

QI QUÆSTIONES INFORMATICÆ

Volume 6 • Number 4

April 1989

D G Kourie	Editorial	137
------------	-----------	-----

VIEWPOINTS and COMMUNICATIONS

B H Venter	Reflections on the Nature and Future of Computer Science in Southern Africa	139
------------	---	-----

MSc/PhD	Abstracts: MSc/PhD Conference held at Dikhololo in 1988	143
---------	---	-----

RESEARCH ARTICLES

P Machanick	Software Design to Meet Third World Requirements: An Experimental Software Engineering Approach	153
-------------	---	-----

G R Finnie	A "Cooperating Expert's" Framework for Business Expert System Design	162
------------	--	-----

P M Q Lay C R Atkinson	The Application of Scientific Method to Information Systems Analysis	169
---------------------------	--	-----

D G Kourie	An Approach to Defining Abstractions, Refinements and Enrichments	174
------------	---	-----

The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

QUÆSTIONES INFORMATICÆ

The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

Editor

Professor D G Kourie
Department of Computer Science
University of the Pretoria
Hatfield
0083

Production

Mr Q H Gee
Department of Computer Science
University of the Witwatersrand
Johannesburg
Wits
2050

Subscriptions

The annual subscription is

	SA	US	UK
Individuals	R20	\$ 7	£ 5
Institutions	R30	\$14	£10

to be sent to:

*Computer Society of South Africa
Box 1714 Halfway House 1685*

Quæstiones Informaticæ is prepared by the Computer Science Department of the University of the Witwatersrand and printed by Printed Matter, for the Computer Society of South Africa and the South African Institute of Computer Scientists.

Editorial

by

Derrick Kourie

It is my privilege to have been requested by the SAICS executive to take over the post of editor of *QI* from Professor Judy Bishop. I think it is in order to thank her on behalf of the readership for the fine job she has done in boosting the quality of the journal during her brief but effective term. It is also appropriate to thank the production editor, Quintin Gee, for his substantial role in producing the journal. I am grateful that he is still in the post, and for all the support and work that he continues to do.

My job as editor is directed towards the overall goal of serving the South African academic community in the various computer-related disciplines in particular, and the computer industry in general. A number of objectives which support this goal include

- ensuring that high quality papers are published, thereby providing a display window for computer-related research in South Africa
- boosting local and international circulation of the journal both within the academic community and in the computer industry at large, thereby promoting a fruitful interchange of ideas
- attempting to do this in a cost-effective fashion so that the limited financial resources of SAICS and the CSSA may be released (perhaps even modestly augmented) to promote their various other service-orientated activities.

A number of measures are planned which are intended to meet these objectives. I shall mention some of them below, while others will become manifest with the passage of time.

After much debate it has been decided to change the name of this journal from *Quæstiones Informaticæ* to *The South African Computer Journal/Die Suid-Afrikaanse Rekenaartydskrif*. It will be abbreviated to *SACJ* in English and *SART* in Afrikaans. Arguments against this name change include the conciseness and uniformity of reference in both official languages provided by *QI*, and a certain kind of catchiness to the name. Those in favour of the name change regard the new proposal as being more descriptive for ordinary mortals (i.e. non-Latin scholars), less pretentious, and therefore more inviting for a wider audience. The fact that the new title identifies the journal as South African is also regarded as important. Many readers would, I surmise, be fairly neutral about the name and adopt a philosophical "a rose by any name" position. Perhaps the divide is between those who opt for a high level of abstraction and information hiding, and

those who feel that a measure of refinement is necessary.

Regarding the quality of papers, I shall continually strive to ensure that papers submitted are reviewed by at least two relevant and competent specialists. It is appropriate here to thank all those who have so enthusiastically reviewed papers to date. This is a time-consuming, altruistic, backroom task, with very little explicit reward. To ensure that the burden is spread more equitably, I would like to appeal to readers to suggest additional names of people who could be approached for reviewing. Names of overseas contacts would be particularly useful.

I should also like to invite as much reader-participation in the journal as possible. There are several levels at which this may be done. The most obvious is by way of letters to the editor. Many people out there have strong ideas about a variety of subjects. In the absence of a decent national network facility (perhaps someday!), please feel free to use *SACJ* as your soapbox.

However, it is also evident that many people read many books for a variety of purposes. Why not share these insights by submitting book reviews to the journal, particularly with respect to books which could be prescribed for courses? If there are any book publishers or distributors out there who perchance may read this editorial, perhaps you should make inspection copies to lecturers contingent on a review being provided to *SACJ*!

I would also encourage researchers to continue providing a steady stream of research papers to the journal. Clearly, *SACJ* is in competition with other international journals for your research results. However, this is not a head-on competition. While it would be sheer hubris to pretend that *SACJ* is precisely equivalent to one of the more prestigious overseas publications, there are considerations which argue in favour of submitting certain kinds of research to *SACJ*. First, *SACJ* will be dedicated to providing a quick turnaround in reviewing and publication. Hence, it is an ideal forum for presenting and testing interim research results, and even for quickly assuring your stamp on potentially important ideas which you hope to flesh out later. Secondly, *SACJ* is the obvious forum to use for locally relevant research. Finally, and quite candidly, the competition for publication in *SACJ* is obviously not as intense as in a more prestigious international journal. However, I need to be most

explicit on the implications of this latter point.

SACJ should not be seen as a soft option in the sense that quality will be sacrificed. By this I mean that on some arbitrary scale of quality measurement, if CACM contains papers above say the 95% percentile, then SACJ should fall into about a 60% percentile category. Put differently, there is clearly a gap to be filled that lies somewhere between poor, inferior drivel and outstanding research contributions – a gap which SACJ will seek to fill. Papers will therefore be rigorously reviewed, and every effort will be made to ensure that the journal is worthy of international recognition – even if such recognition does not come about immediately. This is not the impossible task that some might consider it to be. There are several South African scientific journals that already enjoy a measure of international recognition (the South African Statistical Journal – to name but one). Furthermore, it is my perception that many of our academics who travel overseas discover – perhaps slightly to their amazement – that they are well able to hold their own with academics at peer institutions. This suggests that there is probably sufficient brain power, research ability and research activity in the country to ensure that the

goal of international recognition is attained.

As for the cost-effective functioning of SACJ, two points need to be made. First, SACJ will be available for a limited amount of advertising at R1000 per page and R500 per half-page. The computer industry and book publishers might wish to avail themselves of this offer, as might universities and employment agencies. Enquiries in this regard should be directed to Quintin Gee. Secondly, a modest charge per page (indicated elsewhere in this edition) will be levied on accepted research papers. This has become standard practice for most journals, the rationale being that the SACJ is one of the journals which counts for state subsidy purposes. However, the editor will have the right to waive such charges in deserving cases, as for example in the case of an author from industry whose company is unwilling to provide the financial support.

Ultimately then, SACJ will critically depend on your support. It will become what you, the reader, researcher and reviewer, make it. In a sense the South African Computer Journal will expose you, the South African Computer Academic, to the outside world without a single Latin phrase to hide behind.

An Approach to Defining Abstractions, Refinements and Enrichments

D G Kourie

Department of Computer Science, University of Pretoria, Pretoria 0083

Abstract

A proposal for defining abstractions and refinements is given in terms of three-valued logic applied to a domain of discourse consisting of a property and an entity set. Definitions for several related concepts flow naturally from these, including possible orderings on refinements and abstractions, as well as the notions of non-determinism, enrichment and base abstractions.

Keyword: abstraction, refinement, enrichment, three-valued logic, non-determinism.

Computing Review Categories: D.2.1, D.2.8, D.3.1, F.3.0, I.2.6

Received September 1988, Accepted January 1989

1. Introduction

Most software development methods proposed during the last two decades or so depend in one way or another on the idea of stepwise refinement of abstractions. The notion of abstraction also comes to the fore in the Artificial Intelligence context [1,2,3].

Often the concepts of abstraction and refinement are referred to in the literature in an intuitive and/or imprecise way. In other cases definitions, sometimes differing from one author to the next, have been proposed. Such proposals are, however, either limited to a specific context (such as the development of sequential programs), or a definition is given for one of the concepts but not for the other. It would seem, therefore, that there is a need to develop a unified formal theory of abstractions and refinements so as to sharpen and shape the ideas of those who are involved in software development and in the practice of AI. Towards this end, the present paper proposes generic definitions for these concepts, and for several concepts directly associated with these. In particular, the notion of enrichment seems to belong naturally with that of refinement. However, the emphasis is primarily, but not exclusively on software development, rather than on AI.

The next section gives the basic model within which the definitions are made. The definitions of abstractions, refinement and enrichment are given in section 3 as well as a number of corollaries that follow from these definitions. Section 4 is devoted to possible orderings which arise by virtue of the definitions. The final section briefly discusses some of the more important literature that relates to the present work.

2. Basic Model

2.1 Notation and Conventions

The following notation is used :

$X \cup Y$: the union of sets X and Y

ϕ : the empty set

iff : if and only if

$\vdash P$: Predicate P is true

$P \rightarrow Q$: Predicate Q logically follows from predicate P

Atomic formulae are predicates or well-formed formulae (wff's) in first order predicate calculus without connectives. A literal is either an atomic formula or an atomic formula preceded by a negation symbol. A clause is the disjunction of one or more literals (cf. Nilsson [9]).

Where it is clear from the context, a set of clauses will denote their conjunction, i.e. $\{P_i / P_i$ a clause, $i = 1..n\}$ denotes the predicate $P_1 \& P_2 \dots \& P_n$.

2.2 Domain of Discourse

The context for the discussion below is a *domain of discourse* involving entities and their associated properties. A domain of discourse is represented by a pair of sets : an entity set and a property set respectively. The pair is denoted by $\langle Ent, Prop \rangle$. Apart from being non-empty, no assumption about the cardinality of these sets is made.

Each $E_i \in Ent$ is a constant symbol which is interpreted as a distinct *entity* of interest in the domain of discourse.

Each $p_j(E) \in Prop$ is a clause in which every literal contains the free variable, E , as one of its arguments. E represents some uninstantiated entity in Ent . An interpretation is assumed that identifies each literal of each clause $p_j(E)$ (and therefore each clause as a whole) as a *property* of the uninstantiated entity E . For example, the

appearance of the free variable E in a literal such as $colour(E, blue)$ is construed to mean that the uninstantiated entity E has the colour blue.

When E is substituted by entity $E_i \in Ent$, the resulting clause is denoted by $p_j(E_i)$. $p_j(E_i)$ is called a *property assertion* of entity E_i . It is a clause that asserts that the specific entity E_i possesses the property alluded to in $p_j(E)$.

In general, only properties which are considered directly relevant in the domain of discourse are included in *Prop*. However, it will be assumed that if a clause is considered relevant then each separate literal in the clause together with its negation is also in *Prop*.

An interpretation of property assertions in the domain of discourse is assumed to exist in the form of a total function :

$$I : Ent \times Prop \rightarrow \{true, false, unknown\}.$$

The notation $\vdash p_j(E_i)$, or $\vdash \sim p_j(E_i)$, or $?p_j(E_i)$ will be used for $I(E_i, p_j(E))$, depending on whether $I(E_i, p_j(E))$ has value *true* or *false* or *unknown* respectively. It is assumed that the interpretation I is consistent with the normal (strong) truth tables that apply in 3-valued logic [6]. Hence, for example, the interpretation of a property assertion cannot be that it has truth value of false or unknown if the interpretation of one of its disjuncts is that it has a truth value of true.

Each entity, E_i , thus partitions *Prop* into three sets called the true, unknown or false property sets of E_i , which are respectively denoted by:

$$true(E_i) = \{p_j(E) / p_j(E) \in Prop \ \& \ \vdash p_j(E_i)\}$$

$$unknown(E_i) = \{p_j(E) / p_j(E) \in Prop \ \& \ ?p_j(E_i)\}$$

$$false(E_i) = \{p_j(E) / p_j(E) \in Prop \ \& \ \vdash \sim p_j(E_i)\}$$

It can easily be shown, using the assumption of consistency in 3-valued logic, that if *Prop* and $true(E_i)$ are given, then it is possible to infer $false(E_i)$ and $unknown(E_i)$. Hence, where no further qualification is made, reference to a property set should be taken to indicate the true property set.

The foregoing assumes that the truth value of a property assertion in a domain of discourse is fixed over time, space, environment, etc. It is either true (or false) under all circumstances (or modalities), or else it has value 'unknown' – even it is known to be true (or false) in some circumstances.

Henceforth it will also be assumed that every element of *Prop* is a member of the true property set of at least one entity in *Ent*. Furthermore, we shall say that entity E_i has or does not have the property $p_j(E)$ according as to whether $p_j(E) \in true(E_i)$ or not.

It is interesting to note that entities may be characterized as either deterministic or non-deterministic in terms of the foregoing model. An arbitrary entity $E_i \in Ent$ is *deterministic* iff $unknown(E_i) = \emptyset$. An entity which is not deterministic is *non-deterministic*.

3. Definitions

3.1 Property Definitions

In this section two arbitrary properties $p_x(E)$ and $p_y(E)$ in *Prop* are considered, and for conciseness they are denoted by x and y respectively.

The notation $x \rightarrow y$ is used to assert that y logically follows from x . A necessary condition for this to hold is that every entity in *Ent* which has property x (and by assumption there will be at least one such entity) also has property y .

Conversely, $\sim[y \rightarrow x]$ asserts that y does not logically follow from x . A sufficient condition for this to hold is that some entity in *Ent* which has property y (and again by assumption there will be at least one such entity) does not have property x .

The following definitions will be useful :

- Property x *refines* property y iff
 $\vdash ([x \rightarrow y] \ \& \ \sim[y \rightarrow x])$
- Properties x and y are *equivalent* iff
 $\vdash ([x \rightarrow y] \ \& \ [y \rightarrow x])$
- Properties x and y are *independent* iff
 $\vdash (\sim[x \rightarrow y] \ \& \ \sim[y \rightarrow x])$
- Properties x and y *enrich* one another iff they are independent, but are common properties of at least one entity in *Ent*.
- Properties x and y are *mutually exclusive* iff they are independent but do not enrich one another.

Hence, any two properties in the property set of a given entity are either equivalent, or they are independent and enrich one another, or one of them refines the other.

However, equivalent properties will always be either jointly present or jointly absent in an entity's property set. They are thus not helpful in discriminating between entities. Consequently, it will be assumed that domains of discourse are chosen in such a way that no equivalent properties occur. Thus any pair of properties that an entity has either enrich one another, or one of them refines the other.

3.2 Abstraction

Consider two distinct entities E_i and E_j in *Ent*. E_j is defined as an *abstraction* of E_i iff $true(E_j)$ is a subset of $true(E_i)$.

This definition implies the following about E_i and E_j :

- Since E_i and E_j are distinct entities, the following must hold :
 - $true(E_j)$ is a proper subset of $true(E_i)$,
 - $unknown(E_i)$ is a proper subset of $unknown(E_j)$,
 - $false(E_j)$ is a proper subset of $false(E_i)$.
- Properties in $true(E_j)$ but not in $true(E_i)$ are in $unknown(E_i)$. Similarly properties in $false(E_i)$ but not in $false(E_j)$ are also in $unknown(E_j)$. Informally, therefore, there is a parallel between abstraction and non-determinism. The degree of abstraction is directly related to the degree of non-determinism.
- An entity may be an abstraction of several entities which may be, but need not be abstractions, of one another.
- If an entity E_a is an abstraction of entity E_r , and latter has exactly one more property in its property set than the former, then E_r either refines E_a or E_r enriches E_a .
- Let E_a be some arbitrary abstraction of E_r and let E_{ra} denote some abstraction of E_r which has E_a as one of its abstractions. By choosing E_{ra} appropriately, at least one of the following assertions will be true :
 - E_r refines E_a
 - E_r enriches E_a
 - E_{ra} refines E_a
 - E_{ra} enriches E_a
- If a) holds above, then so does c) for any choice of E_{ra} and similarly for b) and d).

3.3 Refinement/Enrichment

Based on these notions, an entity may sometimes be regarded as a refinement or an enrichment of one or more of its abstractions. Consider the entity E , and one of its abstractions, E_a . Let

$D = true(E) - true(E_a)$, i.e. D is the set of properties that E , has in addition to those in E_a .

- Entity E , is a *refinement* of entity E_a iff every property in D refines some property in $true(E_a)$.
- Entity E , is an *enrichment* of entity E_a if every property in D enriches all properties in $true(E_a)$.

Note that if some (but not all) properties in D refine properties in $true(E_a)$, and the remaining properties in D enrich all properties $true(E_a)$ then E , neither refines nor enriches E_a .

3.4 Corollaries

Several corollaries follow from the foregoing definitions. A non-exhaustive list includes the following :

- The entity set may be partitioned into two sets: a set of entities which are abstractions of other entities and a set containing the remaining entities. Elements of the latter set may appropriately be called *concrete* entities in the domain of discourse, and elements of the former, *abstract* entities.
- The enrichment relationship for properties is reflexive, but not transitive. For entities the enrichment relationship is irreflexive and transitive.
- The refinement relationship is transitive but irreflexive for both entities and properties.
- Furthermore, if $p_k(E)$ refines $p_l(E)$ and $p_k(E) \in true(E_j)$, then $p_l(E) \in true(E_j)$. However, the converse does not necessarily hold, namely if $p_k(E)$ refines $p_l(E)$ and $p_l(E) \in true(E_j)$, then it does not follow that $p_k(E) \in true(E_j)$.
- Entity refinement and entity enrichment are mutually exclusive notions.

3.5 Base Abstractions

If a given property in *Prop* refines another in *Prop*, it will be called a *refining* property. All those properties in *Prop* which are not refining are called *base* properties.

Note the following :

- Every pair of base properties in the property set of an arbitrary entity enrich one another.
- A refining property always refines at least one base property, but may also refine one or more other refining properties.
- The property set of any entity E_i can be uniquely partitioned into a non-empty set of base properties, and a (possibly empty) set of refining properties. These sets will be denoted by $bas(E_i)$ and $ref(E_i)$ respectively.

An abstraction E_j of E_i which is such that $true(E_j) = bas(E_i)$ will be called a *base abstraction* of E_i , and will be denoted by $ba(E_i)$.

However, while every entity has at most one base abstraction, a base abstraction need not always exist. For example, if $ref(E_i) = \phi$, then $true(E_i) = bas(E_i)$, and no base abstraction of E_i exists. Also, $bas(E_i)$ may be such that while individual properties in this set enrich one another, the consistency requirement for 3-valued logic truth tables demands that other refining properties always be present in the true property set of any abstraction of E_i . For example, properties of the form $(a \text{ or } b)$ and $(\sim b)$ enrich one another (neither logically follows from the other), but any abstraction with these properties in its property set, necessarily also contains the refining property (a) in its property set.

Nevertheless, in applying progressive refinement and/or enrichment to arrive at the properties of an entity E_i , much of the initial creative effort will go into determining $bas(E_i)$, and indeed into the base abstraction of E_i if it exists.

4. Ordering

Since refinement of entities is transitive and irreflexive, it defines a strict partial order on entities of *Ent*, which can be denoted by $>_r$. Hence if entity E_p is a refinement of entity E_q , this may be written as $E_p >_r E_q$.

Similarly enrichment of entities is a transitive and irreflexive relation, and defines a strict but partial order $>_e$ on entities of *Ent*. Thus $E_p >_e E_q$ denotes the fact that E_p is an enrichment of E_q .

The number of base properties in the property sets of entities provides a basis for defining a weak partial order on abstractions of entities in *Ent*. The order will be denoted by \geq_b . Given two arbitrary abstractions, E_p and E_q , of an arbitrary entity E_c (and allowing also for the possibility that either E_p and/or E_q is the entity E_c itself) then $E_p \geq_b E_q$ iff the number of base properties in $true(E_p)$ is greater or equal to the number of base properties in $true(E_q)$. $E_p \geq_b E_q$ is read as: ' E_p is at least as basic as E_q '.

Note the following :

- $E_p >_e E_q \rightarrow (E_p \geq_b E_q) \ \& \ \sim(E_q \geq_b E_p)$ since an enrichment, per definition, has more base properties than the entity which it enriches.
- $E_p >_r E_q \rightarrow (E_p \geq_b E_q) \ \& \ (E_q \geq_b E_p)$ since a refinement of an entity per definition adds refining properties, but not base properties, to the property set of the refined entity.

In a given context some base properties may be regarded as more important than others. In such a case, a suitably chosen weighting could be applied to the base properties and the ordering would then be according to the weighted sum of base properties.

Several approaches to defining orderings on abstractions now suggest themselves, some of which are more naive than others.

The most naive approach is to postulate a weak total order on abstractions, based on the number of properties in the respective true property sets. The semantics of such an ordering is, however, not very clear.

A somewhat more meaningful approach is to postulate a strict partial order, $<_a$, such that $E_p <_a E_q$ iff E_q is an abstraction of E_p .

Ideally an ordering for any pair of abstractions of the same entity would be desirable – even if neither one is an abstraction of the other. One approach for achieving this would be to extend the previous ordering to a weak order, \leq_a , whereby:

$$E_p \leq_a E_q \text{ iff } (E_p <_a E_q) \text{ or } (\sim E_p <_a E_q \ \& \ \sim E_q <_a E_p \ \& \ E_p \geq_b E_q). \quad (I)$$

In general, such an ordering is weak when considered over all entities in *Ent* since neither (I)

nor (II) need necessarily hold for two arbitrary entities. However, the ordering is total over abstractions of a given entity.

Note that if (II) holds, but not (I), then the ordering implies that an abstraction with few base properties which may possibly have many refining properties is to be considered more abstract than an abstraction with many base properties which may not be at all refined. Furthermore, two abstractions with disjoint property sets each having the same number of base abstractions would be considered to be equally abstract, even if the property set of one abstraction contained many refining properties, while the other contained none. In other words the number of base properties, rather than the degree of refinement dominates in determining the level of abstraction.

An even more sophisticated ordering of abstractions could be defined based on a metric on each abstraction. This metric should reflect not only the number of base properties (perhaps weighted in some way) in the relevant property set, but also the extent to which the property set shows how individual base properties have been refined. Precisely how such a metric should balance the degree of refinement against the number of base properties will depend, inter alia, on the domain of discourse.

5. Other Work

This paper is a summary of work reported on elsewhere [7]. The latter extends the concepts discussed here to the area of specifications, and proposes definitions for concepts such as valid or invalid, partial or complete specifications, as well as for abstract, refined or enriched specifications.

The notion of abstraction is qualitatively explained in a number of texts on software development. A typical example is :

'By abstraction we mean the act of singling out a few properties of an object for further studies or use, omitting from consideration other properties that don't concern us for the moment.' [4]

Darden [2] spells out a similar idea of abstraction in the AI context in somewhat more detail, claiming that 'abstraction formation involves loss of content'. The present paper's definition of an abstraction is very much in the spirit of these definitions. It is, however, more formal, and given in the context of a domain of discourse.

Benzon and Hayes [1] criticize Darden's approach. However, the latter appear to be more concerned with 'issues related to human cognitive

endeavors in forming abstract concepts' rather than doing 'a conceptual analysis of abstraction as it is used in current computational AI work' [3].

Hoare *et. al.* [5] propose an weak partial ordering between programs which is very similar to the ordering on abstractions $<_a$ discussed above. Their ordering, which will be denoted here by \geq_H is defined as :

$$P \geq_H Q \text{ iff } P \cup Q = P$$

where $P \cup Q$ designates a program that makes a non-deterministic choice to function as either P or as Q . Were it not for the fact that in the present paper $<_a$ has been defined irreflexively, the following would hold for any pair of programs P and Q :

$$P \geq_H Q \text{ iff } Q <_a P.$$

The definition of \geq_H leads to the concept of an *abstract program*, a concept that is stricter but similar to the notion of an abstract specification [7].

Morgan and Robinson [8] propose a definition for refinement in the context of software development which is based on the concepts of pre- and post-conditions. Using the well-known notation $wp(P, post)$ for the weakest precondition of a program P with post-condition $post$, they define refinement as follows:

'For programs P and Q we say P is refined by Q , written $P \leq Q$, iff for all post-conditions $post : wp(P, post) \Rightarrow wp(Q, post)$.'

It can be shown that if Q is a refinement of P (but not vice-versa) according to the definition of Morgan and Robinson then Q is a refinement of P for a suitably chosen domain of discourse, according to the present paper's definition [7].

To the author's knowledge the theme of enrichment has not been formally addressed in the literature on software development.

References

- [1] W Benzon and D Hayes, [1987], *Reactions to Darden*, Letter in AI Magazine, 8(4), 7-8.
- [2] L Darden, [1987], *Viewing the History of Science as Compiled Hindsight*, AI Magazine, 8(2), 33-41.
- [3] L Darden, [1987], *Darden's Response*, Letter in AI Magazine, 8(4), 11.
- [4] D Gries, [1981], *The Science of Programming*, Springer-Verlag.
- [5] C A R Hoare, I J Hayes, J He, C C Morgan, A W Roscoe, J W Sanders, I H Sorensen, J M Spivey and B A Sufrin, [1987], *Laws of programming*, CACM, 30(8), 672-686.
- [6] S C Kleene, [1974], *Introduction to Metamathematics*, Wolters-Noordhof Publishing and North Holland Publishing Company, 7th Reprint.
- [7] D G Kourie, [1988], *Towards a general theory of abstractions, refinements and enrichments*, Internal Report, University of Pretoria.
- [8] C Morgan and K Robinson, [1987], *Specification statements and refinement*, IBM J. Res. Develop., 31(5), 546-555.
- [9] N J Nilsson, [1982], *Principles of Artificial Intelligence*, Springer-Verlag.

NOTES FOR CONTRIBUTORS

The purpose of the journal will be to publish original papers in any field of computing. These may include research, review and exploratory articles of interest to the journal's readers. The preferred language of the journal will be English, although papers in Afrikaans or other congress languages of IFIP will not be precluded. Typed manuscripts for review should be submitted in triplicate to:

Professor D G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
Pretoria

Form of Manuscript

Manuscripts for review should be prepared according to the following guidelines.

- Use double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
 - title (as brief as possible)
 - author's initials and surname
 - author's affiliation and address
 - an abstract of less than 200 words
 - an appropriate keyword list
 - a list of Computing Review Categories.
- Tables should be typed on separate sheets of A4 paper, and should be numbered and titled.
- Figures should also be supplied on separate sheets of A4 paper, and should be identified on the back in pencil with the author's name and the figure number. Original line drawings, and not photocopies, should be submitted.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin. Distinguish clearly between:
 - upper and lower case letters
 - the letter O and figure zero
 - the letter I and the number one
 - the letter K and kappa.

References should be listed after the text in alphabetical order of the (first) author's surname, cited in the text in square brackets. References should take the following form:
[1] E Ashcroft and Z Manna, [1972], The translation of 'GOTO' programs to 'WHILE' programs, *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.

[2] C Bohm and G Jacopini, [1966], Flow diagrams, Turing machines and languages with only two formation rules, *Comm.*

ACM, 9 (3), 366-371.

Manuscripts accepted for publication should comply with the above guidelines, but may be in one of the following three formats:

- typewritten and suitable for scanning
- provided as an ASCII file on diskette
- camera-ready.

Authors wishing to provide camera-ready copy may obtain a page specification from the production editor.

Charges

A page charge, scaled to reflect production costs, will be levied on papers accepted for publication. The costs per final page are as follows:

Typed format, not camera-ready R60

Disk in ASCII format R40

Camera-ready format R20

These charges may be waived upon request of the author, and at the editor's discretion.

Proofs

Proofs of accepted papers will 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.

Copyright

Copyright in published papers will be vested in the publisher.

Letters and Communications

Letters to the editor will be welcomed and will provide a forum for discussion on topical issues. They should be signed, and should be limited to about 500 words.

Communications reflecting minor research contributions will be considered for publication in a separate section of the journal. Such communications will, however, not be regarded as a fully-fledged publication for FRD subsidiary 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.

Advertising

Placement of advertisements at R1000 per full page per issue and R500 per half page per issue will be considered. Enquiries should be directed to the production editor.

