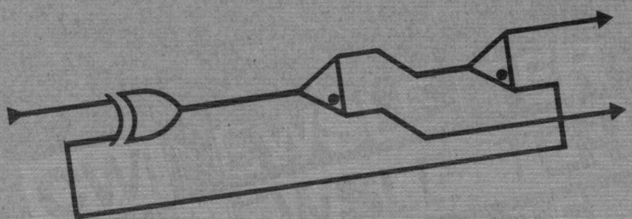
Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

Quarterly, Issue no. 15
June 1987



CWI NEWSLETTER

Number 15, June 1987

The CWI Newsletter is published quarterly by the Centre for Mathematics and Computer Science (Centrum voor Wiskunde en Informatica), Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. The Newsletter will report on activities being conducted at the Centre and will also contain articles of general interest in the fields of Mathematics and Computer Science, including book reviews and mathematical entertainment. The editors encourage persons outside and in the Centre to contribute to the Newsletter. Normal referee procedures will apply to all articles submitted.

The Newsletter is available free of charge to all interested persons. The Newsletter is available to libraries on an exchange basis.

All correspondence should be addressed to: *The CWI Newsletter, Centre for Mathematics and Computer Science, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands.*

ISSN 0168-826x

Contents

Centre for Mathematics and Computer Science
Centrum voor Wiskunde en Informatica

The Centre for Mathematics and Computer Science (CWI) is the research institute of the Stichting Mathematisch Centrum (SMC), which was founded on 11 February 1946.

The goal of CWI is to do fundamental and advanced research in mathematics and computer science, with special attention to those areas to which the research may have relevant applications. The research is fundamental in that it is mainly concerned with those problems for which there are as yet no standard methods of solution. It is advanced in that CWI aims at research work which is of a high level both nationally and internationally. Preference is given to those subjects which, from an international point of view, look likely to have interesting developments.

The research at CWI is organized in eight scientific departments:

- Pure Mathematics;
- Applied Mathematics;
- Mathematical Statistics;
- Operations Research and System Theory;
- Numerical Mathematics;
- Software Technology;
- Algorithms and Architecture;
- Interactive Systems

There are also a number of supporting sectors, in particular the Computer Systems and Telematics Sector, and an extensive Library.

The subdivision of the research into eight departments is less rigid than it appears, for there exists considerable collaboration between the departments. This is a matter of deliberate policy, not only in the selection of research topics, but also in the selection of the permanent scientific staff.

ACP_τ

A Universal Axiom System for Process Specification

J.A. Bergstra
University of Amsterdam, Department of Computer Science
P.O. Box 19268, 1000 GG Amsterdam
State University of Utrecht, Department of Philosophy
P.O. Box 8810, 3508 TA Utrecht

J.W. Klop
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam
Free University, Department of Mathematics and Computer Science
De Boelelaan 1081, 1081 HV Amsterdam

0. Introduction

Following R. Milner's development of his widely known Calculus of Communicating Systems, there have been in the last decade several approaches to *process algebra,* i.e. the algebraic treatment of communicating processes. In this paper we give a short and informal presentation of some developments in process algebra which started five years ago at the Centre for Mathematics and Computer Science, and since two years in cooperation with the University of Amsterdam and the State University of Utrecht[1]. Most of the present paper can be found in the more complete survey [6], where the subjects of specification and verification of processes are treated in so-called bisimulation semantics. Here, we adopt a further restriction by concentrating on the specification issue.

We start with a very simple axiom system for processes called Basic Process Algebra, in which no communication facilities are present. This system is interesting not only because it is a nucleus for all process axiom systems that are devised and analyzed in the 'Algebra of Communicating Processes', but also because it provides a link with the classical and successful theory of formal languages, in particular where regular languages and context-free languages are concerned. In Section 2 we explain this link.

Next, we introduce more and more operators, leading first to the axiom system ACP (Algebra of Communicating Processes) where communication between processes is possible, and finally to ACP_τ (Algebra of Communicating Processes with abstraction). Examples are given showing that the successive extensions yield more and more specification power; and a culmination point

1. Research partially supported by ESPRIT project 432, Meteor.

is the Finite Specification Theorem for ACP_τ, stating that every finitely branching, effectively presented process can be specified in ACP_τ by a finite system of recursion equations. Of course, an algebraic system for processes is only really interesting and useful if also sufficient facilities for process *verification* are present. These require an extension with some infinitary proof rules which will not be discussed here (for these, see the full version of this paper [6]). We refer also to the same paper for a more extensive list of references than the one below.

1. Basic Process Algebra

The kernel of all axiom systems for processes that we will consider, is Basic Process Algebra. Not only is for that reason an analysis of BPA and its models worth-while, but also because it presents a new angle on some old questions in the theory of formal languages, in particular about context-free languages and deterministic push-down automata. First let us explain what is meant by 'processes'.

The processes that we will consider are capable of performing atomic steps or *actions* $a,b,c, \ldots$, with the idealization that these actions are events without positive duration in time; it takes only one moment to execute an action. The actions are combined into composite processes by the operations $+$ and $\cdot$, with the interpretation that $(a+b)\cdot c$ is the process that first chooses between executing a or b and, second, performs the action c after which it is finished. At this stage it does not matter how the choice is made. These operations, *alternative composition* and *sequential composition* (or just sum and product), are the basic constructors of processes. Since time has a direction, multiplication is not commutative; but addition is, and in fact it is stipulated that the options (summands) possible at some stage of the process form a *set.* Formally, we will require that processes $x,y,z, \ldots$ satisfy the following axioms (where the product sign is suppressed):

BPA
$x+y=y+x$
$(x+y)+z=x+(y+z)$
$x+x=x$
$(x+y)z=xz+yz$
$(xy)z=x(yz)$

Table 1

In the Introduction we used the term 'process algebra' in the generic sense of denoting the area of algebraic approaches to concurrency, but we will also adopt the following technical meaning for it: any model of these axioms will be a *process algebra.* The simplest process algebra is the *term model* of BPA, whose elements are BPA-expressions (built from the atoms $a,b,c, \ldots$ by means of the basic constructors) modulo the equality generated by the axioms. The term model itself (let us call it $\mathbb{T}$) is not very exciting: it contains only finite processes. In order to specify also infinite processes, we introduce *recursion variables* $X, Y, Z, \ldots$. Using these, one can specify the process $aaaaaa \cdots$ (performing infinitely many consecutive a-steps) by the recursion equation $X = aX$; indeed, by 'unwinding' we have $X = aX = aaX = aaaX = \cdots$. In general, we will admit simultaneous recursion, i.e. systems of recursion equations. A non-trivial example is the following specification of the process behaviour of a Stack with data 0,1:

STACK
$X = 0{\downarrow}.YX + 1{\downarrow}.ZX$
$Y = 0{\uparrow} + 0{\downarrow}.YY + 1{\downarrow}.ZY$
$Z = 1{\uparrow} + 0{\downarrow}.YZ + 1{\downarrow}.ZZ$

TABLE 2

Here $0{\downarrow}$ and $0{\uparrow}$ are the actions 'push 0' and 'pop 0', respectively; likewise for 1. Now Stack is specified by the first recursion variable, X. Indeed, according to the first equation the process X is capable of performing either the action $0{\downarrow}$, after which the process is transformed into YX, or $1{\downarrow}$, after which the process is transformed into ZX. In the first case we have, using the second equation, $YX = (0{\uparrow} + 0{\downarrow}.YY + 1{\downarrow}.ZY)X = 0{\uparrow}{\cdot}X + 0{\downarrow}.YYX + 1{\downarrow}.ZYX$. This means that the process YX has three options; after performing the first one ($0{\uparrow}$) it behaves like the original X. Continuing in this manner we find a transition diagram or *process graph* as in Figure 1.

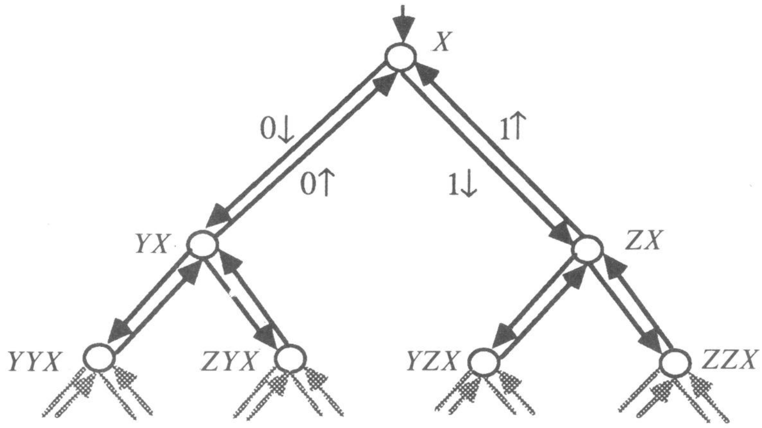


FIGURE 1. Stack

It is not hard to imagine how such a process graph (a rooted, directed, connected, labeled graph) can be associated with a system of recursion equations; we will not give a formal definition here. Actually, one can use such process graphs and build various models *(graph models)* for BPA from them; this will be discussed now.

2. Graph models for BPA

Let **G** be the set of all at most countably branching process graphs $g,h, \ldots$ over the action alphabet $A=\{a,b,c, \ldots\}$. (I.e. a node in such a graph may have at most countably many one-step successors.) On **G** we define operations $+$ and $\cdot$ as follows: $g \cdot h$ is the result of appending (the root of) h at each termination node of g, and $g+h$ is the result of identifying the roots of g and h. (To be more precise, we first have to unwind g and h a little bit so as to make their roots 'acyclic', otherwise the sum would not have the intended interpretation of making an irreversible choice.) Letting **a** be the graph consisting of a single arrow with label a, we now have a structure $\mathcal{G}=\mathbf{G}(+,\cdot,\mathbf{a},\mathbf{b},\mathbf{c}, \ldots)$ which corresponds to the signature of BPA. But it is not a model of BPA. For instance the law $x+x=x$ does not hold in $\mathcal{G}$, since $\mathbf{a}+\mathbf{a}$ is not the same as $\mathbf{a}$; the former is a graph with two arrows and the latter has one arrow.

Here we need the fundamental notion of D. Park (see [13]), called *bisimulation equivalence* or *bisimilarity*. Two graphs g and h are bisimilar if there is a matching between their nodes (i.e. a binary relation with domain the set of nodes of g, and codomain the set of nodes of h) such that (1) the roots are matched; (2) if nodes s,t in g,h respectively are matched and an a-step is possible from s to some s' then in h an a-step is possible from t to some t' *such that s' and t' again are matched* ; (3) likewise with the roles of g,h reversed. A matching satisfying (1-3) is a bisimulation. An example is given in Figure 2, where (part of) the matching is explicitly displayed; another example is given in Figure 3 where the matching is between each pair of nodes on the same horizontal level.

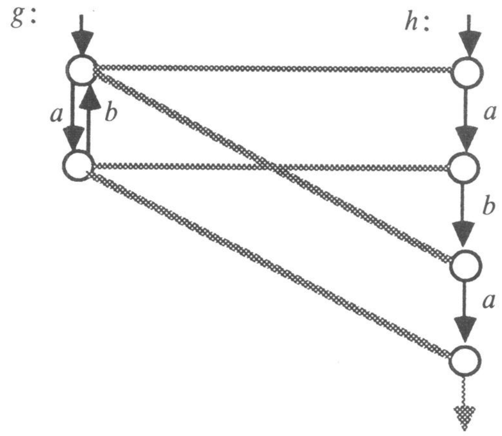
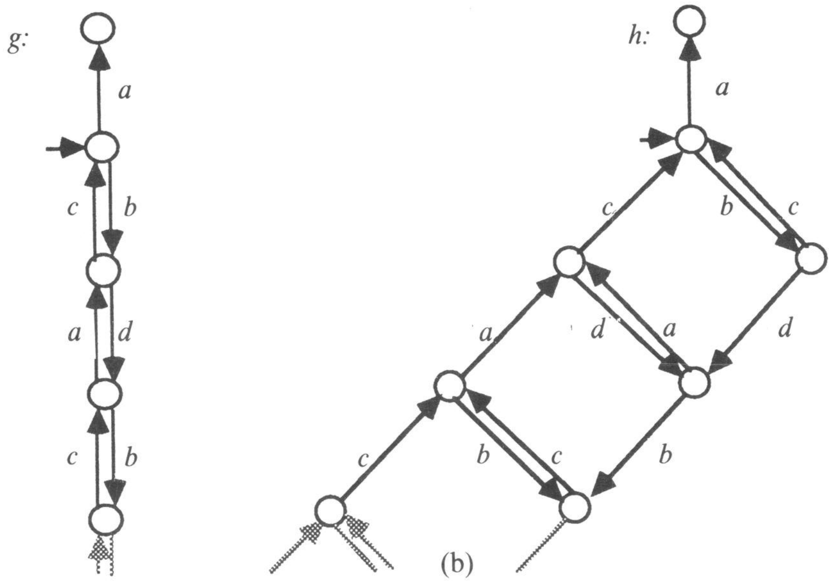


Figure 2



(a)

FIGURE 3

We use the notation $g \underline{\leftrightarrow} h$ to express that g and h are bisimilar. Now one proves that $\underline{\leftrightarrow}$ is not only an equivalence on $\mathbf{G}$, but even a congruence on $\mathcal{G}$. Thus the quotient $\mathbb{G} = \mathcal{G} / \underline{\leftrightarrow}$ is well-defined, and it is a model of BPA. ($\mathbb{G}$ has constants $a = \mathbf{a} / \underline{\leftrightarrow}$ etc., and operations $+, \cdot$ defined by $g + h = (\mathbf{g} + \mathbf{h}) / \underline{\leftrightarrow}$ for $g = \mathbf{g} / \underline{\leftrightarrow}$ and $h = \mathbf{h} / \underline{\leftrightarrow}$; likewise for $\cdot$. (For typographical reasons we will not distinguish between the syntactic $+, \cdot$ and the semantic $+, \cdot$ in our notation.)

Even more, $\mathbb{G}$ is a very nice model of BPA: all systems of recursion equations in the syntax of BPA have a solution in $\mathbb{G}$, and systems of *guarded* recursion equations like in Table 1 have moreover a *unique* solution. 'Guarded' means that in the right-hand sides of the recursion equations no recursion variable can be accessed without passing an atomic action. (E.g. $X = a + X$ is not a guarded equation; it has many solutions: $a + b$, $a + c$, ...)

Some submodels (all satisfying the axioms of BPA) of $\mathbb{G}$ are of interest: $\mathbb{G}_{fb}$, built from *finitely branching* process graphs; $\mathbb{R}$, built from finite (but possibly cycle-containing) graphs; and $\mathbb{F}$, built from finite and acyclic graphs. Also $\mathbb{G}_{fb}$ has the property of providing unique solutions for systems of guarded recursion equations. Without the condition of guardedness, there need not be solutions. E.g. the equation

$$X = Xa + a$$

cannot be solved in $\mathbb{G}_{fb}$. In the model $\mathbb{R}$ of regular processes one can always find unique solutions for guarded recursion equations provided they are *linear,* that is, the expressions (terms) in the equations may only be built by sum and a restricted form of product called *prefix multiplication* $a \cdot s$ ('a' an atom, s a general expression) which excludes products of recursion variables as in Table 1. For a complete proof system for regular processes, see [11].

EXAMPLE

$$\{X = aX + bY, Y = cX + dY\}$$

is a linear system;

$$\{X = aXX + bY, Y = cX + dYXY\}$$

is not.

The model $\mathbb{R}$ contains the *finite-state* processes; hence the notation $\mathbb{R}$ for 'regular' as in formal language theory. Finally, $\mathbb{F}$ contains only finite processes and is in fact isomorphic to the term model $\mathbb{T}$.

Some systems of recursion equations should be taken as equivalent. Clearly, $X = aX$ and $X = aaX$ specify the same process in $\mathbb{G}$. Less clearly, the two systems

$$E_1 = \{X = a + bYX,\ Y = c + dXY\}$$

$$E_2 = \{X = a + bU,\ U = cX + dZX,\ Y = c + dZ,\ Z = aY + bUY\}$$

are equivalent in this sense: E_1 specifies the process graph in Figure 3a above, and E_2 specifies the graph in Figure 3b. Moreover, as we already saw, these two graphs are bisimilar. So E_1 and E_2 denote the same process in $\mathbb{G}$. So the question arises: *Is equivalence of recursion equations over* BPA, *relative to the graph model* $\mathbb{G}$, *decidable?* At the moment this question is wide open. There is an interesting connection here with *context-free languages,* as follows.

A guarded system of recursion equations over BPA corresponds in an obvious way (for details see [2]) to a context-free grammar (CFG) in Greibach Normal Form, and vice versa. Hence each context-free language (CFL) can be obtained as the set of finite traces of a process in $\mathbb{G}$ denoted by a system of guarded recursion equations. (A finite trace is the word obtained by following a path from the root to a termination node.) In fact, to generate a CFL it is sufficient to look at certain restricted systems of recursion equations called 'normed'. A system is normed if in every state (of the corresponding process) there is a possibility to terminate. E.g. $X = aX$ is not normed, but $X = b + aX$ is. There is a simple syntactical check to determine whether a system is normed or not. Clearly, the property 'normed' also pertains to process graphs. In [2] it is proved that the equivalence problem stated above is solvable for such normed systems. This is rather surprising in view of the well-known fact that the equality problem for CFLs is unsolvable. The point is that the process semantics in $\mathbb{G}$ of a CFG bears much more information than the trace set semantics, which is an abstraction from the process semantics.

The link with deterministic context-free languages resides in the following observation from [2]:

THEOREM 2.1. *Let* $g,h \in \mathbf{G}$ *be two normed and deterministic process graphs. Then* $g \leftrightarrow h$ *iff* g *and* h *have the same sets of finite traces.*

Here a graph is 'deterministic' if two arrows leaving the same node always have different label. The CFL (i.e. the set of finite traces) determined by a normed and deterministic graph, corresponding to a system of guarded recursion equations in BPA, is known as a *simple* CFL; the simple CFLs form a proper subclass of the deterministic CFLs.

Summarizing, we can state that BPA and its graph model obtained via the concept of bisimulation provide a new angle on some problems in the theory of formal languages, concerned with context-free languages. Here we think especially of *deterministic* context-free languages (DCFLs), obtained by deterministic push-down automata, with the well-known open problem whether the equality problem for DCFLs is solvable. Thus, even in the absence of the many operators for parallellism, abstraction etc. which are still to be introduced below, we have in BPA and its models an interesting theory with potential implications for the DCFL problem.

3. DEADLOCK

After the excursion to semantics in the preceding section we return to the development of more syntax for processes. A vital element in the present set-up of process algebra is the process δ, signifying 'deadlock'. The process ab performs its two steps and then terminates, succesfully; but the process $ab\delta$ *deadlocks* after the a- and b-action: it wants to do a proper (i.e. non-δ) action but it cannot. So δ is the acknowledgement of stagnation. With this in mind, the axioms to which δ is subject, may be clear:

DEADLOCK
$\delta + x = x$
$\delta \cdot x = \delta$

TABLE 3

The axiom system of BPA (Table 1) together with the present axioms for δ is called BPA_δ. We are now in a position to motivate the absence in BPA of the 'other' distributive law: $z(x+y) = zx + zy$. For, suppose it would be added. Then $ab = a(b+\delta) = ab + a\delta$. This means that a process with deadlock possibility is equal to one without, conflicting with our intention to model also deadlock behaviour of processes.

The essential role of the new process δ will only be fully appreciated after the introduction of communication, below.

4. The merge operator

If x,y are processes, their 'parallel composition' $x \| y$ is the process that first chooses whether to do a step in x or in y, and proceeds as the parallel composition of the remainders of x,y. In other words, the steps of x,y are interleaved or merged. Using an auxiliary operator $|\!\lfloor$ (with the interpretation that $x |\!\lfloor y$ is like $x \| y$ but with the commitment of choosing the initial step from x) the operation $\|$ can be succinctly defined by the axioms:

FREE MERGE
$x \| y = x |\!\lfloor y + y |\!\lfloor x$
$ax |\!\lfloor y = a(x \| y)$
$a |\!\lfloor y = ay$
$(x+y) |\!\lfloor z = x |\!\lfloor z + y |\!\lfloor z$

Table 4

The system of nine axioms consisting of BPA and the four axioms for merge will be called PA. Moreover, if the axioms for δ are added, the result will be PA_δ. The operators $\|$ and $|\!\lfloor$ will also be called *merge* and *left-merge* respectively.

The merge operator corresponds to what in the theory of formal languages is called *shuffle*. The shuffle of the words *ab* and *cd* is the set of words $\{abcd, acbd, cabd, acdb, cadb, cdab\}$. Merging the processes *ab* and *cd* yields the process

$$\begin{aligned} ab \| cd &= ab |\!\lfloor cd + cd |\!\lfloor ab = a(b \| cd) + c(d \| ab) \\ &= a(b |\!\lfloor cd + cd |\!\lfloor b) + c(d |\!\lfloor ab + ab |\!\lfloor d) \\ &= a(bcd + c(d \| b)) + c(dab + a(b \| d)) \\ &= a(bcd + c(db + bd)) + c(dab + a(bd + db)), \end{aligned}$$

a process having as trace set the shuffle above.

An example of a process recursively defined in PA, is $X = a(b \| X)$. It turns out that this process can already be defined in BPA, by the system of recursion equations

$$\{X = aYX, Y = b + aYY\}.$$

To see that both ways of defining X yield the same process, one may 'unwind' according to the given equations:

$$\begin{aligned} X &= a(b \| X) = a(b |\!\lfloor X + X |\!\lfloor b) = a(bX + a(b \| X) |\!\lfloor b) \\ &= a(bX + a((b \| X) \| b)) \\ &= a(bX + a \ldots), \end{aligned}$$

while on the other hand

$$X = aYX = a(b + aYY)X = a(bX + aYYX) = a(bX + a\ldots).$$

So at least up to level 2 the processes are equal. By further unwinding they can be proved equal up to each finite level.

Yet there are processes definable in PA but not in BPA. An example (from [4]) of such a process is given by the recursion equation

$$X = 0{\downarrow}\cdot(0{\uparrow} \| X) + 1{\downarrow}\cdot(1{\uparrow} \| X)$$

describing the process behaviour of a Bag (or multiset), in which arbitrarily many instances of the data 0,1 can be inserted (the actions $0{\downarrow}, 1{\downarrow}$ respectively) or retrieved ($0{\uparrow}, 1{\uparrow}$), with the restriction that no more 0's and 1's can taken from the Bag than were put in first. The difference with a Stack or a Queue is that all order between incoming and outgoing 0's and 1's is lost. The process graph corresponding to the process Bag is as in Figure 4.

We conclude this section on PA by mentioning the following fact (see [4]), which is useful for establishing non-definability results:

THEOREM 4.1. *Every process which is recursively defined in PA and has an infinite trace, has an eventually periodic trace.*

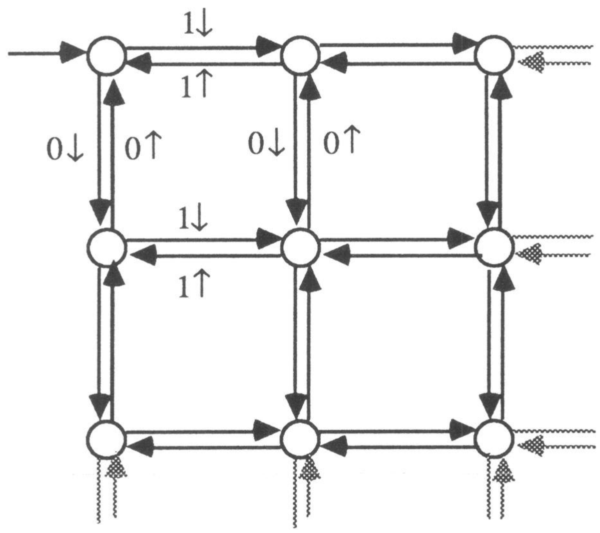


FIGURE 4. Bag

5. COMMUNICATION

So far, the parallel composition or merge ($\|$) did not involve communication in the process $x\|y$: one could say that x and y are 'freely' merged or interleaved. However, some actions in one process may need an action in another process for an actual execution, like the act of shaking hands requires simultaneous acts of two persons. In fact, 'handshaking' is the paradigm for the type of communication which we will introduce now. If $A=\{a,b,c,\ldots,\delta\}$ is the action alphabet, let us adopt a binary communication function $|:A \times A \rightarrow A$ satisfying the axioms in Table 5.

COMMUNICATION FUNCTION
$a\|b=b\|a$
$(a\|b)\|c=a\|(b\|c)$
$\delta\|a=\delta$

TABLE 5

Here a,b vary over A, including δ. We can now specify *merge with communication* ; we use the same notation $\|$ as for the 'free' merge in Section 4 since in fact 'free' merge is an instance of merge with communication by choosing the communication function trivial, i.e. $a|b=\delta$ for all $a,b \in A$. There are now two auxiliary operators, allowing a finite axiomatisation: left-merge ($⌊\!⌊$) as before and $|$ *(communication merge* or simply 'bar'), which is an extension of the communication function in Table 5 to all processes, not only the atoms. The axioms for $\|$ and its auxiliary operators are given in Table 6.

MERGE WITH COMMUNICATION
$x\|\|y=x⌊\!⌊y+y⌊\!⌊x+x\|y$
$ax⌊\!⌊y=a(x\|\|y)$
$a⌊\!⌊y=ay$
$(x+y)⌊\!⌊z=x⌊\!⌊z+y⌊\!⌊z$
$ax\|b=(a\|b)x$
$a\|bx=(a\|b)x$
$ax\|by=(a\|b)(x\|\|y)$
$(x+y)\|z=x\|z+y\|z$
$x\|(y+z)=x\|y+x\|z$

TABLE 6

We also need the so-called *encapsulation* operators ∂_H (for every $H \subseteq A$) for removing unsuccessful attempts at communication:

ENCAPSULATION
$\partial_H(a)=a$ if $a \notin H$
$\partial_H(a)=\delta$ if $a \in H$
$\partial_H(x+y)=\partial_H(x)+\partial_H(y)$
$\partial_H(xy)=\partial_H(x)\cdot\partial_H(y)$

TABLE 7

These axioms express that ∂_H 'kills' all atoms mentioned in H, by replacing them with δ. The axioms for BPA, DEADLOCK together with the present ones in Tables 5-7 constitute the axiom system ACP (Algebra of Communicating Processes). Typically, a system of communicating processes $x_1, \ldots, x_n$ is now represented in ACP by the expression $\partial_H(x_1 \| \cdots \| x_n)$. Prefixing the encapsulation operator says that the system $x_1, \ldots, x_n$ is to be perceived as a separate unit with respect to the communication actions mentioned in H; no communications between actions in H with an environment are expected or intended.

A useful theorem to break down such expressions is the *Expansion Theorem* (first formulated by Milner, for the case of CCS; see [12]) which holds under the assumption of the *handshaking axiom* $x|y|z=\delta$. This axiom says that all communications are binary. (In fact we have to require associativity of '$\|$' first - see Table 8.)

THEOREM 5.1 (EXPANSION THEOREM).

$$x_1 \| \cdots \| x_k = \sum_i x_i \mathbin{⫗} X_k^i + \sum_{i \neq j} (x_i | x_j) \mathbin{⫗} X_k^{i,j}$$

Here X_k^i denotes the merge of $x_1, \ldots, x_k$ except x_i, and $X_k^{i,j}$ denotes the same merge except x_i, x_j $(k \geqslant 3)$. For instance, for $k=3$:

$$x\|y\|z = x \mathbin{⫗} (y\|z) + y \mathbin{⫗} (x\|x) + z \mathbin{⫗} (x\|y) + (y|z) \mathbin{⫗} x + (z|x) \mathbin{⫗} y + (x|y) \mathbin{⫗} z.$$

In order to prove the Expansion Theorem, one first proves by simultaneous induction on term complexity that for all closed ACP-terms (i.e. ACP-terms without free variables) the following *axioms of standard concurrency* hold:

AXIOMS OF STANDARD CONCURRENCY
$(x \llfloor y) \llfloor z = x \llfloor (y \| z)$
$(x \mid y) \llfloor z = x \mid (y \llfloor z)$
$x \mid y = y \mid x$
$x \| y = y \| x$
$x \mid (y \mid z) = (x \mid y) \mid z$
$x \| (y \| z) = (x \| y) \| z$

TABLE 8

As in Section 2 one can construct graph models $\mathbb{G}, \mathbb{G}_{fb}, \mathbb{R}, \mathbb{F}$ for ACP; in these models the axioms in Table 8 are valid. We will discuss the construction of these models in Section 7. (It is however also possible to construct 'non-standard' models of ACP in which these axioms do not hold. We will not be interested in such pathological models.)

The defining power of ACP is strictly greater than that of PA. The following is an example (from [4]) of a process U, recursively defined in ACP, but not definable in PA: let the alphabet be $\{a,b,c,d,\delta\}$ and let the communication function be given by $c|c=a$, $d|d=b$, and all other communications equal to δ. Let $H=\{c,d\}$. Now we recursively define the process U as in Table 9:

$U = \partial_H(dcY \| Z)$
$X = cXc + d$
$Y = dXY$
$Z = dXcZ$

TABLE 9

Then, we claim, $U = ba(ba^2)^2(ba^3)^2(ba^4)^2 \cdots$. Indeed, using the axioms in ACP and putting

$$U_n = \partial_H(dc^n Y \| Z)$$

for $n \geq 1$, a straightforward computation shows that

$$U_n = ba^n ba^{n+1} U_{n+1}.$$

By Theorem 4.1, U is not definable in PA, since the one infinite trace of U is not eventually periodic.

We will often adopt a special format for the communication function, called *read-write communication*. Let a finite set D of *data* d and a set $\{1, \ldots, p\}$ of *ports* be given. Then the alphabet consists of *read* actions $ri(d)$ and *write* actions $wi(d)$, for $i=1, \ldots, p$ and $d \in D$. The interpretation is: read datum d at port i, write datum d at port i respectively. Furthermore, the alphabet contains actions $ci(d)$ for $i=1, \ldots, p$ and $d \in D$, with interpretation: *communicate* *d at i*. These actions will be called *transactions*. The only non-trivial communications (i.e. not resulting in δ) are: $wi(d) | ri(d) = ci(d)$. Instead of $wi(d)$ we will also use the notation $si(d)$ (send d along i). Note that read-write communication satisfies the handshaking axiom: all communications are binary.

EXAMPLE 5.1.
Using the present read-write communication format we can write the recursion equation for a Bag B_{12} (cf. Section 4) which reads data $d \in D$ at port 1 and writes them at port 2 as follows:

$$B_{12} = \sum_{d \in D} r1(d)(w2(d) \| B_{12}).$$

6. ABSTRACTION

A fundamental issue in the design and specification of hierarchical (or modularized) systems of communicating processes is *abstraction*. Without having an abstraction mechanism enabling us to abstract from the inner workings of modules to be composed to larger systems, specification of all but very small systems would be virtually impossible. We will now extend the axiom system ACP, obtained thus far, with such an abstraction mechanism.

Consider two Bags B_{12}, B_{23} (cf. Example 5.1) with action alphabets $\{r1(d), s2(d) | d \in D\}$ and $\{r2(d), s3(d) | d \in D\}$, respectively. That is, B_{12} is a bag-like channel reading data d at port 1, sending them to port 2; B_{23} reads data at 2 and sends them to 3. (That the channels are bags means that, unlike the case of a queue, the order of incoming data is lost in the transmission.) Suppose the bags are connected at port 2; so we adopt communications $s2(d) | r2(d) = c2(d)$ where $c2(d)$ is the transaction of d at 2.

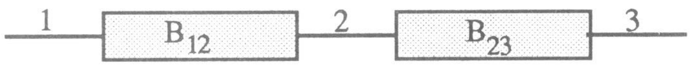


FIGURE 5. Transparent Bag $\mathbb{B}_{13}$

The composite system $\mathbb{B}_{13} = \partial_H(B_{12} \| B_{23})$ where $H = \{s2(d), r2(d) | d \in D\}$, should, intuitively, be again a Bag between ports 1,3. However, from some (rather involved) calculations we learn that

$$\mathbb{B}_{13} = \sum_{d \in D} r1(d) \cdot (c2(d) \cdot s3(d)) \| \mathbb{B}_{13}).$$

So $\mathbb{B}_{13}$ is a 'transparent' Bag: the passage of d through 2 is visible as the transaction event $c2(d)$. (Note that this terminology conflicts with the usual one in the area of computer networks, where a network is called transparent if the internal structure is *not* visible.)

How can we *abstract* from such internal events, if we are only interested in the external behaviour at 1,3? The first step to obtain such an abstraction is to remove the distinctive identity of the actions to be abstracted, that is, to rename them all into one designated action which we call, after Milner, τ: the *silent* action. This renaming is realised by the *abstraction operator* τ_I, parameterized by a set of actions $I \subseteq A$ and subject to the following axioms:

ABSTRACTION
$\tau_I(\tau)=\tau$
$\tau_I(a)=a$ if $a \notin I$
$\tau_I(a)=\tau$ if $a \in I$
$\tau_I(x+y)=\tau_I(x)+\tau_I(y)$
$\tau_I(xy)=\tau_I(x)\cdot\tau_I(y)$

TABLE 10

The second step is to attempt to devise axioms for the silent step τ by means of which τ can be removed from expressions, as e.g. in the equation $a\tau b=ab$. However, it is not possible to remove *all* τ's in an expression if one is interested in a faithful description of deadlock behaviour of processes (at least in bisimulation semantics, the framework adopted in this paper). For, consider the process (expression) $a+\tau\delta$; this process can deadlock, namely if it chooses to perform the silent action. Now, if one would propose naively the equations $\tau x = x\tau = x$, then $a+\tau\delta = a+\delta=a$, and the latter process has no deadlock possibility. It turns out that one of the proposed equations, $x\tau = x$, can be safely adopted, but the other one is wrong. Fortunately, R. Milner has devised some simple axioms which give a complete description of the properties of the silent step (complete with respect to a certain semantical notion of process equivalence called $r\tau\delta$-bisimulation, which does respect deadlock behaviour; this notion is discussed below), as follows.

SILENT STEP
$x\tau=x$
$\tau x = \tau x + x$
$a(\tau x+y)=a(\tau x+y)+ax$

TABLE 11

To return to our example of the 'transparent' Bag $\mathbb{B}_{13}$, after abstraction of the set of transactions $I = \{c\,2(d) \mid d \in D\}$ the result is indeed an 'ordinary' Bag:

$$\tau_I(\mathbb{B}_{13})=\tau_I(\sum_{d \in D} r\,1(d)(c\,2(d)\cdot s\,3(d) \| \mathbb{B}_{13})) \qquad (*)$$

$$= \sum_{d \in D} r\,1(d)(\tau\cdot s\,3(d) \| \tau_I(\mathbb{B}_{13})) = \sum_{d \in D} (r\,1(d)\cdot\tau\cdot s\,3(d)) \lfloor\!\lfloor \tau_I(\mathbb{B}_{13})$$

$$= \sum_{d \in D} (r\,1(d)\cdot s\,3(d)) \lfloor\!\lfloor \tau_I(\mathbb{B}_{13}) = \sum_{d \in D} r\,1(d)(s\,3(d) \| \tau_I(\mathbb{B}_{13}))$$

from which it follows that $\tau_I(\mathbb{B}_{13}) = B_{13}$ (**), the Bag defined by

$$B_{13} = \sum_{d \in D} r1(d)(s3(d) \| B_{13}).$$

Here we were able to eliminate all silent actions, but this will not always be the case. For instance, 'chaining' two Stacks instead of Bags as in Figure 5 yields a process with 'essential' τ-steps. Likewise for a Bag followed by a Stack. (Here 'essential' means: non-removable in bisimulation semantics.) In fact, the computation above is not as straightforward as was suggested: to justify the equations marked with (*) and (**) we need additional proof principles. As to (**), this equation is justified by the *Recursive Specification Principle* (RSP) stating that a *guarded system of recursion equations in which no abstraction operator τ_I appears, has a unique solution.* We will not discuss the justification of equation (*) here. The justification of a principle like RSP is that it is valid in all 'sensible' models of our axioms; however note that for formal computations one has to postulate such a principle explicitly.

Combining all the axioms presented above in Tables 1,3,4,5,6,7,10,11 and a few axioms specifying the interaction between τ and communication merge $|$, we have arrived at the system ACP_τ, *Algebra of Communicating Processes with abstraction* (see Table 12).

Actually, in spite of our restriction to specification of processes as stated in the Introduction, the last computation concerned a very simple process *verification,* showing that the combined system has the desired external behaviour of a Bag. Abstraction, realized in ACP_τ by the abstraction operator and the silent process τ, clearly is of crucial importance for process verification. But also for process specification abstraction is important. Let $f: \mathbb{N} \rightarrow \{a,b\}$ be a sequence of symbols a,b, and let p_f be the proces $f(0) \cdot f(1) \cdot f(2) \ldots$, that is, the unique solution of the infinite system of recursion equations $\{X_n = f(n) \cdot X_{n+1} | n \geqslant 0\}$. Now we have:

THEOREM 6.1. *There is a computable function f such that process p_f is not definable by a finite system of recursion equations in* ACP_τ *without abstraction operator.*

On the other hand, according to the Finite Specification Theorem 8.1, every process p_f with computable f *is* definable by a finite system of recursion equations in full ACP_τ.

ACP$_\tau$			
$x+y=y+x$	A1	$x\tau=x$	T1
$x+(y+z)=(x+y)+z$	A2	$\tau x+x=\tau x$	T2
$x+x=x$	A3	$a(\tau x+y)=a(\tau x+y)+ax$	T3
$(x+y)z=xz+yz$	A4		
$(xy)z=x(yz)$	A5		
$x+\delta=x$	A6		
$\delta x=\delta$	A7		
$a\vert b=b\vert a$	C1		
$(a\vert b)\vert c=a\vert(b\vert c)$	C2		
$\delta\vert a=\delta$	C3		
$x\Vert y=x \mathbin{\lfloor\!\!\lfloor} y+y \mathbin{\lfloor\!\!\lfloor} x+x\vert y$	CM1		
$a \mathbin{\lfloor\!\!\lfloor} x=ax$	CM2	$\tau \mathbin{\lfloor\!\!\lfloor} x=\tau x$	TM1
$ax \mathbin{\lfloor\!\!\lfloor} y=a(x\Vert y)$	CM3	$\tau x \mathbin{\lfloor\!\!\lfloor} y=\tau(x\Vert y)$	TM2
$(x+y) \mathbin{\lfloor\!\!\lfloor} z=x \mathbin{\lfloor\!\!\lfloor} z+y \mathbin{\lfloor\!\!\lfloor} z$	CM4	$\tau\vert x=\delta$	TC1
$ax\vert b=(a\vert b)x$	CM5	$x\vert\tau=\delta$	TC2
$a\vert bx=(a\vert b)x$	CM6	$\tau x\vert y=x\vert y$	TC3
$ax\vert by=(a\vert b)(x\Vert y)$	CM7	$x\vert\tau y=x\vert y$	TC4
$(x+y)\vert z=x\vert z+y\vert z$	CM8		
$x\vert(y+z)=x\vert y+x\vert z$	CM9	$\partial_H(\tau)=\tau$	DT
		$\tau_I(\tau)=\tau$	TI1
$\partial_H(a)=a$ if $a\notin H$	D1	$\tau_I(a)=a$ if $a\notin I$	TI2
$\partial_H(a)=\delta$ if $a\in H$	D2	$\tau_I(a)=\tau$ if $a\in I$	TI3
$\partial_H(x+y)=\partial_H(x)+\partial_H(y)$	D3	$\tau_I(x+y)=\tau_I(x)+\tau_I(y)$	TI4
$\partial_H(xy)=\partial_H(x)\cdot\partial_H(y)$	D4	$\tau_I(xy)=\tau(x)\cdot\tau_I(y)$	TI5

TABLE 12

7. GRAPH MODELS FOR ACP$_\tau$

We will now construct graph models for ACP$_\tau$, in analogy with the construction of these models for BPA in Section 2. Again we start with a domain of at most countably branching process graphs **G**, the only difference being that arrows may now also bear label τ and δ. (By abuse of language we use the same notation **G**.) Next, we define on **G** in addition to $+,\cdot$ operations $\Vert, \mathbin{\lfloor\!\!\lfloor}, \vert, \boldsymbol{\tau_I}, \boldsymbol{\partial_H}$ corresponding to the syntactic operations $\Vert, \mathbin{\lfloor\!\!\lfloor}, \vert, \tau_I, \partial_H$. We will only discuss the definition of the first operation $\Vert$. Let **ab** and **cd** be two process graphs as in Figure 6, and suppose there are communications $a\vert d=f$ and $b\vert c=k$, all other communications being trivial (i.e. resulting in δ). Then **ab**$\Vert$**cd** is the process graph indicated in Figure 6, a cartesian product with diagonal edges for the successful communications.

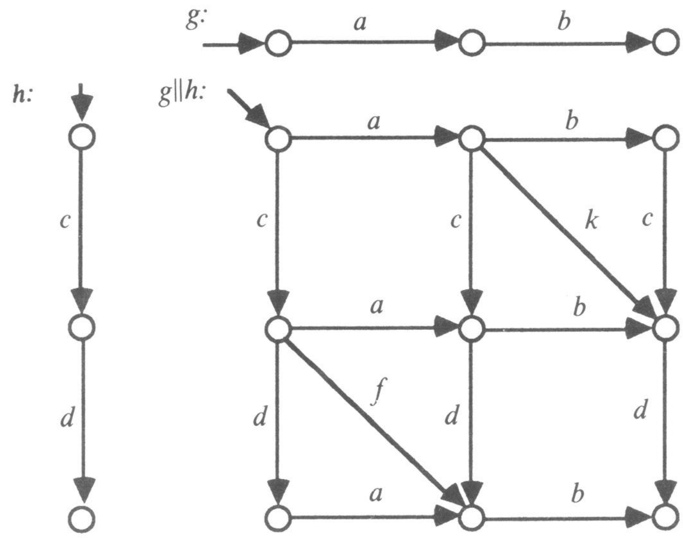


FIGURE 6

We now have a structure $\mathcal{G}=\mathbf{G}(+,\cdot,\|,\llfloor,|,\tau_I,\partial_H,\tau,\delta,\mathbf{a},\mathbf{b},\mathbf{c},\ldots)$, which is not yet a model of ACP_τ but becomes so after dividing out the congruence *$r\tau\delta$-bisimilarity* (notation: $\leftrightarrow_{r\tau\delta}$), a generalization of the 'ordinary' bisimilarity $\leftrightarrow$ of Section 2. Here we say that $g \leftrightarrow_{r\tau\delta} h$ if there is a relation between the nodes of g and the nodes of h such that (1) the roots are related; (2) a non-root node is only related to non-root nodes; (3) if nodes s,t in g,h respectively are related and there is in g an a-step from s to some s', then there is in g a path $\tau\tau\tau \cdots \tau a \tau\tau \cdots \tau$ (i.e. zero or more τ-steps followed by an a-step followed by zero or more τ-steps) from t to some t' such that s' and t' are again related; (4) as (3) with the roles of g,h interchanged. (See for an example of such a $r\tau\delta$-bisimulation Figure 7.) Again, this equivalence is a congruence on $\mathcal{G}$ and putting $\mathbb{G}=\mathcal{G}/\leftrightarrow_{r\tau\delta}$ we have a model for ACP_τ, in which all systems of guarded recursion equations have a solution, and even a unique solution if abstraction operators are absent from the system.

As before in Section 2, $\mathbb{G}$ has submodels $\mathbb{R},\mathbb{F}$ (regular and finite processes, respectively). Remarkably, as observed in [1], there is no model $\mathbb{G}_{fb}$ based on all finitely branching graphs now. (For ACP such a model does exist.) The reason is that there is no structure $\mathcal{G}_{fb}$, since $\mathbf{G}_{fb}$ is not closed under the operations $\|,\llfloor,|,\tau_I$. The auxiliary operator $|$ is the culprit here.

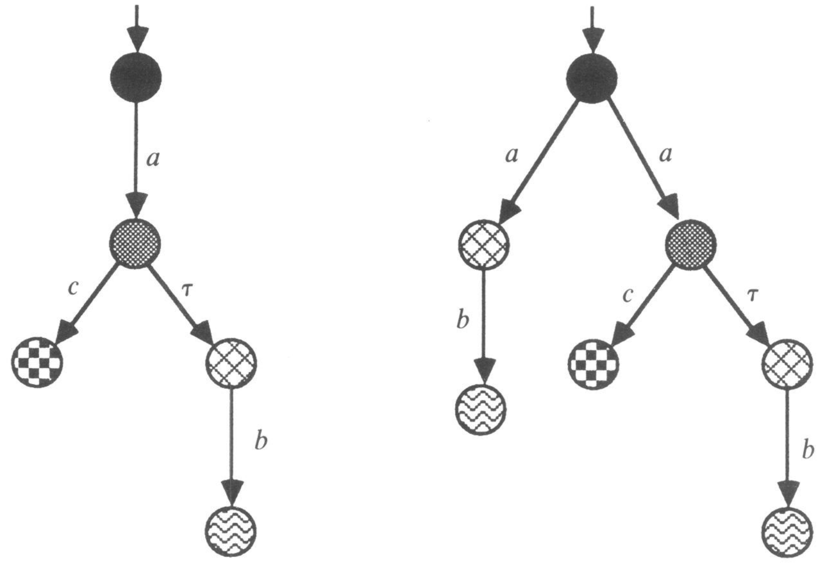


FIGURE 7. Example of $r\tau\delta$-bisimulation: nodes of the same colour are related

8. THE FINITE SPECIFICATION THEOREM

ACP_τ is a powerful specification mechanism; in a sense it is a universal specification mechanism: *every finitely branching, computable process in the graph model* $\mathbb{G}$ *can be finitely specified* in ACP_τ. (We use the word 'specification' for 'system of recursion equations'.) We have to be more precise about the notion of 'computable process'. First, an intuitive explanation: suppose a finitely branching process graph $g \in \mathbf{G}$ is 'actually' given; the labels may include τ, and there may be even infinite τ-traces. That g is 'actually' given means that the process graph g must be 'computable': g can be described by some coding of the nodes in natural numbers and recursive functions giving in-degree, out-degree, edge-labels, etc. This notion of a computable process graph is rather obvious, and we will not give details of the definition here.

Now even if the computable graph g is an infinite process graph, it can trivially be specified by an infinite computable specification, as follows. First rename all τ-edges in g to t-edges, for a 'fresh' atom t. Call the resulting process graph: g_t. Next assign to each node s of g_t a recursion variable X_s and write down the recursion equation for X_s according to the outgoing edges of node s. Let X_{s_0} be the variable corresponding to the root s_0 of g_t. As g is computable, g_t is computable and the resulting 'direct' specification

$$E = \{X_s = T_s(\mathbf{X}) | s \in \mathrm{NODES}(g_t)\}$$

is evidently also computable (i.e.: the nodes can be numbered as s_n ($n \geqslant 0$), and after coding the sequence e_n of codes of equations $E_n : X_{s_n} = T_{s_n}(\mathbf{X})$ is a computable sequence). Now the infinite specification which uniquely determines g, is simply: $\{Y = \tau_{\{t\}}(X_{s_0})\} \cup E$. In fact all specifications below will have the form $\{X = \tau_I(X_0),\ X_n = T_n(\mathbf{X}) | n \geq 0\}$ where the guarded expressions $T_n(X)$ $(= T_n(X_{i_1}, \ldots, X_{i_n}))$ contain no abstraction operators τ_J. They may contain all other process operators. We will say that such specifications have *restricted abstraction.*

However, we want more than a computable infinite specification with restricted abstraction: to describe process graph g we would like to find a *finite* specification with restricted abstraction for g. Indeed this is possible:

THEOREM 8.1 (FINITE SPECIFICATION THEOREM). *Let the finitely branching and computable process graph g determine* **g** *in the graph model* $\mathbb{G}$ *of* ACP$_\tau$. *Then there is a finite specification with restricted abstraction E in* ACP$_\tau$ *such that* $[\![E]\!] = g$. *Here* $[\![E]\!]$ *is the solution of E in* $\mathbb{G}$.

The proof in [1] is by constructing a Turing machine in ACP$_\tau$; the 'tape' is obtained by glueing together two Stacks as defined in Table 2. There does not seem to be an essential difficulty in removing the condition 'finitely branching' in the theorem, in favour of 'at most countably branching'.

9. CONCLUDING REMARKS

Even though the Finite Specification Theorem declares the set of operators of ACP$_\tau$ to be sufficient for all specifications, in practice one will need more operators to make specifications not only theoretically but also practically possible. Therefore some additional operators have been defined and studied in the present branch of process algebra, notably an operator by means of which different priorities can be given to different atomic actions, and a state operator taking into account information from a suitable state space. Using priorities imposed on atomic actions enables us to model interrupts in a system of communicating processes; the state operator has turned out to be indispensable in the construction of process algebra semantics for some object-oriented programming languages. For these developments we refer to [6]. Lately, some thorough studies have been made about extending ACP$_\tau$ with some new constants: ϵ for the empty process and η for an alternative to the silent step τ ([16,3]). The typical equation here is $\tau = \eta + \epsilon$.

A substantial amount of effort has been invested in extending ACP$_\tau$ to a suitable framework also for process verification, which was barely discussed in the present paper. Process verifications have been realized now for several non-trivial protocols ([14,9]), and recently also for some systolic algorithms ([10,15]) for tasks like palindrome recognition, matrix-vector multiplication. Some positive experience was also obtained using process algebra for the specification and verification of a simple production control system for a configuration of workcells.

Finally we mention that bisimulation semantics, as adopted in the present paper, is by no means the only process semantics. It is possible to identify many processes which are different in bisimulation semantics while still retaining an adequate description of relevant aspects such as deadlock behaviour, leading for instance to readiness semantics or failure semantics, embodying different views on processes. For a study in this area we refer to [7]. For an investigation of models of ACP_τ based on Petri Nets, see [8].

REFERENCES

1. J.C.M. BAETEN, J.A. BERGSTRA, J.W. KLOP (1987). On the consistency of Koomen's Fair Abstraction Rule. *TCS 51 (1/2),* 129-176.
2. J.C.M. BAETEN, J.A. BERGSTRA, J.W. KLOP (1987). Decidability of bisimulation equivalence for processes generating context-free languages. J.W. DE BAKKER, A.J. NIJMAN, P.C. TRELEAVEN (eds.). *Proceedings of the PARLE Conference, Eindhoven 1987, Vol. II,* Springer LNCS 259, 94-113.
3. J.C.M. BAETEN, R.J. VAN GLABBEEK (1987). *Abstraction and Empty Process in Process Algebra,* CWI Report CS-R8721, Centre for Mathematics and Computer Science, Amsterdam.
4. J.A. BERGSTRA, J.W. KLOP (1984). The algebra of recursively defined processes and the algebra of regular processes. J. PAREDAENS (ed.). *Proc. 11th ICALP,* Antwerpen 1984, Springer LNCS 172, 82-95.
5. J.A. BERGSTRA, J.W. KLOP (1986). Algebra of communicating processes. J.W. DE BAKKER, M. HAZEWINKEL, J.K. LENSTRA (eds.). *CWI Monograph* I, *Proceedings of the CWI Symposium Mathematics and Computer Science,* North-Holland, Amsterdam, 89-138.
6. J.A. BERGSTRA, J.W. KLOP (1986). Process algebra: specification and verification in bisimulation semantics. M. HAZEWINKEL, J.K. LENSTRA, L.G.L.T. MEERTENS (eds.). *CWI Monograph 4, Proceedings of the CWI Symposium Mathematics and Computer Science* II, North-Holland, Amsterdam, 61-94.
7. J.A. BERGSTRA, J.W. KLOP, E.-R. OLDEROG (1987). Failures without chaos: a new process semantics for fair abstraction. M. WIRSHING (ed.). *Proceedings IFIP Conference on Formal Description of Programming Concepts, Gl. Avernaes 1986,* North-Holland, Amsterdam, 77-103.
8. R.J. VAN GLABBEEK, F.W. VAANDRAGER (1987). Petri net models for algebraic theories of concurrency. J.W. DE BAKKER, A.J. NIJMAN, P.C. TRELEAVEN (eds.). *Proc. PARLE Conference, Eindhoven 1987, Vol. II,* Springer LNCS 259, 224-242.
9. C.P.J. KOYMANS, J.C. MULDER (1986). *A Modular Approach to Protocol Verification using Process Algebra,* Logic Group Preprint Series Nr.6, Dept. of Philosophy, State University of Utrecht.
10. L. KOSSEN, W.P.WEIJLAND (1987). *Correctness Proofs for Systolic Algorithms: Palindromes and Sorting,* Report FVI 87-04, Computer Science Department, University of Amsterdam.
11. S. MAUW (1987). *A Constructive Version of the Approximation Induction Principle,* Report FVI 87-09, Computer Science Department, University of

Amsterdam.
12. R. MILNER (1980). *A Calculus of Communicating Systems*, Springer LNCS 92.
13. D. PARK (1981). Concurrency and automata on infinite sequences. *Proc. 5th GI Conference*, Springer LNCS 104.
14. F.W. VAANDRAGER (1986). *Verification of Two Communication Protocols by Means of Process Algebra*, CWI Report CS-R8608, Centre for Mathematics and Computer Science, Amsterdam.
15. W.P.WEIJLAND (1987). *A Systolic Algorithm for Matrix-Vector Multiplication*, Report FVI 87-08, Computer Science Department, University of Amsterdam.
16. J.L.M. VRANCKEN (1986). *The Algebra of Communicating Processes with Empty Process*, Report FVI 86-01, Computer Science Department, University of Amsterdam.

Translating Programs into Delay-Insensitive Circuits

Jo C. Ebergen
Dept. of Computer Science and Mathematics
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven

1. Introduction

In 1938 Claude E. Shannon wrote his seminal paper [23] entitled 'A Symbolic Analysis of Relay and Switching Circuits'. He demonstrated that Boolean algebra could be used elegantly in the design of switching circuits. The idea was to specify a circuit by a set of Boolean equations, to manipulate these equations by means of a calculus, and to realize this specification by a connection of basic elements. The result was that only a few basic elements, or even one element such as the 2-input NAND gate, suffice to synthesize any switching function specified by a set of Boolean equations. Shannon's idea proved to be very fertile and out of it grew a complete theory, called switching theory, which is used by most circuit designers nowadays.

In the thesis [5] a method is presented for designing *delay-insensitive circuits.* (Operationally speaking, a delay-insensitive circuit is a connection of basic elements whose functional operation is insensitive to delays in wires or elements.) The principal idea of this method is similar to that in Shannon's paper: to design a circuit as a connection of basic elements and to construct this connection with the aid of a formalism. The method of constructing such a circuit, as described in [5], is by translating programs satisfying a certain syntax. The result of such a translation is a delay-insensitive connection of elements chosen from a finite set of basic elements. Moreover, this translation has the property that the number of basic elements in the connection is proportional to the length of the program. Furthermore, in [5] a rigorous formalization is given of what it means for such a connection to be delay-insensitive.

In this paper[1] we briefly describe some of the history of designing delay-

1. The research reported in this paper was carried out while the author was working at CWI in Amsterdam.

insensitive circuits and some of the reasons why we would like to design delay-insensitive circuits. By means of an example we convey the idea of designing delay-insensitive circuits. We conclude with an outline of the method described in [5].

2. Some history

Delay-insensitive circuits are a special type of circuits. We briefly describe their origins and how they are related to other types of circuits and design techniques. The most common distinction usually made between types of circuits is the distinction between *synchronous circuits* and *asynchronous circuits.*

Synchronous circuits are circuits that perform their (sequential) computations based on the successive pulses of the clock. From the time of the first computer designs many designers have chosen to build a computer with synchronous circuits. In [25] Alan Turing, one of the first computer designers, has motivated this choice as follows:

> We might say that the clock enables us to introduce a discreteness into time, so that time for some purposes can be regarded as a succession of instants instead of a continuous flow. A digital machine must essentially deal with discrete objects, and in the case of the ACE (Automatic Computing Engine) this is made possible by the use of a clock. All other digital computing machines except for human and other brains that I know of do the same. One can think up ways of avoiding it, but they are very awkward.

In the past fifty years many techniques for the design of synchronous circuits have been developed and are described by means of switching theory [11, 15]. The correctness of synchronous systems relies on the bounds of delays in elements and wires. The satisfaction of these delay requirements cannot be guaranteed under all circumstances, and for this reason problems can crop up in the design of synchronous systems. (Some of these problems are described in the next section.) In order to avoid these problems interest arose in the design of circuits without a clock. Such circuits have generally been called *asynchronous* circuits.

The design of asynchronous circuits has always been and still is a difficult subject. Several techniques for the design of such circuits have been developed and are discussed in, for example, [11, 15, 28]. For special types of such circuits formalizations and other design techniques have been proposed and discussed. David E. Muller has given a formalization of a type of circuits which he coined by the name of *speed-independent* circuits. An account of this formalization is given in [16].

From a design discipline that was applied in the Macromodules project [3, 4] at Washington University in St. Louis, the concept of a special type of circuit evolved which was given the name *delay-insensitive* circuit. It was realized that a proper formalization of this concept was needed in order to specify and design such circuits in a well-defined manner. A formalization of the concept of a delay-insensitive circuit was later given in [26]. For the design and specification of delay-insensitive circuits several methods have been developed based on, for example, Petri Nets and techniques derived from switching theory [17].

Another name that is frequently used in the design of asynchronous circuits is *self-timed systems.* This name was introduced by C. L. Seitz in [22] in order to describe a method of system design without making any reference to timing except in the design of the self-timed elements.

Recently, Alain Martin has proposed some interesting and promising design techniques for circuits of which the functional operation is unaffected by delays in elements or wires [12, 13]. His techniques are based on the compilation of CSP-like programs into connections of basic elements. The techniques presented in [5] exhibit a similarity with the techniques applied by Alain Martin.

3. Why delay-insensitive circuits ?

The reasons for designing delay-insensitive systems are manifold. One reason why there has always been an interest in asynchronous systems is that synchronous systems tend to reflect a *worst-case* behavior, while asynchronous systems tend to reflect an *average-case* behavior. A synchronous system is divided into several parts, each of which performs a specific computation. At a certain clock pulse, input data are sent to each of these parts and at the next clock pulse the output data, i.e. the results of the computations, are sampled and sent to the next parts. The correct operation of such an organization is established by making the clock period larger than the worst-case delay for any subcomputation. Accordingly, this worst-case behavior may be disadvantageous in comparison with the average-case behavior of asynchronous systems.

Another more important reason for designing delay-insensitive systems is the so-called *glitch phenomenon.* A glitch is the occurrence of metastable behavior in circuits. Any computer circuit that has a number of stable states also has metastable states. When such a circuit gets into a metastable state, it can remain there for an indefinite period of time before it resolves into a stable state. For example, it may stay in the metastable state for a period larger than the clock period. Consequently, when a glitch occurs in a synchronous system, erroneous data may be sampled at the time of the clock pulses. In a delay-insensitive system it does not matter whether a glitch occurs: the computation is delayed until the metastable behavior has disappeared and the element has resolved into a stable state. One frequent cause for glitches are, for example, the asynchronous communications between independently clocked parts of a system.

The first mention of the glitch problem appears to date back to 1952 (cf. [1]). The first publication of experimental results of the glitch problem and a broad recognition of the fundamental nature of the problem came only after 1973 [2, 8] due to the pioneering work on this phenomenon at the Washington University in St. Louis.

A third reason is due to the effects of *scaling.* This phenomenon became prominent with the advent of integrated circuit technology. Because of the improvements of this technology, circuits could be made smaller and smaller. It turned out, however, that if all characteristic dimensions of a circuit are scaled down by a certain factor, including the clock period, delays in long wires do not scale down proportional to the clock period [13, 21]. As a consequence, some VLSI designs when scaled down may no longer work properly anymore, because delays for some computations have become larger than the clock period. Delay-insensitive systems do not have to suffer from this phenomenon if the basic elements are chosen small enough so that the effects of scaling are negligible with respect to the functional behavior of these elements [24].

A fourth reason is the clear separation between functional and physical correctness concerns that can be applied in the design of delay-insensitive systems. The correctness of the behavior of basic elements is proved by means of physical principles only. The correctness of the behavior of connections of basic elements is proved by mathematical principles only. Thus, it is in the design of the basic elements only that considerations with respect to delays in wires play a role. In the design of a connection of basic elements no reference to delays in wires or elements is made. This does not hold for synchronous systems where the functional correctness of a circuit also depends on timing considerations. For example, for a synchronous system one has to calculate the worst-case delay for each part of the system and for any computation in order to satisfy the requirement that this delay must be smaller than the clock period.

As a last reason, we believe that the translation of parallel programs into delay-insensitive circuits offers a number of advantages compared to the translation of parallel programs into synchronous systems. In [5] a method is presented with which the synchronization and communication between parallel parts of a system can be programmed and realized in a natural way.

4. An example

In order to get an idea of designing delay-insensitive circuits we describe in an informal way a small example. Consider the modulo-3 counter specified by the following communication behavior. The modulo-3 counter has three communication actions: one input, denoted by $a?$, and two outputs, denoted by $p!$ and $q!$. The communication behavior is an alternation of inputs and outputs, starting with an input. The outputs depend on the inputs as follows. After the n-th input, where $n>0$, if $n \,\mathbf{mod}\, 3 \neq 0$, then output $q!$ is produced, else output $p!$ is produced. This behavior is expressed in the following program, or so-called *command*,

$$E0 = \mathbf{pref}[a?;q!;a?;q!;a?;p!].$$

Here, $[E]$ denotes repetition of the enclosed behavior E and $E1;E2$ denotes concatenation of $E1$ and $E2$. The notation $\mathbf{pref}\,E$ denotes the prefix-closure of the behavior E, i.e. if the string of symbols (also called *trace*) $a?q!a?q!a?p!$ is a possible behavior of E, then also each prefix of this trace is a possible behavior of **pref** E.

The following physical interpretation may be associated with the symbols. With each symbol corresponds a terminal of the circuit and with each occurrence of that symbol in a trace corresponds a voltage transition (either a high-going or a low-going transition) in that terminal. Voltage transitions corresponding to inputs are caused by the environment of the circuit; voltage transitions corresponding to outputs are caused by the circuit itself.

In the same way the basic TOGGLE and XOR component can be specified as given in Figure 1.

pref$[a?;b!;a?;c!]$

pref$[(a?\,|\,b?);c!]$

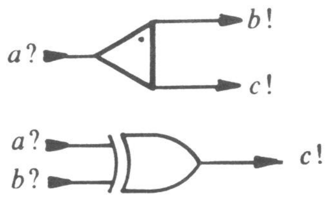


FIGURE 1

The first component is the TOGGLE component and can be considered as a modulo-2 counter. The second component is an XOR component and has the following repetitive behavior. First, the environment provides either an input $a?$ or an input $b?$ (the | separates the alternatives), and then the component produces an output $c!$. After the environment has received an output $c!$ it may produce a new input again, and so on. (Notice that the behaviors of components are specified as orderings of events instead of as logical functions.)

We emphasize that all specifications must be understood as prescriptions for the behavior of the component *and* environment. Consequently, in constructing a decomposition for the modulo-3 counter $E0$ we assume that the environment satisfies the prescribed behavior in $E0$, i.e. the environment provides new inputs $a?$ only when an output has been received. Under this assumption the modulo-3 counter can be decomposed as depicted in Figure 2.

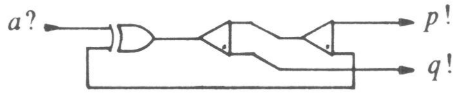


FIGURE 2

Notice that in the decomposition the prescription for the environment of every basic component is not violated. Without much difficulty we can convince ourselves that the functional behavior of this decomposition is unaffected by delays in connection wires or in basic elements. In other words, we could say that the modulo-3 counter is realized by a delay-insensitive connection of basic elements. Knowing how to construct a modulo-3 counter the reader may try, as an exercise, to construct a modulo-17 counter from TOGGLE and XOR components. (There exist several solutions, some more efficient than others.)

5. OUTLINE OF THE METHOD

The method presented in [5] for designing delay-insensitive circuits is briefly described as follows. An abstraction of a circuit is called a *component*; components are specified by programs written in a notation based on *trace theory*. Trace theory was inspired by Hoare's CSP [6, 7] and developed by a number of people at the University of Technology in Eindhoven. It has proven to be a good tool in reasoning about parallel computations [18, 19, 24, 9] and, in particular, about delay-insensitive circuits [10, 20, 21, 26, 27].

The programs are called *commands* and can be considered as an extension of the notation for regular expressions. Any component represented by a command can also be represented by a regular expression, i.e. it is also a *regular* component. The notation for commands, however, allows for a more concise representation of a component due to the additional programming primitives in this notation. These extra programming primitives include operations to express parallelism, tail recursion (for representing finite state machines), and projection (for introducing internal symbols).

Based on trace theory the concepts of *decomposition* of a component and of *delay-insensitivity* are formalized. The decomposition of a component is intended to represent the realization of that component by means of a connection of circuits. Several theorems are presented that are helpful in finding decompositions of a component. Delay-insensitivity is formalized by the definition of *DI decomposition*. A DI decomposition represents a realization of a component by means of a *delay-insensitive* connection of circuits. In order to link decomposition and DI decomposition, the definition of a DI component is introduced. Operationally speaking a DI component represents a circuit for which the communication between circuit and environment takes place in a delay-insensitive way. (It turns out that the definition of a DI component is equivalent with Udding's formalization of a delay-insensitive circuit.) By means of the definition of a DI component one of the fundamental theorems in the thesis can be formulated as follows: DI decomposition and decomposition are equivalent if all components involved are DI components.

This theorem is applied as follows to the example described in the previous section. We showed, informally, that the modulo-3 counter can be decomposed into TOGGLE and XOR components. Furthermore, we have that the TOGGLE component, XOR component, and modulo-3 counter $E0$ are DI components. Consequently, it follows by the above mentioned theorem that the decomposition of Figure 2 forms a DI decomposition of the modulo-3 counter.

Because of the above mentioned theorem, it is important to have techniques to recognize DI components. For this purpose a number of so-called *DI grammars* are developed, i.e. grammars for which any command generated by these grammars represents a (regular) DI component.

Based on these grammars syntax-directed translations of commands into DI decompositions of components represented by these commands are developed. With these grammars the language $\mathcal{L}_4$ of commands is defined. It is shown that any regular DI component represented by a command in the language $\mathcal{L}_4$ can be decomposed in a syntax-directed way into the finite set $\mathbb{B}$ of basic DI components and so-called *CAL components.* CAL components are also DI components. Consequently, since all components involved are DI components, the decomposition into these components is, by the above theorem, also a DI decomposition.

The set of all CAL components is, however, not finite. In order to show that a decomposition into a finite basis of components exists, two decompositions of CAL components are discussed: one decomposition into the finite basis $\mathbb{B}0$ and one decomposition into the finite basis $\mathbb{B}1$. The decomposition of CAL components into the basis $\mathbb{B}1$ is in general *not* a DI decomposition, since not every component in $\mathbb{B}1$ is a DI component. This decomposition, however, is in general simpler than the decomposition into $\mathbb{B}0$ and can be realized in a simple way if so-called *isochronic forks* are used in the realization. The decomposition of CAL components into the basis $\mathbb{B}0$ is an interesting but difficult subject. Since every component in $\mathbb{B}0$ is a DI component, every decomposition into $\mathbb{B}0$ is therefore also a DI decomposition. In [5] a general procedure for the decomposition of CAL components into the basis $\mathbb{B}0$ is described, which is conjectured to be correct

The complete decomposition method can be described as a syntax-directed translation of commands in $\mathcal{L}_4$ into commands of the basic components in $\mathbb{B}0$ or $\mathbb{B}1$. Consequently, the decomposition method is a constructive method and can be completely automated: as soon as we have a specification of a component expressed as a command in $\mathcal{L}_4$ we can find mechanically a decomposition of this component into $\mathbb{B}0$ or $\mathbb{B}1$. Moreover, it is shown that the result of the complete decomposition of any component expressed in $\mathcal{L}_4$ can be linear in the length of the command, i.e. the number of basic elements in the resulting connection is proportional to the length of the command.

Although many regular DI components can be expressed in the language $\mathcal{L}_4$, which is the starting point of the translation method, probably not every regular DI component can be expressed in this way. Nevertheless, it is also shown that for any regular DI component there exists a decomposition into components expressed in $\mathcal{L}_4$, which can then be translated by the method presented.

6. Concluding remarks

The research described in [5] has been fascinating and many-sided. It includes, for example, aspects of

- Language design: which programming primitives do we include in the language in order to be able to present a clear and concise program for a component?
- Programming methodology: do there exist techniques to design programs

from specifications for components in the language of commands?

- Translation techniques: how do we translate programs into connections of basic elements?
- Syntax and semantics: how we can satisfy semantic properties (like a DI component) by imposing syntactic requirements on programs?
- VLSI design: what physical constraints must be met in order to realize the circuit designs obtained in the VLSI medium?

In the thesis the aim of delay-insensitive design has been pursued as far as possible, i.e. correctness arguments based on delay-assumptions have been postponed as far as possible, in order to see what sort of designs such a pursuit would lead to. In this approach our first concern has been the correctness of the designs and only in the second place have we addressed their efficiency. Accordingly, although the number of basic components is already proportional to the length of the program, still many optimizations are possible in translating programs into delay-insensitive circuits.

REFERENCES

1. T.J. CHANEY, *A Comprehensive Bibliography on Synchronizers and Arbiters,* Technical Memorandum No. 306C, Institute for Biomedical Computing, Washington University, St. Louis.
2. T.J. CHANEY, C.E. MOLNAR (1973). Anomalous behavior of synchronizer and arbiter circuits. *IEEE Transactions on Computers C-22,* 421-422.
3. W.A. CLARK (1967). Macromodular computer systems. *Proceedings of the Spring Joint Computer Conference,* AFIPS.
4. W.A. CLARK, C.E. MOLNAR (1974). Macromodular computer systems. R. STACY, B. WAXMAN (eds.). *Computers in Biomedical Research, Vol. IV,* Academic Press, New York.
5. JO C. EBERGEN (1987). *Translating Programs into Delay-Insensitive Circuits,* Ph. D. Thesis, Eindhoven University of Technology.
6. C.A.R. HOARE (1978). Communicating sequential processes. *Communications of the ACM 21,* 666-677.
7. C.A.R. HOARE (1985). *Communicating Sequential Processes,* Prentice-Hall.
8. M. HURTADO (1975). *Dynamic Structure and Performance of Asymptotically Bistable Systems,* D. Sc. Dissertation, Washington University, St. Louis.
9. ANNE KALDEWAIJ (1986). *A Formalism for Concurrent Processes,* Ph.D. Thesis, Eindhoven University of Technology.
10. ANNE KALDEWAIJ (1987). The translation of processes into circuits. J.W. DE BAKKER, A.J. NIJMAN, P.C. TRELEAVEN (eds.). *Proceedings PARLE, Parallel Architectures and Languages Europe, Vol. I,* Springer LNCS, 195-213.
11. ZVI KOHAVI (1970). *Switching and Finite Automata Theory,* McGraw-Hill.
12. ALAIN J. MARTIN (1985). The design of a self-timed circuit for distributed mutual exclusion. H. FUCHS (ed.). *Proceedings 1985 Chapel Hill Conference on VLSI,* Computer Science Press, 247-260.

13. ALAIN J. MARTIN (1986). Compiling communicating processes into delay-insensitive VLSI circuits. *Distributed Computing 1,* 226-234.
14. CARVER MEAD, MARTIN REM (1982). Minimum propagation delays in VLSI. *IEEE Journal of Solid-State Circuits SC-17,* 773-775.
15. R.E. MILLER (1965). *Switching Theory,* Wiley.
16. R.E. MILLER, Chapter 10 in: [15].
17. C.E. MOLNAR, T.P. FANG, F.U. ROSENBERGER (1985). Synthesis of delay-insensitive modules. H. FUCHS (ed.). *Proceedings 1985 Chapel Hill Conference on VLSI,* Computer Science Press, 67-86.
18. MARTIN REM (1985). Concurrent computations and VLSI circuits. M. BROY (eds.). *Control Flow and Data Flow: Concepts of Distributed Computing,* Springer-Verlag, 399-437.
19. MARTIN REM (1987). Trace theory and systolic computations. J.W. DE BAKKER, A.J. NIJMAN, P.C. TRELEAVEN (eds.). *Proceedings PARLE, Parallel Architectures and Languages Europe, Vol. I,* Springer LNCS, 14-34.
20. HUUB M.J.L. SCHOLS (1985). *A Formalisation of the Foam Rubber Wrapper Principle,* Master's Thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology.
21. HUUB M.J.L. SCHOLS, TOM VERHOEFF (1985). *Delay-Insensitive Directed Trace Structures Satisfy the Foam Rubber Wrapper Postulate,* Computing Science Notes 85/04, Department of Mathematics and Computing Science, Eindhoven University of Technology.
22. C.L. SEITZ (1980). System Timing. CARVER MEAD, LYNN CONWAY. *Introduction to VLSI Systems,* Addison-Wesley, 218-262.
23. CLAUDE E. SHANNON (1938). A symbolic analysis of relay and switching circuits. *Trans. AIEE, 57,* 713-723.
24. JAN L.A. VAN DE SNEPSCHEUT (1985). *Trace Theory and VLSI Design,* LNCS 200, Springer-Verlag.
25. ALAN M. TURING (1986). Lecture to the London Mathematical Society on 20 February 1947. B.E. CARPENTAR, R.W. DORAN (eds.). *Charles Babbage Institute Reprint Series for the History of Computing, Vol. 10,* MIT Press.
26. JAN TIJMEN UDDING (1984). *Classification and Composition of Delay-Insensitive Circuits,* Ph.D. Thesis, Eindhoven University of Technology.
27. JAN TIJMEN UDDING (1986). A formal model for defining and classifying delay-insensitive circuits and systems. *Distributed Computing 1,* 197-204.
28. S.H. UNGER (1969). *Asynchronous Sequential Switching Circuits,* Wiley Interscience.

Book Review

E.P. van den Ban
Department of Mathematics, University of Utrecht
Budapestlaan 6, 3584 CD Utrecht, The Netherlands

S. HELGASON (1984). *Groups and Geometric Analysis* (Integral Geometry, Invariant Differential Operators and Spherical Functions), Academic Press, New York, 654 pp.

Nowadays, harmonic analysis on Riemannian symmetric spaces (of Euclidean, compact or non-compact type) is a rather advanced field with many different aspects. Helgason's *Groups and Geometric Analysis* offers an introduction to those aspects which have been among the main research interests of the author in the last thirty years. The diversity of subjects treated is great. Nevertheless the author has managed to achieve coherence of presentation by clearly putting forward a few main themes and basic problems. To illustrate this I intend to systematically go through the contents of the book.

Two main themes of harmonic analysis dominate the first part of the book: firstly the theme of integral transforms (mainly Radon transforms, a few orbital integrals), and secondly that of invariant differential operators. The second part of the book deals with the analysis of spherical functions on Riemannian symmetric spaces, especially those of non-compact type: it provides a beautiful illustration of the themes mentioned.

All of the above are illuminated in an introductory chapter which gives a detailed treatment of the three basic examples: $\mathbb{R}^2 \cong M(2)/O(2)$ = group of isometries of $\mathbb{R}^2$ modulo the stabilizer of the origin (Euclidean type), $S^2 \cong O(3)/O(2)$ (compact type: spherical harmonics) and finally the hyperbolic disk $D = \{z \in \mathbb{C};\ |z| < 1\}$ viewed as the homogeneous space $SU(1,1)/SO(2)$ (of non-compact type). A reader having no background in Lie group theory will get an excellent impression of the role group actions play in harmonic analysis on these spaces.

The next chapter gives a thorough treatment of the d-Radon transform (integration over d-planes) in $\mathbb{R}^n$. Treated are: the inversion, the support and the Plancherel theorem. Applications to PDE's and radiography (X-ray transform) are briefly mentioned. After this the group theoretic structure which underlies the Radon transform is analyzed and formulated in greater generality. Since I found it illuminating, I'll briefly discuss this point of view here.

Let $X = \mathbb{R}^n$ and let Y be the smooth manifold of all hyperplanes in $\mathbb{R}^n$. Then the $(n-1)$-Radon transform R_X is the map $C_c^\infty(X) \to C^\infty(Y)$ defined by

$$R_X f(y) = \int_{\hat{y}} f(x) dm_y(x),$$

for $f \in C_c^\infty(X)$. Here $dm_y(x)$ denotes $(n-1)$-dimensional Euclidean measure on the hyperplane $\hat{y} \subset X$ corresponding to the point $y \in Y$. There is also a dual Radon transform R_Y: $C_c^\infty(Y) \to C^\infty(X)$. If $\phi \in C_c^\infty(Y)$ then $R_Y\phi(x)$ is defined by integrating ϕ over the closed submanifold $\check{x} = \{y \in Y;\ x \in y\}$ of Y. The map R_Y is the transposed of R_X. For $f \in C_c^\infty(X)$, one has the beautiful inversion formula

$$f = \Gamma(\tfrac{1}{2})\Gamma(n/2)^{-1}(-\frac{1}{4\pi}\Delta)^{\frac{1}{2}(n-1)} R_Y R_X f,$$

involving a fractional power of the Laplacian Δ. This formula goes back to RADON [14] for $n=3$ and to JOHN [11] for $n>3$. Its generalization to the d-plane transform is due to HELGASON [7]. We'll now see how group theory enters. In a natural fashion the group $M(n)$ of isometries of $\mathbb{R}^n$ acts transitively on both X and Y. Thus $X \cong M(n)/O(n)$ and $Y \cong M(n)/F$, where $F \cong \mathbb{Z}_2 \times M(n-1)$ is the stabilizer of the hyperplane $x_1 = 0$ in $\mathbb{R}^n$. The crucial observation now is that R_X is equivariant for the natural actions of $M(n)$ on $C_c^\infty(X)$ and $C^\infty(Y)$. Thus representation theory enters the scene. Moreover, the property of equivariance suggests a generalization of the Radon transform to more homogeneous spaces $X = G/H_X$ and $Y = G/H_Y$ for the same Lie group G. Two elements $x \in X$ and $y \in Y$ are called incident if $x \cap y \neq \emptyset$ as cosets in G. A generalized Radon transform can now be defined by integrating functions on X over sets $\hat{y} = \{x \in X;\ x \text{ and } y \text{ incident}\}$. Similarly a dual transform R_Y can be defined. By the way, if $G = U(4)$, $H_X = U(1) \times U(3)$ and $H_Y = U(2) \times U(2)$, then the maps $y \mapsto \hat{y}$ and $x \mapsto \check{x}$ are Penrose correspondences, see PENROSE [13].

Using the general set up indicated above, the author discusses the analysis of Radon transforms for the non-Euclidean Riemannian symmetric spaces of rank 1.

The second chapter deals with the algebra $\mathbb{D}(G/H)$ of invariant differential operators on a homogeneous space G/H of a Lie group G. Geometric constructions such as separation of variables and taking radial parts are discussed in generality. For Riemannian symmetric spaces G/H the algebra $\mathbb{D}(G/H)$ is analyzed in great detail. From this point on the book may be considered as a continuation of Helgason's previous book [10].

Chapter 3 deals with linear group actions (in particular by finite reflection groups), and the corresponding invariant and harmonic polynomials. At the end the Kostant-Rallis theory of adjoint orbits in a symmetric space is discussed.

Chapter 4 is devoted to the study of spherical functions and spherical transforms on a Riemannian symmetric space $X=G/K$ of the non-compact type. Here G is a real semisimple Lie group with a maximal compact subgroup K. The algebra $\mathbb{D}(X)$ of invariant differential operators is commutative. Its joint eigenspaces

$$E(X,\chi) = \{f \in C^{\infty}(X);\ Df=\chi(D)f,\ D \in \mathbb{D}(X)\}$$

are parametrized by characters $\chi \in \mathbb{D}(X)\hat{}$. The spaces $E(X,\chi)$ are invariant for the left regular representation L of G on $C^{\infty}(X)$. Basic problems put forward by the author are:

(1) to describe the joint eigenspaces $E(X,\chi)$,

(2) to determine for which $\chi \in \mathbb{D}(X)\hat{}$, the restriction of L to $E(X,\chi)$ is irreducible,

(3) to decompose functions on X in terms of joint eigenfunctions (Fourier decomposition).

Historically, the third of these problems was solved first, by Harish-Chandra [5]. The set $\mathbb{D}(X)\hat{}$ can be parametrized in a natural fashion by $a_{\mathbb{C}}^{*}/W$, where a is a maximal abelian linear subspace of the Killing orthocomplement of Lie(K) in Lie(G), and where W is the finite reflection group determined by the a-roots in Lie(G). If $\lambda \in a_{\mathbb{C}}^{*}$, then the corresponding element of $\mathbb{D}(X)\hat{}$ is denoted χ_{λ}. The space $E_{\lambda}(X)=E(X,\chi_{\lambda})$ contains a unique left K-invariant function ϕ_{λ} with $\phi_{\lambda}(e)=1$, the so-called elementary or zonal spherical function. Explicitly, ϕ_{λ} can be given as a Radon transform of a function of exponential type. Any K-invariant function $f \in C_c^{\infty}(X)$ can be decomposed as

$$f(x) = \int_{ia^{*}} \phi_{\lambda}(x)\tilde{f}(\lambda)|c(\lambda)|^{-2}d\lambda.$$

Here $d\lambda$ denotes suitably normalized Lebesgue measure on ia^{*}. Moreover, $\tilde{f}(\lambda)=\int_X f(x)\phi_{-\lambda}(x)dx$ is the so called spherical Fourier transform of f. Finally, $c(\lambda)$ is the famous c-function, which occurs as leading coefficient in a converging series expansion describing the asymptotics of $\phi_{\lambda}(x)$ as x tends to infinity in X. Originally, Harish-Chandra proved this result in [5] for a space $\mathbb{S}$ of K-invariant rapidly decreasing functions on X (the proper analogue of the Euclidean Schwartz space), subject to two conjectures being true. One of these conjectures involved an estimate for the c-function, the other density of a space of wave packets in $\mathbb{S}$. The first conjecture was solved by GINDIKIN and KARPELEVIČ [4], who expressed the c-function as a product of quotients of Γ-functions. The other was solved by HARISH-CHANDRA [6].

Using the above inversion formula for C_c^{∞}-functions, HELGASON [8] proved a Paley-Wiener theorem for the spherical Fourier transform, except for certain estimates for the coefficients in the series expansion of ϕ_{λ}. The missing estimates were provided by GANGOLLI [3]. ROSENBERG [15] discovered that it was possible to first prove a support result on wave packets

$$\int_{ia^*} \phi_\lambda(x) A(\lambda) |c(\lambda)|^{-2} d\lambda,$$

with A a W-invariant entire function of Paley-Wiener type on $a^*_{\mathbf{C}}$, and to use this to give a much simpler proof of the inversion formula for C_c^∞-functions. The present book is the first to give a self contained account of these short proofs of the inversion and the Paley-Wiener theorem. It may be interesting to know that after the appearance of this book an even shorter proof of the Paley-Wiener theorem has been discovered by FLENSTED-JENSEN [2]. His proof completely avoids the consideration of asymptotics of spherical functions: instead via an ingenious variation on Hermann Weyl's unitary trick a reduction to the complex and then the Euclidean case is given. A drawback of this method is that it does not give the inversion formula.

The above questions (1) and (2), taken up first by HELGASON [9] have also given rise to some beautiful developments in the subject. In the book they are only dealt with for the case of the hyperbolic disk D, in the introductory chapter. It turns out that $E_\lambda(D)$ can be characterized as the image under a generalized Poisson transformation of the space of hyperfunctions on the boundary $\partial D = \{z \in \mathbf{C}: |z| = 1\}$: the classical integral representation of harmonic functions on the disk is a special case of this. The analogue of the above description of eigenfunctions by Poisson transformations for a general Riemannian symmetric space of the non-compact type was conjectured and partially proved by HELGASON [9] and finally proved by KASHIWARA ET AL. [12]. An excellent introduction to this material can be found in SCHLICHTKRULL [16].

The book ends with a chapter on (the relatively standard) Fourier analysis on a Riemannian symmetric space of the compact type.

Each chapter of the book concludes with a set of exercises and in addition a set of historical notes which is usually very complete and helpful. In fact I noticed only one omission: in the discussion of asymptotics of zonal spherical functions a reference to the enlightening paper of CASSELMAN and MILIČIĆ [1] is missing.

The first third of the book can certainly be used as a textbook for beginning graduate students. The rest requires a greater knowledge of Lie group theory which however nowhere goes beyond the contents of the author's previous book [10]. The present book will also be an excellent source of reference for experts. No doubt it will become a new standard in the field.

REFERENCES

1. W. CASSELMAN, D. MILIČIĆ (1982). Asymptotic behavior of matrix coefficients of admissible representations. *Duke Math. J. 49,* 869-930.
2. M. FLENSTED-JENSEN (1986). Analysis on non-Riemannian symmetric spaces. *AMS Reg. Conf. Ser. 61.*
3. R. GANGOLLI (1971). On the Plancherel formula and the Paley-Wiener theorem for spherical functions on semisimple Lie groups. *Ann. of Math. 93,* 150-165.
4. S.G. GINDIKIN, F.I. KARPELEVIČ (1962). Plancherel measure for Riemann

symmetric spaces of nonpositive curvature. *Dokl. Akad. Nauk. SSSR 145,* 252-255.

5. HARISH-CHANDRA (1958). Spherical functions on a semisimple Lie group, I and II. *Amer. J. Math. 80,* 241-310 and 553-613.
6. HARISH-CHANDRA (1966). Discrete series for semisimple Lie groups, II. *Acta Math. 116,* 1-111.
7. S. HELGASON (1964). A duality in integral geometry; some generalizations of the Radon transform. *Bull. AMS 70,* 435-446.
8. S. HELGASON (1966). An analog of the Paley-Wiener theorem for the Fourier transform on certain symmetric spaces. *Math. Ann. 165,* 297-308.
9. S. HELGASON (1970). A duality for symmetric spaces with applications to group representations. *Adv. Math. 5,* 1-154.
10. S. HELGASON (1978). *Differential Geometry, Lie Groups and Symmetric Spaces,* Academic Press, New York.
11. F. JOHN (1955). *Plane Waves and Sperical Means,* Wiley (Interscience), New York.
12. M. KASHIWARA, A. KOWATA, K. MINEMURA, M. OKAMOTO, T. OSHIMA, M. TANAKA (1978). Eigenfunctions of invariant differential operators on a symmetric space. *Ann. Math. 107,* 1-39.
13. R. PENROSE (1967). Twistor algebra. *J. Math. Phys. 8,* 345-366.
14. J. RADON (1917). Ber. Verh. Sächs. Akad. Wiss. Leipzig. *Math.-Nat. kl. 69,* 262-277.
15. J. ROSENBERG (1977). A quick proof of Harish-Chandra's Plancherel theorem for spherical functions on a semisimple Lie group. *Proc. Amer. Math. Soc. 63,* 143-149.
16. H. SCHLICHTKRULL (1984). *Hyperfunctions and Harmonic Analysis on Symmetric Spaces,* Birkhäuser, Boston.

The Radon Transform: First Steps

N.M. Temme

In this note we discuss some aspects of the Radon transform mentioned in the review of Helgason's book in this Newsletter.

In short, the Radon transform of a function $f(x,y)$ of two variables is the set of line integrals, with obvious generalizations to higher dimensions. It plays a fundamental role in a large class of applications which fall under the heading of tomography. In a narrow sense, tomography is the problem of reconstructing the interior of an object by passing radiation through it and recording the resulting intensity over a range of directions. It is the problem of finding f from the above mentioned line integrals and is related to the inversion of the Radon transform.

Before discussing a simple example of how to compute the Radon transform, we will tell more about the background of the applications. In mathematical physics there is a notorious class of difficult problems: the ill-posed problems. The notion of a *well*-posed problem is due to Hadamard: a solution must exist, be unique, and depend continuously on the data. In ill-posed problems the last condition may be violated, and then important difficulties may arise, especially when the data are not complete or not accurate. Tomography falls in this class of ill-posed problems.

Probably the most widely known applications of tomography are in medicine. Computer assisted tomography (CAT-scan) uses X-rays directed from a range of directions to reconstruct the density in a thin slice of the body (the Greek word *τομος* means *section*). Recent advances in medical tomography include nuclear magnetic resonance (NMR), where strong magnetic fields are used to make hydrogen atoms resonate. One advantage over the CAT-scan is that the use of potentially harmful X-rays can be avoided.

An important feature of the medical applications of tomography is its 'nondestructive' character. Also in industry there is a considerable need to investigate the integrity and remaining reliable lifetime of components and structures by using nondestructive evaluation. Once again the components are subjected to penetrating radiation with the aim of deducing information about their internal states.

The search for oil depends heavily upon the analysis of seismic data. This is another example of the reconstruction of internal features of a body from monitored reflections of radiation or energy flows.

The Radon transform is an interesting example of a mathematical problem that was considered and solved long before its applicability was seen. In fact, this problem, as well as its three-dimensional version, was solved by J. Radon in 1917 and later rediscovered in various settings such as probability theory (recovering a probability distribution from its marginal distributions) and astronomy (determining the velocity distribution of stars from the distribution of radial velocities in various directions). Of course, much work was needed to adapt the Radon inversion formula to the incomplete information available in practice. The computational solution of ill-posed problems of the form arising in the general area of tomography is a very active research topic in computational mathematics. Although the last decade has yielded very useful algorithms, much work remains to be done; for instance in 3-D problems. At CWI research on reconstruction problems started quite recently. There are promissing contacts with industry (on NMR and seismic problems) and with researchers from medical disciplines.

To describe the rôle of line integrals in tomography we start with the equation

$$I = I_0 e^{-\mu x},$$

for the beam density of a narrow beam of X-ray photons through some homogeneous material, where I_0 is the input intensity (number of photons per second per unit cross-sectional area) and I is the observed intensity after the beam passes the distance x through the material. The linear *attenuation coefficient* μ depends, among other things, on the density of the material. This formula has to be changed for material that is inhomogeneous, where μ depends on a space parameter. In two variables the analogue of the above equation becomes

$$I = I_0 \exp[-\int_L \mu(x,y)ds],$$

where the line integral is along the beam path L, which is parametrized by s (see figure 1).

By moving the source and detector it is possible to obtain a set of line integrals. Taking logarithms, this constitutes a sampling of the Radon transform. Then an appropriate inversion or reconstruction algorithm is applied to recover an approximation to the attenuation coefficient distribution over a transverse section of some portion of the human body.

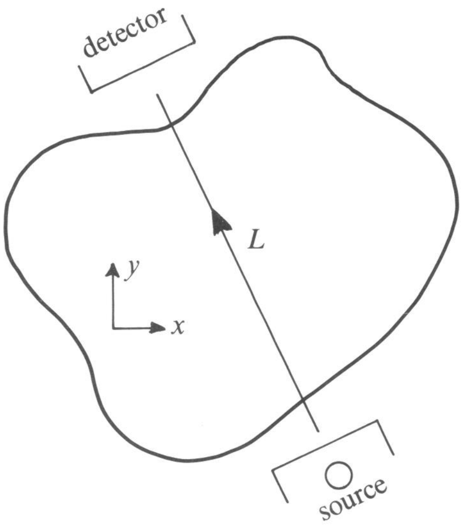
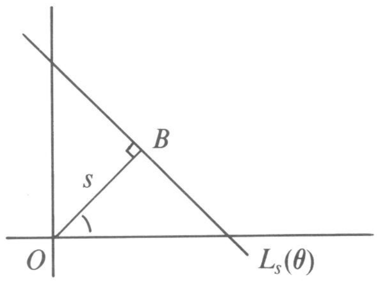


FIGURE 1. The beam passes through the region characterized by $\mu(x,y)$ along the line L

A typical coordinate system for setting up the Radon transform is the following. A line L_s in $\mathbb{R}^2$ with distance s from the origin $O = (0,0)$ in x,y-plane is further characterized by an angle θ (see figure 2)



$B = (s\cos\theta,\ s\sin\theta)$ is a fixed point on L

FIGURE 2

So each line has two parameters s, θ which look like polar coordinates, but in fact they are not: if $s = 0$ different values of θ yield different lines $L_0(\theta)$.

Let $A = (x,y) = (r\cos\phi, r\sin\phi)$ (polar coordinates) denote a variable point on $L_s(\theta)$. Then, if the distance from A to B equals t, we have

$$\begin{cases} x = s\cos\theta - t\sin\theta, \\ y = s\sin\theta + t\cos\theta. \end{cases} \tag{1}$$

To define the Radon transform we assume that $f : \mathbb{R}_2 \rightarrow \mathbb{R}$ is continuous and integrable, and we write

$$Rf(s,\theta) = \int_{-\infty}^{\infty} f(x,y)dt, \quad (x,y) \in L_s(\theta), \tag{2}$$

where t appears in the above notation for $(x,y) \in L_s(\theta)$. By allowing negative values of s, we can restrict the θ-domain to $[0,\pi]$, since

$$Rf(s,\theta) = Rf(-s,\pi+\theta).$$

In practical problems f is compactly supported, that is $f = 0$ if r is large enough, say $r \geqslant 1$.

For higher dimensions a vector notation is very useful. For $\mathbb{R}_2$ we start with the unit vector ω with polar angle θ. So the point B in figure 2 can be written as the vector $B = s\omega$, with scalar s. $L_0(\theta)$ runs through the origin and is parallel with $L_s(\theta)$. Let $y \in L_0(\theta)$. Then $\omega \cdot y = 0$. So $L_s(\theta)$ is characterized by the set of end points of vectors x that can be written as

$$x = y + s\omega, \quad \text{with} \quad \omega \cdot y = 0,$$

or as the set of end points of vectors $x \in \mathbb{R}_2$ satisfying $x \cdot \omega = s$. Hence, the Radon transform can be written as

$$Rf(s,\theta) = \int_{y \cdot \omega = 0} f(y + s\omega)dy = \int_{x \cdot \omega = s} f(x)dx,$$

where, for convenience, we now suppose that the argument of f is a vector. The above definition is for $x, y, \omega \in \mathbb{R}_2$. However, by integrating over (hyper) planes, the same notation can be used for $\mathbb{R}_n$.

Especially fruitful is the introduction of the δ-function notation. Recall that this generalized function has the property

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0)$$

for smooth functions f. So the line integral over the X-axis can be expressed as

$$Rf(0,0) = \int_{-\infty}^{\infty} f(x,0)dx = \int_{-\infty}^{\infty} \{ \int_{-\infty}^{\infty} f(x,y)\delta(y)dy \} dy.$$

In general we can write

$$Rf(s,\theta) = \int_{-\infty}^{\infty}\int f(x,y)\delta(s - x\cos\theta - y\sin\theta)dxdy$$

$$= \int\int f(x)\delta(s - x \cdot \omega)dx, \quad x \in \mathbb{R}_2,$$

and for n-dimensions we can use the same notation

$$Rf(s,\theta) = \int\int f(x)\delta(s - x \cdot \omega)dx,$$

$s \in \mathbb{R}$, $x, \omega \in \mathbb{R}_n$; θ is now a $(n-1)$-dimensional vector containing the polar angles for defining a $(n-1)$-dimensional hyperplane in $\mathbb{R}_n$, and ω is the corresponding unit vector.

EXAMPLE. Let $f(x,y) = \exp(-x^2-y^2) = \exp(-r^2)$. We use

$$\int\int_{-\infty}^{\infty} f(x,y)\delta(s - x\cos\theta - y\sin\theta)dxdy.$$

The transformation (rotation)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

yields directions of integration parallel and perpendicular to $L_s(\theta)$. From $x^2+y^2 = u^2+v^2$ (the mapping is an isometry), we obtain

$$Rf(s,\theta) = \int\int_{-\infty}^{\infty} \exp(-u^2-v^2)\delta(s-u)dudv = \sqrt{\pi}e^{-s^2}.$$

On the other hand, using (1), $r^2 = x^2+y^2 = s^2+t^2$, and definition (2) we easily obtain the same result. It follows that the Radon transform of the Gaussian distribution yields again a Gaussian.

The theory of the Radon transform can be put in the framework of the Fourier transforms. Since for the latter inversion formulas are readily available, the inversion of Radon transformations is, in principle, established. Radon was not aware of this link with Fourier transformation, and he put the inversion in the following form: Let $F_Q(q)$ be the mean of $Rf(s,\theta)$ over all $L_s(\theta)$ on a distance q from a point $Q = (x,y) = (r\cos\phi, r\sin\phi)$, i.e.,

$$F_Q(q) = \frac{1}{2\pi}\int_0^{2\pi} Rf[q + r\cos(\phi-\theta),\theta]d\theta,$$

then

$$f(x,y) = -\frac{1}{\pi}\int_0^{\infty} q^{-1}dF_Q(q) \tag{3}$$

(in the notation of a Stieltjes integral).

The conditions on f are: continuous and compactly supported. For a recent elementary proof see NIEVERGELT [4].

For the pure mathematician this may give an end to the matter. For the applied mathematician there are two important difficulties:

- the inversion of Radon transforms is an ill-posed problem (inaccurate data may produce instabilities)
- the number of line integrals (i.e., data) is limited; also the directions (θ-values) may be restricted to a narrow range.

The numerical analyst usually applies algebraic inversion techniques for integral equations, instead of using the analytical inversion theorem. The latter, however, plays a fundamental rôle in diverse areas of Radon transformations and tomography, especially when it is written in terms of Fourier transforms.

For a very nice introductory monograph the reader is referred to DEANS [1]. The book of NATTERER [3] goes further in the direction of mathematical foundation of this topic. The IEEE-Special Issue [5] gives an interesting introduction to both the mathematical and the applied aspects of tomography. In the references below excellent bibliographics are included. Recent contributions in which reconstruction is considered as a statistical problem by modelling both noise and the to be reconstructed object f as stochastic processes are described in VARDI ET AL. [6] and GEMAN and MCCLURE [2].

REFERENCES

1. S.R. DEANS (1983). *The Radon Transform and Some of its Applications*, Wiley.
2. S. GEMAN, D.E. MCGLURE (1985). Bayesian image analysis: an application to single photon emission tomography. *1985 Stat. Comp. Sec., Proc. Amer. Stat. Assoc.*
3. F. NATTERER (1986). *The Mathematics of Computer Tomography*, Wiley.
4. Y. NIEVERGELT (1986). *SIAM Review 28*, 79-84.
5. PROCEEDINGS OF THE IEEE, March 1983, Special Issue on Computerized Tomography.
6. Y. VARDI, L. SHEPP, L. KAUFMAN (1985). A statistical model for positron emission tomography. *J. Amer. Statist. Assn. 80,* 8-37.

Abstracts
of Recent CWI Publications

When ordering any of the publications listed below please use the order form at the back of this issue.

CWI Tract 35. M.L. Eaton. *Lectures on Topics in Probability Inequalities.*
AMS 60E15, 62H05; 197 pp.
Abstract: The material in this book is based on a set of lectures given at the University of Amsterdam in the first half of 1985. The lectures highlighted the following topics: i) majorization results and their extensions to reflection groups; ii) association and the FKG inequality; iii) log concavity, Anderson's theorem and related topics. To a large extent the treatment of the material is mathematically self-contained, although the examples sometimes require a bit of specialized statistical knowledge.

CWI Tract 36. A.H.P. van der Burgh, R.M.M. Mattheij (eds.). *Proceedings of the First International Conference on Industrial and Applied Mathematics (ICIAM 87). Contributions from the Netherlands.* (433 pp.).
Abstract: This tract contains the contributions from the Netherlands to the First International Conference on Industrial and Applied Mathematics (ICIAM 87). The papers cover the following topics: applied mathematical analysis, scientific computing, control theory and signal processing, computational geometry, applied probability and statistics, mathematics of natural sciences, software and hardware aspects.

CWI Tract 37. L. Stougie. *Design and Analysis of Algorithms for Stochastic Integer Programming.*
AMS 90C15, 90C27, 90C39, 90B35; 92 pp.
Abstract: Stochastic programming problems are mathematical programming problems in which some of the parameters are modeled as random variables to represent uncertainty about their

value. Integrality constraints imposed on some of the decision variables leads us into the field of stochastic integer programming. Due to the stochasticity and the combinatorial nature, the problems are extremely hard to solve and therefore the ideal of optimality is usually abolished and an approximation of the optimal solution is settled for. A framework for the theoretical analysis of the quality of approximations is presented. For various specific stochastic integer programming problems approximation algorithms are designed and analyzed. For some specific problems of small size dynamic programming algorithms are presented to obtain an optimal solution.

CWI Tract 38. J.B.G. Frenk. *On Banach Algebras, Renewal Measures and Regenerative Processes.*
AMS 60K05, 13J05; 201 pp.

Abstract: Consider the following problem: Suppose $\mathbf{X}_1, \mathbf{X}_2, \ldots$ is a sequence of independent and identically distributed random variables with probability distribution F. Set $\mathbf{S}_0 = 0$, $\mathbf{S}_n = \sum_{k=1}^{n} \mathbf{X}_k$, $n \geqslant 1$ and consider for every bounded Borel set $\mathcal{B}$ the measure $U(\mathcal{B}) := \mathcal{E}[\sum_{n=0}^{\infty} 1_{\{S_n \in \mathcal{B}\}}]$. This measure is called the *renewal* measure and it plays an important role in the analysis of regenerative processes. In this CWI Tract a self-contained treatment of the asymptotic behaviour of the renewal measure under various assumptions on the probability distribution F is given using Fourier analysis and the theory of commutative Banach algebras.

CWI Tract 39. H.J.M. Peters, O.J. Vrieze (eds.). *Surveys in Game Theory and Related Topics.*
AMS 90Dxx, 90D05, 90D10, 90D12, 90D13, 90D15, 90D30, 90D35, 90D40; 330 pp.

Abstract: This book consists of eleven surveys in game theory and two on related topics (2). Each chapter is self-contained. The authors are specialists in the respective fields and in the last section of each chapter they discuss some of their latest results. The whole field of game theory is covered. We mention: Refinements of equilibrium concepts, Games with incomplete information, Stochastic games, Combinatorial games, Games of linear optimization problems, Simple games, Solution concepts for cooperative games, Bargaining solutions. The 'related topics' are concerned with social choice theory and the relation between decision theory and game theory.

CWI Tract 40. J.L. Geluk, L. de Haan. *Regular Variation, Extensions and Tauberian Theorems.*
AMS 26A12, 40E05; 132 pp.

Abstract: Functions of regular variation were invented by Karamata in 1930 as a suitable class of functions in connection with a Tauberian theorem for Laplace transforms. Many other applications are known. The present text intends to give a self-contained, smooth and coherent introduction to the theory of regular variation and its main extensions. Disregarding the possible applications we show how these classes of functions are a natural setting for Tauberian theorems of the Laplace type. Also some results are given for general kernel transforms. In the text there is a clear separation between the various classes of functions. We have tried to stick to the main line of the theory putting little emphasis on various refinements, minimality of conditions and other specialized topics. The theory is built in circles. After a full treatment of regularly varying (RV) functions sections on the function classes Π and Γ follow. The theory of these function classes parallels closely the theory of regular variation. Next (Chapter 2) Tauberian theorems for Laplace transforms are treated in which these function classes (RV, Π and Γ) play a central role. Finally (Chapter 3 and 4) the theory is further extended. Here limits are replaced by upper and lower bounds. Chapter 3 gives the theory of these further generalizations of regular variation and in

Chapter 4 Tauberian theorems are given in which these generalizations play a central role.

CWI Tract 41. S.J. Mullender (ed.). *The Amoeba Distributed Operating System: Selected papers 1984-1987.*
AMS 68A05; CR C0, C2, D4; 308 pp.
Abstract: This tract contains selected papers relating to the Amoeba Distributed Operating System which were published between 1984 and 1987. The papers reflect a joint effort between the Centre for Mathematics and Computer Science, and the Vrije Universiteit, both located in Amsterdam, the Netherlands.

CWI Tract 42. P.R.J. Asveld, A. Nijholt (eds.). *Essays on Concepts, Formalisms, and Tools.*
AMS 68-02, 68-03, 68B05, 68C01, 68C05, 68C20, 68C30, 68C40, 68D22, 68D25, 68D35,68Fxx, 68F05, 68F10, 68F20, 68F25, 68G15, 68G99; CR D.1.2, D.1.4, D.2.2, D.3.1-4, F.0, F.1.1-3, F.2.2, F.3.2-3, F.4.0-3, G.1.0, I.1.1-2, I2.3, K.2.; 278 pp.
Abstract: This book contains a collection of papers in which different aspects of modeling subject matters can be recognized. Most papers in this volume deal with 'artificial' situations. Their subject matters are human-defined or human-constructed languages and systems. The authors introduce and study formalisms, show how a subject matter can be modeled, or discuss the building and usefulness of tools for the generation of programs that facilitate the writing or processing of user programs. Ultimately, the introduction and study of the formalisms that are discussed in these papers have been inspired by practical considerations. Practical considerations may lead to intriguing theoretical problems, though sometimes no foreseeable practical application of the results of investigations into these problems can be given.

CWI Syllabus 13. M.J. Bergvelt, G.M. Tuynman, A.P.E. ten Kroode. *Proceedings Seminar 1983-1985 Mathematical Structures in Field Theories, Vol. 2.*
AMS 70Hxx, 53Bxx; 206 pp.
Abstract: These proceedings cover part of the lectures given in the seminar 'Mathematical Structures in Field Theories', held at the University of Amsterdam during the academic years 1983-1984 and 1984-1985 (see CWI-Syllabi 2, 6 and 8). Chapter 1 gives an introduction to classical mechanics and symplectic geometry and is an introduction to the next two chapters. In the second chapter Yang-Mills theory is treated as a classical, albeit singular, dynamical system; the mathematical framework is in terms of differential geometry and the paper is an application of the work by Gotay, Nester and Hinds to the Yang-Mills system. The third chapter is devoted to the geometrical description of the Toda lattice. This lattice is described as a Hamiltonian system on a coadjoint orbit in the dual of a Lie algebra. The symplectic structure is the Kostant-Kirillov symplectic form.

CWI Syllabus 14. *Vacantiecursus 1987: De personal computer en de wiskunde op school.*
Abstract: This course (for high school mathematics teachers) contains four possible applications of a personal computer in teaching mathematics. The topics are fractals, number theory, numerical mathematics and general mathematics. This syllabus has been written in Dutch.

CS-R8707. B. Awerbuch, L.M. Kirousis, E. Kranakis & P.M.B. Vitányi. *On proving register atomicity.*
AMS 68C05, 68C25, 68A05, 68B20; CR B.3.2, B.4.3, D.4.1, D.4.4; 24 pp.; **key**

words: register, run, atomic, regular, reader, writer, proof method.

Abstract: Concurrent access of shared variables by asynchronous processes does not require mutual exclusion, but can be solved with no waiting. A fruitful paradigm in this context is the notion of a shared register satisfying a niceness condition called atomicity. The model is rigorously presented, and then a method is given for proving register atomicity. It is then used to give simple proofs of the atomicity of two register constructions, the matrix register and the Bloom register, without assuming the existence of a global clock. (The matrix construction shows how to implement an atomic, n-writer, n-reader register with value domain V from n^2 atomic - or even regular - 1-writer, 1-reader registers with value domain $\mathbb{N} \times V$, with $\mathbb{N}$ the set of nonnegative integers. Bloom's construction shows how to implement an atomic, 2-writer, n-reader register with value domain V, from two atomic, 1-writer, $n-1$-reader registers with value domain $\{0,1\} \times V$. These constructions are so simple that they may be even practical.)

CS-R8713. S.J. Mullender. *Process management in a distributed operating system.*
CR D.4, C.2.4, D.2.5; 12 pp.; **key words:** distributed operating system, process management, migration, Amoeba.

Abstract: The Distributed Systems Group at the Centre for Mathematics and Computer Science and the Vrije Universiteit in Amsterdam has designed a collection of services for the management of processes in the Amoeba distributed operating system. With a small set of kernel operations, it is possible to download, debug, migrate, and checkpoint processes. First, the basic kernel mechanisms are described, followed by the description of a number of supporting user-space services. The paper ends with a discussion of the properties of Amoeba that made this design possible.

CS-R8714. P.H. Rodenburg. *Algebraic specifications for parametrized data types: the case of minimal computable algebras and parameters with equality.*
AMS 03D80, 68Q65; CR F.3.2, D.3.3; 14 pp.; **key words:** parametrized data type, computable minimal algebra, Kreisel-Lacombe-Shoenfield theorem, algebra with equality, finite equational specification, effective operation.

Abstract: For minimal algebras, and under certain assumptions on the domain of parameters, it is shown that a persistent parametrized data type with computable parameters is effective if is has a finite equational specification.

CS-R8715. L.M. Kirousis. *On effectively labeling planar projections of polyhedra.*
AMS 68T10, 51M20, 68Q20, 68Q25, 68R10; CR I.2.10, F.2.2, G.2.2; 16 pp.; **key words:** polyhedron, labeling, 2-dimensional image of a polyhedron.

Abstract: A well-known method for interpreting planar projections (images) of 3-dimensional polyhedra is to label their lines by the Clowes-Huffman scheme. However, the question of whether there is such a labeling has been shown to be NP-complete. In this paper an algorithm is given linear in time that answers the labelability question under the assumption that some information is known about those edges of the polyhedron both of whose faces are visible. In many cases, this information can be derived from the image itself. Moreover, the algorithm has an effective parallel version, i.e., with polynomially many processors it can be executed in time polynomial in $\log n$.

CS-R8716. J.C.M. Baeten & R.J. van Glabbeek. *Merge and termination in process algebra.*
AMS 68Q10, 68Q55, 68Q45, 68N15; CR F.1.2, F.3.2, F.4.3, D.3.3; 25 pp.; **key words:** concurrency, process algebra, empty process, termination, tick.

Abstract: In Vrancken (reference 14 of this paper), the empty process ϵ was added to the Algebra of Communicating Processes of Bergstra and Klop. Reconsidering the definition of the parallel composition operator merge, we found that it is preferable to explicitly state the termination option. This gives an extra summand in the defining equation of merge, using the auxiliary operator $\surd$ (tick). We find that tick can be defined in terms of the encapsulation operator ∂_H. We give an operational and a denotational semantics for the resulting system ACP$\surd$, and prove that they are equal. We consider the Limit Rule, and prove it holds in our models.

CS-R8717. T. Tomiyama & P.J.W. ten Hagen. *The concept of intelligent integrated interactive CAD systems.*
AMS 69H12, 69H21, 69K10, 69K36, 69L60; CR H.2.1, I.2.1, I.3.6, J.6; 29 pp.; **key words:** CAD, machine design, design theory, knowledge engineering, interactive systems, conceptual modeling.

Abstract: In this report we first propose the concept of Intelligent Integrated Interactive CAD (IIICAD) systems, after having analyzed problems of present Computer Aided Design (CAD) systems. IIICAD is expected to be a large software system based on knowledge engineering technology. In order to develop such a complicated system we need to put emphasis on the importance of theoretical work besides implementational techniques, since these techniques cannot even solve all problems of conventional CAD systems.

CS-R8718. P.J.W. ten Hagen & R. van Liere. *A model for graphical interaction.*
AMS 69K32, 69K36; CR I.3.2, I.3.6; 26 pp.; **key words:** graphics systems, methodology and techniques.

Abstract: A model for graphical interaction is presented which will allow us to precisely and formally describe many important aspects of graphical input, graphical output and various correlations between these two. The model encapsulates the fundamental properties of elementary graphics input devices and their feedback. It also encompasses the operational modes of such devices and the screen resources they occupy. On top of this, the model allows the description of compound inputs leading to the description of arbitrary complex interaction techniques. Moreover, these interaction techniques can be very precisely controlled by the application program. The latter is of importance for incorporating such techniques in a variety of methods. One of the main achievements of the model is the encapsulation of the concept called I/O-symmetry. Finally, it is shown how the model can be used to describe various concepts such as user freedom, direct manipulation, error recovery and dialogue scheduling. Directions for further development of the I/O-unit model will be outlined.

CS-R8719. P.J.W. ten Hagen & H.J. Schouten. *Parallel graphical output from dialogue cells.*
CR I.3.4, D.1.3; 15 pp.; **key words:** computer graphics, parallelism, user interface management systems.

Abstract: A system that accepts and processes graphical output from parallel processes using a single workstation, and its use in a User Interface Management System called Dialogue Cells, is described. It allows a programmer to easily, concisely and precisely describe pictures and operations on them in a highly interactive environment. Each parallel process can do output independently, but pictures can also be moved from one process to another. Each process has its own graphical output environment. The description and implementation of the system is based on GKS.

CS-R8720. T. Tomiyama & P.J.W. ten Hagen. *Organization of design knowledge in an intelligent CAD environment.*

AMS 69H12, 69H21, 69K14, 69L60; CR H.1.2, H.2.1, I.2.4, J.6; 28 pp.; **key words:** conceptual modeling, data modeling, knowledge engineering, knowledge representation, CAD.

Abstract: More and more attention has been paid to the concept of so-called intelligent CAD systems as a promising approach towards the next generation of integrated engineering environments. In this paper, first we propose the concept of Intelligent Integrated Interactive CAD systems. We introduce a language based on predicate logic and the object oriented programming paradigm, and describe the mechanisms of the system in this language. Then we discuss the possibility of describing and organizing design knowledge using this language. From this discussion we will see how design knowledge should be embedded and work in an intelligent, integrated, and interactive CAD environment.

CS-R8721. J.C.M. Baeten & R.J. van Glabbeek. *Abstraction and empty process in process algebra.*

AMS 68Q10, 68Q45, 68Q55, 68N15; CR F.1.2, F.3.2, F.4.3, D.3.3; 24 pp.; **key words:** concurrency, process algebra, hidden step, hiding abstraction, silent step, internal action, empty process, termination.

Abstract: In this paper, we combine the hidden step η of the authors' paper (reference 2 of this paper) with the empty process ϵ of Vrancken and the authors (references 12 and 3 of this paper). We formulate a system ACP_c, which is a conservative extension of the systems ACP_η, $ACP\surd$, but also of ACP_τ. This is a general system, in which most relevant issues can be discussed. Abstraction from internal steps can be achieved in two ways, in two stages: we can abstract to the hidden step η, and then from η to Milner's silent step τ.

CS-R8722. L.M. Kirousis, E. Kranakis & P.M.B. Vitányi. *Atomic multireader register.*

AMS 68M10, 68P15, 68Q25; CR B.3.2, B.4.3, D.4.1, D.4.2, D.4.4; 29 pp.; **key words:** register, shared register, atomic, regular, reader, writer.

Abstract: We give implementations for atomic, shared, asynchronous, wait-free registers: (i) A new implementation of an atomic, 1-writer, 1-reader, b-bit register from $O(b)$ safe, boolean registers (i.e. from scratch). The solution uses neither repeated writing of the input nor repeated reading of the output. (ii) An implementation of an atomic, 1-writer, n-reader, multibit register from $O(n^2)$ atomic, 1-writer, 1-reader, multibit registers. Both constructions rely on the same idea. In a sense (ii) is a generalization of (i). These results show how to construct atomic, multireader registers from - basically - elementary hardware like flip-flops.

CS-R8723. I. Shizgal. *An Amoeba replicated service organization.*

CR C.2.4, D.4.4; 7 pp.; **key words:** Amoeba, multicast, 2-phase commit, distributed service.

Abstract: A technique is described which allows replicated instances of servers on the Amoeba operating system to dynamically detect other currently active server sites, and for these to cooperate in the maintenance of replicated data.

CS-R8724. J.N. Kok. *A fully abstract semantics for data flow nets.*

AMS 68B10; CR D.3.1, F.3.2, F.3.3; 18 pp.; **key words:** data flow programming, data flow networks, denotational semantics, multivalued functions, concurrency.

Abstract: Two semantic models for data flow nets are given. The first model is an intuitive operational model. This model has an important drawback: it is not compositional. An example shows

the non-compositionality of our model. There exist two nets that have the same semantics, but when they are placed in a specific context, the semantics of the resulting nets differ. The second one is obtained by adding information to the first model. The amount of information is enough to make it compositional. Moreover, we show that we have added the minimal amount of information to make the model compositional: the second model is fully abstract with respect to the equivalence generated by the first model. To be more specific: the first model describes the semantics of a data flow net as a function from (tuples of) sequences of tokens to sets of (tuples of) sequences of tokens. The second one maps a data flow net to a function from (tuples of) infinite sequences of finite words to sets of (tuples of) infinite sequences of finite words.

CS-R8725. J.A. Bergstra & J.W. Klop. ACP_τ: *A universal axiom system for process specification.*

AMS 68Q10, 68Q55, 68Q45, 68N15; CR F.1.2, F.3.2, F.4.3, D.3.3; 17 pp.; **key words:** communicating processes, process algebra, bisimulation semantics, graph models, recursive specifications.

Abstract: Starting with Basic Process Algebra (BPA), an axiom system for alternative composition (+) and sequential composition (·) of processes, we give a presentation in several intermediate stages leading to ACP_τ, Algebra of Communicating Processes with abstraction. At each successive stage an example is given showing that the specification power is increased. Also some graph models for the respective axiom systems are informally presented. We conclude with the Finite Specification Theorem for ACP_τ, stating that each finitely branching, effectively presented process (as an element of the graph model) can be specified in ACP_τ by means of a finite system of guarded recursion equations.

CS-R8726. V. Akman. *Steps into a geometer's workbench.*

CR F.2.2, G.1.5-6, I.1.2, I.2.9, I.3.4-5; 21 pp.; **key words:** minimal paths, Voronoi diagram, polyhedra, computational geometry, geometer's workbench, algorithm animation, workstation, prototyping, Macsyma, Smalltalk, Model-View-Controller, Lisp.

Abstract: As computational geometry matures, it becomes crucial to use its techniques in the professional environment of graphics and robotics. This however is a nontrivial task since (i) computational geometry concerns itself with asymptotic analysis, and (ii) in search of elegance it ignores the special cases which are the bugbear of practical applications. I see experimentation as a way to resolve these difficulties and propose a software system to act as a 'Geometer's Workbench'. This entails the integration of geometric knowledge with algorithm animation and object-oriented graphics. The workbench should allow improvisation with geometric objects and is expected to broaden the way geometry is used in the same style as Macsyma did this for algebra.

CS-R8727. V. Akman. *Geometry and graphics applied to robotics.*

CR F.2.2, G.1.5-6, I.1.2, I.2.9, I.3.4-5; 20 pp.; **key words:** minimal paths, Voronoi diagram, continuum method, visibility, polyhedra, computational geometry, concrete complexity, geometer's workbench, algorithm animation, workstation, prototyping, Macsyma, Smalltalk, Model-View-Controller, Lisp.

Abstract: In the first part of this paper, I review the recent efforts on integrating robotics and computer science ideas. Specifically, I advocate the view that applying results from areas of computer science such as concrete complexity, symbolic computation, and computational geometry will simplify the work of robot programmers. In the second part, the discussion takes place in the context of model-based robotics. I argue that the time has come to build a 'Geometer's Workbench', a system integrating geometric knowhow with algorithm animation techniques and interactive graphics

to visualize complex situations as encountered in robotics. Such a system is expected to broaden the way geometry is practiced in the same style as Macsyma did this for algebra.

CS-R8728. T. Tomiyama & P.J.W. ten Hagen. *Representing knowledge in two distinct descriptions: extensional vs. intensional.*
CR H.1.2, H.2.1, I.2.4, J.6; 19 pp.; **key words:** conceptual modeling, data modeling, knowledge engineering, knowledge representation, CAD.
Abstract: This paper describes a theory of knowledge on which future CAD systems can be based. First, we present two distinct description methods, viz. extensional and intensional. Second, these two are compared in the context of CAD applications and their advantages and disadvantages are clarified. Finally, we propose a new data description method which combines extensional and intensional description methods.

CS-R8729. J.C. Mulder & W.P. Weijland. *Verification of an algorithm for log-time sorting by square comparison.*
AMS 68Q35, 68Q60, 68Q10, 68Q55; CR B.7.1, D.2.4, F.1.2, F.3.2; 25 pp.; **key words:** concurrency, process algebra, sorting, correctness proof, rank sort, orthogonal tree network, asynchronous cooperation, delay-insensitive.
Abstract: In this paper a concurrent sorting algorithm called RANKSORT is presented, able to sort an input sequence of length n in $\log n$ time, using n^2 processors. The algorithm is formally specified as a delay-insensitive circuit. Then, a formal correctness proof is given, using bisimulation semantics in the language ACP of Bergstra & Klop. The algorithm has area $\cdot$ time^2 $= O(n^2 \log^4 n)$ complexity which is slightly suboptimal with respect to the lower bound of $AT^2 = \Omega(n^2 \log n)$.

CS-R8730. J.W. Klop & A. Middeldorp. *Strongly sequential term rewriting systems.*
AMS 68Q50; CR F.4.1, F.4.2; 38 pp.; **key words:** regular term rewriting systems, normalizing reduction strategy, needed redex.
Abstract: For regular term rewriting systems, G. Huet and J.-J. Lévy have introduced the property of 'strong sequentiality'. A strongly sequential regular term rewriting system admits an efficiently computable normalizing one-step reduction strategy. As shown by Huet and Lévy, strong sequentiality is a decidable property. In this paper we present a structural analysis of strongly sequential term rewriting systems, leading to two new and simplified proofs of the decidability of this property.

CS-R8731. B. Veth. *An integrated data description language for coding design knowledge.*
CR D.2.m, F.4.1, I.2.1, I.2.4, J.6; 19 pp.; **key words:** CAD, design theory, logic, theory of knowledge, theory of design objects, qualitative reasoning, object-oriented programming, logic programming, knowledge engineering, software engineering, prototyping.
Abstract: We present in a unifying framework the basic notions of IDDL (Integrated Data Description Language) to code design knowledge in the IIICAD system. IIICAD is an intelligent, integrated and interactive computer-aided design environment we are currently developing at the Centre for Mathematics and Computer Science.

CS-R8732. P. America & J.W. de Bakker. *Designing equivalent semantic models for process creation.*

AMS 68B10, 68C01; CR D.1.3, D.3.1, D.3.3, F.1.2, F.3.2; 75 pp.; **key words:** operational semantics, denotational semantics, parallelism, process creation, continuations, domain equations, metric spaces, fixed points.

Abstract: Operational and denotational semantic models are designed for languages with process creation, and the relationships between the two semantics are investigated. The presentation is organized in four sections dealing with a uniform and static, a uniform and dynamic, a nonuniform and static, and a nonuniform and dynamic language, respectively. Here uniform/nonuniform refers to a language with uninterpreted/interpreted elementary actions, and static/dynamic to the distinction between languages with a fixed/growing number of parallel processes. The contrast between uniform and nonuniform is reflected in the use of linear time versus branching time models, the latter employing a version of Plotkin's resumptions. The operational semantics make use of Hennessy and Plotkin's transition systems. All models are built on metric structures, and involve continuations in an essential way. The languages studied are abstractions of the parallel object-oriented language POOL for which we have designed separate operational and denotational semantics in earlier work. The paper provides a full analysis of the relationship between the two semantics for these abstractions. Technically, a key role is played by a new operator which is able to decide dynamically whether it should act as sequential or parallel composition.

CS-R8733. J.A. Bergstra & J.W. Klop. *A convergence theorem in process algebra.*

AMS 68Q05, 68Q10, 68Q55, 68Q45; CR F.1.2, F.3.1, F.3.2, F.3.3; 32 pp.; **key words:** process algebra, projective limit model, merge, left-merge, recursion equations, complete metric space, process graph, Approximation Induction Principle.

Abstract: We study a convergence phenomenon in the projective limit model $\mathbf{A}^{\infty}$ for PA, an axiom system in the framework of process algebra for processes built from atomic actions by means of alternative composition $(+)$ and sequential composition $(\cdot)$, and subject to the operations $\|$ (merge) and $\|\!\!\_$ (left-merge). The model $\mathbf{A}^{\infty}$ is also a complete metric space. Specifically, it is shown that for every element $q \in \mathbf{A}^{\infty}$ the sequence $q, s(q), s^2(q), \ldots, s^n(q), \ldots$ converges to a solution of the (possibly unguarded) recursion equation $X = s(X)$ where $s(X)$ is an expression in the signature of PA involving the recursion variable X. As the convergence holds for arbitrary starting points q, this result does not seem readily obtainable by the usual convergence proof techniques. Furthermore, the connection is studied between projective models and models based on process graphs. Also these models are compared with the process model introduced by De Bakker and Zucker.

CS-R8734. L. Kossen & W.P. Weijland. *Verification of a systolic algorithm for string comparison.*

AMS 68Q35, 68Q60, 68Q10, 68Q55; CR B.7.1, D.2.4, F.1.2, F.3.2; 34 pp.; **key words:** concurrency, process algebra, synchronous communication, asynchronous cooperation, self-timed system, systolic system, VLSI, correctness proof.

Abstract: A self-timed systolic system computing the edit distance between two strings is proved correct by means of an algebraical concurrency theory ACP (Algebra of Communicating Processes). A systolic system is a system consisting of a great number of concurrently operating and cooperating elements. In the system described here, the flow of control is regulated by the elements themselves: the system is self-timed. A formal approach can be helpful to construct complex systems such as VLSI-circuits.

CS-N8701. J.W. Klop. *Term rewriting systems: a tutorial.*

AMS 03B40, 68Q99; CR D.1.1, F.4.1; 37 pp.; **key words:** Abstract Reduction Systems, Term Rewriting Systems, Combinatory Logic, reduction strategies, regular Term Rewriting Systems.

Abstract: Term Rewriting Systems play an important role in various areas, such as abstract data type specifications, implementations of functional languages and automated deduction. In this tutorial we introduce some of the basic concepts and facts for TRS's. No attempt is made to present a comprehensive survey: e.g. the tutorial does not contain material about conditional TRS's or equational TRS's. The spirit of the material presented here is syntactic rather than semantic. An emphasis is put on Abstract Reduction Systems, of which not only TRS's are instances, but also Semi-Thue Systems, tree replacement systems, and graph rewrite systems. As an example of an important termination proof technique we describe the recursive path orderings in a new presentation.

OS-R8708. J.L. van den Berg, O.J. Boxma & W.P. Groenendijk. *Sojourn times in the M/G/1 queue with deterministic feedback.*
AMS 60K25, 68M20; 9 pp.; **key words:** M/G/1 queue, feedback, sojourn times.

Abstract: In this paper we consider an M/G/1 queueing model, in which each customer is fed back a fixed number of times. For the case of negative exponentially distributed service times at each visit, we determine the joint distribution of the sojourn times of the consecutive visits. As a by-product we obtain the total sojourn time distribution; it can be related to the sojourn time distribution in the M/D/1 queue with processor sharing. For the case of generally distributed service times at each visit, a set of linear equations is derived, from which the mean sojourn times per visit can be calculated.

OS-R8709. G.A.P. Kindervater, J.K. Lenstra & A.H.G. Rinnooy Kan. *Perspectives on parallel computing.*
AMS 68M05, 68Q10, 90Bxx, 90Cxx; 5 pp.; **key words:** operations research, parallelism, architectures, computations, computational models.

Abstract: Operations research is one of the areas that is likely to benefit from advances in parallel computing. We briefly review what has been achieved in recent years and try to sketch what may be expected in the near future. More realism in theoretical models of parallel computation and more uniformity in available architectures will be required. Formal techniques will have to be developed for the design and implementation of efficient parallel algorithms. Only then can parallelism fulfill its promise and considerably expand the range of effectiveness of operations research methods.

OS-R8710. J.L. van den Berg & O.J. Boxma. *Sojourn times in feedback queues.*
AMS 60K25, 68M20; 16 pp.; **key words:** M/M/1 queue, feedback, sojourn times.

Abstract: This paper considers an M/M/1 queue with a very general feedback mechanism. When a customer completes his i-th service, he departs from the system with probability $1-p(i)$ and he cycles back with probability $p(i)$. The main result of the paper is a formula for the joint distribution of the successive sojourn times of a customer in the system. As a by-product, it is shown that the sojourn times in all individual cycles are identically, negative exponentially, distributed. Also, the correlation between the sojourn times of the j-th and k-th cycle of a customer is calculated; furthermore, the distribution of the total sojourn time is derived.

OS-R8711. M.W.P. Savelsbergh. *Local search for constrained routing problems.*

AMS 90B05, 90B35, 90C27; 10 pp.; **key words:** traveling salesman problem, vehicle routing problem, local search, iterative improvement.

Abstract: We develop local search algorithms for routing problems with various side constraints such as time windows on vertices and precedence relations between vertices. The algorithms are based on the k-exchange concept. The presence of side constraints introduces feasibility problems. Checking the feasibility of a given solution in the straightforward way requires time which is linear in the number of vertices. Our method reduces this effort to constant time.

NM-R8708. W.H. Hundsdorfer. *Stability results for θ-methods applied to a class of stiff differential-algebraic equations.*

AMS 65L05, 65L20; 10 pp.; **key words:** differential-algebraic equations, stiff initial value problems.

Abstract: In this paper we consider some simple numerical methods for a class of stiff differential-algebraic equations (with index 2). The methods are based on the well-known θ-method for ordinary differential equations. The stability and some convergence properties of the methods are discussed.

NM-R8709. M. Bergman. *Implementation of elementary functions in Ada.*

AMS 69D49, 65-04; 15 pp.; **key words:** Ada, elementary mathematical functions, portability, scientific libraries.

Abstract: Unlike many other languages, Ada does not define elementary mathematical functions. Therefore a package of basic mathematical functions has been designed and implemented, which meets requirements like portability, general usefulness and efficiency. For these purposes Ada offers a number of interesting and useful features, which will be discussed in brief. Further, a detailed description is given of the way the elementary functions have been implemented.

NM-R8710. J. Kok. *Design and implementation of elementary functions in Ada.*

AMS 69D49, 65-04; 20 pp.; **key words:** Ada, high level language, basic mathematical functions, scientific libraries, portability.

Abstract: This report describes the design and implementation of the elementary functions in Ada for the EC-funded project 'Pilot Implementations of Basic Modules for Large Portable Numerical Libraries in Ada'. It presents the specification of a generic package for the declaration of the elementary functions in Ada which employs the project's library error handling mechanism. Further, it describes the portable implementation of this package, and the work done on testing and documenting the package.

NM-R8711. J.G. Verwer. *Some stability results for the hopscotch difference method when applied to convection-diffusion equations.*

AMS 65M10; CR 5.7.; 9 pp.; **key words:** partial differential equations, convection-diffusion equations, hopscotch method, linear stability.

Abstract: The hopscotch method is a time stepping scheme applicable to wide classes of spatially discretized, multi-space-dimensional, time-dependent partial differential equations (PDEs). In this contribution attention is focussed on the simple odd-even hopscotch method (OEH). Our aim is to present some interesting stability properties of this method for convection-diffusion equations where the space discretization is carried out by standard symmetrical and/or one-sided finite differences. First, we give a general formulation of the time-stepping scheme and outline its main computational features. Next, we discuss linear stability properties of the method in the multi-dimensional case. We present explicit expressions for the critical time step based on the Von Neumann condition. We show that in certain cases an increase of diffusion may render the process

unstable, an observation which is in clear contrast to the common practice. This strange phenomenon can occur only in the higher dimensional case. Finally, we discuss the spectral condition, but only in one space dimension. In accordance with results for the explicit Euler rule, we conclude that the spectral condition is misleading in the sense that it does not prevent large error growth.

NM-R8712. W. Hoffmann. *NUMVEC FORTRAN Library manual. Chapter : Simultaneous linear equations Update # 1.*
AMS 65V05, 65F05, 15A06; CR 5.14; 8 pp.; **key words:** Gauss-Jordan elimination, linear equations, software.
Abstract: This document describes two NUMVEC FORTRAN Library routines, INVGJ and GJPCF. INVGJ calculates the approximate inverse of a real square matrix by Gauss-Jordan elimination with partial pivoting using column interchanges. GJPCF calculates the approximate solution of a set of real linear equations with multiple right hand sides, **AX** = **B**, by Gauss-Jordan elimination with partial pivoting using column interchanges.

NM-R8713. J.G. Blom, J.M. Sanz-Serna, J.G. Verwer. *A Lagrangian moving grid scheme for one-dimensional evolutionary partial differential equations.*
AMS 65M50, 65M99; CR 5.17; 18 pp.; **key words:** numerical analysis, partial differential equations, time-dependent problems, moving grid methods, space-time finite differences.
Abstract: A Lagrangian moving grid finite difference method for one-space-dimensional, evolutionary partial differential equations which exhibit sharp transitions in space and time is developed. The method is based on a Crank-Nicolson type difference scheme derived via a coordinate transformation governed by equidistribution of the second space derivative. Each time step of our method involves two stages. First, a static grid numerical integration is carried out, immediately followed by a de Boor type redistribution of nodes at the forward time level. This stage serves only to compute the transformation. Second, a moving grid numerical integration is carried out with the Crank-Nicolson scheme. Numerical experiments show that the method automatically concentrates the grid in regions of high spatial activity and is also able to step in time with stepsizes larger than those needed by static methods, that is, methods which for intervals of time work on a fixed, nonuniform grid. As a result the method achieves high accuracy with few gridpoints in space and time.

NM-R8714. P.J. van der Houwen. *Stabilization of explicit difference schemes by smoothing techniques.*
AMS 65M, 65N; 11 pp.; **key words:** numerical analysis, partial differential equations, difference schemes, smoothing, stability, convergence.
Abstract: We give a survey of applications of smoothing techniques to difference schemes for partial differential equations. Smoothing techniques may improve the stability or convergence of the difference scheme considerably. We distinguish smoothing of the numerical solution itself, smoothing of the right-hand side of the differential equation, and residue smoothing. Examples are given for parabolic, hyperbolic and elliptic equations.

NM-N8702. B.P. Sommeijer, W.H. Hundsdorfer, C.T.H. Everaars, P.J. van der Houwen, J.G. Verwer. *A numerical study of a 1D stationary semiconductor model.*
AMS 65N20, 65H10; **key words:** numerical analysis, partial differential equations, nonlinear systems, initial approximations.

Abstract: Based on a 1D model describing the stationary basic semiconductor device equations, several numerical methods have been investigated to solve this type of problems. The main part of this work discusses the solution of the nonlinear systems which result from discretization of the PDE's. Emphasis is placed upon finding suitable initial approximations when iterative processes are employed.

MS-R8703. M.L. Eaton. *Admissibility in fair Bayes prediction problems. I. General theory.*

AMS 62C05, 62C15; 37 pp.; **key words:** prediction, predictive distribution, Bayes rules, improper prior distributions, fair Bayes loss functions.

Abstract: This paper is concerned with sufficient conditions for the admissibility of posterior predictive distributions which are derived from improper proper distributions. It is argued that the relevant loss functions are the so called fair Bayes loss functions. The admissibility results are valid for a wide class of such loss functions. The main result is applied to the one-dimensional translation problem with Lebesgue measure as the improper prior to give sufficient conditions for admissibility of the resulting predictive distribution. A connection with recurrence of an induced Markov chain is described.

MS-R8704. T.A. Louis, J.K. Bailey. *Controlling error rates using prior information and marginal totals to select tumor sites.*

AMS 62F15, 62H17, 62K99, 62P10; 31 pp.; **key words:** Bayes methods, bioassay, conditional power, multiplicity.

Abstract: Carlin & Louis proposed in 1985 a selection procedure designed to control the problems of multiplicity associated with *P*-values reported from carcinogen bioassays. Instead of searching the data for statistically significant tumor site/type combinations, the procedure uses site/type specific prior information and conditioning statistics to select sites and types with potentially significant *P*-values. Any single *P*-value selected by this method retains its usual meaning, and the size of the test procedure is controlled. We apply the Carlin & Louis procedure to a random sample of bioassays using male and female mice and rats from the National Cancer Institute's data base. From these data we estimate priors for lifetime incidence and dose effect. Then, we compare the performance of the selection procedure to use of Bonferroni adjusted and unadjusted minimum observed *P*-values.

MS-R8705. T.A. Louis. *Efficient monotone sequential design.*

AMS 62K05, 62L05, 62L15; 26 pp.; **key words:** optimal design, sequential methods, monotone follower problem, regret, scale invariance.

Abstract: Optimal designs for nonlinear problems depend on unknown parameters. With no constraints, sequential methods can pick the next design point using prior and accruing information and 'home in' on the optimal design. In monotone designs, observations must be ordered in time (or other metameter), so design options decrease. For example, in rodent bioassay experiments where a group of n rodents are simultaneously put on test, sacrifices to discover the presence or absence of tumors can occur only at ages greater than or equal to the current age. So, if data are taken beyond the optimal age, it is not possible subsequently to go back to it.
For the class of problems studied, statistical information depends on the time data are taken and unknown parameters. In this report we consider a class of monotone designs based on a scale invariant design objective, and two structures for statistical information. One factors into a function of time and a function of the unknown parameter and results from data from a scale family for each t. The other depends on time divided by the unknown parameter, and results, for example, from destructive life tests. We develop and investigate two- and three-stage adaptive rules, compute the asymptotic order of the regret for these rules relative to the rule taking all data at the

optimal time, and show that the order is close to the best possible. We find the rate at which the performance of these rules departs from scale invariance, and report on a simulation study comparing the current rule to others in the literature.

MS-R8706. M.L. Eaton. *Concentration inequalities for Gauss Markov estimators.*
AMS 62H12, 62J05, 62F10; 33 pp.; **key words:** Gauss Markov estimators, concentration inequalities, elliptical densities, log concave densities, majorization, group induced orderings, reflection groups.

Abstracts: Let M be the regression subspace and γ the set of possible covariances for a random vector Y. The linear model determined by M and γ is *regular* if the identity is in γ and if $\Sigma(M) \subseteq M$ for all $\Sigma \in \gamma$. For such models, concentration inequalities are given for the Gauss Markov estimator of the mean vector under various distributional and invariance assumptions on the error vector. Also, invariance is used to establish monotonicity results relative to a natural group induced partial ordering.

AM-R8702. H. Roozen. *An asymptotic solution to a two-dimensional exit problem arising in population dynamics.*
AMS 35C20, 35J25, 60J60, 60J70, 92A15; 26 pp.; **key words:** a two-dimensional exit problem, diffusion matrix singular at boundary.

Abstract: This paper deals with a two-dimensional stochastic system, which is the diffusion approximation to a birth-death process. At the boundary of the state space, the diffusion matrix becomes singular. The stochastic fluctuations are assumed to be small. By an asymptotic analysis, expressions are derived that determine the probability of exit at each of the two boundaries and the expectation and variance of the exit time. These expressions contain constants that can be computed numerically.

AM-R8703. J.B.T.M. Roerdink. *The biennial life strategy in a random environment.*
AMS 15A52, 60H99, 92A15; 17 pp.; **key words:** population biology, random matrix products.

Abstract: A discrete-time population model with two age classes is studied which describes the growth of biennial plants in a randomly varying environment. A fraction of the oldest age class delays its flowering each year. The solution of the model involves products of random matrices. We calculate the exact mean and variance of the long-run geometric growth rate assuming a gamma distribution for the random number of offspring per flowering plant after one year. It is shown, both by analytical calculation and numerical examples, that it is profitable for the population to delay its flowering, in the sense that the average growth rate increases and the extinction probability decreases. The optimal values of the flowering fraction depend upon the environmental and model parameters.

AM-R8704. J. Grasman, J.B.T.M. Roerdink. *Stochastic and chaotic relaxation oscillations.*
AMS 34C15, 34E15, 58F13, 60H10; 15 pp.; **key words:** relaxation oscillation, stochastic perturbation, chaos.

Abstract: For relaxation oscillators stochastic and chaotic dynamics are investigated. The effect of random perturbations upon the period is computed. For an extended system with additional state variables chaotic behaviour can be expected. As an example the Van der Pol oscillator is changed into a third order system admitting period doubling and chaos in a certain parameter range. The distinction between chaotic oscillation and oscillation with noise is explored by studying a time

series of one of the variables. Return maps, power spectra and Lyapunov exponents are analyzed for that purpose.

AM-R8705. H.E. de Swart. *Low-order spectral models of the atmospheric circulation: a survey.*

AMS 86A10, 76E20, 35A35, 34C35; 40 pp.; **key words:** interaction between planetary waves and synoptic eddies, low-order models, bifurcation analysis, vacillation between weather regimes.

Abstract: A quasi-geostrophic potential vorticity equation is derived from the Navier-Stokes equations for atmospheric motions. It describes the evolution of a quasi-horizontal flow at time scales of a few days and more. The associated boundary value problem is analyzed by projection of the equation onto orthonormal eigenfunctions (modes) of a Sturm-Liouville operator. The result is a spectral model, consisting of an infinite number of nonlinear ordinary differential equations for the evolution of the mode amplitudes. Low-order spectral models, in which only a few modes are resolved, appear to have properties which agree with observations of the atmospheric circulation. However, little justification is available for truncating the spectral expansion at low resolution numbers. It is argued that stochastic forcing terms should be added to the equations, but it is not priori clear how they should be specified. A derivation is presented of a specific low-order spectral model of the quasi-geostrophic potential vorticity equation. Some of its subsystems are analyzed for their physical and mathematical properties. It appears that topography can act as a triggering mechanism to generate multiple equilibria. The corresponding flow patterns resemble preference states of the atmospheric circulation. The systems can vacillate between three characteristic regimes with transitions provided either by external or internal mechanisms. A discussion is presented on the validity of stochasticly forced spectral models and deterministic chaotic models for the atmospheric circulation.

AM-N8701. M. van Herwijnen. *IMAGES: A study of Serra's mathematical morphology (in Dutch).*

44 pp.

Abstract: This report gives an overview of J. Serra's theory of mathematical morphology, which provides a powerful, organized and uniform approach to computer image analysis. Images are considered as subsets of $\mathbb{R}^n$ and the theory considers the class of 'morphological transformations' of these subsets. Various imperfections in the theory are pointed out. Also implementation of the transformations on a personal computer is dicussed.

PM-R8703. T.H. Koornwinder. *Group theoretic interpretations of Askey's scheme of hypergeometric orthogonal polynomials.*

AMS 33A65, 33A30, 33A75, 44A20, 22E70; 29 pp.; **key words:** classical orthogonal polynomials, Askey's scheme of hypergeometric orthogonal polynomials, limit transitions between families of special functions, integral transforms with special function kernels, group theoretic interpretations of special functions.

Abstract: This paper gives a survey of various group theoretical interpretations of the classical hypergeometric orthogonal polynomials which occur in the Askey scheme. Furthermore, a natural extension of the scheme is discussed, where families of hypergeometric functions are allowed which may be orthogonal in a generalized sense and which may be of nonpolynomial nature. Limit transitions and group theoretic interpretations are also given for this extended scheme. It turns out that the scheme contains many triples of kernels of integral transforms for which the composition yields identity.

PM-R8704. S.N.M. Ruijsenaars. *Action-angle maps and scattering theory for some finite-dimensional integrable systems. The pure solution case.*
AMS 70F10, 70H15, 70H40; 34 pp.; **key words:** action-angle maps, solitons, relativistic particle dynamics, scattering theory.

Abstract: We construct an action-angle transformation for the Calogero-Moser systems with repulsive potentials, and for relativistic generalizations thereof. This map is shown to be closely related to the wave transformations for a large class $\mathcal{C}$ of Hamiltonians, and is shown to have remarkable duality properties. All dynamics in $\mathcal{C}$ lead to the same scattering transformation, which is obtained explicitly and exhibits a soliton structure. An auxiliary result concerns the spectral asymptotics of matrices of the form $M\exp(tD)$ as $t \to \infty$. It pertains to diagonal matrices D whose diagonal elements have pairwise different real parts and to matrices M for which certain principal minors are non-zero.

CWI Activities

Winter 1987

With each activity we mention its frequency and (between parentheses) a contact person at CWI. Sometimes some additional information is supplied, such as the location if the activity will not take place at CWI.

Study group on Analysis on Lie groups. Jointly with University of Leiden. Biweekly. (T.H. Koornwinder)

Seminar on Integrable Systems. Once a month. (M. Hazewinkel)

Seminar on Algebra and Geometry. Once a month. (A.M. Cohen)
The Cohomology of the Schubert variety and Coxeter groups.

Cryptography working group. Monthly. (J.H. Evertse)

Colloquium 'STZ' on System Theory, Applied and Pure Mathematics. Twice a month. (J. de Vries)

Seminar on Mathematical Morphology. 21,28 October, 4,11,13,17 November. (H.J.A.M. Heijmans)

Study group Biomathematics. Lectures by visitors or members of the group. Jointly with University of Leiden. Bimonthly. (O. Diekmann)

Progress meetings of the Applied Mathematics Department. Weekly. (N.M. Temme)
New results and open problems on the research topics of the department: biomathematics, mathematical physics, asymptotic and applied analysis, image analysis.

Study group on Statistical and Mathematical Image Analysis. Every three weeks. (R.D. Gill)

Progress meetings of the Mathematical Statistics Department. Biweekly. (K.O. Dzhaparidze)
Talks by members of the department on recent developments in research and consultation.

Workshop on Statistics in Image Analysis. Jointly with University of Aarhus, Denmark. 28-29 September. (R.D. Gill)

Study group on Empirical Processes. Jointly with University of Amsterdam. Biweekly. (R.D. Gill)

Lunteren meeting on Stochastics. 16, 17, and 18 November 1987 at 'De Blije Werelt', Lunteren. (R. Helmers)

Invited speakers: A.D. Barbour (Zürich, Switzerland), M.T. Barlow (Cambridge, USA), P.J. Bickel (Berkeley, USA), R.W. Keener (Ann Arbor, USA), E. Nummelin (Helsinki, Finland), J.A. Wellner (Seattle, USA).

System Theory Days. Irregular. (J.H. van Schuppen, J.M. Schumacher)

Study group on System Theory. Biweekly. (J.M. Schumacher)

Colloquium on Queueing Theory and Performance Evaluation. Irregular. (O.J. Boxma)

Progress meetings on Numerical Mathematics. Weekly. (H.J.J. te Riele)

Study group on Numerical Software for Vector Computers. Monthly. (H.J.J. te Riele)

Study group on Differential and Integral Equations. Lectures by visitors or group members. Irregular. (H.J.J. te Riele)

Conference on Numerical Mathematics. 5 - 7 October 1987 at Zeist. (W.H. Hundsdorfer)

Invited speakers: B. Gustafsson (Uppsala, Sweden), A. Lerat (Paris, France), H.D. Mittelman (Tempe, USA), P.L. Roe (Cranfield, UK), H. Schwetlick (Halle, DDR), A. Spence (Bath, UK).

Study group on Graphics Standards. Monthly. (M. Bakker)

National Study Group on Concurrency. Jointly with Universities of Leiden & Eindhoven and several industrial research establishments. 25 September, 23 October, 20 November, 11 December. (J.W. de Bakker)

Post-academic Course on Modern Techniques in Software Engineering. (J.C. van Vliet)

Various lectures present modern techniques and methods for the construction of complex software systems. The course is meant for persons with a background in computer science, who are or will be actively involved in the construction of those systems.

Post-academic Course on PROLOG. 20-21 October. (P.J.F. Lucas)

In this course, both the theoretical foundations of logic programming and the applications of the programming language PROLOG are discussed. The course is meant for researchers and engineers who consider using PROLOG in their projects.

Study group on Logical Aspects on Artificial Intelligence. 5,6,9,10,11 November. (P.J.F. Lucas)

Study group on Dialogue Programming. (P.J.W. ten Hagen)

Process Algebra Meeting. Weekly. (J.W. Klop)

Visitors to CWI from Abroad

W. van Assche (Leuven, Belgium) 11 June. M.L. Balinski (Ecole Polytechnique, Paris, France) 6-7 April. M. Borsboom (Van Karmann Institute, Rhode Sr. Genese, Belgium) 1 April. A. Camina (University of Norwich) 18-20 May. R. Charron (St. John's, New Foundland) 27-29 April. H.R. Gail (IBM Yorktown Heights, USA) 27 April. L. Gatteschi (Torino, Italy) May. M. Gevers (Louvain University, Louvain La Neuve, France) 23-24 April. M.L. Glasser (Clarkson Univesity, Potsdam, USA) 11-13 May. I. Guessarian (Lab. Informatique Théoretique et Programmation, Paris, France) 22-26 June. Guo Ben-yu (Technical University Shanghai, China) 22-23 June. J. Halpern (IBM Research, San Jose, USA) 14-15 April. C. Jennison (University of Bath, England) 26 June - 3 July. W.M. Kantor (Eugene, USA) 16-19 May. V.K. Klonias (University of Crete, Greece) 20-21 May. G. Latouche (University of Brussels, Belgium) 24 June. Lin Qun (Xiamen University, China) 22-23 June. J. Mau (University of Tübingen, West Germany) 17-19 June. H. Matano (Hiroshima University, Japan) June. A. Märtz (Humboldt University, Berlin, East Germany) May. M. Mimura (Hiroshima University, Japan) June. F. Morain (University of Limoges, France) 13 April. A.M. Odlyzko (AT&T Bells Labs, Murray Hill, USA) 10 April. J. Pinter (Budapest) 5 June. J. Prüss (Gesamthochschule Paderborn, West Germany) 12 May. G. Robin (University of Limoges, France) 13 April. F. Rouvière (Nice, France) 21-25 April. R.L. Smith (University of Michigan, Ann Arbor, USA) 24 April. G. Steiner (Mc Master University, Hamilton, Canada) April. D. Venable (Rochester, USA) 26-30 April. J. Walrand (University of California, Berkeley, USA) 9 June. W.A. Woyczynski (Case Western Reserve Research Institute, Cleveland, USA) 15-17 June. A.I. Zayed (California Polytechnic State University, USA) 29 June - 14 August. R.A. Moyeed (Dhaka, Bangla Desh) May-June. E.V. Khamaladze (Steklov Inst. Moscow, USSR) June-July.

Order Form for CWI Publications

Sales Department
Centre for Mathematics and Computer Science
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

☐ Please send the publications marked below on an exchange basis
☐ Please send the publications marked below with an invoice

	Publication code	Price per copy	Number of copies wanted
☐	CWI Tract 35 *)	Dfl. 30.40	....
☐	CWI Tract 36 *)	56.80	....
☐	CWI Tract 37 *)	14.10	....
☐	CWI Tract 38 *)	31.60	....
☐	CWI Tract 39 *)	50.40	....
☐	CWI Tract 40 *)	20.30	....
☐	CWI Tract 41 *)	46.70	....
☐	CWI Tract 42 *)	42.90	....
☐	CWI Syllabus 13 *)	31.60	....
☐	CWI Syllabus 14 *)	16.50	....
☐	CS-R8707	4.--	....
☐	CS-R8713	4.--	....
☐	CS-R8714	4.--	....
☐	CS-R8715	4.--	....
☐	CS-R8716	4.--	....
☐	CS-R8717	5.20	....
☐	CS-R8718	4.--	....

*) not available on exchange

	Publication code	Price per copy	Number of copies wanted
☐	CS-R8719	4.--	....
☐	CS-R8720	5.20	....
☐	CS-R8721	4.--	....
☐	CS-R8722	5.20	....
☐	CS-R8723	4.--	....
☐	CS-R8724	4.--	....
☐	CS-R8725	4.--	....
☐	CS-R8726	4.--	....
☐	CS-R8727	4.--	....
☐	CS-R8728	4.--	....
☐	CS-R8729	4.--	....
☐	CS-R8730	6.40	....
☐	CS-R8731	4.--	....
☐	CS-R8732	11.60	....
☐	CS-R8733	5.20	....
☐	CS-R8734	5.20	....
☐	CS-N8701	6.40	....
☐	OS-R8708	4.--	....
☐	OS-R8709	4.--	....
☐	OS-R8710	4.--	....
☐	OS-R8711	4.--	....
☐	NM-R8708	4.--	....
☐	NM-R8709	4.--	....
☐	NM-R8710	4.--	....
☐	NM-R8711	4.--	....
☐	NM-R8712	4.--	....
☐	NM-R8713	4.--	....
☐	NM-R8714	4.--	....
☐	NM-N8702	4.--	....
☐	MS-R8703	6.40	....
☐	MS-R8704	5.20	....
☐	MS-R8705	4.--	....
☐	MS-R8706	5.20	....
☐	AM-R8702	4.--	....
☐	AM-R8703	4.--	....
☐	AM-R8704	4.--	....
☐	AM-R8705	6.40	....
☐	PM-R8703	5.20	....
☐	PM-R8704	5.20	....

If you wish to order any of the above publications please tick the appropriate boxes and return the completed form to our Sales Department.
Don't forget to add your name and address!
Prices are given in Dutch guilders and are subject to change without notice. Foreign payments are subject to a surcharge per remittance to cover bank, postal and handling charges.

Name ....

Street ....

City ....

Country ....

Signature ....

Date ....

Contents