given —i.e. some derived and some postulated— such as

$$[\mathit{tr\dot{u}e}] \equiv \mathit{true}$$

$$[\mathit{f\dot{a}lse}] \equiv \mathit{false}$$

$$[x \mathrel{\dot{\wedge}} y] \equiv [x] \wedge [y] \quad .$$

- next we observe a one-to-one correspondence between theorems about boolean scalars and boolean structures, e.g. we have in scalars

$$x \wedge (x \vee y) \equiv x \vee false$$

as we have in structures

$$[x \dot{\wedge} (x \dot{\vee} y) \dot{\equiv} x \dot{\vee} \dot{false}] \quad ,$$

and this is the moment that we yield to the temptation to omit all the dots: we overload $\equiv \wedge \vee$ etc., we also make no longer a distinction between the boolean scalars true and false and the "constant boolean structures" $\dot{true}$ and $\dot{false}$, which we now write as true and false respectively. By thus embedding the boolean scalars in the boolean structures, we have made the former a subtype of the latter, very much in the way in which the integers can be made a subtype of the reals. Not only have we symplified our notation, we have also embellished properties, e.g. [ ] is now idempotent, and it distributes over $\wedge$.

Austin, 4 October 1993

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188 , USA