Regardless of the size of your computer system it is a strategic tool. We are committed to helping you use it most effectively. All it needs is a little imagination.

UNIDATA
Telephone (012) 45-1152

THERE'S A GIANT IN EVERY ONE OF US
The Department of Computer Science at Pretoria University is inviting applications for two annually awarded bursaries for MSc study. These bursaries are aimed at outstanding students who have a Computer Science Honours degree (or equivalent qualification). As far as possible, one such bursary will be awarded to a South African student, while the other will be awarded to a student from another African state. The hope is to thereby promote interaction between South African computer scientists and computer scientists elsewhere in Africa.

The bursaries are funded by the Department's contract research work and will amount to a minimum of R20000 per year. They are intended to cover costs for tuition, housing, transport, books, and an adequate allowance for other living expenses. Successful candidates will be required to work for 20 hours per week as part of the Computer Science Department's contract research team. The work assigned to each candidate will be aimed at providing him with practical experience in his field of interest.

Study fields for which bursaries will be considered include:
- Computer Networking
- Distributed Systems
- Software Engineering
- Artificial Intelligence
- Intelligent Computer Aided Instruction
- Machine Aided Translation

Students should aim to complete the dissertation that is required for an MSc within two years of study (i.e. one year of full time study). The dissertation should preferably be written in English but it may also be written in Afrikaans.

The continuation of the bursary and adjustments to the allowance will be decided upon annually and will depend on progress. It is a condition of the bursary that, after completing his/her studies, the student should return to his/her country of origin for the same number of years for which the bursary was held.

Applicants may request application forms from:

MSc. Bursaries
The Computer Science Department
University of Pretoria
Hillcrest
Pretoria
0083
South Africa
Editor's Notes

It is with sincere gratitude that SACJ takes leave of Dr Peter Lay who, until recently, was the assistant editor dealing with Information Systems. He has left academia for what sounds like a more gentle lifestyle. (He has gone farming!) Under Peter’s stewardship the number of high-quality IS papers in SACJ grew steadily. In general, IS papers tend to be accessible and relevant to a wide spectrum of computer professionals, and the quality of IS papers that have been appearing in SACJ has significantly contributed to the increased interest being shown in the journal by the local computer industry.

If this growth in interest is to be sustained, it is urgent and important to find a suitable replacement assistant editor. The ideal candidate should not only be respected as an academic by his peers, but should also be disposed to enthusiastically promote SACJ in the private sector. Since a shortlist of candidates is currently being compiled, I would like issue a general appeal for names that might be included on it. Please contact me urgently if you would like to be considered for the job, or if you would like to nominate someone that you consider to be particularly suitable.

My three year term of office as editor expires in October. I have always considered it a great privilege to hold this position, and as a result, I felt honoured when the SAICS executive committee requested that I stay on for a further term. Nevertheless, I initially declined the request on the grounds that the time-demands of the job were significantly eroding my ability to fulfil other duties. Particularly demanding has been the task of seeing to the typesetting of the various contributions - either by doing it myself, or by ensuring that it is adequately done by someone else. Recently, however, Prof G de V Smit (Riel Smit) at UCT has offered to assume the role of production editor. This generous offer so much changes the complexion of what is being asked of me that I am now both willing and honoured to continue as editor for another term. I am very grateful to Riel for his offer and I look forward to working with him. In future, authors whose papers have been accepted for publication will be asked to liaise directly with him regarding the precise form in which the final contribution should be submitted.

The next issue of SACJ will consist largely of a selection of papers that were presented at the 6th South African Computer symposium. The selection will be based on comments from the referees who, at the time, were asked to adjudicate the papers in terms of their appropriateness for both the conference as well as for SACJ publication. Papers which, in the opinion of one or more referees, required major revision will have to be resubmitted to SACJ for refereeing purposes. Authors will soon be contact in this regard.

At the time of writing, the updated list of "approved" publications for the first half of 1991 had not yet been released by the relevant authorities. For the sake of past, present and future contributors I sincerely hope that SACJ will be on the list when it eventually comes out. However, I have become increasingly aware that there is a real danger of laying too much store on papers published in so-called approved journals as a basis for evaluating and rewarding research. I hope to expand more fully on this theme in a future edition of SACJ. Keep watching this space!

Derrick Kourie
Editor

This SACJ issue is sponsored by a generous donation from UNIDATA
Knowledge Representation using Formal Grammars

S H von Solms, E M Ehlers and D J Enslin
Department of Computer Science, Rand Afrikaans University, Johannesburg, 2000

Abstract

In this paper, formal grammars, specifically random context grammars, are used as a vehicle for knowledge representation in expert systems. Random context grammars and the manner in which they are used to represent knowledge are discussed. An example of a knowledge base is given.

Received September 1990, Accepted January 1991.

1. Introduction

It is generally accepted that effective representation of domain knowledge is a key to the success of expert systems [1,2]. Well known ways presently used for knowledge representation are semantic networks, production rules, frames and logic [3].

In this paper we describe a new way of presenting knowledge, based on the theory of formal grammars, specifically random context grammars [4]. In this approach, a knowledge base consists of a set of production rules, with each rule having a permitting and forbidding context, controlling the situation in which the specific rule may be used ("fired").

2. Random Context Grammars

A random context grammar (RCG) is a 4-tuple:
\[ G = (V_N, V_T, P, S) \]
where \( V_N, V_T \) and \( S \) have the usual formal language meaning. \( P \) is a finite set of rules of the form:
\[ A \rightarrow \alpha (U; T) \]
where \( A \in V_N, \alpha \in (V_N \cup V_T)^*, U \subseteq V_N, T \subseteq V_N \) and \( U \cap T = \emptyset \). \( U \) is called the permitting context, and \( T \) the forbidding context.

A production like the one above can be applied to the sentential string: \( B \ A \gamma \), resulting in \( B \alpha \gamma \) if:
- all elements of \( U \) appear in \( B \gamma \), and
- no element of \( T \) appears in \( B \gamma \).

3. Permitting and forbidding context conditions in knowledge representation rules

In our representation, we use the idea of a permitting and forbidding context in a rule to control the application of a rule to a specific word (situation). Permitting conditions will demand that certain aspects be true (present), while forbidding conditions demand that certain other aspects be untrue (not present) for any rule to be applicable.

Consider the following rules from our knowledge base presented in the next paragraph. (This knowledge base defines an animal classification system.)

1. ANIMAL \( \rightarrow \) REPTILE (EGGS; FISH, BIRD, MAMMAL).
2. ANIMAL \( \rightarrow \) BIRD (LUNGS; MAMMAL, REPTILE).

Rule 1 is interpreted as follows:

An ANIMAL (the starting symbol of the derivation), is (can be) a REPTILE if it lays EGGS (permitting context), unless it is a FISH, or a MAMMAL or a BIRD (forbidding context).

Rule 2 is interpreted as follows:

An ANIMAL is a BIRD if it has LUNGS, unless it is a MAMMAL or a REPTILE.

The motivation for this approach is that it conforms more naturally to the way a human thinks and reasons when confronted with a problem.

To prove a premise, certain aspects known to support the premise are looked for (permitting contexts), and certain aspects known to disprove the premise (forbidding context), are eliminated. If the context conditions are satisfied, the premise holds.

It is also conjectured that this approach will simplify
the process of acquiring knowledge from an expert by prompting the expert for permitting and forbidding context conditions during the process of creating the knowledge base.

4. An example knowledge base

In this section we provide an example (only partial) for an animal classification system. The example will be used in the following sections, and is presented in table 1 below.

5. Non-determinism in reasoning

In many (complex) cases, a derivation can lead to different answers, depending on the available knowledge. In an animal classification situation, an expert may deduce that a specific animal may be a leopard or a cheetah, because no more information to make the final decision is available. With even less information he may only be able to deduce that the animal is a mammal. The knowledge representation and deduction formalism presented, follows this way of reasoning because it is precisely the characteristic of a non deterministic random context grammar to end a derivation in more than one terminal symbol, depending on the derivation path and the available (or unavailable) context conditions. We therefore also feel that the derivation mechanism used by the proposed formalism agrees very naturally with that of the human way of reasoning.

6. The derivation process

In a derivation process, we start with the start symbol, say ANIMAL, together with a set of context conditions, permitting and forbidding, supplied by the user, say
- VERY-SHARP-SENSES
- TEETH-ARE-FANGS
- HAIR
as permitting conditions.

The start symbol ANIMAL, is placed on the start list. Every rule with ANIMAL as the left hand side is investigated to see if the context conditions are satisfied. If so, the rule is applied (fired).

Consider rule 1.1 in table 1. This rule's permitting context (SUCKLES) does not appear in the list above, so the rule cannot be used. The next rule considered is 1.2. Because the context conditions of this rule is satisfied by the list above (HAIR does appear in the permitting context of rule 1.2), the rule is applied, and ANIMAL is replaced by MAMMAL.

All other rules with ANIMAL on the left are also considered, to see if any other application would be possible. In this case, it would not be.

All rules with MAMMAL on the left is now investigated. In this case rule 7.1 is applicable, and after that rule 8.1. The derivation therefore looks as follows:

ANIMAL => MAMMAL => CARNIVORE => CAT.

CAT is a terminal symbol because it does not appear on the left of any rule. The derivation stops successfully.

7. Inheritance

Consider the derivation in 6. CAT was derived from CARNIVORE, therefore CAT inherits all the permitting context conditions from rules 7.2 to 7.4.

8. Implementation

A tool for knowledge representation using the formalism described above, was developed.

This tool allows the creation of a knowledge base using context conditions, and reasons according to the method described in 6.

This is a full MS-DOS-based system, providing facilities to create a knowledge base of the type described above, to make derivations, to edit the knowledge base and to explain the derivation path used.

For more information on the system, the authors can be contacted.

9. Summary

A new formalism for presenting knowledge in a knowledge base was presented. This formalism is based on random context grammars, and uses permitting and forbidding context conditions.

A full implementation of the formalism on a personal computer exists.

Bibliography

### Table 1: An Example Knowledge Base

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Source</th>
<th>Permitting Context</th>
<th>Forbidding Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>ANIMAL → MAMMAL</td>
<td>(SUCKLES)</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>ANIMAL → MAMMAL</td>
<td>(HAIR)</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>ANIMAL → MAMMAL</td>
<td>(WARM-BLOODED)</td>
<td>BIRD</td>
</tr>
<tr>
<td>1.4</td>
<td>ANIMAL → MAMMAL</td>
<td>(LUNGS)</td>
<td>BIRD, REPTILE</td>
</tr>
<tr>
<td>2.1</td>
<td>ANIMAL → BIRD</td>
<td>(WINGS)</td>
<td>MAMMAL (BAT)</td>
</tr>
<tr>
<td>2.2</td>
<td>ANIMAL → BIRD</td>
<td>(BEAK)</td>
<td>MAMMAL (PLATYPUS)</td>
</tr>
<tr>
<td>2.3</td>
<td>ANIMAL → BIRD</td>
<td>(FEATHERS)</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>ANIMAL → BIRD</td>
<td>(EGGS)</td>
<td>MAMMAL (PLATYPUS)</td>
</tr>
<tr>
<td>2.5</td>
<td>ANIMAL → BIRD</td>
<td>(LUNGS)</td>
<td>REPTILE, FISH</td>
</tr>
<tr>
<td>2.6</td>
<td>ANIMAL → BIRD</td>
<td>(FLY)</td>
<td>MAMMAL (BAT)</td>
</tr>
<tr>
<td>2.7</td>
<td>ANIMAL → BIRD</td>
<td>(WARM-BLOODED)</td>
<td>MAMMAL</td>
</tr>
<tr>
<td>3.1</td>
<td>ANIMAL → FISH</td>
<td>(AQUATIC)</td>
<td>MAMMAL (WHALE), ) ( REPTILE, BIRD</td>
</tr>
<tr>
<td>3.2</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td>MAMMAL, BIRD,</td>
</tr>
<tr>
<td>3.3</td>
<td>ANIMAL → FISH</td>
<td>(STICKY-SCALES)</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td>REPTILE</td>
</tr>
<tr>
<td>3.6</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td>FISH</td>
</tr>
<tr>
<td>3.8</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td>FISH, BIRD,</td>
</tr>
<tr>
<td>3.9</td>
<td>ANIMAL → FISH</td>
<td>(TOES-ARE-CLAWS)</td>
<td></td>
</tr>
<tr>
<td>3.10</td>
<td>ANIMAL → FISH</td>
<td>(LUNGS)</td>
<td>MAMMAL, BIRD</td>
</tr>
<tr>
<td>3.11</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.12</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.13</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.14</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.15</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td></td>
</tr>
<tr>
<td>3.16</td>
<td>ANIMAL → FISH</td>
<td>(TOES-ARE-CLAWS)</td>
<td></td>
</tr>
<tr>
<td>3.17</td>
<td>ANIMAL → FISH</td>
<td>(LUNGS)</td>
<td></td>
</tr>
<tr>
<td>3.18</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.19</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.20</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.21</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.22</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td></td>
</tr>
<tr>
<td>3.23</td>
<td>ANIMAL → FISH</td>
<td>(TOES-ARE-CLAWS)</td>
<td></td>
</tr>
<tr>
<td>3.24</td>
<td>ANIMAL → FISH</td>
<td>(LUNGS)</td>
<td></td>
</tr>
<tr>
<td>3.25</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.26</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.27</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.28</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.29</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td></td>
</tr>
<tr>
<td>3.30</td>
<td>ANIMAL → FISH</td>
<td>(TOES-ARE-CLAWS)</td>
<td></td>
</tr>
<tr>
<td>3.31</td>
<td>ANIMAL → FISH</td>
<td>(LUNGS)</td>
<td></td>
</tr>
<tr>
<td>3.32</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.33</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.34</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.35</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.36</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td></td>
</tr>
<tr>
<td>3.37</td>
<td>ANIMAL → FISH</td>
<td>(TOES-ARE-CLAWS)</td>
<td></td>
</tr>
<tr>
<td>3.38</td>
<td>ANIMAL → FISH</td>
<td>(LUNGS)</td>
<td></td>
</tr>
<tr>
<td>3.39</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.40</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.41</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.42</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.43</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td></td>
</tr>
<tr>
<td>3.44</td>
<td>ANIMAL → FISH</td>
<td>(TOES-ARE-CLAWS)</td>
<td></td>
</tr>
<tr>
<td>3.45</td>
<td>ANIMAL → FISH</td>
<td>(LUNGS)</td>
<td></td>
</tr>
<tr>
<td>3.46</td>
<td>ANIMAL → FISH</td>
<td>(GILLS)</td>
<td></td>
</tr>
<tr>
<td>3.47</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.48</td>
<td>ANIMAL → FISH</td>
<td>(FINNS)</td>
<td></td>
</tr>
<tr>
<td>3.49</td>
<td>ANIMAL → FISH</td>
<td>(COLD-BLOODED)</td>
<td></td>
</tr>
<tr>
<td>3.50</td>
<td>ANIMAL → FISH</td>
<td>(EGGS)</td>
<td></td>
</tr>
</tbody>
</table>

In the knowledge base, the symbol in brackets denotes a typical example.
Book Review


Reviewer: Mr N L O Cowley, University of Port Elizabeth

Andrew Tanenbaum is internationally known as an author of outstanding and popular computer science textbooks. Structured Computer Organisation was first published in 1976. It is a measure of the book's success that it has survived so long in the rapidly changing world of computing education. The third edition of the book has been extensively revised to reflect current directions in computer organisation, without losing the positive characteristics of the previous two editions.

The central principle of the book is that a computer can be viewed as a hierarchy of levels, each of which performs some major function. The bottom level is the physical hardware, and the levels above this are virtual machines, each with its own machine language. In order of increasing abstraction, the virtual machine levels are the microprogramming level, the conventional machine level, the operating system machine level, the assembly language level and the problem-oriented language level. Using these six levels structures the material in the book effectively and simplifies understanding (because Miller's Law is satisfied).

Each of these levels (except the problem oriented language level) is discussed in a separate chapter. Adopters may find it necessary to supplement the chapter on digital logic with extra notes, especially if they follow a more formal and detailed approach. The Intel and Motorola families of processors are used as illustrative examples throughout. Only two example operating systems are considered (UNIX and OS/2), but one must remember that the text is not an operating system text.

Although most of the book is devoted to a study of the conventional von Neumann machine, the second last chapter deals with advanced computer architectures (RISC machines and parallel architectures) and was very exciting to read. The last chapter consists of a comprehensive collection of suggested readings and literature citations which lead one on to further studies in the field, although this really applies to senior and graduate students. One appendix deals with finite precision numbers and arithmetic, and another with floating-point (IEEE Standard 754) numbers and arithmetic. The index is adequate, although not outstanding.

This logically structured and comprehensive book focuses on concepts rather than details or complex mathematics. Explanations are clear and complete. All chapters are suffixed by summaries and sets of problems (some very open-ended and challenging - a typical feature of Tanenbaum's books). A problem solutions manual is available, as well as a toolbox to accompany the chapter on microprogramming. The material in the book can be covered in one year.

I recommend this book as a sound and proven text on which to base an undergraduate course in computer organisation.

Books Received

The following books are available for review by appropriate reviewers:

J Loecks, K Mehlorn and R Wilhelm, [1988], Foundations of Programming Languages, J Wiley & Sons.

Call for Reviews

Readers are invited to submit reviews of books which might be of interest to colleagues. If, for example, you have recently used a new book for a course and consider that it is particularly good (or bad - for that matter), please feel free to share your insights through the medium of these pages. Likewise, a short review of a book that you might recently have read as part of your research could well be of interest to others.
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 papers. However, non-refereed review and exploratory articles of interest to the journal’s readers will be considered for publication under sections marked as Communications or 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 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 relevant Computing Review Categories.
- Tables and figures should be on separate sheets of A4 paper, and should be numbered and titled. Figures should be submitted as original line drawings, and not photocopies.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.
- 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 in square brackets. References should thus take the following form:

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

- in a typed form (i.e. suitable for scanning);
- as an ASCII file on diskette; or
- as a WordPerfect, \TeX or \LaTeX file; or
- in camera-ready format.

A page specification is available on request from the editor, for authors wishing to provide camera-ready copies. A styles file is available from the editor for Wordperfect, \TeX or \LaTeX documents.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R20-00 per final page for camera-ready contributions and the maximum is R100-00 per page for contributions in typed format.

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

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.

Note that, in the case of camera-ready submissions, it is the author’s responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

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 specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.
## Contents

**GUEST CONTRIBUTION**

Why all the Fuss About Neural Networks?
G Barth ........................................... 4

**RESEARCH ARTICLES**

The Placement of Subprograms by an Automatic Programming System
J P du Plessis and H J Messerschmidt ............................... 9

An Investigation into the Separation of the Application from its User Interface
J H Greyling and P R Warren ............................................. 16

The Physical Correlates of Local Minima
L F A Wessels, E Barnard and E van Rooyen .................. 22

An Efficient Primal Simplex Implementation for the Continuous 2-Matching Problem
T H C Smith, T W S Meyer and L Leenen .................. 28

Concept Network Framework for a Multi-paradigm Knowledge Base
J Kambanis .................................................. 32

PEST - A Microcomputer Pascal Based Expert System Shell
A G Sartori-Angus and R Neville .................................. 39

A Linda Solution to the Evolving Philosophers Problem
S E Hazelhurst ............................................ 44

**TECHNICAL NOTE**

Knowledge Representation using Formal Grammars
S H von Solms, E M Ehlers and D J Enslin .................. 54

**COMMUNICATIONS AND REPORTS**

Book Review & Books Received ........................................ 57
Cost-effective Visual Simulation Based on Graphics Workstations
A Caduri and D G Kourie ........................................... 58
A Method of Controlling Quality of Application Software
T D Crossman .................................................. 70
An Update on UNINET-ZA: The Southern African Academic and Research Network
V Shaw .................................................. 75