Calculational Mathematics

Writings on the predicate, relational, and other calculi

by
Edsger W. Dijkstra
Wim Feijen
Netty van Gasteren

Predicate Calculus

EWD1123	The unification of three calculi
EWD1160	How subtypes should enter the picture
EWD1184	Our book's omission on quantification over scalar subtypes

Relational Calculus

EWD1047	A relational summary
AvG92 / WF140	An introduction into the relational calculus
EWD1136	A comparison of relational proofs
EWD1139	A relational bagatelle
EWD1141	Notational considerations and the relational calculus

Other Calculi

WF122	Untitled summary of various calculi
WF175	Playing with dagger and star, i.e. with transitive closures
WF211	A very beginning of lattice theory
AvG136 / WF227	Exercises in Calculating

Calculational Proof Design

WF147	Designing a proof for R.S.Bird's theorem on pre-orders
WF148	One down for the relational calculus (whether you like it or not)
EWD1143	Some annotated proofs
EWD1144	More annotated proofs (a sequel to 1143)
EWD1150	On the design of calculational proofs
WF219	A very clever and impressive (but highly intractable) proof of a beautiful theorem

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
  ¬ (not, negation)
- the binary (infix) operators
  ∨ (or, disjunction), ∧ (and, conjunction)
  ⇒ (implies, implication), ⇐ (follows from, consequence)
  ≡ (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 ∨ and ∧ .

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 "⟨∀s: s∈S: (" we write "[" and instead of ").s⟩" we write "]" .

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

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

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

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

* * *

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 postu-

lated 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.

Legenda The axioms and theorems that follow should be universally quantified over their global variables of type boolean structure. The marks Ax and Th 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

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

Remark 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 $\bullet$ – .

Ax. $\equiv$ is associative

Remark 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.)

Ax. $\equiv$ is symmetric

Ax. $[x \equiv true \equiv x]$ 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.)

Th. $[true] \equiv true$.

For the disjunction $\vee$ we give the axioms

Ax. $\vee$ is associative
Ax. $\vee$ is symmetric
Ax. $\vee$ is idempotent .

For equivalence and disjunction together we give

Ax. $\vee$ distributes over $\equiv$, i.e.
$[x \vee (y \equiv z) \equiv x \vee y \equiv x \vee z]$.

Th. $[x \vee true \equiv true]$ or $[x \vee true]$.

Proof We observe for any boolean structures x, y

$x \vee true$
$=$ { Ax. with $x := y$ yields $[(y \equiv y) \equiv true]$ }
$x \vee (y \equiv y)$
$=$ { $\vee$ dist. over $\equiv$ }
$x \vee y \equiv x \vee y$
$=$ { Ax. with $x := x \vee y$ }
$true$.

(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

Ax. $[x \wedge y \equiv x \equiv y \equiv x \vee y]$,

an axiom that is also known as The Golden Rule.

Th. $\wedge$ is associative

Th. $\wedge$ is symmetric

Th. $\wedge$ is idempotent

Th. $[x \wedge true \equiv x]$

Proof. 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.)

Th. $[x \wedge (y \equiv z) \equiv x \wedge y \equiv x \wedge z \equiv x]$

Proof 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 $\equiv$ }

$$
\begin{array}{ll}
 & x \equiv y \equiv z \equiv x \vee (y \equiv z) \\
= & \quad \{ \text{Golden Rule} \} \\
 & x \wedge (y \equiv z) \quad .
\end{array}
$$

(End of Proof.)

A corollary of the last theorem is

Th. $\wedge$ distributes over "$\equiv \equiv$", i.e.

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

Th. $\wedge$ and $\vee$ distribute over each other

Proof 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

$$
\begin{array}{ll}
 & x \wedge (y \vee z) \\
= & \quad \{ \text{Golden Rule} \} \\
 & x \wedge (y \equiv z \equiv y \wedge z) \\
= & \quad \{ \wedge \text{ dist. over } \equiv \equiv \} \\
 & x \wedge y \equiv x \wedge z \equiv x \wedge y \wedge z \\
= & \quad \{ \text{properties of } \wedge \} \\
 & x \wedge y \equiv x \wedge z \equiv x \wedge y \wedge x \wedge z \\
= & \quad \{ \text{Golden Rule} \} \\
 & (x \wedge y) \vee (x \wedge z) \quad .
\end{array}
$$

(End of Proof.)

Th. The Laws of Absorption

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

Proof To demonstrate the first law of absorption we observe for any x, y

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

(End of Proof?)

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

Ax. $[x \Rightarrow y \equiv y \Leftarrow x]$

Ax. $[x \Rightarrow y \equiv x \vee y \equiv y]$

from which we immediately deduce with the Golden Rule

Th. $[x \Rightarrow y \equiv x \wedge y \equiv x]$.

A simple rewriting of the Laws of Absorption now yields

Th. $[x \Rightarrow x \vee y]$ and $[x \wedge y \Rightarrow x]$,

which are also known as Laws of Absorption .

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

Proof We observe for any x, y, z

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

$$
\begin{array}{ll}
= & \{\text{eliminate inner } \Rightarrow \text{ with } \wedge \equiv\} \\
& x \wedge (y \wedge z \equiv y) \equiv x \\
= & \{\wedge \text{ almost distributes over } \equiv\} \\
& x \wedge y \wedge z \equiv x \wedge y \\
= & \{\text{reintroduction of } \Rightarrow \text{ with } \wedge \equiv\} \\
& x \wedge y \Rightarrow z
\end{array}
$$

(End of Proof)

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

Proof We observe for any x, y

$$
\begin{array}{ll}
& x \wedge (x \Rightarrow y) \\
= & \{\text{eliminate } \Rightarrow \text{ with } \wedge \equiv\} \\
& x \wedge (x \wedge y \equiv x) \\
= & \{\wedge \text{ almost distributes over } \equiv\} \\
& x \wedge x \wedge y \equiv x \wedge x \equiv x \\
= & \{\wedge \text{ is idempotent}\} \\
& x \wedge y \equiv x \equiv x \\
= & \{\text{identity of } \equiv\} \\
& x \wedge y
\end{array}
$$

(End of Proof.)

Remark 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. $\Rightarrow$ is transitive, i.e.

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

Proof We observe for any x, y, z

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

(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

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

(End of Proof.)

Our earlier proofs all consisted of sequences of equivalent expressions. Now we have es-

tablished 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 ¬ (not, negation) is defined by the two axioms that connect it to the equivalence and the disjunction respectively

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

From the first axiom by itself one can deduce

Th. $[\neg x \equiv y \equiv x \equiv \neg y]$
Th. $[x \equiv \neg\neg x]$ (i.e. ¬ is an involution).

Together they yield another way to write the implication:

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

Proof We observe for any x, y

```
    true
=     { Excluded Middle }
    [(x≡y) ∨ ¬(x≡y)]
=     { ¬≡ }
    [(x≡y) ∨ (¬x ≡ y)]
=     { ∨ distributes over ≡ }
    [x ∨ ¬x ≡ x∨y ≡ y∨¬x ≡ y∨y]
=     { Excluded Middle; ∨ idempotent }
    [x ∨ y ≡ y ∨ ¬x ≡ y]
=     { ≡ ∨ symmetric; ⇒ in ∨ and ≡ }
    [x ⇒ y ≡ ¬x ∨ y]
```

(End of Proof?)

An immediate consequence is what is known

as the Law of the Contrapositive

Th. $[x \Rightarrow y \equiv \neg x \Leftarrow \neg y]$.

Furthermore the "Laws of de Morgan" are important

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

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

Proof. We prove the first one by observing for any x,y :

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

(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

Ax. [false ≡ ¬true]

and to the everywhere operator by

Ax. [false] ≡ false .

We leave to the reader the proofs of

Th. [false ≡ x ≡ ¬x]
Th. [false ∨ x ≡ x]
Th. [false ∧ x ≡ false] .

* * *

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

⟨∀x: r.x: t.x⟩ ;

for the existentially quantified expression we write

⟨∃x: r.x: t.x⟩ .

Instead of the angular brackets ⟨⟩, 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 de-

lineate 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

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

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

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

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":

Ax $\quad [\langle \forall x : r.x : t.x \rangle \equiv \langle \forall x : \text{true} : r.x \Rightarrow t.x \rangle]$

Th $\quad [\langle \exists x : r.x : t.x \rangle \equiv \langle \exists x : \text{true} : r.x \wedge t.x \rangle]$

Note 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:

Ax $\quad [\langle \forall x,y : r.x \wedge s.x.y : t.x.y \rangle \equiv$
$\langle \forall x : r.x : \langle \forall y : s.x.y : t.x.y \rangle \rangle]$

Th $\quad [\langle \exists x,y : r.x \wedge s.x.y : t.x.y \rangle \equiv$
$\langle \exists x : r.x : \langle \exists y : s.x.y : t.x.y \rangle \rangle]$

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

Th $\quad [\langle \forall x : r.x : \langle \forall y : s.y : t.x.y \rangle \rangle \equiv$
$\langle \forall y : s.y : \langle \forall x : r.x : t.x.y \rangle \rangle]$

and similarly for the existential quantification.

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

Ax $\quad [\langle \forall x: [r.x]: t.x \rangle] \equiv \langle \forall x: [r.x]: [t.x] \rangle$.

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":

Ax $\quad [\langle \forall x: r.x: s.x \wedge t.x \rangle \equiv$
$\langle \forall x: r.x: s.x \rangle \wedge \langle \forall x: r.x: t.x \rangle]$

Th $\quad [\langle \exists x: r.x: s.x \vee t.x \rangle \equiv$
$\langle \exists x: r.x: s.x \rangle \vee \langle \exists x: r.x: t.x \rangle]$

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 -

Th $\quad [\langle \forall x: r.x \vee s.x: t.x \rangle \equiv$
$\langle \forall x: r.x: t.x \rangle \wedge \langle \forall x: s.x: t.x \rangle]$

Th $\quad [\langle \exists x: r.x \vee s.x: t.x \rangle \equiv$
$\langle \exists x: r.x: t.x \rangle \vee \langle \exists x: s.x: t.x \rangle]$

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

Ax $\quad [q \vee \langle \forall x: r.x: t.x \rangle \equiv \langle \forall x: r.x: q \vee t.x \rangle]$

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

Th $\quad [\, q \wedge \langle \exists x: r.x: t.x \rangle \equiv \langle \exists x: r.x: q \wedge t.x \rangle \,]$.

Of the following theorems we prove the first

Th $\quad [\langle \forall x: \text{false}: t.x \rangle \equiv \text{true}\,]$

Th $\quad [\langle \exists x: \text{false}: t.x \rangle \equiv \text{false}\,]$.

Proof We observe for any t

$\langle \forall x: \text{false}: t.x \rangle$

$=$ {trading}

$\langle \forall x: \text{true}: \text{false} \Rightarrow t.x \rangle$

$=$ {predicate calculus}

$\langle \forall x: \text{true}: \text{true} \vee t.x \rangle$

$=$ {$\vee$ over $\forall$}

$\text{true} \vee \langle \forall x: \text{true}: t.x \rangle$

$=$ {predicate calculus}

true.

(End of Proof.)

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

Th $\quad [\, q \wedge \langle \forall x: r.x \vee [x=y]: t.x \rangle \equiv$

$\langle \forall x: r.x \vee [x=y]: q \wedge t.x \rangle \,]$

Th $\quad [\, q \vee \langle \exists x: r.x \vee [x=y]: t.x \rangle \equiv$

$\langle \exists x: r.x \vee [x=y]: q \vee t.x \rangle \,]$

They follow from the "1-point rule":

Ax $[\langle \forall x: [x=y]: t.x \rangle \equiv t.y]$

Th $[\langle \exists x: [x=y]: t.x \rangle \equiv t.y]$.

We shall prove the first, after establishing the Lemma

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

Proof 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}

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

* * *

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

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

from which follows that the everywhere operator is idempotent:

Th. $$[[x]] \equiv [x] \quad .$$

We also conclude from the above

Th $$[x \vee [y]] \equiv [x] \vee [y] \quad ,$$

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

Th $\quad [x \wedge y] \equiv [x] \wedge [y]$,

from which the monotonicity now follows

Th $\quad [x \Rightarrow y] \Rightarrow ([x] \Rightarrow [y])$.

Remark Of the two ways of rewriting an equivalence

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

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

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

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

i.e. mutual implication translates a proof obligation of the form $[x \equiv y]$ to two independent 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

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

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 .

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

For (ii) we observe

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

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

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

(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

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

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

* * *

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

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

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.

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

(i) ⇒ (iii)

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

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

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

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

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

(i) $\Leftarrow$ (iii)

Using (iii), we show

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

by mutual implication.

LHS ⇐ RHS

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

LHS ⇒ RHS

We observe for any x,y

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

(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 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 $[\sim x \Rightarrow y] \equiv [x \Rightarrow \sim y]$ 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 $[\sim\neg x \equiv \neg\sim x]$

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

$$
\begin{array}{l|l}
\quad [\sim\neg x \Rightarrow \neg\sim x] & \quad [\sim\neg x \Leftarrow \neg\sim x] \\
= \quad \{\text{pred. calc.}\} & = \quad \{\text{pred. calc}\}
\end{array}
$$

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

(End of Proof.)

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

Th. Transposition distributes over all logical operators.

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

Proof We observe for any x

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

(End of Proof.)

One axiom deals with composition only:

Ax 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:

Ax $$[x;\rceil \equiv x]$$.

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

Ax $$[x;y \Rightarrow z] \equiv [\sim x;\neg z \Rightarrow \neg y]$$.

(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

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

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

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

Th $\quad [\sim(x;y) \equiv \sim y ; \sim x]$

Remark Note how 5 of the above 6 steps are unavoidable: 2 appeals to the right-identity element, viz. one to introduce $\mathbb{J}$ and one to eliminate $\mathbb{J}$ 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

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

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:

$$\begin{aligned} & [\, x;z \Rightarrow y \,] \\ = & \quad \{\text{left-exchange}\} \\ & [\, \neg y \,;\, \sim z \Rightarrow \neg x \,] \\ = & \quad \{\text{contrapositive}\} \\ & [\, x \Rightarrow \neg(\neg y;\sim z) \,] \end{aligned}$$

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

Th $\quad [x \Rightarrow x;\text{true}]$ and $[x \Rightarrow \text{true};x]$,

from which

Th $\quad [\text{true};\text{true} \equiv \text{true}]$

immediately follows.

Remark 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:

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

Note that any left-condition p can be written as $x;\text{true}$ 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

Ax 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 rela-

tional 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

Th (p is a left-condition) ≡ (¬p is a right-condition).

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

Th Expressions built from left-conditions and logical operators are left-conditions.

Proof 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
(p is a left-condition) ⇒ (¬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)

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

Proof of (ii)

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

End of Proof.

Th For all x, y, z

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

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

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

$(x \wedge y); z$

$\Rightarrow$ {monotonicity of ;}

$x; z \wedge y; z$

$\Rightarrow$ {condition A: $[x; z \Rightarrow x]$}

$x \wedge y; z$

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

$\neg x \vee (x \wedge y); z$

$\Leftarrow$ {condition B: $[\neg x; z \Rightarrow \neg x]$}

$\neg x; z \vee (x \wedge y); z$

$=$ { ; over $\vee$}

$(\neg x \vee (x \wedge y)); z$

$=$ {pred. calc.}

$(\neg x \vee y); z$

$\Leftarrow$ {monotonicity of ;}

$y; z$

Turning our attention to the conditions, we observe

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

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

$$
\begin{array}{lcl}
p(\neg x)q & \equiv & \neg\ p(x)q \\
p(x \vee y)q & \equiv & p(x)q \vee p(y)q \\
p(x \wedge y)q & \equiv & p(x)q \wedge p(y)q \\
p(x \equiv y)q & \equiv & p(x)q \equiv p(y)q \\
p(x \Rightarrow y)q & \equiv & p(x)q \Rightarrow p(y)q \\
p(\sim x)q & \equiv & q(x)p \\
p(x;y)q & \equiv & \langle \exists r :: p(x)r \wedge r(y)q \rangle \\
p(\mathfrak{I})q & \equiv & p = q
\end{array}
$$

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) $[x; \text{true} \vee \text{true}; y] \Rightarrow [x; \text{true}] \vee [\text{true}; y]$ for all x, y (or, alternatively, A. Tarski's formulation of the Cone Rule:
$[x; \text{true}] \vee [\text{true}; \neg x]$ 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

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

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 "$\rceil$" is modelled by

$$\rceil.r \equiv r = \varepsilon \quad \text{for all } r \in \Sigma .$$

and this $\rceil$ satisfies

$$[\rceil \Rightarrow x] \equiv \neg [\rceil \Rightarrow \neg x] \quad \text{for all } x .$$

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

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

$$\begin{array}{ll} & [x ; \text{true}] \vee [\text{true}; \neg x] \\ = & \quad \{\text{predicate calculus}\} \\ & [\text{true} \Rightarrow x ; \text{true}] \vee [\text{true} \Rightarrow \text{true}; \neg x] \\ = & \quad \{\rceil \text{ is identity element of composition}\} \\ & [\rceil ; \text{true} \Rightarrow x ; \text{true}] \vee [\text{true}; \rceil \Rightarrow \text{true}; \neg x] \\ \Leftarrow & \quad \{ ; \text{ is disjunctive, hence monotonic}\} \end{array}$$

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

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 $\mathcal{S}$ to $\mathcal{S}$.

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

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

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

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

(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

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

Thus

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

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

$$r = s \# t \quad \equiv \quad t = inv.s \# r$$

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

$$\varepsilon = S \,+\!\!+\, inv.S \quad \text{and} \quad \varepsilon = inv.S \,+\!\!+\, S \ .$$

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)

$$y: [\, y \equiv \rceil \vee x;y \,] \ .$$

Intermezzo By way of illustration we show the equivalence of the two formulations of the Cone Rule:

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

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

(*) $\Rightarrow$ (**) We observe for any x

$[x;true] \vee [true;\neg x]$

$\Leftarrow$ {(*) with $y := x$}

$[x;true \vee true;\neg x]$

$\Leftarrow$ {monotonicity of ;}

$[x;J \vee J;\neg x]$

$=$ {J is identity of ;}

$[x \vee \neg x]$

$=$ {pred. calc}

$true$

(*) $\Leftarrow$ (**) We observe for any x, y

$[x;true \vee true;y]$

$=$ {pred. calc.}

$[\neg(x;true) \Rightarrow true;y]$

$\Rightarrow$ {monotonicity of ;}

$[true;\neg(x;true) \Rightarrow true;true;y]$

$=$ {$[true;true \equiv true]$ (twice) and (**) with $x := x;true$}

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

$=$ {pred. calc.}

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

$([true;\neg(x;true) \Rightarrow true;y] \wedge [true;\neg(x;true)])$

$\Rightarrow$ {pred. calc.}

$[x;true] \vee [true;y]$.

(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

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

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

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

(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) $\qquad [\langle \exists p :: p \rangle]$

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

(PSP') $\qquad [\langle \exists p : [p \Rightarrow Q] : p \rangle \Leftarrow Q]$ 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.)

Proof The proof is by mutual implication.

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

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

$$
\begin{array}{ll}
 & [\langle \exists p :: p \rangle] \\
= & \quad \{\text{definition of point-predicate}\} \\
 & [\langle \exists p : \langle \forall Q :: [p \Rightarrow Q] \equiv \neg[p \Rightarrow \neg Q] \rangle : p \rangle] \\
\Rightarrow & \quad \{\text{instantiate } \forall \text{ with } Q := R \text{; } \exists \text{ monotonic in range}\} \\
 & [\langle \exists p : [p \Rightarrow R] \equiv \neg[p \Rightarrow \neg R] : p \rangle] \\
\Rightarrow & \quad \{\text{pred. calc.}\} \\
 & [\langle \exists p : [p \Rightarrow R] \vee [p \Rightarrow \neg R] : p \rangle] \\
= & \quad \{\text{range split}\} \\
 & [\langle \exists p : [p \Rightarrow R] : p \rangle \vee \langle \exists p : [p \Rightarrow \neg R] : p \rangle] \\
\Rightarrow & \quad \{\text{pred. calc., as noted above}\} \\
 & [\langle \exists p : [p \Rightarrow R] : p \rangle \vee \neg R \rangle
\end{array}
$$

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

(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:

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

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

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

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

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

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

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

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

$$[x \dot{\equiv} \dot{true} \dot{\equiv} x]$$

- $\dot{\vee}$ 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 $\dot{false}$ are introduced, and the properties of the everywhere operator are given — i.e. some derived and some postulated — such as

$$[\dot{true}] \equiv true$$

$$[\dot{false}] \equiv false$$

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

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

$$x \wedge (x \vee y) \equiv x \vee \mathit{false}$$

as we have in structures

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

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" $\dot{\mathit{true}}$ and $\dot{\mathit{false}}$, 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

Our book's omission on quantification over scalar subtypes

In our book [DS90], we failed to introduce scalars of type T as a subtype of structures of type T, and consequently, our use of the expression "of the same type" is sometimes ambiguous. In the rest of this text, x, y stand for dummies of type structure of type T, and c for a scalar dummy of type T.

In [AB36], Lex Bijlsma rightly points out that our text can be interpreted as suggesting the truth of the generally false

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

and that our text fails to deal explicitly with the theorem that for punctual f

(0) $$[\langle \forall c : c = x : f.c \rangle \equiv f.x]$$

(a theorem which is used!). The purpose of this note is to remedy this situation.

* * *

We shall use for the subtype relation between x and c the postulate

(1) $$\langle \forall c :: \langle \exists x :: [x = c] \rangle \rangle \quad ,$$

from which – since $[[X] \Rightarrow X]$ –

(2) $\langle \forall c :: [\langle \exists x :: x = c \rangle] \rangle$

follows. Furthermore, the fact that c is not just a subtype of the structure x but is the corresponding scalar type leads to the postulate

(3) $\langle \forall x :: [\langle \exists c :: c = x \rangle] \rangle$.

Finally we recall the definition of f's punctuality:

(4) $[\langle \forall x,y :: x=y \Rightarrow f.x = f.y \rangle]$;

for boolean f, predicate calculus allows us to rewrite (4) as

(5) $[\langle \forall x,y :: x=y \Rightarrow f.x \equiv x=y \Rightarrow f.y \rangle]$.

Just to be on the -very- safe side, we check that a formula universally quantified over x may be instantiated with x := c, i.e. we shall prove for non necessarily punctual g

(6) $[\langle \forall x :: g.x \rangle \Rightarrow \langle \forall c :: g.c \rangle]$.

To this end we observe

$$
\begin{array}{ll}
 & \langle \forall c :: g.c \rangle \\
= & \quad \{(1)\} \\
 & \langle \forall c :: \langle \exists x :: [x=c] \rangle \Rightarrow g.c \rangle \\
= & \quad \{\text{predicate calculus}\} \\
 & \langle \forall c :: \langle \forall x :: [x=c] \Rightarrow g.c \rangle \rangle
\end{array}
$$

$$
\begin{array}{ll}
= & \{\text{interchange; Leibniz}\} \\
 & \langle \forall x :: \langle \forall c :: [x=c] \Rightarrow g.x \rangle \rangle \\
= & \{\text{predicate calculus}\} \\
 & \langle \forall x :: \langle \exists c :: [x=c] \rangle \Rightarrow g.x \rangle \\
\Leftarrow & \{\text{predicate calculus}\} \\
 & \langle \forall x :: g.x \rangle
\end{array}
$$

Note The first step is not such a rabbit when we realize (i) that we have to use (1) in a strengthening chain, (ii) that we have to introduce a quantification over x, and (iii) Leibniz is needed to relate $g.c$ to $g.x$ (End of Note.)

The proof of (0) is by a ping-pong argument; pong being the easiest, we do that one first.

Proof of (0), [LHS $\Leftarrow$ RHS]

$$
\begin{array}{ll}
 & [\langle \forall c : c=x : f.c \rangle \Leftarrow f.x] \\
\Leftarrow & \{(6)\} \\
 & [\langle \forall y : y=x : f.y \rangle \Leftarrow f.x] \\
= & \{\text{pred. calc.}\} \\
 & [\langle \forall y : y=x : f.y \Leftarrow f.x \rangle] \\
\Leftarrow & \{\text{pred. calc.}\} \\
 & [\langle \forall y :: y=x \Rightarrow (f.y \equiv f.x) \rangle] \\
= & \{(4),\ f \text{ is punctual}\} \\
 & \text{true.}
\end{array}
$$

(End of Proof of (0), [LHS $\Leftarrow$ RHS].)

We have not made use yet of (3). We are going to do that by showing, in preparation of the proof of ping, that for punctual f

(7) $\quad [\langle \forall x :: f.x \rangle \equiv \langle \forall c :: f.c \rangle]$.

The proof is remarkably similar to the earlier proof of (6). We observe for punctual f :

$\langle \forall x :: f.x \rangle$

$=$ { (3) }

$\langle \forall x :: \langle \exists c :: c = x \rangle \Rightarrow f.x \rangle$

$=$ { pred. calc. }

$\langle \forall x :: \langle \forall c :: c = x \Rightarrow f.x \rangle \rangle$

$=$ { interchange; (5) & (6), f is punctual }

$\langle \forall c :: \langle \forall x :: x = c \Rightarrow f.c \rangle \rangle$

$=$ { pred. calc. }

$\langle \forall c :: \langle \exists x :: x = c \rangle \Rightarrow f.c \rangle$

$=$ { (2) }

$\langle \forall c :: f.c \rangle$,

and after this demonstration of (7), the proof of ping is a walk-over.

Proof of (0), [LHS $\Rightarrow$ RHS]

We observe for punctual f

$\langle \forall c : c = x : f.c \rangle$

$=$ { (7), $?=x$ and f both punctual }

$\langle \forall y : y = x : f.y \rangle$

$\Rightarrow$ {instantiation $y := x$}
$x = x \Rightarrow f.x$
$=$ {predicate calculus}
$f.x$

and this concludes the proof of (0).

This proof became longer than I had expected. I don't feel guilty about postulating (1) and (3) here, but it is bad that they don't occur in our book.

[AB36] A. Bijlsma, "A case of context dependence in predicate calculus", September 14, 1993, Technical University Eindhoven

[DS90] Edsger W. Dijkstra & Carel S. Scholten "Predicate Calculus and Program Semantics", Springer-Verlag, 1990

Austin, 22 August 1994

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

A relational summary

Here we present the relational calculus as a specialization of the predicate calculus: to the latter we add two operators and one constant.

To begin with we introduce a unary prefix operator, which –following Hesselink– we call the "estrangement" and denote by "$*$". It is given the same high binding power as $\neg$ – i.e. higher than all infix operators except functional application – under the usual rule that unary prefix operators are right-associative.

The estrangement is introduced by the following two axioms. (Note. A summary of formulae is given at the end of this paper.)

(0) $[P \Rightarrow *Q] \equiv [Q \Rightarrow *P]$

(1) $[P \Leftarrow *Q] \equiv [Q \Leftarrow *P]$

From (0) follow

(2) $[*\mathit{false} \equiv \mathit{true}]$ and

(3) $[*(\underline{E}Q :: Q) \equiv (\underline{A}Q :: *Q)]$.

From (1) follow

(4) $[*\mathit{true} \equiv \mathit{false}]$

(5) $[\ *(\underline{A}Q :: Q) \equiv (\underline{E}Q :: *Q)\ ]$

Proof of (2): $[\ *false \equiv true\ ]$

$= \{ pred.\ calc. \}$

$[\ true \Rightarrow *false\ ]$

$= \{ (0) \}$

$[\ false \Rightarrow *true\ ]$

$= \{ pred.\ calc. \}$

$true$ (End of Proof of (2))

Proof of (3): We observe for any P, and Q ranging over some set,

$[\ P \Rightarrow *(\underline{E}Q :: Q)\ ]$

$= \{ (0) \}$

$[\ (\underline{E}Q :: Q) \Rightarrow *P\ ]$

$= \{ pred.\ calc. \}$

$[\ (\underline{A}Q :: Q \Rightarrow *P)\ ]$

$= \{ interchange \}$

$(\underline{A}Q :: [\ Q \Rightarrow *P\ ])$

$= \{ (0) \}$

$(\underline{A}Q :: [\ P \Rightarrow *Q\ ])$

$= \{ interchange \}$

$[\ (\underline{A}Q :: P \Rightarrow *Q)\ ]$

$= \{ pred.\ calc. \}$

$[\ P \Rightarrow (\underline{A}Q :: *Q)\ ]$

and since the equivalence of first and last lines hold for any P, (3) follows.

(End of Proof of (3))

Note. Formula (2) follows from (3) by choosing for Q an empty range. (End of Note.)

From both axioms we conclude

(6) $[P \equiv *Q] \equiv [Q \equiv *P]$.

Proof
$$
\begin{aligned}
& [P \equiv *Q] \\
= & \quad \{\text{pred. calc.}\} \\
& [P \Rightarrow *Q] \wedge [P \Leftarrow *Q] \\
= & \quad \{(0) \text{ and } (1)\} \\
& [Q \Rightarrow *P] \wedge [Q \Leftarrow *P] \\
= & \quad \{\text{pred. calc.}\} \\
& [Q \equiv *P] \quad .
\end{aligned}
$$
(End of Proof.)

Instantiating (6) with $Q := *P$ yields the corollary

(7) $[P \equiv **P]$,

i.e. the estrangement is an involution.

Next we introduce an asymmetric binary operator called "composition" and denoted by an infix ";", which – following Scholten – we give a binding power less than the unary operators and higher than the binary logical operators. Composition occurs in three axioms, viz.

(8) $[P;(Q;R) \equiv (P;Q);R]$,

which states that composition is associative;

(9) $[P;Q \Rightarrow D] \equiv [P \Rightarrow *Q]$,

which links composition and estrangement to the constant D – following Hesselink called "the diversity" – , and

(10) $[*(*P ; *Q) \equiv \neg(\neg Q ; \neg P)]$,

which links composition and estrangement to the negation.

Axiom (8) we exploit by omitting semantically superfluous parentheses in continued compositions. Axiom (9), which is our only axiom about the constant D, is mainly used to rewrite implications.

(11) $[P;Q \Rightarrow D] \equiv [Q;P \Rightarrow D]$

Proof
$$
\begin{aligned}
& [P;Q \Rightarrow D] \\
= & \quad \{(9)\} \\
& [P \Rightarrow *Q] \\
= & \quad \{(0)\} \\
& [Q \Rightarrow *P] \\
= & \quad \{(9) \text{ with } P,Q := Q,P\} \\
& [Q;P \Rightarrow D] \quad .
\end{aligned}
$$
(End of Proof.)

What in (11) might look like an interchange of P and Q is better interpreted as a rotation. We leave to the reader to show, for instance, the equivalence of

(i) $[P;Q;R \Rightarrow D]$,
(ii) $[Q;R;P \Rightarrow D]$, and
(iii) $[R;P;Q \Rightarrow D]$.

The proof has to use the associativity of the composition. In hints, I intend to refer to (11) and its generalizations under the name "rotation".

Our second use of (9) is to rewrite

$$(\underline{A}Z :: [P \Rightarrow Z] \equiv [Q \Rightarrow Z]) \equiv [P \equiv Q]$$

from predicate calculus. Because $*$ is an involution – i.e. (7) – we may rewrite the left-hand side of the above as

$$(\underline{A}Z :: [P \Rightarrow *Z] \equiv [Q \Rightarrow *Z]) \quad .$$

On account of (9) we deduce

(12) $(\underline{A}Z :: [P;Z \Rightarrow D] \equiv [Q;Z \Rightarrow D]) \equiv [P \equiv Q]$.

Now it is simple to show that $*D$ is the (left- and right-hand) identity of composition, i.e.

(13) $[*D;P \equiv P]$

(14) $[P;*D \equiv P]$.

Proof. We observe for any Z

$$[*D;P;Z \Rightarrow D]$$

$$
\begin{array}{ll}
= & \{\text{rotation}\} \\
 & [P;2;{*D} \Rightarrow D] \\
= & \{(9) \text{ with } P,Q := P;2,\ {*D}\} \\
 & [P;2 \Rightarrow {**D}] \\
= & \{(7), \text{ i.e. } * \text{ is an involution}\} \\
 & [P;2 \Rightarrow D] \ ,
\end{array}
$$

which, according to (12), establishes (13). The very similar proof of (14) is left as an exercise to the reader.

(End of Proof.)

We now explore with the aid of (9) the boolean expression

$$[X;Y \Rightarrow Z] \ .$$

We observe for any X, Y, Z

$$
\begin{array}{ll}
 & [X;Y \Rightarrow Z] \\
= & \{* \text{ is an involution}\} \\
 & [X;Y \Rightarrow {**Z}] \\
= & \{(9)\} \\
 & [X;Y;{*Z} \Rightarrow D]
\end{array}
$$

$$
\begin{array}{ll}
= & \{(9)\} \\
 & [X \Rightarrow {*(Y;*Z)}]
\end{array}
\qquad\qquad
\begin{array}{ll}
= & \{\text{rotation}\} \\
 & [Y;{*Z};X \Rightarrow D] \\
= & \{(9)\} \\
 & [Y \Rightarrow {*(*Z;X)}]
\end{array}
$$

and thus we have derived

(15) $[X;Y \Rightarrow Z] \equiv [X \Rightarrow *(Y;*Z)]$

and -using $\Leftarrow$ and renaming-

(16) $[X \Leftarrow Y;Z] \equiv [*(*X;Y) \Leftarrow Z]$.

Next, we use the above to show that composition is universally disjunctive in both arguments. In order to show that composition is universally disjunctive in its left operand, we define for fixed Y the functions f and g by

$$[f.X \equiv X;Y] \quad \text{and} \quad [g.Z \equiv *(Y;*Z)] \ ,$$

and shall derive the universal disjunctivity of f from the rewritten (15):

$$[f.X \Rightarrow Z] \equiv [X \Rightarrow g.Z] \quad \text{for all } X,Z \qquad (*).$$

In order to show that composition is universally disjunctive in its right operand, we define for fixed Y the functions f and g by

$$[f.Z \equiv Y;Z] \quad \text{and} \quad [g.X \equiv *(*X;Y)] \ ,$$

and shall derive the universal disjunctivity of f from the rewritten (16):

$$[X \Leftarrow f.Z] \equiv [g.X \Leftarrow Z] \quad \text{for all } X,Z \qquad (*)$$

Note that, X and Z being dummies, the two formulae marked (*) express the same constraint

on f and g. That for an f and g satisfying (*), f is universally disjunctive is a well-known theorem, whose proof is given below. Note that in the design of this proof, one has hardly any freedom: at each step, there is only one thing one can do.

Proof In order to derive for any range of P

$$[f.(\underline{E}P::P) \equiv (\underline{E}P::f.P)]$$

we observe for any R

$$\begin{aligned}
& [f.(\underline{E}P::P) \Rightarrow R] \\
= & \quad \{(*)\} \\
& [(\underline{E}P::P) \Rightarrow g.R] \\
= & \quad \{\text{predicate calculus}\} \\
& [(\underline{A}P::P \Rightarrow g.R)] \\
= & \quad \{\text{predicate calculus}\} \\
& (\underline{A}P::[P \Rightarrow g.R]) \\
= & \quad \{(*)\} \\
& (\underline{A}P::[f.P \Rightarrow R]) \\
= & \quad \{\text{predicate calculus}\} \\
& [(\underline{A}P::f.P \Rightarrow R)] \\
= & \quad \{\text{predicate calculus}\} \\
& [(\underline{E}P::f.P) \Rightarrow R]
\end{aligned}$$

from which observation the demonstrandum follows.

(End of Proof.)

(17) ; is universally disjunctive in both operands.

The reader is supposed to be familiar with the best-known consequences of f being universally disjunctive: $[f.false \equiv false]$ and f is monotonic with respect to $\Rightarrow$.

We now turn to (10), our third and last axiom about composition. It connects estrangement and negation, and does so in a symmetrical way. One of our near purposes is to show that negation and estrangement commute, but because the constant $\mathbb{D}$ is so closely associated with composition, we first investigate whether we can find a nice, simple relation between $*$, $\neg$, and $\mathbb{D}$. Because $*$ and $\neg$ are involutions and $*\mathbb{D}$ is the identity element of composition, (10) is likely to have an instantiation that admits simplification. It has indeed. (Note that the instantiation of (10) is all but dictated by the desire to exploit at both sides that $*\mathbb{D}$ is the identity element of composition.) We observe

$$
\begin{array}{ll}
 & true \\
= & \{(10) \text{ with } P,Q := \neg * \mathbb{D}, \mathbb{D}\} \\
 & [*(*\neg * \mathbb{D} \,;\, *\mathbb{D}) \equiv \neg(\neg \mathbb{D} \,;\, \neg\neg * \mathbb{D})] \\
= & \{(14)\ ;\ \neg \text{ is an involution}\} \\
 & [**\neg * \mathbb{D} \equiv \neg(\neg \mathbb{D} \,;\, *\mathbb{D})] \\
= & \{* \text{ is an involution};\ (14)\} \\
 & [\neg * \mathbb{D} \equiv \neg\neg \mathbb{D}] \\
= & \{\neg \text{ is an involution}\}
\end{array}
$$

$$[*\mathbb{D} \equiv \neg\mathbb{D}] \quad .$$

Thus we have derived

(18) $[*\mathbb{D} \equiv \neg\mathbb{D}]$.

And now we are ready to show that $\neg$ and $*$ commute, i.e.

(19) $[\neg * X \equiv * \neg X]$.

Proof Aiming to apply (12), we observe for arbitrary X, Z

$[*\neg X ; Z \Rightarrow \mathbb{D}]$
= { $*$ is an involution, twice }
$[**(*\neg X ; ** Z) \Rightarrow \mathbb{D}]$
= { (10) with $P, Q := \neg X, * Z$ }
$[*\neg(\neg * Z ; \neg\neg X) \Rightarrow \mathbb{D}]$
= { (1) }
$[* \mathbb{D} \Rightarrow \neg(\neg * Z ; \neg\neg X)]$
= { (18) ; $\neg$ is an involution }
$[\neg\mathbb{D} \Rightarrow \neg(\neg * Z ; X)]$
= { contrapositive }
$[\neg * Z ; X \Rightarrow \mathbb{D}]$
= { (9) with $P, Q := \neg * Z, X$ }
$[\neg * Z \Rightarrow * X]$
= { contrapositive }
$[\neg * X \Rightarrow * Z]$
= { (9) with $P, Q := \neg * X, Z$ }
$[\neg * X ; Z \Rightarrow \mathbb{D}]$

Predicate Z being arbitrary, (19) now follows on account of (12) from the equivalence of the first and last terms above.

(End of Proof.)

Inspired by the commutativity of $*$ and $\neg$, we now introduce a special symbol and name for their functional composition. It is denoted by the tilde $\sim$ and called the "transposition". It is formally defined by

(20) $[\sim P \equiv *\neg P] \quad [\sim P \equiv \neg * P]$.

Also $\sim$ is an involution and it commutes with $*$ and $\neg$. More precisely

(20) $[\sim\sim P \equiv P]$

(20) $[*P \equiv \neg\sim P] \quad [*P \equiv \sim\neg P]$

(20) $[\neg P \equiv \sim * P] \quad [\neg P \equiv * \sim P]$;

the proofs are left to the reader. Formulae (10) and (18) can be rewritten as

(21) $[\sim(P;Q) \equiv \sim Q;\sim P]$

(22) $[\sim D \equiv D]$.

Finally, we introduce the constant J satisfying -see (13), (14), (18)-

(23) $[J \equiv \neg D] \quad [J \equiv *D] \quad [J \equiv \sim J]$

(24) $[J;P \equiv P] \quad [P;J \equiv P]$,

the feeling being that the identity element of the composition deserves its own name.

We observe for any range of Q

$$
\begin{aligned}
& \sim (\underline{E} Q :: Q) \\
= \quad & \{(20), \text{definition of } \sim\} \\
& \neg * (\underline{E} Q :: Q) \\
= \quad & \{(3)\} \\
& \neg (\underline{A} Q :: * Q) \\
= \quad & \{\text{de Morgan}\} \\
& (\underline{E} Q :: \neg * Q) \\
= \quad & \{(20)\} \\
& (\underline{E} Q :: \sim Q) \quad ,
\end{aligned}
$$

i.e. transposition distributes over existential quantification (i.e. is universally disjunctive). Since -see (20)- transposition distributes over negation as well and negation and existential quantification suffice for all logical expressions, we have derived

(25) transposition distributes over the logical operators and the quantifications

Remark Note that composition -see (21)- is excluded from the logical operators; composition is a relational operator. (End of Remark.)

* * *

By definition

(26) (P is a precondition) ≡ [P; true ≡ P]
(P is a postcondition) ≡ [true; P ≡ P] ;

the proof of

(27) (P is a postcondition) ≡
(~P is a precondition)

is left to the reader. In what follows we shall deal with postconditions; the exploration of the dual theorems is left to the reader.

An important theorem about postconditions is that a logical expression of postconditions is again a postcondition. As in the proof of (25), we prove it for the special cases of existential quantification and negation. First we show that if P ranges over a set of postconditions – i.e. (A P:: [true; P ≡ P]) –

[true; (E P:: P) ≡ (E P:: P)]

Proof We observe for P ranging over postconditions

true; (E P:: P)
= { ; is universally disjunctive}
(E P:: true; P)
= { P is a postcondition}
(E P:: P)

(End of Proof.)

To prove that the negation of a postcondition is a postcondition, we first observe

$$
\begin{array}{ll}
 & [\,true\,;\,P \equiv P\,] \\
= & \quad \{\text{pred. calc.}\} \\
 & [\,true\,;\,P \Rightarrow P\,] \;\wedge\; [\,true\,;\,P \Leftarrow P\,] \\
= & \quad \{(24)\} \\
 & [\,true\,;\,P \Rightarrow P\,] \;\wedge\; [\,true\,;\,P \Leftarrow \rceil\,;P\,] \\
= & \quad \{[\,true \Leftarrow \rceil\,] \text{ and } ; \text{ is monotonic}\} \\
 & [\,true\,;\,P \Rightarrow P\,] \quad .
\end{array}
$$

Hence our proof obligation is

$$[\,true\,;P \Rightarrow P\,] \equiv [\,true\,;\neg P \Rightarrow \neg P\,]$$

Proof We observe for any P

$$
\begin{array}{ll}
 & [\,true\,;\,\neg P \Rightarrow \neg P\,] \\
= & \quad \{(16) \text{ with } X,Y,Z := \neg P, true, \neg P\} \\
 & [\,*(*\neg P\,;\,true) \Leftarrow \neg P\,] \\
= & \quad \{\text{contrapositive}, (20)\} \\
 & [\,\sim(\sim P\,;\,true) \Rightarrow P\,] \\
= & \quad \{(21), (20) \text{ and } [\,true \equiv \sim true\,]\} \\
 & [\,true\,;\,P \Rightarrow P\,] \quad .
\end{array}
$$

(End of Proof.)

And thus we have established

(28) logical expressions built from postconditions are postconditions

Next we prove

(29) $[\text{true};P \Rightarrow P] \Rightarrow [X;(Y \wedge P) \equiv X;Y \wedge P]$

Proof The proof of the equivalence is by mutual implication. The one direction uses that P is a postcondition, the other one that $\neg P$ is one. In both cases we use the lemma

(*) $(Q \text{ is a postcondition}) \Rightarrow [X;Q \Rightarrow Q]$

which follows from the monotonicity of composition.

LHS $\Rightarrow$ RHS We observe for any X, Y and postcondition P

$$
\begin{array}{ll}
 & X;(Y \wedge P) \\
\Rightarrow & \{[Y \wedge P \Rightarrow Y], [Y \wedge P \Rightarrow P], \text{monotonicity of } ;\} \\
 & X;Y \wedge X;P \\
\Rightarrow & \{(*) \text{ with } Q := P; \text{ monotonicity of } \wedge\} \\
 & X;Y \wedge P
\end{array}
$$

LHS $\Leftarrow$ RHS

$$
\begin{array}{ll}
 & X;(Y \wedge P) \vee \neg P \\
\Leftarrow & \{(*) \text{ with } Q := \neg P\} \\
 & X;(Y \wedge P) \vee X;\neg P \\
= & \{; \text{ distributes over } \vee\} \\
 & X;((Y \wedge P) \vee \neg P) \\
= & \{\text{pred. calc.}\} \\
 & X;(Y \vee \neg P) \\
\Leftarrow & \{[Y \vee \neg P \Leftarrow Y] \text{ and monotonicity of } ;\} \\
 & X;Y
\end{array}
$$

(End of Proof.)

Notational Convention From now onwards the reader is supposed to be so familiar with (20) that in the use of $\neg$ $*$ and $\sim$ the composition of any two will immediately be rendered by the third. Prior to the introduction of $\sim$, one has the choice between $\neg * X$ and $* \neg X$; the introduction of the "superfluous" $\sim$ enables us to introduce $\sim X$ as the canonical representation. I should have introduced this notational convention earlier. (End of Notational Convention.)

In order to relate the above to Tarski's Axiomatization, we derive as a theorem Tarski's axiom

$$[P;Q \wedge \sim R \equiv \mathit{false}] \equiv [Q;R \wedge \sim P \equiv \mathit{false}]$$

Proof We observe for any P, Q, R

$$\begin{aligned}
& [P;Q \wedge \sim R \equiv \mathit{false}] \\
= & \quad \{\text{(predicate and) relational calculus}\} \\
& [\neg(P;Q) \vee *R] \\
= & \quad \{\text{predicate calculus}\} \\
& [P;Q \Rightarrow *R] \\
= & \quad \{\text{relational calculus}\} \\
& [P;Q;R \Rightarrow D] \\
= & \quad \{\text{rotation}\} \\
& [Q;R;P \Rightarrow D] \\
= & \quad \{\text{as above}\}
\end{aligned}$$

$$[Q;R \wedge \sim P \equiv \mathit{false}] \quad .$$

(End of Proof.)

To quote Hoare and He Jifeng on Tarski's formulation: "Replacement of this last axiom will not be lamented. (To this I can add that, by our current standards, also notationally Tarski's text is atrocious.)

In order to relate the above to the axiomatization of Hoare and He Jifeng, we shall derive as a theorem their axiom

$$[\neg(P \backslash Q) \equiv (\neg Q \backslash \neg P) \backslash \neg J]$$

where $\backslash$ is given by

$$[X \backslash Y \equiv *(X; *Y)]$$

Proof We observe for any P, Q

$$\begin{aligned}
& (\neg Q \backslash \neg P) \backslash \neg J \\
= & \quad \{\text{elimination outer } \backslash\} \\
& *((\neg Q \backslash \neg P); \sim J) \\
= & \quad \{\sim J \text{ is identity element of } ;\} \\
& *(\neg Q \backslash \neg P) \\
= & \quad \{\text{elimination } \backslash\} \\
& \neg Q ; \sim P \\
= & \quad \{(21)\} \\
& \sim(P; *Q) \\
= & \quad \{\text{introduction } \backslash\} \\
& \neg(P \backslash Q)
\end{aligned}$$

(End of Proof.)

To which I am tempted to add "Replacement of this last axiom will not be lamented".

* * *

Our axiomatization of the relational calculus introduced the triple $(*\ ;\ \mathbb{D})$, but all axioms given thus far are satisfied if we equate the triple $(*\ ;\ \mathbb{D})$ with the triple $(\neg \wedge \text{false})$. To distinguish the two we now introduce a last axiom for composition that is not satisfied by conjunction

(30) $[P;\text{true} \vee \text{true};Q] \Rightarrow [P;\text{true}] \vee [\text{true};Q]$

Remark Note, firstly, that the disjuncts are any precondition and any postcondition respectively. Note, secondly, that LHS $\Leftarrow$ RHS, so we also could have written

(30') $[P;\text{true} \vee \text{true};Q] \equiv [P;\text{true}] \vee [\text{true};Q]$

(End of Remark.)

To relate the above to Tarski we shall prove as a theorem Tarski's axiom

(31) $[P;\text{true}] \vee [\text{true};\neg P]$

Proof We observe for any P

$[P;\text{true}] \vee [\text{true};\neg P]$

$\Leftarrow$ {(30) with $Q := \neg P$}

$$
\begin{array}{ll}
 & [\,P;\mathit{true}\ \vee\ \mathit{true};\neg P\,] \\
\Leftarrow & \quad \{;\ \text{is monotonic}\} \\
 & [\,P;\,\mathbb{J}\ \vee\ \mathbb{J};\neg P\,] \\
= & \quad \{\mathbb{J}\ \text{is}\ ;\text{'s unit}\} \\
 & [\,P\ \vee\ \neg P\,] \\
= & \quad \{\text{predicate calculus}\} \\
 & \mathit{true}
\end{array}
$$

(End of Proof.)

Remark I did not succeed in proving (30) from (31). (End of Remark)

Next we shall prove what C.S. Scholten postulated to distinguish the triples:

(32) $[\mathit{true};\neg X;\mathit{true}]\ \vee\ [X]$.

Proof We observe for any X

$$
\begin{array}{ll}
 & [\mathit{true};\neg X;\mathit{true}]\ \vee\ [X] \\
\Leftarrow & \quad \{(31)\ \text{with}\ P := \mathit{true};\neg X\ ,\ \text{monotonicity}\ \vee\} \\
 & [\mathit{true};\neg(\mathit{true};\neg X)]\ \Rightarrow\ [X] \\
\Leftarrow & \quad \{\text{monotonicity}\ [\,]\} \\
 & [\,\mathit{true};\neg(\mathit{true};\neg X)\ \Rightarrow\ X\,] \\
= & \quad \{\text{rel. calc.}\} \\
 & [\,{*X};\mathit{true}\ ;\neg(\mathit{true};\neg X)\ \Rightarrow\ \mathbb{D}\,] \\
= & \quad \{\text{rel. calc}\} \\
 & [\,{*X};\mathit{true}\ \Rightarrow\ \sim(\mathit{true}\ ;\neg X)\,] \\
= & \quad \{\text{rel. calc}\} \\
 & [\,{*X};\mathit{true}\ \Rightarrow\ {*X};\mathit{true}\,] \\
= & \quad \{\text{pred. calc.}\} \\
 & \mathit{true}
\end{array}
$$

Q.E.D.

Finally we prove C.S. Scholten's

(33) $[\neg(X; true; Y)] \Rightarrow [\neg X] \vee [\neg Y]$

Proof We observe for any X

$[\neg(X; true; Y)]$
$=$ {pred. calc}
$[\neg(X; (true \wedge true; Y))]$
$=$ { true; Y is a postcondition; (29)}
$[\neg(X; true \wedge true; Y)]$
$=$ { de Morgan}
$[\neg(X; true) \vee \neg(true; Y)]$
$\Rightarrow$ {(28) and its dual; (30)}
$[\neg(X; true)] \vee [\neg(true; Y)]$
$\Rightarrow$ { $[J \Rightarrow true]$ and monotonicity }
$[\neg(X; J)] \vee [\neg(J; Y)]$
$=$ { J }
$[\neg X] \vee [\neg Y]$

Rather trivially, (32) and (33) can be strengthened to

(32') $[true; \neg X; true] \neq [X]$ and
(33') $[\neg(X; true; Y)] \equiv [\neg X] \vee [\neg Y]$

A possibly useful formula, which follows rather directly from (33') is

(34) $[\neg(X; true)] \equiv [\neg X]$

As axioms, (30) and (33) are interchangeable.

(0) $[P \Rightarrow *Q] \equiv [Q \Rightarrow *P]$

(1) $[P \Leftarrow *Q] \equiv [Q \Leftarrow *P]$

(2) $[*\mathit{false} \equiv \mathit{true}]$

(3) $[*(\underline{E}Q :: Q) \equiv (\underline{A}Q :: *Q)]$

(4) $[*\mathit{true} \equiv \mathit{false}]$

(5) $[*(\underline{A}Q :: Q) \equiv (\underline{E}Q :: *Q)]$

(6) $[P \equiv *Q] \equiv [Q \equiv *P]$

(7) $[P \equiv **P]$

(8) $[P;(Q;R) \equiv (P;Q);R]$

(9) $[P;Q \Rightarrow D] \equiv [P \Rightarrow *Q]$

(10) $[*(*P;*Q) \equiv \neg(\neg Q;\neg P)]$

(11) $[P;Q \Rightarrow D] \equiv [Q;P \Rightarrow D]$

(12) $(\underline{A}Z :: [P;Z \Rightarrow D] \equiv [Q;Z \Rightarrow D]) \equiv [P \equiv Q]$

(13) $[*D;P \equiv P]$

(14) $[P;*D \equiv P]$

(15) $[X;Y \Rightarrow Z] \equiv [X \Rightarrow *(Y;*Z)]$

(16) $[X \Leftarrow Y;Z] \equiv [*(*X;Y) \Leftarrow Z]$

(17) ; is universally disjunctive in both operands

(18) $[*D \equiv \neg D]$

(19) $[\neg *X \equiv *\neg X]$

(20) $\neg$ (negation), $*$ (estrangement) and $\sim$ (transposition)

- are all three involutions
- distribute over each other
- satisfy $[\neg * \sim P \equiv P]$ for all P

(21) $[\sim(P;Q) \equiv \sim Q;\sim P]$

(22) $[\sim \mathbb{D} \equiv \mathbb{D}]$

(23) $[\mathbb{J} \equiv \neg \mathbb{D}]$ $[\mathbb{J} \equiv *\mathbb{D}]$ $[\mathbb{J} \equiv \sim \mathbb{J}]$

(24) $[\mathbb{J};P \equiv P]$ $[P;\mathbb{J} \equiv P]$

(25) transposition distributes over the logical operators and the quantifications

(26) (P is a precondition) $\equiv [P;\text{true} \equiv P]$
(P is a postcondition) $\equiv [\text{true};P \equiv P]$

(26') (P is a precondition) $\equiv [P;\text{true} \Rightarrow P]$
(P is a postcondition) $\equiv [\text{true};P \Rightarrow P]$

(27) (P is a postcondition) $\equiv$
($\sim P$ is a precondition)

(28) logical expressions built from postconditions are postconditions.

(29) $[\text{true};P \Rightarrow P] \Rightarrow [X;(Y \wedge P) \equiv X;Y \wedge P]$

(30) $[P;\text{true} \vee \text{true};Q] \Rightarrow [P;\text{true}] \vee [\text{true};Q]$

(31) $[P;\text{true}] \vee [\text{true};\neg P]$

(32) $[\text{true}; \neg X; \text{true}] \vee [X]$

(32') $[\text{true}; \neg X; \text{true}] \not\equiv [X]$

(33) $[\neg(X; \text{true}; Y)] \Rightarrow [\neg X] \vee [\neg Y]$

(33') $[\neg(X; \text{true}; Y)] \equiv [\neg X] \vee [\neg Y]$

(34) $[\neg(X; \text{true})] \equiv [\neg X]$

Acknowledgements

My indebtedness to A. Tarski, to C.A.R. Hoare and He Jifeng, to W.H. Hesselink and to C. S. Scholten is obvious. I did not much more than sort the material and present it in a homogeneous notation.

Austin, 8 November 1990

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78750-8138

An introduction into the relational calculus

The relational calculus is a calculus of mathematical relations. Pioneering work on the subject was done by Alfred Tarski as early as the forties of this century. In more recent times computing scientists have shown a renewed interest in the calculus, because of its potential adequacy for dealing with program semantics and even program development. The seminal paper by C.A.R. Hoare and He Jifeng [HH86], written in 1985, clearly marks the beginning of a period.

By a number of local circumstances we ourselves became involved with the relational calculus as well; being programmers, we came to ask ourselves how the calculus can be presented in a fashion so smooth and simple that it can be reconstructed with little effort at any moment, be it tomorrow, next week, or even a year from now. This text is an effort towards that goal. It presents the calculus in a number of small steps, each step comprising the introduction of one postulate and exploring the consequences of its incorporation.

The text was written with the uninitiated reader in mind. However, since we will present the relational calculus as a specialization of the predicate calculus, familiarity

with the latter is needed. Our presentation is based on the notational conventions and the calculational mode laid down by Edsger W. Dijkstra and Carel S. Scholten in [DS 90].

There are two papers from which we have benefitted greatly. The one is a technical report by Edsger W. Dijkstra [EWD 1047] and the other a technical note by C. S. Scholten [CSS 164]. In fact, we have tried to combine the best of both while adding our own cadenzas.

At earlier stages of our struggles with the relational calculus Jaap van der Woude has been very helpful, giving support and technical assistance. As always, the members of the Eindhoven Tuesday Afternoon Club have been a continuous and indispensable forum. There is one member in particular who initiated our involvement with the relational calculus and without whom we would never have written this paper. To him, to Carel Steven Scholten, we respectfully and gratefully dedicate this text, on the occasion of his becoming an honorary doctor.

0 A preamble on junctivity

When dealing with predicate transformers, i.e. functions from predicates to predicates, it may be quite useful to know their so-called junctivity properties. In this presentation of the relational calculus we will encounter some predicate transformers that have outstanding junctivity properties. Hence this preamble.

Usually one distinguishes two kinds of junctivity, viz. disjunctivity and conjunctivity.

The strongest form of disjunctivity is so-called universal disjunctivity. Predicate transformer f is called universally disjunctive if it distributes over arbitrary disjunctions, i.e. if for all possible ranges of dummy x,

$$[\, f.(\mathbin{\Big[\!\Big]}x :: x) \equiv (\mathbin{\Big[\!\Big]}x :: f.x)\,] \quad .$$

The best-known consequences of f being universally disjunctive are

- $[\, f.\mathit{false} \equiv \mathit{false}\,]$
- f is monotonic with respect to $\Rightarrow$, i.e. $[\, x \Rightarrow y\,] \Rightarrow [\, f.x \Rightarrow f.y\,]$.

The strongest form of conjunctivity is so-called universal conjunctivity. Predicate transformer g is called universally

conjunctive if it distributes over arbitrary conjunctions. i.e. if for all possible ranges of dummy y,

$$[\, g.(\underline{A}y :: y) \equiv (\underline{A}y :: g.y) \,] .$$

The best-known consequences of g being universally conjunctive are

- $[\, g.\mathrm{true} \equiv \mathrm{true} \,]$
- g is monotonic with respect to $\Rightarrow$.

In what follows we shall freely use these facts.

* * *

Sometimes the universal junctivity of a predicate transformer comes for free, thanks to the following lemma.

The junctivity lemma

Let predicate transformers f and g be such that (for all x, y)

$$(*) \qquad [\, f.x \Rightarrow y \,] \equiv [\, x \Rightarrow g.y \,] .$$

then f is universally disjunctive
and g is universally conjunctive.

Proof In order to prove f's universal disjunctivity we must prove, for arbitrary range of x,

$$(**) \qquad [\, f.(\underline{E}x :: x) \equiv (\underline{E}x :: f.x) \,] .$$

To that end we observe for any y .

$$
\begin{array}{ll}
 & [\, f.(\mathbin{\underline{\exists}} x :: x) \Rightarrow y \,] \\
= & \quad \{ (*) \text{ with } x := (\mathbin{\underline{\exists}} x :: x) \} \\
 & [\, (\mathbin{\underline{\exists}} x :: x) \Rightarrow g.y \,] \\
= & \quad \{ \text{predicate calculus, i.e. distribution of } \Rightarrow g.y \text{ over } \mathbin{\underline{\exists}} \} \\
 & [\, (\mathbin{\underline{\forall}} x :: x \Rightarrow g.y) \,] \\
= & \quad \{ \text{interchange of } [\,] \text{ and } \mathbin{\underline{\forall}} \} \\
 & (\mathbin{\underline{\forall}} x :: [\, x \Rightarrow g.y \,]) \\
= & \quad \{ (*) \} \\
 & (\mathbin{\underline{\forall}} x :: [\, f.x \Rightarrow y \,]) \\
= & \quad \{ \text{interchange of } \mathbin{\underline{\forall}} \text{ and } [\,] \} \\
 & [\, (\mathbin{\underline{\forall}} x :: f.x \Rightarrow y) \,] \\
= & \quad \{ \text{predicate calculus} \} \\
 & [\, (\mathbin{\underline{\exists}} x :: f.x) \Rightarrow y \,] \quad ,
\end{array}
$$

and since the equivalence between the first and the last line holds for any y . (**) follows . Note that in the above proof the internal structure of the expression $g.y$ is completely irrelevant . That expression could equally well be something outlandish like $\neg(\sim z \,;\, \neg y)$. provided it is a predicate .

The - very similar - proof of g's universal conjunctivity is left to the reader .

(End of Proof .)

1 The transposition

In this presentation of the relational calculus relations are considered to be predicates, and the relational calculus will emerge from the predicate calculus by extending the latter with two new operators and a new constant.

The first operator to be added is the so-called transposition. It is a unary operator to be denoted by a prefix $\sim$ – pronounced "tilde" – and it is given the same binding power as $\neg$. By postulate, the transposition satisfies

(0) $\quad [\sim x \Rightarrow y] \equiv [x \Rightarrow \sim y] \qquad$ (for all x, y).

From (0) and the junctivity lemma with $f.g := \sim . \sim$ we conclude

$\sim$ is universally disjunctive, and
$\sim$ is universally conjunctive,

and hence

(1) $\quad [\sim \text{false} \equiv \text{false}]$, $[\sim \text{true} \equiv \text{true}]$,
$\sim$ is monotonic with respect to $\Rightarrow$.

After thus having settled $\sim$'s junctivity properties, we now investigate how it "behaves" in combination with negation.

the square brackets, and itself.

As for the latter we will show that $\sim$ is an involution, i.e.

(2) $\quad [\sim\sim x \equiv x] \qquad$ (for all x)

Proof We observe for any x

$$
\begin{aligned}
& [\sim\sim x \Rightarrow x] \\
= & \quad \{(0), \text{ see Remark below}\} \\
& [\sim x \Rightarrow \sim x] \\
= & \quad \{(0)\} \\
& [x \Rightarrow \sim\sim x]
\end{aligned}
$$

Since the middle line equivales true, so do the outer lines, which establishes (2).

(End of Proof.)

Remark In the hints of the above little calculation we deliberately left out the particular instantiations of (0) because including them would not agree with our view of (0): formula (0) expresses that we can freely move the symbol $\sim$ back and forth between the antecedent and the consequent of an implication between square brackets. It is by means of this simple "symbol dynamics" that we remember and use (0). The situation is very similar to the way most people deal with e.g. the rules of de Morgan.
(End of Remark.)

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

(3) $[\neg\sim x \equiv \sim\neg x]$ (for all x).

Proof The proof is by mutual implication:

$[\neg\sim x \Rightarrow \sim\neg x]$
$=$ { shunting the antecedent to the consequent }
$[\text{true} \Rightarrow \sim x \vee \sim\neg x]$,

and the validity of the latter follows from

$\sim x \vee \sim\neg x$
$=$ { $\sim$ over $\vee$: $\sim$ is universally disjunctive }
$\sim(x \vee \neg x)$
$=$ { predicate calculus }
$\sim$ true
$=$ { (1) }
true .

The _very similar_ proof of the implication in the other direction is left to the reader.

(End of Proof.)

Because the transposition is universally junctive and distributes over the negation, we have as a corollary:

(4) Transposition distributes over all logical operators including the two logical quantifiers.

Finally, from (0) with $x, y := true, x$ together with $[\sim true \equiv true]$, we conclude

(5) $\quad [x] \equiv [\sim x]$;

i.e. we can freely transpose or "de-transpose" an expression between square brackets.
This rule is useful for calculational practice, especially in combination with (4).

This completes our introduction of the transposition.

2 The composition and the transposition

The second - and last! - operator to be added to the predicate calculus is the so-called composition. It is a binary operator to be denoted by an infix ";" - pronounced "semi" - and it is given a higher binding power than the other binary operators and a lower binding power than the unary operators.

Our first postulate about the composition is very important but not too interesting in its own right. It reads

(0) the composition is associative, i.e.

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

The second postulate is, in fact, a pair of postulates linking composition and transposition:

(1a) the "left-exchange"

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

(1b) the "right-exchange"

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

At first, these rules look horrifying but they no longer are once we become aware

of their "symbol dynamics". First observe that all antecedents are compositions of two operands. In the left-exchange the one side of the equivalence is transformed into the other by exchanging the left operand and the consequent while negating both – i.e. taking their contrapositive – and transposing the operand that remains in place. Note that this recipe works in both directions since removing a negation or transposition is the same as adding one – both are involutions –. The right-exchange has the same dynamics, except that this time the right operand is exchanged.

The above exchange rules can be combined – about which more later – into Jaap van der Woude's rule:

(2) the "middle-exchange"

$$[\, x ; u ; y \Rightarrow v \,] \equiv [\, \neg x ; \neg v ; \neg y \Rightarrow \neg u \,] .$$

Proof

$$\begin{aligned} & [\, x ; u ; y \Rightarrow v \,] \\ = & \quad \{ (0) , \text{ i.e. } ; \text{ is associative} \} \\ & [\, (x ; u) ; y \Rightarrow v \,] \\ = & \quad \{ \text{left-exchange} \} \\ & [\, \neg v ; \neg y \Rightarrow \neg (x ; u) \,] \\ = & \quad \{ \text{contrapositive} \} \end{aligned}$$

$$
\begin{array}{ll}
 & [\, x;u \;\Rightarrow\; \neg(\neg v;\sim y)\,] \\
= & \quad \{\text{right-exchange}\} \\
 & [\, \sim x;(\neg v;\sim y) \;\Rightarrow\; \neg u\,] \\
= & \quad \{(0)\} \\
 & [\, \sim x;\neg v;\sim y \;\Rightarrow\; \neg u\,] \,.
\end{array}
$$

(End of Proof.)

Remark The exchange rules may bring about all sorts of functional compositions of $\neg$ and $\sim$. In our calculations we will not hesitate to simplify these compositions in one go along with other manipulations, even without saying so explicitly. Thus, the negation of $\sim\neg x$ simply is $\sim x$.
(End of Remark.)

* * *

Next we investigate whether $\sim$ has any distributivity properties with respect to composition. To that end we observe for any x, y, z

$$
\begin{array}{ll}
 & [\, \sim(x;y) \;\Rightarrow\; z\,] \\
= & \quad \{\text{move } \sim \text{ to consequent (see (1.0))}\} \\
 & [\, x;y \;\Rightarrow\; \sim z\,] \\
= & \quad \{\text{left-exchange}\} \\
 & [\, \neg\sim z;\sim y \;\Rightarrow\; \neg x\,] \\
= & \quad \{\text{right-exchange}\} \\
 & [\, \neg z;x \;\Rightarrow\; \neg\sim y\,] \\
= & \quad \{\text{left-exchange}\} \\
 & [\, \sim y;\sim x \;\Rightarrow\; z\,]
\end{array}
$$

and since the equivalence between the first

and the last line holds for any z, we conclude

(3) the "$\sim$ over $;$" - rule

$$[\ \sim(x\,;\,y) \ \equiv\ \sim y\,;\,\sim x\ ]\ .$$

* * *

Next we show that composition is universally disjunctive in both operands We do so by exploiting the junctivity lemma dealt with in the preamble. For arbitrary z, we define predicate transformer f as follows :

$$[\ f.x \ \equiv\ z\,;\,x\ ] \qquad \text{(for all } x\text{)}\ .$$

By the junctivity lemma, we can prove f's universally disjunctivity - and, hence, the composition's universal disjunctivity in its second operand - provided we manage to establish

$$[\ f.x \Rightarrow y\ ] \ \equiv\ [\ x \Rightarrow \text{something}\ ]\ .$$

This turns out to be very simple :

$$\begin{array}{ll} & [\ f.x \Rightarrow y\ ] \\ = & \quad \{\ \text{definition of } f\ \} \\ & [\ z\,;\,x \Rightarrow y\ ] \\ = & \quad \{\ \text{right-exchange in order to isolate } x\ \} \\ & [\ \sim z\,;\,\neg y \Rightarrow \neg x\ ] \\ = & \quad \{\ \text{contrapositive}\ \} \\ & [\ x \Rightarrow \neg(\sim z\,;\,\neg y)\ ] \\ = & \quad \{\ \} \\ & [\ x \Rightarrow \text{something}\ ]\ . \end{array}$$

The -very similar- proof of the composition's universal disjunctivity in the first operand is left to the reader. Thus we have established

(4) the composition is universally disjunctive in both operands.

As a result, we also have

(5) [false ; x ≡ false] . [x ; false ≡ false],

and

composition is monotonic with respect to ⇒ in both operands.

* * *

Finally, we wish to mention that composition and negation do not relate nicely and that -hence- there are no nice conjunctivity properties of the composition.

3 The identity of composition

We would like composition to have an identity element. Therefore, we postulate the existence of a constant predicate J satisfying

(0) $[x ; J \equiv x]$ (for all x),

which conveys that J is a right-identity element.

Remark We cannot hope to prove the existence of such a J from our previous postulates, because - as Rob Hoogerwoord observed - all the latter are satisfied if the composition is implemented by $[x ; y \equiv \text{false}]$ for all x, y : in this case (0) reduces to the highly improbable $[\text{false} \equiv x]$ for all x.
(End of Remark.)

From (0) we derive that composition has $\sim J$ as a left-identity element:

(1) $[\sim J ; x \equiv x]$ (for all x)

Proof We observe for any x,

$$
\begin{aligned}
& \sim J ; x \\
= & \quad \{ [\sim\sim x \equiv x], \text{ see (1.2)} \} \\
& \sim J ; \sim\sim x \\
= & \quad \{ \sim \text{ over } ; \text{ , see (2.3)} \}
\end{aligned}
$$

$$
\begin{array}{ll}
 & \sim (\sim x \,;\, \bar{J}) \\
= & \quad \{ (0) \text{ with } x := \sim x \} \\
 & \sim (\sim x) \\
= & \quad \{ \sim\sim \} \\
 & x
\end{array}
$$

(End of Proof.)

If existent, left- and right-identities, are equal, so we have

(2) $\quad [\sim J \equiv J]$,

which follows from

$$
\begin{array}{ll}
 & \sim J \\
= & \quad \{ (0) \text{ with } x := \sim \bar{J} \} \\
 & \sim \bar{J} \,;\, J \\
= & \quad \{ (1) \text{ with } x := J \} \\
 & J \ .
\end{array}
$$

In summary we conclude from (0), (1), and (2)

(3) $\quad [x \,;\, J \equiv x]$ and $[J \,;\, x \equiv x]$,

i.e. J is the (two-sided) identity element of composition.

The one and only theorem of this section that will be appealed to later on is

(4) $\quad [x \Rightarrow x \,;\, \text{true}]$, and its dual

$[x \Rightarrow \text{true} \,;\, x]$.

Proof We prove $[x \Rightarrow x ; true]$

$$
\begin{array}{ll}
 & x ; true \\
\Leftarrow & \quad \{ ; \text{ is monotonic, using } [true \Leftarrow J] \} \\
 & x ; J \\
= & \quad \{ \text{identity of } ; \} \\
 & x \ .
\end{array}
$$

(End of Proof.)

With the introduction of J an overwhelming number of new theorems waits to be formulated or invented. In this introductory text we will, however, hardly touch on any of these. Presumably, the enormous growth in "manipulative potential" is best explained by the shape of formulae (3) and (4): they are the only rules, so far, for introducing (and removing) compositions.

* * *

The relational calculus can, of course, be developed in various ways. More traditional presentations, for instance, postulate at one fell swoop

- ; is associative
- J is the two-sided identity element of ;
- $[J \equiv \sim J]$
- the middle-exchange rule,

and this indeed suffices to prove all the results that we have established so far. The problem with this set of postulates,

however, is that hardly anything useful can be derived from a proper subset of them. It is only when the whole bunch is taken into account that the calculus opens up like Pandora's Box. It is precisely this lack of disentangledness that has prompted us to diverge from such a more traditional presentation.

4 A little theory of left- and right-conditions

In this section we offer no new postulates but instead deal with two special kinds of predicates, to be called "left-conditions" and "right-conditions". We do so because in every treatment of the relational calculus these "conditions" pop up every so often and because an elementary mastery of their algebraic properties may have a beneficial effect on our relational calculations as a whole.

By definition,

(0a) p is a left-condition $\equiv$ $[p; \mathit{true} \equiv p]$

(0b) q is a right-condition $\equiv$ $[\mathit{true}; q \equiv q]$.

Because for all x, we have — see (3.4) —

$$[x; \mathit{true} \Leftarrow x] \quad \text{and} \quad [\mathit{true}; x \Leftarrow x],$$

we may conclude

(1a) p is a left-condition $\equiv$ $[p; \mathit{true} \Rightarrow p]$

(1b) q is a right-condition $\equiv$ $[\mathit{true}; q \Rightarrow q]$.

Note that formulae (1) are formally weaker characterizations of conditions than formulae (0).

Remark It is in general quite useful to be aware of the various equivalent forms of a formula, in particular if some of these

forms are formally weaker than others. For proving the validity of a formula, the weak version is usually to be preferred, whereas in exploiting the validity the strong version is preferable.
(End of Remark.)

Because of (2) below, all theorems on left- and right-conditions come in pairs; in what follows we therefore focus on left-conditions mainly.

(1) p is a left-condition
$\equiv$ $\sim p$ is a right-condition

Proof

p is a left-condition
$=$ { (0a) }
$[\, p ; \mathit{true} \equiv p \,]$
$=$ { $[x] \equiv [\sim x]$, see (1.5), and
$\sim$ over $\equiv$, see (1.4) }
$[\, \sim (p ; \mathit{true}) \equiv \sim p \,]$
$=$ { $\sim$ over $;$, see (2.3), and
$[\, \sim \mathit{true} \equiv \mathit{true} \,]$ }
$[\, \mathit{true} ; \sim p \equiv \sim p \,]$
$=$ { (0b) }
$\sim p$ is a right-condition

(End of Proof.)

* * *

First we investigate how to form new left-conditions from existing ones. In fact,

we can show

(3) $x : p$ is a left-condition
$\Leftarrow$ p is a left-condition . for all x,

and

(4) all logical expressions built from left-conditions only are left-conditions .

The proof of (3) is left to the reader.
As for (4), existential quantification, negation, and the constant true suffice to build all logical expressions, so that (4) follows if we can demonstrate

(4a) true is a left-condition

(4b) $\neg p$ is a left-condition
$\equiv$ p is a left-condition

(4c) $(\mathrm{E}p :: p)$ is a left-condition
$\Leftarrow$ $(\mathrm{A}p ::$ p is a left-condition $)$.

The proof of (4a) is left to the reader.

Proof of (4b)

By (1a), the result follows from

$$
\begin{array}{ll}
 & [\neg p \,;\, \text{true} \Rightarrow \neg p] \\
= & \quad \{ \text{left-exchange, and } [\neg \text{true} \equiv \text{true}] \} \\
 & [p \,;\, \text{true} \Rightarrow p] \; .
\end{array}
$$

(End of Proof.)

Proof of (4c)

By (0a), the result follows from

$$
\begin{array}{ll}
 & (\mathbin{\Box} p :: p) \,;\, \mathit{true} \\
= & \quad \{ \;;\; \text{is universally disjunctive} \} \\
 & (\mathbin{\Box} p :: p \,;\, \mathit{true}) \\
= & \quad \{ \text{from the antecedent of (4c):} \\
 & \qquad [\, p \,;\, \mathit{true} \equiv p \,] \; \} \\
 & (\mathbin{\Box} p :: p) \;.
\end{array}
$$

(End of Proof.)

* * *

Among all theorems involving left- or right-conditions, the following one ranks very high in importance.

(5) for left-condition p,

$$[\, (p \wedge x) \,;\, y \;\equiv\; p \wedge x \,;\, y \,]$$

(for all x, y).

Its dual is

(6) for right-condition q,

$$[\, x \,;\, (y \wedge q) \;\equiv\; x \,;\, y \wedge q \,]$$

(for all x, y).

(It is the shape of these rules that has triggered the names "left-condition" and right-condition.)

Proof of (5)

The proof is by mutual implication. In

the one direction we use that p is a left-condition and in the other direction we use that $\neg p$ is a left-condition.

- $[\ (p \wedge x) ; y \ \Rightarrow \ p \wedge x ; y\ ]$

  This part is established by two independent weakenings of the antecedent, viz.

  $$\begin{array}{ll} & (p \wedge x) ; y \\ \Rightarrow & \quad \{ \ ; \text{ is monotonic} \} \\ & x ; y \end{array}$$

  and

  $$\begin{array}{ll} & (p \wedge x) ; y \\ \Rightarrow & \quad \{ \ ; \text{ is monotonic} \} \\ & p ; \text{true} \\ = & \quad \{ \ p \text{ is a left-condition} \} \\ & p \ . \end{array}$$

- $[\ (p \wedge x) ; y \ \Leftarrow \ p \wedge x ; y\ ]$

  Here we follow a heuristical advice that we owe to Lex Bijlsma - see Remark below - and rewrite the demonstrandum into the equivalent

  $[\ \neg p \vee (p \wedge x) ; y \ \Leftarrow \ x ; y\ ]$.

  This, now, follows from

  $$\begin{array}{ll} & \neg p \vee (p \wedge x) ; y \\ = & \quad \{ \ \neg p \text{ is a left-condition, i.e.} \\ & \qquad [\ \neg p \equiv \neg p ; \text{true}\ ] \ \} \\ & \neg p ; \text{true} \vee (p \wedge x) ; y \\ \Leftarrow & \quad \{ \ ; \text{ is monotonic} \} \\ & \neg p ; y \vee (p \wedge x) ; y \end{array}$$

$$
\begin{array}{ll}
= & \{\ ;\ \text{over}\ \vee\ \} \\
 & (\neg p \vee (p \wedge x)) ; y \\
= & \{\ \text{predicate calculus}\ \} \\
 & (\neg p \vee x) ; y \\
\Leftarrow & \{\ ;\ \text{is monotonic}\ \} \\
 & x ; y
\end{array}
$$

(End of Proof of (5).)

Remark As the reader will have noticed, many of our proofs take the form of linear calculations. Whenever the demonstrandum is an implication or an equivalence, i.e. an expression with two sides, the question arises at which side to start calculating. A very useful heuristic is to start at the more complicated side. But what if both sides are equally complicated? Bijlsma's advice is to try to massage the entire demonstrandum so as to make one side more complicated than the other. How this can be done is a separate concern. We think that in the above proof we have shown a successful application of Bijlsma's Principle.
(End of Remark.)

Now we are ready to prove what may be called the main theorem on "conditions", which reads

(7) the "left/right - composition":

for left-condition p and
right-condition q,

$$[\ p ; q \ \equiv \ p \wedge q\ ] \ .$$

Together with $[x ; J \equiv x]$ and $[J ; x \equiv x]$, it is one of the few simple rules from the relational calculus that admit the introduction or elimination of the composition operator.

Proof of (7)

$p ; q$

$=$ { predicate calculus }

$(p \wedge \text{true}) ; q$

$=$ { p is a left-condition, hence (5) applies with $x, y := \text{true}, q$ }

$p \wedge \text{true} ; q$

$=$ { q is a right-condition }

$p \wedge q$.

(End of Proof.)

And this concludes our treatment of "conditions".

5 The Cone Rules

All our postulates so far, viz.

- $[\sim x \Rightarrow y] \equiv [x \Rightarrow \sim y]$
- $;$ is associative
- $[x ; y \Rightarrow z] \equiv [\neg z ; \sim y \Rightarrow \neg x]$
  $[x ; y \Rightarrow z] \equiv [\sim x ; \neg z \Rightarrow \neg y]$
- $[x ; J \equiv x]$,

are satisfied if the triple $(\sim \mid ; \mid J)$ is implemented by the triple (identity transformer $\mid \wedge \mid$ true). This possibility will now be excluded by our next and last postulate, that we owe to C.S. Scholten and has been dubbed "the Cone Rule" by Jaap van der Woude. It reads

(0) $[\text{true} ; \neg x ; \text{true}] \Leftarrow \neg [x]$ (for all x).

It can be formally strengthened into

(1) $[\text{true} ; \neg x ; \text{true}] \equiv \neg [x]$,

because the implication

(2) $[\text{true} ; \neg x ; \text{true}] \Rightarrow \neg [x]$

follows from the previous postulates. The proof of (2), which is by a case distinction between $[x] \equiv \text{true}$ and $[x] \equiv \text{false}$, is

left to the reader

× × ×

Edsger W. Dijkstra designed another rule with the effect of distinguishing the triples. It reads

(3a) $[\, x\,;\, true \;\vee\; true\,;\, y\,] \;\Rightarrow\; [\, x\,;\, true\,] \;\vee\; [\, true\,;\, y\,]$.

or -equivalently-

(3b) for all left-conditions p and right-conditions q,

$[\, p \vee q\,] \;\Rightarrow\; [\, p\,] \;\vee\; [\, q\,]$.

Alfred Tarski invented yet another rule with the same effect, viz.

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

In the remaining part of this section we shall show that Scholten's (0), Dijkstra's (3), and Tarski's (4) are all equivalent. We do so by showing

$(0) \Leftarrow (4)$, $(4) \Leftarrow (3a)$, and $(3b) \Leftarrow (0)$

in this (irrelevant) order.

Proof of $(0) \Leftarrow (4)$

First we rewrite (4) into an equivalent

implicative shape, viz.

(*) $[x ; \text{true}] \Leftarrow \neg [\text{true} ; \neg x]$

and prove (0) $\Leftarrow$ (*). We observe for any x

(0)

$=$ { definition of (0) }

$[\text{true} ; \neg x ; \text{true}] \Leftarrow \neg [x]$

$=$ { $;$ is associative }

$[(\text{true} ; \neg x) ; \text{true}] \Leftarrow \neg [x]$

$\Leftarrow$ { (*) with $x := (\text{true} ; \neg x)$ }

$\neg [\text{true} ; \neg (\text{true} ; \neg x)] \Leftarrow \neg [x]$

$=$ { contrapositive }

$[\text{true} ; \neg (\text{true} ; \neg x)] \Rightarrow [x]$

$\Leftarrow$ { monotonicity of $[\,]$ }

$[\text{true} ; \neg (\text{true} ; \neg x) \Rightarrow x]$

$=$ { right-exchange, and $[\sim \text{true} \equiv \text{true}]$ }

$[\text{true} ; \neg x \Rightarrow \text{true} ; \neg x]$

$=$ { predicate calculus }

true .

(End of Proof.)

Proof of (4) $\Leftarrow$ (3a)

We observe for any x

(4)

$=$ { definition of (4) }

$[x ; \text{true}] \vee [\text{true} ; \neg x]$

$\Leftarrow$ { (3a) with $y := \neg x$ }

$[x ; \text{true} \vee \text{true} ; \neg x]$

$\Leftarrow$ { $[x ; \text{true} \Leftarrow x]$ and $[\text{true} ; \neg x \Leftarrow \neg x]$, see (3.4) }

$[x \vee \neg x]$

$=$ { predicate calculus }
true .

(End of Proof.)

Proof of (3b) $\Leftarrow$ (0)

For any left-condition p and right-condition q we observe

$[p] \vee [q]$

$\Leftarrow$ { contrapositive of (0), i.e. $[x] \Leftarrow \neg [true ; \neg x ; true]$ }

$\neg [true ; \neg p ; true] \vee \neg [true ; \neg q ; true]$

$=$ { $\neg p$ is a left-condition, i.e. $[\neg p ; true \equiv \neg p]$, and $\neg q$ is a right-condition, i.e. $[true ; \neg q \equiv \neg q]$ }

$\neg [true ; \neg p] \vee \neg [\neg q ; true]$

$=$ { de Morgan }

$\neg ([true ; \neg p] \wedge [\neg q ; true])$

$=$ { $[]$ over $\wedge$ }

$\neg [true ; \neg p \wedge \neg q ; true]$

$=$ { $true ; \neg p$ is a left-condition because $\neg p$ is - see (4.3) - and $\neg q ; true$ is a right-condition because $\neg q$ is . Hence the left/right - composition applies - see (4.7) - }

$\neg [true ; \neg p ; \neg q ; true]$

$=$ { ; is associative }

$\neg [\ true ; (\neg p ; \neg q) ; true\ ]$

$=$ { left/right - composition on $\neg p ; \neg q$ }

$\neg [\ true ; (\neg p \wedge \neg q) ; true\ ]$

$=$ { de Morgan }

$\neg [\ true ; \neg(p \vee q) ; true\ ]$

$\Leftarrow$ { contrapositive of (2), i.e.
$\neg [\ true ; \neg x ; true\ ] \Leftarrow [x]$,
with $x := p \vee q$ }

$[\ p \vee q\ ]$.

Remark The two strengthening steps in the above calculation could have been replaced with equality-preserving steps had we used the formally stronger (1) rather than (0) and (2) separately. This, however, would have obfuscated that the first strengthening is a genuine reference to the newly introduced postulate (0) or (1), whereas the second strengthening does not rely on that postulate at all. This confirms Lincoln A. Wallen's warning that the notion of "equivalence" should be handled with care: equivalences hide why the one side implies the other, and these reasons can radically differ for the two directions.
(End of Remark.)

(End of Proof.)

And this concludes our introduction of the relational calculus.

5 Etudes

For the reader who has travelled this far we include a small set of exercises. We have not tried to sort them according to section or difficulty, simply because we do not really know how to do that.

From our own experience we know that many of these exercises can be discouragingly hard, even though all of them admit relatively short and technically simple solutions. Our advice to the reader who gets stuck is to stop trying, and reinvestigate the problem much more consciously than before, keeping one eye on the formula to be manipulated and the other on the manipulative possibilities offered by the calculus. During that process, application of some of the heuristical rules that we scattered through the text may help. The reader will then experience that the exercises, when carried out along these lines, are much more than just exercises in relational calculation: they are exercises in proof construction.

0. $$[x \Rightarrow J] \wedge [y \Rightarrow J] \Rightarrow [x ; y \equiv x \wedge y]$$

1. $$[x \Rightarrow J] \Rightarrow [x \equiv \sim x]$$

2. For p a left- or right-condition
$[\, p ; p \equiv p \,]$

3. $[\, \neg (x ; \mathit{true}) \,] \equiv [\, \neg x \,]$

4. Alfred Tarski's rotation rule:
$[\, x ; y \Rightarrow \neg \sim z \,] \equiv [\, y ; z \Rightarrow \neg \sim x \,]$.

5. For p a left-condition
$[\, x ; (p \wedge y) \equiv (x \wedge \sim p) ; y \,]$

6. $[\, x \Rightarrow x ; \sim x ; x \,]$

7. For q a right-condition
$[\, q ; \mathit{true} ; \sim q \equiv q ; \sim q \,]$

8. $[\, J \equiv (\underline{A} x :: \neg (\sim x ; \neg x)) \,]$

9. $[\, \neg (x ; \mathit{true} ; y) \,] \equiv [\, \neg x \,] \vee [\, \neg y \,]$
(this is yet another Cone Rule)

10. $[\, x \Rightarrow J \,] \Rightarrow [\, x \equiv (x ; \mathit{true} ; x) \wedge J \,]$

11. $[\, x ; y \wedge z \Rightarrow x ; (y \wedge \sim x ; z) \,]$

12. $[x \Rightarrow J] \Rightarrow [x ; (y \wedge z) \equiv x ; y \wedge z]$

13. For p and q left-conditions
$[p ; q \equiv p] \vee [q]$

14. $[J] \vee [\neg J ; \mathrm{true}]$

6 References

[DS 90] Predicate Calculus and Program Semantics, Edsger W. Dijkstra and Carel S. Scholten, Springer Verlag 1990

[EWD1047] A relational summary, Edsger W. Dijkstra, November 1990, Austin TX, USA

[HH86] The weakest prespecification, C.A.R. Hoare and He Jifeng, Fundamenta Informaticae 9: 51-84, 217-252, 1986

[CSS164] Introduction of transposition and composition, C.S. Scholten, January 1991, Beekbergen, The Netherlands.

Eindhoven,
8 March 1991

W.H.J. Feijen and A.J.M. van Gasteren,
Department of Mathematics and
Computing Science,
Eindhoven University of Technology,
P.O. Box 513,
5600 MB Eindhoven,
The Netherlands

A comparison of relational proofs

The purpose of this note is to compare different proofs for the same theorem of the relational calculus, so that we may get some feeling for the sources of manipulative advantages and disadvantages. The theorem chosen is simple enough that it can be proved from first principles; yet I hope it is "deep" enough to make the experiment interesting.

Theorem Let t be the strongest solution of

(*) $\quad x: [\,] \vee a;x \Rightarrow x\,]$

i.e. -in view of Knaster-Tarski-

(0) $\quad [\,] \vee a;t \equiv t\,]$

(1) $\quad [\,] \Rightarrow x\,] \wedge [\,a;x \Rightarrow x\,] \Rightarrow [\,t \Rightarrow x\,]$ for all x;

let s be the strongest solution of

(**) $\quad x: [\,b \vee a;x \Rightarrow x\,]$, i.e.

(2) $\quad [\,b \vee a;s \equiv s\,]$

(3) $\quad [\,b \Rightarrow x\,] \wedge [\,a;x \Rightarrow x\,] \Rightarrow [\,s \Rightarrow x\,]$; then

(4) $\quad [\,s \equiv t;b\,]$

Proof 0 In this proof we try to find closed expressions for t and s, whereafter we establish (4) by manipulating these expressions. Comparing (*) with (**), we see that (*) is obtained from (**) with for b the special choice

$\mathcal{J}$; so we focus on (**) and try to find an expression for s in terms of (a,b) . We observe

```
    s
=   {(2)}
    b ∨ a;s
=   {(2)}
    b ∨ a;(b ∨ a;s)
=   { ; over ∨}
    b ∨ a;b ∨ a;a;s
=   {(2) and ; over ∨}
    b ∨ a;b ∨ a;a;b ∨ a;a;a;s      etc.
```

which suggests for s the closed form, viz.

(5) $[\, s \equiv \langle \exists n : n \geq 0 : q.n \,;\, b \rangle \,]$, where

(6) $[\, q.0 \equiv \mathcal{J} \,]$ and $[\, q.(n+1) \equiv a;q.n \,]$ for all n.

To check that conjecture (5) verifies (2) we observe

```
    b ∨ a;⟨∃n: n≥0 : q.n ; b⟩
=     { ; over ∃}
    b ∨ ⟨∃n: n≥0 : a;q.n ; b⟩
=     { (6)}
    q.0;b ∨ ⟨∃n: n≥0 : q.(n+1);b⟩
=     {p.c., transforming the dummy}
    q.0;b ∨ ⟨∃n: n≥1 : q.n ; b⟩
=     { p.c.}
    ⟨∃n: n≥0 : q.n;b⟩      .
```

To check that conjecture (5) verifies (3) for

any x, we observe

$$
\begin{array}{ll}
 & [\langle \exists n: n \geq 0: q.n ; b \rangle \Rightarrow x] \\
= & \quad \{ \text{pred. calc.} \} \\
 & \langle \forall n: n \geq 0: [q.n ; b \Rightarrow x] \rangle \\
\Leftarrow & \quad \{ \text{by mathematical induction, see below} \} \\
 & [b \Rightarrow x] \wedge [a;x \Rightarrow x]
\end{array}
$$

For the mathematical induction we observe for the base:

$$
\begin{array}{ll}
 & [q.0 ; b \Rightarrow x] \\
= & \quad \{ (6) \} \\
 & [\,] ; b \Rightarrow x] \\
= & \quad \{ \text{rel. calc.} \} \\
 & [b \Rightarrow x]
\end{array}
$$

for the step:

$$
\begin{array}{ll}
 & [q.(n+1) ; b \Rightarrow x] \\
= & \quad \{ (6) \} \\
 & [a ; q.n ; b \Rightarrow x] \\
\Leftarrow & \quad \{ [a;x \Rightarrow x] \} \\
 & [a ; q.n ; b \Rightarrow a;x] \\
\Leftarrow & \quad \{ \text{monotonicity } ; \} \\
 & [q.n ; b \Rightarrow x] \,.
\end{array}
$$

By substituting $b := [\,]$ we derive for t

(7) $\quad [t \equiv \langle \exists n: n \geq 0: q.n \rangle]$,

and now we are ready to establish (4)

$$
\begin{array}{ll}
 & s \\
= & \quad \{ (5) \} \\
 & \langle \exists n: n \geq 0: q.n ; b \rangle \\
= & \quad \{ ; \text{ over } \exists \} \\
 & \langle \exists n: n \geq 0: q.n \rangle ; b \\
= & \quad \{ (7) \} \\
 & t ; b
\end{array}
$$

(End of Proof 0)

Proof 0 uses that relational composition is associative, sufficiently disjunctive and has J as its neutral element. It is the type of proof that I may have preferred 15 years ago, but now it annoys me a little, as its mathematical induction over the naturals strikes me as a foreign element.

Proof 1 Here (4) is established by means of a ping-pong argument; ping is easy.

$$
\begin{array}{ll}
 & [\, s \Rightarrow t;b \,] \\
\Leftarrow & \quad \{ (3) \text{ with } x := t;b \} \\
 & [\, b \Rightarrow t;b \,] \;\wedge\; [\, a;t;b \Rightarrow t;b \,] \\
\Leftarrow & \quad \{ \text{monotonicity of } ; \} \\
 & [\, \mathsf{J} \Rightarrow t \,] \;\wedge\; [\, a;t \Rightarrow t \,] \\
= & \quad \{ (0) \} \\
 & \text{true} \quad .
\end{array}
$$

Pong uses the exchange rules, primarily to get t all by itself in the antecedent position:

$$
\begin{array}{ll}
 & [\, t;b \Rightarrow s \,] \\
= & \quad \{ \text{left exchange} \} \\
 & [\, \neg s;\sim b \Rightarrow \neg t \,] \\
= & \quad \{ \text{contrapositive} \} \\
 & [\, t \Rightarrow \neg(\neg s;\sim b) \,] \\
\Leftarrow & \quad \{ (1) \text{ with } x := \neg(\neg s;\sim b) \} \\
 & [\, \mathsf{J} \Rightarrow \neg(\neg s;\sim b) \,] \;\wedge\; [\, a;\neg(\neg s;\sim b) \Rightarrow \neg(\neg s;\sim b) \,] \\
= & \quad \{ \text{contrapositive ; right exchange} \} \\
 & [\, \neg s;\sim b \Rightarrow \neg \mathsf{J} \,] \;\wedge\; [\, \sim a;\neg s;\sim b \Rightarrow \neg s;\sim b \,]
\end{array}
$$

$\Leftarrow$ {left exchange and monotonicities}

$[\ ];b \Rightarrow s] \ \wedge \ [\neg a;\neg s \Rightarrow \neg s]$

$=$ {$[\ ]$ and right exchange}

$[b \Rightarrow s] \wedge [a;s \Rightarrow s]$

$=$ {(2)}

true

(End of Proof 1).

In a global sense, Proof 1 isn't too bad: the way in which ping uses (0) & (3) and pong uses (1)&(2) is absolutely standard, and since, say, 1985, I can write that down without thinking. There is, however, a dual problem with pong: firstly, all those negations giving rise to contrapositives that don't really contribute to the argument — Rutger M. Dijkstra's objection — and the $\neg$, which now can be regarded as a foreign element.

Proof 2 Ping as above. For pong, we first observe that on account of monotonicities

$$[t \Rightarrow u] \wedge [u;b \Rightarrow s] \Rightarrow [t;b \Rightarrow s] \ .$$

In order to ease the demonstration of $[t \Rightarrow u]$, we want to choose a weak u. Fortunately, $x: [x;b \Rightarrow s]$ has a weakest solution (because $;$ is universally disjunctive); calling it u we have

(8) $\quad [u;b \Rightarrow s]$

(9) $\quad [x;b \Rightarrow s] \Rightarrow [x \Rightarrow u]$ for all x .

And now we observe

$$
\begin{array}{ll}
 & [t;b \Rightarrow s] \\
\Leftarrow & \quad \{\text{monotonicities}\} \\
 & [t \Rightarrow u] \wedge [u;b \Rightarrow s] \\
= & \quad \{(8)\} \\
 & [t \Rightarrow u] \\
\Leftarrow & \quad \{(1) \text{ with } x := u\} \\
 & [\,] \Rightarrow u] \wedge [a;u \Rightarrow u] \\
\Leftarrow & \quad \{(9) \text{ with } x := \,] \text{ , with } x := a;u\} \\
 & [\,];b \Rightarrow s] \wedge [a;u;b \Rightarrow s] \\
\Leftarrow & \quad \{\,] \text{ and } (8)\} \\
 & [b \Rightarrow s] \wedge [a;s \Rightarrow s] \\
= & \quad \{(2)\} \\
 & \text{true}
\end{array}
$$

(End of Proof 2)

Formulae (8) and (9) capture in the traditional manner that u is the weakest solution of $x: [x;b \Rightarrow s]$. They do not capture the consequence of the fact that $[x;b \Rightarrow s]$ is an antimonotonic function of x, i.e. that any predicate stronger than the weakest solution solves the equation as well. We take that in account by replacing (9) by

(10) $\quad [x;b \Rightarrow s] \equiv [x \Rightarrow u]$ for all x

which reduces (8) to a consequence (instantiate (10) with $x := u$).

The tradition is emerging to eliminate the identifier u and to write s/b instead - read: "s over b" - . The infix operator $/$ is now defined by

(11) $[x;y \Rightarrow z] \equiv [x \Rightarrow z/y]$ for all x,y,z.

(Notice the difference between (10) and (11): (10) was used to define u in terms of b and s, (11) defines the operator $/$.) The analogue of (8),

(12) $[(z/y);y \Rightarrow z]$ for all y,z ,

becomes a consequence, known as the law of "cancelation". We can now rewrite our last calculation as follows

$[t;b \Rightarrow s]$

$=$ { def. of $/$ }

$[t \Rightarrow s/b]$

$\Leftarrow$ { (1) with $x := s/b$ }

$[\,] \Rightarrow s/b] \;\wedge\; [a;(s/b) \Rightarrow s/b]$

$=$ { def. of $/$, twice }

$[\,];b \Rightarrow s] \;\wedge\; [a;(s/b);b \Rightarrow s]$

$\Leftarrow$ { $[\,]$ and cancelation + monotonicities }

$[b \Rightarrow s] \;\wedge\; [a;s \Rightarrow s]$

$=$ { (2) }

true .

We could do without the step that introduced u, as the monotonicities exploited have now been captured by the introduction of $/$.

Similarly, we can define $\backslash$ -read "under"- by

(13) $[x;y \Rightarrow z] \equiv [y \Rightarrow x\backslash z]$

with its own law of cancelation

(14) $[x;(x\backslash z) \Rightarrow z]$.

The fact that using formulae like (8) and (9) or (10) is a general technique raises the question: are the equations $x:[x;y\Rightarrow z]$ and $y:[x;y\Rightarrow z]$ so special that their solutions deserve special operators (or functions) to denote them?

(11) is of the form $[f.x \Rightarrow z] \equiv [x \Rightarrow g.z]$, i.e. the form of a Galois connection, and so is (13). Given an f, the introduction of its Galois partner g may be a simple way of capturing that it exists (and that f is universally disjunctive), but it is most rewarding if g has much nicer manipulative properties than f. It looks as if the answer to the question raised can only be given by the outcome of a systematic study of how we can manipulate with $/$ and $\backslash$.

Austin, 16 October 1992

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

A relational bagatelle

The first observation is, that, given the exchange rules in the form

(0) $[x;y \Rightarrow z] \equiv [\neg z;\sim y \Rightarrow \neg x]$ for all x,y,z

(1) $[x;y \Rightarrow z] \equiv [\sim x;\neg z \Rightarrow \neg y]$ for all x,y,z,

it stands to reason to rewrite "$\sim$ over ;"

$$[\sim(x;y) \equiv \sim y;\sim x] \quad \text{for all } x,y$$

in the more similar form

(2) $[x;y \Rightarrow z] \equiv [\sim y;\sim x \Rightarrow \sim z]$ for all x,y,z.

Note We use freely - i.e. without mentioning it even - that $\sim$ is an involution that distributes over the logical connectives. (End of Note.)

* * *

The second, more important observation, which I owe to Rutger M. Dijkstra, is that we can eliminate the $\neg$-signs by (i) transforming the dummy z into $\neg z$, and (ii) shunting the consequents towards the antecedents; we then get for all x,y,z

$$[x;y \wedge z \Rightarrow \text{false}] \equiv [z;\sim y \wedge x \Rightarrow \text{false}]$$

$$[x;y \wedge z \Rightarrow \text{false}] \equiv [\sim x;z \wedge y \Rightarrow \text{false}]$$

$$[x;y \wedge z \Rightarrow \text{false}] \equiv [\sim y;\sim x \wedge \sim z \Rightarrow \text{false}]\ .$$

Remark This observation is of importance because, besides eliminating many negation signs, it eliminates many calculational steps of taking the contrapositive by a simple appeal to the symmetry of $\wedge$. Rutger went one step further and eliminated the "$\Rightarrow$false" by the introduction of the "somewhere operator", the conjugate of the "everywhere operator" $[\,]$. (End of Remark.)

* * *

We now introduce 3 functions from triples (of predicates) to triples (of predicates), in which definitions "$\sim$" denotes an involution:

$$L.(x,z,y) = (z,x,\sim y)$$
$$R.(x,z,y) = (\sim x,y,z)$$
$$C.(x,z,y) = (\sim y,\sim z,\sim x)$$

It now follows that with α,β,γ any permutation of L,R,C:

(i) $\alpha\circ\alpha$ = the identity function

(ii) $\alpha = \beta\circ\gamma\circ\beta$.

I find it a bit annoying that my proof for this theorem is rather elaborate. Showing (i) requires showing that $L\circ L$, $R\circ R$, and $C\circ C$ are all three the identity function, (ii) follows from the fact that $L\circ R$, $R\circ C$ and $C\circ L$ are the same function.

By way of example we compute $L\circ L$, $L\circ R$, $R\circ C$

$$
\begin{array}{ll}
 & (L \circ L).(a,b,c) \\
= & \{\text{def. of } L \text{ with } x,y,z := a,c,b\} \\
 & L.(b,a,\sim c) \\
= & \{\text{def. of } L \text{ with } x,y,z := b,\sim c,a\} \\
 & (a,b,\sim\sim c) \\
= & \{\sim \text{ is an involution}\} \\
 & (a,b,c)
\end{array}
$$

$$
\begin{array}{ll}
 & (L \circ R).(a,b,c) \\
= & \{\text{def. of } R\} \\
 & L.(\sim a,c,b) \\
= & \{\text{def. of } L\} \\
 & (c,\sim a,\sim b)
\end{array}
\qquad
\begin{array}{ll}
 & (C \circ L).(a,b,c) \\
= & \{\text{def. of } L\} \\
 & C.(b,a,\sim c) \\
= & \{\text{def. of } C\} \\
 & (c,\sim a,\sim b)
\end{array}
$$

And now we can, for instance, conclude

$$
\begin{array}{ll}
 & R \\
= & \{L \circ L \text{ is the identity function}\} \\
 & L \circ L \circ R \\
= & \{L \circ R = C \circ L\} \\
 & L \circ C \circ L
\end{array}
$$

From (ii): $\alpha = \beta \circ \gamma \circ \beta$ we conclude that of the triple (0), (1), and (2), we can derive each from the two others, and that we can do so in (precisely) two ways. To show how, for instance (0) can be derived from (1) and (2), we observe

$$
\begin{array}{ll}
\quad [\,x;y \Rightarrow z\,] & \quad [\,x;y \Rightarrow z\,] \\
= \quad \{(1)\} & = \quad \{(2)\} \\
\quad [\,\neg x;\neg z \Rightarrow \neg y\,] & \quad [\,\neg y;\neg x \Rightarrow \neg z\,] \\
= \quad \{(2)\} & = \quad \{(1)\} \\
\quad [\,\neg\neg z;\, x \Rightarrow \neg\neg y\,] & \quad [\,y;\neg\neg z \Rightarrow \neg\neg x\,] \\
= \quad \{(1)\} & = \quad \{(2)\} \\
\quad [\,\neg\neg z;\neg y \Rightarrow \neg x\,] & \quad [\,\neg z;\,\neg y \Rightarrow \neg x\,] \,,
\end{array}
$$

a guise in which these proofs may look quite surprising, but now we know why they could be constructed on the principle "there is only one thing you can do".

The nice thing of this bagatelle is that the algebraic theorem $\alpha = \beta \circ \gamma \circ \beta$ yields the proof-theoretic result that (0), (1), and (2) are <u>not</u> independent results, as each follows from the other two.

Austin, 22 October 1992

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

Notational considerations and the relational calculus

W.H.J. Feijen and A.J.M. van Gasteren -and others- wrote the exchange rules of the relational calculus

(0a) the "left-exchange"

$$[\, x;y \Rightarrow z \,] \equiv [\, \neg z;{\sim}y \Rightarrow \neg x \,]$$

(0b) the "right-exchange"

$$[\, x;y \Rightarrow z \,] \equiv [\, {\sim}x;\neg z \Rightarrow \neg y \,]$$

The appeal to an exchange rule of the above form is often followed or preceded by shunting or taking the contrapositive. Rutger M. Dijkstra realized that these manipulations are caused by the fact that the implication is an asymmetric way of writing the disjunction, but instead of writing $[\, \neg(x;y) \vee z \,]$, which would have introduced a negation in front of the composition, he pushed that negation as far to the outside as he could. After the introduction of the "somewhere operator" $\langle\, \rangle$, which is the conjugate of the "everywhere operator"

$$\langle x \rangle \equiv \neg[\, \neg x \,] \;,$$

he formulated the exchange rules

(1a) $$\langle x;y \wedge z \rangle \equiv \langle z;{\sim}y \wedge x \rangle$$

(1b) $$\langle x;y \wedge z \rangle \equiv \langle {\sim}x;z \wedge y \rangle \;.$$

For a number of calculations this was a con-

siderable improvement.

Formulae (0) tell us that equations

$$x:[x;y \Rightarrow z] \quad \text{and} \quad y:[x;y \Rightarrow z] \ .$$

For the weakest solution of the former, C.A.R. Hoare and He Jifeng introduced $y\backslash z$ and for the weakest solution of the latter z/x. R.C. Backhouse interchanged the role of $\backslash$ and $/$, and everybody - Hoare and He Jifeng included - agreed that this was an improvement. The $/$ (read: "over") and $\backslash$ (read: "under") are now introduced by the "factorization rules"

$$(2a) \qquad [x;y \Rightarrow z] \equiv [x \Rightarrow z/y]$$

$$(2b) \qquad [x;y \Rightarrow z] \equiv [y \Rightarrow x\backslash z] \ .$$

Instantiating these definitions with $x := z/y$ and $y := x\backslash z$ respectively, we get the "cancellation rules"

$$(3a) \qquad [(z/y);y \Rightarrow z]$$

$$(3b) \qquad [x;(x\backslash z) \Rightarrow z] \ .$$

In the above form, the cancellation rules have an undeniable appeal, but so far, $/$ and $\backslash$ are only part of nomenclature - viz. a way of writing the weakest solutions of special types of equations. The $/$ and $\backslash$ can only contribute really, provide we know their properties, but which? To quote R.M. Dykstra "The number of

lemmas that can be formulated about factors is vast. I will refrain from listing them because the properties of "under" and "over" are much less nice than those of composition and -given the latter- factors are in fact redundant."

The above-mentioned redundancy can be made more explicit by applying to (0) the contrapositive:

$$[x;y \Rightarrow z] \equiv [x \Rightarrow \neg(\neg z;\sim y)]$$

$$[x;y \Rightarrow z] \equiv [y \Rightarrow \neg(\sim x;\neg z)] \quad .$$

Comparing these formulae with (2), we conclude

(4a) $\quad [z/y \equiv \neg(\neg z;\sim y)]$

(4b) $\quad [x\backslash z \equiv \neg(\sim x;\neg z)]$.

Observing all the above, we see that R.M. Dykstra introduced the somewhere operator and that Hoare, He Jifeng introduced, and Backhouse maintained "over" and "under", all because no one knew how to negate a composition.

* * *

Before pursuing the above line of thought, we turn for a moment to our unary operators and their notation. Tarski denotes negation

by an "over bar", so that, for instance, de Morgan's law can be rendered by

$$[\overline{x \wedge y} \equiv \bar{x} \vee \bar{y}] \quad .$$

The attraction of this notation is that it introduces neither the need for parentheses nor priority rules. There is the typographical disadvantage that the bars can get long and the formulae high — I quote from a report I recently received the following formula

$$\overline{\overline{x + x + \bar{x}} + \overline{x + \bar{x}} + \bar{x} + \overline{\bar{x} + \bar{x}}} \quad - \quad ,$$

there is the methodological disadvantage that it does not really generalize — it is only the line segment that lends itself nicely to extension —, and finally the fundamental disadvantage that there is — at least to the best of my knowledge — no manageable formalism (like BNF) for the definition of such multi-dimensional syntaxes. Hoare and He Jifeng stick to Tarski's over bar, R.C. Backhouse adopts the convention of the prefix "¬" with a binding power higher than ";".

Then there is the transposition. Tarski uses the "over cup", so that its being an involution can be rendered by

$$[\overset{\smile}{x} = x] \quad ,$$

but the rendering of $[\sim(x;y) \equiv \sim y;\sim x]$ requires the stretched cup:

$$[\overbrace{x;y}^{\cup} \equiv y^{\cup};x^{\cup}] .$$

Hoare and He Jifeng define $\breve{R} = \bar{R}\backslash\bar{I}$ and then use the right-hand side in their calculations — I quote, for instance,

$$(\bar{Q}\backslash\bar{I}) \leq \overline{((\bar{\bar{R}}\backslash\bar{\bar{I}})\backslash(\bar{\bar{P}}\backslash\bar{\bar{I}}))\backslash\bar{I}} \quad ;$$

I think that we can agree that here something went wrong. Commutation with negation can be rendered by

$$[\breve{\bar{x}} \equiv \bar{\breve{x}}] .$$

Because transposition commutes with negation, Backhouse opts for a postfix cup. Commutation can now be rendered by

$$[(\neg x)^{\cup} \equiv \neg(x^{\cup})] ,$$

i.e. for the combination of the two we get by mere omission of the parentheses the unique form $\neg x^{\cup}$, whereas, so far, I have to choose between $\sim\neg x$ and $\neg\sim x$. An alternative way of avoiding this choice is the introduction of a third operator. Denoting it by "//" and calling it "mirror", we can define it by

$$(5) \qquad [//x \equiv \sim\neg x] .$$

We give it the same high binding power as $\sim$ and $\neg$.

In EWD982 "Relational Calculus according to ATAC", the remark was already made that when you have 2 commuting involutions, you have a 3rd, in particular, with the triple (α, β, γ) any permutation of the triple $(\neg, \sim, /\!/)$, we have

$$[\alpha\alpha x \equiv x] \quad \text{and} \quad [\alpha\beta x = \gamma x] \quad \text{for all } x,$$

from which it follows that we never need to apply more than one of them. The introduction of $/\!/$ can be defended on the grounds that it does justice to the underlying symmetry between the three operators.

* * *

After this interlude on the operator $/\!/$, we continue our considerations triggered by "somewhere", "over", and "under". It is quite possible to negate a composition! As a matter of fact, Tarski has shown us how to do it: introduce a new operator for the conjugate of the composition. We call it the "confrontation" and denote it by an infix "!" that we give the same binding power as the ";". As conjugate of composition, confrontation, defined by

$$(6) \qquad [x \,!\, y \equiv \neg(\neg x ; \neg y)] \quad ,$$

- is associative
- is universally conjunctive in both arguments

- has $\neg J$ as its neutral element.

Moreover, we have now the "vocabulary" to describe how our three unary operators interact with composition and with confrontation:

$$(7)\quad [\neg(x;y) \equiv \neg x \,!\, \neg y]$$

$$[\sim(x;y) \equiv \sim y ; \sim x]$$

$$[/\!/(x;y) \equiv /\!/y \,!\, /\!/x]$$

$$[\neg(x\,!\,y) \equiv \neg x ; \neg y]$$

$$[\sim(x\,!\,y) \equiv \sim y \,!\, \sim x]$$

$$[/\!/(x\,!\,y) \equiv /\!/y ; /\!/x] \quad .$$

Instead of R.M. Dykstra's exchange rules (1) with the "somewhere operator", we can now write

$$(8a)\qquad [x\,!\,y \vee z] \equiv [z\,!\,\sim y \vee x]$$

$$(8b)\qquad [x\,!\,y \vee z] \equiv [\sim x\,!\,z \vee y] \quad .$$

Moreover, we have now a shorter definition of "over" and "under": instead of (4) we could now write

$$(9a)\qquad [z/y \equiv z\,!\,/\!/y]$$

$$(9b)\qquad [x\backslash z \equiv /\!/x\,!\,z]$$

and the factorization and cancellation rules (2)

and (3) can be rewritten accordingly, e.g.

(10a) $$[x;y \Rightarrow z] \equiv [x \Rightarrow z!\!/\!/y]$$

(10b) $$[x;y \Rightarrow z] \equiv [z \Rightarrow /\!/x!y] \ .$$

and instantiations like

$$[(z!\!/\!/y);y \Rightarrow z]$$

$$[x \Rightarrow (x;y)!\!/\!/y] \ .$$

Formulae (9) suggest to me to forget about "over" and "under" and to trade them for "mirror" and "confrontation." I ended EWD 1136 with the suggestion of "a systematic study of how we can manipulate with / and \". Some results already emerge. For instance, when we ask ourselves how associative / is - it isn't - we find

$$[(x/y)/z \equiv x/(z;y)]$$

which is nothing but the associativity of ! in disguise. This obfuscation is very similar to the complications of the contrapositive and shunting, where the implication sign has disguised the associativity of the disjunction.

* * *

When I introduced the conjugate of the composition into my calculations, I was temporarily unaware that Tarski had already introduced the operator. David A. Naumann reminded me of it.

Tarski called the composition "relative multiplication" and the confrontation "relative addition". He denoted the former, like us, by ";" and, in a graphical pun, the latter by "$\dagger$". It is surprising to see that Tarski's "relative addition" was ignored and "over" and "under" were introduced, as all the people involved knew Tarski's article. Under the assumption that trading $/$ and $\backslash$ for $/\!/$ and $\backslash\!\backslash$ is a manipulative simplification —and, to say the truth, I am almost certain it is—, I can only think of the following explanations.

(i) People don't really manipulate with $/$ and $\backslash$, i.e. z/y and $x\backslash z$ are more notations for the weakest solutions of two special equations —as exemplified by factorization and cancellation rules— than that $/$ and $\backslash$ are viewed as operators with pleasant properties.

(ii) The lack of $/\!/$ for $\neg\nu$ made $/$ and $\backslash$ more effective abbreviations.

(iii) People were put off —and rightly so— by Tarski's misleading terminology: the analogy with arithmetic is too shallow. (Of course, $/$ and $\backslash$ are also inspired by arithmetic analogy; the analogy makes me suspicious, but I have seen others attracted by it.)

(iv) With $\ddagger$, Tarski chose the wrong symbol. I first used $\dagger$, and that worked, but then I realized that I did not have a role for its obvious partner d, so I switched to ϕ, and that worked too. After D.A. Naumann had reminded me that my invention had been anticipated by Tarski, I felt that the least I could do was to use Tarski's symbol, but, no matter how hard I tried, it did not work. (I tried a smaller +, e.g. $\overset{+}{;}$, but still the symbol was too big for its high binding power. For formulae with Tarski's symbol, I found no way of spacing that my hand-eye coordination could adopt, I gave up and, upon analysis of my difficulties, decided to try $!$.)

About the shape of the character // for "mirror" I still have doubts.

Austin, 5 November 1992

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

PREDICATE CALCULUS - PART 0

- Operators in the order of decreasing binding power:

.	functional application
¬	negation
∧ , ∨	conjunction, disjunction
⇒ , ⇐	implication, consequence
≡ , ≠	equivalence, difference

- Leibniz's Rule

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

- Associativity

Equivalence, difference, conjunction, and disjunction are associative. Equivalence and difference are mutually associative. Functional application is left-associative, i.e.

$$[f.x.y = (f.x).y]$$

- Symmetry and Idempotence

Equivalence, difference, conjunction, and disjunction are symmetric. Conjunction and disjunction are idempotent.

- Unit elements and zero elements

Units:
$$[X \equiv true \equiv X]$$
$$[X \not\equiv false \equiv X]$$
$$[X \wedge true \equiv X]$$
$$[X \vee false \equiv X]$$

Zeroes:
$$[X \wedge false \equiv false]$$
$$[X \vee true \equiv true]$$

- ## Elementary Rules

Distribution

$[\ X \vee (Y \equiv Z) \equiv X \vee Y \equiv X \vee Z\ ]$

$[\ X \wedge (Y \equiv Z) \equiv X \wedge Y \equiv X \wedge Z \equiv X\ ]$

$[\ X \vee (Y \wedge Z) \equiv (X \vee Y) \wedge (X \vee Z)\ ]$

$[\ X \wedge (Y \vee Z) \equiv (X \wedge Y) \vee (X \wedge Z)\ ]$

Absorption

$[\ X \vee (X \wedge Y) \equiv X\ ]$

$[\ X \wedge (X \vee Y) \equiv X\ ]$

$[\ X \vee Y \Leftarrow X\ ]$

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

Complement

$[\ X \vee (\neg X \wedge Y) \equiv X \vee Y\ ]$

$[\ X \wedge (\neg X \vee Y) \equiv X \vee Y\ ]$

Golden Rule

$[\ X \wedge Y \equiv X \equiv Y \equiv X \vee Y\ ]$

Implication

$[\ X \Rightarrow Y \equiv X \vee Y \equiv Y\ ]$

$[\ X \Rightarrow Y \equiv X \wedge Y \equiv X\ ]$

$[\ X \Rightarrow Y \equiv \neg X \vee Y\ ]$

$[\ X \Leftarrow Y \equiv Y \Rightarrow X\ ]$

Negation

$[\ \neg (X \equiv Y) \equiv \neg X \equiv Y\ ]$

$[\ \neg X \vee X \equiv \text{true}\ ]$

$[\ \neg\neg X \equiv X\ ]$

$[\ \neg (X \vee Y) \equiv \neg X \wedge \neg Y\ ]$

$[\ \neg (X \wedge Y) \equiv \neg X \vee \neg Y\ ]$

- Exercises

0. $[\ P \wedge (X \equiv Y \equiv Z) \equiv P \wedge X \equiv P \wedge Y \equiv P \wedge Z\ ]$
1. $[\ X \wedge (X \equiv Y) \equiv X \wedge Y\ ]$
2. $[\ (X \equiv X \wedge Y) \vee (Y \equiv X \wedge Y)\ ]$
3. $[\ X \wedge (X \Rightarrow Y) \equiv X \wedge Y\ ]$
4. $[\ X \wedge Y \Rightarrow Z \equiv X \Rightarrow (Y \Rightarrow Z)\ ]$
5. $[\ (X \Rightarrow Y) \wedge (Y \Rightarrow Z) \Rightarrow (X \Rightarrow Z)\ ]$
6. $[\ (X \Rightarrow Y) \vee (Y \Rightarrow Z)\ ]$
7. $[\ (X \Rightarrow Y) \wedge (Y \Rightarrow X) \equiv X \equiv Y\ ]$
8. $[\ X \Rightarrow \text{true}\ ]$
9. $[\ \text{true} \Rightarrow X \equiv X\ ]$
10. $[\ (X \equiv Y) \Rightarrow (X \Rightarrow Y)\ ]$
11. $[\ X \vee Y \Rightarrow Z \equiv (X \Rightarrow Z) \wedge (Y \Rightarrow Z)\ ]$
12. $[\ X \Rightarrow Y \wedge Z \equiv (X \Rightarrow Y) \wedge (X \Rightarrow Z)\ ]$
13. $[\ X \wedge Y \Rightarrow Z \equiv (X \Rightarrow Z) \vee (Y \Rightarrow Z)\ ]$
14. $[\ X \Rightarrow Y \vee Z \equiv (X \Rightarrow Y) \vee (X \Rightarrow Z)\ ]$
15. $[\ X \Rightarrow (Y \equiv Z) \equiv X \Rightarrow Y \equiv X \Rightarrow Z\ ]$
16. $[\ (X \equiv Y) \Rightarrow Z \equiv X \Rightarrow Z \equiv Y \Rightarrow Z \equiv Z\ ]$
17. $[\ (X \equiv Y \equiv Z) \Rightarrow P \equiv X \Rightarrow P \equiv Y \Rightarrow P \equiv Z \Rightarrow P\ ]$
18. $[\ (X \Rightarrow Y) \Rightarrow ((Y \Rightarrow Z) \Rightarrow (X \Rightarrow Z))\ ]$
19. $[\ (Y \Rightarrow Z) \Rightarrow ((X \Rightarrow Y) \Rightarrow (X \Rightarrow Z))\ ]$
20. $[\ X \Rightarrow (Y \Rightarrow Z) \equiv Y \Rightarrow (X \Rightarrow Z)\ ]$
21. $[\ X \Rightarrow (Y \Rightarrow Z) \equiv (X \Rightarrow Y) \Rightarrow (X \Rightarrow Z)\ ]$
22. $[\ X \Rightarrow (Y \Rightarrow Z) \equiv X \wedge Y \Rightarrow X \wedge Z\ ]$
23. $[\ X \vee (Y \Rightarrow Z) \equiv X \vee Y \Rightarrow X \vee Z\ ]$
24. $[\ (X \Rightarrow Y) \Rightarrow (X \vee Z \Rightarrow Y \vee Z)\ ]$

25. $[(X \Rightarrow Y) \Rightarrow (X \wedge Z \Rightarrow Y \wedge Z)]$

26. $[((X \Rightarrow Y) \Rightarrow Z) \Rightarrow (X \Rightarrow (Y \Rightarrow Z))]$

27. $[\text{false} \Rightarrow X]$

28. $[X \Rightarrow \text{false} \equiv \neg X]$

29. $[\neg X \equiv X \equiv \text{false}]$

30. $[\neg X \Rightarrow X \equiv X]$

31. $[X \Rightarrow Y \equiv \neg X \Leftarrow \neg Y]$

32. $[X \wedge Y \equiv X \wedge Z \equiv \neg X \vee (Y \equiv Z)]$

33. $[X \wedge \neg Y \equiv Y \wedge \neg X \equiv X \equiv Y]$

34. $[X \wedge (Y \vee Z) \equiv X \wedge Y \equiv (X \wedge Z \Rightarrow Y)]$

35. $[X \vee (Y \wedge Z) \equiv X \vee Y \equiv (X \vee Z \Leftarrow Y)]$

36. $[(X \wedge Y) \vee (\neg X \wedge Z) \equiv (\neg X \vee Y) \wedge (X \vee Z)]$

37. $[[X]] \equiv [X]$

38. $[X \Rightarrow Y] \Rightarrow ([X] \Rightarrow [Y])$

39. $[X] \vee [Y] \Rightarrow [X \vee Y]$

40. $[X] \wedge [Y] \equiv [X \wedge Y]$

41. $[X \equiv Y] \Rightarrow ([X] \equiv [Y])$

/.

PREDICATE CALCULUS - PART 1

- Elementary Rules

Trading

$$[\ (\underline{A}x : r.x \wedge s.x : p.x) \equiv (\underline{A}x : r.x : \neg s.x \vee p.x)\ ]$$
$$[\ (\underline{E}x : r.x \wedge s.x : p.x) \equiv (\underline{E}x : r.x : s.x \wedge p.x)\ ]$$

Splitting the range

$$[\ (\underline{A}x : r.x \vee s.x : p.x) \equiv (\underline{A}x : r.x : p.x) \wedge (\underline{A}x : s.x : p.x)\ ]$$
$$[\ (\underline{E}x : r.x \vee s.x : p.x) \equiv (\underline{E}x : r.x : p.x) \vee (\underline{E}x : s.x : p.x)\ ]$$

Distribution

$$[\ Q \vee (\underline{A}x :: p.x) \equiv (\underline{A}x :: Q \vee p.x)\ ]$$
$$[\ Q \wedge (\underline{E}x :: p.x) \equiv (\underline{E}x :: Q \wedge p.x)\ ],$$

and for non-empty ranges :

$$[\ Q \wedge (\underline{A}x :: p.x) \equiv (\underline{A}x :: Q \wedge p.x)\ ]$$
$$[\ Q \vee (\underline{E}x :: p.x) \equiv (\underline{E}x :: Q \vee p.x)\ ],$$

and for boolean range :

$$[\ (\underline{A}x :: p.x)\ ] \equiv (\underline{A}x :: [p.x])$$

One-point-rules

$$[\ (\underline{A}x : [x=y] : p.x) \equiv p.y\ ]$$
$$[\ (\underline{E}x : [x=y] : p.x) \equiv p.y\ ]$$

Dummy renaming

$$[\ (\underline{Q}x : r.x : p.x) \equiv (\underline{Q}y : r.y : p|y)\ ]$$

- Exercises

0. $[\ (\underline{A}x :: p.x \wedge q.x) \equiv (\underline{A}x :: p.x) \wedge (\underline{A}x :: q.x)\ ]$
1. $[\ (\underline{E}x :: p.x \vee q.x) \equiv (\underline{E}x :: p.x) \vee (\underline{E}x :: q.x)\ ]$

2. $[(\underline{\forall} x :: p.x \equiv q.x) \Rightarrow (\underline{\forall} x :: p.x) \equiv (\underline{\forall} x :: q.x)]$

3. $[(\underline{\forall} x :: p.x \equiv q.x) \Rightarrow (\underline{\exists} x :: p.x) \equiv (\underline{\exists} x :: q.x)]$

4. $[(\underline{\forall} x :: p.x \Rightarrow q.x) \Rightarrow (\underline{\forall} x :: p.x) \Rightarrow (\underline{\forall} x :: q.x)]$

5. $[(\underline{\forall} x :: p.x \Rightarrow q.x) \Rightarrow (\underline{\exists} x :: p.x) \Rightarrow (\underline{\exists} x :: q.x)]$

6. $[(\underline{\forall} x :: p.x) \Rightarrow (\underline{\forall} x :: p.x \vee q.x)]$

7. $[(\underline{\exists} x :: p.x) \Rightarrow (\underline{\exists} x :: p.x \vee q.x)]$

8. $[(\underline{\forall} x : r.x \vee s.x : p.x) \Rightarrow (\underline{\forall} x : r.x : p.x)]$

9. $[(\underline{\exists} x : r.x \vee s.x : p.x) \Leftarrow (\underline{\exists} x : r.x : p.x)]$

10. $[(\underline{\forall} x :: p.x) \vee (\underline{\forall} x :: q.x) \Rightarrow (\underline{\forall} x :: p.x \vee q.x)]$

11. $[(\underline{\exists} x :: p.x) \wedge (\underline{\exists} x :: q.x) \Leftarrow (\underline{\exists} x :: p.x \wedge q.x)]$

12. $[(\underline{\forall} x : r.x : p.x) \equiv (\underline{\forall} x : r.x : r.x \wedge p.x)]$

13. $[(\underline{\exists} x : r.x : p.x) \equiv (\underline{\exists} x : r.x : r.x \Rightarrow p.x)]$

14. $[Q \Rightarrow (\underline{\forall} x :: p.x) \equiv (\underline{\forall} x :: Q \Rightarrow p.x)]$

15. $[(\underline{\exists} x :: p.x) \Rightarrow Q \equiv (\underline{\forall} x :: p.x \Rightarrow Q)]$

16. $[(\underline{\forall} x : r.x : r.x \vee p.x) \equiv \text{true}]$

17. $[(\underline{\exists} x : r.x : \neg r.x \wedge p.x) \equiv \text{false}]$

18. $[(\underline{\forall} y : r.y : (\underline{\forall} x : r.x : p.x) \Rightarrow p.y)]$

19. $[(\underline{\forall} y : r.y : p.y \Rightarrow (\underline{\exists} x : r.x : p.x))]$

20. $[(\underline{\forall} x :: Q \wedge p.x) \equiv (Q \wedge (\underline{\forall} x :: p.x)) \vee (\underline{\forall} x :: \text{false})]$

21. $[(\underline{\exists} x :: Q \vee p.x) \equiv (Q \vee (\underline{\exists} x :: p.x)) \wedge (\underline{\exists} x :: \text{true})]$

22. For non-empty range,
$[(\underline{\forall} x :: p.x) \Rightarrow Q \equiv (\underline{\exists} x :: p.x \Rightarrow Q)]$

23. For non-empty range,
$[\ Q \Rightarrow (\underline{E}x :: p.x) \equiv (\underline{E}x :: Q \Rightarrow p.x)\ ]$

24. $[\ (\underline{A}y : r.y : (\underline{E}x :: r.x))\ ]$

25. $[\ (\underline{A}x :: p.x \vee q.x) \Rightarrow (\underline{E}x :: p.x) \vee (\underline{A}x :: q.x)\ ]$

26. $[\ (\underline{A}x :: p.x) \Rightarrow (\underline{E}x :: p.x) \equiv (\underline{E}x :: true)\ ]$

27. $[\ (\underline{E}x :: p.x) \Rightarrow (\underline{A}x :: p.x) \equiv (\underline{A}x, y :: p.x \equiv q.x)\ ]$

28. $[\ (\underline{A}x : r.x \wedge [x = y] : p.x) \equiv \neg r.y \vee p.y\ ]$

29. $[\ (\underline{E}x : r.x \wedge [x = y] : p.x) \equiv r.y \wedge p.y\ ]$

30. $[\ (\underline{A}x :: (\underline{E}y :: p.x.y)) \vee (\underline{A}y :: (\underline{E}x :: \neg p.x.y))\ ]$

31. $[\ (\underline{A}x, y : r.y \wedge [x = y] : p.x.y) \equiv (\underline{A}y : r.y : p.y.y)\ ]$

32. $[\ (\underline{A}x : 0 \leq x \wedge x < y+1 : p.x)$
$\equiv (\underline{A}x : 0 \leq x \wedge x < y : p.x) \wedge (y < 0 \vee p.y)\ ]$

33. $[\ (\underline{E}x : 0 \leq x \wedge x < y+1 : p.x)$
$\equiv (\underline{E}x : 0 \leq x \wedge x < y : p.x) \vee (0 \leq y \wedge p.y)\ ]$

34. $[\ (\underline{A}x, y : r.x.y \wedge x < y+1 : p.x.y)$
$\equiv (\underline{A}x, y : r.x.y \wedge x < y : p.x.y) \wedge (\underline{A}x : r.x.x : p.x.x)\ ]$

35. $[\ (\underline{A}x : 0 \leq x : 1 \leq x) \equiv false\ ]$

36. $[\ (\underline{A}y :: y \leq 0 \Rightarrow (\underline{A}x : 0 \leq x \wedge x < y : p.x))\ ]$

37. Rewrite $(\underline{A}x : 0 \leq x \wedge x \leq y : x < |y|)$ into an equivalent expression that no longer contains a quantifier

38. $[\ (\underline{A}x, y : x < y : (\underline{A}z :: x < z \vee z < y))\ ]$

39. Find all left-zero, right-zero, left-unit, and right-unit elements of operator $\Rightarrow$.

40. Prove that a conjunctive (disjunctive) predicate transformer is monotonic.

41. Prove that, for monotonic f,
$[\ f.(\underline{A}X :: X) \Rightarrow (\underline{A}X :: f.X)\ ]$, and
$[\ (\underline{E}X :: f.X) \Rightarrow f.(\underline{E}X :: X)\ ]$.

42. $[X \Rightarrow Y] \equiv (\underline{A}H :: [H \Rightarrow X] \Rightarrow [H \Rightarrow Y])$

43. $[X \Rightarrow Y] \equiv (\underline{A}H :: [Y \Rightarrow H] \Rightarrow [X \Rightarrow H])$

44. $[X \equiv Y] \equiv (\underline{A}H :: [H \vee X] \equiv [H \vee Y])$

45. Prove that predicate transformers f and g satisfying
$[f.X \vee Y] \equiv [X \vee g.Y]$,
for all X and Y , are universally conjunctive.

/.

MISCELLANEOUS

• A little calculus of Δ and ∇

For any two numbers x and y we deem to be defined the expressions $x \Delta y$ and $x \nabla y$. They satisfy – by postulate –

$$[\, z \geq x \Delta y \;\equiv\; z \geq x \;\wedge\; z \geq y \,]$$

$$[\, z \leq x \nabla y \;\equiv\; z \leq x \;\wedge\; z \leq y \,] \;;$$

here, the square brackets denote universal quantification over x, y, and z. We now can prove a lot of properties, relating Δ, ∇, and other arithmetic operators.

0. Addition distributes over both Δ and ∇.

1. Δ and ∇ are associative, symmetric, and idempotent

2. $[\, x \Delta y \geq x \;\wedge\; x \Delta y \geq y \,]$
   $[\, x \nabla y \leq x \;\wedge\; x \nabla y \leq y \,]$

3. $[\, x \Delta y = x \;\vee\; x \Delta y = y \,]$
   $[\, x \nabla y = x \;\vee\; x \nabla y = y \,]$

4. $[\, z \leq x \Delta y \;\equiv\; z \leq x \;\vee\; z \leq y \,]$
   $[\, z \geq x \nabla y \;\equiv\; z \geq x \;\vee\; z \geq y \,]$

5. Δ and ∇ distribute over each other.

6. $[\, -(x \Delta y) = (-x) \nabla (-y) \,]$
   $[\, -(x \nabla y) = (-x) \Delta (-y) \,]$

7. $[\, x \Delta y = x \nabla y \;\equiv\; x = y \,]$

8. $[\, x \Delta y = x \;\equiv\; y = x \nabla y \,]$

9. $[\, x \Delta y = x \;\equiv\; x \geq y \,]$

10. $[\, x \nabla y = x \;\equiv\; x \leq y \,]$

11. $[\, z \geq 0 \;\Rightarrow\; z \times (x \Delta y) = (z \times x) \Delta (z \times y) \,]$

And so on.

Sometimes it is useful to have on hand the two outlandish, non-numeric values "pinf" and "minf", which satisfy

$$[\ x \mathbin{\Delta} pinf = pinf\ ] \qquad [\ x \mathbin{\Delta} minf = x\ ]$$
$$[\ x \mathbin{\nabla} pinf = x\ ] \qquad [\ x \mathbin{\nabla} minf = minf\ ] .$$

A highly common interpretation is $\underline{max}$ for Δ and $\underline{min}$ for ∇. We shall stick to this interpretation and notation in what follows.

• On _MAX_ and _MIN

For finite and (mostly) nonempty ranges $r.x$ and for integer expressions $f.x$, we will consider expressions

$$(\underline{MAX}\ x :\ r.x :\ f.x) \qquad \text{and} \qquad (\underline{MIN}\ x :\ r.x :\ f.x) .$$

They can be defined by (, omitting the range)

$$[\ z \geq (\underline{MAX}\ x ::\ f.x) \ \equiv\ (\underline{A}x ::\ z \geq f.x)\ ]$$
$$[\ z \leq (\underline{MIN}\ x ::\ f.x) \ \equiv\ (\underline{A}x ::\ z \leq f.x)\ ] .$$

They inherit many properties from universal quantification and from $\underline{max}$ and $\underline{min}$. We will list some, for $\underline{MAX}$ mainly.

12. Splitting the range

$$[(\underline{MAX}\ x :\ r.x \vee s.x :\ f.x)$$
$$=\ (\underline{MAX}\ x :\ r.x :\ f.x)\ \underline{max}\ (\underline{MAX}\ x :\ s.x :\ f.x)\ ]$$

13. Splitting the term

$$[\ (\underline{MAX}\ x ::\ f.x\ \underline{max}\ g.x)$$
$$=\ (\underline{MAX}\ x ::\ f.x)\ \underline{max}\ (\underline{MAX}\ x ::\ g.x)\ ]$$

14. One-point-rule

$$[\ (\underline{MAX}\ x :\ [x = y] :\ f.x) \ =\ f.y\ ]$$

15. Distribution

$$[\ (\underline{MAX}\ x ::\ f.x\ \underline{min}\ g.y) = (\underline{MAX}\ x ::\ f.x)\ \underline{min}\ g.y\ ]$$
$$[\ (\underline{MAX}\ x ::\ f.x\ \underline{max}\ g.y) = (\underline{MAX}\ x ::\ f.x)\ \underline{max}\ g.y\ ]$$
$$[\ (\underline{MAX}\ x ::\ f.x + g.y) = (\underline{MAX}\ x ::\ f.x) + g.y\ ]$$

It should be noted that latter two rules hold for non-empty ranges only.

16. $[\ z = (\underline{MAX} x :: f.x)$
$\equiv \quad (\underline{\exists} x :: z = f.x) \ \wedge \ (\underline{A} x :: f.x \leq z)\ ]$

17. $[\ f.z = (\underline{MAX} x : r.x : f.x)$
$\equiv \quad r.z \ \wedge \ (\underline{A} x : r.x : f.x \leq f.z)\ ]$

18. $[\ -(\underline{MAX} x :: f.x) \quad = \quad (\underline{MIN} x :: -f.x)\ ]$

Etcetera.

- On $\underline{S}$ and $\underline{N}$

For finite ranges $r.x$ and integer expression $f.x$, we will consider expressions

$(\underline{S} x : r.x : f.x)$ and $(\underline{N} x : r.x)$

For the former we have - by definition -

$[\ (\underline{S} x : false : f.x) = 0\ ]$
$[\ (\underline{S} x : [x = y] : f.x) = f.y\ ]$
$[\ (\underline{S} x : r.x \vee s.x : f.x)$
$\quad = (\underline{S} x : r.x : f.x) + (\underline{S} x : s.x : f.x)$
$\quad\quad - (\underline{S} x : r.x \wedge s.x : f.x)\ ]$

Furthermore, all familiar rules about finite summation apply.

The latter is defined by

$[\ (\underline{N} x : r.x) = (\underline{S} x : r.x : 1)\ ]$,

so that it is natural-valued. For the corresponding one-point-rule we will often use the format

$[\ (\underline{N} x : [x = y] \wedge r.x)$
$\quad =$ if $r.y \rightarrow 1$
$\quad\quad [\!] \ \neg r.y \rightarrow 0$
$\quad\quad$ fi
$\ ]$

* * *

Playing with dagger and star, i.e. with transitive closures

There are many different ways in which one can present the regularity calculus, but in any such presentation the unary operator $*$ – star –, to denote the reflexive transitive closure of a relation, will see the light in a pretty early stage. Over the last years (or decades?) it has become customary to introduce $*r$ as the strongest solution of the equation

(0) $\quad x : [\, J \vee r ; x \Rightarrow x \,]$.

But, in fact, this is a little bit unfortunate because it can also be introduced as the strongest solution of

(1) $\quad x : [\, J \vee x ; r \Rightarrow x \,]$.

The choice between (0) and (1) being immaterial, we have to make an immaterial choice in an early stage of such a development of the regularity calculus, and that is not very satisfactory.

Instead of starting from the skew equations (0) or (1), it is much nicer to preserve symmetry and to introduce $*r$ as the strongest solution of

(2) $\quad x : [\, J \vee r \vee x ; x \Rightarrow x \,]$.

From this definition it is, for instance, immediately clear that any solution x satisfies

$$[\, x;x \Rightarrow x \,] \quad ,$$

i.e. any solution is transitive, in particular $*r$. The only little disadvantage of (2) over (1) or (0) is that the defining equation for $*r$ has become slightly more complicated. Therefore, Edsger W. Dijkstra proposed to start our investigations from $\dagger$ - dagger -, to denote the ordinary transitive closure. His proposal is to define $\dagger r$ as the strongest solution of

(3) $\quad x : [\, r \vee x;x \Rightarrow x \,]$,

and then $*r$ by, for instance,

(4) $\quad [\, *r \equiv \dagger(J \vee r) \,]$.

During one-and-a-half of its session, the ETAC has explored this proposal (in the presence of EWD), and the purpose of this note is to record our findings

$\dagger \quad \dagger \quad \dagger$

Dagger all by itself

The first thing to be done is to spell out the defining properties of $\dagger r$:

(5a) $\quad [r \Rightarrow x] \wedge [x;x \Rightarrow x] \Rightarrow [\dagger r \Rightarrow x] \qquad (\forall x)$

- $\dagger r$'s extremity -

(5b) $\quad [\, r \vee \dagger r ; \dagger r \Rightarrow \dagger r \,]$

Remark By the theorem of Knaster & Tarski we can strengthen implication (5b) into an

equivalence, but we refrain from doing so because we can travel a long long way without having to appeal to Knaster & Tarski.
(End of Remark.)

Next we do justice to the name that has been attached to tr, viz. that tr is transitive and that t is a closure.

The transitivity of tr, i.e.

(6) $[\ tr\ ;\ tr \Rightarrow tr\ ]$.

immediately follows from (5b).

For t to be a closure we have to prove that it is weakening, monotonic, and idempotent.

It being weakening, i.e.

(7) $[\ r \Rightarrow tr\ ]$,

immediately follows from (5b).

It being monotonic, i.e.

(8) $[\ r \Rightarrow s\ ] \Rightarrow [\ tr \Rightarrow ts\ ]$,

follows from the fact that (3) – the defining equation for tr – has the shape
$x: [\ f.r.x \Rightarrow x\ ]$, with f monotonic in r.

Remark Here we have appealed to the very general theorem that for f monotonic in its first argument and for g.r the strongest solution of $x: [\ f.r.x \Rightarrow x\ ]$, g is monotonic as well.
(End of Remark.)

The idempotence of t, i.e.

(9) $[\ \dagger\dagger r \equiv \dagger r\ ]$,

is an immediate consequence of Lemma (10), with $s := \dagger r$:

(10) for transitive s , $[\ \dagger s \equiv s\ ]$

Proof of (10)

Pong: $[\ \dagger s \Leftarrow s\ ]$ is okay since $\dagger$ is weakening

Ping: $[\ \dagger s \Rightarrow s\ ]$

$\Leftarrow$ { $\dagger s$'s extremity, see (5a) }

$[\ s \Rightarrow s\ ] \wedge [\ s ; s \Rightarrow s\ ]$

$\equiv$ { pred. calc. }

$[\ s ; s \Rightarrow s\ ]$

$\equiv$ { s is transitive }

true .

(End of Proof)

This concludes our demonstration that $\dagger$ is a closure.

$\dagger \quad \dagger \quad \dagger$

There are a number of special relations within our calculus, to wit the constants true, false, and J, and the left-, right-, and middleconditions – the latter also called monotypes –. They are all transitive, and therefore – by lemma (10) – we conclude

(11) $[\ \dagger \mathrm{true} \equiv \mathrm{true}\ ]$ $[\ \dagger \mathrm{false} \equiv \mathrm{false}\ ]$

(12) $[\ \dagger J \equiv J\ ]$

(13) for r a left- or right-condition, $[\ \dagger r \equiv r\ ]$

(14) $[\, r \Rightarrow J \,] \;\Rightarrow\; [\, \dagger r \equiv r \,]$

In what follows, we will not have much use for (11) and (13), but we do for (12) and (14)

Remark For the proofs of the transitivity of these special relations, the only properties of $;$ (semi) that we need are that false is a zero-element, J is the identity-element of $;$, and that $;$ is monotonic in either argument. In particular we want to point out that the only junctivity property of $;$ used so far, is its monotonicity.
(End of Remark.)

$\dagger \quad \dagger \quad \dagger$

Next, we investigate how $\dagger$ behaves towards disjunctions. Well, it does not behave well, except when middle-conditions are involved:

(15) $[\, r \Rightarrow J \,] \;\Rightarrow\; [\; \dagger(r \vee s) \;\equiv\; \dagger r \;\vee\; \dagger s \;]$

Proof $[\mathrm{LHS} \Leftarrow \mathrm{RHS}]$ is okay by monotonicity of $\dagger$.
$[\mathrm{LHS} \Rightarrow \mathrm{RHS}]$ follows from

$$
\begin{array}{ll}
 & [\, \dagger(r \vee s) \;\Rightarrow\; \dagger r \;\vee\; \dagger s \,] \\
\Leftarrow & \quad \{ \text{ extremity of } \dagger(r \vee s) \text{, see (5a) } \} \\
 & [\, r \vee s \;\Rightarrow\; \dagger r \;\vee\; \dagger s \,] \\
 & \qquad \wedge\; [\, (\dagger r \vee \dagger s) \,;\, (\dagger r \vee \dagger s) \;\Rightarrow\; \dagger r \vee \dagger s \,] \\
\equiv & \quad \{ \dagger \text{ is weakening, on the first conjunct} \\
 & \qquad ; \text{ over } \vee \text{ on the second conjunct} \} \\
 & [\, \dagger r ; \dagger r \;\vee\; \dagger r ; \dagger s \;\vee\; \dagger s ; \dagger r \;\vee\; \dagger s ; \dagger s \\
 & \qquad \Rightarrow\; \dagger r \vee \dagger s \,] \\
\equiv & \quad \{\, [\dagger r ; \dagger r \Rightarrow \dagger r] \text{ and } [\dagger s ; \dagger s \Rightarrow \dagger s] \,\}
\end{array}
$$

$$
\begin{array}{ll}
 & [\, tr ; ts \;\vee\; ts ; tr \;\Rightarrow\; tr \vee ts \,] \\
\Leftarrow & \quad \{\, [r \Rightarrow J] \text{ from the antecedent of (15), hence } [tr \Rightarrow J] \text{ from (14)} \,\} \\
 & [\, J ; ts \;\vee\; ts ; J \;\Rightarrow\; tr \vee ts \,] \\
\equiv & \quad \{\, \text{rel. calc.} \,\} \\
 & [\, ts \;\Rightarrow\; tr \vee ts \,] \\
\equiv & \quad \{\, \text{pred. calc.} \,\} \\
 & \text{true} \,.
\end{array}
$$

(End of Proof.)

By (15) with $r, s := J, r$, and by (12), we have as a corollary

(16) $\quad [\, t(J \vee r) \equiv J \vee tr \,]$,

which will be used shortly. Also we shall use the "lifted" version of (16), which is

(17) $\quad t \circ (J \vee) = (J \vee) \circ t$,

and which expresses quite clearly that functions $(J \vee)$ and t commute.

Remark We want to point out that up to this point the only junctivity property of $;$ that we have used is its finite disjunctivity. (End of Remark.)

So much for t all by itself.

* * *

Star is coming up

Now, following Dijkstra's proposal, we let $*$ see the light by defining $*r$ by

(18a) $[\ *r \equiv \dagger(J \vee r)\ ]$.

Thanks to (16) we now also have

(18b) $[\ *r \equiv J \vee \dagger r\ ]$,

and in exploring properties of $*$ we now can freely choose definition (18a) or (18b), whichever comes in most handy. We also continue our flirt with lifting, and we lift (18) towards

(19a) $* = \dagger \circ (J \vee)$

(19b) $* = (J \vee) \circ \dagger$.

For doing justice to the usual name attached to $*$, we have to prove that $*r$ is reflexive and transitive, and that $*$ is a closure

The reflexivity of $*r$, i.e.

(20) $[\ J \Rightarrow *r\ ]$,

immediately follows from (18b).

The transitivity of $*r$, i.e.

(21) $[\ *r \,;\, *r \Rightarrow *r\ ]$,

is an immediate consequence of the transitivity of $\dagger$, thus:

$$
\begin{array}{ll}
 & *r \,;\, *r \\
\equiv & \quad \{\ (18a)\ \} \\
 & \dagger(J \vee r) \,;\, \dagger(J \vee r) \\
\Rightarrow & \quad \{\ \text{transitivity of } \dagger(J \vee r)\ \} \\
 & \dagger(J \vee r) \\
\equiv & \quad \{\ (18a)\ \} \\
 & *r \qquad .
\end{array}
$$

For $*$ to be a closure, we have to demonstrate that it is weakening, monotonic, and idempotent. For this purpose we shall use one of the lifted formulae (19).

Because $\dagger$ and $(\mathrm{J}\vee)$ are both weakening and monotonic, so is their functional composition, and, hence, so is $*$.

Because $\dagger$ and $(\mathrm{J}\vee)$ are each idempotent, and because they commute – see (17) –, their functional composition is idempotent as well:

$$
\begin{array}{ll}
 & *\circ* \\
= & \quad \{ (19a) \} \\
 & \dagger\circ(\mathrm{J}\vee)\circ\dagger\circ(\mathrm{J}\vee) \\
= & \quad \{ (17) \text{ on the two middle terms} \} \\
 & \dagger\circ\dagger\circ(\mathrm{J}\vee)\circ(\mathrm{J}\vee) \\
= & \quad \{ \text{idempotence of } \dagger \text{ and } (\mathrm{J}\vee) \} \\
 & \dagger\circ(\mathrm{J}\vee) \\
= & \quad \{ (19a) \} \\
 & * \quad .
\end{array}
$$

In summary.

(22) $\quad [\, r \Rightarrow *r \,]$

(23) $\quad [\, r \Rightarrow s \,] \Rightarrow [\, *r \Rightarrow *s \,]$

(24) $\quad [\, **r \equiv *r \,] \qquad \text{or} \qquad *\circ* = *$

$*$ $\quad$ $*$
$\quad *$

In retrospection, the idempotence of $*$ can

be dealt with in very much the same way we dealt with the idempotence of $\dagger$, viz. it is a direct consequence of Lemma (25):

(25) for reflexive and transitive s, $[{*}s \equiv s]$

Proof of (25)

$$
\begin{array}{ll}
 & {*}s \\
\equiv & \quad \{ \text{(10b)} \} \\
 & J \vee ts \\
\equiv & \quad \{ s \text{ is transitive, hence } [ts \equiv s] \} \\
 & J \vee s \\
\equiv & \quad \{ s \text{ is reflexive, i.e. } [J \Rightarrow s] \} \\
 & s
\end{array}
$$

(End of Proof.)

Finally, we mention without proof

(26) $[{*}\text{false} \equiv J]$ $\qquad$ $[{*}\text{true} \equiv \text{true}]$

(27) $[{*}J \equiv J]$

(28) $[r \Rrightarrow J] \Rightarrow [{*}r \equiv J]$

$$
\begin{array}{ccc}
 & * & \\
* & & * \\
 & * &
\end{array}
$$

The transitivity of ${*}r$ — see (21) — is subsumed in the stronger

(29) $[{*}r \,;\, {*}r \equiv {*}r]$.

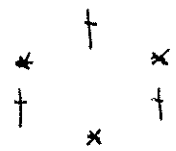
Proof

$$
\begin{array}{ll}
 & {*}r \,;\, {*}r \\
\equiv & \quad \{ \text{(18b)} \} \\
 & (J \vee tr) \,;\, (J \vee tr) \\
\equiv & \quad \{ \,;\, \text{over } \vee \} \{ \text{rel. calc.} \}
\end{array}
$$

$$
\begin{array}{ll}
 & J \vee tr \vee tr ; tr \\
\equiv & \quad \{ \text{pred. calc. using } [\, tr ; tr \Rightarrow tr \,] \} \\
 & J \vee tr \\
\equiv & \quad \{ (18b) \} \\
 & *r
\end{array}
$$

(End of Proof.)

Remark Still, we have not used more of ; than its finite disjunctivity
(End of Remark.)



Dagger and Star together

Here we examine how † and * act among each other. We find

(30a) $[\, \dagger * r \equiv * r \,]$ or $\dagger \circ * = *$

(30b) $[\, * \dagger r \equiv * r \,]$ or $* \circ \dagger = *$

Proofs

$$
\begin{array}{ll}
 & \dagger \circ * \\
= & \quad \{ (19a) \} \\
 & \dagger \circ \dagger \circ (J \vee) \\
= & \quad \{ \dagger \text{ is idempotent} \} \\
 & \dagger \circ (J \vee) \\
= & \quad \{ (19a) \} \\
 & *
\end{array}
\qquad
\begin{array}{ll}
 & * \circ \dagger \\
= & \quad \{ (19b) \} \\
 & (J \vee) \circ \dagger \circ \dagger \\
= & \quad \{ \dagger \text{ is idempotent} \} \\
 & (J \vee) \circ \dagger \\
= & \quad \{ (19b) \} \\
 & *
\end{array}
$$

(End of Proofs.)

Next, in combination with ; we find

(31a) $[\ *r \,;\, tr \ \equiv \ tr\ ]$

(31b) $[\ tr \,;\, *r \ \equiv \ tr\ ]$

Proof of (31a)

$$
\begin{array}{ll}
 & *r \,;\, tr \\
\equiv & \quad \{\ (18b)\ \} \\
 & (J \vee tr) \,;\, tr \\
\equiv & \quad \{\ ;\ \text{over}\ \vee\ \}\{\ \text{rel. calc.}\ \} \\
 & tr \ \vee \ tr \,;\, tr \\
\equiv & \quad \{\ tr\ \text{is transitive}\ \} \\
 & tr
\end{array}
$$

(End of Proof.)

Sad Intermezzo

The above was written "aus einem Guß" one day early January 1994. It is early October now. The reason for that enormous delay is that fun is over now. In what follows, we wish to relate the symmetric and the skew equations for dagger and star, but the ensuing proofs are far from nice or crisp.

A very beginning of lattice theory

Let's start at the very beginning.
A very good place to start.

Julie Andrews in *The Sound of Music*

For a large part, mathematics consists of exploring concepts and of investigating and proving their properties. The art of proving plays a major rôle in this game. Since the advent of modern computing science, it has become clear that in many branches of elementary mathematics, proofs can be beneficially rendered in a calculational format. The benefits comprise greater precision and lucidity —without loss of concision—, an enhanced view on how to separate one's concerns, and hence an improved economy of thought. Unfortunately, most textbooks on elementary mathematical issues have not (yet) adopted such a calculational style, so that yet another generation of young people will receive a mathematical education without having experienced the joy and usefulness of calculating. And this is a pity.

The purpose of this note is to transmit some of the flavour of calculation. We have selected a topic from the very beginning of lattice theory and we intend to present a treatment that can be read, understood, and hopefully enjoyed by a reasonable university freshman.

* * *

Our universe of discourse will be some fixed, anonymous set of things on which a binary relation $\leq$ (" at most") is defined. This relation we postulate to be

- reflexive, i.e. $x \leq x$ $(\forall x)$
- antisymmetric, i.e. $x \leq y \;\wedge\; y \leq x \;\Rightarrow\; x = y$ $(\forall x, y)$

Remark In the standard literature we usually find the additional postulate that $\leq$ is

transitive, i.e. $x \leq y \;\wedge\; y \leq z \;\Rightarrow\; x \leq z$ $(\forall x, y, z)$

For the time being though, we do not need the transitivity of $\leq$. Therefore, we do not introduce it now. And apart from that, it will — as we

shall see — enter the picture in a totally different way.
End Remark .

Equality of things is a very important concept to have. It is as important as the notion of a function. Equality and functions are at the heart of mathematics, and they are beautifully related by the

Rule of Leibniz

For any function f[1], $x = y \Rightarrow f.x = f.y$.

End

The two postulates that we have of $\leq$ do not reveal very much about equality; only the antisymmetry mentions it. Therefore, the first thing to do is to collect some more facts concerning equality. The most common one is the

Rule of Mutual Inequality

$$x = y \equiv x \leq y \wedge y \leq x \quad .$$

End

It is an immediate restatement of $\leq$'s reflexivity and antisymmetry.

A very useful but less common statement about equality is the so-called

Rule of Indirect Equality

$$x = y \equiv \langle\, \forall z :: z \leq x \equiv z \leq y \,\rangle \quad .$$

End

Let us prove it. We prove it by mutual implication. (In our jargon we refer to the implication $LHS \Rightarrow RHS$ from left to right by "ping" and to $RHS \Rightarrow LHS$ by "pong".)

Proof of ping

$$\begin{array}{ll} & x = y \Rightarrow \langle\, \forall z :: z \leq x \equiv z \leq y \,\rangle \\ \equiv & \quad \{\ (P\Rightarrow) \text{ distributes over } \forall\ \} \\ & \langle\, \forall z :: x = y \Rightarrow (z \leq x \equiv z \leq y) \,\rangle \\ \equiv & \quad \{\ \text{Rule of Leibniz, see below}\ \} \\ & \mathit{true} \end{array}$$

The function f involved in this application is the boolean function given by $f.a \equiv z \leq a$.

[1] We denote function application by an infix dot.

End

Proof of pong We have to prove

$$\langle\ \forall z ::\ z \leq x \equiv z \leq y\ \rangle\ \Rightarrow\ x = y\ \ ,$$

and we do so by setting up a weakening chain of predicates that begins with the antecedent and ends with the consequent. Notice that in this chain we will quite likely have to refer to the antisymmetry of $\leq$ because this is the only property of $\leq$ that mentions the $=$-symbol; and we have not used it in the ping-part yet. (The latter remark is a very simple example of the kind of bookkeeping that has proven to be very useful in proof design.) Here is the chain

$$\begin{array}{ll} & \langle\ \forall z ::\ z \leq x\ \equiv\ z \leq y\ \rangle \\ \Rightarrow & \qquad \{\ \text{instantiate with}\ \ z := x\ \ \text{and with}\ \ z := y\ \ \} \\ & (x \leq x\ \equiv\ x \leq y)\ \ \wedge\ \ (y \leq x\ \equiv\ y \leq y) \\ \equiv & \qquad \{\ \text{reflexivity of}\ \ \leq\ \ \} \\ & x \leq y\ \ \wedge\ \ y \leq x \\ \Rightarrow & \qquad \{\ \text{antisymmetry of}\ \ \leq\ \ \} \\ & x = y \end{array}$$

Notice that the first step —the instantiation— is not brilliant at all: the first line contains symbol $\forall$ and the target line does not, so that somewhere along the way we must eliminate $\forall$. In fact, about the only rule from the predicate calculus with which one can eliminate the universal quantifier $\forall$, is the rule of instantiation. Once we are aware of this, the step is no longer a surprise. Furthermore, there is not much we can instantiate z with, viz. just x and y ; and we did both in order to make the next line as strong as possible, which is beneficial if one has to construct a weakening chain.
End Proof of pong .

The rule of Indirect Equality has a companion, also called the Rule of Indirect Equality. It reads

$$x = y\ \equiv\ \langle\ \forall z ::\ x \leq z\ \equiv\ y \leq z\ \rangle\ \ .$$

The difference is the side of the $\leq$-symbol at which x and y reside. Which of the two is to be used depends on the particular application.

* *
*

So much for $\leq$ and for $=$ in our universe. We now enter lattice theory

by postulating that in our universe the equation in p

$$p : \langle\, \forall z ::\ p \leq z \ \equiv\ x \leq z \ \wedge\ y \leq z \,\rangle$$

has, for each x and y, at least one solution. (The inexperienced reader should not feel daunted here: in case our universe is just the universe of real numbers with the usual $\leq$-relation, the maximum of x and y may be recognized as a good candidate for p.)

The first thing we do is to show that the equation has at most one solution. This is done by showing $p = q$, for p and q solutions of the equation. Here, one of the rules of Indirect Equality comes in handy: for any z, we have

$$\begin{array}{ll} & p \leq z \\ \equiv & \qquad \{\ p \text{ is a solution } \} \\ & x \leq z \ \wedge\ y \leq z \\ \equiv & \qquad \{\ q \text{ is a solution } \} \\ & q \leq z, \end{array}$$

and, hence, $p = q$. So our equation has exactly one solution for each x and y. Therefore, that solution is a function of x and y, which we propose to denote by $x \uparrow y$ (x "up" y). In summary, we have the beautiful

$$(0) \qquad x \uparrow y \leq z \ \equiv\ x \leq z \ \wedge\ y \leq z \qquad (\forall x, y, z)$$

(In the standard literature we find $\uparrow$ under entries like "sup" or "join" or "lub".)

Examples

- A well-known instance of (0) can be found in set theory. If we take set inclusion $\subseteq$ as an instance of $\leq$ —it is reflexive and antisymmetric!— , set union $\cup$ is the corresponding $\uparrow$. Indeed, we have for all sets x, y, and z,

$$x \cup y \subseteq z \ \equiv\ x \subseteq z \ \wedge\ y \subseteq z \ .$$

- Also, if we take set containment $\supseteq$ for $\leq$, set intersection is the corresponding $\uparrow$. Indeed,

$$x \cap y \supseteq z \ \equiv\ x \supseteq z \ \wedge\ y \supseteq z \ .$$

- Another well-known instance is in the predicate calculus where we have

$$[\, x \vee y \Rightarrow z \,] \ \equiv\ [\, x \Rightarrow z \,] \wedge [\, y \Rightarrow z \,] \ , \quad \text{and}$$

$$[\, x \wedge y \Leftarrow z \,] \;\equiv\; [\, x \Leftarrow z \,] \wedge [\, y \Leftarrow z \,] \;.$$

- From number theory we know the reflexive, antisymmetric relation denoted $|$ ("divides"). Now, the least common multiple of x and y can see the light via

  $$(x \;\mathsf{lcm}\; y) \,|\, z \;\equiv\; x \,|\, z \;\wedge\; y \,|\, z \quad ,$$

  and the greatest common divisor of x and y by

  $$z \,|\, (x \;\mathsf{gcd}\; y) \;\equiv\; z \,|\, x \wedge z \,|\, y \quad .$$

  Both are instances of (0). (How?)

- But probably the best-known instance of (0) is when we take for $\leq$ the usual order between numbers. Then $\uparrow$ is the familiar maximum operator. We will return to this later.

End Examples .

Now let us investigate (0). We can rather straightforwardly deduce from it that

- $\uparrow$ is idempotent, i.e. $\quad x \uparrow x \;=\; x$
- $\uparrow$ is symmetric, i.e. $\quad x \uparrow y \;=\; y \uparrow x$
- $\uparrow$ is associative, i.e. $\quad x \uparrow (y \uparrow z) \;=\; (x \uparrow y) \uparrow z \quad .$

Let us prove the symmetry. We appeal to Indirect Equality :

$$\begin{array}{ll} & x \uparrow y \leq z \\ \equiv & \qquad \{\ (0)\ \} \\ & x \leq z \;\wedge\; y \leq z \\ \equiv & \qquad \{\ \wedge \text{ is symmetric}\ \} \\ & y \leq z \;\wedge\; x \leq z \\ \equiv & \qquad \{\ (0) \text{ with } x, y := y, x\ \} \\ & y \uparrow x \;\leq\; z, \end{array}$$

and the conclusion follows. From this proof we see that $\uparrow$ inherits its symmetry from $\wedge$. The same holds for $\uparrow$'s idempotence and $\uparrow$'s associativity, as the reader may verify.

The next thing we do with (0) is to study it for some simple instantiations. For instantiation $z := y$ we find

$$\begin{array}{ll} & x \uparrow y \;\leq\; y \\ \equiv & \qquad \{\ (0)\ \} \\ & x \leq y \;\;\wedge\;\; y \leq y \end{array}$$

$\equiv$ $\{$ $\leq$ is reflexive $\}$

$x \leq y$.

Thus we have derived the

Rule of Absorption

$x \uparrow y \leq y \;\equiv\; x \leq y$

End

Next, from (0) with $z := x \uparrow y$, we find the

Rule of Expansion

$y \leq x \uparrow y$

End

Using Mutual Inequality, we can combine the rules of Absorption and Expansion into

(1) $\qquad x \uparrow y = y \;\equiv\; x \leq y \qquad (\forall x, y)$

Remark Almost every established treatment of lattice theory starts from (1), but that is not nearly as nice as the treatment given here, because the pleasing symmetry exhibited by (0) is completely hidden.

End

So much for some simple instantiations of (0).

* *

*

Now the time has come to prove the beautiful[2]

Theorem 1 For reflexive and antisymmetric $\leq$, and for $\uparrow$ as defined by (0), we have that

$\leq$ is transitive .

Proof We have to prove that for all x , y , and z

$x \leq y \;\wedge\; y \leq z \;\Rightarrow\; x \leq z$.

Using (1), this we can rewrite as

$x \uparrow y = y \;\wedge\; y \uparrow z = z \;\Rightarrow\; x \uparrow z = z$,

and we shall prove this latter by showing the consequent — $x \uparrow z = z$ — thereby using the antecedent — $x \uparrow y = x \;\wedge\; y \uparrow z = z$ — :

[2]We owe this theorem to Edsger W. Dijkstra. It seems to be not generally known to lattice theorists.

$$
\begin{array}{ll}
 & x \uparrow z \\
= & \quad \{ \text{ since } y \uparrow z = z \text{ , from the antecedent } \} \\
 & x \uparrow (y \uparrow z) \\
= & \quad \{ \uparrow \text{ is associative } \} \\
 & (x \uparrow y) \uparrow z \\
= & \quad \{ \text{ since } x \uparrow y = y \text{ , from the antecedent } \} \\
 & y \uparrow z \\
= & \quad \{ \text{ since } y \uparrow z = z \text{ , from the antecedent } \} \\
 & z \ .
\end{array}
$$

And we are done. (We ask the reader to notice that each individual step in the above calculation is almost forced upon us. This is a very typical characteristic of many a calculation.)
□

Now that we have obtained the transitivity of $\leq$, we shall feel free to use it. For the sake of completeness we mention that a reflexive, antisymmetric, and transitive relation is commonly called a *partial order*, and that a universe equipped with a partial order is called a *partially ordered set* —a "poset" for short—.

* *
*

Definition (0) of $\uparrow$ tells us when $x \uparrow y \leq z$. We now may ask when $z \leq x \uparrow y$. We leave to the reader to verify that

(2) $\quad z \leq x \uparrow y \ \Leftarrow \ z \leq x \ \vee \ z \leq y$,

and we investigate the converse :

$$
\begin{array}{ll}
 & z \leq x \uparrow y \ \Rightarrow \ z \leq x \ \vee \ z \leq y \\
\equiv & \quad \{ \text{ predicate calculus } \} \\
 & (z \leq x \uparrow y \ \Rightarrow \ z \leq x) \ \vee \ (z \leq x \uparrow y \ \Rightarrow \ z \leq y) \\
\Leftarrow & \quad \{ \leq \text{ is transitive } \} \\
 & x \uparrow y \leq x \ \vee \ x \uparrow y \leq y \\
\equiv & \quad \{ \text{ Rule of Absorption, twice } \} \\
 & y \leq x \ \vee \ x \leq y \ .
\end{array}
$$

For this last line to be valid for any x and y , we require that $\leq$ be a so-called linear or total order: by definition a total order is a partial order

with the additional property that, for all x and y, $x \leq y \ \vee \ y \leq x$.
So, in combination with (2) we find

(3) for $\leq$ a total order,
$z \leq x \uparrow y \ \equiv \ z \leq x \ \vee \ z \leq y$.

Furthermore, we deduce from (1) that

(4) for $\leq$ a total order, operator $\uparrow$ as defined by (0) satisfies
$x \uparrow y = x \ \vee \ x \uparrow y = y$.
(In words : $\uparrow$ is a selector.)

* *
*

From here, we can proceed in many different directions. After all, lattice theory is a huge mathematical terrain, with many ins and outs. We conclude this introduction by confronting $\uparrow$ with other functions.

We consider functions from and to our anonymous universe. For f such a function we have, by definition,

- f is monotonic $\equiv \ \langle \ \forall x, y :: \ x \leq y \ \Rightarrow \ f.x \leq f.y \ \rangle$.
- f distributes over $\uparrow \ \equiv \ \langle \ \forall x, y :: \ f.(x \uparrow y) \ = \ f.x \uparrow f.y \ \rangle$.

We can now formulate the well-known, yet beautiful, theorem

(5) f distributes over $\uparrow \ \Rightarrow \ f$ is monotonic .

Proof For any x and y, we observe

$f.x \leq f.y$
$\equiv$ { (1) }
$f.x \uparrow f.y \ = \ f.y$
$\equiv$ { f distributes over $\uparrow$ }
$f.(x \uparrow y) = f.y$
$\Leftarrow$ { Leibniz's Rule }
$x \uparrow y = y$
$\equiv$ { (1) }
$x \leq y$,

and the result follows from the outer two lines.
End

Small Intermezzo (on proof design)

We would like to draw the reader's attention to the fact that the above proof —no matter how simple it is— displays a great economy of thought. Let us analyze it in some detail. Given that f distributes over $\uparrow$, we have to construct a calculation of the form

$$f.x \leq f.y \ \ldots \Leftarrow \ldots \ x \leq y \ .$$

Right at the outset we can argue that such a calculation will require at least four steps, viz.

- a step to introduce symbol $\uparrow$, in order to be able to exploit the given about f
- a step in which the given about f is actually used
- a step to eliminate symbol $\uparrow$ again, because the target line $x \leq y$ does not mention it
- a step to eliminate symbol f, for which Leibniz's Rule is our only means so far.

Our proof contains precisely (these) four steps, so it cannot be shortened. In fact, it was designed with these four considerations in mind. When we wrote above "no matter how simple it is", this may have sounded paradoxical, but it isn't. On the contrary, the proof derives its simplicity from the consciously considered shapes of the formulae and from the manipulative possibilities available. Nowadays, many more proofs can be and are being designed following such a procedure.
End Small Intermezzo .

A direct consequence of (5) concerns monotonicity properties of $\uparrow$. Because function f defined by $f.x = c \uparrow x$, for whatever c, distributes over $\uparrow$ —as the reader may verify—, theorem (5) tells us that $\uparrow$ is monotonic in its second argument. Since $\uparrow$ is symmetric, we therefore have

(6) $\uparrow$ is monotonic in both arguments.

What about the converse of (5)? Does it hold as well? In order to find out, we try to prove

$$f.(x \uparrow y) \ = \ f.x \uparrow f.y$$

on the assumption that f is monotonic. We do this by Mutual Inequality:

$$\begin{array}{ll} & f.x \uparrow f.y \ \leq \ f.(x \uparrow y) \\ \equiv & \qquad \{ \text{ definition of } \uparrow \text{, see (0) } \} \\ & f.x \leq f.(x \uparrow y) \ \ \wedge \ \ f.y \leq f.(x \uparrow y) \end{array}$$

$\Leftarrow$ { monotonicity of f , twice }

$x \leq x \uparrow y \ \wedge \ y \leq x \uparrow y$

$\equiv$ { Rule of Expansion, twice }

true ,

$f.(x \uparrow y) \ \leq \ f.x \uparrow f.y$

$\Leftarrow$ { (2) }

$f.(x \uparrow y) \leq f.x \ \vee \ f.(x \uparrow y) \leq f.y$

$\Leftarrow$ { monotonicity of f , twice }

$x \uparrow y \leq x \ \vee \ x \uparrow y \leq y$

$\equiv$ { Rule of Absorption, twice }

$y \leq x \ \vee \ x \leq y$,

and the validity of this last line requires $\leq$ to be total. Thus, we have derived, in combination with (5),

(7) for $\leq$ a total order,,
f is monotonic $\equiv$ f distributes over $\uparrow$.

* *
*

In lattice theory, one always introduces a companion to $\uparrow$; it is $\downarrow$ ("down"). (In the standard literature we find $\downarrow$ under entries like "inf", or "meet", or "glb".) It sees the light via

(8) $z \leq x \downarrow y \ \equiv \ z \leq x \ \wedge \ z \leq y$ $(\forall x, y, z)$,

i.e. in a way that is very similar to (0). It has very similar —dual— properties to $\uparrow$. In fact, it has the same properties if we simply flip $\leq$ into $\geq$, and $\uparrow$ into $\downarrow$: just compare (0) and (8). With this symbol dynamics in mind the companion properties for $\downarrow$ come for free. We mention

- $\downarrow$ is idempotent, symmetric, and associative
- $y \leq x \downarrow y \ \equiv \ y \leq x$ Absorption
- $x \downarrow y \ \leq \ y$ Contraction (= the dual of Expansion)
- $x \downarrow y = y \ \equiv \ y \leq x$
- $x \downarrow y \ \leq \ z \ \Leftarrow \ x \leq z \ \vee \ y \leq x$
- $\downarrow$ is monotonic in both arguments
- etcetera .

Of course, we can now also investigate formulae containing both $\uparrow$ and $\downarrow$. We mention

$$x \downarrow (x \uparrow y) = x \quad , \quad x \uparrow (x \downarrow y) = x \quad , \quad \text{and}$$

$$x \downarrow y = x \;\equiv\; x \uparrow y = y \quad .$$

The proofs are left as exercises. We will not continue these investigations now.

In case we take for $\leq$ the usual order between real numbers, $\downarrow$ is the familiar minimum operator.

* *
*

Let us, to conclude this story, consider the real numbers with the usual order $\leq$. This is a total order. The foregoing little theory now grants us quite a number of useful arithmetical results.

- In order to find out which part of the $(x\,,y)$-plane satisfies $x \uparrow y \leq x+y$, we simply calculate :

$$\begin{array}{ll} & x \uparrow y \leq x+y \\ \equiv & \qquad \{ \text{ definition of } f \ \} \\ & x \leq x+y \;\wedge\; y \leq x+y \\ \equiv & \qquad \{ \text{ arithmetic } \} \\ & 0 \leq y \;\wedge\; 0 \leq x \quad . \end{array}$$

  So the answer is: the first quadrant.
  (Ask one of your colleagues or students to solve this little problem, and observe how he does it. This could be a very instructive experiment.)

- Since function f, defined by $f.x = c+x$, is monotonic, we infer from (7):

  addition distributes over the maximum.

- Likewise, multiplication with a nonnegative number distributes over the maximum.

- And also

$$2^{x \uparrow y} = 2^x \uparrow 2^y \quad , \quad \text{and}$$

$$(x \uparrow y)^2 = x^2 \uparrow y^2 \quad (\text{for } x\,, y \geq 0\,, 0) \quad , \quad \text{and}$$

$$z \downarrow (x \uparrow y) = (z \downarrow x) \uparrow (z \downarrow y) \quad .$$

- And now the reader should prove —with a minimal amount of case analysis—

$$x^2 \downarrow y^2 \ \leq \ x*y \ \Leftarrow \ 0 \leq x*y \quad .$$

- Perhaps, we can also learn to handle absolute values more readily, because we have $\mid x \mid \ = x \uparrow -x$. Try to use it to prove the triangular inequality

$$\mid x+y \mid \ \leq \mid x \mid + \mid y \mid \quad .$$

- Etcetera.

* *
*

This really was the beginning of lattice theory. Was it difficult? We hope that most of our readers will say: no! We believe that elementary lattice theory —which goes beyond this note— can and should be taught to reasonable freshmen or, in any case, to sophomores, of computing science and mathematics alike. Many of our colleagues, world-wide, especially computing science colleagues, will shudder at the thought, because lattice theory is regarded far too abstract to be useful or to be teachable to the average student. And abstract stands for frightening, doesn't it? We really must disagree with such a point of view, because —as we tried to show— the game is completely under control by the use of a modest repertoire of simple calculational rules. It is the peaceful calculational style which does away with the fear for abstract things. And also, it is the peaceful calculational style which lets the subject matter sink in much more profoundly than would have been the case otherwise. If still in doubt, remember Newton and Leibniz: they took away the deep difficulties attending the notions of limits and derivatives by ...proposing a symbolism to denote them and a set of formula rewrite rules to manipulate and ...to master them. By now these notions are high-school topics.

Exercises in Calculating

W.H.J. Feijen & A.J.M van Gasteren

Dear Masters, Colleagues, and Students
at the 1996 Marktoberdorf Summer School,

When Edsger W. Dijkstra indicated that he did not feel up to attending this year's Summer School, one of us (WF) was invited to fill in his slots. Prof. Dijkstra had planned to address "The Design of Calculations", an important issue for modern computing science. Due to shortage of time we were not able to compose a coherent and informative text dealing with this topic. So we decided to fill in the empty slots with some talks on Multiprogramming.

Yet, we felt that, in one way or another, we had to remain faithful to Dijkstra's original proposal, and that is why we decided to concoct a small set of exercises that can serve as a carrier for discussing all sorts of aspects concerning the art of calculating. Our plan is to distribute the exercises few by few —i.e. not all at a time—, and ask you to solve and discuss them during dinner or lunch, or in a lost hour, or even late at night in a pub. We can ask this because the bulk of our

exercises do not require more than, say, 10 calculational steps, and most of them even far less than that.

There is one important thing, though, that you must promise us <u>not</u> to do, namely regard them as exercises in problem solving. Of course you are welcome to feel excited and delighted when you found/designed a calculational solution, but you are not allowed to leave it at that, because it is only then when the real game starts. Because it is only then that you, in discussion with others who tackled the problem, should address questions like

- which have been my design considerations?
- what is the overall structure of my calculation, and could it have been done differently?
- what is the quality of the hints justifying the correctness of the individual steps? (An important quality criterion is that a reader can read (and understand) your calculation at a steady, not too slow pace.)
- are the individual steps too small, too naive, or, on the contrary, too large?
- how is the layout of my calculation on paper? Is the spacing attractive to the human mind? (For instance: "$x^2+y^2 = (x+y)^2 \equiv \{\text{expansion}\}\ x^2+y^2 = x^2+2xy+y^2 \equiv \{\text{cancellation}\}\ 0 = 2xy \equiv \{\text{algebra}\}\ 0=x \vee 0=y$" is not!)
- etcetera

Of course, the people present at this Summer School have lots of different backgrounds. They have different mother tongues, different academic educations, and different ages. And – there is no escaping it – these differences will pop up in the discussions. In dealing with these exercises, however, we must agree on something, and this is

- that we are supposed to be more or less familiar with elementary predicate calculus, for instance as described in [DS90]
- that we stick to the kind of calculational format as used in, for instance, [Gas90], [DS90], and [GS93]

Among the people present at this Summer School, there are a number who are familiar with the intended calculational style, such as David Gries, Rutger M. Dijkstra, Wim Feijen, Netty van Gasteren, Markus Kaltenbach, Burghard von Karger, and definitely some others. Please, don't hesitate to contact them/us.

We conclude this little letter with the warning that some of the exercises are extremely simple, and that some are quite hard. Good luck, and ... enjoy!

References

[DS90] E.W. Dijkstra and C.S. Scholten,
Predicate Calculus and
Program Semantics,
Springer-Verlag, Berlin, 1990

[Gas90] A.J.M. van Gasteren,
On the Shape of
Mathematical Arguments,
Lecture Notes in Computer Science 445,
Springer-Verlag, New York, 1990

[GS93] D. Gries and F.B. Schneider,
A Logical Approach to Discrete Math,
Springer-Verlag, New York, 1993

* * *

Exercise 0

We consider an anonymous universe equipped with a relation $\leq$.

- For reflexive and antisymmetric $\leq$, we have the two so-called "Rules of Indirect Equality", viz. for all x, y :

$$x = y \quad \equiv \quad \langle \forall z :: \; z \leq x \equiv z \leq y \rangle \quad ,$$

$$x = y \quad \equiv \quad \langle \forall z :: \; x \leq z \equiv y \leq z \rangle \quad .$$

Prove one of them. (Hint: give a ping-pong proof, i.e. a proof by mutual implication.)

- For reflexive and transitive $\leq$, we have the two so-called "Rules of Indirect Inequality", viz. for all x, y :

$$x \leq y \quad \equiv \quad \langle \forall z :: \; z \leq y \Leftarrow z \leq x \rangle \quad ,$$

$$x \leq y \quad \equiv \quad \langle \forall z :: \; x \leq z \Leftarrow y \leq z \rangle \quad .$$

Prove one of them.

Exercise 1 (from the very beginning of lattice-calculus)

We consider an anonymous universe equipped with a reflexive, antisymmetric relation $\leq$ and a binary infix operator $\uparrow$ ("up"), related by, for all x, y, z :

$$x \uparrow y \leq z \quad \equiv \quad x \leq z \ \wedge \ y \leq z \ .$$

Prove

a) $y \leq x \uparrow y$

b) $\uparrow$ is associative

c) $x \uparrow y = y \quad \equiv \quad x \leq y$

d) $\leq$ is transitive

e) For each function f to and from our universe,

f distributes over $\uparrow$

$\Rightarrow$ f is monotonic with respect to $\leq$.

f) $z \leq x \uparrow y \quad \Leftarrow \quad z \leq x \ \vee \ z \leq y$

g) For $\leq$ total as well, i.e. $p \leq q \ \vee \ q \leq p$ for all p, q :

$$z \leq x \uparrow y \quad \equiv \quad z \leq x \ \vee \ z \leq y \ .$$

Exercise 2 (from the beginning of lattice-calculus)

We consider a universe with reflexive, antisymmetric, and transitive relation $\leq$. Furthermore, for each predicate R and endo-function t, i.e. a function to and from the universe,

$$\langle \uparrow x : R.x : t.x \rangle$$

is given to be an element of the universe; by definition it satisfies

$$\langle \uparrow x : R.x : t.x \rangle \leq z \quad \equiv \quad \langle \forall x : R.x : t.x \leq z \rangle \ .$$

Prove

a) [Instantiation]

$$R.x \quad \Rightarrow \quad t.x \leq \langle \uparrow y : R.y : t.y \rangle$$

b) [Term monotonicity]

$$\langle \forall x :: s.x \leq t.x \rangle$$
$$\Rightarrow$$
$$\langle \uparrow x :: s.x \rangle \leq \langle \uparrow x :: t.x \rangle \quad , \quad \text{for each range}$$

c) [Range Monotonicity]

$$\langle \forall x :: R.x \Rightarrow S.x \rangle$$
$$\Rightarrow$$
$$\langle \uparrow x : R.x : t.x \rangle \leq \langle \uparrow x : S.x : t.x \rangle$$

d) f is monotonic with respect to $\leq$

$\equiv$

for each x, $f.x = \langle \uparrow y : y \leq x : f.y \rangle$

Similar, dual properties hold for

$\langle \downarrow x : R.x : t.x \rangle$,

defined by

$$z \leq \langle \downarrow x : R.x : t.x \rangle \quad \equiv \quad \langle \forall x : R.x : z \leq t.x \rangle \quad .$$

(Operator $\uparrow$ is commonly called "supremum" and $\downarrow$ "infimum".)

Exercise 3 (from the theory of Galois Connections)

We consider a universe with reflexive, antisymmetric, and transitive relation $\leq$. Furthermore, for each predicate R and endofunction t, supremum $\langle \uparrow x : R.x : t.x \rangle$ and infimum $\langle \downarrow x : R.x : t.x \rangle$ are defined. (See Exercise 2.)

Let f and g be two endofunctions coupled by the so-called Galois Connection: for all x, y

$$(*) \qquad f.x \leq y \quad \equiv \quad x \leq g.y \ .$$

Prove

a) f distributes over arbitrary suprema. (This is commonly called: "f is universally $\uparrow$-junctive".)

b) g distributes over arbitrary infima.

c) [Rules of Cancellation]

$$x \leq g.(f.x)$$

$$f.(g.x) \leq x \ .$$

d)
$$f.x = \langle \downarrow z : x \leq g.z : z \rangle$$

$$g.y = \langle \uparrow z : f.z \leq y : z \rangle$$

e) Monotonic functions f and g that satisfy the Rules of Cancellation are Galois-Connected as in (*) above.

f) For universally $\uparrow$-junctive f, there exists a function g satisfying . (In the jargon, g is called an "upper-adjoint" of f.)

g) Upper (and lower) adjoints are unique .

Exercise 4 (from the beginning of extremity-calculus)

We consider a universe with a reflexive, antisymmetric, and transitive relation $\leq$. Furthermore, all suprema and infima are defined. (See Exercise 2.)

Let $\mathcal{B}$ be a predicate on our universe, and consider equation

(*) $\quad x : \mathcal{B}.x$

(This is our notation for an equation $\mathcal{B}.x$, in which x is the "unknown".)

a) Give a formal characterization of "equation (*) has a least (w.r.t. $\leq$) solution".

b) Prove that equation (*) has a least solution precisely whenever the infimum of all solutions solves (*)

Next, let f be an endofunction on our universe, and consider equation

(**) $\quad x : f.x \leq x$

c) Prove that for monotonic f, equation (**) has a least solution

d) [Theorem of Knaster & Tarski]
Prove that for monotonic f, equations

$x : f.x \leq x$ and

$x : f.x = x$

have the same least solution

e) Let q be such that

$f.q \leq q$ and

$\langle \forall x : f.x = x : q \leq x \rangle$.

Prove that, for monotonic f,
q is the least solution of $x : f.x \leq x$.

Exercise 5 (from relation calculus)

We extend the predicate calculus with a new binary infix operator, denoted ; ("semi"), of which we postulate that

- it is disjunctive (and hence monotonic, cf. Exercise 1e) in each of its arguments
- it has J as its two-sided identity element, i.e. $[x ; J \equiv x]$ and $[J ; x \equiv x]$.

Prove $[x \Rightarrow J] \wedge [y \Rightarrow J] \Rightarrow [x ; y \equiv x \wedge y]$

(This has proven to be a difficult exercise for the uninitiated.)

Exercise 6 (from relation/regularity calculus)

We extend the predicate calculus with a new binary infix operator, denoted ; ("semi"), of which we postulate that

- it is associative
- it has J as its two-sided identity-element, i.e. $[x ; J \equiv x]$ and $[J ; x \equiv x]$
- it is universally disjunctive (and hence monotonic, cf. Exercise 1e) in each of its arguments.

We now consider, for arbitrary r, equation

(*) $\quad x : [J \vee x ; r \Rightarrow x]$

Prove

a) Equation (*) has a strongest solution (i.e. a least solution with respect to $\Rightarrow$).

b) For any s, equation $x : [s ; x \Rightarrow s]$ has a weakest solution.

c) For s the strongest solution of (*), $[s ; s \Rightarrow s]$ ("s is transitive")

d) Equations (*) and $x : [J \vee r ; x \Rightarrow x]$ have the same strongest solution.

Designing a proof for R.S. Bird's theorem on pre-orders.

We are given some fixed, anonymous universe of things. For any binary relation R on that universe and for any subset S on that same universe subset $Min.R.S$ is defined as follows: for any x,

$$x \in Min.R.S$$
$$\equiv$$
$$x \in S \;\wedge\; (\forall y :: y \in S \Rightarrow x\,R\,y) .$$

The theorem to be proven is that for any two pre-orders X and Y, there exists a pre-order Z satisfying

$$Min.Z = Min.Y \circ Min.X ,$$

or -equivalently-, for all S

$$(0) \qquad Min.Z.S = Min.Y.(Min.X.S) .$$

(A pre-order is a binary relation that is both reflexive and transitive.)

* * *

The above was communicated to us by Richard S. Bird as an exercise in the predicate calculus. It was accompanied by the warning that proving the theorem had been "surprisingly difficult". In view of our recent involvement in the relational calculus and in view of the announced difficulty of the exercise. Bird's theorem came -more or less-

as a gift from heaven. because now there was an opportunity to put the relational calculus at work for solving a nontrivial problem. Therefore we took the liberty of changing Bird's exercise into an exercise in the relational calculus, with the purpose of testing the latter's potential.

However, it turned out that, more than anything else, the exercise became an exercise in proof development. In our first effort to prove the theorem we were insufficiently aware of this, and in finding a proof we would proceed so uncautiously that, in the end, we got completely stuck. Warned and alarmed by such a miserable performance, we started afresh, this time obeying all our current rules for proof construction. The ensuing result was a proof of Bird's theorem with almost every step pre-ordained, thus offering no surprises at all. The main purpose of this note has now become to exhibit that development.

* * *

This text will not be self-contained in that it uses the relational calculus. The appeal to that calculus will, however, be quite modest and not go beyond e.g. AvG92/WF140: "An introduction into the relational calculus". For completeness's sake we supply an appendix mentioning the most important calculational rules employed in this note.

* * *

Translating the problem

The problem, as stated, is formulated in terms of subsets of and binary relations on a given universe. Since we wish to tackle the problem using the relational calculus we first translate sets into binary relations.

There are two standard ways of pairing sets and relations. One is to associate sets with relations called monotypes, and the other is to couple them to relations called leftconditions. Because there is a one-to-one correspondence between monotypes and leftconditions, the choice is irrelevant from a mathematical point of view. However, we choose to represent sets by leftconditions because it so happens that with that choice the ensuing formulae become simpler by almost one order of magnitude.

By convention, set S and leftcondition S – no confusion will arise from overloading name S – will be coupled by the rule

$$(\forall x,z :: \quad x \in S \quad \equiv \quad x \; S \; z) \; .$$

(The fact that the binary relation S defined by this rule indeed matches the notion of S being a leftcondition as we know it from the relational calculus, is not demonstrated here.)

Theorem (0) is entirely expressed in expressions of the form $\text{Min.}\, z.S$. For

set & leftcondition S, we therefore seek to translate set $Min.Z.S$ into a relation that is a leftcondition. We propose that

$$Min.Z.S = A$$

where A is the largest set satisfying
- see definition of Min -

$$(\forall x :: x \in A \Rightarrow x \in S \wedge (\forall y :: y \in S \Rightarrow x \not Z y)) ,$$

or - equivalently -

(i) $(\forall x :: x \in A \Rightarrow x \in S)$, and

(ii) $(\forall x :: x \in A \Rightarrow (\forall y :: y \in S \Rightarrow x \not Z y))$.

First we translate (i) into the relational format:

$$\begin{array}{ll} & (i) \\ = & \{ \text{introduction of an additional dummy} \} \\ & (\forall x,z :: x \in A \Rightarrow x \in S) \\ = & \{ \text{using the coupling rule for } S \} \\ & (\forall x,z :: x \in A \Rightarrow x\, S\, z) \\ = & \{ \text{on the premise that also } A \text{ satisfies the coupling rule, i.e. relation } A \text{ is a leftcondition} \} \\ & (\forall x,z :: x\, A\, z \Rightarrow x\, S\, z) \\ = & \{ \text{definition of } [\,] \} \\ & [\, A \Rightarrow S \,] \end{array} .$$

Next we observe

$$\begin{array}{ll} & (ii) \\ = & \{ \text{pred. calc.} \} \\ & (\forall x,y :: x \in A \wedge y \in S \Rightarrow x \not Z y) \end{array}$$

$$
\begin{array}{ll}
= & \{ \text{additional dummy} \} \\
 & (\forall x,y :: (\exists z :: x \in A \wedge y \in S) \Rightarrow x \, Z \, y) \\
= & \{ \text{coupling rule: given for } S \text{ and demanded for } A \} \\
 & (\forall x,y :: (\exists z :: x \, A \, z \wedge y \, S \, z) \Rightarrow x \, Z \, y) \\
= & \{ \text{definition of } \sim \} \\
 & (\forall x,y :: (\exists z :: x \, A \, z \wedge z \, (\sim S) \, y) \Rightarrow x \, Z \, y) \\
= & \{ \text{definition of } ; \} \\
 & (\forall x,y :: x \, (A ; \sim S) \, y \Rightarrow x \, Z \, y) \\
= & \{ \text{definition of } [\,] \} \\
 & [A ; \sim S \Rightarrow Z] \; .
\end{array}
$$

Summarizing, we have that set Min.Z.S translates into the weakest relation A satisfying

(*) $[A \Rightarrow S] \wedge [A ; \sim S \Rightarrow Z]$ and

(**) A is a leftcondition .

At this point we have a stroke of good luck, since the weakest A satisfying just (*) is a leftcondition whenever S is – shown below – so that demand (**) of A is for free. For future use we now completely spell out the fact that A is the weakest relation satisfying (*) :

(a0) $[A \Rightarrow S]$

(a1) $[A ; \sim S \Rightarrow Z]$

(a2) $[W \Rightarrow S] \wedge [W ; \sim S \Rightarrow Z] \Rightarrow [W \Rightarrow A]$ $(\forall W)$

Now we show that (**) follows from properties (a) and from S being a leftcondition.

Proof We show that A is a leftcondition by using the relational definition [A; true ⇒ A] rather than a pointwise definition of leftconditions.

```
    [ A; true  ⇒  A ]
⇐       { (a2) with  W := A; true }
    [ A; true  ⇒  S ]  ∧  [ A; true; ~S  ⇒  Z ]
=       { S is a leftcondition and
          ~S is a rightcondition }
    [ A; true  ⇒  S; true ]  ∧  [ A; ~S  ⇒  Z ]
⇐       { monotonicity of ; }  { (a1) }
    [ A  ⇒  S ]
=       { (a0) }
    true .
```

(End of Proof.)

* * *

Now we are ready to translate our target expression

(0) $\quad Min.Z.S = Min.Y.(Min.X.S)$

With $\quad A = Min.Z.S$
$\quad B = Min.Y.C$
$\quad C = Min.X.S$,

(0) can be rewritten as

$$A = B \ .$$

With our representation of sets by leftconditions, (0) can be rewritten as

$$[A \equiv B] \ ,$$

with as givens that S. A, B, and C are leftconditions satisfying properties (a), (b), and (c) – the latter two to be displayed in a moment – .

Bird's theorem is that for preorders X and Y there exists a preorder Z such that [A ≡ B] holds. For the definition of X being a preorder we will use the relational definitions

[J ⇒ X] for the reflexivity of X,

and [X;X ⇒ X] for the transitivity of X.

× × ×

Before we embark on the construction of a proof we first tabulate all the givens. Our advice to the reader is to physically isolate this table from the rest of the text – for instance by Xeroxing it – and to keep it ready for inspection all through the process of proof construction. Also he should use it to keep track of which givens have been used in the proof "so far", because at some point during the proof construction this record plays an important heuristical rôle.

Here is the table.

The table of givens

(a0) $[A \Rightarrow S]$

(a1) $[A ; \sim S \Rightarrow Z]$

(a2) $[W \Rightarrow S] \wedge [W ; \sim S \Rightarrow Z] \Rightarrow [W \Rightarrow A]$

(b0) $[B \Rightarrow C]$

(b1) $[B ; \sim C \Rightarrow Y]$

(b2) $[W \Rightarrow C] \wedge [W ; \sim C \Rightarrow Y] \Rightarrow [W \Rightarrow B]$

(c0) $[C \Rightarrow S]$

(c1) $[C ; \sim S \Rightarrow X]$

(c2) $[W \Rightarrow S] \wedge [W ; \sim S \Rightarrow X] \Rightarrow [W \Rightarrow C]$

S, A, B, and C are leftconditions

X and Y are preorders, i.e.

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

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

Constructing a proof

Our proof will be entirely calculational. For just checking the correctness of the calculations the hints suffice. But we want to do more. For a number of crucial steps we wish to explain why we did those steps. Therefore we will annotate our calculations with "reasons" containing heuristical considerations. These heuristical considerations are important to the extent that they may reveal that certain steps that look like rabbits are, upon closer scrutiny, not rabbits at all. In fact, they may reveal that our forthcoming proof is largely pre-ordained.

* * *

We have to prove the existence of a relation Z such that

$$[A \equiv B] \quad \wedge \quad Z \text{ is a preorder} .$$

We do this by constructing at least one witness. Because the second conjunct is too general a requirement, we start focussing on the first one. Because all our givens about A and B occur in implications, we rewrite $[A \equiv B]$ as the conjunction of $[A \Rightarrow B]$ and $[B \Rightarrow A]$, and we tackle these conjuncts separately.

Re $[A \Rightarrow B]$

$$\begin{array}{ll} & [A \Rightarrow B] \\ \Leftarrow & \quad \{ \text{ (b2) with } W := A \ \} \end{array}$$

$$[A \Rightarrow C] \wedge [A ; \sim C \Rightarrow Y]$$

Reason Expression $[A \Rightarrow B]$ has A in the antecedent and B in the consequent. The only rule that can handle a B in the consequent is rule (b2) and the only rule that can handle an A in the antecedent is rule (a0). The reason to reject an application of (a0) is that it would genuinely strengthen $[A \Rightarrow B]$, whereas the application of (b2) is, in fact, an equivalence preserving step: the whole bunch of formulae (b0), (b1), and (b2) can be rewritten as the single and equivalent

$$(*) \quad [W \Rightarrow C] \wedge [W ; \sim C \Rightarrow Y] \equiv [W \Rightarrow B] \qquad (\forall W)$$

- as the reader may verify - . In case of choice, equivalence preserving steps are always to be preferred over other ones, that is to say: there should be very good reasons for neglecting this heuristic rule.

In the meantime, the reader may wonder why in the first place we tabulated the expanded formulae (b) instead of the much more compact (*). The argument for this is that (*) is too entangled. It acts as a container and hampers direct access to the properties (b0) and (b1). The reason why we rendered (b2) as an implication rather than an equivalence is that, without loss of mathematical content, it reduces our manipulative possibilities and thereby the search-space in which our ultimate proof is to be found.
(End of Reason.)

Next we tackle the conjuncts $[A \Rightarrow C]$ and $[A ; \sim C \Rightarrow Y]$ separately.

- $[A \Rightarrow C]$
  $\Leftarrow$ { (c2) with $W := A$, heuristics as before }
  $[A \Rightarrow S] \wedge [A ; \sim S \Rightarrow X]$
  $=$ { (a0) for the first conjunct }
  $[A ; \sim S \Rightarrow X]$
  $\Leftarrow$ { (a1) }
  $[Z \Rightarrow X]$.

Reason The expression $[A ; \sim S \Rightarrow X]$ could also have been strengthened by weakening A via (a0) or by strengthening X via (c1). However, we have to bear in mind that the theorem should hold for all S, and that sooner or later we have to remove all occurrences of S - and, in its wake, all occurrences of A, B, and C - from our demonstrandum. The rules (a1), (b1), and (c1) are the only ones that can do this for us.
(End of Reason.)

Meanwhile we have encountered the first constraint to be imposed on Z, viz.

(1) $[Z \Rightarrow X]$.

- $[A ; \sim C \Rightarrow Y]$
  $=$ { pred. calc. }
  $[A ; \sim C \wedge A ; \sim C \Rightarrow Y]$
  $\Leftarrow$ { (a0), and (c0) in the form $[\sim C \Rightarrow \sim S]$ }
  $[S ; \sim C \wedge A ; \sim S \Rightarrow Y]$

Reason The table of givens provides three possibilities for strengthening $[A ; \sim C \Rightarrow Y]$. One of them strengthens consequent Y via (b1), but this removes Y from the expression which is not to be recommended. The two other possibilities weaken the antecedent by weakening A via (a0) and $\sim C$ via (c0). Which one do we choose? Here we follow a rule that we owe to Edsger W. Dijkstra, and apply both possibilities simultaneously while seeing to it that the resulting expression is as weak as possible It is this latter goal that explains the emergence of the $\wedge$ in the antecedent of the newly formed expression.

At this point it is nice to add that, had we allowed ourselves a coarser strengthening than the one resulting from Dijkstra's rule, the rest of the proof would have come to an unsuccessful end. Here we may have an explanation for Richard Bird's characterization of the problem: "surprisingly difficult".
(End of Reason.)

We continue our calculation:

$$
\begin{array}{ll}
 & [\, S ; \sim C \;\wedge\; A ; \sim S \;\Rightarrow\; Y \,] \\
\Leftarrow & \quad \{ \text{(c1) in the form } [\, S ; \sim C \Rightarrow \sim X \,] \} \\
 & \quad \{ \text{(a1)} \} \\
 & [\, \sim X \;\wedge\; Z \;\Rightarrow\; Y \,] \\
= & \quad \{ \text{pred. calc.} \} \\
 & [\, Z \;\Rightarrow\; \neg \sim X \vee Y \,] ,
\end{array}
$$

and here we encounter our second constraint to be imposed on Z, viz.

(2) $[\, Z \Rightarrow \neg\sim X \vee Y \,]$.

(End of Re $[A \Rightarrow B]$.)

Re $[B \Rightarrow A]$

$[B \Rightarrow A]$
$\Leftarrow$ { (a2) with $W := B$, heuristics as before }
$[B \Rightarrow S] \wedge [B; \sim S \Rightarrow Z]$,

and we tackle the conjuncts separately.

- $[B \Rightarrow S]$
  $\Leftarrow$ { (c0) }
  $[B \Rightarrow C]$
  $\Leftarrow$ { (b0) }
  true .

- $[B; \sim S \Rightarrow Z]$

The reader who has recorded which of the givens (a), (b), and (c) have been used so far, will observe that (b1) is the only one that has not yet been used. It is a fair guess that (b1) has to play a rôle. But it contains Y and our demonstrandum does not! The question is how to drag an occurrence of Y into the picture. Requirement (2) of Z is the only possibility! If we put

(3) $[\, Z \equiv Z' \wedge (\neg\sim X \vee Y) \,]$

we obtain

$$
\begin{array}{ll}
 & [B;\sim S \Rightarrow Z] \\
= & \quad \{(3)\} \\
 & [B;\sim S \Rightarrow Z' \wedge (\neg\sim X \vee Y)] \\
= & \quad \{\text{pred. calc.}\} \\
 & [B;\sim S \Rightarrow Z'] \wedge [B;\sim S \wedge \sim X \Rightarrow Y].
\end{array}
$$

Again we tackle these conjuncts separately.

$$
\begin{array}{ll}
\bullet\bullet & [B;\sim S \Rightarrow Z'] \\
\Leftarrow & \quad \{(b0)\} \\
 & [C;\sim S \Rightarrow Z'] \\
\Leftarrow & \quad \{(c1)\} \\
 & [X \Rightarrow Z'] .
\end{array}
$$

which gives us our third contraint to be imposed on Z, viz.

(4) $\quad [X \Rightarrow Z']$.

$$
\begin{array}{ll}
\bullet\bullet & [B;\sim S \wedge \sim X \Rightarrow Y] \\
\Leftarrow & \quad \{(b1)\} \\
 & [B;\sim S \wedge \sim X \Rightarrow B;\sim C]
\end{array}
$$

Reason We must use (b1) and the above step is the only possibility.
(End of Reason.)

Now we are left with an expression that is entirely formulated in a "B & C - nomenclature". In particular, it no longer contains a reference to A or Z, and none of the rules (b) and (c) allow us to reimport these names into the expression. This means that in showing its validity we will not encounter new constraints

on Z. For us this is the main reason isolate it as a separate

Lemma $[B;{\sim}S \wedge {\sim}X \Rightarrow B;{\sim}C]$,

to be shown later. (Another reason to isolate it that there are so many different proofs for it, all of them relatively long, and none of them really fascinating, i.e. coming close to being ugly.)

(End of Re $[B \Rightarrow A]$.)

× × ×

This concludes our design of a proof for $[A \equiv B]$ and it also concludes the most fascinating and most critical part of our proof for Bird's theorem. Of course, it remains to be shown that our constraints on Z, viz.

(1) $[Z \Rightarrow X]$

(2) $[Z \Rightarrow \neg{\sim}X \vee Y]$

(3) $[Z \equiv Z' \wedge (\neg{\sim}X \vee Y)]$

(4) $[X \Rightarrow Z']$.

admit of a solution that is a preorder. (Constraint (2) is subsumed in (3), but we leave it as is.) It was Lex Bijlsma who at this point observed that the constraints admit precisely 1 solution, viz.

$[Z \equiv X \wedge (\neg{\sim}X \vee Y)]$,

- proof left to the reader - . Now we show that this solution is a preorder.

Z is reflexive . i.e. $[J \Rightarrow Z]$

$$
\begin{array}{ll}
 & Z \\
= & \quad \{(3)\} \\
 & Z' \wedge (\neg\sim X \vee Y) \\
\Leftarrow & \quad \{(4) \text{ and pred. calc.}\} \\
 & X \wedge Y \\
\Leftarrow & \quad \{X \text{ and } Y \text{ are reflexive}\} \\
 & J \quad .
\end{array}
$$

Z is transitive . i.e. $[Z ; Z \Rightarrow Z]$

$$
\begin{array}{ll}
 & [Z ; Z \Rightarrow Z] \\
= & \quad \{(3)\} \\
 & [Z ; Z \Rightarrow Z' \wedge (\neg\sim X \vee Y)] \\
= & \quad \{\text{pred. calc.}\} \\
 & [Z ; Z \Rightarrow Z'] \wedge [Z ; Z \Rightarrow \neg\sim X \vee Y] \quad .
\end{array}
$$

As for the first conjunct we observe.

$$
\begin{array}{ll}
 & Z' \\
\Leftarrow & \quad \{(4)\} \\
 & X \\
\Leftarrow & \quad \{X \text{ is transitive}\} \\
 & X ; X \\
\Leftarrow & \quad \{(1)\} \\
 & Z ; Z \quad .
\end{array}
$$

As for the second conjunct we observe.

$$
\begin{array}{ll}
 & [Z ; Z \Rightarrow \neg\sim X \vee Y] \\
= & \quad \{\text{pred. calc.}\} \\
 & [Z ; Z \wedge \sim X \Rightarrow Y]
\end{array}
$$

$\Leftarrow$ { Y is transitive }
$[Z; Z \wedge \sim X \Rightarrow Y; Y]$
$\Leftarrow$ { the Grand Dedekind, see Appendix for the rule and its "symbol dynamics" }
$[\sim X; \sim Z \wedge Z \Rightarrow Y]$
$\wedge$
$[\sim Z; \sim X \wedge Z \Rightarrow Y]$

For the first of these conjuncts - the second is left to the reader - we have

$[\sim X; \sim Z \wedge Z \Rightarrow Y]$
$\Leftarrow$ { (1) in the form $[\sim Z \Rightarrow \sim X]$ }
$[\sim X; \sim X \wedge Z \Rightarrow Y]$
$\Leftarrow$ { X's transitivity in the form $[\sim X; \sim X \Rightarrow \sim X]$ }
$[\sim X \wedge Z \Rightarrow Y]$
$=$ { pred. calc. }
$[Z \Rightarrow \neg\sim X \vee Y]$
$=$ { (2) }
true,

which concludes our demonstration of Z being transitive.

* * *

Finally, we give a proof of the lemma. It contains - surprisingly and disappointingly - one more appeal to the transitivity of X.

Lemma $[B; \sim S \wedge \sim X \Rightarrow B; \sim C]$

Proof

$[B; \sim S \wedge \sim X \Rightarrow B; \sim C]$

$\Leftarrow$ { the Grand Dedekind-rule }

$[\sim X ; S \wedge B \Rightarrow B]$

$\wedge$

$[\sim B ; \sim X \wedge \sim S \Rightarrow \sim C]$

$=$ { the first conjunct "vanishes". the second is transposed }

$[X ; B \wedge S \Rightarrow C]$

$\Leftarrow$ { (c2) with $W := X ; B \wedge S$ }

$[X ; B \wedge S \Rightarrow S]$

$\wedge$

$[(X ; B \wedge S) ; \sim S \Rightarrow X]$

The first conjunct vanishes. For the second one we observe

$(X ; B \wedge S) ; \sim S$

$\Rightarrow$ { pred. calc. }

$X ; B ; \sim S$

$\Rightarrow$ { (b0) }

$X ; C ; \sim S$

$\Rightarrow$ { (c1) }

$X ; X$

$\Rightarrow$ { X is transitive }

X .

(End of Proof.)

Remark As mentioned before, the above lemma admits of many different proofs. All previous proofs we had lavishly used that S, B, and C are leftconditions, but the above proof - which was developed while being written down - does not use these facts at all. The net result is that our proof of Bird's theorem nowhere uses the givens that S, A, B, and C are leftconditions:

they only enter the picture for the benefit of translating the original problem statement into the relational notation. This surprise at the very end of this note will be food for further thought, and perhaps necessitate a rewrite. Peccavimus.
(End of Remark.)

This concludes our derivation of a proof for Bird's theorem.

* * *

Final Remarks

We wish to describe the flavour of our first effort to prove the theorem in order to understand better why that effort failed. Formulae (a) define A as the weakest relation satisfying

(a0) $[\, A \Rightarrow S \,]$ and

(a1) $[\, A ; \sim S \Rightarrow Z \,]$.

We can give a closed expression for that relation by rewriting (a1) as follows

$$
\begin{array}{ll}
 & [\, A ; \sim S \Rightarrow Z \,] \\
= & \quad \{ \text{left-exchange} \} \\
 & [\, \neg Z ; S \Rightarrow \neg A \,] \\
= & \quad \{ \text{contrapositive} \} \\
 & [\, A \Rightarrow \neg (\neg Z ; S) \,] \; .
\end{array}
$$

The closed expression for A now is

$$[\, A \equiv S \wedge \neg(\neg Z ; S) \,] \quad .$$

Similarly, we have closed expressions for B and C, viz.

$$[\, B \equiv C \wedge \neg(\neg Y ; C) \,]$$

$$[\, C \equiv S \wedge \neg(\neg X ; S) \,] \quad .$$

Theorem $[A \equiv B]$ can now be formulated without any reference to the auxiliary names A, B, or C, namely as: for all S

$$(\ast) \quad [\, S \wedge \neg(\neg Z ; S) \equiv S \wedge \neg(\neg X ; S) \wedge \neg(\neg Y ; (S \wedge \neg(\neg X ; S))) \,] \, ,$$

and our task is to solve it for Z, using the relational calculus as our exclusive tool. As mentioned before, we did not succeed. Presumably, the prime reason for our failure is that expression $(\ast)$ and its manipulative descendants are so long and so fine-grained that they offer far too many manipulative possibilities and far too little guidance for how to continue calculation. Especially in an exercise like the current one, where – in retrospect – the solution space is so tight, an abundance of manipulative freedom at the same time implies an abundance of dead alleys. (In our first effort, the situation was even worse because we had coupled sets to monotypes, rendering $(\ast)$ more complicated by almost one order of magnitude.)

The method that we then applied by attaching auxiliary names to the subexpressions of $(\ast)$ and by completely spelling out their

properties, did the job. It was only then that it became apparent how little freedom we had in proving Bird's theorem. (At a later stage Perry Moerland - student member of the ETAC - successfully repeated the experiment for the monotype representation of sets.)

A viable, quite different way to prove Bird's theorem was exemplified by Henk Doornbos - graduate student to R.C. Backhouse - He introduces a binary operator "/", defined by

$$[\ P/Q \ \equiv \ \neg(\neg P \,;\, Q)\ ],$$

in terms of which (x) can be rewritten as, for all S

$$[\ S \wedge Z/S \ \equiv \ S \wedge X/S \wedge Y/(S \wedge X/S)\ ].$$

With the purpose of solving this for Z, he then develops - in a rather goal directed fashion - a theory of "/". This method is viable especially when - as in Doornbos's case - the interest in operator "/" goes beyond the current exercise.

x x x

A final Final Remark concerns an investigation of Lex Bijlsma's. He has shown, using a pointwise argument, that $X \wedge (\neg\circ X \vee Y)$ is the only candidate relation that can solve Bird's problem. It is nice for us that, with our pointless reasoning, we have encountered

this solution, but at the same time it is a pity that we have not received any signal of its uniqueness.

(End of Final Remarks.)

Acknowledgements

In the first place we would like to acknowledge Richard S. Bird for communicating such a fascinating, easily stated little problem. This text is written for and dedicated to him, mainly.

Secondly, we would like to thank Jaap van der Woude who - after our initial failure - showed us that the theorem can be proven within the relational calculus as we know it.

Finally, we wish to express our gratitude for the many critical remarks of Lex Bijlsma, Ronald Bulterman, Netty van Gasteren, Rob Hoogerwoord, Perry Moerland, John Segers, and Carel Scholten, all members of the ETAC.

(End of Acknowledgements.)

W.H.J. Feijen,

7 November 1991,
Eindhoven.

Appendix

All rules of the relational calculus that we used in this note can be derived from the following set of postulates

- $[\sim x \Rightarrow y] \equiv [x \Rightarrow \sim y]$
- $;$ is associative
- $[x ; y \Rightarrow z] \equiv [\neg z ; \sim y \Rightarrow \neg x]$
  $[x ; y \Rightarrow z] \equiv [\sim x ; \neg z \Rightarrow \neg y]$,
  the left - and right - exchange rules
- $[\mathrm{J} ; x \equiv x]$.

The most important consequences are

- $\sim$ distributes over all logical expressions ,
  $[x] \equiv [\sim x]$, $[\sim\sim x \equiv x]$
  $[\sim \mathrm{true} \equiv \mathrm{true}]$, $[\sim \mathrm{false} \equiv \mathrm{false}]$,
  $[\sim (x ; y) \equiv \sim y ; \sim x]$
- $;$ is universally disjunctive in both arguments, and hence monotonic
- the Grand Dedekind rule :

$$[\, x;y \;\wedge\; z \;\Rightarrow\; r;s \,]$$
$$\Leftarrow$$
$$[\, z;\sim y \;\wedge\; x \;\Rightarrow\; r \,]$$
$$\wedge$$
$$[\, \sim x;z \;\wedge\; y \;\Rightarrow\; s \,] \quad .$$

Its "symbol dynamics" – an aide-memoire towards remembering the rule – is as follows. It is a rule for separating the two operands in $r;s$. To separate the left operand, the left operand in $x;y$ is swapped with conjunct z and operand in $x;y$ that remains is transposed. The separate the right operand of $r;s$ the right operand of $x;y$ is swapped and the remaining one transposed.

This rule, which is very useful for the practice of relational calculation, has been designed by Henk Doornbos and appears in "A relational theory of data types", a technical report by R. Backhouse, E. Voermans, and J. van der Woude.

- There are two definitions of x being a leftcondition, a weak and a strong one:

  $[\, x;\, true \;\Rightarrow\; x \,]$ weak

  $[\, x;\, true \;\equiv\; x \,]$ strong

- x is leftcondition $\equiv$ $\sim x$ is rightcondition

(End of Appendix.)

(End of WF147.)

One down for the relational calculus

(whether you like it or not)

The following problem was communicated by Richard S. Bird. For any binary relation R and subset S on some universe, set $Min.R.S$ is defined by, for all x

$$x \in Min.R.S$$
$$\equiv$$
$$x \in S \;\wedge\; (\forall y: y \in S: x\,R\,y) \;.$$

The problem is to prove the

Theorem

For any two preorders X and Y there exists a preorder Z such that for all S

$$Min.Z.S = Min.Y.(Min.X.S) \;.$$

(A preorder is a reflexive and transitive binary relation.)

(End of Theorem.)

More specifically, Bird's assignment was to prove the theorem using the predicate calculus.

* * *

The very first thing we (= WF) did to the problem was change it into an exercise in the relational calculus rather than the predicate calculus. After all, Bird's theorem was a theorem on relations and we (= WF and many others) had

acquired some experience in relational calculation recently. So here we had a good opportunity for experimentation. Meanwhile these experiments have been done and some of them have been reported – e.g. [HD], [WF147], [JvdW] –, and the outcome should be sobering, to put it mildly. Now the time has come to return to Richard Bird's original assignment, and see what the ordinary predicate calculus can do for us.

* * *

The following proof is at best a minor variation of the proofs Lex Bijlsma, Anne Kaldewaij, and Jaap van der Woude have designed earlier – and independently –. As far as notation is concerned we shall stay closest to Kaldewaij's. (It need not amaze us that the three gentlemen have designed the "same" proof, since – as we shall see – the task is largely of the form "there is only one thing you can reasonably do".)

Our design starts focussing on the equation

$$\text{for all } S: \quad \mathrm{Min}.Z.S = \mathrm{Min}.Y.(\mathrm{Min}.X.S)$$

that we try to solve for Z. To that end we consider a calculation that begins with: for any u and S,

$$\begin{array}{ll} & u \in \mathrm{Min}.Y.(\mathrm{Min}.X.S) \\ = & \quad \{\text{definition of } \mathrm{Min}.Y\} \\ & u \in \mathrm{Min}.X.S \;\wedge\; (\forall v: v \in \mathrm{Min}.X.S : u\ Y\ v) \\ = & \quad \{\text{definition of } \mathrm{Min}.X\text{, twice}\} \\ & u \in S \;\wedge\; (\forall w: w \in S: u\ X\ w) \\ & \wedge\; (\forall v: v \in S \;\wedge\; (\forall w: w \in S: v\ X\ w): u\ Y\ v) \end{array}$$

$=$ {trading in the last conjunct}

(*0) $u \in S$

(*1) $\wedge$ $(\forall w : w \in S : u \mathrel{X} w)$

(*2) $\wedge$ $(\forall v : v \in S : (\forall w : w \in S : v \mathrel{X} w) \Rightarrow u \mathrel{Y} v)$,

and that ends with

(**) $u \in S \wedge (\forall w : w \in S : u \mathrel{Z} w)$

$=$ {definition of Min. Z}

$u \in \text{Min.}\, Z.\, S$.

The task that remains is to bridge the gap between the expressions (*) and (**). The two universal quantifications (*1) and (*2) are ready to be joined. The only problem is in the term of (*2) which still mentions S, in particular subexpression $(\forall w : w \in S : v \mathrel{X} w)$. For that expression we now observe

$(\forall w : w \in S : v \mathrel{X} w)$

$\Rightarrow$ { $u \in S$, i.e. (*0) }

$v \mathrel{X} u$

$\Rightarrow$ {predicate calculus}

$(\forall w : w \in S : v \mathrel{X} u)$

$=$ { (*1) }

$(\forall w : w \in S : v \mathrel{X} u \wedge u \mathrel{X} w)$

$\Rightarrow$ { X is transitive }

$(\forall w : w \in S : v \mathrel{X} w)$, i.e. the first line.

Hence, $(\forall w : w \in S : v \mathrel{X} w) \equiv v \mathrel{X} u$.
and now we can continue our calculation from (*) .

Remark The first step in the above little calculation is not a rabbit at all. The patented way to remove S from the expression $(\forall w : w \in S : v \; X \; w)$ is by a proper instantiation. In the context in which that expression resides there are only two elements of S available, viz. v and u. Instantiation with v would yield $v \; X \; v$ which, in view of X's reflexivity, equivales true, and therefore would make no sense.
(End of Remark.)

We now have

$$
\begin{array}{ll}
 & (*) \\
= & \quad \{ \text{by the above result} \} \\
 & u \in S \\
 & \wedge \; (\forall w : w \in S : u \; X \; w) \\
 & \wedge \; (\forall v : v \in S : v \; X \; u \; \Rightarrow \; u \; Y \; v) \\
= & \quad \{ \text{joining the terms} \} \\
 & u \in S \\
 & \wedge \; (\forall w : w \in S : u \; X \; w \\
 & \qquad \wedge \; (w \; X \; u \; \Rightarrow \; u \; Y \; w)) \\
= & \quad \{ \text{calculus of relations} \} \\
 & u \in S \\
 & \wedge \; (\forall w : w \in S : u \; (X \wedge (\sim X \Rightarrow Y)) \; w) \\
= & \quad \{ \text{choose:} \\
 & \qquad [\, Z \equiv X \wedge (\sim X \Rightarrow Y) \,] \quad \} \\
 & (**)
\end{array}
$$

* * *

The second part of our proof of Bird's theorem consists of showing that the result relation Z given by

$$[\ Z \equiv X \wedge (\sim X \Rightarrow Y)\ ]$$

is a preorder whenever X and Y are preorders. We supply this proof for completeness's sake (and for reasons of comparison with the corresponding proof in the relational calculus), although nothing is fascinating about it — a machine or a freshman could probably do it —.

- Z is reflexive, i.e. $u\ Z\ u$ for all u:

$$\begin{array}{ll} & u\ Z\ u \\ = & \quad \{\text{definition of } Z\} \\ & u\ X\ u \ \wedge\ u\ (\sim X \Rightarrow Y)\ u \\ \Leftarrow & \quad \{\text{calculus}\} \\ & u\ X\ u \ \wedge\ u\ Y\ u \\ = & \quad \{X \text{ and } Y \text{ are reflexive}\} \\ & \text{true}\ . \end{array}$$

- Z is transitive, i.e for all u, v, w

$$u\ Z\ v \ \wedge\ v\ Z\ w \ \Rightarrow\ u\ Z\ w$$

First we spell out the antecedent. It is the conjunction of

(a) $u\ X\ v$ (b) $v\ X\ w$

(c) $u\ (\sim X)\ v \Rightarrow u\ Y\ v$ (d) $v\ (\sim X)\ w \Rightarrow v\ Y\ w$

Next we spell out the consequent. It is the conjunction of

(e) $u\ X\ w$ (f) $u\ (\sim X)\ w \Rightarrow u\ Y\ w$.

Now we observe for (e)

$$
\begin{array}{ll}
 & u \mathbin{X} w \\
\Leftarrow & \quad \{ X \text{ is transitive} \} \\
 & u \mathbin{X} v \;\wedge\; v \mathbin{X} w \\
= & \quad \{ \text{(a) and (b)} \} \\
 & \text{true} \quad .
\end{array}
$$

and for (f)

$$
\begin{array}{ll}
 & u \mathbin{Y} w \\
\Leftarrow & \quad \{ Y \text{ is transitive} \} \\
 & u \mathbin{Y} v \;\wedge\; v \mathbin{Y} w \\
\Leftarrow & \quad \{ \text{(c) and (d)} \} \\
 & u \,(\sim X)\, v \;\wedge\; v \,(\sim X)\, w \\
= & \quad \{ \text{definition of } \sim \} \\
 & v \mathbin{X} u \;\wedge\; w \mathbin{X} v \\
\Leftarrow & \quad \{ X \text{ is transitive} \} \\
 & w \mathbin{X} u \\
= & \quad \{ \text{definition of } \sim \} \\
 & u \,(\sim X)\, w \quad , \text{ i.e. the antecedent of (f).}
\end{array}
$$

* * *

This is it. Is there a moral ? Perhaps, but in the first place there is the observation that a proof of Bird's theorem is far more easily constructed when using the predicate calculus than in case of using the relational calculus. This is a surprising outcome since one would guess that the relational calculus were the symbolism par excellence to tackle such a theorem on relations. Quod non.

We cannot leave it at this because we would

like to have a technical explanation for the observed phenomenon. One possibility is this. At some point we derived a very crucial equivalence, viz.

$$(***) \qquad (\forall w : w \in S : v\ X\ w) \equiv v\ X\ u \ .$$

It provided the link between the head and the tail of our main calculation. And it is precisely here, in expressing (***), where the relational calculus falls down. In a "dummy-free" notation like the relational calculus an expression like $v\ X\ u$ simply cannot see the light, so that nothing else can be done than to "program around it"

It can be argued that a formalism with reduced manipulative possibilities has the advantage that it reduces the solution spaces in solving problems. But at the same time we should then be willing to accept that the reduction may be so dramatic that solution spaces can collapse to empty. Here we wish to transmit to the relational calculators a question posed by Carel S. Scholten. Give five relations PQ, QR, RS, PR, and QS, find a relational expression Z such that

$$\begin{array}{l} p\ Z\ s \\ \equiv \\ (\exists q,r :: p\ PQ\ q \ \wedge\ q\ QR\ r \ \wedge\ r\ RS\ s \\ \qquad \wedge\ p\ PR\ r \ \wedge\ q\ QS\ s\ ) \ . \end{array}$$

It looks as if the straightjacket of the relational calculus, if not for theoretical reasons then for practical reasons anyway, cannot be

maintained.

W.H.J. Feijen,
14 November 1991
Eindhoven

[HD] Richard Bird's Problem,
Technical Report by Henk Doornbos

[WF147] Designing a proof for R.S. Bird's theorem on pre-orders,
Technical Report by W.H.J. Feijen

[JvdW] Free style spec wrestling II: preorders,
Technical Report by Jaap van der Woude
(to be published in The Squigollist.)

Some annotated proofs

Without the work of W.H.J. Feijen, A.J.M van Gasteren and R.M. Dijkstra, the following would not have been written; I owe them a lot.

The topic of this note is the design of formal proofs. Besides giving a number of proofs I shall give the considerations that are of relevance for their construction.

To begin with, we consider the beautiful theorem -from the collection of études composed by WF&AvG, and dubbed "the see-saw lemma by RMD -

(0) $[x \Rightarrow x;\sim x;x]$ for all x ,

to be proven from first principles.

Which first principles? Well, the consequent $x;\sim x;x$ suggests two things: firstly, because of the 2 semicolons, that the associativity of the composition will be involved, and, secondly, that we shall need the exchange rules as they provide the link between composition and transposition.

Because of the involvement of the exchange rules, we follow the suggestion of EWD1141 to switch to composition's conjugate, the confrontation, in terms of which the exchange rules are a little bit cleaner.

Extract I recall from EWD1141, that the "confrontation" is denoted by "!", is defined by $[\, x!y \equiv \neg(\neg x; \neg y)\,]$, and, hence, is associative, has an identity element $\neg\,]\, -$, and is universally conjunctive. In terms of !, the exchange rules are

$$[x!y \vee z] \equiv [z!\sim y \vee x] \qquad \text{left-exchange}$$

$$[x!y \vee z] \equiv [\sim x!z \vee y] \qquad \text{right-exchange.}$$

(End of Extract)

In terms of the confrontation, the see-saw lemma reads

(1) $[\, x!\sim x!x \Rightarrow x\,]$ for all x .

Remark In this note I do not intend to spend much attention to things such as the equivalence of (0) and (1). That part of predicate calculus - including $(f \circ g)^* = f^* \circ g^*$ - I assume here available. (End of Remark).

The three main proof patterns for establishing $[p \Rightarrow q]$ are schematically represented by

(i) $[p \Rightarrow q] \Leftarrow \text{true}$

(ii) $[p \vee z] \Rightarrow [q \vee z]$ for all z

(iii) $[\neg p \vee z] \Leftarrow [\neg q \vee z]$ for all z .

(Under (i) I capture here all notational variations such as

- transforming $p \wedge q$ in a sequence of value-preserving transformations into p
- transforming p in a sequence of weakening transformations into q (or $q \vee \neg p$), etc.

(ii) and (iii) are different for the universal quantification over the new dummy. Note that (ii) and (iii) represent two intrinsically different ways of introducing the dummy z: in (ii) "p" becomes a disjunct, in (iii) "$\neg p$" does so.)

The shape of the exchange rules suggests (ii), i.e. to establish (1) by establishing

(2) $[x\,!\,{\sim}x\,!\,x \vee z] \Rightarrow [x \vee z]$ for all x,z.

Remembering that the whole purpose of the introduction of z was to create the opportunity of applying an exchange rule, we now suggest to apply an exchange rule – for reasons of symmetry it probably does not matter which one –, i.e. to start our weakening chain as follows:

$$
\begin{aligned}
& [x\,!\,{\sim}x\,!\,x \vee z] \\
= & \quad \{\text{right-exchange}\} \\
& [{\sim}x\,!\,z \vee {\sim}x\,!\,x]
\end{aligned}
$$

(The alternative application of the right-exchange rule would have yielded $[\sim(x\,!\sim x)\,!\,z \vee x]$, which looks less inspiring, so we pursue the chosen application first.)

Had $!$ been disjunctive, the next step would have been easy: we would have rewritten our last result as $[\sim x\,!\,(z \vee x)]$. But $!$ is not disjunctive, but conjunctive, i.e. at least monotonic, and this we can exploit, remembering that we are building a weakening chain! We can continue

$$\begin{array}{ll} & [\sim x\,!\,z \vee \sim x\,!\,x] \\ \Rightarrow & \quad \{\,!\ \text{monotonic in its 2nd argument}\} \\ & [\sim x\,!\,(z \vee x)] \quad . \end{array}$$

Remark Here are three alternative formulations for "f is monotonic":

(i) $[p \Rightarrow q] \Rightarrow [f.p \Rightarrow f.q]$ for all p, q

(ii) $[f.p \vee f.q \Rightarrow f.(p \vee q)]$ for all p, q

(iii) $[f.p \wedge f.q \Leftarrow f.(p \wedge q)]$ for all p, q .

Monotonicity is too important a property to be known by (i) only; (ii) and (iii) can be generalized to existential and universal quantification respectively. (End of Remark.)

In view of the fact that our target expression $[x \vee z]$ contains no $\sim$, our last trans-

formation was a step in the right direction: the number of ~'s has been reduced from 2 to 1. The right-exchange is a proper mechanism for that, but is only applicable after we have made the confrontation into a disjunct. So we proceed

$$
\begin{array}{ll}
 & [\sim x\,!\,(z \vee x)] \\
= & \quad \{\text{predicate calculus}\} \\
 & [\sim x\,!\,(x \vee z) \vee \text{false}] \\
= & \quad \{\text{right-exchange}\} \\
 & [x\,!\,\text{false} \vee x \vee z]
\end{array}
$$

Remark An alternative for the first step, that would have turned the confrontation into a disjunct would have been an appeal to the idempotence of $\vee$: it would have yielded

$$[\sim x\,!\,(z \vee x) \vee \sim x\,!\,(z \vee x)] \quad ,$$

reintroducing a second ~ (and more complication). (End of Remark)

Now the only thing left is to get rid of the first disjunct x!false. However,

$$[x\,!\,\text{false} \vee x \vee z] \Rightarrow [x \vee z] \quad \text{for all } z$$

is equivalent to

$$[x\,!\,\text{false} \vee x \Rightarrow x]$$

or

$$[x\,!\,\text{false} \Rightarrow x]$$

which holds because ! has a neutral element. (Vide the conjugate relation [x ⇒ x;true].) So our proof can be completed:

```
    [ x!;false ∨ x ∨ z ]
=     { [ x!;false ⇒ x ]}
    [ x ∨ z ]
```

Summarizing the proof, we observe for any x,z

```
    [ x!;~x!;x ∨ z ]
=     { right-exchange }
    [~x!;z ∨ ~x!;x ]
⇒     {!, monotonic in second argument}
    [~x!;(z ∨ x)]
=     {predicate calculus}
    [~x!;(x ∨ z) ∨ false]
=     { right-exchange }
    [ x!;false ∨ x ∨ z]
=     { [ x!;false ⇒ x ]}
    [ x ∨ z]
```

Remark in retrospect The only rôle of the associativity of the composition has been to give us the choice between using the left- or the right-exchange. (End of Remark in retrospect.)

The reader is kindly invited to note that my "annotated" version of a 5-step argument took about 5 pages of manuscript. The moral of the story is that a formal proof can be a very

compact deposit of our considerations. Furthermore I would like to point out that most of our considerations seemed more general than just the relational calculus. The only really specific element was the lemma $[x!, false \Rightarrow x]$, but the need for that was the result of calculation.

* * *

The next theorem - also from WF&AvG - I want to deal with is

For any right-condition q

(3) $\quad [q; true; \neg q \equiv q; \neg q]$.

A moment's consideration shows that a ping-pong argument is indicated since for pong we don't even need to know that q is a right-condition:

$$
\begin{array}{ll}
 & [q; true; \neg q \Leftarrow q; \neg q] \\
\Leftarrow & \quad \{ \; ; \text{ monotonic} \} \\
 & [q; true \Leftarrow q] \\
= & \quad \{ \text{lemma} \} \\
 & true
\end{array}
$$

(The lemma follows thus:

$$
\begin{array}{ll}
 & x; true \\
\Leftarrow & \quad \{ [true \Leftarrow J] \text{ and } ; \text{ monotonic} \} \\
 & x; J \\
= & \quad \{ J \text{ neutral element of } ; \} \\
 & x \quad .
\end{array}
$$

)

With proving ping

(4) $[q;\mathit{true};\sim q \Rightarrow q;\sim q]$

I had serious problems, which in retrospect could be traced to an unsufficiently convenient characterization of "q is a right-condition". The original characterization was

$$[\mathit{true};q \equiv q] \quad ,$$

which is good to know, but since $[\mathit{true};x \Leftarrow x]$ holds for any x, the specific quality of being a right-condition is expressed by

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

and it was this equivalence with which I started my work on the current theorem. But it is more convenient its mutual implications separately, depending on whether one wants to show or to use that q is a right-condition. In the former case we use

(5) $[\mathit{true};q \Rightarrow q] \Rightarrow (q \text{ is a right-condition})$;

in the latter case we also exploit the monotonicity of composition, and get that for any x

(6) $[x;q \Rightarrow q] \Leftarrow (q \text{ is a right-condition})$

with the immediate consequence that for any x

(6') $[\sim q;x \Rightarrow \sim q] \Leftarrow (q \text{ is a right-condition})$.

The charm of (6), which almost looks like an

absorption rule, is the dummy x that we can instantiate as we see fit. In particular we can conclude (4) from

(7) $[q;\mathit{true};\sim q \Rightarrow q;\sim q;x]$

In particular -and here is still some sort of rabbit - : if x starts with "q;" we see in the consequent of (7) as subexpression the consequent of the see-saw lemma. So let us start with that:

$$
\begin{array}{ll}
 & \mathit{true} \\
= & \quad \{\text{see-saw lemma}\} \\
 & [q \Rightarrow q;\sim q;q] \\
\Rightarrow & \quad \{\text{monotonicity of ; in 1st argument}\} \\
 & [q;\mathit{true};\sim q \Rightarrow q;\sim q;q;\mathit{true};\sim q] \\
\Rightarrow & \quad \{(6') \text{ with } x := q;\mathit{true};\sim q \text{ and monotonicities}\} \\
 & [q;\mathit{true};\sim q \Rightarrow q;\sim q] \quad .
\end{array}
$$

After the introduction of the see-saw lemma, we compose both sides with "true;~q" to get the target antecedent, in the proces constructing the x with which to apply (6') in the last step.

Austin, 9 November 1992

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

More annotated proofs (a sequel to 1143)

From the collection of WF & AvG, we consider the theorem

(0) $[x \Rightarrow J] \wedge [y \Rightarrow J] \;\Rightarrow\; [x;y \equiv x \wedge y]$.

How do we prove this? Well, to the left we have "J", which does not occur to the right, where we have "$;$", which does not occur to the left. Hence we need a connection between "J" and "$;$", which is, of course, that "J" is the neutral element of "$;$". We can go further: because what the antecedent states about "J" is symmetric in x and y, which in the consequent occur as left- and right-hand argument of the "$;$", we can expect to need that "J" is, both, the left-neutral and the right-neutral element of the composition.

In view of the above it is tempting to start a weakening chain at the antecedent and head for the expression $x;y$. Thus we observe

$$
\begin{array}{ll}
 & [x \Rightarrow J] \wedge [y \Rightarrow J] \\
\Rightarrow & \quad \{; \text{ monotonic in both arguments}\} \\
 & [x;y \Rightarrow J;y] \wedge [x;y \Rightarrow x;J] \\
= & \quad \{J \text{ is left- and right-neutral element of } ;\} \\
 & [x;y \Rightarrow y] \wedge [x;y \Rightarrow x] \\
= & \quad \{\text{predicate calculus}\} \\
 & [x;y \Rightarrow x \wedge y]
\end{array}
$$

So ping has been proved; note that, in the mean time, we have used composition's monotonicity. We still have to show pong:

(1) $[x \Rightarrow \mathcal{J}] \wedge [y \Rightarrow \mathcal{J}] \Rightarrow [x \wedge y \Rightarrow x;y]$.

The antecedent - and we have used this in the proof of ping - implies that the prefix operators like "$x;$" and the postfix operators like "$;y$" $\sqsubseteq$. If we wish to use that to strengthen $[x \wedge y \Rightarrow x;y]$, we have to strengthen a term in the consequent. Hence some shunting seems indicated:

$\sqsubseteq$ are strengthening.

$$
\begin{array}{ll}
 & [x \wedge y \Rightarrow x;y] \\
= & \quad \{\text{shunting}\} \\
 & [x \Rightarrow x;y \vee \neg y] \\
= & \quad \{\mathcal{J} \text{ neutral element}\} \\
 & [x \Rightarrow x;y \vee \mathcal{J};\neg y] \\
\Leftarrow & \quad \{[x \Rightarrow \mathcal{J}]\} \\
 & [x \Rightarrow x;y \vee x;\neg y] \\
= & \quad \{\ ; \text{ over } \vee\} \\
 & [x \Rightarrow x;(y \vee \neg y)] \\
= & \quad \{\text{predicate calculus}\} \\
 & [x \Rightarrow x;\text{true}] \\
= & \quad \{\text{relational calculus}\} \\
 & \text{true}
\end{array}
$$

The way in which the symmetry is destroyed in the shunting step is strange and surprising. (My first proof shunted to $[x;y \vee \neg x \vee \neg y]$ and maintained the symmetry between x and y as

much as possible.) The surprise reflects that if we drop in the antecedent one of the conjuncts, it is still a theorem. In the mean time, we have also used composition's distribution over $\vee$; in the sequence "monotonic", "distributing over $\vee$", "universally disjunctive" was that the next stronger property of composition to be taken into account, thus presenting a further invitation to do some shunting.

* * *

The next theorem we owe to the WF&AvG is

For p a left-condition

(2) $\quad [\, p;p \equiv p \,]$

where - see EWD1143 - we characterize p's being a left-condition by

(3) $\quad [\, p;x \Rightarrow p \,]$ for all x

Remark Note similarity and difference between (3) and our consequences of $[\, y \Rightarrow J \,]$ in the previous example, such as

$$[\, x;y \Rightarrow x \,] \quad \text{for all } x \; .$$

Not surprisingly, our current proof shows structural similarities with the previous one. (End of Remark.)

Again, ping is a walk-over:

$$
\begin{array}{ll}
 & \text{true} \\
= & \quad \{(3) \text{ with } x := p\} \\
 & [\, p;p \Rightarrow p \,] \quad .
\end{array}
$$

When we tackled pong in class -without all these preliminaries- and looked at our proof obligation

$$[\, p \Rightarrow p;p \,] \quad \text{for left-condition } p,$$

we took the standard approach of listing the "direct consequences" or "simple properties" of the ingredients

- logical expressions built from left-conditions are left-conditions; having only left-condition p this leads to $\neg p$ being a left-condition:

(4) $\quad [\neg p;\, x \Rightarrow \neg p] \quad$ for all x .

- $\Rightarrow$ can be expressed with $(\wedge \equiv)$, $(\vee \equiv)$, or $(\neg \vee)$; further shunting and contrapositive;
  $\Rightarrow$ is transitive

- $;$ is monotonic, distributes over $\vee$, is universally disjunctive;
  $;$ has an identity element
  $;$ is associative

Looking at these properties we concluded that they suggested to use the implication in the proof obligation for the introduction of $\neg$ and $\vee$, and (after some polishing) we came up with

$$
\begin{array}{ll}
 & [\,p \Rightarrow p;p\,] \\
= & \quad \{\text{sort of shunting}\} \\
 & [\,p \Rightarrow p;p \vee \neg p\,] \\
\Leftarrow & \quad \{(4) \text{ with } x := p\} \\
 & [\,p \Rightarrow p;p \vee \neg p;p\,] \\
= & \quad \{\,;\text{ over } \vee\} \\
 & [\,p \Rightarrow (p \vee \neg p);p\,] \\
= & \quad \{\text{pred. calc.}\} \\
 & [\,p \Rightarrow \text{true};p\,] \\
= & \quad \{\text{rel. calc.}\} \\
 & \text{true.}
\end{array}
$$

Remark The "sort of shunting" - instead of eliminating the $\Rightarrow$ by rewriting the demonstrandum as $[\neg p \vee p;p]$ - was introduced in the polishing phase; it is purely cosmetic. (End of Remark.)

Remark In the context in which this problem was posed,

(5) $[(p \wedge x);y \equiv p \wedge x;y]$ for left-condition p

had been established. Rutger M. Dijkstra used it to establish (2) without ping-pong argument:

$$
\begin{array}{ll}
 & \text{true} \\
= & \quad \{(5) \text{ with } x,y := \text{true}, p\} \\
 & [(p \wedge \text{true});p \equiv p \wedge \text{true};p] \\
= & \quad \{\text{pred. calc. ; rel. calc.}\} \\
 & [\,p;p \equiv p\,]
\end{array}
$$

* * * (End of Remark.)

It is customary to denote the strongest solution of

$$x: [\,] \vee a;x \Rightarrow x\,]$$

by a^* - read "a star" and called "the reflexive transitive closure of a" - . In EWD1136 we compared proofs for the theorem that, in our new terminology, the strongest solution of

$$x: [\,b \vee a;x \Rightarrow x\,]$$

is $a^*;b$. Knowing the theorem of Knaster-Tarski, we see that all this is expressed by

(6) $\quad [\,] \vee a;a^* \equiv a^*\,]$

(7) $\quad [\,b \vee a;x \Rightarrow x\,] \Rightarrow [\,a^*;b \Rightarrow x\,]$

(Formula (6) expresses that a^* solves the first equation; (7) with $b := \mathcal{J}$ expresses that a^* implies all its solutions. Applying the postfix operator ";b" to both sides of (6) tells us that $a^*;b$ solves the second equation, while (7) expresses that $a^*;b$ implies all its solutions.)

Following the traditions, we shall give the postfix operator * in its exponential position a higher binding power than all other (relational and logical) operators. (In a note about notation, I should probably argue in favour of a prefix operator, but this note is about proof design.)

We are now going to show that $\sim$ and * commute -distribute over each other-, i.e.

(8) $\quad [(\sim s)^* \equiv \sim s^*]$

Because of the structure of (7), proofs of theorems about extreme solutions are often ping-pong arguments. We shall first demonstrate ping: it is a proof largely constructed on the principle "there is only one thing you can do". [It is therefore not a tribute to RMD, who produced the following proof almost verbatim.]

$$
\begin{array}{ll}
 & [(\sim s)^* \Rightarrow \sim s^*] \\
\Leftarrow & \quad \{(7) \text{ with } a, b, x := \sim s, J, \sim s^*\} \\
 & [J \vee \sim s ; \sim s^* \Rightarrow \sim s^*] \\
= & \quad \{\text{relational calculus, applying } \sim \text{ to both sides}\} \\
 & [J \vee s^* ; s \Rightarrow s^*]
\end{array}
$$

The first step is dictated, because (7) is our only tool for concluding that a closure implies something. The "intuitive" calculator will justify the next step as "cleaning up", as a way of removing a large number of tildes. The conscious calculator will realize, that we can deal - see (7)- with a demonstrandum where the closure is the <u>left</u> argument of a composition; hence the decision to apply $\sim$ to both sides. Next, heading for a re-application of (7), we had better find a way of removing "$J \vee$" from the antecedent.

The only other place where J occurs is (6), so that is what we appeal to — in the hint I have recorded the direction of the appeal —

$$
\begin{array}{ll}
 & [\mathrm{J} \vee s^*;s \Rightarrow s^*] \\
\Leftarrow & \quad \{(6)\Rightarrow \text{ with } a := s\} \\
 & [\mathrm{J} \vee s^*;s \Rightarrow \mathrm{J} \vee s;s^*] \\
\Leftarrow & \quad \{\text{monotonicity of } \vee\} \\
 & [s^*;s \Rightarrow s;s^*]
\end{array}
$$

and now we have again a demonstrandum with an antecedent to which (7) is applicable. So full of faith we continue:

$$
\begin{array}{ll}
 & [s^*;s \Rightarrow s;s^*] \qquad \dagger \\
\Leftarrow & \quad \{(7) \text{ with } a,b,x := s,s,s;s^*\} \\
 & [s \vee s;s;s^* \Rightarrow s;s^*] \\
= & \quad \{\ ;\ \text{over } \vee\} \\
 & [s;(\mathrm{J} \vee s;s^*) \Rightarrow s;s^*] \\
\Leftarrow & \quad \{;\ \text{monotonic in 2nd argument}\} \\
 & [\mathrm{J} \vee s;s^* \Rightarrow s^*] \\
= & \quad \{(6)\Rightarrow \text{ with } a := s\} \\
 & true \qquad .
\end{array}
$$

In passing we note that we did not need the theorem of Knaster-Tarski: the proof could as well have been carried out, had (6) been replaced by the — formally much! — weaker

(6') $\qquad [\mathrm{J} \vee a;a^* \Rightarrow a^*]$.

I interpret this observation as a further confirmation of my suspicion that it is misleading

to call a^* "the strongest fixpoint of $\langle \lambda x: \rceil \vee a; x \rangle$" : there are too many arguments in which the "being a fixpoint" is not relevant.

Having proved ping: $[(\sim s)^* \Rightarrow \sim s^*]$ for all s, pong is now easy: the calculation below

$$
\begin{array}{ll}
 & \text{true} \\
= & \quad \{\text{ping with } s := \sim s\} \\
 & [(\sim\sim s)^* \Rightarrow \sim(\sim s)^*] \\
= & \quad \{\sim \text{ is an involution}\} \\
 & [s^* \Rightarrow \sim(\sim s)^*] \\
= & \quad \{\text{transposition}\} \\
 & [\sim s^* \Rightarrow (\sim s)^*]
\end{array}
$$

establishes pong, and hence (8) has been proved.

Reading the proof on the previous page, line ≠ shows that we have proved a next ping: $[s^*; s \Rightarrow s; s^*]$ for all s. The calculation below

$$
\begin{array}{ll}
 & \text{true} \\
= & \quad \{\text{ping with } s := \sim s\} \\
 & [(\sim s)^*; \sim s \Rightarrow \sim s; (\sim s)^*] \\
= & \quad \{(8)\} \\
 & [\sim s^*; \sim s \Rightarrow \sim s; \sim s^*] \\
= & \quad \{\text{rel. calc.}\} \\
 & [\sim(s; s^*) \Rightarrow \sim(s^*; s)] \\
= & \quad \{\text{transposition}\}
\end{array}
$$

$$[\ s;s^* \Rightarrow s^*;s\ ]$$

establishes the corresponding pong, hence

(9) $$[\ s^*;s \equiv s;s^*\ ] \quad .$$

* * *

As a final example of the use of (6) and (7) we shall show that * is a closure, i.e. monotonic, weakening, and idempotent.

<u>* is monotonic</u>. We have to show for arbitrary s and t that

$$[\ s \Rightarrow t\ ] \Rightarrow [\ s^* \Rightarrow t^*\ ] \quad .$$

The proof is standard:

$$\begin{array}{ll} & [\ s^* \Rightarrow t^*\ ] \\ \Leftarrow & \quad \{(7) \text{ with } a,b,x := s, \rceil, t^*\} \\ & [\ \rceil \vee s;t^* \Rightarrow t^*\ ] \\ \Leftarrow & \quad \{(6)\Rightarrow \text{with } a := t\} \\ & [\ \rceil \vee s;t^* \Rightarrow \rceil \vee t;t^*\ ] \\ \Leftarrow & \quad \{\text{monotonicities}\} \\ & [\ s \Rightarrow t\ ] \quad . \end{array}$$

Note that we did not need Knaster-Tarski.

<u>Remark</u> The fact that we first appealed to (7) and then to (6) is not essential, but in this order it works better; in the other order we would need a second appeal to (6). (End of Remark.)

* is weakening We have to show for arbitrary s that

$$[\, s \Rightarrow s^* \,] \quad .$$

Since this is a demonstrandum with s^* as consequent, (7) is in this context irrelevant, and we shall only appeal to (6). This is perhaps the place to remark that (when we are not interested in Knaster-Tarski) we can draw to separate conclusions from (6)

(10) $\quad [\, \rceil \Rightarrow a^* \,]$

(11) $\quad [\, a;a^* \Rightarrow a^* \,]$.

We now observe

$$\begin{array}{ll} & s^* \\ \Leftarrow & \quad \{(11) \text{ with } a := s\} \\ & s;s^* \\ \Leftarrow & \quad \{(10) \text{ with } a := s \text{, monotonicity of } ;\} \\ & s;\rceil \\ = & \quad \{\text{relational calculus}\} \\ & s \end{array} \quad .$$

* is idempotent We have to show for arbitrary s that $[\, s^{**} \equiv s^* \,]$, but since we have just shown that * is weakening, it suffices to show that for any s

$$[\, s^{**} \Rightarrow s^* \,] \quad .$$

To this end we observe for any s

$[s^{**} \Rightarrow s^*]$

$\Leftarrow$ {(7) with $a,b,x := s^*, J, s^*$}

$[J \vee s^*;s^* \Rightarrow s^*]$

$=$ {(10) with $a := s$}

$[s^*;s^* \Rightarrow s^*]$

$\Leftarrow$ {(7) with $a,b,x := s, s^*, s^*$}

$[s^* \vee s;s^* \Rightarrow s^*]$

$=$ {predicate calculus}

$[s;s^* \Rightarrow s^*]$

$=$ {(11) with $a := s$}

true.

* *

In the above proof I can think of only one place where we could have gone wrong:

$[s^*;s^* \Rightarrow s^*]$

$\Leftarrow$ {monotonicity}

$[s^* \Rightarrow J]$.

With the amount of annotation decreasing so rapidly, this EWD had better be concluded.

Austin, 13 November 1992

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

On the design of calculational proofs

§ 0. By their brevity and unexpected turns, calculational proofs often strike the reader as being very ingenious constructions. Far be it from us to belittle the ingenuity of their inventors, but that should not prevent us from studying a number of representative proofs in the hope of learning why those proofs have the structure they have. This paper reports on such a study.

Among the things that will emerge, I already mention a few.

Firstly, many steps that may seem surprising at first sight, are, in fact, (almost) dictated, as they are the only (or by far the simplest) transformation that will enable us to exploit one of the givens that has to be taken into account. In this connection I recommend that we maintain as fine-grained a bookkeeping as possible of what of the givens we have used: what has not been used yet often indicates the direction in which the proof should be completed

Secondly, when faced with an antisymmetric and reflexive relation, i.e. a relation $\leq$ (with transpose $\geq$) such that

$$x = y \;\equiv\; x \leq y \wedge x \geq y \quad \text{for all } x, y \;,$$

we must be prepared for a proof with a possibly subtle interplay between calculating with equalities and with inequalities: any hints we can extract as to when to stress which of the two approaches will obviously be most welcome.

§ 1. Let me show you, as a first indication of the kind of considerations I would like to highlight, a calculational proof of which each step is forced (as a result of which the proof is very well known, not to say canonical).

About predicate transformers f and g is given

(0) $\quad [f.x \Rightarrow y] \equiv [x \Rightarrow g.y] \quad$ for all x, y ;

we are asked to show that g is universally conjunctive, i.e.

(1) $\quad [g.\langle \forall y : y \in W : y \rangle \equiv \langle \forall y : y \in W : g.y \rangle]$
for any bag W of predicates.

Remark We indicate function application not implicitly by juxtaposition but explicitly by an infix period of high syntactic binding power. With the square brackets we denote the "everywhere operator", a function from "predicates" to the "boolean scalars" true and false . For further elaboration

we refer the reader for instance to [0]. (End of Remark.)

For brevity's sake, the range "$y \in W$" is omitted in the following calculation; we should remember that it is a scalar range. We shall first give the proof and then discuss it.

Proof We observe for any x

$$
\begin{array}{ll}
 & [\, x \Rightarrow g.\langle \forall y :: y \rangle \,] \\
= & \quad \{(0) \text{ with } y := \langle \forall y :: y \rangle\} \\
 & [\, f.x \Rightarrow \langle \forall y :: y \rangle \,] \\
= & \quad \{\text{pred. calc.: } Q\Rightarrow \text{ over } \forall\} \\
 & [\, \langle \forall y :: f.x \Rightarrow y \rangle \,] \\
= & \quad \{\text{pred. calc.: interchange (NB. range is scalar)}\} \\
 & \langle \forall y :: [\, f.x \Rightarrow y \,] \rangle \\
= & \quad \{(0)\} \\
 & \langle \forall y :: [\, x \Rightarrow g.y \,] \rangle \\
= & \quad \{\text{pred. calc.: interchange (NB. range is scalar)}\} \\
 & [\, \langle \forall y :: x \Rightarrow g.y \rangle \,] \\
= & \quad \{\text{pred. calc.: } Q\Rightarrow \text{ over } \forall\} \\
 & [\, x \Rightarrow \langle \forall y :: g.y \rangle \,] \qquad ;
\end{array}
$$

having thus established

$$[\, x \Rightarrow g.\langle \forall y :: y \rangle \,] \equiv [\, x \Rightarrow \langle \forall y :: g.y \rangle \,] \quad \text{for all } x,$$

we conclude (1) on account of

$$(2) \quad \langle \forall x : \text{true} : [\, x \Rightarrow P \,] \equiv [\, x \Rightarrow Q \,] \rangle \equiv [\, P \equiv Q \,] \quad ,$$

which is a well-known lemma that follows from implication's reflexivity and antisymmetry.

(End of Proof.)

Let us now analyse the extent to which the structure of the above proof has been forced upon us.

Firstly, do we need (0) in order to show (1)? Yes, we do, because (with $W := \emptyset$) a consequence of (1) is $[g.true \equiv true]$, and that definitely does not hold for any g, for instance not for the predicate transformer g defined by $[g.x \equiv false]$ for all x.

Next, being obliged to use (0), we observe that our <u>only</u> way of exploiting that the g in the left-hand side $g.\langle \forall y :: y \rangle$ of (1) satisfies (0), is to instantiate the latter by $y := \langle \forall y :: y \rangle$; this <u>forces</u> us to explore $[x \Rightarrow g.\langle \forall y :: y \rangle]$, this now being the smallest expression that contains the left-hand side of (1) and can be rewritten using (0). The first step then applies (0) with that instantiation.

In view of the right-hand side of (1), it is from here on our duty to eliminate f and to reintroduce g, which can <u>only</u>

be done by a second appeal to (0). Because there is no point in undoing the first step, we must apply (0) to a subexpression of the form $[f.x \Rightarrow ?]$ with a consequent different from $\langle \forall y :: y \rangle$. The next two steps of the predicate calculus form such a subexpression, and then our second appeal to (0) reintroduces g.

Our final task is to form $\langle \forall y :: g.y \rangle$, the right-hand side of (1), as subexpression; the next two steps of the predicate calculus form $\langle \forall y :: g.y \rangle$ as consequent, and, on account of (2), we are done.

The moral of the story is that, apart from the choice of syntax for proof presentation, we had no choice.

§ 2. The full-blown principle of Leibniz states (to the best of my knowledge) that for x and y of some type, and f ranging over the functions on that type

$$(3) \qquad x = y \equiv \langle \forall f :: f.x = f.y \rangle \quad .$$

Reading the above equivalence as a mutual implication, we observe that "$\Leftarrow$" follows by instantiating the right-hand side with for f the identity function. For this con-

clusion, even the existence of other functions is irrelevant. For the general function it is the implication in the other direction that is relevant:

$$(4) \qquad x = y \;\Rightarrow\; f.x = f.y \quad ,$$

and this is what is usually referred to as "Leibniz's Principle".

Remark Appealing to

$$q \Rightarrow \langle \forall x :: x \rangle \;\equiv\; \langle \forall x :: q \Rightarrow x \rangle$$

— itself an immediate consequence of "$\vee$ distributes over $\forall$" —, we have taken the universal quantification over f outside. The possibility of thus removing a universal quantification from a consequent often provides the incentive to split up an equivalence involving a universally quantified equivalent into a mutual implication. (End of Remark.)

The connection between function application and equality is very tight: function application preserves equality and, conversely, — besides the two constant relations *true* and *false* — equality is the *only* relation that is preserved by function application. In Section 0, we referred to the interplay between calculating with equalities and with

inequalities; the above observation tells us that when we have to capture the essence of function application, i.e. when we have to apply Leibniz's Principle, we may be forced to introduce an equality so as to be able to do so. The next section gives two examples of this phenomenon.

§ 3. In a wider context, Jamey Leifer had to deal with two functions f and g whose domains and ranges are all the same, and whose applications commute, i.e.

(5) $$f.(g.x) = g.(f.x) \quad \text{for "all" } x \ .$$

In order to appeal to (5), we have to apply -say- g to an f-value, say $f.x$. In order to appeal to Leibniz's Principle, that f-value has to equal something else, say y, for the time being. So we observe for any x, y

$$\begin{array}{ll} & f.x = y \\ \Rightarrow & \quad \{ \text{Leibniz's Principle} \} \\ & g.(f.x) = g.y \\ = & \quad \{(5)\} \\ & f.(g.x) = g.y \ . \end{array}$$

The last line tells us that a nice choice for y is x, and thus we have proved

$$f.x = x \;\Rightarrow\; f.(g.x) = g.x \quad ,$$

or, in words: if x is a fixed point of f, so is $g.x$. After the above observation, Jamey Leifer conducted all the rest of his (beautiful) argument in terms of fixed points, a concept that did not occur at all in the theorem he had set out to prove.

§ 4. The other example I would like to show is the proof of Perry Moerland's lemma. (Also this came up in a wider context.) About predicate transformer f we are given that it is monotonic and has a left-inverse g, i.e.

(6) $\quad [x \Rightarrow y] \Rightarrow [f.x \Rightarrow f.y]$ for all x,y

(7) $\quad [g.(f.x) \equiv x]$ for all x;

we have to show that as a consequence

(8) $\quad [x] \Leftarrow [f.x]$ for all x.

We are looking for a strengthening chain of booleans, starting at $[x]$ and ending with $[f.x]$. Along the way we have to introduce f; for doing so, (6) is not a good candidate, but (7) is. The advantage of introducing f with the aid of (7) is that then g is introduced as

well. (Note that (6) by itself does not suffice for the demonstration of (8), i.e. that (7) has to be exploited, but that exploitation obviously requires that g enters the picture.)

Having introduced g and still aiming for $[f.x]$, we have to get rid of g again. Since the only other thing known about g is that it is a function, the proper candidate for the removal of g is Leibniz's Principle. Since an appeal to the latter requires equalities, the introduction of an equality was Perry Moerland's first concern; one observes for any x

$$
\begin{array}{ll}
 & [\,x\,] \\
= & \quad \{\text{predicate calculus}\} \\
 & [\,x \equiv \text{true}\,] \\
= & \quad \{(7)\,;\,(7) \text{ with } x := \text{true}\} \\
 & [\,g.(f.x) \equiv g.(f.\text{true})\,] \\
\Leftarrow & \quad \{\text{Leibniz's Principle}\} \\
 & [\,f.x \equiv f.\text{true}\,] \\
= & \quad \{\text{predicate calculus}\} \\
 & [\,f.x \Leftarrow f.\text{true}\,] \wedge [\,f.x \Rightarrow f.\text{true}\,] \\
= & \quad \{[\,x \Rightarrow \text{true}\,] \text{ and } (6) \text{ with } y := \text{true}\} \\
 & [\,f.x \Leftarrow f.\text{true}\,] \\
\Leftarrow & \quad \{\text{predicate calculus}\} \\
 & [\,f.x\,]
\end{array}
$$

§5. With a relation like $\geq$ (and its transpose $\leq$) comes the notion of monotonicity with respect to it, i.e. "preserving it": "f is monotonic with respect to $\geq$" means

(9) $\quad x \geq y \;\Rightarrow\; f.x \geq f.y \quad$ for all x, y .

(Note that in this terminology, Leibniz's Principle states that each function is "monotonic with respect to $=$".)

We now restrict ourselves to such domains and relations $\geq$ such that for any set of values the lowest higher bound "$\uparrow$" —and similarly the highest lower bound "$\downarrow$"— exists, i.e. we assume that for any range of the dummy x and any function t of the appropriate type we can define $\langle \uparrow x :: t.x \rangle$ by

(10) $\quad w \geq \langle \uparrow x :: t.x \rangle \;\equiv\; \langle \forall x :: w \geq t.x \rangle$ for all w.

The important theorem is that, for any f that is monotonic with respect to $\geq$,

(11) $\quad f.\langle \uparrow x :: t.x \rangle \geq \langle \uparrow x :: f.(t.x) \rangle$;

its proof is another example of "there is only one thing you can do".

Proof We observe for any monotonic f, etc.

$$
\begin{array}{ll}
 & f.\langle \uparrow x :: t.x\rangle \geq \langle \uparrow x :: f.(t.x)\rangle \\
= & \quad \{ \text{(10) with } w,t := f.\langle \uparrow x :: t.x\rangle, f\circ t \} \\
 & \langle \forall x :: f.\langle \uparrow x :: t.x\rangle \geq f.(t.x)\rangle \\
\Leftarrow & \quad \{ f \text{ monotonic with respect to } \geq , \\
 & \qquad \forall \text{ monotonic with respect to } \Leftarrow \} \\
 & \langle \forall x :: \langle \uparrow x :: t.x\rangle \geq t.x\rangle \\
= & \quad \{ \text{(10) with } w := \langle \uparrow x :: t.x\rangle \} \\
 & \langle \uparrow x :: t.x\rangle \geq \langle \uparrow x :: t.x\rangle \\
= & \quad \{ \geq \text{ is reflexive} \} \\
 & \text{true}
\end{array}
$$

Note that (10), when viewed as mutual implication, is used in either direction.

(End of Proof.)

The above proof is so simple, and so completely forced by the definitions, that the usefulness of the theorem may come as a surprise. Predicate transformers that are monotonic with respect to implication are very common, and for such an f, which satisfies (6), we have

(12) $[f.\langle \forall x :: x\rangle \Rightarrow \langle \forall x :: f.x\rangle]$ for all ranges

(13) $[f.\langle \exists x :: x\rangle \Leftarrow \langle \exists x :: f.x\rangle]$ " ,

with the special cases

(14) $[f.(x \wedge y) \Rightarrow f.x \wedge f.y]$ for all x,y

(15) $[f.(x \vee y) \Leftarrow f.x \vee f.y]$ " .

In fact, (12) through (15) are more than useful consequences of f's monotonicity, they are restatements of it.

§6. In section 1 we proved - in our appeal to (2) - $[P \equiv Q]$ by showing for any x

$$[x \Rightarrow P] \equiv [x \Rightarrow Q] \quad .$$

We may well ask what has been gained by the introduction of the dummy x. A possible answer is given by the observation that, in order to derive $[P \equiv Q]$ from the above, the latter has to be instantiated twice, viz. with $x := P$ and with $x := Q$. This double instantiation strongly suggests that the the introduction of dummy x has enabled us to avoid a ping-pong argument (i.e. a twofold case analysis).

But this is not the whole story, for, analogously to (2), we have, for instance,

(16) $\langle \forall x :: [P \Rightarrow x] \Leftarrow [Q \Rightarrow x] \rangle \equiv [P \Rightarrow Q]$.

This authorizes us to conclude $[P \Rightarrow Q]$ by showing for any x

(17) $[P \Rightarrow x] \Leftarrow [Q \Rightarrow x]$;

the conclusion, however, requires a single instantiation only, viz. $x := Q$. Yet, the introduc-

tion of dummy x as in (17) is justifiable in more than one way. We can (i) prove (17) for arbitrary x, and then instantiate (17) with $x := Q$, or (ii) carry out the argument not for arbitrary x, but right from the start for Q, thus deriving, instead of (17),

(18) $$[P \Rightarrow Q] \Leftarrow [Q \Leftarrow Q] \quad .$$

Here (i) is to be preferred above (ii) for more than one reason:

- the dummy x is shorter than the expression Q, hence (i) saves ink and paper;
- argument (ii) obscures the existence of demonstrandum (17), obscures the fact that the internal structure of the first Q and of the last Q in (18) is totally irrelevant for the latter's demonstration; argument (i) is clearer than (ii) in that it heads for (17), which clearly states that the first and the last Q's in (18) could be "anything", provided they are the same. In the jargon: argument (i) is better disentangled than argument (ii).
- the full force of this disentanglement manifests itself when, besides the proof, we design the theorem as well and Q emerges as the result of our calculations.

We shall illustrate the findings of the last two sections (and a little bit more) in the next section.

§7. In this section, we are looking, by way of example, for a nontrivial theorem about relational composition. Composition will be denoted by an infix ";" with a syntactic binding power higher than that of the logical infix operators but lower than that of the unary operators "¬" and "~".

We shall use that the transposition "~" distributes over the logical operators, that composition is monotonic in both its arguments, and that composition and transposition are coupled by the "right-exchange", for which we choose here the formulation that for all x, y, z

(19) $[x;y \wedge z \Rightarrow \text{false}] \equiv [\sim x; z \wedge y \Rightarrow \text{false}]$.

The exchange rules enable us to manipulate a composition that occurs as antecedent, but there are <u>no</u> analogous manipulative opportunities for a composition that occurs as consequent, and we may therefore expect a problem when trying to prove a theorem of the form

(20) $[a \Rightarrow b;c]$.

This problem, however, can be overcome with the aid of (16), which states that we can demonstrate (20) by proving - in a variation

on (17) – for arbitrary z

(21) $[b;c \wedge z \Rightarrow \mathit{false}] \Rightarrow [a \wedge z \Rightarrow \mathit{false}]$,

and in this formulation of our proof obligation, the composition "b;c" occurs as (conjunct in an) antecedent, and is thus amenable to manipulation via an exchange rule!

We shall now first give our calculation that transforms $[b;c \wedge z \Rightarrow \mathit{false}]$ without strengthening into $[a \wedge z \Rightarrow \mathit{false}]$, and shall discuss the proof later. In the course of the calculation, suitable values for c and a will be chosen. We observe for any z

$$
\begin{array}{lll}
 & [b;c \wedge z \Rightarrow \mathit{false}] & \\
= & \quad \{\text{right-exchange}\} & (A) \\
 & [\sim b;z \wedge c \Rightarrow \mathit{false}] & \\
= & \quad \{\text{choose } c\text{: } [c \equiv \sim d;e \wedge f]\} & (B) \\
 & [\sim b;z \wedge \sim d;e \wedge f \Rightarrow \mathit{false}] & \\
\Rightarrow & \quad \{\text{monotonicities}\} & (C) \\
 & [(\sim b \wedge \sim d);(z \wedge e) \wedge f \Rightarrow \mathit{false}] & \\
= & \quad \{\sim \text{ distributes over } \wedge\} & (D) \\
 & [\sim(b \wedge d);(z \wedge e) \wedge f \Rightarrow \mathit{false}] & \\
= & \quad \{\text{right-exchange}\} & (E) \\
 & [(b \wedge d);f \wedge e \wedge z \Rightarrow \mathit{false}] & \\
= & \quad \{\text{choose } a\text{: } [a \equiv (b \wedge d);f \wedge e]\} & (F) \\
 & [a \wedge z \Rightarrow \mathit{false}] \quad , &
\end{array}
$$

and thus we have proved (20) with the choices made in steps (B) and (F), i.e.

(22) $[(b \wedge d);f \wedge e \Rightarrow b;(\sim d;e \wedge f)]$,

a theorem, even uglier than "the first ugly theorem" of [1].

Remark We have used (19) instead of the formulation of the right-exchange from [2]

(23) $[x;y \Rightarrow z] \equiv [\sim x;\neg z \Rightarrow \neg y]$

and (21) instead of what (17) would have given, viz.

$$[b;c \Rightarrow z] \Rightarrow [a \Rightarrow z] \quad ,$$

thus avoiding the need of introducing negations and moving terms back and forth with the shunting rule

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

In [1], this economy is achieved by the introduction of the "somewhere" operator. (End of Remark.)

Step A is not surprising because (21) was intentionally chosen so as to make the right-exchange applicable to its antecedent.

The purpose of step B is to apply the most general substitution for b or c

that makes continued manipulation possible. For b, I could not find a productive substitution: after b:= d ∨ e, we can use composition's monotonicity as in (15) but that creates 2 occurrences of dummy z, and after b:= d ∧ e, we are stuck because then composition's monotonicity as in (14) works in the wrong direction. If, however, we substitute a composition like ~d;e for c, then monotonicity as in (14) works in the right direction. The inclusion of the additional "∧ f" is, in a sense, obligatory if we wish to keep our options open: since c occurs as conjunct and conjunction is associative, the additional "∧ f" does not hamper the combination of the two compositions, and, furthermore, omission of the conjunct "∧ f" can be interpreted as a possibly premature instantiation f:= true. (The choice of "~d;e" instead of "d;e" is an irrelevant matter of elegance; it allows us to ignore that "~" is its own inverse.)

Step C' exploits the monotonicity of ";" (plus the usual ones from predicate calculus), and, in view of our

target, our remaining duty is to extricate dummy z from the composition.

Step E performs this extraction by – what else? – a second appeal to right-exchange after step D has prepared the ground. (Step D could have been postponed, but then later simplification would have needed that "~" is its own inverse.)

Our final step F embodies the recognition of what to choose for "a" and thus we have designed and proved theorem (22).

§ 8. In [2], left- and right-conditions are defined by

(24a) (p is a left-condition) ≡ [p; true ≡ p]

(24b) (q is a right-condition) ≡ [true; q ≡ q] ;

we now focus our attention on the right-condition.

In [2], Feijen & van Gasteren point out that – because [true; x ⇐ x] for all x – definition (24b) is equivalent to

(25) (q is a right-condition) ≡ [true; q ⇒ q] ,

the right-hand side of which is formally weaker than that of (24b). Lincoln A. Wallen has taught us a greater awareness of monotonicity arguments and has urged us to notice when we used only one direction of a mutual implication. To prove that a relation is a right-condition, (25) is more convenient for the formulation of the demonstrandum than (24b).

When trying to use that a relation is a right-condition, formally stronger characterizations are, in general, to be preferred. The stronger (24b) captures $[true; x \Leftarrow x]$ – which follows from the existence of a neutral element of the composition and the latter's monotonicity –. Here I propose another strengthening of (25), one that captures composition's monotonicity (in its left argument):

(26) (q is a right-condition) $\equiv \langle \forall x :: [x;q \Rightarrow q] \rangle$.

We shall use this definition (twice) to prove the following theorem:

(q is a right-condition) $\Rightarrow$ ($\neg q$ is a right-condition)

Proof We observe for arbitrary relation q

$$
\begin{array}{ll}
 & (\neg q \text{ is a right-condition}) \\
= & \quad \{(26) \text{ with } q := \neg q\} \\
 & \langle \forall x :: [x; \neg q \Rightarrow \neg q] \rangle \\
= & \quad \{(23), \text{ i.e. right-exchange}\} \\
 & \langle \forall x :: [\sim x; q \Rightarrow q] \rangle \\
\Leftarrow & \quad \{\text{instantiation } x := \sim x\} \\
 & \langle \forall x :: [x; q \Rightarrow q] \rangle \\
= & \quad \{(26)\} \\
 & (q \text{ is a right-condition})
\end{array}
$$

(End of Proof.)

In both [1] and [2], the proof of this theorem appeals to $[\sim true \equiv true]$, whereas the above proof does not use any properties of $\sim$ or true. The manipulative disadvantage of (25) is that it contains a constant whose properties we have to use.

The heuristic significance of formulation (26) is that it almost dictates monotonicity arguments because the fact that prefixing q by an arbitrary "$x;$" weakens it, fully captures that q is a right-condition. In other words, if, for instance, we set ourselves to prove that for a right-condition q

$$[q; true; \sim q \equiv q; \sim q] \quad , \tag{27}$$

the implication sign in (26) tells us that a ping-pong argument is appropriate.

Remark When faced with a demonstrandum of the form $[A \equiv B]$ we have a demonstrandum that does not depend (anti)monotonically on either A or B. Hence, proving this directly can only be done by appealing to Leibniz's Principle, i.e. by value-preserving transformations. (We have seen an example of this in section 1.) Rewriting the demonstrandum as mutual implication $[A \Rightarrow B] \wedge [A \Leftarrow B]$ yields two conjuncts that do depend (anti)monotonically on both A and B! Because we like to avoid avoidable ping-pong arguments, it is nice to recognize circumstances under which they are indicated. (End of Remark.)

In proving (27) we observe for ping:

$$
\begin{array}{ll}
 & [q;true;\sim q \Leftarrow q;\sim q] \\
\Leftarrow & \quad \{\text{monotonicity of } ;\} \\
 & [q;true \Leftarrow q] \\
= & \quad \{[x;true \Leftarrow x]\} \\
 & true\ ,
\end{array}
$$

a demonstration in which we did not use that q is a right-condition. For pong we observe that we can strengthen the right-hand side:

$$[q;true;\sim q \Rightarrow q;\sim q]$$

$\Leftarrow$ { (26) with $x := q;true;\sim q$ }
$[q;true;\sim q \Rightarrow q;true;\sim q;q;\sim q]$
$\Leftarrow$ { monotonicity of ; }
$[\sim q \Rightarrow \sim q;q;\sim q]$
$=$ { (28) with $b := \sim q$ }
true ,

where the "seesaw lemma" – see [1] –

(28) $[b \Rightarrow b;\sim b;b]$

follows from (22) with $d,e,f := b,b,true$.
(Knowledge of the seesaw lemma admittedly helps in the choice of how to instantiate x in (26) .)

§ 9. In [1] "middle-conditions" are introduced by

(29) (c is a middle-condition) $\equiv [c \Rightarrow J]$,

where "J" is the neutral element of ";" .
Also here the constant can be eliminated from the definition: we could have defined

(30) (c is a middle-condition) $\equiv \langle \forall x :: [c;x \Rightarrow x] \rangle$

or

(31) (c is a middle-condition) $\equiv \langle \forall x :: [x;c \Rightarrow x] \rangle$.

In other words, c being a middle-condition is captured by the fact that the prefix operator "c;" and the postfix operator ";c"

are strengthening operators. The implication signs in these definitions tell us that, for instance,

(32) $[c \equiv \sim c]$ for middle-condition c

has to be proved by mutual implication. Since $[\sim x \Rightarrow x] \equiv [x \Rightarrow \sim x]$, it suffices to prove $[c \Rightarrow \sim c]$. We observe to this end

$$
\begin{array}{ll}
 & \sim c \\
\Leftarrow & \quad \{(30) \text{ with } x := \sim c\} \\
 & c ; \sim c \\
\Leftarrow & \quad \{(31) \text{ with } x := c ; \sim c\} \\
 & c ; \sim c ; c \\
\Leftarrow & \quad \{(28) \text{ with } b := c\} \\
 & c \qquad .
\end{array}
$$

Our wish to apply strengthening operators suggested to start at the consequent and the first two steps then followed. (And we were just lucky that we did not try to prove $[\sim c \Rightarrow c]$ according to the above scheme.)

§ 10. Let me conclude. We set out to explore for antisymmetric and reflexive relations the interplay between calculating with equalities and inequalities, the latter drawing our attention to monotonicity arguments. In passing we encountered a number of instances – (2), (3), (16), (26), (30), (31) –

where, sometimes to clear advantage, a formula or definition could be rewritten as a universally quantified expression.

A closely related topic is the design of calculational proofs of theorems about extreme solutions. Since this topic has been dealt with in [0] we shall not pursue it here. To close I shall borrow an example from [1] to show that also here universal quantification can be used to eliminate a constant. Instead of defining the reflexive transitive closure by

$$(s^* \text{ is the reflexive transitive closure of } s) \equiv (s^* \text{ is the strongest solution of } x: [\,] \vee s;x \Rightarrow x])$$

we can define s^* equivalently by

$$\langle \forall t :: s^*;t \text{ is the strongest solution of } x: [\,t \vee s;x \Rightarrow x] \rangle \quad .$$

We shall not repeat the standard proof here (which is, as is to be expected, a ping-pong argument). We do wish to point out that the instantiation $t := J$ returns the original definition.

Acknowledgements Anyone familiar with their writings will realize that I am greatly endebted to Rutger M. Dijkstra,

W.H.J. Feijen, and A.J.M. van Gasteren.

[0] Predicate Calculus and Program Semantics,
Edsger W. Dijkstra & Carel S. Scholten,
Springer-Verlag, 1990

[1] Relational Calculus and Relational
Program Semantics,
Rutger M. Dijkstra
1992

[2] An Introduction to the Relational
Calculus,
W.H.J. Feijen & A.J.M. van Gasteren
Appeared in:
C.S. Scholten Dedicata: Van oude
machines en nieuwe rekenwijzen,
W.H.J. Feijen & A.J.M. van Gasteren (Eds.)
Schoonhoven, Academic Service, 1991

Austin, 11 April 1993

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

A very clever and impressive (but highly intractable) proof of a beautiful theorem.

With r and s relations, the beautiful theorem reads

$$0 \wedge 1 \wedge 2 \Rightarrow 3 ,$$

where

0: r is well-founded
1: s is well-founded
2: $r \vee s$ is transitive
3: $r \vee s$ is well-founded

* * *

It was

Henk Doornbos

who communicated this theorem. Quite a number of people have tried to prove it within the relational calculus, and they all failed: except for Henk Doornbos who after a long, long while and after many serious efforts encountered a proof. Henk incorporated his proof in his forthcoming PhD-thesis and provided a heuristics. But that heuristics is so unconvincing that we leave it out here. We just present Henk's proof as one long calculation. The main characteristic of the proof is that the right thing

is done at the right moment, and that nobody knows when what is right. The reason for extracting this passage from Henk's PhD-thesis, is the problem's right of existence all by itself.

* * *

We formalize 0, 1, 2, and 3, thereby using Rutger M. Dijkstra's formulation of well-foundedness.

(0) $\langle \forall x :: [x \Rightarrow x ; r] \Rightarrow [x \Rightarrow \text{false}] \rangle$

(1) $\langle \forall x :: [x \Rightarrow x ; s] \Rightarrow [x \Rightarrow \text{false}] \rangle$

(2) $[(r \vee s) ; (r \vee s) \Rightarrow r \vee s]$

(3) $\langle \forall x :: [x \equiv x ; (r \vee s)] \Rightarrow [x \Rightarrow \text{false}] \rangle$

Explanation Relation r is well-founded means that equation $x : [x \equiv x ; r]$ has false as its weakest solution. By Knaster-Tarski, (0) is the "extremity part" of r's well-foundedness. There is little point in writing down the "solves-part" in this case.
End of Explanation.

We now prove (3) that, for any x, (3)'s consequent is true. We thereby use (0), (1), (2), and

(4) $[x \equiv x ; (r \vee s)]$,

the antecedent of (3).
Off we go.

$[\, x \Rightarrow \mathit{false} \,]$

$\Leftarrow$ { (0) }

$[\, x \Rightarrow x ; r \,]$

$\equiv$ { shunting }

$[\, x \wedge \neg(x ; r) \Rightarrow \mathit{false} \,]$

$\equiv$ { • let y be such that $[\, y \equiv x \wedge \neg(x ; r) \,]$

$[\, y \Rightarrow \mathit{false} \,]$

$\Leftarrow$ { (1) }

$[\, y \Rightarrow y ; s \,]$

$\equiv$ { Lemma 0, to be shown later :
$[\, y \equiv x ; s \wedge \neg(x ; r) \,]$
(proof uses (4)) }

$[\, x ; s \wedge \neg(x ; r) \Rightarrow y ; s \,]$

$\equiv$ { shunting }

$[\, x ; s \Rightarrow x ; r \vee y ; s \,]$

$\equiv$ { by (4), $[\, x \equiv x ; (r \vee s)^* \,]$ }

$[\, x ; s \Rightarrow x ; (r \vee s)^* ; r \vee y ; s \,]$

$\Leftarrow$ { $[x \Leftarrow y]$, from definition of y }

$[\, x ; s \Rightarrow y ; (r \vee s)^* ; r \vee y ; s \,]$

$\Leftarrow$ { • let u be such that
$[\, u ; s \Rightarrow (r \vee s)^* ; r \vee s \,]$ }

$[\, x ; s \Rightarrow y ; u ; s \,]$

$\Leftarrow$ { $(;s)$ is monotonic }

$[\, x \Rightarrow y;u \,]$

$\equiv$ { shunting }

$[\, x \wedge \neg(y;u) \Rightarrow \text{false} \,]$

$\Leftarrow$ { (0) with $x := x \wedge \neg(y;u)$ }

$[\, x \wedge \neg(y;u) \Rightarrow (x \wedge \neg(y;u));r \,]$

$\Leftarrow$ { Lemma 1, to be shown below:
$[(a \wedge \neg b);c \Leftarrow a;c \wedge \neg(b;c)]$ }

$[\, x \wedge \neg(y;u) \Rightarrow x;r \wedge \neg(y;u;r) \,]$

$\equiv$ { shunting }

$[\, x \Rightarrow y;u \vee (x;r \wedge \neg(y;u;r)) \,]$

$\equiv$ { predicate calculus }

$[x \Rightarrow y;u \vee x;r]$
$\wedge\ [x \Rightarrow y;u \vee \neg(y;u;r)]$

$\Leftarrow$ { definition of y & shunting }

$[y \Rightarrow y;u] \wedge [x \wedge y;u;r \Rightarrow y;u]$

$\Leftarrow$ { relational and predicate calculus }

$[J \Rightarrow u] \wedge [u;r \Rightarrow u]$

$\equiv$ { predicate calculus }

$[J \vee u;r \Rightarrow u]$

$\Leftarrow$ { definition of $*$ }

$[u \equiv r^*]$

$\equiv$ { • let u be such that $[u \equiv r^*]$ }

true .

And this completes the "main calculation"

The remaining obligations are to check the solvability of the requirements on u, and to prove Lemma0 and Lemma1.

Proof of Lemma0

$$
\begin{array}{ll}
 & y \\
\equiv & \quad \{\text{definition of } y\} \\
 & x \wedge \neg(x;r) \\
\equiv & \quad \{(4)\} \\
 & x;(r \vee s) \wedge \neg(x;r) \\
\equiv & \quad \{; \text{ over } \vee\} \\
 & (x;r \vee x;s) \wedge \neg(x;r) \\
\equiv & \quad \{\text{predicate calculus}\} \\
 & x;s \wedge \neg(x;r)
\end{array}
$$

End

Proof of Lemma1

For any a, b, c we have

$$
\begin{array}{ll}
 & [(a \wedge \neg b);c \Leftarrow a;c \wedge \neg(b;c)] \\
\equiv & \quad \{\text{shunting}\} \\
 & [(a \wedge \neg b);c \vee b;c \Leftarrow a;c] \\
\Leftarrow & \quad \{\text{relational calculus}\} \\
 & [(a \wedge \neg b) \vee b \Leftarrow a] \\
\equiv & \quad \{\text{predicate calculus}\} \\
 & \text{true}
\end{array}
$$

End

Re u We have to show $[r^*; s \Rightarrow (r \vee s)^*; r \vee s]$
This is the place where the transitivity of rvs, i.e. (2), will enter the game. And the only thing we use of it is

(5) $[r; s \Rightarrow r \vee s]$

We observe

$[r^*; s \Rightarrow (r \vee s)^*; r \vee s]$
$\equiv$ { • let z be such that $[z \equiv (r \vee s)^*; r \vee s]$ }
$[r^*; s \Rightarrow z]$
$\Leftarrow$ { $r^*; s$ is the strongest solution of $x: [s \vee r; x \Rightarrow x]$ }
$[s \vee r; z \Rightarrow z]$
$\equiv$ { $[s \Rightarrow z]$ }
$[r; z \Rightarrow z]$
$\equiv$ { definition of z }
$[r; (r \vee s)^*; r \Rightarrow z] \wedge [r; s \Rightarrow z]$
$\Leftarrow$ { (5) }
$[r; (r \vee s)^*; r \Rightarrow z] \wedge [r \vee s \Rightarrow z]$
$\equiv$ { right conjunct $\equiv$ true }
$[r; (r \vee s)^*; r \Rightarrow z]$
$\Leftarrow$ { $[r \Rightarrow r \vee s]$ }
$[(r \vee s); (r \vee s)^*; r \Rightarrow z]$
$\equiv$ { definition of z }
true.

End

* * *

Of course, the question is: Can we do better?

WHJ Feijen
21 January 1996