South African Computer Journal
Number 22
March 1999

Suid-Afrikaanse Rekenaar-tydskrif
Nommer 22
Maart 1999

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarvereniging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

World-Wide Web: `http://www.cs.up.ac.za/sacj/`

Subscriptions

	Annual	Single copy
Southern Africa	R80.00	R40.00
Elsewhere	US$40.00	US$20.00

An additional US$15 per year is charged for airmail outside Southern Africa

to be sent to:

Computer Society of South Africa
Box 1714, Halfway House, 1685
Phone: +27 (11) 315-1319 Fax: +27 (11) 315-2276

WOFACS 98
IFIP WG2.3 and UNU/IIST
WINTER SCHOOL on PROGRAMMING METHODOLOGY
University of Cape Town, 6-17 July 1998

Chris Brink

Laboratory for Formal Aspects and Complexity in Computer Science, Department of Mathematics and Applied Mathematics, University of Cape Town

Since 1992 there has been a biennial Winter School on Formal and Applied Computer Science (WOFACS) at the University of Cape Town. Each of these have resulted in a *Proceedings* volume: *SACJ* **9**, **13**, **19**, and the volume you hold in your hand right now. All WOFACS events have had more or less the same structure. A group of eminent academics come to Cape Town during the winter vacation, and each of them offers, over a 2-week period, a course of 10 lectures on a particular topic. These short courses are pitched at about Honours level, and have some evaluation mechanism built in: short tests, or exercises, or assignments. At a student's request, and by arrangement with the Head of Department at his/her home institution, these courses can then be offered as part of the student's Honours degree. In this way WOFACS makes a contribution to beginning postgraduate studies on a wide geographical front. Typically such an event would attract students and young staff members not only from across South Africa, but also from many sub-Saharan African countries. Each WOFACS was organised by the UCT Laboratory for Formal Aspects and Complexity in Computer Science (FACCSLab).

WOFACS 98 had a distinctly international flavour. The entire event was, in fact, three things at once (which explains the somewhat complicated title at the top of this page). Besides being, by our reckoning, the 4th WOFACS, it was also the third in a series of outreach offerings of IFIP Working Group 2.3 on Programming Methodology. All the speakers were from WG2.3, and the entire event was built around the topic of programming methodology. Thirdly, the event was also an offering of the International Institute for Software Technology, situated at the United Nations University in Macao. The three role players, FACCSLab, WG2.3 and UNU/IIST, shared an agenda of trying to service specifically possible participants from disadvantaged communities and other African countries, and the UNU/IIST made available some generous grants for this purpose. In keeping with the tradition of these events there were no course fees: except for a fairly modest registration charge WOFACS has always been a free service to the community.

The speakers at WOFACS 98 and their topics were:

- Prof Dines Bjørner, Technical University Denmark: *Domains and Requirements, Software Architectures and Program Organisation.*
- Prof David Gries, Cornell University: *Logic as a Tool.*
- Prof Michael Jackson: *Problem Frames and Principles of Description.*
- Prof Jayadev Misra, University of Texas: *Toward an Applied Theory of Concurrency.*
- Dr Carroll Morgan, Oxford: *Predicate Transformers and Probabilistic Programs.*

Professor Gries' course was regarded as foundational, and recommended to all participants. It was offered during the first week only, at double tempo, thus giving the other four courses the opportunity to make use of concepts and techniques introduced there. Each of the speakers made available a Course Reader of their material. These were printed and bound before the event, and were handed out to participants upon registration. WOFACS 98 was attended by more than 60 participants, inter alia from Angola, Malawi, the Congo, Gabon, Cameroon, Malawi and Uganda.

From South African Universities we had representation, besides UCT, also from the University of Stellenbosch, the Transkei, the Qwa-qwa branch of the University of the North, the Witwatersrand, Pretoria, RAU, the North-West, Vista, UNISA, the Mangosothu Technikon and the Eastern Cape Technikon. Cape Town weather can be pretty stormy in July, but there were sufficiently many beautifully clear winter days to allow participants the opportunity to do some

sightseeing and exploring, after lectures or over the weekend.

We are grateful to our financial sponsors, and pleased to acknowledge their contributions.

- UNU/IIST.
- The Foundation for Research Development.
- The Chairman's Fund of Anglo American.

We thank the University of Cape Town for the use of its premises and facilities. We are particularly grateful to the 5 speakers, who put a lot of thought and preparation into their lectures, and tackled with great success the difficult task of conveying state-of-the-art material to a heterogeneous audience. It is only fair that specific thanks should be given to Carroll Morgan, whose idea it was in the first place to have a combined event, and to Dines Bjorner, who kickstarted the fundraising campaign. Finally, I would like to add my personal thanks to my colleagues and staff who worked so hard behind the scenes to make a success of WOFACS 98.

Teaching Math More Effectively, Through the Design of Calculational Proofs*

David Gries[a] Fred B. Schneider

Computer Science, Cornell University

[a] *Supported by Darpa under ONR grant N00014-91-J-4123.*

[b] *Supported by ONR under contract N00014-91-J-1219, AFOSR under proposal 93NM312, NSF under grant CCR-8701103, and DARPA/NSF grant CCR-9014363.*

1 Introduction

Lower-level college math courses usually teach informal methods, presumably because students can't handle formalism. Neophytes need a gentle introduction to math, it is felt. Later, perhaps, when they have mastered definitions and proofs written largely in English, they can be taught a bit more about rigor and formalism. The design of proofs is also not taught, because it is difficult. Sure, students see some proofs, and they may be asked to develop a few themselves, but they are not ready for principles and strategies for designing proofs.

But there is general dissatisfaction with the current situation. Even after several math courses, students have inadequate reasoning abilities, many students still fear math, math notation, and rigor, and the development of proofs remains a mystery, a black art.

We conjecture that a good part of the reason for this state of affairs is the reliance on informality and English and the lack of rigor. A case in point is the proof given in Table 1, which is taken from a math text. In order to recognize how bad this proof is, let's investigate the qualities of a good proof.

A proof of a theorem should provide evidence for belief in the validity of the theorem, where the evidence consists of the facts (e.g. previously proved theorems) on which it rests and on how these facts interact to convince. We understand a proof when we understand which facts are used and how they interact. Understanding also implies the ability to explain the proof to others and perhaps to prove other theorems with similar proofs.

Now look at the proof in Table 1. It does not state the facts on which it rests. (For example, it says, "If $y \notin A$, then, since $y \in A \cup B$ we must have $y \in B$", but there is no reference to the theorem that explains why this inference holds.) Also, it is difficult to see precisely how the facts interact —the sequence and subsequences of inferences and all the case analyses in the proof cannot be easily digested and then explained to others. Finally, having seen this proof, students have difficulty proving similar theorems. In fact, this proof yields little insight into its development —how did it arise?

And yet, in spite of its inadequacies, this proof (and others like it) is held up as a model for students to emulate.

2 A more effective approach

We believe that mathematics can be taught more effectively by first teaching the design of rigorous proofs using a formal logic. The result will be a better educated student —a student who is not so afraid of math, who is able to handle mathematical notation, who can reason more effectively, and who, therefore, can learn later material more easily.

Of course, students have to be convinced that the use of formalism is actually helpful. The rigorous approach of logic should help them solve problems they cannot easily solve without it, and a suitable use of logic should not lead to overwhelming complexity of detail.

In our experience, for this purpose, a *calculational* logic is best. A sample proof in our calculational propositional logic is given in Table 2. Equivalence $\equiv$ is treated associatively, so that the theorem in the table can be viewed either as $(p \vee q \equiv p \vee \neg q) \equiv p$ or as $p \vee q \equiv (p \vee \neg q \equiv p)$. Use of associativity of equivalence helps reduce formal detail immensely. Also, symbol $=$ is used for equality over any type, including type boolean. Symbol $=$ is used conjunctionally: $b = c = d$ is equivalent to $b = c \wedge c = d$.

Each step of the proof has the form

$$\begin{array}{ll} & E[v := P] \\ = & \langle\, P = Q \,\rangle \\ & E[v := Q] \end{array}$$

Such a step shows equality of two formulas using the rule of substitution of equals for equals. The hint between the two formulas shows the equality being used in the substitution ($E[v := P]$ denotes expression E with every free occurrence of variable v replaced by expression P). (In a text, each hint would contain the number of the axiom or theorem being referenced.) Transitivity of equality allows us to conclude that the first and last formulas of the proof of Table 2 are equal.

A theorem of the logic is either an axiom or a theorem that is proved equal to an already-existing theorem. Also, we have a metatheorem: To prove $P \equiv Q$ it suffices to transform P into Q (or Q into P) as shown in Table 2.

Calculational propositional logic, along with preliminaries (e.g. the definition of textual substitution) can be taught in four weeks. Students will see many proofs and will develop many themselves, in the rigid proof style illustrated in Table 2. They will also learn a few strategies and principles to be used in designing proofs. Here are two strategies. (1) In proving a theorem of the form $P \equiv Q$, transform the more complicated of P and Q into the simpler one. (2) Use the shape of the formulas to guide the development. For example, the shape of the formula in the first line of Table 2 cries out for simplification using distribution of $\vee$ over $\equiv$. As students develop a skill in proving theorems, they learn that attention to rigor may be a simplifying force —and not an onerous burden.

One can also discuss the connection between typical informal presentations of proofs and propositional logic, thus providing some understanding for informality as well. For example, proof by contradiction is based on the theorem $p \equiv \neg p \Rightarrow \mathit{false}$.

A key in making rigor and formalism palatable is to keep the notation simple and consistent and to explain every new notation (as well as give rules for manipulating it). It is this attention to basic detail that helps students lose their fear of mathematics. For example, traditionally, students are not shown rules for manipulating summations like $\Sigma_{i=1}^{3} i^2$ —in fact, even the syntax may be unexplained. Also, students are confused by the many different notations for quantification —see the left column of Table 3. Presentation of a single notation for all quantifications eliminates this confusion and provides an elegant uniformity. For any operator $\star$ that is associative, is symmetric, and has an identity, the notation[1]

$$(\star i \mid R.i : P.i)$$

denotes the "accumulation" using operator $\star$ of the values of expression $P.i$ over all values of i that satisfy range-predicate $R.i$. For example, Table 3 gives the conventional notation and our notation for three different quantifications. With a single notation, one can discuss issues of scope, free occurrences of variables, and bound occurrences of variables for all quantifications, once and for all. One can also present general axioms and theorems for manipulating such quantifications.

Thereafter, pure predicate logic is easy to introduce. Operators $\wedge$ and $\vee$ are associative, are symmetric, and have identities, so $(\wedge i \mid R.i : P.i)$ and $(\vee i \mid R.i : P.i)$ makes sense; the first is universal quantification and the second is existential quantification. A few axioms specific to these two quantifications can be introduced.

Once logic and proof have been thoroughly presented, other topics can be discussed —e.g. set theory, a theory of integers, and mathematical induction. Each topic can be presented in the same rigorous manner, by extending the logic: give axioms to characterize new operators and build up a library of theorems. For example, set theory can be introduced by adding to pure predicate logic the axioms that characterize set comprehension and the set operators. In this manner, the notions of proof and proof style become the unifying force in the course, the glue that binds together arguments in all domains.

Table 4 contains a calculational proof of distributivity of union over intersection. In contrast to the proof of Table 1, this proof has all the good qualities mentioned earlier. It refers to the facts it uses (e.g. the definition of $\cup$). Its structure is simple. And it uses a strategy that is used over and over in mathematics: To prove something about operators (here, $\cup$ and $\cap$), eliminate them using their definitions, perform some manipulation, and reintroduce the operators.

3 Rigor provides crisper explanations

Attention to rigor and formal detail can provide a measure of clarity that is impossible to obtain with an informal approach. For example, one can rigorously prove the metatheorem that a formula P is a theorem iff the formula $(\forall x \mid : P)$ is a theorem. Using this metatheorem, one can discuss the different ways in which theorems might be expressed. Thus, the following three statements are equivalent. In the first, it is assumed informally that a and b are integers —perhaps this is mentioned in the accompanying text; in the second, the type is given informally; in the third, the type is made formally explicit.

$$a+b=b+a$$

$$a+b=b+a \qquad (\text{for } a,b \text{ integers})$$

$$(\forall a,b:\mathbb{Z} \mid : a+b=b+a)$$

As another example where formalism can clarify, consider proving $b^{m+n} = b^m \cdot b^n$, for n,m natural numbers, by mathematical induction. Without a well-explained notation for quantification and the rules for manipulating it, no amount of informal explanation will clarify for students the different roles of m and n in the proof. Formally, however, this formula is expressed as $(\forall m,n \mid 0 \le n \wedge 0 \le m : b^{m+n} = b^m \cdot b^n)$, which can be rewritten (using an axiom of quantification and the ability to name a formula) as

$$(\forall n \mid 0 \le n : P.n)$$

$$\text{where } P.n: \ (\forall m \mid 0 \le m : b^{m+n} = b^m \cdot b^n) \ .$$

Now it is clear that n will be the "induction variable" of an inductive proof and that the induction hypothesis is a universal quantification over m.

Also, an early careful study of quantification makes many proofs by induction easier. For example, without knowing the rules for manipulating summations, the proof by induction of $n^2 = (\Sigma i \mid 1 \le i \le n : 2 \cdot i - 1)$ can be difficult.

Finally, since the students understand quantification, one can prove the following quite easily —using a calculational proof. Let U be a set and $\prec$ a binary relation over U. Then $(U, \prec)$ admits induction iff $(U, \prec)$ is well founded. This theorem, which is rarely mentioned in informal presentations of induction, provides a deeper un-

[1] Bound variable i can be annotated with a type to indicate the range of values it may assume. A discussion of types is outside the scope of this article.

derstanding of induction.

4 Discussion

The rigorous approach to teaching math has not, as yet, been accepted. Two criticisms of it are heard frequently: (1) students can't handle rigor and formalism, and (2) teaching syntactic manipulation impedes the understanding that a more semantic and informal approach provides.

Our own experience belies the first criticism; in fact, the criticism should go the other way. Teaching mathematics through informalism is like driving in a fog. One sees dim figures in the distance, and every once in a while some of them suddenly appear clearly, but usually everything is veiled and mysterious. It's dangerous to drive in the fog, and one has to drive slowly; even so, one is not always sure where one is. Teaching rigor and precision, provided it is done without the veil of complexity interfering, burns away the fog, leaving everything crisp and clear and making it possible to drive faster.

We reply to the criticism concerning semantic versus syntactic reasoning as follows. An informal proof, like that in Table 1, can be translated into a proof in a natural-deduction or Hilbert logic. The resulting proof is every bit as syntactic as the calculational proof of Table 4. Thus, the English proof is simply an informal version of a syntactic proof —and, as we have seen, a poor one at that. The informal proof has no more meaning or semantics than the calculational proof.

Perhaps criticism concerning semantics arises because formal statements are sometimes difficult to understand. However, presenting a formal definition or theorem does not preclude giving alternative views as well. For example, a presentation of the axiomatic definition of set union can be supplemented with a Venn diagram, an English description, and an informal notion of evaluation. Nevertheless, it should be realized that for purposes of reasoning —constructing proofs— it is the axiomatic definition that is important. In fact, the axiomatic definition should be viewed as encoding all the meaning of the object being defined.

New ideas in teaching are often met with inertia. People don't like changing what they have been doing for a long time —especially if it requires them to change their way of thinking. And our approach does require a change how one thinks of mathematics and proof. However, because of the general ineffectiveness of current teaching methods in turning on students to math, alternatives should be seriously considered. The rigorous approach of calculational proofs bears looking into by all who want to teach mathematics effectively.[2]

[2]The author's 500-page text, *A Logical Approach to Discrete Math* (Springer Verlag, NY, 1993), uses the approach described in this article in teaching the usual topics in discrete math —logic, set theory, a theory of integers, induction, functions and relations, combinatorics, solving recurrence relations, modern algebra, and graph theory. The 300-page Instructor's Manual contains four other essays concerning the approach, as well as answers to the exercises. The Instructor's Manual also provides evidence of the success of our approach. Together, the text and Instructor's Manual contain over 700 calculational proofs, most of which are short and simple. Contact the authors (gries@cs.cornell.edu) to obtain the Instructor's Manual.

Table 1: Conventional Proof of $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

We first show that $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$. If $x \in A \cup (B \cap C)$, then either $x \in A$ or $x \in B \cap C$. If $x \in A$, then certainly $x \in A \cup B$ and $x \in A \cup C$, so $x \in (A \cup B) \cap (A \cup C)$. On the other hand, if $x \in B \cap C$, then $x \in B$ and $x \in C$, so $x \in A \cup B$ and $x \in A \cup C$, so $x \in (A \cup B) \cap (A \cup C)$. Hence, $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$.

Conversely, if $y \in (A \cup B) \cap (A \cup C)$, then $y \in A \cup B$ and $y \in A \cup C$. We consider two cases: $y \in A$ and $y \notin A$. If $y \in A$, then $y \in A \cup (B \cap C)$, and this part is done. If $y \notin A$, then, since $y \in A \cup B$ we must have $y \in B$. Similarly, since $y \in A \cup C$ and $y \notin A$, we have $y \in C$. Thus, $y \in B \cap C$, and this implies $y \in A \cup (B \cap C)$. Hence $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$. The theorem follows.

Table 2: Proof of $p \vee q \equiv p \vee \neg q \equiv p$

$$
\begin{array}{ll}
 & p \vee q \equiv p \vee \neg q \\
= & \quad \langle \text{Distr. of } \vee \text{ over } \equiv,\ p \vee (q \equiv r) \equiv p \vee q \equiv p \vee r \rangle \\
 & p \vee (q \equiv \neg q) \\
= & \quad \langle \neg q \equiv q \equiv \mathit{false} \rangle \\
 & p \vee \mathit{false} \\
= & \quad \langle \text{Identity of } \vee,\ p \vee \mathit{false} \equiv p \rangle \\
 & p
\end{array}
$$

Table 3: A Uniform Notation for Quantification

Conventional notation	Uniform notation
$\Sigma_{i=1}^{3} i^2$	$(+i \mid 1 \leq i \leq 3 : i^2)$
$(\forall x).1 \leq x \leq 3 \Rightarrow b[x] = 0$	$(\wedge x \mid 1 \leq x \leq 3 : b[x] = 0)$
$(\exists x).1 \leq x \leq 3 \wedge b[x] = 0$	$(\vee x \mid 1 \leq x \leq 3 : b[x] = 0)$

Table 4: Calculational Proof of $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Below, we prove that $v \in A \cup (B \cap C) \equiv v \in (A \cup B) \cap (A \cup C)$. By Extensionality (the definition of equality of sets), $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

$$
\begin{array}{ll}
 & v \in A \cup (B \cap C) \\
= & \quad \langle \text{Definition of } \cup \rangle \\
 & v \in A \vee v \in B \cap C \\
= & \quad \langle \text{Definition of } \cap \rangle \\
 & v \in A \vee (v \in B \wedge v \in C) \\
= & \quad \langle \text{Distr. of } \vee \text{ over } \wedge \rangle \\
 & (v \in A \vee v \in B) \wedge (v \in A \vee v \in C) \\
= & \quad \langle \text{Definition of } \cup, \text{ twice} \rangle \\
 & (v \in A \cup B) \wedge (v \in A \cup C) \\
= & \quad \langle \text{Definition of } \cap \rangle \\
 & v \in (A \cup B) \cap (A \cup C)
\end{array}
$$

Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications of Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript
Manuscripts for *review* should be prepared according to the following guidelines:

- Use wide margins and $1\frac{1}{2}$ or double spacing.
- The first page should include:
- the title (as brief as possible)
- the author's initials and surname
- the author's affiliation and address
- an abstract of less than 200 words
- an appropriate keyword list
- a list of relevant Computing Review Categories
- Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text according to the Harvard. References should also be according to the Harvard method.

Manuscripts accepted for publication should comply with guidelines as set out on the SACJ web page,

`http://www.cs.up.ac.za/sacj`

which gives a number of examples.

SACJ is produced using the LaTeX document preparation system, in particular LaTeX 2_ε. Previous versions were produced using a style file for a much older version of LaTeX, which is no longer supported. Please see the web site for further information on how to produce manuscripts which have been accepted for publication.

Authors of accepted publications will be required to sign a copyright transfer form.

Charges
Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30.00 per final page for contributions which require no further attention. The maximum is R120.00, prices inclusive of VAT.

These charges may be waived upon request of the author and the discretion of the editor.

Proofs
Proofs of accepted papers may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Letters and Communications
Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words. Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book Reviews
Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement
Placement of advertisements at R1000.00 per full page per issue and R500.00 per half page per issue will be considered. These charges exclude specialised production costs, which will be borne by the advertiser. Enquiries should be directed to the editor.