DESUPER DEORSUM

CELERIORE EXISTIMATE ET RECTIORE FACTO

Quaestiones Informaticae

Vol. 2 No. 3 September, 1983

Quaestiones Informaticae

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

'n Amptelike tydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

Subscriptions

Annual subscriptions are as follows:

	SA	US	UK
Individuals	R6	$7	£3,0
Institutions	R12	$14	£6,0

Circulation manager
Mr E Anderssen
Department of Computer Science
Rand Afrikaanse Universiteit
P O Box 524
Johannesburg 2000
Tel.: (011) 726-5000

Quaestiones Informaticae is prepared for publication by Thomson Publications South Africa (Pty) Ltd for the Computer Society of South Africa and the South African Institute of Computer Scientists.

Nebulas as Structural Data Models

H.O. van Rooyen and D.J. Weiermans
University of South Africa, South Africa

Abstract
Nebula modelling is a generalised structural modelling technique which applies to any finite set of objects (entities) and n-ary relationships ($n \geq 2$). A nebula model can be presented in either diagrammatic or tabular form, and combines the properties of graph (binary) data models with those of tabular (relational) data models. We give a brief introduction to nebula data modelling by means of a simple illustration.

1. Introduction

Modern binary ($n = 2$ only) structural modelling is based on the work of Harary, Norman and Cartwright [1]. Lendaris has defined a structural model as "a collection of elements and their relationships" and stated that "structural modeling holds the promise of converting a completely intuitive process of model building into a more systematic approach, and enhancing communication within a heterogeneous group" [2]. McLean and Shepherd defined the term "structure" as "the way in which the component parts of the complex whole are inter-related" and stated that "a structural model focuses on the task of selecting the components of a model and explicitly stating the interactions between them" [3]. To date, structural modelling has dealt only with binary relationships, but nebula modelling extends it to n-ary relationships with $n \geq 2$ [4,5,6,7,8,9 and 10]. Graph theory and network theory are special cases of nebula theory. Nebula based structural modelling provides an unquantified description of any real world situation in the following way. First the situation is reduced (simplified) to a description in terms of a set (countable or countably infinite) of object (entity/concept) names and (sentential) statements of relationships among the objects named. Each statement is then encoded as an n-tuple of object names, and these object names and n-tuples constitute the model.

2. Basic concepts, and an example

By an abstract *nebula* of type λ we mean a pair $\langle A,F \rangle$ where the underlying set A is denumerable, F is a function which correlates with every $i \in$ dom F a function

$$F(i)\colon A^{\lambda(i)-2} \Rightarrow P(AxA), \text{ and } \lambda\colon \text{dom } F \Rightarrow (\omega\text{-}2)$$

where $\omega = \{0,1,2,\ldots\}$ and $n = \{0,1,\ldots,n\text{-}1\} \in \omega$.

In this section we use an example to illustrate some basic concepts of nebula theory, namely

- diagrammatic presentation of $\langle A,F \rangle$
- occurrence x, i;j, y of label i;j in a diagram of $\langle A,F \rangle$
- tabular presentation of $\langle A,F \rangle$.

Formal definitions of these concepts can be found in [5,6,7,8,9 and 10].

Example

The degree curricula in the Science Faculty at UNISA are composed of small course units called modules. Prerequisite and parallel requirements are defined for each module, specifying the conditions under which a student may register for that module. A *prerequisite* requirement for a module v is a set E_v of one or more modules for which a student shall have obtained credit before he may enrol for module v. A *parallel* requirement for a module v is defined to be a set L_v of one or more modules which is such that the student must enrol for every $u \in L_v$ for which he has not yet obtained credit at the same time as he enrols for v.

Consider the following selection of data, where COS denotes Computer Science and COS000 represents the entrance qualification.

Module	*Prerequisites*	*Parallel modules*
COS111	COS000	—
COS121	COS000	—
COS131	COS000	INF101
COS132	COS000	COS131
COS211	COS111,121,INF101	—
COS212	COS111,121,INF101	COS211,221
COS221	COS111,121,INF101	COS211
COS231	COS131,132	—
COS231	COS111,MAT101,102	—
COS311	COS211,221	—

Table 1

For every module b in the table an n-tuple [occurrence/datum]

$$\langle a, r_1, r_2, \cdots r_k, S_1, S_2, .., S_\ell, b\langle$$

can be constructed, representing a relationship between the prerequisites and the parallels for the relevant module, where

$E_b = \{a, r_1, r_2, \ldots, r_k\}$ represents the prerequisites, and
$L_b = \{s_1, s_2, \ldots, s_\ell\}$ represents the parallels, for the module b.

We choose a nebula model $\langle A,F \rangle$ of the data given in table 1. Let the members of A represent the modules. Each $i \in$ dom F represents a registration relationship, and has the form $i = \langle k,\ell \rangle$, $k, \ell \in \omega$, with k = (number of prerequisites) — 1, and ℓ = number of parallels, for some module. Then

$$F(i)\colon A^{\lambda(i)-2} \Rightarrow P(AxA), \text{ where } \lambda(i) = k + \ell + 2, \text{ and}$$

$$(F(i))(j) = \begin{cases} \{\langle a,b \rangle \in AxA \mid \langle a, r_1, \ldots, r_k, s_1, \ldots, s_\ell, b \rangle \text{ belongs to the } \lambda(i)\text{-ary relation named by } i \in \text{dom } F \text{ and } j = \langle r_1, \ldots, r_k, s_1, \ldots, s_\ell \rangle\}; \\ \phi \text{ otherwise.} \end{cases}$$

This nebula $\langle A,F \rangle$, is such that

A = {COS000, COS111, COS121, COS131, COS132, COS211, COS212, COS221, COS231, COS311, INF101, MAT101, MAT102}

dom F = $\{\langle 0,0 \rangle, \langle 0,1 \rangle, \langle 1,0 \rangle, \langle 2,0 \rangle, \langle 2,1 \rangle, \langle 2,2 \rangle\}$

and, for example,

$$F(\langle 2,0 \rangle)(\langle x_1,x_2 \rangle) = \begin{cases} \{\langle \text{COS111, COS211} \rangle\} \text{ if } x_1 = \text{COS121 and } x_2 = \text{INF101}; \\ \{\langle \text{COS111, COS231} \rangle\} \text{ if } x_1 = \text{MAT101 and } x_2 = \text{MAT102}; \\ \phi \text{ otherwise.} \end{cases}$$

A diagrammatic presentation of $\langle A,F \rangle$ is given in figure 1, and the tabular presentation in table 2.

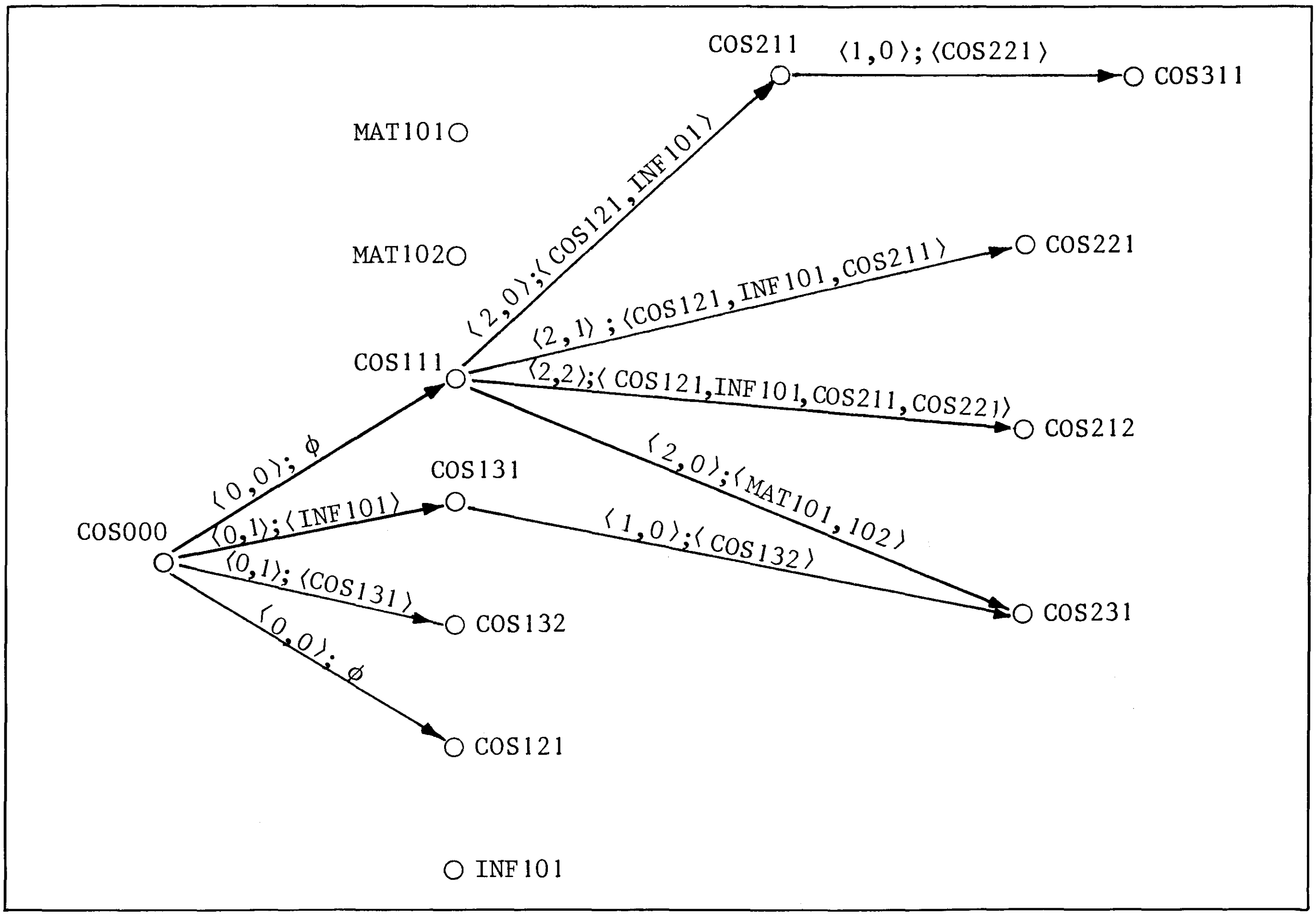

FIGURE 1 — Diagram of nebula ⟨A,F⟩

3. Substructures and Cascades*

Let ⟨A,F⟩ be a nebula of type λ and ⟨B,G⟩ be a nebula of type ρ. We say that ⟨B,G⟩ is a *subnebula* of ⟨A,F⟩, written $\langle B,G\rangle \leq \langle A,F\rangle$, iff

1. $B \leq A$ and
2. dom $G \leq$ dom F and
3. $\rho = \lambda|$ dom G where $|$ denotes restriction and
4. for each $i \in$ dom G and for every $j \in B^{\lambda(i)\text{-}2}$we have $(G(i))(j) \leq (F(i))(j)$.

⟨B,G⟩ is called a *spanning subnebula* of ⟨A,F⟩ iff $B = A$. ⟨B,G⟩ is a *reduct* of ⟨A,F⟩ iff $B = A$ and, for every $i \in$ dom G, we have $G(i) = F(i)$. Every reduct is a spanning subnebula, but the converse is not generally true.

Supremum (∨), infimum (∧), and complement of certain subnebulas of given nebula ⟨A,F⟩, can be formally defined, as can homomorphisms between nebulas [see 5, 6, 7, 8, 9, and 10].

The set X of all subnebulas of a given nebula ⟨A,F⟩, with partial order ≤ and corresponding operations ∧ and ∨, constitutes a distributive lattice with universal element ⟨A,F⟩ and null element $\langle \phi,\phi \rangle$. See [5].

In a nebula data model ⟨A,F⟩ *all* data manipulation is achieved by applying the operations ∨ and ∧ to subnebulas of ⟨A,F⟩, and data retrieval is attained by generating an appropriate subnebula using an operator scheme called "cascade".

4. Definition of a Cascade*

In general a *cascade* in ⟨A,F⟩ is a sequence $\{\langle B_k,G_k\rangle | k\epsilon\omega\}$ of subnebulas of ⟨A,F⟩ such that if $i \leq j$ then $\langle B_i,G_i\rangle$ is a subnebula

RELATION TABLE

i	j	arrow
⟨0,0⟩	ϕ	⟨COS000, COS111⟩
	ϕ	⟨COS000, COS121⟩
⟨0,1⟩	⟨INF101⟩	⟨COS000, COS131⟩
	⟨COS131⟩	⟨COS000, COS132⟩
⟨1,0⟩	⟨COS132⟩	⟨COS131, COS231⟩
	⟨COS221⟩	⟨COS211, COS311⟩
⟨2,0⟩	⟨COS121, INF101⟩	⟨COS111, COS211⟩
	⟨MAT101, MAT102⟩	⟨COS111, COS231⟩
⟨2,1⟩	⟨COS121, INF101, COS211⟩	⟨COS111, COS221⟩
⟨2,2⟩	⟨COS121, INF101, COS211, COS221⟩	⟨COS111, COS212⟩

ENTITY TABLE

COS000
COS111
COS121
COS131
COS132
COS211
COS212
COS221
COS231
COS311
INF101
MAT101
MAT102

TABLE 2 — Nebula ⟨A,F⟩ in tabular form

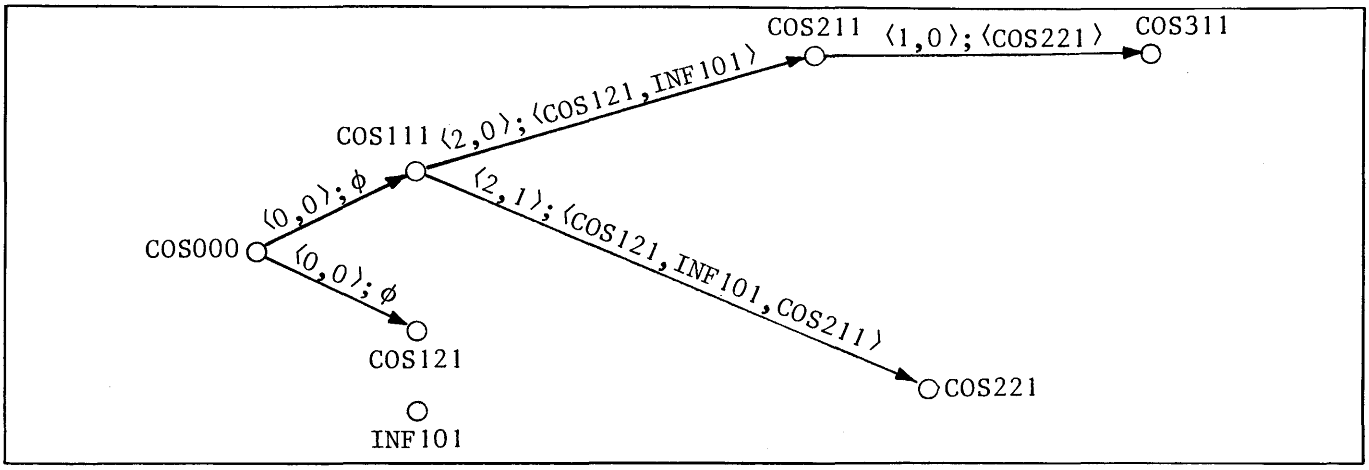

FIGURE 2 — Diagram of (B_3,G_3).

of $\langle B_j,G_j\rangle$. This definition can be limited in such a way that cascades are computer generable. An important example of this is the following.

Definition

The sequence $\{\langle B_k,G_k\rangle | k \epsilon \omega\}$ of subnebulas of $\langle A,F\rangle$ is called a *cascade generated from* $\langle B_0,G_0\rangle$ by Γ and τ iff

1. $B_o \leq A$ and G_0 is specified so that $\langle B_0,G_0\rangle \leq \langle A,F\rangle$ and
2. $B_{k+1} = B_k \cup (\Gamma_{R_k}(C_k) \cup \tau_{R_k}(C_k))$ where $C_0 = B_0$ and $C_k \leq B_k$ so that $\langle B_k,G_k\rangle \leq \langle B_{k+1},G_{k+1}\rangle$, and
3. dom $G_k \leq$ dom $G_{k+1} \leq$ dom F and for each $i \epsilon$ dom G_{k+1} we have $G_{k+1}(i) : (B_{k+1})^{\lambda(i)-2} \rightarrow P((B_{k+1})^2)$ where $(G_{k+1}(i))(j) = (G_k(i))(j) \cup ((F_{R_k}(i))(j) \cap (C_k \times B_{k+1}))$ and $i \epsilon$ dom G_{k+1} — dom G_k iff there is at least one $j \epsilon (B_{k+1})^{\lambda(i)-2}$ for which $(F_{R_k}(i))(j) \cap (C_k \times B_{k+1}) \neq \phi$, where the subscript R_k denotes the Γ and τ functions of some subnebula $\langle A,F_{R_k}\rangle$ of $\langle A,F\rangle$, with $R_k \leq \bigcup_{i,j}\{(F(i))(j)\}$, at each step k. Such a cascade is said to be *limited* iff, at each step k, R_k is chosen in such a way that $\tau_{R_k}(C_k) \leq C_k$, i.e. if $(F(i))(j) \epsilon R_k$ then $j \epsilon (C_k)^{\lambda(i)-2}$.

Strictly speaking it is not the cascade which is generated, but there is an $m \epsilon \omega$ such that for every $n \geq m$ we have $\langle B_n,G_n\rangle = \langle B_m,G_m\rangle$, and it is in fact $\langle B_m,G_m\rangle$ which is generated by Γ and τ from $\langle B_0,G_0\rangle$ in $\langle A,F\rangle$. Thus such a cascade generates a subnebula of $\langle A,F\rangle$, and with the appropriate choices of C_k and R_k at each step, every subnebula of $\langle A,F\rangle$ can be generated as the end product of a non-trivial cascade from some $\langle B_0,G_0\rangle$. Note that $\langle B_k,G_k\rangle \leq \langle B_{k+1},G_{k+1}\rangle$ is an overriding condition, so a cascade must stop when every choice of $C_k \leq B_k$ and $R_k \leq \bigcup_{i,j}\{(F(i))(j)\}$ yields $\langle B_{k+1},G_{k+1}\rangle \leq \langle B_k,G_k\rangle$. Cascades were originally defined to model thinking processes. Here every query defines a cascade, or sequence of cascades, which generates a subnebula containing the information relevant to that query.

To illustrate the use of "cascade" we return to our example.

Assume that a student specifies module COS311 as a goal for his degree. He has to know which modules lead to the specified module. This information can be retrieved by means of a cascade generated by Γ^{-1} and τ^{-1}, which is defined as follows:

$i=0$: $B_0 = \{COS311\}$ and $G_0 = \phi$ with dom $G_0 = \phi$.

$i=1$: $B_1 = B_0 \cup \Gamma_{R_0}^{-1}(C_0) \cup \tau_{R_0}^{-1}(C_0)$ where $C_0 = B_0$ and we take $R_k = \bigcup_{i,j}\{(F(i))(j)\}$ for each $k \epsilon \omega$ and drop the subscript R_k in this case.

Now $\Gamma_{R_0}^{-1}(C_0) = \{COS311, COS211\}$ and

$\tau_{R_0}^{-1}(C_0) = \{COS221\}$,

therefore $B_1 = \{COS311, COS211\} \cup \{COS221\}$

$= \{COS311, COS211, COS221\}$.

$(G_1)_{10}(\langle x\rangle) = \{\langle COS211, COS311\rangle\}$ if $x = COS221$; ϕ otherwise.

dom G_1 = dom $G_0 \cup \{\langle 1,0\rangle\} = \phi \cup \{\langle 1,0\rangle\} = \{\langle 1,0\rangle\}$.

Alternatively, we can take dom G_k = dom F for each k.

Proceeding in this way we find that $\langle B_3,G_3\rangle = \langle B_4,G_4\rangle$. A diagram of $\langle B_3,G_3\rangle$ is given in figure 2.

Several variations and combinations of cascades can be used, and appear to provide an adequate retrieval mechanism. Cascades are discussed in more detail in chapters 4 and 5 of [6].

5. Concluding Remarks

Nebula modelling is being used, with increasing success, in the field of education [7, 8 and 12], and has great potential for the modelling of knowledge structures, learning, and thought processes.

Many claims have been made as to the superiority of graphs and tables, one to another, for data modelling. Tsichritzis [13], for example, says the following: "Each approach has its strength and weaknesses. Neither is intrinsically simpler or more powerful. Relational data models . . . are elegant in their simplicity, but very limited semantically. Binary data models can represent complex relationships concisely, but are not very user oriented." About comparison of these data models he states: "The basis of comparison, therefore, should not be whether relations or graphs are simpler or more natural. Instead, one should consider the constructs that need to be added to tables and graphs to enable them to model adequately complex situations." Different data models provide us with different data modelling tools, and their usefulness depends on the problem to which they are applied. In this paper we have indicated, very briefly, that nebula theory provides a data model which has both a graph (binary) data model facet and a tabular (relational) data model facet in its representation of the structure of data. It retains the advantages, and some of the disadvantages, of both.

Tsichritzis [13] also states: "the ultimate data modeling tool which mixes data and knowledge about data is the predicate calculus . . . Strictly typed data models make some very ad hoc assumptions about what kinds of categories of objects will be allowed and what kinds of relationships between these categories can be specified. They force data to be homogenous. As a matter of fact, most data model research deals with finding good, general categories which can enforce this homogeneity requirement (e.g. the notion of a relation and a record type)." In the predicate calculus, "the emphasis is on allowing everything to be expressed in a uniform and formal environment without artificial restrictions on typing or categorizing objects". This last remark also applies to nebula modelling, because the members of the set A in an abstract nebula can be used to represent *any* "object", i.e. any "kinds of categories of objects" [13] are allowed. In nebula data modelling queries must be formulated in terms of subnebulas and paths, and thus the choice of an appropriate nebula model for a given problem may be difficult, or even impossible. Nevertheless, we feel that nebula data modelling deserves further investigation.

Experimental nebula models of a registration advice system

[6] and a syllabus (learning structure) design and analysis system [7,8] have been successfully implemented at UNISA.

References

[1] F. Harary, et al. *Structural Models: An Introduction to the Theory of Directed Graphs.* John Wiley & Sons. 1965.

[2] G. G. Lendaris, *Structural Modeling — A Tutorial Guide.* IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-10, No. 12, December 1980, pp. 807-840.

[3] J. M. McLean & P. Shepherd, *The importance of model structure. Futures.* Febr. 1976.

[4] H. O. van Rooyen, *Binary Networks in Graph Theory.* Ph.D. thesis, UNISA, 1976.

[5] H. O. van Rooyen, in collaboration with D. E. Wolvaardt and D. J. Weiermans, *Introduction to Nebula Theory.* Com. Sci. and Info. Systems Colloquium of the Univ. of S.A., Vol. 2, Nov. 1981.

[6] D. J. Weiermans, *Development of a Nebula Database for a Student Advice System.* Ph.D. thesis submitted for examination, UNISA, 1983.

[7] D. E. Wolvaardt, *Development of Nebula Theory with Application to Syllabus Databases.* Ph.D. thesis, UNISA, 1982.

[8] H. O. van Rooyen, D. E. Wolvaardt & H. Helm, *Formalized Concept Maps in a Science of Education.* Working Paper NEB$\neq$4. UNISA and RHODES, 1982.

[9] W. A. Labuschagne & H. O. van Rooyen, *On a Generalisation of Cayley Diagrams.* UNISA, 1980. Accepted for publication in Q.I., 1983.

[10] H. O. van Rooyen, *Menger's Theorem in a Nebula.* Proc. Second South African Computer Symposium, 1981. Accepted for publication in Q.I., 1983.

[11] Calendar 1981 — University of South Africa.

[12] R. J. van den Heever, S. H. von Solms & D. E. Wolvaardt, *Programontwerp: Fundamentele Beginsels.* To be published, HAUM, 1983.

[13] D. C. Tsichritzis & F. H. Lochovsky, *Data Models.* Prentice Hall, Inc., 1982.

*In this paper $\leq$ is used to denote "less than or equal to", "subject of", or "subnebula of", depending on the context.

Notes for Contributors

The purpose of this Journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles, exploratory articles of general interest to readers of the Journal. The preferred languages of the Journal will be the congress languages of IFIP although papers in other languages will not be precluded.

Manuscripts should be submitted in triplicate to: Prof. G. Wiechers at:

Department of Computer Science
University of South Africa
P.O. Box 392
Pretoria 0001
South Africa

Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins. The original ribbon copy of the typed manuscript should be submitted. Authors should write concisely.

The first page should include the article title (which should be brief), the author's name, and the affiliation and address. Each paper must be accompanied by a summary of less than 200 words which will be printed immediately below the title at the beginning of the paper, together with an appropriate key word list and a list of relevant Computing Review categories.

Tables and figures

Illustrations and tables should not be included in the text, although the author should indicate the desired location of each in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Illustrations should also be supplied on separate sheets, and each should be clearly identified on the back in pencil with the Author's name and figure number. Original line drawings (not photoprints) should be submitted and should include all relevant details. Drawings, etc., should be submitted and should include all relevant details. Drawings, etc., should be about twice the final size required and lettering must be clear and "open" and sufficiently large to permit the necessary reduction of size in block-making.

Where photographs are submitted, glossy bromide prints are required. If words or numbers are to appear on a photograph, two prints should be sent, the lettering being clearly indicated on one print only. Computer programs or output should be given on clear original printouts and preferably not on lined paper so that they can be reproduced photographically.

Figure legends should be typed on a separate sheet and placed at the end of the manuscript.

Symbols

Mathematical and other symbols may be either handwritten or typewritten. Greek letters and unusual symbols should be identified in the margin. Distinction should be made between capital and lower case letters between the letter O and zero; between the letter l, the number one and prime; between K and kappa.

References

References should be listed at the end of the manuscript in alphabetical order of author's name, and cited in the text by number in square brackets. Journal references should be arranged thus:

1. ASHCROFT, E. and MANNA, Z. (1972). The Translation of 'GOTO' Programs to 'WHILE' Programs, in *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.
2. BÖHM, C. and JACOPINI, G. (1966). Flow Diagrams, Turing Machines and Languages with only Two Formation Rules, *Comm. ACM*, **9**, 366-371.
3. GINSBURG, S. (1966). *Mathematical Theory of context-free Languages*, McGraw Hill, New York.

Proofs and reprints

Galley proofs will be sent to the author to ensure that the papers have been correctly set up in type and not for the addition of new material or amendment of texts. Excessive alterations may have to be disallowed or the cost charged against the author. Corrected galley proofs, together with the original typescript, must be returned to the editor within three days to minimize the risk of the author's contribution having to be held over to a later issue.

Fifty reprints of each article will be supplied free of charge. Additional copies may be purchased on a reprint order form which will accompany the proofs.

Only original papers will be accepted, and copyright in published papers will be vested in the publisher.

Letters

A section of "Letters to the Editor" (each limited to about 500 words) will provide a forum for discussion of recent problems.

Hierdie notas is ook in Afrikaans verkrygbaar.

Quaestiones Informaticae

Contents/Inhoud

*Presented at the second South African Computer Symposium held on 28th and 29th October, 1981.