For brevity's sake

Since the turn of the century, Russell's Paradox is a standard ingredient of all texts on the foundation of mathematics. Its purpose is to illustrate the trouble one can run into by admitting the seemingly innocent notion of "the set of all sets". We shall not pursue that trouble here, since this is not a book on the foundation of mathematics. Today we are not interested in Russell's Paradox, we are solely interested in its presentation. Being essentially a one-liner, it is a very simple example to make our point; it has the added advantage of being familiar to many. Its usual presentation is along the following lines.

A set may be a member of itself or not. Consider now the set of all sets that are not a member of themselves. Calling this set R , its formal definition would be

$$(0) \qquad R = \{x \mid x \notin x\}$$

—to be read as "the set of all sets x such that $x \notin x$ —. The question we now try to answer is: "Is R a member of itself?". Suppose that R is not a member of itself; since, according to (0), R contains all such sets, R would contain

---

Mathematical Methodology

itself, which contradicts our assumption that R is not a member of itself. Conversely, assume that R is a member of itself; since, according to (0), R contains <u>only</u> sets that are not members of themselves, R would not contain itself, which contradicts our assumption. So, our question "Is R a member of itself?" admits neither answer.

This concludes the first presentation of Russell's Paradox. Consider now the following alternative.

A set may be a member of itself or not. Consider now the set of all sets that are not a member of themselves. Calling this set R, the formal definition of membership of it would be

(1) $\quad x \in R \;=\; x \notin x \quad$ for all x .

Instantiation of (1) with $x := R$ yields

$$R \in R \;=\; R \notin R \quad ,$$

which is a contradiction.

This concludes the second presentation of Russell's Paradox. It is so much shorter and simpler than the first one that we had better fully understand how this tremendous gain could

---

be obtained.

A major difference between (0) and (1) is that (0) states the equality

$$\mathcal{R} = \{x \mid x \notin x\} \quad ,$$

in which x is a "bound variable" — i.e., local to the right-hand side — whereas (1) states the equality

$$x \in \mathcal{R} \;=\; x \notin x \quad ,$$

in which x is a "free variable" — i.e., global to the whole expression — . The latter equality can now be instantiated by substituting an arbitrary set for x ! The latter manipulative freedom is exploited to great advantage by the instantiation $x := \mathcal{R}$. Instantiation is a very simple operation, and the example suggests that free variables in definitions (axioms, theorems, etc.) are an asset. (Later examples will confirm this as yet tentative moral: the availability of free variables can significantly contribute to the economy of manipulation.)

Another major difference between (0) and (1) is the different status of their equality signs. In (0), the equality sign expresses equality between sets and is only acceptable after sets have been admitted as first-class citizens; in (1), the equality sign expresses equality between boolean expressions and is only acceptable after boolean expressions have been admitted as first-class citizens. Let me explain what is meant here by "first-class citizens".

Prior to the introduction of the boolean domain, people used what we now recognize as boolean expressions, but they were viewed as statement of fact — $3<4$ — or as condition — $x>0$ — or as equation — $x^2+bx+c=0$ — or as laws — $(a+b)^2 = a^2+2ab+b^2$ —, and things like $3>4$, $2+2=5$ and $n=n+3$ were considered "wrong". All this changed with the introduction of the two-element boolean domain {true, false} which provides the vocabulary needed to assign <u>values</u> to boolean expressions: $3<4$ is a way for writing true, $3>4$ is a way for writing false, whereas the value of $x>0$ depends on the value of x: for positive x it yields true, for nonpositive x it yields false. As syntactic units, boolean expressions existed after 1558, when Robert Recorde introduced the now familiar equality sign = as infix operator; they became "first-class citizens" in 1858, when George Boole introduced the possibility of evaluating them.

Mathematical Methodology

In retrospect, one might be tempted to regard the introduction of something as simple as the boolean domain as a minor invention, but I think that that would be a grave mistake: it is a great invention because, being so simple, it is such a powerful simplifier. It is of the same level as the introduction of natural numbers, which enabled us to add 3 to 5, regardless of whether we are adding apples or pears.

George Boole made a radical invention, so radical, in fact, that now, more than a century later, the scientific community has not absorbed it yet. (To stay with the metaphor: officially, boolean expressions may have reached the status of first-class citizens, in practice —because old habits and prejudices die hard— they are still the victims of discrimination.) Let me give you a few examples.

In the programming language FORTRAN, as conceived a century after Boole published his invention, boolean expressions are allowed, but there are no boolean variables! Their introduction into programming had to wait until the design of ALGOL 60. (In fact, the inclusion of boolean variables and operators has been one of the main reasons for the educational significance of ALGOL 60; the inclusion of recursion was another reason.)

Very tellingly, their inclusion as first-class citizens did not prevent programmers - even programmers that had skipped FORTRAN - from writing

if $h \geq N$ <u>then</u> eq := true <u>else</u> eq := false

instead of the so much crisper

eq := $h \geq N$,

or from writing

if eq $\equiv$ true <u>then</u> ......

instead of the simpler

if eq <u>then</u> ..... .

Not only as programmers, but also as mathematicians we have to learn to avoid such spurious case analyses and superfluous comparisons.

Another example of the slow acceptance of the boolean domain is provided by the tacit assumption of the mathematical community that the Pythagorean Theorem is a theorem about right-angled triangles. But is it? In the usual terminology for sides and angles it states

$$\gamma = 90^\circ \Rightarrow a^2 + b^2 = c^2$$

and the right angle seems to stare you in the

Mathematical Methodology

face. But everyone familiar with boolean expressions knows that the above has the same value as

$$a^2+b^2 \neq c^2 \Rightarrow \gamma \neq 90^\circ \quad .$$

The Pythagorean Theorem is a theorem about all triangles (and there exists a nice, elementary proof of it for which the complementary picture displays an arbitrary triangle).

The last remark to be made about the two presentations of Russell's Paradox is that the instantiation of (1) enabled us to establish the contradiction without the case analysis of the first presentation.

* * *

In the last programming example from ALGOL 60 I wrote eq ≡ true , using ≡ to denote equality of boolean operands rather than the = I used before. In a moment we shall see why a special equality sign in the case of boolean operands could be justified. For this purpose, I turn to a problem I owe to Roland C. Backhouse.

We are shown a golden vase and a silver vase, each carrying an inscription. We are told that one of the two vases contains a treasure, placed so that the two inscriptions don't lead to a contradiction. The two inscriptions are:

on the silver vase: "The treasure is not in this vase."

and

on the golden vase: "Of these two inscriptions precisely 1 is true."

Can we locate the treasure?

Denoting the truth values of these two statements by S and G respectively, the challenge is to determine the value of S. To this end we turn to the inscription on the golden vase, which in this terminology can be rephrased as

(2) $$S \equiv \neg G \quad .$$

(The symbol $\neg$ denotes negation, and saying that of two boolean values precisely 1 is true is the same as saying that the one is the negation of the other.)

Expression (2) being a formalization of the statement we had defined to have the same value as G, we conclude

$$(S \equiv \neg G) \equiv G \quad .$$

Besides being symmetric, the operator $\equiv$, denoting equality of boolean operands, has the nice property of being associative, i.e., it does not matter how we place the parentheses, and our conclusion could also have been formulated as

$$S \equiv (\neg G \equiv G) \quad , \tag{3}$$

which immediately yields

$$S \equiv \mathit{false} \ ,$$

i.e., the treasure is in the silver vase.

Equation (3) in the unknowns S and G determines the value of S, but leaves the value of G undetermined. The above argument is so short because it avoids a spurious case analysis on the irrelevant value of G .

In a way, this example is very similar to the previous one: they both employ

$$\neg X \equiv X \equiv \mathit{false} \quad ,$$

whatever the value of X (thereby avoiding an avoidable case analysis). It has been included because it illustrates more.

I would like you to know that when I

called the above argument —which has been designed by G. Wiltink — "so short", I did not exaggerate. Backhouse draws attention to the fact that about two dozen steps are required if one tackles this problem in the style of Gentzen's "Natural Deduction", whereas Wiltink solves the problem in two (no less formal) steps. Backhouse concludes from the two dozen steps "It is clear from this example that a large amount of detail is required in a completely formal proof".

Wiltink's alternative strongly suggests that Gentzen, who tried to formalize "the natural way of drawing conclusions" —in German: "das natürliche Schliessen"—, neglected the economy of manipulation. (Before having seen Wiltink's alternative, Backhouse already suspected something of this sort: the quoted problem and its cumbersome solution occur in a section, tellingly titled "Natural Deduction?".) I am afraid that by neglecting the economy of manipulation, Gentzen has done the mathematical community a disservice: calculational proofs have got an undeserved bad name.

The fact that equality between boolean operands is associative, i.e., has a property not enjoyed

---

Mathematical Methodology

by equality in general, justifies a special name and a special symbol for it. A usual name is "equivalence" and a usual symbol is "$\equiv$". I adopt both and propose to pronounce the symbol as "equivales".

Remark One also encounters the symbol "$\Leftrightarrow$", pronounced as "if and only if", but I shall not follow that example. Symbol and name promote the view of equivalence as a shorthand for mutual implication, and it is precisely that view that tends to induce the case analyses that I would like to avoid. Moreover the "if and only if" linguistically hides the associativity of the equivalence so effectively that many people are unaware of it. Just try to understand the sentence "John sees with both eyes if and only if he sees with one eye if and only if he is blind.". According to all linguistic standards, this sentence is, by virtue of its syntactic ambiguity, total gibberish. (End of Remark.)

Wiltink's solution for locating the treasure makes one more thing clear. It shows that it does not suffice to be familiar with just the idea of the boolean domain. To really reap the benefits, we also have to be familiar with the boolean operators —such as equivalence,

Mathematical Methodology

negation, etc. – and with their algebraic properties – such as symmetry, associativity, etc. – . Acquiring that familiarity requires what in these financial times is known as "intellectual investment"; you can take my word for it that this investment quickly pays off.

Austin 15 November 1989

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

Mathematical Methodology