The unification of three calculi

The purpose of this note is to show how much the predicate calculus, the relational calculus, and the regularity calculus have in common, or, more precisely, how much commonality we can design into them. Because the predicate calculus will emerge as a subcalculus of the other two calculi, we discuss the predicate calculus first.

To begin with, I assume the reader to be familiar with

- the boolean domain {true, false}
- the unary (prefix) operator
  $\neg$ (not, negation)
- the binary (infix) operators
  $\vee$ (or, disjunction), $\wedge$ (and, conjunction)
  $\Rightarrow$ (implies, implication), $\Leftarrow$ (follows from, consequence)
  $\equiv$ (equivales or equals, equivalence or equality).

Above, the order from top to bottom coincides for the operators with the order of decreasing syntactic binding power; notice that —in order not to destroy the symmetry between them— we have given the same binding power to $\vee$ and $\wedge$.

In the following, s is variable ranging over a domain S and (primarily) x, y, z will be used to denote boolean functions whose domain is S. Function application will be denoted by an infix dot, preceded by a denotation of the function and followed by a denotation of the argument to which the function is applied. The application dot has the highest binding power and associates to the left — i.e. "$f.p.q$" is short for "$(f.p).q$" — .

To express that x and y are the same function, one writes traditionally

$$\langle\forall s: s\in S: \ x.s \equiv y.s\rangle \quad .$$

Here, the functions applied to s, viz. x and y, are denoted by single identifiers. With the rule that application to s distributes over the logical connectives, we can write the above as

$$\langle\forall s: s\in S: (x \equiv y).s\rangle \quad ,$$

with the understanding that $x \equiv y$ is now an expression denoting the function that is true wherever x and y are equal and is false elsewhere. The distribution can be applied repeatedly, e.g.

$$\langle\forall s: s\in S: \ x.s \vee y.s \equiv z.s\rangle$$

can be written — remember that $\equiv$ has the lowest binding power — as

$$\langle\forall s: s\in S: \ (x \vee y).s \equiv z.s\rangle$$

or even as

$$\langle\forall s: s\in S: (x \vee y \equiv z).s\rangle \quad .$$

For reasons that will become clear later, we now propose the following abbreviation:

instead of "$\langle\forall s: s\in S: ($" we write "[" and instead of "$).s\rangle$" we write "]" .

Using the square brackets, our formula expressing that the disjunction of x and y equivales z reduces to

$$[x \vee y \equiv z] \qquad ;$$

our former formula, expressing that x and y are the same function, now simply becomes

$$[x \equiv y] \qquad .$$

*
*
*

We now take what is a standard step in mathematical theory building. The step is taken after the introduction of a notational novelty — such as a new abbreviation or a "mathematical macro" — for formulae that were interpreted in a familiar domain of discourse. The step consists of starting with a clean slate and axiomatizing afresh the manipulations of the new formulae. In doing so, one creates a new domain of discourse; the rôle of the old, familiar domain of discourse, that used to constitute the subject matter, is thereby reduced to that of providing a possible model for the newly postulated theory. It is essential that the axioms of the new theory —which can be interpreted as theorems in the old universe of discourse— are clearly postulated as such and that the new theory is derived from them without reference to the model of the old universe of discourse. This is the only way to assure that the axioms of the new theory provide an interface that is independent of the old universe of discourse and that, hence, the new theory is safely applicable to alternative models.

Remark The above paragraph sketches in a bird's-eye view the process of mathematical abstraction. The result of being more abstract is not being more vague, on the contrary: the purpose of abstraction is the creation of a new semantic level at which one can again be absolutely precise, but with less commitment. The virtue of the new theory is that one can work in it, unburdened by the irrelevant details of the model that inspired it. Experience has shown that people's first confrontation with mathematical abstraction is often emotionally disturbing; the rest of the educational process hardly teaches the potential intellectual advantages of ignoring available knowledge and the manifest freedom of creating one's own universe of discourse could very well be frightening. (End of Remark.)

We begin our axiomatization by introducing (variables x, y, z of) a new type that we call "boolean structures". It contains the traditional boolean domain {true, false} as subtype, whose two values are sometimes referred to as "the boolean scalars".

We introduce a function from boolean structures to boolean scalars; it is called "the everywhere operator" and is denoted by surrounding the boolean structure it is applied to by a pair of square brackets.

<u>Legenda</u> The axioms and theorems that follow should be universally quantified over their global variables of type boolean structure. The marks <u>Ax</u> and <u>Th</u> distinguish axioms from theorems; the choice is somewhat arbitrary. (End of Legenda.)

In order to express that two boolean structures x and y are the same we write $[x \equiv y]$ instead of the more usual $x = y$. So, Leibniz's Principle

$$x = y \ \Rightarrow\ f.x = f.y$$

we render for arguments and function values of type boolean structure as

<u>Ax</u> $\quad [x \equiv y] \Rightarrow [f.x \equiv f.y]$.

<u>Remark</u> It is understood that an expression is a

function of any of its subexpressions. (End of Remark.)

For the general infix operator $\bullet$ we define the following terms:

"$\bullet$ is associative" means $[(x \bullet y) \bullet z \equiv x \bullet (y \bullet z)]$
"$\bullet$ is symmetric" means $[x \bullet y \equiv y \bullet x]$
"$\bullet$ is idempotent" means $[x \bullet x \equiv x]$,
(the three formula to be universally quantified as usual over x, y, z.) In what follows, we shall exploit associativity by omitting semantically superfluous parentheses whenever convenient — i.e. we don't hesitate to write $x \bullet y \bullet z$ for any associative $\bullet$ — .

<u>Ax.</u> $\equiv$ is associative

<u>Remark</u> For nonboolean types, equality is not associative — its associativity would lead to a type conflict — . The associativity of equality for boolean types is therefore very special, so special that it deserves a special symbol ("$\equiv$") with a special pronunciation ("equivales"). (End of Remark.)

<u>Ax.</u> $\equiv$ is symmetric

<u>Ax.</u> $[x \equiv true \equiv x]$ or $[x \equiv x \equiv true]$

In the first version, this axiom expresses that true is the left and right identity of $\equiv$ —note that it admits the parsings $[x \equiv (true \equiv x)]$ and $[(x \equiv true) \equiv x]$— . In the second version —parsed $[(x \equiv x) \equiv true]$— it expresses that $x \equiv x$ is also the identity element of $\equiv$. (The existence of an identity element of $\equiv$ follows from the previous two axioms.)

<u>Th</u>. $[true] \equiv true$.

For the disjunction $\vee$ we give the axioms

<u>Ax</u>. $\vee$ is associative
<u>Ax</u>. $\vee$ is symmetric
<u>Ax</u>. $\vee$ is idempotent .

For equivalence and disjunction together we give

<u>Ax</u>. $\vee$ distributes over $\equiv$, i.e.

$$[x \vee (y \equiv z) \equiv x \vee y \equiv x \vee z] \quad .$$

<u>Th</u>. $[x \vee true \equiv true]$ or $[x \vee true]$.

<u>Proof</u> We observe for any boolean structures x, y

$$
\begin{array}{ll}
 & x \vee true \\
= & \quad \{\text{Ax. with } x := y \text{ yields } [(y \equiv y) \equiv true]\} \\
 & x \vee (y \equiv y) \\
= & \quad \{\vee \text{ dist. over } \equiv\} \\
 & x \vee y \equiv x \vee y \\
= & \quad \{\text{Ax. with } x := x \vee y\} \\
 & true \; .
\end{array}
$$

(End of Proof.)

(Note that in view of the symmetry of $\vee$, we did not need to distinguish between left and right distribution over $\equiv$.)

The conjunction $\wedge$ is defined by

<u>Ax.</u> $[x \wedge y \equiv x \equiv y \equiv x \vee y]$,

an axiom that is also known as The Golden Rule.

<u>Th.</u> $\wedge$ is associative

<u>Th.</u> $\wedge$ is symmetric

<u>Th.</u> $\wedge$ is idempotent

<u>Th.</u> $[x \wedge true \equiv x]$

<u>Proof.</u> We observe for any boolean structure x

$x \wedge true$
$=$ { Golden Rule with $y := true$ }
$x \equiv true \equiv x \vee true$
$=$ { $[x \vee true]$ }
$x \equiv true \equiv true$
$=$ { identity element of $\equiv$ }
x .

(End of Proof.)

<u>Th.</u> $[x \wedge (y \equiv z) \equiv x \wedge y \equiv x \wedge z \equiv x]$

<u>Proof.</u> We observe for any (boolean structures) x, y, z

$x \wedge y \equiv x \wedge z \equiv x$
$=$ { Golden Rule twice }
$x \equiv y \equiv x \vee y \equiv z \equiv x \vee z$
$=$ { $\equiv$ symmetric; $\vee$ dist. over $\equiv$ }

$$x \equiv y \equiv z \;\equiv\; x \vee (y \equiv z)$$
$$= \quad \{\text{Golden Rule}\}$$
$$x \wedge (y \equiv z) \quad .$$

(End of Proof.)

A corollary of the last theorem is

<u>Th</u>. $\wedge$ distributes over "$\equiv\equiv$", i.e.
$$[u \wedge (x \equiv y \equiv z) \equiv u\wedge x \equiv u\wedge y \equiv u \wedge z] \quad .$$

<u>Th</u>. $\wedge$ and $\vee$ distribute over each other

<u>Proof</u> Because of the symmetry of $\wedge$ and $\vee$, we need not distinguish between left and right distribution. We show that $\wedge$ distributes over $\vee$ by observing for any x, y, z

$$x \wedge (y \vee z)$$
$$= \quad \{\text{Golden Rule}\}$$
$$x \wedge (y \equiv z \equiv y \wedge z)$$
$$= \quad \{\wedge \text{ dist. over } \equiv\equiv\}$$
$$x \wedge y \;\equiv\; x \wedge z \;\equiv\; x \wedge y \wedge z$$
$$= \quad \{\text{properties of } \wedge\}$$
$$x \wedge y \;\equiv\; x \wedge z \;\equiv\; x \wedge y \;\wedge\; x \wedge z$$
$$= \quad \{\text{Golden Rule}\}$$
$$(x \wedge y) \vee (x \wedge z) \quad .$$

(End of Proof.)

<u>Th</u>. The Laws of Absorption
$$[x \wedge (x \vee y) \equiv x] \;,\; [x \vee (x \wedge y) \equiv x] \quad .$$

<u>Proof</u> To demonstrate the first law of absorption we observe for any x, y

$$
\begin{array}{ll}
 & x \wedge (x \vee y) \equiv x \\
= & \quad \{\text{Golden Rule with } y := x \vee y\} \\
 & x \vee y \equiv x \vee x \vee y \\
= & \quad \{\vee \text{ idempotent}\} \\
 & x \vee y \equiv x \vee y \\
= & \quad \{\text{identity of } \equiv\} \\
 & true \quad .
\end{array}
$$

(End of Proof.)

Implication $\Rightarrow$ and consequence $\Leftarrow$ are defined by the following two axioms

<u>Ax.</u> $[x \Rightarrow y \;\equiv\; y \Leftarrow x]$

<u>Ax.</u> $[x \Rightarrow y \;\equiv\; x \vee y \equiv y]$

from which we immediately deduce with the Golden Rule

<u>Th.</u> $[x \Rightarrow y \;\equiv\; x \wedge y \equiv x]$.

A simple rewriting of the Laws of Absorption now yields

<u>Th.</u> $[x \Rightarrow x \vee y]$ and $[x \wedge y \Rightarrow x]$,

which are also known as Laws of Absorption.

<u>Th.</u> $[x \Rightarrow (y \Rightarrow z) \;\equiv\; x \wedge y \Rightarrow z]$

<u>Proof</u> We observe for any x, y, z

$$
\begin{array}{ll}
 & x \Rightarrow (y \Rightarrow z) \\
= & \quad \{\text{eliminate outer } \Rightarrow \text{ with } \wedge \equiv\} \\
 & x \wedge (y \Rightarrow z) \equiv x
\end{array}
$$

$= \quad \{\text{eliminate inner} \Rightarrow \text{with} \wedge \equiv\}$

$\quad x \wedge (y \wedge z \equiv y) \equiv x$

$= \quad \{\wedge \text{ almost distributes over } \equiv\}$

$\quad x \wedge y \wedge z \equiv x \wedge y$

$= \quad \{\text{reintroduction of} \Rightarrow \text{with} \wedge \equiv\}$

$\quad x \wedge y \Rightarrow z$

(End of Proof)

<u>Th.</u> $\quad [x \wedge (x \Rightarrow y) \equiv x \wedge y]$

<u>Proof</u> We observe for any x, y

$\quad x \wedge (x \Rightarrow y)$

$= \quad \{\text{eliminate} \Rightarrow \text{with} \wedge \equiv\}$

$\quad x \wedge (x \wedge y \equiv x)$

$= \quad \{\wedge \text{ almost distributes over } \equiv\}$

$\quad x \wedge x \wedge y \equiv x \wedge x \equiv x$

$= \quad \{\wedge \text{ is idempotent}\}$

$\quad x \wedge y \equiv x \equiv x$

$= \quad \{\text{identity of } \equiv\}$

$\quad x \wedge y$

(End of Proof)

<u>Remark</u> From the above and a Law of Absorption we can derive $[x \wedge (x \Rightarrow y) \Rightarrow y]$, a law that is at least since the Middle Ages known as the Modus Ponens. These days it no longer deserves a special name. (End of Remark.)

One of the most important properties of the implication is

Th. $\quad \Rightarrow$ is transitive, i.e.

$$[(x \Rightarrow y) \wedge (y \Rightarrow z) \Rightarrow (x \Rightarrow z)] \quad .$$

Proof We observe for any x, y, z

$$
\begin{array}{ll}
 & (x \Rightarrow y) \wedge (y \Rightarrow z) \Rightarrow (x \Rightarrow z) \\
= & \{\text{Theorem before last}\} \\
 & x \wedge (x \Rightarrow y) \wedge (y \Rightarrow z) \Rightarrow z \\
= & \{\text{last theorem}\} \\
 & x \wedge y \wedge (y \Rightarrow z) \Rightarrow z \\
= & \{\text{Last theorem}\} \\
 & x \wedge y \wedge z \Rightarrow z \\
= & \{\text{Law of Absorption}\} \\
 & \mathit{true} \quad .
\end{array}
$$

(End of Proof.)

Th. $\Rightarrow$ is reflexive, i.e. $[x \Rightarrow x]$.

Th. $\Rightarrow$ is antisymmetric, i.e.

$$[(x \Rightarrow y) \wedge (y \Rightarrow x) \Rightarrow (x \equiv y)]$$

Proof We observe for any x, y

$$
\begin{array}{ll}
 & (x \Rightarrow y) \wedge (y \Rightarrow x) \\
= & \{\Rightarrow \text{ in } \wedge \text{ and } \equiv, \text{ twice}\} \\
 & (x \wedge y \equiv x) \wedge (x \wedge y \equiv y) \\
= & \{\text{Punctual Leibniz, see below}\} \\
 & (x \wedge y \equiv x) \wedge (x \equiv y) \\
\Rightarrow & \{\text{Absorption}\} \\
 & x \equiv y \quad .
\end{array}
$$

(End of Proof.)

Our earlier proofs all consisted of sequences of equivalent expressions. Now we have established that $\Rightarrow$ is transitive, we also accept, as in the last proofs, proofs of implications in which some expressions in the sequence are connected by $\Rightarrow$ to their successor.

The implication is not a symmetric operator, its two operands have therefore different names: in $x \Rightarrow y$ and in $y \Leftarrow x$, x is called the "antecedent" and y is called the "consequent". In our last proof we started with the antecedent of the demonstrandum and ended up with its consequent; hence the occurrence of $\Rightarrow$ in the left-most column. We sometimes arrange implication proofs the other way round; in that case we start with the consequent and end up with the antecedent; such proofs have occurrences of $\Leftarrow$ in the left-most column.

Our last 3 theorems state that implication is transitive, reflexive, and antisymmetric, i.e. that implication is a "partial order". The adjectives "weaker" and "stronger" are in common usage to describe the "direction" of this partial order: in [antecedent $\Rightarrow$ consequent] the antecedent is said to be "stronger than the consequent" and the consequent is said to be "weaker than the antecedent". (The use of the comparatives is not entirely fortunate since antecedent and consequent could be equivalent.)

The unary operator $\neg$ (not, negation) is defined by the two axioms that connect it to the equivalence and the disjunction respectively

<u>Ax.</u> $[\neg(x \equiv y) \equiv \neg x \equiv y]$
<u>Ax.</u> $[x \vee \neg x]$ (i.e. Law <u>of</u> the Excluded Middle).

From the first axiom by itself one can deduce

<u>Th.</u> $[\neg x \equiv y \equiv x \equiv \neg y]$
<u>Th.</u> $[x \equiv \neg\neg x]$ (i.e. $\neg$ is an involution).

Together they yield another way to write the <u>im</u>plication:

<u>Th.</u> $[x \Rightarrow y \equiv \neg x \vee y]$

<u>Proof</u> We observe for any x, y

$$
\begin{array}{ll}
 & true \\
= & \{\text{Excluded Middle}\} \\
 & [(x \equiv y) \vee \neg(x \equiv y)] \\
= & \{\neg \equiv\} \\
 & [(x \equiv y) \vee (\neg x \equiv y)] \\
= & \{\vee \text{ distributes over } \equiv\} \\
 & [x \vee \neg x \equiv x \vee y \equiv y \vee \neg x \equiv y \vee y] \\
= & \{\text{Excluded Middle; } \vee \text{ idempotent}\} \\
 & [x \vee y \equiv y \vee \neg x \equiv y] \\
= & \{\equiv \vee \text{ symmetric; } \underline{\Rightarrow \text{ in } \vee \text{ and } \equiv}\} \\
 & [x \Rightarrow y \equiv \neg x \vee y]
\end{array}
$$

(End of Proof)

An immediate consequence is what is known as the Law of the Contrapositive

<u>Th.</u> $[x \Rightarrow y \;\equiv\; \neg x \Leftarrow \neg y]$.

Furthermore the "Laws of de Morgan" are important

<u>Th.</u> $[\neg x \vee \neg y \;\equiv\; \neg(x \wedge y)]$

<u>Th.</u> $[\neg x \wedge \neg y \;\equiv\; \neg(x \vee y)]$.

<u>Proof.</u> We prove the first one by observing for any x,y :

$$
\begin{array}{ll}
 & \neg x \vee \neg y \\
= & \{\Rightarrow\ \neg\ \vee\} \\
 & x \Rightarrow \neg y \\
= & \{\Rightarrow\ \vee\ \equiv\} \\
 & x \vee \neg y \equiv \neg y \\
= & \{\Rightarrow\ \neg\ \vee\} \\
 & y \Rightarrow x \;\equiv\; \neg y \\
= & \{\Rightarrow\ \vee\ \equiv\} \\
 & y \vee x \equiv x \equiv \neg y \\
= & \{\neg\ \equiv\} \\
 & \neg\,(y \vee x \equiv x \equiv y) \\
= & \{\text{Golden Rule}\} \\
 & \neg\,(x \wedge y)
\end{array}
$$

. (End of Proof).

A useful equivalence is

Th. $[x \wedge y \Rightarrow z \;\equiv\; x \Rightarrow \neg y \vee z]$;

an appeal to it is called "shunting".

Closely connected to the negation is the boolean constant "false". It is connected to true by

<u>Ax</u>. $[false \equiv \neg true]$

and to the everywhere operator by

<u>Ax</u>. $[false] \equiv false$.

We leave to the reader the proofs of

<u>Th</u>. $[false \equiv x \equiv \neg x]$
<u>Th</u>. $[false \vee x \equiv x]$
<u>Th</u>. $[false \wedge x \equiv false]$.

* * *

We write the quantified expressions as follows. For the universally quantified expression we write

$$\langle \forall x : r.x : t.x \rangle \quad ;$$

for the existentially quantified expression we write

$$\langle \exists x : r.x : t.x \rangle \quad .$$

Instead of the angular brackets $\langle\rangle$, one can also use the normal parentheses () without introducing syntactic ambiguity. For many years I used, in fact, the normal parentheses; later I learned to appreciate the angular brackets as a welcome visual aid to parsing. In the above schemata:

x -called "the dummy"- is a local variable of the quantified expression; the angular brackets delineate the scope of the dummy, whose type we tend to define in the environment of the quantified expression; instead of a single dummy we admit a list of dummies, say "x,y,z" instead of just "x".

r.x stands for an expression of type "predicate" which, in general, depends on the dummy. (Using the infix dot - with the highest binding power - to denote functional application, we denoted the expression in functional form, all other syntactic forms of a predicate expression are allowed as well.) The expression represented in the schemata by r.x is called "the range".

t.x - called "the term" - is also an expression of type "predicate" which, like the range, in general, depends on the dummy.

Universal quantification and existential quantification are for all r, t connected by

$\underline{Ax}.\ \ [\langle \exists x : r.x : t.x \rangle \equiv \neg\langle \forall x : r.x : \neg t.x \rangle]$.

The syntactic categories "range" and "term" are inspired by the analogy with the notation for summation that would denote the sum of the squares of the first K natural numbers by

$$\langle \Sigma n : 0 \leq n \wedge n < K : n^2 \rangle \quad ,$$

the only difference being that here the term is not of type predicate (but of type integer).

With the term of type predicate , we could restrict ourselves to ranges that are true ; the following two rewrite rules are known under the name "trading":

<u>Ax</u> $[\langle\forall x: r.x: t.x\rangle \equiv \langle\forall x: true: r.x \Rightarrow t.x\rangle]$

<u>Th</u> $[\langle\exists x: r.x: t.x\rangle \equiv \langle\exists x: true: r.x \wedge t.x\rangle]$

<u>Note</u> It is customary to omit the range (and to write :: after the dummy) if the range is true or constant all through a calculation. (End of Note.)

The following rewrite rules, known under the names "nesting" and "unnesting", are in a way the analogue of the associativity of conjunction and disjunction:

<u>Ax</u> $[\langle\forall x,y: r.x \wedge s.x.y: t.x.y\rangle \equiv$
$\langle\forall x: r.x: \langle\forall y: s.x.y: t.x.y\rangle\rangle]$

<u>Th</u> $[\langle\exists x,y: r.x \wedge s.x.y: t.x.y\rangle \equiv$
$\langle\exists x: r.x: \langle\exists y: s.x.y: t.x.y\rangle\rangle]$

As a consequence we have the following "interchanges" of quantification

<u>Th</u> $[\langle\forall x: r.x: \langle\forall y: s.y: t.x.y\rangle\rangle \equiv$
$\langle\forall y: s.y: \langle\forall x: r.x: t.x.y\rangle\rangle]$

and similarly for the existential quantification.

The analogue of the distribution of [] over $\wedge$ is given by

<u>Ax</u> $[\langle\forall x: [r.x]: t.x\rangle] \equiv \langle\forall x: [r.x]: [t.x]\rangle$.

It is referred to by "interchange" or "[] over $\forall$"; the [ ] around r.x indicate that the range has to be a boolean scalar. Note the <u>absence</u> of "[ ] over $\exists$"!

The following two are known as "splitting the term":

<u>Ax</u> $[\langle\forall x: r.x: s.x \wedge t.x\rangle \equiv$
$\langle\forall x: r.x: s.x\rangle \wedge \langle\forall x: r.x: t.x\rangle]$

<u>Th</u> $[\langle\exists x: r.x: s.x \vee t.x\rangle \equiv$
$\langle\exists x: r.x: s.x\rangle \vee \langle\exists x: r.x: t.x\rangle]$

and with trading we derive the two known as "splitting the range" – note that in <u>both</u> cases the range being split is a disjunction –

<u>Th</u> $[\langle\forall x: r.x \vee s.x: t.x\rangle \equiv$
$\langle\forall x: r.x: t.x\rangle \wedge \langle\forall x: s.x: t.x\rangle]$

<u>Th</u> $[\langle\exists x: r.x \vee s.x: t.x\rangle \equiv$
$\langle\exists x: r.x: t.x\rangle \vee \langle\exists x: s.x: t.x\rangle]$

Important are the distributions known as "$\vee$ over $\forall$":

<u>Ax</u> $[q \vee \langle\forall x: r.x: t.x\rangle \equiv \langle\forall x: r.x: q \vee t.x\rangle]$

and as "$\wedge$ over $\exists$":

<u>Th</u> $[q \wedge \langle \exists x : r.x : t.x \rangle \equiv \langle \exists x : r.x : q \wedge t.x \rangle]$.

Of the following theorems we prove the first

<u>Th</u> $[\langle \forall x : false : t.x \rangle \equiv true]$

<u>Th</u> $[\langle \exists x : false : t.x \rangle \equiv false]$.

<u>Proof</u> We observe for any t

$$
\begin{array}{ll}
 & \langle \forall x : false : t.x \rangle \\
= & \quad \{\text{trading}\} \\
 & \langle \forall x : true : false \Rightarrow t.x \rangle \\
= & \quad \{\text{predicate calculus}\} \\
 & \langle \forall x : true : true \vee t.x \rangle \\
= & \quad \{\vee \text{ over } \forall\} \\
 & true \vee \langle \forall x : true : t.x \rangle \\
= & \quad \{\text{predicate calculus}\} \\
 & true .
\end{array}
$$

(End of Proof.)

The above two theorems deal with "empty ranges". For nonempty ranges - symbolized by $r.x \vee [x = y]$ - we have

<u>Th</u> $[q \wedge \langle \forall x : r.x \vee [x = y] : t.x \rangle \equiv$
$\langle \forall x : r.x \vee [x = y] : q \wedge t.x \rangle]$

<u>Th</u> $[q \vee \langle \exists x : r.x \vee [x = y] : t.x \rangle \equiv$
$\langle \exists x : r.x \vee [x = y] : q \vee t.x \rangle]$

They follow from the "1-point rule":

<u>Ax</u> $[\langle\forall x: [x=y]: t.x\rangle \equiv t.y]$

<u>Th</u> $[\langle\exists x: [x=y]: t.x\rangle \equiv t.y]$.

We shall prove the first, after establishing the Lemma

$$[\langle\forall x: r.x \vee [x=y]: q\rangle \equiv q]$$

<u>Proof</u> We observe

$[\langle\forall x: r.x \vee [x=y]: q\rangle \equiv q]$
= {splitting the range}
$[\langle\forall x: r.x: q\rangle \wedge \langle\forall x: [x=y]: q\rangle \equiv q]$
= {1-point rule}
$[\langle\forall x: r.x: q\rangle \wedge q \equiv q]$
= {predicate calculus}
$[\neg q \vee \langle\forall x: r.x: q\rangle]$
= {$\vee$ over $\forall$}
$[\langle\forall x: r.x: \neg q \vee q\rangle]$
= {Excluded Middle}
$[\langle\forall x: r.x: true\rangle]$
= {trading}
$[\langle\forall x: false: \neg r.x\rangle]$
= {empty range}
true .

(End of Proof of Lemma)

The first unproved theorem now follows:

$q \wedge \langle\forall x: r.x \vee [x=y]: t.x\rangle$
= {above Lemma}

$$\langle \forall x: r.x \vee [x=y]: q \rangle \wedge \langle \forall x: r.x \vee [x=y]: t.x \rangle$$
$=$ {splitting the term}
$$\langle \forall x: r.x \vee [x=y]: q \wedge t.x \rangle \quad .$$

* * *

We have introduced the everywhere operator [ ] as a function from boolean structures to boolean scalars, and have used it all the time, without being very specific about its properties. We recall

$$[true] \equiv true \qquad \text{and} \qquad [false] \equiv false \ ,$$

from which follows that the everywhere operator is idempotent:

<u>Th</u>. $\qquad [[x]] \equiv [x] \quad .$

We also conclude from the above

<u>Th</u> $\qquad [x \vee [y]] \equiv [x] \vee [y] \quad ,$

which is not a very useful theorem in this form; it is a little bit more palatable as

$$[x] \Rightarrow [y] \ \equiv [\, [x] \Rightarrow y] \quad .$$

A function that for <u>any</u> scalar range distributes over universal quantification, i.e. an f for which for <u>any</u> scalar range

$$[\, f.\langle \forall x :: x \rangle \equiv \langle \forall x :: f.x \rangle]$$

holds, is called "universally conjunctive". (If it distributes for <u>any</u> scalar range over existential quantification, i.e.

$$[f.\langle \exists x :: x\rangle \equiv \langle \exists x :: f.x\rangle] \quad ,$$

it is said to be "universally disjunctive".)

Boolean structures are also called "predicates" and functions from predicates to predicates — like the above f — are known as "predicate transformers". Monotonicity of predicate transformers is defined by

$$(f \text{ is monotonic}) \equiv$$
$$\langle \forall x,y :: [x \Rightarrow y] \Rightarrow [f.x \Rightarrow f.y]\rangle$$

and the connection with the above is that a (con- or dis-)junctive function is monotonic. We shall show this for a function f that distributes over conjunction. We observe for any x, y

$$\begin{array}{ll} & [f.x \Rightarrow f.y] \\ = & \quad \{\text{predicate calculus}\} \\ & [f.x \wedge f.y \equiv f.x] \\ = & \quad \{f \text{ distributes over } \wedge\} \\ & [f.(x \wedge y) \equiv f.x] \\ \Leftarrow & \quad \{\text{Leibniz}\} \\ & [x \wedge y \equiv x] \\ = & \quad \{\text{predicate calculus}\} \\ & [x \Rightarrow y] \quad . \end{array}$$

Since a universally conjunctive function distributes over $\wedge$, and $[\ ]$ is universally conjunctive, we conclude

<u>Th</u> $\quad [x \wedge y] \equiv [x] \wedge [y]$,

from which the monotonicity now follows

<u>Th</u> $\quad [x \Rightarrow y] \Rightarrow ([x] \Rightarrow [y])$.

<u>Remark</u> Of the two ways of rewriting an equivalence

$$[x \equiv y \equiv (x \Rightarrow y) \wedge (x \Leftarrow y)] \quad \text{and}$$
$$[x \equiv y \equiv (x \wedge y) \vee (\neg x \wedge \neg y)] \quad ,$$

the former - known as "mutual implication" - is much more popular than the latter. The explanation is to be found in the distributive properties of [ ], which yield

$$[x \equiv y] \equiv [x \Rightarrow y] \wedge [x \Leftarrow y] \quad ,$$

i.e. mutual implication translates a proof obligation of the form $[x \equiv y]$ to two indepent proof obligations, while rewriting with

$$[x \equiv y] \equiv [(x \wedge y) \vee (\neg x \wedge \neg y)]$$

does not yield such disentanglement. (End of Remark.)

In view of the monotonicity of [ ] the rule of Leibniz

$$[x \equiv y] \Rightarrow [f.x \equiv f.y]$$

follows from

$$[(x \equiv y) \Rightarrow (f.x \equiv f.y)] \quad ;$$

a function f satisfying for any x,y the latter stronger relation is called a "punctual function". Since

$$[(x \equiv y) \Rightarrow (f.x \equiv f.y)]$$
$$= \quad \{\text{predicate calculus}\}$$
$$[(x \equiv y) \wedge (f.x \equiv f.y) \equiv (x \equiv y)]$$
$$= \quad \{\text{predicate calculus: } \wedge \text{ and } \equiv\}$$
$$[(x \equiv y) \wedge f.x \equiv (x \equiv y) \wedge f.y]$$

an alternative definition of punctuality is

$$(f \text{ is punctual}) \equiv$$
$$\langle \forall x,y :: [(x \equiv y) \wedge f.x \equiv (x \equiv y) \wedge f.y] \rangle \quad .$$

We are now ready to demonstrate that expressions formed from variables with the logical operators and quantification are punctual functions of the variables. Since negation and existential quantification — which includes disjunction — suffice to write down the expressions, it suffices to show that

(i) the identity function is punctual
(ii) the negation of a punctual function is punctual
(iii) an existential quantification over punctual functions is punctual .

<u>Proof</u> For (i) we observe $[(x \equiv y) \Rightarrow (x \equiv y)]$.

For (ii) we observe

$$[(x\equiv y) \Rightarrow (f.x \equiv f.y)] \equiv [(x\equiv y) \Rightarrow (\neg f.x \equiv \neg f.y)] \quad .$$

For (iii) we observe - with f ranging over some set of punctual functions -

$$
\begin{array}{ll}
 & (x\equiv y) \wedge \langle \exists f :: f.x\rangle \\
= & \quad \{ \wedge \text{ over } \exists \} \\
 & \langle \exists f :: (x\equiv y) \wedge f.x\rangle \\
= & \quad \{ f \text{ is punctual} \} \\
 & \langle \exists f :: (x\equiv y) \wedge f.y\rangle \\
= & \quad \{ \wedge \text{ over } \exists \} \\
 & (x\equiv y) \wedge \langle \exists f :: f.y\rangle \quad .
\end{array}
$$

(End of Proof)

By way of illustration of the use of punctuality we now give another proof of the transitivity of the implication. We observe for any x,y,z

$$
\begin{array}{ll}
 & (x\Rightarrow y) \wedge (y\Rightarrow z) \wedge (x\Rightarrow z) \\
= & \quad \{ \text{predicate calculus} \} \\
 & (x\wedge y \equiv x) \wedge (y\wedge z \equiv y) \wedge (x\wedge z \equiv x) \\
= & \quad \{ x\wedge z \equiv x \text{ is a punctual function of } x \} \\
 & (x\wedge y \equiv x) \wedge (y\wedge z \equiv y) \wedge (x\wedge y\wedge z \equiv x\wedge y) \\
= & \quad \{ x\wedge (y\wedge z) \equiv x\wedge y \text{ is a punctual function} \\
 & \quad\ \text{of } (y\wedge z) \} \\
 & (x\wedge y \equiv x) \wedge (y\wedge z \equiv y) \wedge (x\wedge y \equiv x\wedge y) \\
= & \quad \{ \text{predicate calculus} \} \\
 & (x\Rightarrow y) \wedge (y\Rightarrow z) \quad ,
\end{array}
$$

hence $\quad (x\Rightarrow y) \wedge (y\Rightarrow z) \Rightarrow (x\Rightarrow z) \quad .$

* * *

The "Galois connection" is a property than an ordered pair of predicate transformers may enjoy. For the ordered pair (f,g) this state of affairs is denoted by $gal.(f,g)$, which is defined by

$$gal.(f,g) \equiv \langle \forall x,y :: [f.x \Rightarrow y] \equiv [x \Rightarrow g.y] \rangle \quad .$$

The central theorem about the Galois connection states that the following 3 assertions are equivalent:

(i) $gal.(f,g)$

(ii) f is universally disjunctive and $[g.y \equiv \langle \exists x : [f.x \Rightarrow y] : x \rangle]$ for all y

(iii) g is universally conjunctive and $[f.x \equiv \langle \forall y : [x \Rightarrow g.y] : y \rangle]$ for all x.

<u>Proof</u> The proof of (i) ≡ (ii) is left to the reader; we prove (i) ≡ (iii) by mutual implication.

<u>(i) ⇒ (iii)</u>

The universal conjunctivity of g states that for any scalar range of y

$$[g.\langle \forall y :: y \rangle \equiv \langle \forall y :: g.y \rangle] \quad ;$$

this will be demonstrated by showing that for any x $[x \Rightarrow g.\langle \forall y :: y \rangle] \equiv [x \Rightarrow \langle \forall y :: g.y \rangle]$.

To this end we observe, using $gal.(f,g)$

$$
\begin{array}{ll}
 & [x \Rightarrow g.\langle\forall y :: y\rangle] \\
= & \quad \{\text{gal.}(f,g)\} \\
 & [f.x \Rightarrow \langle\forall y :: y\rangle] \\
= & \quad \{\text{pred. calc.}: \vee \text{ over } \forall\} \\
 & [\langle\forall y :: f.x \Rightarrow y\rangle] \\
= & \quad \{[\,] \text{ over } \forall\} \\
 & \langle\forall y :: [f.x \Rightarrow y]\rangle \\
= & \quad \{\text{gal.}(f,g)\} \\
 & \langle\forall y :: [x \Rightarrow g.y]\rangle \\
= & \quad \{[\,] \text{ over } \forall\} \\
 & [\langle\forall y :: x \Rightarrow g.y\rangle] \\
= & \quad \{\vee \text{ over } \forall\} \\
 & [x \Rightarrow \langle\forall y :: g.y\rangle] \quad .
\end{array}
$$

In order to prove the second conjunct of (iii), using gal.(f,g), we observe for any x

$$
\begin{array}{ll}
 & \langle\forall y : [x \Rightarrow g.y] : y\rangle \\
= & \quad \{\text{gal.}(f,g)\} \\
 & \langle\forall y : [f.x \Rightarrow y] : y\rangle \\
= & \quad \{\text{predicate calculus}\} \\
 & f.x \quad .
\end{array}
$$

<u>(i) $\Leftarrow$ (iii)</u>

Using (iii), we show

$$[f.x \Rightarrow y] \equiv [x \Rightarrow g.y] \quad \text{for all } x,y$$

by mutual implication.

LHS $\Leftarrow$ RHS

$$\langle\forall x,y :: [f.x \Rightarrow y] \Leftarrow [x \Rightarrow g.y]\rangle$$
$$= \quad \{\text{nesting, trading}\}$$
$$\langle\forall x :: \langle\forall y : [x \Rightarrow g.y] : [f.x \Rightarrow y]\rangle\rangle$$
$$= \quad \{\text{pred. calc.}\}$$
$$\langle\forall x :: [f.x \Rightarrow \langle\forall y : [x \Rightarrow g.y] : y\rangle]\rangle$$
$$= \quad \{\text{(iii), 2nd conjunct}\}$$
$$\langle\forall x :: [f.x \Rightarrow f.x]\rangle$$
$$= \quad \{\text{pred. calc}\}$$
$$true$$

LHS $\Rightarrow$ RHS

We observe for any x,y

$$[f.x \Rightarrow y]$$
$$\Rightarrow \quad \{\text{(iii), 1st conjunct, hence } g \text{ is monotonic}\}$$
$$[g.(f.x) \Rightarrow g.y]$$
$$= \quad \{\text{(iii), 2nd conjunct}\}$$
$$[g.\langle\forall y : [x \Rightarrow g.y] : y\rangle \Rightarrow g.y]$$
$$= \quad \{\text{(iii), 1st conjunct}\}$$
$$[\langle\forall y : [x \Rightarrow g.y] : g.y\rangle \Rightarrow g.y]$$
$$\Rightarrow \quad \{\text{pred. calc}\}$$
$$[x \Rightarrow g.y] \quad .$$

(End of Proof.)

Exercise Let $gal.(f,g)$. Show

(i) $f\circ g$ and $g\circ f$ are monotonic — with $\circ$ we denote functional composition: $[(f\circ g).y \equiv f.(g.y)]$.
(ii) $f\circ g$ is strengthening and $g\circ f$ is weakening

— i.e. $[(f \circ g).y \Rightarrow y]$ and $[(g \circ f).x \Leftarrow x]$ —
(iii) $f \circ g$ and $g \circ f$ are idempotent — i.e. $[(f \circ g \circ f \circ g).y \equiv (f \circ g).y]$ and $[(g \circ f \circ g \circ f).x \equiv (g \circ f).x]$. (Actually $(f \circ g \circ f) = f$, from which both idempotences follow.) (End of Exercise.)

* * *

The relational calculus emerges from the predicate calculus by extending the latter with two operators and a constant. In the context of the relational calculus, the predicates are usually called "relations".

The first additional operator is the unary "transposition". It is denoted by the prefix "~" —pronounced: "tilde"— , which is given the same syntactic binding power as the negation.

The second additional operator is the binary "composition". It is denoted by the infix ";" — pronounced: "semi"— , which is given a binding power between the binary logical operators and the unary operators.

The constant is denoted by $\mathbb{J}$; it will emerge as the identity element of the composition.

Though the axioms defining the new operators are few, the relational calculus has a much richer structure than the predicate calculus, a major difference being that, in contrast to the logical operators, the two relational operators yield expressions that are not punctual functions of their arguments.

* * *

By postulate the transposition satisfies

Ax $\quad [\sim x \Rightarrow y] \equiv [x \Rightarrow \sim y]$ for all x, y .

A shorter formulation would have been: gal.$(\sim,\sim)$. In any case we can conclude:

$\sim$ is universally conjunctive and disjunctive, hence $[\sim \text{true} \equiv \text{true}]$ and $[\sim \text{false} \equiv \text{false}]$ and $\sim$ is monotonic; $\sim$ is an involution — i.e. $[\sim\sim x \equiv x]$ for all x — because (see Exercise) $\sim\sim$ is both weakening and strengthening.

Next we show that negation and transposition distribute over each other, i.e.

Th $\quad [\sim\neg x \equiv \neg\sim x]$

Proof The proof is by mutual implication. We observe (in parallel) for any x

$[\sim\neg x \Rightarrow \neg\sim x]$	$[\sim\neg x \Leftarrow \neg\sim x]$
= {pred. calc.}	= {pred. calc}

$$\begin{array}{ll|ll}
 & [\sim\neg x \wedge \sim x \Rightarrow false] & & [\sim\neg x \vee \sim x \Leftarrow true] \\
= & \{\sim \text{ over } \wedge, \text{ for } \sim \text{ is conjunctive}\} & = & \{\sim \text{ over } \vee, \text{ for } \sim \text{ is disjunctive}\} \\
 & [\sim(\neg x \wedge x) \Rightarrow false] & & [\sim(\neg x \vee x) \Leftarrow true] \\
= & \{\text{pred. calc.}\} & = & \{\text{pred. calc.}\} \\
 & [\sim false \Rightarrow false] & & [\sim true \Leftarrow true] \\
= & \{\text{rel. calc.}\} & = & \{\text{rel. calc.}\} \\
 & true & & true
\end{array}$$

(End of Proof.)

As quantification and negation suffice for all other logical operators we conclude

<u>Th.</u> Transposition distributes over all logical operators.

<u>Th</u> $\quad [x] \equiv [\sim x]$

<u>Proof</u> We observe for any x

$$\begin{array}{ll}
 & [x] \\
= & \{\text{pred. calc.}\} \\
 & [true \Rightarrow x] \\
= & \{\text{rel. calc.}\} \\
 & [\sim true \Rightarrow x] \\
= & \{\text{gal.}(\sim,\sim)\} \\
 & [true \Rightarrow \sim x] \\
= & \{\text{pred. calc.}\} \\
 & [\sim x]
\end{array}$$

(End of Proof.)

One axiom deals with composition only:

<u>Ax</u> composition is associative, i.e.
$[(x;y);z \equiv x;(y;z)]$

The associativity is the main justification for introducing an infix operator to denote composition. We shall exploit the associativity (as usual) implicity by omission of parentheses.

The next axiom deals with a right-identity element of composition:

<u>Ax</u> $[x;J \equiv x]$.

The last axiom postulates what is known as the "right-exchange":

<u>Ax</u> $[x;y \Rightarrow z] \equiv [\sim x; \neg z \Rightarrow \neg y]$.

(The way to memorize this to observe that it is a combination of taking the contrapositive (as far as y and z are concerned) and transposing the prefix x of the antecedent.)

We now observe for any x, y, z

$$
\begin{array}{ll}
 & [\sim(x;y) \Rightarrow z] \\
= & \quad \{ J \text{ is right-identity for } ; \} \\
 & [\sim(x;y);J \Rightarrow z] \\
= & \quad \{ \text{right-exchange; associativity} \} \\
 & [x;y;\neg z \Rightarrow \neg J]
\end{array}
$$

$$
\begin{array}{ll}
= & \{\text{associativity ; right-exchange}\} \\
 & [\sim x ; J \Rightarrow \neg(y ; \neg z)] \\
= & \{J \text{ is right-identity}\} \\
 & [\sim x \Rightarrow \neg(y ; \neg z)] \\
= & \{\text{contrapositive}\} \\
 & [y ; \neg z \Rightarrow \neg\sim x] \\
= & \{\text{right-exchange}\} \\
 & [\sim y ; \sim x \Rightarrow z]
\end{array}
$$

and since the equivalence of first and last term holds for all z, we have proved our "$\sim$ over ;"

<u>Th</u> $\qquad [\sim(x;y) \equiv \sim y ; \sim x]$

<u>Remark</u> Note how 5 of the above 6 steps are unavoidable: 2 appeals to the right-identity element, viz. one to introduce J and one to eliminate J again, and 3 appeals to the right-exchange, our only tool to introduce or eliminate a tilde. (End of Remark.)

Next we observe for any x

$$
\begin{array}{ll}
 & \text{true} \\
= & \{J \text{ right-identity}\} \\
 & [\sim x ; J \equiv \sim x] \\
\Rightarrow & \{\text{Leibniz}\} \\
 & [\sim(\sim x ; J) \equiv \sim\sim x] \\
= & \{\text{rel. calc.}\} \\
 & [\sim J ; x \equiv x]
\end{array}
$$

and from this we conclude that $\sim J$ is a left-identity element of composition. Now

Th $\quad [J \equiv \sim J]$

immediately follows. We leave to the reader to derive from "$\sim$ over ;" and the right-exchange

Th $\quad [x;y \Rightarrow z] \equiv [\neg z ; \sim y \Rightarrow \neg x]$,

known as the "left-exchange".

From the exchange rules and the theory of the Galois connection we now derive that

Th Composition is universally disjunctive in both operands .

In view of the theory of the Galois connection we can conclude that, for all z, $x;z$ is universally disjunctive in its first argument, if we can establish

$$[x;z \Rightarrow y] \equiv [x \Rightarrow \text{something}] \text{ for all } x,y,z$$

where the "something" (which plays the role of "$g.y$") does not depend on x . The left-exchange does the job:

$[x;z \Rightarrow y]$
$=$ {left-exchange}
$[\neg y ; \sim z \Rightarrow \neg x]$
$=$ { contrapositive }
$[x \Rightarrow \neg(\neg y ; \sim z)]$

Similarly, the right-exchange establishes composition's universal disjunctivity in its right operand.

* * *

From its disjunctivity follows that composition is monotonic in both its arguments. From that and the existence of J , the reader may conclude

<u>Th</u> $[x \Rightarrow x;true]$ and $[x \Rightarrow true;x]$,

from which

<u>Th</u> $[true;true \equiv true]$

immediately follows.

<u>Remark</u> Hence, postfixing a relation with ";true" or prefixing it with "true;" is what is known as a closure: a monotonic, weakening, and idempotent operator. (End of Remark.)

If the implication in the other direction holds, there is something special with the relation, which is recognized by the introduction of special terminology:

$$(p \text{ is a left-condition}) \equiv [p;true \Rightarrow p]$$
$$(q \text{ is a right-condition}) \equiv [true;q \Rightarrow q] \quad .$$

Note that any left-condition p can be written as x;true for some x (for instance p).

The notions of left- and right-conditions are closely related to the last axiom of the relational calculus (to which J.C.P.S. van der Woude gave the name "Cone Rule"). In one of its formulations it is

<u>Ax</u> For left-condition p and right-condition q

$$[p \vee q] \Rightarrow [p] \vee [q] \quad .$$

(Note that the inverse implication holds for unrestricted p, q because $[\,]$ is monotonic.)

The Cone Rule is of importance because, without it, a satisfactory model of the relational calculus, is obtained by taking

- the identity function for $\sim$
- the conjunction for $;$
- true for $\mathcal{J}$.

Hence, if a lemma of the relational calculus is invalidated by the above substitution, its proof requires the Cone Rule. Conversely, that part of the relational calculus that can be derived without the Cone Rule is a true generalization of the predicate calculus. My experience with the relational calculus is undoubtedly incomplete and one-sided, but I think that, so far, I had no use for the Cone Rule. And that, of course, is very gratifying.

Left- and right-conditions emerge quite naturally from considering the standard model of the relational calculus: there a relation is modelled by a boolean function defined on the Cartesian square of some space, left- and right-conditions are modelled by either a boolean function on that (single) space or, more accurately, as a relation that does not depend on one of its two arguments. It is the mixture of boolean functions of 1 and of 2 arguments that has introduced the conditions into the relational calculus.

In the following we shall only formulate a theorem for left-conditions if a similar one holds for right-conditions. The similarity follows from

<u>Th</u> $\quad$ (p is a left-condition) $\equiv$ ($\sim p$ is a right-condition).

Important for the identification between left-conditions and boolean functions on the single space is the following

<u>Th</u> $\quad$ Expressions built from left-conditions and logical operators are left-conditions.

<u>Proof</u> By induction over the syntax it suffices to show this for negation and existential quantification, as all logical operators and constants can be expressed in those two. Our obligations are therefore:

(i) for all p
$\quad$ (p is a left-condition) $\Rightarrow$ ($\neg p$ is a left-condition)

(ii) for p ranging over a bag of relations
$\langle \forall p :: p \text{ is a left-condition} \rangle \Rightarrow (\langle \exists p :: p \rangle \text{ is a left-condition})$

Proof of (i)

$$
\begin{array}{ll}
 & (\neg p \text{ is a left-condition}) \\
= & \quad \{\text{def.}\} \\
 & [\neg p ; true \Rightarrow \neg p] \\
= & \quad \{\text{left-exchange}\} \\
 & [p ; \sim true \Rightarrow p] \\
= & \quad \{[\sim true \equiv true]\} \\
 & [p ; true \Rightarrow p] \\
= & \quad \{\text{def.}\} \\
 & (p \text{ is a left-condition})
\end{array}
$$

Proof of (ii)

$$
\begin{array}{ll}
 & (\langle \exists p :: p \rangle \text{ is a left-condition}) \\
= & \quad \{\text{def.}\} \\
 & [\langle \exists p :: p \rangle ; true \Rightarrow \langle \exists p :: p \rangle] \\
= & \quad \{ ; \text{ universally disjunctive}\} \\
 & [\langle \exists p :: p ; true \rangle \Rightarrow \langle \exists p :: p \rangle] \\
\Leftarrow & \quad \{\exists \text{ is monotonic}\} \\
 & \langle \forall p :: [p ; true \Rightarrow p] \rangle \\
= & \quad \{\text{def.}\} \\
 & \langle \forall p :: p \text{ is a left-condition} \rangle \;.
\end{array}
$$

End of Proof.

Th For all x, y, z

(i) $[(x \vee \sim x);z \Rightarrow z] \Rightarrow [x;y \wedge z \equiv x;(y \wedge z)]$

(ii) $[x;(z \vee \sim z) \Rightarrow x] \Rightarrow [x \wedge y;z \equiv (x \wedge y);z]$

Proof We only prove (ii); by transposition, (i) then follows. We are going to establish the right-hand side by mutual implication, collecting the needed conditions as we go along.

For showing $[x \wedge y;z \Leftarrow (x \wedge y);z]$ we observe

$$
\begin{array}{ll}
 & (x \wedge y);z \\
\Rightarrow & \{\text{monotonicity of } ;\} \\
 & x;z \wedge y;z \\
\Rightarrow & \{\text{condition A: } [x;z \Rightarrow x]\} \\
 & x \wedge y;z
\end{array}
$$

For showing $[x \wedge y;z \Rightarrow (x \wedge y);z]$, we show the equivalent $[y;z \Rightarrow \neg x \vee (x \wedge y);z]$ by observing

$$
\begin{array}{ll}
 & \neg x \vee (x \wedge y);z \\
\Leftarrow & \{\text{condition B: } [\neg x;z \Rightarrow \neg x]\} \\
 & \neg x;z \vee (x \wedge y);z \\
= & \{;\text{ over } \vee\} \\
 & (\neg x \vee (x \wedge y));z \\
= & \{\text{pred. calc.}\} \\
 & (\neg x \vee y);z \\
\Leftarrow & \{\text{monotonicity of } ;\} \\
 & y;z
\end{array}
$$

Turning our attention to the conditions, we observe

$$\begin{aligned}
& A \wedge B \\
= \quad & \{\text{definitions}\} \\
& [x;z \Rightarrow x] \;\wedge\; [\neg x;z \Rightarrow \neg x] \\
= \quad & \{\text{left-exchange}\} \\
& [x;z \Rightarrow x] \;\wedge\; [x;\sim z \Rightarrow x] \\
= \quad & \{\text{pred. calc.}\} \\
& [\,x;z \vee x;\sim z \;\Rightarrow\; x\,] \\
= \quad & \{\, ; \text{ over } \vee \} \\
& [\,x;(z \vee \sim z) \;\Rightarrow\; x\,]
\end{aligned}$$

and this concludes the proof.

(End of Proof.)

* * *

In the standard model of the relational calculus, the relational variables in the above range over boolean functions of 2 variables of the same type. We will denote the boolean function corresponding to the relational expression x by an infix (x). The relational calculus is then modelled by

$$\begin{aligned}
p(\neg x)q &\equiv \neg\, p(x)q \\
p(x \vee y)q &\equiv p(x)q \vee p(y)q \\
p(x \wedge y)q &\equiv p(x)q \wedge p(y)q \\
p(x \equiv y)q &\equiv p(x)q \equiv p(y)q \\
p(x \Rightarrow y)q &\equiv p(x)q \Rightarrow p(y)q \\
p(\sim x)q &\equiv q(x)p \\
p(x;y)q &\equiv \langle \exists r :: p(x)r \wedge r(y)q \rangle \\
p(J)q &\equiv p = q
\end{aligned}$$

The verification that this model satisfies all the axioms of the relational calculus is left as an exercise for the reader. These axioms, we repeat, are

(i) ";" is universally disjunctive in both arguments
(ii) ";" is associative
(iii) ";" has an identity element called J
(iv) $\sim$ is a monotonic involution
(v) $[x;y \Rightarrow \neg z] \equiv [\sim x; z \Rightarrow \neg y]$ for all x, y, z
(vi) $[x;true \vee true;y] \Rightarrow [x;true] \vee [true;y]$ for all x, y (or, alternatively, A. Tarski's formulation of the Cone Rule:
$[x;true] \vee [true;\neg x]$ for all x)

We shall now show how an alternative model is provided by the regularity calculus (also known as the calculus of regular expressions).

We interpret predicates as boolean-valued functions over some non-empty set S, and "everywhere" as universal quantification over S:

$$[x] \equiv \langle \forall s : s \in S : x.s \rangle \quad .$$

In the regularity calculus S is the set of finite strings of symbols from some alphabet. With variables r, s, and t ranging over S the composition ";" is given by

$(x;y).r \equiv \langle \exists s,t : r = s \mathbin{+\!\!+} t : x.s \wedge y.t \rangle$ for all $r \in \mathcal{S}$.

where $+\!\!+$ usually denotes concatenation. From the above definition of ";" in terms of "$+\!\!+$" follows

- ";" is universally disjunctive in both arguments
- ";" is associative $\Leftarrow$ "$+\!\!+$" is associative
- ";" has a right-identity $\Leftarrow$ "$+\!\!+$" has a right-identity.

Since concatenation is associative and has a right-identity, viz. ε, the empty string, our composition with for $+\!\!+$ the concatenation satisfies (i),(ii),(iii). Furthermore "J" is modelled by

$$J.r \equiv r = \varepsilon \qquad \text{for all } r \in \mathcal{S} \ .$$

and this J satisfies

$$[J \Rightarrow x] \equiv \neg[J \Rightarrow \neg x] \qquad \text{for all } x \ .$$

(Later we shall formulate this as: "in the regularity calculus, J is a point-predicate".)

We shall now show that this model satisfies the Cone Rule (vi), in Tarski's form:

$$[x;true] \vee [true;\neg x]$$
$= \quad \{\text{predicate calculus}\}$
$$[true \Rightarrow x;true] \vee [true \Rightarrow true;\neg x]$$
$= \quad \{J \text{ is identity element of composition}\}$
$$[J;true \Rightarrow x;true] \vee [true;J \Rightarrow true;\neg x]$$
$\Leftarrow \quad \{; \text{ is disjunctive, hence monotonic}\}$

$$\begin{array}{ll} & [J \Rightarrow x] \vee [J \Rightarrow \neg x] \\ \Leftarrow & \quad \{\text{pred. calc.}\} \\ & [J \Rightarrow x] \equiv \neg [J \Rightarrow \neg x] \\ = & \quad \{\text{this } J \text{ is a point-predicate}\} \\ & true \end{array}$$

The problem lies with (iv) and (v), which mention the $\sim$, which does not occur in the regularity calculus. Our task is to invent a $\sim$ (and, in the process, to extend the notion of concatenation) so that (iv) and (v) are met.

A way to assure that $\sim$ is a monotonic involution is to define

$$[\sim x \equiv (x \circ inv)]$$

where inv is some involution from S to S.

<u>Proof</u> To show that $\sim$ is an involution, we observe for any x

$$\begin{array}{ll} & \sim(\sim x) \\ = & \quad \{\text{def. of } \sim\} \\ & (x \circ inv) \circ inv \\ = & \quad \{\circ \text{ is associative}\} \\ & x \circ (inv \circ inv) \\ = & \quad \{inv \text{ is an involution}\} \\ & x \end{array}$$

To show monotonicity, i.e. $[x \Rightarrow y] \Rightarrow [\sim x \Rightarrow \sim y]$ we observe

$$
\begin{array}{ll}
 & [\sim x \Rightarrow \sim y] \\
= & \quad \{\text{def of } [\ ] \text{ and } \sim\} \\
 & \langle \forall s : s \in S : x.(inv.s) \Rightarrow y.(inv.s) \rangle \\
= & \quad \{\text{transforming the dummy: } inv \text{ has an inverse}\} \\
 & \langle \forall t : t \in S : x.t \Rightarrow y.t \rangle \\
= & \quad \{\text{def. of } [\ ]\} \\
 & [x \Rightarrow y]
\end{array}
$$

(End of Proof.)

We now turn our attention to (v) in order to derive the proper constraint on inv. We first investigate the expression of $[x ; y \Rightarrow \neg z]$ in terms of our model

$$
\begin{array}{ll}
 & [x ; y \Rightarrow \neg z] \\
= & \quad \{\text{model of } [\,] \text{ and of } ;\} \\
 & \langle \forall r :: \langle \exists s, t : r = s \mathbin{+\!\!+} t : x.s \wedge y.t \rangle \Rightarrow \neg z.r \rangle \\
= & \quad \{\text{predicate calculus}\} \\
 & \langle \forall r, s, t : r = s \mathbin{+\!\!+} t : \neg(x.s \wedge y.t \wedge z.r) \rangle \qquad (*)
\end{array}
$$

Thus

$$
\begin{array}{ll}
 & [\sim x ; z \Rightarrow \neg y] \\
= & \quad \{\text{above result; } (\sim x).s \equiv x.(inv.s)\} \\
 & \langle \forall r, s, t : r = s \mathbin{+\!\!+} t : \neg(x.(inv.s) \wedge z.t \wedge y.r) \rangle \\
= & \quad \{\text{changing dummies : } r, s, t := t, inv.s, r\} \\
 & \langle \forall r, s, t : t = inv.s \mathbin{+\!\!+} r : \neg(x.s \wedge y.t \wedge z.r) \rangle \qquad (**)
\end{array}
$$

We satisfy (v) by seeing to it that $(*) \equiv (**)$, an equivalence that follows from equivalence of the ranges:

$$r = s \mathbin{+\!\!+} t \quad \equiv \quad t = inv.s \mathbin{+\!\!+} r$$

By substitutions $r := \varepsilon$ and $t := \varepsilon$, respectively, we derive

$$\varepsilon = s \mathbin{+\!\!+} inv.s \quad \text{and} \quad \varepsilon = inv.s \mathbin{+\!\!+} s \quad .$$

Instead of considering all finite strings of symbols from some alphabet A, we consider a larger set of strings, composed from an alphabet twice as big. Each original letter of alphabet A is introduced in two versions –black and white, say–. Our new S is the set of all finite strings of symbols from the duplicated alphabet such that the two versions of a letter from the single alphabet don't occur next to each other. In the concatanation process, the two versions of the same symbol from the original alphabet annihilate each other; this annihilation does not destroy concatenation's associativity, nor the existence of an identity element. The operation inv is the combination of two commuting operation: reversing the string and inverting the colour of each symbol, and the functional composition of two commuting inversions is, again, an inversion. Since the standard regularity calculus does not use the negation, it is a subsystem in which we only need to distinguish between strings of symbols of one colour. The standard regularity calculus does use a star: "*"; x^* denotes the strongest solution of the equation (in y)

$$y: [y \equiv J \vee x;y] \quad .$$

<u>Intermezzo</u> By way of illustration we show the equivalence of the two formulations of the Cone Rule:

(*) $[x;true \vee true;y] \Rightarrow [x;true] \vee [true;y]$

(**) $[x;true] \vee [true;\neg x]$.

<u>(*) ⇒ (**)</u> We observe for any x

$$
\begin{array}{ll}
 & [x;true] \vee [true;\neg x] \\
\Leftarrow & \{(*) \text{ with } y:=x\} \\
 & [x;true \vee true;\neg x] \\
\Leftarrow & \{\text{monotonicity of };\} \\
 & [x;J \vee J;\neg x] \\
= & \{J \text{ is identity of };\} \\
 & [x \vee \neg x] \\
= & \{\text{pred. calc}\} \\
 & true
\end{array}
$$

<u>(*) ⇐ (**)</u> We observe for any x, y

$$
\begin{array}{ll}
 & [x;true \vee true;y] \\
= & \{\text{pred. calc.}\} \\
 & [\neg(x;true) \Rightarrow true;y] \\
\Rightarrow & \{\text{monotonicity of };\} \\
 & [true;\neg(x;true) \Rightarrow true;true;y] \\
= & \{[true;true \equiv true] \text{ (twice) and (**) with } x:=x;true\} \\
 & [true;\neg(x;true) \Rightarrow true;y] \wedge ([x;true] \vee [true;\neg(x;true)]) \\
= & \{\text{pred. calc.}\} \\
 & ([true;\neg(x;true) \Rightarrow true;y] \wedge [x;true]) \vee \\
 & ([true;\neg(x;true) \Rightarrow true;y] \wedge [true;\neg(x;true)]) \\
\Rightarrow & \{\text{pred. calc.}\} \\
 & [x;true] \vee [true;y] \quad .
\end{array}
$$

(End of Intermezzo.)

* * *

Point-Predicates

The traditional model for the predicate calculus is known as "the powerset model". The underlying space is a nonempty set S, and each predicate Q corresponds to a subset of S. The predicates that correspond to singleton sets, i.e. to the elements of S, are called "point-predicates", and being a point-predicate is formally defined by

$$(p \text{ is a point-predicate}) \equiv$$
$$\langle \forall Q :: [p \Rightarrow Q] \equiv \neg[p \Rightarrow \neg Q] \rangle \quad .$$

An immediate consequence is that *false* is <u>not</u> a point-predicate. A next consequence is that there is no predicate "between" *false* and a point-predicate, more precisely, for point-predicate p

$$[Q \Rightarrow p] \;\Rightarrow\; [Q \equiv \mathit{false}] \vee [Q \equiv p]$$

<u>Proof</u> We observe for any predicate Q and any point-predicate p

$$\begin{array}{ll}
 & [Q \Rightarrow p] \\
\Rightarrow & \quad \{ p \text{ is point-predicate} \} \\
 & [Q \Rightarrow p] \wedge ([p \Rightarrow \neg Q] \vee [p \Rightarrow Q]) \\
\Rightarrow & \quad \{\text{pred. calc.}\} \\
 & [Q \equiv \mathit{false}] \vee [Q \equiv p]
\end{array}$$

(End of Proof)

In what follows, p is understood to range over point-predicates.

The crucial decision is whether or not we add as axiom the "power set postulate" – "PSP" for short –

(PSP) $\quad [\langle\exists p::p\rangle]$

which postulates the existence of so many "points" that they fill up the underlying space. An alternative formulation is

(PSP') $\quad [\langle\exists p:[p\Rightarrow Q]:p\rangle \Leftarrow Q]$ for all Q

(Note that $[\langle\exists p:[p\Rightarrow Q]:p\rangle \Rightarrow Q]$ for all Q is a direct consequence of predicate calculus.)

<u>Proof</u> The proof is by mutual implication.

(PSP) $\Leftarrow$ (PSP') By instantiating PSP' with $Q := true$

(PSP) $\Rightarrow$ (PSP') We observe for any R

$[\langle\exists p::p\rangle]$

= {definition of point-predicate}

$[\langle\exists p:\langle\forall Q::[p\Rightarrow Q]\equiv\neg[p\Rightarrow\neg Q]\rangle:p\rangle]$

$\Rightarrow$ {instantiate $\forall$ with $Q:=R$; $\exists$ monotonic in range}

$[\langle\exists p:[p\Rightarrow R]\equiv\neg[p\Rightarrow\neg R]:p\rangle]$

$\Rightarrow$ {pred. calc.}

$[\langle\exists p:[p\Rightarrow R]\vee[p\Rightarrow\neg R]:p\rangle]$

= {range split}

$[\langle\exists p:[p\Rightarrow R]:p\rangle \vee \langle\exists p:[p\Rightarrow\neg R]:p\rangle]$

$\Rightarrow$ { pred. calc., as noted above}

$[\langle\exists p:[p\Rightarrow R]:p\rangle \vee \neg R\rangle$

$$= \quad \{\text{pred. calc.}\}$$
$$[\langle \exists p: [p \Rightarrow R]: p \rangle \Leftarrow R] \quad .$$

(End of Proof.)

Point-predicates are of interest, as they provide a concept in terms of which the relational calculus and the regularity calculus can be distinguished: in the latter, J is —as we have seen— a point-predicate, whereas in the former it is not. More interesting, however, is how much can be proved without PSP!

Remark It makes sense to talk about a predicate calculus without PSP, for PSP is truly an independent axiom, as is shown by a model of the predicate calculus that satisfies all axioms of the predicate calculus. (Rough sketch: predicates are associated with sets of open intervals of the real-number line such the their union equals the interior of the closure of that union. Negation, disjunction, and conjunction are modelled by complement, union, and intersection respectively, but always followed by the idempotent operation of taking the interior of the closure. The catchword in the mathematical literature seems to be "regular open". Unless I am mistaken, this construction can be generalized for the relational calculus in such a way that J still exists, for instance by confining the regularization operation (of taking the interior of the closure) to the direction parallel to the diagonal.) (End of Remark.)

The calculi without PSP are known as "pointless predicate calculus" or "pointless logic" and "pointless relational calculus" respectively, and it is amazing —over the years, it was an eye-opener at least for me!— how much can be done in these pointless calculi.

I knew, for instance, what I thought were three equivalent expressions for the well-foundedness of a relation

- the validity of a proof by mathematical induction
- the existence of minimal elements for nonempty subsets
- the finiteness of all decreasing sequences of elements.

In contrast to the last two definitions, which explicitly mention "elements", the first one admits a pointless formulation:

$$(S \text{ is left-wellfounded}) \equiv$$
$$\langle \forall P :\ [P \Rightarrow false] \Leftarrow [P \Rightarrow S;P] \rangle \quad ,$$

a pointless definition of well-foundedness that sufficed to prove a number of fundamental theorems about well-foundedness quite elegantly.

In reasoning about programs, the annotations play the role of the predicates, machine states the role of "points". Machine states are what individual computations are about, and this observation gives us another way of appreciating the transition from set theory to the "pointless logic":

without PSP, the machine states have disappeared from the picture and, of necessity, our considerations become nonoperational.

Acknowledgements For direct or indirect contributions I thank A. Tarski, C.A.R. Hoare, He Jifeng, Stephen Brien, Carel S. Scholten, W.H.J. Feijen, A.J.M. van Gasteren, Rutger M. Dijkstra, W.H. Hesselink, Josyula R. Rao, David A. Naumann and Roland C. Backhouse and his collaborators.

References

A. Tarski, "On the calculus of relations", Journal of Symbolic Logic 6,3 (1941) 73-89

Jaap van der Woude, "Calculations with Relations, an Example" in Beauty is our Business, eds. W.H.J. Feijen, A.J.M. van Gasteren, D. Gries, J. Misra (Springer-Verlag, 1990) pp. 435-441

C.A.R. Hoare and He Jifeng, Fundamenta Informaticae 9: pp. 51-84, 217-252, 1986

W.H.J. Feijen and A.J.M. van Gasteren "An Introduction to the Relational Calculus" in "C.S. Scholten Dedicata: Van oude machines en nieuwe rekenwijzen", eds. W.H.J. Feijen and A.J.M. van Gasteren, Academic Service, 1991, pp. 57-81

Edsger W. Dijkstra and Carel S. Scholten, "Predicate Calculus and Program Semantics", Springer-Verlag, 1990

Roland C. Backhouse et al., "A Relational Theory of Datatypes", December 13, 1990, Department of Mathematics and Computing Science, Eindhoven University of Technology, The Netherlands

Rutger M. Dijkstra, Private Communications.

This note has been specially written for the 1992 International Summerschool in Marktoberdorf, Germany

Austin, 10 June 1992

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