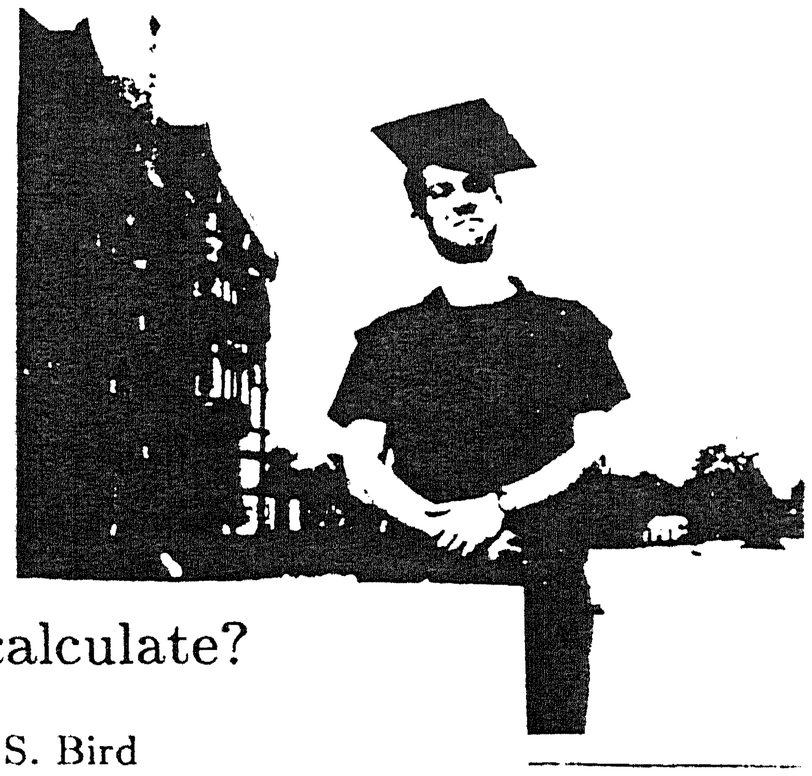
Founded 1989
Editor in chief:
Johan T. Jeuring

The Squiggolist

GRONINGEN — Categorical gang caught. The Committee for Unsquiggolic Activities dealt the subversive Categorical Movement a severe blow when a gang of members was exposed that had succeeded in masquerading as avid squiggolists. "After our successes in fighting the Diagram Chasers, we will not condone such perverted practices," the principal investigator of the CUA told our reporter. It appears that the gang members did themselves in when they categorically denied accusations of unsquiggoliceness raised by loyal supporters of the True Squiggol Spirit. The gang leader, who had taught his followers that t practice was a "natural trans mation" of the True Rites, is lieved to be still fugitive.

NIJMEGEN — Demonstration against pollution. The Association against Software Pollution (WASP) demonstrated at Toernooiveld, Nijmegen yesterday, against the pollution of specifications brought about by Nijmegen scientists. The banner on which a typical example (according to the organizers of the demonstration) of such a polluted specification had been depicted had to be carried by over fifteen people. Though it is a truth universally acknowledged that scientist have feeble constitutions, this was obviously caused by the format and weight of the banner.



Shall we calculate?

Richard S. Bird
Programming Research Group, Oxford University

OXFORD — Room temp red tion rumour. Scientists at on the oldest Squiggol Labs in Bri..... have succeeded in designing a squiggle that induces reduction at room temperatures, according to a rumour that a spokesman of Oxford University declined to confirm or deny. The rumour has caused much excitement and consternation among computer scientists. Researchers in Amsterdam were reportedly unable to repeat the experiment over a feverish weekend. As one of them said, this r cated Meta

international conference on the Mathematics of Program Construction, prof. van de Snapsheet held a revolutionary speech about a system which can derive the most efficient algorithms for all problems for which such an algorithm exists. The audience was very enthousiastic and applauded for a long time, but some of the scientists were afraid that they should be compelled to find other jobs now.

AM — Terse, terser, ew World Record was established last Friday in Amsterdam, when a formal specification of the entire Dutch Tax Code was constructed with only one so-called squiggle. Guinness officials confirmed that the record would be upheld under the new strict rules for specification terseness, which were instated after the scandal last year with the one-line three Msquiggle specification of the U.S. Tax Code in SETL. Recent simplifications of the Dutch Tax ode were not instrumental, the ppy holder of the new record aimed, which was made possible a recent breakthrough in cification technology. The allenge is now to get rid of the st squiggle.

Reducing hopeful majority

Lambert Meertens
Centre for Mathematics and Computing Science, Amsterdam

INI pers thou best

beneficial as believed. In one of the worst disasters in computing history, an automaton gave persistently wrong results during the demonstration of a new software system. The error could be traced to an invariant that had gone berserk. "Without invariants, the error would not have been as persistent as it was," one of the local professors of computing science admitted. "This may force us to reconsider our complete methodology."

continues. For the third time in one week, shopkeepers in the vicinity of Blewbury were baffled to find their shops lifted--lock, stock and barrel. Not even the foundation remained, reported one of the victims. The police admit that they have not a single clue, beyond the fact that a greedy strategy is employed

and NP

address: Centre for Mathematics and Computer Science
Dept. of Algorithmics & Architecture
P.O. Box 1079
1009 AB Amsterdam
The Netherlands
(jt@cwi.nl)

The Squiggolist, number 1, July 1989

Shall we calculate?

Richard S. Bird
Programming Research Group, Oxford University

Calculeamus!
(Leibniz)

Introduction

Suppose we want to group the red-headed members of a set of people into classes depending on their sex. Imagine that Alice, Charles and Diana have red hair, while Bill and Edward do not. One obvious procedure is first to take the subset of red-headed people and then do the grouping by sex. Thus we obtain the partition

$$\{\{\text{Alice}, \text{Diana}\}, \{\text{Charles}\}\}$$

of our set of red-headed people.

Here is another procedure: first classify the original set of people by sex, and then take the red-headed members of each group. Thus we first group the five as

$$\{\{\text{Alice}, \text{Diana}\}, \{\text{Bill}, \text{Charles}, \text{Edward}\}\}$$

and then filter each component for red-headedness.

Are these two procedures the same? Obviously yes you might think, but there is a slight discrepancy that first needs attention. Suppose Charles loses his hair. Applied again after this unfortunate occurrence, the first procedure leads to the single component partition

$$\{\{\text{Alice}, \text{Diana}\}\}$$

while the second procedure, as stated, leads to the set of sets

$$\{\{\text{Alice}, \text{Diana}\}, \emptyset\}$$

It is clear, therefore that we need to amend the second procedure by removing empty sets.

We can formalise these procedures using Bird-Meertens notation. Let $Eq(f)$ denote the function that groups its argument into equivalence classes under the function f (so, with $f = sex$, there are two possible values of f and upto two equivalence classes). Suppose p is a predicate (for example, $p = redheaded$). The equation to be proved is then:

$$Eq(f) \cdot (p\lhd) = (\neq \emptyset) \lhd \cdot (p\lhd)* \cdot Eq(f) \tag{1}$$

The question we are interested in is: can we prove (1) just by calculation, that is, by a process of equational reasoning?

Equivalence classes

Unless we give a formal definition of $Eq(f)$ the answer to the first question is clearly no. In texts on Algebra, $Eq(f)x$ is usually defined in words as the collection of equivalence classes of x under the equivalence relation $\sim_f$, where

$$a \sim_f b \Leftrightarrow f\ a = f\ b$$

Define

$$equiv\ f\ x\ a = (a \sim_f) \lhd x$$

so that $equiv\ f\ x\ a$ is the equivalence class of x with common f-value $f\ a$. We can now define

$$Eq(f)\ x = (\neq \emptyset) \lhd equiv\ f\ x * Dom\ f$$

where $Dom\ f$ is the domain of the function f. Other definitions of $Eq(f)$ will be considered below.

To prove (1) we need some subsidiary results. First:

$$\begin{array}{ll} & equiv\ f\ (p \lhd x)\ a \\ = & \text{definition of } equiv \\ & (a \sim_f) \lhd p \lhd x \\ = & \text{commutativity of } \lhd \\ & p \lhd (a \sim_f) \lhd x \\ = & \text{definition of } equiv \\ & p \lhd equiv\ f\ x\ a \end{array}$$

so, expressing this as a functional identity, we have

$$equiv\ f\ (p \lhd x) = (p\lhd) \cdot (equiv\ f\ x) \tag{2}$$

Second, we need the law

$$(p \wedge q)\lhd = p\lhd \cdot q\lhd \tag{3}$$

where $\wedge$ is interpreted as an operation on predicates rather than on propositions.

Third, we need the $(\lhd, *)$ interchange law:

$$p \lhd \cdot f* = f* \cdot (p \cdot f)\lhd \tag{4}$$

Fourth, we need the law that if $p\ (f\ a) \Rightarrow p\ a$ for all a (equivalently, if $p \cdot f = (p \cdot f) \wedge p$), then

$$p \lhd \cdot f* = p\lhd \cdot f* \cdot p\lhd \tag{5}$$

The proof of (5) is as follows:

$$\begin{array}{ll} & p\triangleleft \cdot f* \\ = & \quad (4) \\ & f* \cdot (p.f)\triangleleft \\ = & \quad \text{hypothesis} \\ & f* \cdot ((p \cdot f) \wedge p)\triangleleft \\ = & \quad (3) \\ & f* \cdot (p.f)\triangleleft \cdot p\triangleleft \\ = & \quad (4) \\ & p\triangleleft \cdot f* \cdot p\triangleleft \end{array}$$

Fifth, we need the fact that

$$(p \triangleleft x) \neq \emptyset \Rightarrow x \neq \emptyset \tag{6}$$

Finally, we need the $*$-distributivity law:

$$(f \cdot g)* = (f*) \cdot (g*). \tag{7}$$

The proof of (1) is now:

$$\begin{array}{ll} & (Eq(f) \cdot (p\triangleleft))\ x \\ = & \quad \text{definition of composition} \\ & Eq(f)(p \triangleleft x) \\ = & \quad \text{definition of } Eq \\ & (\neq \emptyset) \triangleleft equiv\ f\ (p \triangleleft x) * Dom\ f \\ = & \quad (2) \\ & (\neq \emptyset) \triangleleft ((p\triangleleft) \cdot (equiv\ f\ x)) * Dom\ f \\ = & \quad (7) \\ & (\neq \emptyset) \triangleleft (p\triangleleft) * (equiv\ f\ x) * Dom\ f \\ = & \quad (5) \text{ and } (6) \\ & (\neq \emptyset) \triangleleft (p\triangleleft) * (\neq \emptyset) \triangleleft (equiv\ f\ x) * Dom\ f \\ = & \quad \text{definition of } Eq \\ & (\neq \emptyset) \triangleleft (p\triangleleft) * Eq(f)\ x \\ = & \quad \text{definition of composition} \\ & ((\neq \emptyset) \triangleleft \cdot (p\triangleleft)* \cdot Eq(f))\ x \end{array}$$

Exercises

If you thought the above derivation was rather long, try the following ones. Both stem from different definitions of $Eq(f)$. In the first, we define

$$Eq(f)\ x = (\neq \emptyset) \triangleleft equiv\ f\ x * x$$

This is similar to the first definition, but $Dom\ f$ has been replaced by the set x (the reader may have wondered why this was not done originally). The third definition of $Eq(f)\ x$ is

as the coarsest (i.e. smallest size) partition of x with the property that every component in the partition consists of elements with the same f-value. This definition can be expressed formally as:

$$Eq(f) = \sqcap_{\#} / \cdot \text{all } q_f \lhd \cdot \text{Inv}(\cup /)$$

where

$$q_f\ x = (\#(f * x) = 1)$$

and

$$\text{Inv}(f)\ y = \{x | f\ x = y\}$$

Now, for each of the new definitions, prove (1). Alternatively, prove the equivalence of the three definitions of $Eq(f)$. Of course, you are only allowed calculation. I wish you luck.

Reducing hopeful majority

Lambert Meertens

We are given a non-empty bag of (votes on) 'candidates', and are asked to determine if some candidate has the majority.

Several derivations of linear-time algorithms have been given, all of which work in two phases: first find a 'hopeful majority' candidate, and next check if it really has the majority. A 'hopeful majority' candidate is any candidate c in the bag such that if some candidate has the majority, it is c.

I consider here only the problem of finding some hopeful majority candidate. All previous algorithms I have seen basically scan the bag. The purpose of this note is to show that there is a divide-and-rule approach. In a previous note I have given a derivation, mainly based on predicate calculus. Here only the solution is presented.

Let C stand for the type of the candidates, and N for the naturals. The operation $\oplus: (C \times N) \times (C \times N) \to C \times N$ is defined by:

$$(c0, d0) \oplus (c1, d1) \;=\; (c0, d0 + d1) \lhd c0 = c1 \rhd ((c0, d0 - d1) \uparrow_{\pi_2} (c1, d1 - d0)) \quad ,$$

where $a \lhd p \rhd b$ stands for **if** p **then** a **else** b **fi**. We also define $f: C \to C \times N$:

$$f \bullet c \;=\; (c, 1) \quad .$$

Now a hopeful majority candidate is determined by $\pi_2 \bullet h$, where h is the bag homomorphism defined by

$$h \;=\; \oplus/ \bullet f\bullet \quad .$$

However, something funny is going on here. Even under the hot indeterminate interpretation of $\uparrow_{\pi_2}$ the operator $\oplus$ is not associative. This can be seen by considering the different ways to compute h on a bag of three distinct candidates. According to the current definitions this would mean that h is not a proper bag homomorphism. Associativity is not required for consistency if we consider the bag splitting itself also as indeterminate. This is already mentioned in the *Algorithmics* paper, but I had not come across a clear (non-contrived) example before. The definition of the indeterminate bag reduce $\oplus/$ is then that it is the 'thinnest' (most determinate) indeterminate function r satisfying

$$r \bullet \tau \rightsquigarrow c \quad ;$$

$$r \bullet \uplus \rightsquigarrow \oplus \bullet r \| r \quad .$$