Tomus 6. Fasciculus 1.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1983

Curat: Universitas Szegediensis de Attila József nominata

Tomus 6.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1984

Curat: Universitas Szegediensis de Attila József nominata

6. kötet 1. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK
KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1983. április
A Szegedi József Attila Tudományegyetem gondozásában

INDEX

Tomus 6.

О разработке инструментальных систем, ориентированных на решение информационно-логических задач

Р. Г. Бухараев, А. И. Еникеев, И. И. Макаров

Практика использования вычислительных машин для автоматизации исследовательских и проектных работ выдвинула на первое место среди множества классов решаемых задач задачи информационно-логического типа. Основные из них — это формульные преобразования, машинные эксперименты по моделированию различных процессов, информационный поиск и т.п. Сложность и многообразие структур данных, а также программных средств их обработки, характерные для упомянутого класса задач, обусловили необходимость эффективного управления процессами использования и создания соответствующих пакетов прикладных программ. Средства, имеющиеся в существующих операционных системах, не обладают достаточной гибкостью и эффективностью для генерации программ в пакетах. Использование только традиционных языков программирования в качестве входных для пакетов программ также не обеспечивает упомянутые выше возможности из-за отсутствия в них средств, позволяющих осуществлять качественный диалог, динамическое планирование решения задач, эффективное структурирование программ и адекватность представления моделей предметных областей конструкциями этих языков.

Все это привело к необходимости создания концепции инструментальных систем программирования, позвольяющих эффективное создание пакетов прикладных программ и систем программирования с проблемной ориентацией. Эта концепция, наиболее полно рассмотренная в работе [1], получила свое развитие в разработках интегрированных систем программирования, представляющих разновидность инструментальных систем с совместимостью видов проблемноориентированных данных в разных пакетах [8], расширяющейся системы автоматизации проектирования вычислительных машин [2] и диалоговых систем для решения информационно-логических задач [4].

В докладе предлагается один из подходов к разработке такого рода систем с диалоговыми возможностями, обеспечивающих: 1) эффективное создание проблемно-ориентированных языков для пакетов прикладных программ; 2) эффективную генерацию новых программ пакета на основании имеющихся; 3) динамическое планирование решения задач на вычислительной машине в режиме диалога. Этот подход был использован при создании системы МАТИСС [6]. Особенность последней состоит в едином концептуальном подходе к разработке системы, основанном на оптимальном сочетании таких требований, как расширяемость входного языка системы, возможность эффективного и гибкого управления диалоговыми средствами, наличие средств структурирования и обработки данных (в том числе и программ), обеспечивающих, в частности, возможности динамического преобразования программ в режиме диалога, выполнения смешанных вычислений и формульных преобразований. Одним из основных принципов построения системы явилось создание базисного языка и средств расширения, позволяющих последовательно надстраивать над базисным языком множество проблемно-ориентированных подсистем для решения информационно-логических задач.

Базисный язык является входным языком первой очереди реализации системы и позволяет:

— описание синтаксиса и семантики вводимых элементов расширения;

— непосредственное составление программ для решения как вычислительных, так и информационно-логических задач, связанных с обработкой сложных структур данных;

— в режиме диалога динамическое изменение структуры программы на стадии ее выполнения с целью гибкой адаптации алгоритма к условиям, возникающим в процессе решения задачи.

Для обеспечения перечисленных возможностей в базисный язык вколючены средства:

а) выполнения арифметических, логических и текстовых операций;
в) обработки структур данных и программ;
с) расширения каждого уровня языка;
д) диалогового взаимодействия;
е) формульных преобразований.

К особенностям языка можно отнести следующие:

— допускается возможность смешанного (частичного) вычисления выражений, в которых не для всех переменных к моменту вычислений определены значения;

— допускается возможность динамического описания переменных; в этом случае тип переменной однозначно определяется в текущий момент работы программы;

— в операторах формульных преобразований допускается использование условных соотношений;

— язык позволяет реализацию рекурсивных обращений (реализация рекурсии выполнена аналогично ЛИСП-системам [3]);

— диалоговые средства языка обеспечивают планирование точек выхода на диалог по заданной в программе системе условий;

— синтаксис языка описывается $LL[K]$ грамматикой [5], позволяющей иметь однопроходной синтаксический анализатор.

Последняя особенность языка вызвана необходимостью повышения реактивности диалога, в котором динамическое изменение структуры программы требует компиляции фрагментов изменения.

Средства расширения системы позволяют расширять входной язык путем введения в него новых операций (функций), отношений и операторов. Расширение сводится к введению пары (s, g), где s — синтаксическое, а g — семантическое описание элемента расширения входного языка. Введенная пара запоминается в таблице соответствий между синтаксическими и семантическими описаниями, которая служит исходной информацией для настройки компилятора на соответствующий уровень расширения входного языка системы. Семантические описания, задаваемые в виде исходных программ перед записью в таблицу предварительно компилируются. Для составления семантических программ можно использовать базисный язык, любой уровень расширения входного языка, а также языки, входящие в стандартное математическое обеспечение.

Одним из основных принципов расширения языка является принцип модульности, позволяющий выделение отдельных уровней расширения системы в автономные подсистемы, работающие независимо от всей системы. При реализации средств расширения был использован опыт разработки системы ПРОЕКТ [2].

Важным моментом реализации диалоговой системы является соотношение компиляции и интерпретации, поскольку чистая интерпретация увеличивает время выполнения программ, а обычная компиляция часто бывает неприспособленной к диалогу.

Это обусловило выбор способа компиляции, предусматривающего получение объектной программы в специальном структурированном виде, удобном для интерпретации и динамического изменения программы в режиме диалога. В отличие от реализации ЛИСП-подобных систем [3] здесь предлагается способ внутреннего представления программы, основанный на обратной польской записи выражений, сокращающий время интерпретации за счёт ликвидации лишних проходов по структуре программы и позволяющий эффективную реализацию режимов коллективного доступа.

Программы, реализующие функции и операторы базисного языка, представляются в виде машинных модулей. Функциям и операторам расширенного языка, семантические программы которых были составлены на входном языке системы, соответствуют структурированные представления программ.

Подобное представление программ достаточно удобно реализуется интерпретатором с использованием стэка. В процессе прохода по структуре программы интерпретатор заносит в стэк данные и выполняет встречающиеся модули. Входные параметры для своей работы каждый такой модуль соответственно выбирает из стэка. Результат работы модуля также заносится в стэк.

Такой подход к внутреннему представлению и интерпретации программ естественным образом позволяет применить методику разработки систем коллективного доступа, предложенную в работе [7] для реализации режима разделения времени, основанного на принципе квантования по модулям соответствующих представлений программ. В отличие от традиционного квантования по времени, упомянутая выше реализация достигается за счёт разрешения прерываний для перехода на обслуживание другого процесса только после полного завершения работы очередного модуля и позволяет: а) упростить структуру управления процессами; б) минимизировать количество запоминаемой при прерывании информации.

Однако такой принцип реализации режима разделения времени является целесообразным, если продолжительность работы соответствующих модулей сравнительно невелика и может быть оценена на стадии их разработки. Поэтому на разработку системных программ были наложены следующие требования:

1) программная конструкция вида α_1; CALL, β; α_2, где α_1 начальный участок программы до оператора вызова подпрограммы β, а α_2 конечный участок, должна быть заменена цепочкой программ α_1, β, α_2, выполняемых в перечисленной последовательности (если время работы программы β сравнительно невелико);

2) exли время работы какой-либо из программ увеличивается в зависимости от величины обрабатываемых данных, то конструкция такой программы должна обеспечивать выход после обработки определенной порции информации на управляющую программу, которая после перехода на обслуживание данного процесса может запустить ту же самую программу для обработки следуюшей порции данных.

Следует отметить, что указанный выше подход к интерпретации программ позволил также естественным образом реализовать возможность динамического планирования точек перехода в программах в зависимости от задаваемой совокупности условий (аналог планирования программных прерываний в операционной системе). Последняя возможность обеспечивает: а) эффективное тестирование программ; б) планирование точек выхода на диалог.

Дополнительно к рассмотренному выше представлению программ в систему включены средства получения и обработки полной структурированной формы представления выражений, основанной на прямой польской записи (аналогично представлению выражений в ЛИСПе). Последняя форма представления является эффективной в формульных преобразованиях и смешанных вычислениях выражений.

В состав программного обеспечения системы входят: 1) управляющая программа, обеспечивающая режимы разделения времени и мультипрограммирования; 2) макропроцессор, обеспечивающий процессы расширения и компиляции входного языка; 3) интерпретатор, реализующий программы в специальном внутреннем представлении и управляющий диалоговыми средствами; 4) обрабатывающие программы, непосредственно реализующие операторы и функции входного языка; 5) служебные программы, устанавливающие связь с операционной системой.

Первая очередь системы реализована в рамках дисковой операционной системы ДОС ЕС.

КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ТЕОРЕТИЧЕСКОЙ КИБЕРНЕТИКИ
УЛ. ЛЕНИНА 18
420008 КАЗАНЬ, СССР

Литература

[1] Тамм, Б. Г., Э. Х. Тыугу, О создании проблемно-ориентированного программного обеспечения, *Кибернетика,* № 4, К., 1975.

[2] Глушков, В. М., Ю. В. Капитонова, А. А. Летичевский, Автоматизация проектирования вычислительных машин, *Наукова думка,* Киев, 1975.

[3] Лавров, С. С., Г. С. Силагадзе, Автоматическая обработка данных, Язык ЛИСП и его реализация, *Наука,* Москва, 1978.

[4] Брябрин, В. М. Г. В. Сенин, Руководство к системе ЛИЛОС-БЭСМ—6, ВЦ АН СССР, Москва, 1977.

[5] Ахо, А. Дж. Ульман, Теория синтаксического анализа, перевода и компиляции, Т. I., *Мир,* Москва, 1978.

[6] Бухараев, Р. Г., А. И. Еникеев, И. И. Макаров, Вопросы эффективного управления пакетами прикладных программ в режиме диалога, *Труды Советско-финского симпозиума по интерактивным системам,* Тбилиси, 1979.

[7] Замов, Н. К., Н. И. Звягина, Р. К. Самитов, Автоматизированная обучающая система Гамма, Тезисы докл, 1 Всесоюзной конференции *Человеко-машинное обучение системы,* (г. Телави), Москва, 1979.

[8] Lopez, L. A., POLO-problem oriented language organizer, *Comput. & Structures,* v. 2, 1972.

(Поступило 3-ого декабря 1981 г.)

On the verification of abstract data types

By L. VARGA

This paper describes a method for verifying the correctness of an abstract data type specification according to a concept about the abstract data type. The abstract data types are formally defined in terms of the algebraic specification technique. General rules are given for constructing theorems about a given abstract data type and for proving the theorems. These theorems serve to convince us of that the given specification correctly takes the meaning of our concept. The method is illustrated by an example.

1. Introduction

One of the most important achievements of programming methodology is the data abstraction. During the recent years, a number of different specification techniques for abstract data types has been proposed. Among these is the algebraic specification method which is described in detailed in Guttag's and Horning's common paper [1] and in an excellent tutorial paper [2].

It is known that a specification must be able to provide separate pictures to its user and its implementor. This two requirements can be satisfied by double specification [4]. A double specification has an abstract and a concrete specification part. An abstract specification is to serve the user's view and a concrete specification is to serve the implementor's view.

In the case of a double specification a verification of the correctness of an implementation according to the concept which is in our mind about the given data type can be carried out in two steps: First we show that an abstract specification correctly reflects the concept and next we verify the correctness of a concrete specification according to the abstract one.

In this paper we will be concerned only with the first phase of this verifictaion procedure when an abstract specification is given in terms the algebraic specification technique.

In section 2 a practical example of algebraic specification is given. In section 3 we provide a method for constructing theorems about the abstract data type and general rules for proving the theorems. The verification is illustrated by an example.

2. Algebraic specification, an example

Data abstraction includes a set of objects and a group of functions that operate upon this object set. An access to an object is only possible through one of the functions. A given set of objects with the functions or operations forms an abstract data type.

A composite abstract data type consists of elementary objects too. We must distinguish between an object to be defined and an elementary object, whose properties are assumed to be known. The functions of an abstract data type may include parameters. Hence a simple model of abstract data type can be characterized by three set (a set of objects, a set of elementary objects, a set of parameters) and a group of functions. Let the names of the above sets be *object, elem* and *parameter* respectively. The domain of a function is generally a cartesian product of the above sets, and its range may be a set of objects or the set of elementary objects.

The group of functions can be divided into two blocks. The first block consists of *constructor functions,* that can be used to build every values of the object set. Hence the range of a constructor function should be the set of object. The second block consists of non constructor function, called *selector functions* because these functions can be used to select parts of an object, for example an elementary object from the object structure or the remains. The range of a selector function generally is the set of elementary objects or the object set itself.

We must distinguish a *generator object* from the remains. The generator object is distinguished by the property that each value of an object set can be generated from it by applying constructor functions one after the others.

A specification method must be suitable for specifying both the syntax and the semantics of the operations.

Now let us see a solution of the specification problem given by the algebraic specification method and choose a simple case of the general abstract data model as an example for illustrating both the specification and verification method.

Let the name of the abstract type be "object"

Syntax

null: $\to$ object
assign: object$\times$parameter$\times$elem$\to$object
delete: object$\times$parameter$\to$object
read: object$\times$parameter$\to$elem

Constructors: null, assign

Semantics

s: object; *p:* parameter; *e:* elem;
read (assign (s, p, e), p')=
    if $p=p'$ *then* e *else* read (s, p')
delete (assign (s, p, e), p')=
    if $p=p'$ *then* delete (s, p)
        else assign (delete (s, p'), p, e)
read (null, p)=readerror
delete (null, p)=null

Auxiliary function

length: object → integer
length (null) = 0
length (assign (s, p, e)) =
length (delete $(s, p)+1$)

Abstract invariant

$I_a(s)$: $0 \leqq \text{length}(s) \leqq n$

Equality

$s_1 = s_2 \equiv (s_1 = \text{null} \wedge s_2 = \text{null}) \vee$
$(\forall p)(\text{read}(s_1, p) = \text{read}(s_2, p) \wedge \text{delete}(s_1, p) = \text{delete}(s_2, p))$.

The auxiliary function is distinguished from the other functions by the property that it is not used by the programs using the abstract data type. It is only a specification tool.

The abstract invariant is used to define a bounded object set:

$$\{s | I_a(s)\}$$

The equality axiom reduces the equality of two objects to the equality of their appropriate parts.

3. A proof method

Given an algebraic data type specification and a concept about the same data type, we have to show that the given specification correctly takes the meaning of verifying the correctness of a specification according to a concept, but we can convince ourself of the correctness by proving theorems about an abstract data type given by an algebraic specification. General rules for deriving such theorems are the followings:

1. The semantics of an abstract type is defined by the effects of each selector operation on an abstract object when this object is produced by a constructor operation. Other relations between two operations can be formulated as theorems and can be proved.

For example, in the case of our abstract data type all the theorems generated in this way are:

Theorem a,
assign (assign $(s, p_1, e_1), p_2, e_2)$ = assign (assign $(s, p_2, e_2), p_1, e_1)$, if $p_1 \neq p_2$

Theorem b,
delete (delete $(s, p_1), p_2)$ = delete (delete $(s, p_2), p_1)$,

Theorem c,
read (delete $(s, p_1), p_2)$ = read (s, p_2) if $p_1 \neq p_2$

2. Selector functions map an abstract object to its components. Therefore we have to show that an abstract object can be reconstructed from its selected components by constructor operations. In the case of our example we can generate only one theorem in this way:

Theorem d,

assign (delete (s, p), p, read (s, p))$=s$

We have two general rules for proving these theorems. One of them is the induction and another is that of applying the equality formula for both sides of a theorem.

The induction steps are the following:

First we show that the theorem holds for the generator object. Then supposing that the theorem holds for each element of a subset of the object set we have to prove that the theorem holds for each object generated by a constructor operation from the given subset.

We now use these rules for verifying the above theorems.

Proof of Theorem c,

We prove the Theorem by induction on s.

Basis. If $s=$null, then

delete (null, p_1)$=$null

and the result is immediate.

Induction step. Now choose

$s'=$assign (s, q, e)

and suppose that our theorem is true for s. We have to prove the theorem for s' too. It follows from semantics axioms that

read (delete (s', p_1), p_2)$=$

$=$read (delete (s, q), p_2), if $q=p_1 \wedge p_2 \neq p_1$

$=e$, if $q \neq p_1 \wedge q=p_2$

$=$read (delete (s, p_1), p_2), if $q \neq p_1 \wedge q \neq p_2$

We have

read (delete (s, q), p_2)$=$read (s, p_2), if $q \neq p_2$

read (delete (s, p_1), p_2)$=$read (s, p_2) if $p_1 \neq p_2$

by our induction hypothesis. The equation

read (assign (s, q, e), p_2)$=e$, if $p_2=q$

follows from the semantics of the read operation.

Proof of Theorem a,

The equality definition can be simplified by using Theorem c,. Hence the new equality formula is

$s_1=s_2=(s_1=\text{null} \wedge s_2=\text{null}) \vee (\forall p)$ (read $(s_1, p)=$read (s_2, p)).

Using this definition of equality we have to show that the read operation for both sides of Theorem a, gives the same result what ever is p. The problem can be broken up into three cases corresponding to the relations among the three parameters

1. $p=p_1$

For the left hand side:

read (assign (assign (s, p_1, e_1), p, e_2), p)$=e_2$

and for the right side we have

read (assign (assign (s, p, e_2), p_1, e_1), p)$=$

read (assign (s, p, e_2), p)$=e_2$

by using the axiom for the semantics of a read operation.

2. $p \neq p_1$ $p \neq p_2$

For the left side:

read (assign (assign (s, p_1, e_1), p_2, e_2), p)=
read (assign (s, p_1, e_1), p)=read (s, p)

and for the right side:

read (assign (assign (s, p_2, e_2), p_1, e_1), p)=
read (assign (s, p_2, e_2), p)=read (s, p)

Proof of Theorem b,

Induction on s:

Basis. s=null. Then the theorem is trivially true:

Induction step. Suppose s=assign (s_0, p, e), where the theorem is true for s_0. Then

1. $p=p_1$
delete (delete (assign (s_0, p, e), p_1), p_2)=
delete (delete (s_0, p_1), p_2)

and

delete (delete (assign (s_0, p, e), p_2), p_1)=
delete (assign (delete (s_0, p_2), p, e), p_1)=
delete (delete (s_0, p_2), p_1)=
delete (delete (s_0, p_1), p_2).

2. $p=p_2$. Symmetrical case.
3. $p \neq p_1 \wedge p \neq p_2$.
delete (delete (assign (s_0, p, e), p_1), p_2)=
delete (assign (delete (s_0, p_1), p, e), p_2)=
assign (delete (delete (s_0, p_1), p_2) p, e)

and similarly for the other side we have

delete (delete (assign (s_0, p, e), p_1), p_2)=
assign (delete (delete (s_0, p_2), p_1), p, e)=
assign (delete (delete (s_0, p_1), p_2), p, e).

Proof of Theorem d,

We can prove the theorem by the axiom of equality. For the right hand side:

1. $q=p$
read (assign (delete (s, p), p, read (s, p)), q)=
read (s, q)
2. $q \neq p$
read (assign (delete (s, p), p, read (s, p)), q)=
read (delete (s, p), q)=read (s, q)

4. Closing comments

The procedural nature of a data type, which is important for the implementor, can be described by the Hoare-like specification [3]. In this case we have an abstract specification in algebraic form and a concrete specification in Hoare-like form. We have no direct method to verify the correctness of a Hoare-like specification according to an algebraic specification. However it is easy to transform an algebraic specification into Hoare-like form and then we can use the verification method given by Hoare [3].

EÖTVÖS LORÁND UNIVERSITY,
MÚZEUM KRT. 6—8.
BUDAPEST, HUNGARY
H—1088

References

[1] GUTTAG, J. V. and J, HORNING, The algebraic specification of abstract data types, *Acta Inform.*, v. 10, 1978, pp. 27—52.
[2] GUTTAG, J. V., Notes on type abstraction, *IEEE Trans. Software Engrg.*, SE—6, 1980, pp. 13—23.
[3] HOARE, C. A. R., Proofs of correctness of data representations, *Acta Inform.*, v. 1, 1972, pp. 271—281.
[4] WULF, W. A., R. L. LONDON and M. SHAW, An introduction to the construction and verification of Alphard programs, *IEEE Trans. Software Engrg.*, SE—2, 1976, pp. 253—265.

(Received March 2, 1982)

Data structures and storage structures in scientific data base for multistage experiment

By L. BORZEMSKI

The aim of this paper is twofold. One of them is to present a multistage identification experiment and the other is concerned with the design of data base organization in that application. Some of the essential characteristics of multistage experiment are outlined. On the basis of the multiple name structure the data model is introduced. Then a framework for data representation within records is proposed and analyzed. It is shown that well-known multiple attribute retrieval methods can be applied in this case.

1. Introduction

The fast growing computer capability for data handling focused attention on the construction of powerful information laboratory systems which could carry out the process of automated experimentation, especially with the enormous amount of data. In scientific laboratory calculations there are many examples of tasks that require the processing of large data sets.

The aim of this paper is twofold. One of them is to present the multistage identification experiment environment and the other is concerned with the design of storage structures in that application. The multistage identification is vital to the efficient data manipulation in system identification. There are many aspects of data manipulation that require data base supporting.

The data model is presented in terms of Turski's data structure theory [7]. Physical representation of the data within an information system in the multistage experiment environment is proposed and analyzed. It is shown that well-known multiple attribute retrieval methods can be applied in this case.

2. The multistage experiment environment

The following short description is only intended to indicate the nature of the tasks undertaken in the multistage identification. The details have been published in [3, 5]. Fig. 1 shows a flow chart for the multistage identification experiment. In the multistage identification we perform the identification of the system in such a way

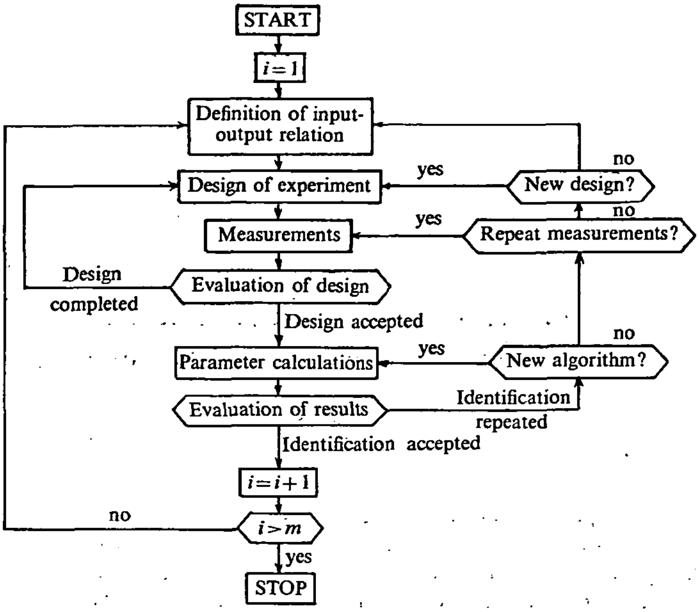


Fig. 1
Flow chart for the multistage identification experiment

that the global model $\bar{y}=\Phi(x_1, x_2, \ldots, x_m, a_m)$ is considered to be decomposed into m submodels $a_{i-1}=\Phi_i(x_i, a_i)$, $a_0 \triangleq \bar{y}$, $i=1, 2, \ldots, m$. We can measure input signals at each stage and output signals at the first stage only. The strategy of experiment is falling into m stages where at each stage we perform N_i elementary experiments, where

$$N_i = \begin{cases} 1, & \text{for} \quad i=m, \\ \prod\limits_{l=i+1}^{m} n_l & \text{for} \quad i<m. \end{cases}$$

Throughout this paper each elementary experiment is assumed to have the following experiment attributes: identifier of the experiment NR_i, number of observations n_i, identification criterion Q_{i,n_i}, matrix of inputs X_{i,n_i}, matrix of outputs A_{i-1,n_i}, vector of model parameters a_{i,n_i}, table of vectors M_i which consists of the input vectors at the $(i+1)$, $(i+2)$, ..., m-th stages assumed constant during the identification at the i-th stage. The above ordering of experiment attributes will be assumed through this study. To obtain the matrix of observations of a_{i-1}, $i=\overline{2, m}$, it is necessary to repeat the experiments at the $(i-1)$-th stage n_i times assuming different values of x_i. It can be seen that for $i<m$ vector of model parameters a_i at the i-th stage is a column of output matrix $A_{i,n_{i+1}}=[a_{i,1}, \ldots, a_{i,n_{i+1}}]$ at the $(i+1)$-th stage. For every table M_{i-1} at the $(i-1)$-th stage there exists the table of vectors at the i-th stage which shares the same input vectors. For $i<m-1$ there exists for every table M_i' the table M_i'' at the same stage which differs only in one element, i.e. one vector.

If the designs of elementary experiments at the first stage are the same then all X_{1,n_1} matrices contain the same data.

The list of the system users consists of the experiment design programs, programs for identification, control of experiment and statistical calculations, and of experimentators, as well.

There exists a number of anticipated user requests [2, 4]. Almost all users are rather non-computer oriented and the total task coding in high level language is divided between them. Experimental data are collected and processed in a complex way. Tasks involve the iterative execution of several computer programs, each requiring data generated by the others in addition to user input data. The data are generalized FORTRAN arrays of numeric data. Multiple read/write requests for the same data in different applications programs are observed. Users manipulate on different data aggregates (e.g. single numeric data, vectors, arrays). Some data are collected and demanded in different ways, for example input matrix is generated vector by vector but requested also row by row. The data in a multistage experiment have some distinct characteristics, namely the regularity in their multiple relationships, redundancy and constant growth. Usually, moderate size data bases are involved but the size is increased when the data from experiments carried out at the different laboratories are to be bound together. In [2] an experiment with three stages is considered where the size of data base is of the order of 10^7 bytes of "pure" information, without any organizational data.

These characteristics give rise to the need for data base to support data manipulation in computer-based multistage experiment laboratory system. It should be noted that in laboratories we use minicomputers or microcomputers that considerably restrict us in data base facility choise.

3. Data description

The data model will be presented in terms of Turski's data structure theory [7]. Then data are considered to be ordered pairs (n, v) such that $n \in \mathcal{N}$, $v \in \vartheta$, where $\mathcal{N}$ is a denumerable set of elements called "names" which distinguish the entity in the real or abstract world and ϑ is any set of values considered as the collection of the information pertaining to the properties termed by the name.

We shall use a multiple name [2] which is defined as $\mathbf{n}_\alpha = \bigcap_{i \in \alpha} (n_i = a_i)$, where α is a string of integers selected unrepetitively from the set $\{1, 2, ..., r\}$, $a_i \in A_i$ — domain value of name part n_i, $\mathbf{n}_\alpha \in \mathcal{N}$. In this paper we assume that each name part takes value from the finite subset of natural numbers with cardinality $\bar{\bar{A}}_i$.

Using the above approach we can construct r — level ordered unbalanced multiple name tree (sorted lexicographically) with the dummy root at the top so that the unique path connecting the root node to a terminal node corresponds to a distinct multiple name $\mathbf{n}_\alpha$ with $\alpha = 1, 2, ..., r$.

Let us first define domain sets A_i in such a way that name part n_i describes the multistage experiment number, stage number, experiment number at given stage, experiment attribute number, observation number and element number, for $i = \overline{1, 6}$, respectively. It is obvious that n_5, n_6 or n_6 are greater than one only for arrays or vector data, according to the n_4 value. The cardinal numbers of the sets A_i, $i=\overline{1,6}$ are then defined in the following way

$$\bar{A}_1 = E, \qquad \bar{A}_4 = 7$$

$$\bar{A}_2 = \max_{k=\overline{1,E}} \{m_k\}, \qquad \bar{A}_5 = \max_{\substack{k=\overline{1,E}\\ i=\overline{1,m_k}}} \{n_{k,i}\},$$

$$\bar{A}_3 = \max_{k=\overline{1,E}} \{N_{k,1}\}, \qquad \bar{A}_6 = \max_{\substack{k=\overline{1,E}\\ i=\overline{1,m_k}}} \{s_{k,i}, k_{k,i}, l_{k,0}\},$$

where the following parameters for the i-th stage of the k-th multistage experiment, $i=\overline{1,m_k}$, $k=\overline{1,E}$ are given:

$n_{k,i}$ the number of observations,
$s_{k,i}$ the number of inputs,
$l_{k,i-1}$ the number of outputs,
$k_{k,i}$ the number of unknown parameters,
$N_{k,i}$ the number of elementary experiments.

E and m_k denote the number of multistage experiments and the number of stages of the k-th multistage experiment, respectively.

We also define in Table 1 three additional parameters $q_{k,i}^{(p)}$, $h_{k,i}^{(p)}$, $g_{k,i}^{(p)}$, where $k=\overline{1,E}$, $i=\overline{1,m_k}$, $p=\overline{1,\bar{A}_4}$.

Table 1. Definition of $q_{k,i}^{(p)}, h_{k,i}^{(p)}, g_{k,i}^{(p)}$ parameters

p	attribute	$q_{k,i}^{(p)}$	$h_{k,i}^{(p)}$	$g_{k,i}^{(p)}$
1	NR_i	1	1	1
2	n_i	1	1	1
3	Q_{i,n_i}	1	1	1
4	X_{i,n_i}	$n_{k,i}s_{k,i}$	$n_{k,i}$	$s_{k,i}$
5	A_{i-1,n_i}	$n_{k,i}l_{k,i-1}$	$n_{k,i}$	$l_{k,i-1}$
6	a_{i,n_i}	$k_{k,i}$	1	$k_{k,i}$
7	M_i	s_z	m_k-i	s_M

where

$$s_z = \begin{cases} \sum_{j=i+1}^{m_k} s_{k,j}, & \text{for} \quad i < m_k, \\ 0, & \text{for} \quad i = m_k. \end{cases}$$

$$s_M = \begin{cases} \max_{\substack{k=\overline{1,E}\\ i=\overline{2,m_k}}} \{s_{k,i}\}, & \text{for} \quad i < m_k, \\ 0, & \text{for} \quad i = m_k. \end{cases}$$

The model concept can be extended for any set of known experiment attributes with appropriate $q_{k,i}^{(p)}$, $h_{k,i}^{(p)}$, $g_{k,i}^{(p)}$ parameters.

The great advantage of this data description method is that the construction of the multiple name tree reflects the structural properties of data aggregates in a multistage experiment. To illustrate this consider a data base pertaining to a multistage experiment in which we have three stages with four, two, three observations at each stage, respectively. At each stage the models have two, one and one input signals respectively, and four, two, three model parameters. We consider only one output at the first stage. Then, we have got six, three and one experiments at the first, second and third stage, respectively. Within this example, $\mathbf{n}=(1, 1, 2, 4, 3, *)$ describes the third vector of the input matrix in the second experiment at the first stage (Fig. 2). This way we can indicate all data aggregates, not necessary to be logically clustered but others, partitioning through a data base as well. For instance, $\mathbf{n}=(1, 1, *, 3, *, *)$ is related to all identification criterion values at the first stage.

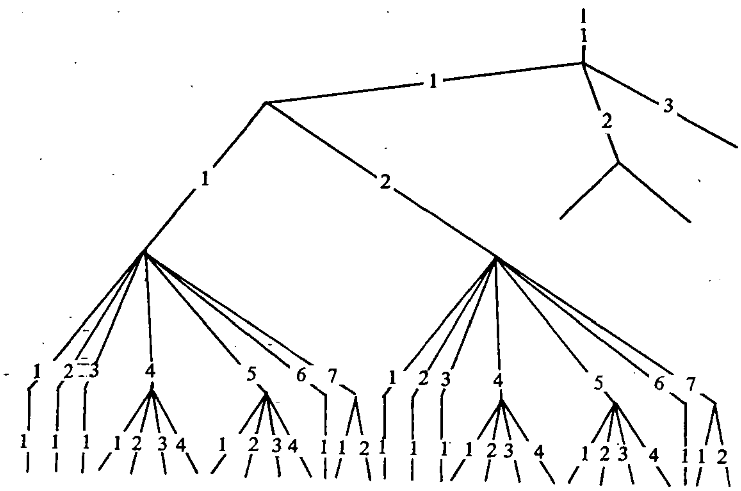


Fig. 2
A part of the multiple name tree for $\mathfrak{z}=5$.

In defining our name structure (the name admissibility verification algorithm is given in [1]) we note that the logical structure of data is stable in the sense of its construction and data relationships. As far as we could do, we have took advantage of this in designing the multiple name tree which exactly describes the real world. Another property which is observed is that the multiple name tree ensures the hierarchical clustering of data in the so-called "top-down" searching. Since this tree is given when all required multistage experiment parameters are known then almost all data management issues can be handled in arithmetic way [2]. In particular, data base storage structures can be constructed using this view what we will present in the next section. In general, the user of a data base may ask a wide variety of questions about the data that are stored. However, in this application the types of queries can be anticipated and the storage structures may be designed to handle them with suitable cost.

4. Physical data organization

The data within an information system may be stored in very different forms. Traditionally, the data base is a series of "physical" records which are formed by interconnecting some set of data items via storage structure. Each data item is a representation of (attribute, value) pair which characterizes an entity. Precise definitions for that can be found in [6]. In a business system environment there are many data base organizations which support the classification of user concepts into entities, attributes and relationships. In the multistage experiment environment we need some way of having unifying framework to represent data described in the previous section. The way this is done is to have a set of records such that each record attached to a terminal node of a multiple name tree represents a data corresponding to the name path $\mathbf{v}$ connecting the root node to that terminal node. One can easily see that only one record is attached to each terminal node. Then, for a multiple name tree with all leaves at depth $\mathfrak{z}$[1] a record stores information pertaining to the following entities:

— complete multistage experiment,
— all experiments at a given stage,
— an elementary experiment,
— an experiment attribute,
— a vector,
— an element (variable)

for $\mathfrak{z}=\overline{1, 6}$, respectively.

Note, that for the particular values of $\mathfrak{z}$ we found that different terminal nodes can refer to the same physical data, for example if $\mathfrak{z}=4$, $h=\overline{1, N_i}$ then $\mathbf{n}=(1, i, h, 4, *, *)$ indicates the same input matrix shared in h experiments at the i-th stage. To avoid redundancy there can be one storage copy of the record which stores appropriate values according to the multistage experiment strategy. Appropriate algorithm for recognizing these situations has been developed [2].

Other meaningful parameter e is to describe structural properties of data stored in a record. It is defined in a similar way and indicates how the data within a record is structurally divisible. If $\mathfrak{z}=e$, we assume that physical record stores structurally nondivisible data, e.g. for $\mathfrak{z}=4$, a value of experiment attribute. If $e>\mathfrak{z}$, it results that every record at level $\mathfrak{z}$ stores a sequence of data items which can have either one or more elements, i.e. for $\mathfrak{z}=4$, $e=5$ we obtain vectors or variables according to the $\mathbf{v}_4$ value. The data items are ordered in lexicographically ascending order of their multiple names. We also define parameter f which depends on the manner in which the physical boundaries of data items are fixed in a record, considering four storage formats, namely positional, relational, indexed and labeled [6].

Now we develop a series of equations for evaluating the space requirements under each of combination of $\mathfrak{z}$, e, f, $\mathbf{v}$ values. The results are given in Table 2. The storage allocation scheme assumes that all elements are allotted the same number of machine storage units (e.g. words, bytes). It depends on the storage allocation for REAL/DOUBLE PRECISION variables. We obtain $b_j(z, \mathfrak{z}, \mathbf{v}_j)$ storage units for representing the data of name $\mathbf{v}_j$ within the j-th record, $j=\overline{1, F(z, \mathfrak{z})}$ at the $\mathfrak{z}$-level

---

[1] The root of a tree lies at depth 0; the son of a node at depth $(i-1)$ lies at depth i.

Table 2. Equations of space allocation

$\mathfrak{z}$	$F(z, \mathfrak{z})$	$b_j(z, \mathfrak{z}, \mathbf{v}_j)$
1	E	$\sum_{l=1}^{m_{v_1}} N_{v_1,l} \sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{v_1,l}$
2	$\sum_{k=1}^{E} m_k$	$N_{v_1,v_2} \sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{v_1,v_2}$
3	$\sum_{k=1}^{E} N_k$	$\sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{v_1,v_2}$
4	$\bar{\bar{A}}_4 \sum_{k=1}^{E} N_k - E$	$g^{(v_4)}_{v_1,v_2}$
5	$\sum_{k=1}^{E} \sum_{i=1}^{m_k} N_{k,i} \sum_{p=1}^{\bar{\bar{A}}_4} q^{(p)}_{k,i}$	$h^{(v_4)}_{v_1,v_2}$
6	$\sum_{k=1}^{E} \sum_{i=1}^{m_k} N_{k,i} \sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{k,i}$	1

	$F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$			
$\mathfrak{z}$ \ e	1	2	3	4
1	1	m_{v_1}	N_{v_1}	$\bar{\bar{A}}_4 N_{v_1} - E$
2		1	N_{v_1,v_2}	$\bar{\bar{A}}_4 N_{v_1,v_2} - E\xi(v_2)$
3			1	$\bar{\bar{A}}_4 - \xi(v_2)$
4				1
5				
6				

$\mathfrak{z}$ \ e	$F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$	
	5	6
1	$\sum_{i=1}^{m_{v_1}} N_{v_1,i} \sum_{p=1}^{\bar{\bar{A}}_4} q^{(p)}_{v_1,i}$	$\sum_{i=1}^{m_{v_1}} N_{v_1,i} \sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{v_1,i}$
2	$N_{v_1,v_2} \sum_{p=1}^{\bar{\bar{A}}_4} q^{(p)}_{v_1,v_2}$	$N_{v_1,v_2} \sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{v_1,v_2}$
3	$\sum_{p=1}^{\bar{\bar{A}}_4} q^{(p)}_{v_1,v_2}$	$\sum_{p=1}^{\bar{\bar{A}}_4} g^{(p)}_{v_1,v_2}$
4	$q^{(v_4)}_{v_1,v_2}$	$g^{(v_4)}_{v_1,v_2}$
5	1	$h^{(v_4)}_{v_1,v_2}$
6		1

Table 2. Equations of space allocation (cd)

of data aggregation. The number of records $F(z, \mathfrak{z})$ is the number of leaf nodes at depth $\mathfrak{z}$. We assume z is a vector of parameters which characterizes the multistage experiment i.e. number of observations, number of inputs, outputs and model parameters, and number of multistage experiments within an information system. The total length of the j-th record w_j is $b_j + c_j$, where c_j is equal to $I + \delta(\mathfrak{z}, f) \cdot \cdot F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$. Within a file every record is assigned a label (identifier) which occupies I storage units. $\delta(\mathfrak{z}, f)$ is the additional space required by storage technique of data items within a record per data item. $F_j(z, \mathfrak{z}, e, \mathbf{v}_j)$ is a number of data items within the j-th record and $\xi(i) = 1$, if $i = m_k$, and $\xi(i) = 0$, otherwise. In the similar manner we also find number of storage units for representation each data item within a record [2].

It is pointed out that the above physical representation results in variable length of records. In the event that records may not be divided between buckets (a restriction posed by the majority of operating systems) then one can quickly determine the $\mathfrak{z}$'s value which satisfies this constraint.

The multiple attribute retrieval methods have been found useful for physical record positioning. In the most of them the storage scheme is dictated by the primary key. In [2] we describe an algorithm which generates the identifier on the basis of a mapping from the space of admissible multiple names into the space of integers.

Next, assuming that each part name correlates with an attribute we can obtain six versions of each multiple attribute method. One can determine the $\mathfrak{z}$'s, e's and f's that minimize the expected operational cost of the system [2]. Most of current multiple attribute access methods (for example, Inverted List, Multilist) require storing attribute values in the records [6]. Then a record additionaly contains the identifier and values of indexed attributes. However, in the doubly-chained tree organization the indexed attribute value are not stored in the records. Each case is met with appropriate value of $\delta(\mathfrak{z}, f)$. A detailed comparison of the average retrieval time per query and storage requirements of several current methods can be done.

Considering the limited space of this paper we refer the reader to [2] for the results. Moreover, the mapping function mentioned above establishes a new record addressing technique which further improve the system performance. It will be published elsewhere.

The data base management system which provides a high level access to the data base in the multistage experiment has been implemented at the Technical University of Wrocław [3].

5. Conclusions

Data base organization displays the potential profit in data management efficiency in the multistage experiment environment. Since the characteristics of this application strongly motivate a new strategy for data storage and retrieval, this data base environment was analyzed. The data model has been proposed. In this paper we have also introduced physical data organization which has been found useful for any multiple attribute retrieval method.

TECHNICAL UNIVERSITY OF WROCŁAW
INSTITUTE OF CONTROL AND SYSTEMS ENGINEERING
JANISZEWSKIEGO 11/17,
50—372 WROCŁAW, POLAND

References

[1] BORZEMSKI, L., Data management in a multistage experiment control systems, Proc. Int. Conf. Systems Engineering, Coventry, Sept. 1980.
[2] BORZEMSKI, L., A methodology and data base management algorithms for a multistage experiment control (In Polish), Ph. D. Thesis, Wrocław, 1979.
[3] BORZEMSKI, L., S. LEBIEDIEWA, Data base organization in the multistage identification experiment, IFAC Workshop on Scientific Experiments and Laboratory Procedures SELPA, Smolenice, Nov. 12—14, 1980.
[4] BUBNICKI, Z., Identification of control plants, Elsevier — PWN, Amterdam — Warszawa, 1980.
[5] BUBNICKI, Z., On the multistage identification, *Systems Sci.*, v. 3, 1977, pp. 207—210.
[6] MARTIN, J., *Computer data-base organization*, Prentice-Hall, New York, 1977.
[7] TURSKI, W. M., A model for data structures and its applications, *Acta Inform.*, v. 1, 1971, pp. 26—34.

(Received Dec. 3, 1981)

The early bird problem is unsolvable in a one-dimensional cellular space with 4 states

By H. KLEINE BÜNING

Legendi and Katona (1981) have shown that the early bird problem in a one-dimensional space is solvable with 5 states. The proof is based on a sophisticated concept of waves introduced by Vollmar. We will show that 5 states is a sharp bound for solvability.

1. Early bird problem

Vollmar (1977) defined the problem for a one-dimensional cellular space allowing more than one cell to be excited at a given time step. Only quiescent cells may be excited. Before the first time step at least one cell should be excited. After a certain period the first birds should be in a distinguished state while all the others in a different state.

2. Unsolvability with 4 states

Theorem. The early bird problem is unsolvable in a one-dimensional cellular space with 4 states.

Proof. Assume: There exists a four-state solution, say with a set of states $\{0, B, 2, 3\}$, where

O = initial state

B = bird state (arises only from state 0, spontaneously). Then there is a set of transitions — called A — solving the problem. After a certain period the first bird(s) should be in a distinguished state. The initial state 0 cannot be the distinguished state, because the space is unbounded and after a finite number of steps we obtain a finite configuration.

Case a: B is the distinguished state. There are no transitions $0B0 \to i$, $0BB \to i$, $BB0 \to i$, $BBB \to i$ $(i=0, 2, 3)$ in A, since a bird B cannot be generated by transitions. The set of transitions A must contain a transition $B0B \to 2$ or $B0B \to 3$, otherwise the initial configurations

$K_1 = \dots 0BBBB\underset{\sim}{0}BBBB0\dots$ and

$K_1' = \dots 0BBBBBBBBB0\dots$, where for K_1

a later bird (second step) occurs at the cell marked by $\sim$ would imply the same configuration sequence (after 3 steps). Without loss of generality we assume $B0B \to 2$ belongs to A. Then A contains no transition of the below defined set of transitions D, because a first bird would be killed.

$$D := \{BB2 \to i,\ 2BB \to i,\ 2B2 \to i,\ BB0 \to i,\ 0BB \to i,\ 0B0 \to i,\ BBB \to i \quad (i=0, 2, 3)\}.$$

Now it is investigated a case distinction. Let

$$L_2 := B00 \to 2, \quad R_2 := 00B \to 2,$$

$$L_3 := B00 \to 3, \quad R_3 := 00B \to 3.$$

Case 1: $L_2, R_2 \in A$

Let $K_1 = \ldots 0B000B0\ldots$ be an initial configuration. Then we obtain after one step $K_2 = \ldots 02B202B20\ldots$. In case of birth of a bird we have $K_2^* = \ldots 02B2B2B20\ldots$. Furthermore let $K_1' = \ldots 00B0B0B0\ldots$ be another initial configuration, then we obtain after one step $K_2' = \ldots 02B2B2B20\ldots$. We see that $K_2' = K_2^*$. This shows that a later bird survives. This is a contradiction.

Case 2: $L_3, R_2 \in A$

Let the initial configuration $K_1 = \ldots 0B0B00BB0\ldots$ be given. Then we get after one step $K_2 = \ldots 02B2B32BB30\ldots$. Thus we see that A does not contain the transitions

$$\left.\begin{array}{l} 2B2 \to i \\ 2B3 \to i \\ BB3 \to i \end{array}\right. \quad (i=0, 2, 3) \text{ (otherwise a first bird is killed).}$$

Now let (later birth of birds)

$$K_2' = \ldots 0B0B00B00\ldots 02B2B32BB30\ldots$$

then we obtain

$$K_3' = \ldots 02B2B32BB30\ldots i_1 \ldots i_{12}0\ldots$$

for some $i_j \in \{0, B, 2, 3\}$. Eliminating these later birds is only possible from the right side. Since $BB3 \to i$, $BB2 \to i$, $BB0 \to i$ $(i=0, 2, 3)$ do not belong to A (see above and set D), the later birds cannot be killed. This is a contradiction.

Case 3: $L_2, R_3 \in A$ (analogue to case 2, symmetry)

Case 4: $L_3, R_3 \in A$

Let $K_1 = \ldots 0BB00B0B000B0\ldots$ be an initial configuration, then we get (after one step) $K_2 = \ldots 03BB33B2B303B30\ldots$. Thus we see that A does not contain the transitions

$$\left.\begin{array}{l} 3BB \to i \\ BB3 \to i \\ 3B2 \to i \\ 2B3 \to i \\ 3B3 \to i \end{array}\right. \quad (i=0, 2, 3) \text{ (otherwise a first bird is killed).}$$

Since $BB0 \leftarrow i$, $BB2 \rightarrow i$ $(i=0, 2, 3) \notin A$ (see set D) and above we have seen $BB3 \rightarrow i \notin A$ two later birds — left from the first birds — survive. This is a contradiction.

Case 5: $L_2 \in A$, R_2, $R_3 \notin A$

Let the initial configuration be given

$K_1 = ...0B00B0...$, then we obtain (after one step)
$K_2 = ...00B20B20....$ Now let (birth of bird)
$K_2' = ...0B2BB20...$ and
$K_1^* = ...0B0BB0...$ another initial configuration.

Then we get after one step $K_2^* = ...0B2BB20....$ Since $K_2^* = K_2'$ the later bird survives. This is a contradiction.

Case 6: $R_2 \in A$, L_2, $L_3 \notin A$ (analogue to case 5, symmetry)

Case 7: $L_3 \in A$, R_2, $R_3 \notin A$

Let the initial configuration $K_1 = ...0BBB0...$ be given, then after one step we get $K_2 = ...0BBB30$. Thus we see that $BB3 \rightarrow i \notin A$ $(i=0, 2, 3)$. Since $BB2 \rightarrow i$, $BB0 \rightarrow i \in D$ and therefore not in A, later birds far enough left from the first birds survive. This is a contradiction.

For example:

$K_2' := 0...0\underset{\sim}{B}\underset{\sim}{B}\underset{\sim}{B}0...0...0BBB30...$ ($\sim$birth of birds)

then we get

$K_3' := 0...0BBB30...0i_1...i_{10}0...$ for some $i_j \in \{0, B, 2, 3\}$

Case 8: $R_3 \in A$, L_2, $L_3 \notin A$ (analogue to 7, symmetry)

Case 9: L_2, L_3, R_3, $R_3 \notin A$

Let $K_1 = ...0B00...$ be an initial configuration, then no transition is applicable to K_1. In case of birth of a bird $K_1' = ...0B000B00$, again we cannot apply a transition to K_1'. This is a contradiction, because K_1 and K_1' have the same configuration sequence. Altogether we have shown that the early bird problem is unsolvable with 4 states, where B is the distinguished state..

Next we will consider the distinguished states 2 or 3. Without loss of generality we assume

Case b: 2 is the distinguished state.

Before starting with a case distinction we will prove

Proposition 1. If the set of transitions A solves the problem, then

a) $\exists i_1, i_2 \in \{0, 3\}$: $(220 \rightarrow i_1 \in A,\ 022 \rightarrow i_2 \in A)$ and

$\forall i=0,2$: $(322 \rightarrow i \notin A,\ 223 \rightarrow i \notin A)$ or

b) $\exists i_1, i_2 \in \{0, 3\}$: $(223 \rightarrow i_1 \in A,\ 322 \rightarrow i_2 \in A)$ and

$\forall i=0,3$: $(022 \rightarrow i \notin A,\ 220 \rightarrow i \notin A)$.

Proof. Let us begin with an initial configuration $K_1 = \ldots 0BB0BB0BB0\ldots$ without later birds. After a finite number of steps we obtain a configuration.

$$K_n = \ldots i_2 i_1\, 22\, l\, 22\, m\, 22\, j_1 j_2 \ldots \quad \text{for some} \quad i_t, j_t, l, m \in \{0, 3\}$$

and state 2 remains in the next steps in these cells (no birth of birds).

Thus we see that $l22 \to i$, $22l \to i \notin A$ $(i = 0, 3)$, otherwise the distinguished state 2 is changed.

IF $l = 0$, then $022 \to i$, $220 \to i \notin A$.

IF $322 \to i$ resp. $223 \to i \notin A$ for $i = 0, 3$, then it holds

$$\begin{array}{lll} 322 \to i \text{ resp.} & 223 \to i & \\ 222 \to i & 222 \to i & \text{not in } A. \\ 022 \to i & 220 \to i. & \end{array}$$

Thus we see that two later birds for enough right resp. left from the first birds survive. Therefore $322 \to i_1$ and $223 \to i_2 \in A$ for some $i_1, i_2 \in \{0, 3\}$. If $l = 3$ the proof is similar.

Now we write $XYZ \to$ instead of $\exists i \in \{0, B, 2, 3\} - \{Y\}$: $XYZ \to i$ and $XYZ \to \in A$ means $\exists i \in \{0, B, 2, 3\} - \{Y\}$: $XYZ \to i \in A$.

Let

$$E_1 := 323 \to \qquad E_3 := 023 \to$$

$$E_2 := 320 \to \qquad E_4 := 020 \to .$$

Next we will consider a case distinction.

Case 1: $E_1 \in A$; $E_2, E_3, E_4 \notin A$

Case 1a: $\{223 \to, 322 \to\} \subset A$, $220 \to$, $022 \to \notin A$ (see Prop. 1)

Let $K_1 = \ldots 0BB000\ldots$ be the initial configuration, then we obtain after a finite number of steps

$$K_n = \ldots i_1\, 22\, i_2 \ldots \quad \text{for some} \quad i_1, i_2 \in \{0, 3\}$$

and from hence cells with state 2 remain in state 2. Then $i_2 = i_1 = 0$, because $223 \to$, $322 \to \in A$. Since $E_3 = 023 \to$, $E_4 = 020 \to$, $022 \to \notin A$ two later birds far enough left from the early birds reach state 2 and remain in this state.

Case 1b: $\{220 \to, 022 \to\} \subset A$, $223 \to$, $322 \to \notin A$

If $K_1 = \ldots 0BB0B0BB0$ is an initial configuration, we obtain for some n, $K_n = = \ldots i_1\, 22\, i_3\, \tilde{2} i_4\, 22\, i_5 \ldots$ where state 2 remains (no birth of birds) in these cells. Then $i_1 = i_3 = i_4 = i_5 = 3$, because $220 \to$, $022 \to \in A$. This is a contradiction to $323 \to \in A$. The cell marked by $\sim$ changes its state 2.

If K_1 is an initial configuration (with birds) and there are no later birds, then K_{n+1} denotes the configuration after n steps. If the bird-cells are in the distinguished state 2 and there is no change of states in these bird-cells in the following (without birth of birds) then the configuration is called K^*_{n+1}.

Case 2: $E_2 \in A$; $E_3, E_1, E_4 \notin A$

ad a: $\{220 \to, 022 \to\} \subset A$, $223 \to$, $322 \to \notin A$

If $K_1 = \ldots 0B0BB0B0\ldots$, then $\exists n$:

$K_n^* = \ldots i_1\, 2i_2\, 22i_3\, 2i_4$ for some $i_1, i_2, i_3, i_4 \in \{0, 3\}$.

Since $022\!\to$, $220\!\to\, \in A$, we see $i_2 = i_3 = 3$ and $i_4 = 3$, because $E_2 = 320\!\to\, \in A$. Since $E_1 = 323\!\to$, $E_3 = 023\!\to$, $223\!\to\, \notin A$, later birds ($0B0BB0B0$) far enough right from the first birds reach state 2 and remain in this state.

ad b: $\{223\!\to, 322\!\to\} \subset A$, $220\!\to$, $022\!\to\, \notin A$

If $K_1 = \ldots 0BB0B0\ldots$, then $\exists n$:

$K_n^* = \ldots i_1\, 22i_2\, 2i_3$ for some $i_1, i_2, i_3 \in \{0, 3\}$.

Since $223\!\to$, $322\!\to\, \in A$, it holds $i_1 = i_2 = 0$. Since $E_4 = 020\!\to$, $E_3 = 023\!\to$, $022\!\to\, \notin A$ later birds ($0BB0B$) far enough left from the first one reach state 2 and remain in this state.

Case 3: $E_3 \in A$; $E_1, E_2, E_4 \notin A$ (analogue to case 2)

Case 4: $E_4 \in A$; $E_1, E_2, E_3 \notin A$

ad a: $\{223\!\to, 322\!\to\} \subset A$, $220\!\to$, $022\!\to\, \notin A$

If $K_1 = \ldots 0BB0B0BB0\ldots$ then $\exists n$:

$K_n^* = \ldots i_1\, 22i_2\, 2i_3\, 22i_4 \ldots$ for some $i_1, \ldots, i_4 \in \{0, 3\}$.

Since $223\!\to$, $322\!\to\, \in A$, it holds $i_1 = i_2 = i_3 = i_4 = 0$, but $E_4 = 020\!\to\, \in A$ changes in the next step state 2. This is a contradiction to K_n^*.

ad b: $\{220\!\to, 022\!\to\} \subset A$, $223\!\to$, $322\!\to\, \notin A$

If $K_1 = \ldots 0B0BB0BB0\ldots$ then $\exists n$:

$K_n^* = \ldots i_0 2i_1 22i_2 22i_3 \ldots$ for some $i_0, \ldots, i_4 \in \{0, 3\}$.

Since $220\!\to$, $022\!\to\, \in A$, it holds $i_1 = i_2 = i_3 = 3$. Since $E_1 = 323\!\to$, $E_3 = 023\!\to$, $223\!\to\, \notin A$ later birds ($0B0BB0BB0$) far enough right from the first one reach state 2 and remain in this state.

Case 5: $E_3, E_4 \in A$; E_1, $E_2 \notin A$

If $K_1 = \ldots 0B0BB0B0\ldots$ then $\exists n$:

$K_n^* = \ldots i_1\, 2i_2\, 22i_3\, 2i_4\ldots$ for some $i_1, \ldots, i_4 \in \{0, 3\}$.

ad a: $\{220\!\to, 022\!\to\} \subset A$; $223\!\to$, $322\!\to\, \notin A$. Then it holds $i_2 = i_3 = 3$ and because of $023\!\to\, \in A$, $i_1 = 3$ holds. Since $E_1 = 323\!\to$, $E_2 = 320\!\to$, $322\!\to\, \notin A$ later birds far enough left from the first one reach state 2 and state 2 cannot be changed.

ad b: $\{223\!\to, 322\!\to\} \subset A$; $220\!\to$, $022\!\to\, \notin A$. Then it holds $i_2 = 0 = i_3$, but $E_3 = 023\!\to$, $E_4 = 020\!\to\, \in A$. Thus for one bird-cell (left from i_4) state 2 is changed in the next step.

Case 6: $E_2, E_4 \in A$; $E_1, E_3 \notin A$ (analogue to case 5)

Case 7: $E_2, E_3 \in A$; E_1, $E_4 \notin A$

ad a: $\{223\!\to, 322\!\to\} \subset A$; $220\!\to$, $022\!\to\, \notin A$

If $K_1 = \ldots 0B0BB00BB0B0B0BB0B0\ldots$ then $\exists n$:

$K_n = \ldots i_1\, 2i_2\, 22i_3\, i_4\, 22i_5\, 2i_6\, 2i_7\, 22i_8\, 2i_9 \ldots$ for some $i_1, \ldots, i_8 \in \{0, 3\}$.

Since $223\to$, $322\to\in A$ and $E_3=023\to$, $E_2=320\to\in A$ it holds $i_j=0$ $(1\leqq j\leqq 9)$ and $200\to$, $002\to$, $202\to\notin A$. Birds must send out signals to the right or to the left. So we can assume that a cell in state 3 is left or right from the cell with state i_1 or i_9 — say left —. Since $002\to$, $202\to\notin A$ it holds $302\to\in A$, otherwise later birds ($0B0BB$) far enough right from the first birds survive (in state 2). Now let $K_n^*=$ $=\dots 3\underbrace{0\dots 0}_{m}20220022020202022020$ be and $m=1$. This leads to a contradiction. If $302\to 3\in A$, then $E_3=320\to\in A$ eliminates state 2. If $302\to 2\in A$, then a new state 2 occurs. Now let $m>1$. Because $200\to\notin A$ and later birds (double the configuration K_1) far enough right from the first ones must be eliminated, the set A contains the transition $300\to$. This leads to a contradiction, because in case of $300\to 3$ we reach case $m=1$ and in case of $300\to 2$ a new distinguised state 2 occurs.

ad b: $\{220\to, 022\to\}\subset A$; $223\to$, $322\to\notin A$

If $K_1=\dots 0B0BB00BB0$ then $\exists n$:

$$K_n^*=\dots i_1\,2i_2\,22i_3\,i_4\,22i_5\dots \quad \text{for some} \quad i_1,\dots,i_5\in\{0,3\}.$$

Then it holds $i_1=i_2=i_3=i_4=i_5=3$, because $220\to$, $022\to$, $E_3=023\to$ and $E_2=$ $=320\to\in A$.

Furthermore we see that $232\to$, $332\to$, $233\to\notin A$ (otherwise a new state 2 arises or a state 2 is eliminated one or two steps later). Since $232\to$, $332\to\notin A$ and later birds ($0B0BB00BB0$) far enough right from the first birds must be killed, the transition $032\to$ *belongs to* A.

Now we consider $K_1'=\dots 0BB000BB0\dots$ then $\exists t$:

$$K_t'^* = \dots j_1\,22j_2\,j_3\,j_4\,22j_5 \quad \text{for some} \quad j_i\in\{0,3\} \quad (1\leqq i\leqq 5).$$

Then it holds $j_i=3$ $(i\neq 3)$ $(220\to$, $022\to\in A)$. If $j_3=0$ then $032\to\in A$ and $022\to\in A$ lead to a contradiction (new state 2 or elimination). Thus we see that $j_3=3$ and $333\to\notin A$.

Going back to K_1 it holds

$$K_n^* = \dots 0\underbrace{3\dots 3}_{m}\,232233223\dots$$

If $m=1$ and if $032\to 0\in A$, then $E_2=023\to\in A$ eliminates the distinguished state 2 and if $032\to 2\in A$ we reach a new state 2. Let $m>1$. It holds $033\to\notin A$, otherwise we obtain after some steps the situation $m=1$ or a new state 2. Altogether we get $333\to$, $033\to$, $233\to\notin A$ and therefore a later bird (state 2) ($0B0BB0$) far enough from the first birds survive.

Case 8: $E_1, E_4\in A$; $E_2, E_3\notin A$

If $K_1=\dots 0BB0B0BB0\dots$ then $\exists n$:

$$K_n^* = \dots i_1\,22i_2\,2i_3\,22i_4 \quad \text{for some} \quad i_1,\dots,i_4\in\{0,3\}.$$

ad a: $\{220\to, 022\to\}\subset A$; $223\to$, $322\to\notin A$, then it holds $i_1=i_2=i_3=i_4=3$, but $E_1=323\to\in A$ contradicts K_n^*.

ad b: $\{223\to, 322\to\}\subset A$; $220\to$, $022\to\notin A$, then it holds $i_1=i_2=i_3=i_4=0$, but $E_4=020\to\in A$ contradicts K_n^*.

Case 9: $E_1, E_2\in A$; $E_3, E_4\notin A$

If $K_1 = \ldots 0BB0B0BB0B0 \ldots$ then $\exists n$:

$K_n^* = \ldots i_1\, 22 i_2\, 2 i_3\, 22 i_4\, 2 i_5 \ldots$ for some $i_1, \ldots, i_5 \in \{0, 3\}$.

ad a: $\{220\to, 022\to\} \subset A$; $223\to, 322\to \notin A$, then it holds $i_1 = i_2 = i_3 = i_4 = 3$, but $E_1 = 323\to \in A$ contradicts K_n^*.

ad b: $\{223\to, 322\to\} \subset A$; $220\to, 022\to \notin A$, then it holds $i_1 = i_2 = i_3 = i_4 = 0$. Since $E_4 = 020\to$, $E_3 = 023\to$, $022\to \notin A$ later birds ($0BB0B0BB0B0$) far enough left from the first one reach state 2 and remain in this state.

Case 10: $E_1, E_3 \in A$; $E_2, E_4 \notin A$ (analogue to case 9)

Case 11: $E_1, E_2, E_4 \in A$; $E_3 \notin A$
If $K_1 = \ldots 0B0B0B0 \ldots$ then $\exists n$

$K_n^* = \ldots i_1\, 2 i_2\, 2 i_3\, 2 i_4 \ldots$ for some $i_1, \ldots, i_4 \in \{0, 3\}$.

If $i_2 = 3$, then $i_3 = 0$ or 3, but $E_1 = 323\to$, $E_2 = 320\to \in A$ contradicts K_n^*. If $i_2 = 0$, then $i_3 = 3$ and then $i_4 = 0$ or 3, but $E_1 = 323\to$, $E_2 = 320\to \in A$ contradicts K_n^*.

Case 12: $E_1, E_3, E_4 \in A$; $E_2 \notin A$ (analogue to case 11)

Case 13: $E_1, E_2, E_3 \in A$; $E_4 \notin A$
ad a: $\{220\to, 022\to\} \subset A$; $223\to, 322\to \notin A$
If $K_1 = \ldots 0BB0B0BB0 \ldots$ then $\exists n$:

$K_n^* = \ldots i_1\, 22 i_2\, 2 i_3\, 22 i_4$ for some $i_1, \ldots, i_4 \in \{0, 3\}$.

Since $220\to$, $022\to \in A$, it holds $i_1 = i_2 = i_3 = i_4 = 3$. But $E_1 = 323\to \in A$ contradicts K_n^* $(i_2\, 2 i_3)$.
ad b: $\{223\to, 322\to\} \subset A$; $220\to, 022\to \notin A$
If $K_1 = \ldots 0B0BB00BB0B0 \ldots$ then $\exists n$:

$K_n^* = \ldots i_1\, 2 i_2\, 22 i_3\, i_4\, 22 i_5\, 2 i_6 \ldots$ for some $i_1, \ldots, i_6 \in \{0, 3\}$.

Since $223\to$, $322\to \in A$, it holds $i_2 = i_3 = i_4 = i_5 \to 0$ and because of $E_3 = 023\to$, $E_2 = 320\to \in A$ it holds $i_1 = i_6 = 0$. Furthermore $i_1 = \ldots = i_6 = 0$ implies $202\to$, $200\to$, $002\to \notin A$, otherwise a state 2 is changed. Birds must send out signals to the right or to the left. Therefore we have a transition $300\to 3$ or $003\to 3$ in A.

Suppose: $300\to 3, 003\to 3 \in A$.

Case b1: Left from the cell with state i_1 in K_n^* state 3 occurs.

$K_n^* = \ldots 3\underbrace{0\ldots}_{m}020220022020\ldots$

Let $m=1$: $302\to 3$ or $2 \in A$ contradicts K_n^*, because of $E_2 = 320\to \in A$ resp. a new state 2 occurs. Thus $302\to \notin A$ and because $202\to$, $002\to \notin A$ a later bird marked by $\sim$ ($0\tilde{B}0BB0$) far enough right from the first remains in state 2.

If $m>1$, we reach after $m-1$ steps case $m=1$, because $300\to 3 \in A$.

Case b2: Right from the cell with state i_6 in K_n^* state 3 occurs. This leads to a contradiction similar to case b1.

Thus we see that $300\to 3\notin A$ or $003\to 3\notin A$. Without loss of generality we assume $300\to 3\in A$ and $003\to 3\notin A$. Then there is no cell left from the cell with state i_1 in K_n^* which has state 3 (apply case b_1 again). Therefore a cell right from the cell i_8 must have state 3.

$$\ldots 020220022020\underbrace{\ldots}_{m}03$$

If $m=1$ for all further steps, then $203\to 3\notin A$, because $E_3=023\to\ \in A$ and $203\to 2\notin A$, because a distinguished state 2 arises. Altogether we obtain $203\to$, $200\to$, $202\to\ \notin A$. This shows that later birds ($0BB0B0$) far enough left from the first reach state 2 and remain in this state.

If $m>1$, then $003\to 2\in A$, otherwise there is no feedback from a meeting with birds far enough right from the origin, but $003\to 2$ generates new distinguished states in case of no birth of birds. This is a contradiction.

Case 14: $E_2, E_3, E_4 \in A$; $E_1 \notin A$

ad a: $\{223\to,\ 322\to\}\subset A$; $220\to$, $022\to\ \notin A$

If $K_1=\ldots 0BB0B0BB0\ldots$ then $\exists n$:

$$K_n^* = i_1\,22 i_2\,2 i_3\,22 i_4 \ldots \quad \text{for some } i_1, \ldots, i_4 \in \{0, 3\}.$$

Because $223\to$, $322\to\ \in A$ it holds $i_1=i_2=i_3=i_4=0$, but $E_4=020\to\ \in A$ contradicts K_n^*.

ad b: $\{220\to,\ 022\to\}\subset A$; $223\to$, $322\to\ \notin A$

If $K_1=\ldots 0B0BB00BB0B0BB0B0\ldots$ then $\exists n$:

$$K_n = i_1\,2 i_2\,22 i_3\, i_4\,22 i_5\,2 i_6\,22 i_7\,2 i_8 \quad \text{for some } i_1, \ldots, i_8 \in \{0, 3\}.$$

Because $220\to$, $022\to\ \in A$ it holds $i_j=3$ $(1\leqq j\leqq 8)$. This implies $232\to$, $332\to$, $233\to\ \notin A$.

If $K_1'=\ldots 0BB000BB0\ldots$ then $\exists n$:

$$K_n^* = j_1\,22 j_2\, j_3\, j_4\,22 j_5 \ldots \quad \text{for some } j_1, \ldots, j_5 \in \{0, 3\}.$$

Because of $220\to$, $022\to\ \in A$ it holds $032\to\ \in A$, otherwise a later bird ($0B0BB0$) far enough right from the first bird (starting with K_1) reaches state 2 and remains in this state. This implies $j_3=3$, because $022\to\ \in A$ and $E_3=023\to\ \in A$. Furthermore it follows from $j_2=j_3=j_4=3$ that $333\to\ \notin A$. Now we consider K_1 and K_n^* again.

$$K_n^* = \ldots 0\underbrace{3\ldots 3}_{m}\,2 i_2\,22 i_3\, i_4\,22 i_5\,2 i_6\,22 i_7\,2 i_8 \ldots$$

$m=1$ implies a contradiction, because $023\to 0\in A$ (then $023\to\ \in A$ eliminates state 2) or $032\to 2\in A$ (then a new distinguished state arises). Let $m>1$. Since $333\to\ \notin A$ and $233\to\ \notin A$ and later birds ($0B0BB0$) far enough from the first birds must be killed (state 2 changed) the transition $033\to$ belongs to A.

This transition leads to a configuration K_t^* (from K_n^*)

$$\ldots 0\tilde{3}2322332232322323\ldots$$

and in the next step we obtain a new state 2 ($032\to 2$) or a state 0 ($032\to 0$) in the cell marked by $\sim$ and then we eliminate state 2, because of $E_3=023\to\ \in A$. This shows that case 14 is impossible.

Case 15: $E_1, E_2, E_3, E_4 \in A$

If $K_1 = \ldots 0B0\ldots$, then $\exists n$: $K_n^* = \ldots i_1 2 i_2 \ldots$ for some $i_1, i_2 \in \{0, 3\}$, but $E_i \in A$ $(1 \leqq i \leqq 4)$ contradicts K_n^*.

Case 16: $E_i \notin A$ $(1 \leqq i \leqq 4)$

If $K_1 = \ldots 0B0BB0B0\ldots$ then $\exists n$:

$$K_n^* = \ldots i_1 2 i_2 22 i_3 2 i_4 \ldots \text{ for some } i_1, \ldots, i_4 \in \{0, 3\}.$$

ad a: $\{220\rightarrow, 022\rightarrow\} \subset A$; $223\rightarrow$, $322\rightarrow \notin A$, then it holds $i_2 = i_3 = 3$. Since $E_2 = 320\rightarrow$, $E_1 = 323\rightarrow$, $322\rightarrow \notin A$ later birds $0B0BB0B0$ far enough left from the first birds reach state 2 and remain in this state.

ad b: $\{223\rightarrow, 322\rightarrow\} \subset A$; $220\rightarrow$, $022\rightarrow \notin A$, then it holds $i_2 = i_3 = 0$. Since $E_4 = 020\rightarrow$, $E_3 = 023\rightarrow$, $022\rightarrow \notin A$ later birds $(0B0BB0B0)$ far enough left from the first birds reach state 2 and remain in this state.

Altogether we have proved that the one-dimensional early bird problem is unsolvable with set of states $\{0, B, 2, 3\}$ and with distinguished state 2. □

INSTITUT FÜR MATHEMATISCHE LOGIK
UND GRUNDLAGENFORSCHUNG
D—4400 MÜNSTER, GERMANY

References

[1] LEGENDI, T., E. KATONA, A 5-state solution of the early bird problem in a one-dimensional cellular space, *Acta Cybernet.*, v. 5, 1981, pp. 173—179.

[2] ROSENSTIEHL, P., J. R. FIKSEL, A. HOLLINGER, Intelligent graphs: Network of finite automata capable of solving graph problems, ED. READ, R. C., *Graph Theory and Computing*, Academic Press, New York, 1972, pp. 219—265.

[2] VOLLMAR, R., On two modified problems of synchronization in cellular automata, *Acta Cybernet.*, v. 3, 1978, pp. 293—300.

(Received August 27, 1981)

On the real-time recognition of formal languages in cellular automata

By J. PECHT*

1. Previous approaches to use cellular automata to recognize formal languages: appreciation and critic

To recognize formal languages by cellular automata (ca) already some approaches have been developed. A ca is defined by a d-dimensional Euclidean space $\mathbf{Z}^d$ ($d \geqq 0$) ($\mathbf{Z}$ denoting the set of integers), where each lattice point is occupied by a finite deterministic automaton, and all automata are identical and work synchronously. Each automaton is connected with a fixed finite number ($\geqq 2$) of neighbours, where all automata use the same interconnection scheme T, called *template* or *neighbourhood.* The best known templates are the (d-dimensional) *von Neumann templates* $T = H^d = \{\mathbf{0}, \pm u_1, \pm u_2, \ldots, \pm u_d\}$, where $\mathbf{0} = (0, 0, \ldots, 0)$ is the (d-dimensional) origin and u_i is the i^{th} d-dimensional unit vector and the (d-dimensional) *Moore templates* $T = M^d = \{-1, 0, 1\}^d$. In each transition step the behaviour of the automaton at point x depends only on the states of the automata at points $x+t$, where t ranges over T. In that sense, we will consider homogeneously occupied, homogeneously interconnected deterministic, single transition function ca. For details see, e.g., [13], [1], [3] or [14].

Using ca to recognize formal languages [9], one has to decide how to input the words, or chains of symbols. As in other abstract recognition devices, there are two main possibilities to do this. First, we have the "on-line" ca as defined by Cole [2]. In this case, the automaton at the origin is equipped with an additional input line from which it reads the input word, one letter at each time step. Let us take the state of the (distinguished) automaton at origin, immediately after having received the last symbol of the input word, in order to decide whether or not the word belongs to the language considered. The class of languages which can be recognized by such a d-dimensional ca is called "the class of the d-dimensional 'on-line' real-time recog-

* Now with: Kernforschungszentrum Karlsruhe, Institut für Material- und Festkörperforschung III, Postfach 3640, D—75, Karlsruhe, FRG.

nizable languages" and is denoted by $\mathscr{L}_d^{\text{on}}$ ($d \geqq 0$). In [2], the following is obtained:

$$\mathscr{L}_0^{\text{on}} = \mathscr{L}_3,$$

$$\forall d \geqq 0: \mathscr{L}_d^{\text{on}} \subsetneqq \mathscr{L}_{d+1}^{\text{on}},$$

$$\exists L \in \mathscr{L}_2: \forall d \geqq 0: L \notin \mathscr{L}_d^{\text{on}},$$

where $\mathscr{L}_3$ denotes the class of regular languages and $\mathscr{L}_2$ the class of context-free languages.

As in Cole's approach the word to be analyzed has to be input letter by letter, it is in general impossible to get a (maximal) recognition time less than $1 \cdot n$, where n is the length of the actual input word. Moreover, as the information disperses only with finite speed from the origin into the space (in a pyramidal manner), most of the automata are activated "too late" and only few of the capabilities of parallelism are exploited. Therefore this approach of real-time recognition causes an exploitation factor of (approximately) only $1/d!$

These disadvantages can be removed, if the word to be analyzed is not read sequentially in n steps, but in a parallel manner, using only one step. In other words: The information is supposed to be written into the ca (i.e.: distributed over the single automata) at the beginning of the recognition process. The way to embed the words must be simple and as independent as possible from the actual word (in some sense). Moreover, no two symbols of the same word are allowed to occupy the same automaton. Smith [10] considers this procedure for the one-dimensional case. He presupposes that the input word is inscribed from left to right, beginning at the origin 0 and with no gaps allowed. Automata not occupied by the input are assumed to remain in a "boundary state" which does not alter during the whole evaluation process. After n steps (n as above) the state of the automaton at origin gives the decision whether the word belongs to the language or not.

If one considers only the von Neumann template, H^1, the languages recognizable in such a manner are called "one-dimensional 'off-line' real-time recognizable" and their class is denoted by $\mathscr{L}_1^{\text{off}}$. Smith proved that

$$\mathscr{L}_1^{\text{on}} \subsetneqq \mathscr{L}_1^{\text{off}}$$

and

$$\exists L \in \mathscr{L}_1^{\text{off}}: \forall d \geqq 0: L \notin \mathscr{L}_d^{\text{on}},$$

concluding that "off-line" ce are inherently faster than "on-line" ones. He explains this phenomenon with the higher degree of parallelism now available from the beginning of the analyzing process. In this approach, however, remains the fact that recognition times less than real-time are generally not achievable, too, because the most distant symbol cannot influence the cell at origin before the n^{th} step.

Generalizing the results of Smith, Seiferas [12] achieves recognition times of the form $d\left\lceil \sqrt[d]{n} \right\rceil$ in d-dimensional off-line ca. To do this, the word to be analyzed is inscribed into the cube $\left\{0, 1, \ldots, \left\lceil \sqrt[d]{n} \right\rceil - 1\right\}^d$ row by row and then surrounded by a special boundary symbol (sc. Fig. 1). Seiferas uses the templates $N_+^d := \{0, u_1, 2u_1, u_2, 2u_2, \ldots, u_d, 2u_d\}$. He proves that all regular languages can be recognized in such a way within the cited time. But it is easy to verify that, using this type of inscription

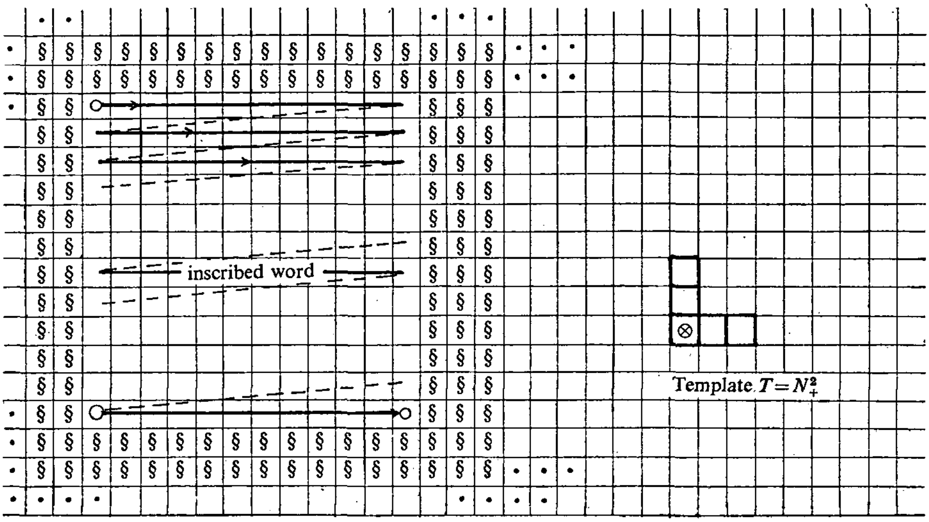


Fig. 1

Off-line recognition of regular languages in 2-dimensional cellular automata according to SEIFERAS [12]

and this template N_+^d, all symbols of the inscribed word can influence the origin already within time $\left\lceil \frac{d}{2} \left\lceil \sqrt[d]{n} \right\rceil \right\rceil$. This implies that Seiferas does not meet the lowest possible recognition time which perhaps could be reached in these structures. The aim of the following is to investigate and generalize this aspect in a more detailed way.

2. Introduction to a systematic approach: *T*-recognition of *T*-languages

If we consider, for example, the template N_+^d, then after the k^{th} step, the automaton at origin can be influenced by (approximately) $(2k)^d/d!$ other automata. This means that Seiferas uses only

$$\frac{(\lceil \sqrt[d]{n} \rceil)^d}{(2d\lceil \sqrt[d]{n} \rceil)^d/d!} \quad \left(= \frac{d!}{2^d d^d}\right)$$

of the supplied space. Similarly, we state that n points can be "reached" from the origin within

$$\left\lceil \frac{1}{2} \lceil \sqrt[d]{d!} \rceil \lceil \sqrt[d]{n} \rceil \right\rceil$$

steps. This fact implies that a speedup factor of (approximately)

$$\frac{2d}{\sqrt[d]{d!}} \quad \left(\geqq \frac{4d}{d+1}\right)$$

can possibly be achieved, without changing the template N_+^d, if we replace the cubic representation of the word by one which is more adapted to the shape of the region eontaining all the points reachable by the oirigin within k steps. In this case it turns out to be a simplex. But let us consider these problems in an even more general way:

Given any template T, the region which can influence the origin within k steps is the set kT recursively defined by

$$0T := \{0\}$$

and

$$(k+1)T := kT+T \quad (k \geqq 0)\ ^1.$$

This means that, after the k^{th} step, the state at the automaton at origin $\mathbf{0}$ can be used to decide some property of that part of the input pattern which is contained in region kT. Vice versa, we can (ab-) use any ca with template T to classify patterns of the shape kT $(k \geqq 0)$. This is done in the following way: let us assume that the patterns to be classified contain only symbols of some subset A of the state set Z of the ca. Then, given any such pattern with shape kT, extend it to an (infinite) pattern in an arbitrary way and make work the ca exactly k steps. Afterwards the state of the automaton at origin is taken to classify the original finite pattern.

To formalize these ideas, let us call any finite, nonempty set A an *alphabet* and any mapping $w: kT \to A$ a *$T-A$-word* (*T-word* or, simply, *word*) *with shape kT* or *with T-diameter k* $(k \geqq 0)$. Then, formally, A^{kT} denotes the set of all such words with shape kT [2]. Furthermore, let $(T, A)^*$, defined by

$$(T, A)^* := \bigcup_{k \geqq 0} A^{kT}$$

denote the *set of all $T-A$-words (of any T-diameter)*. It is true that, depending on the underlying template T, the words of $(T, A)^*$ may have somewhat strange shapes (sc. Fig. 2). Any subset L of $(T, A)^*$ is called a *$T-A$-language* (or, simply, *T-language*). We say that a certain ca (with the same template T) *T-recognizes L* if its state set, Z, contains A and if there is a subset F of Z, the *set of accepting states*, such that for any word w of $(T, A)^*$ with shape kT it holds: w is an element of L iff, extending w as cited above, and starting running the ca exactly k steps[3], the automaton at origin enters a state of F. L is called *T-recognizable*, if there exists a ca (with template T) which T-recognizes L. Obviously, this notion of recognizability is the strongest real-time recognizability definable in off-line ca, because a pattern must be classified as soon as the whole information to be classified can have influenced the deciding cell.

Furthermore, if we want to apply this approach to the recognition of formal languages, we have to define how to represent the (conventional) words (e.g. of A^*) as $T-A$-words. Therefore we introduce the following notation: Any sequence $h=(h^k)_{k \geqq 0}$, where each member h^k represents some bijection[4]

$$h^k\colon kT \to \{1, 2, 3, \ldots, \text{card}\,(kT)\},$$

[1] For two subsets M and N of $\mathbf{Z}^d$ and any element $x \in \mathbf{Z}^d$ let $M+N := \{y+z / y \in M$ and $z \in N\}$ and $x+M (:= M+x) := \{x\}+M$. $+(-)$ is the componentwise sum (difference).

[2] For two sets M and N let M^N denote the set of all mappings f from N into M (i.e.: $M^N := \{f / f\colon N \to M\}$).

[3] Note that $k \neq k' \Rightarrow kT \neq k'T$ $(k, k' = 0, 1, 2, \ldots)$ (cp. [6, 7]).

[4] A mapping is said to be a bijection if it is onto and one-to-one.

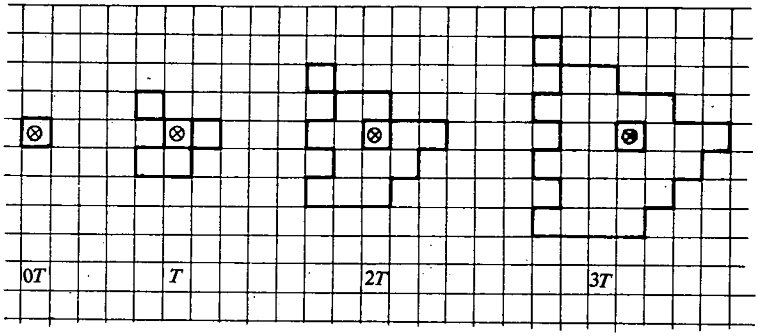


Fig.2

The shapes $0T=\{0\}$, $1T=T$, $2T$ and $3T$ for some 2-dimensional template $T(=\{(-1,-1),(0,-1),(0,0),(1,0),(-1,1)\})$

is called a *T-wrap*. Such a T-wrap $h=(h^k)_{k\geq 0}$ permits us to represent any word $p=a_1a_2a_3\ldots a_{\mathrm{card}(kT)}$ (i.e. of the length card (kT)) as the $T-A$-word w $(\in A^{kT}!)$, defined by

$$w(x) := a_{(h^k(x))}, \quad (x \in kT)\ ^5.$$

Let us denote this (uniquely defined) word w as $\hat{h}(p)$. Thus, $\hat{h}$ can be considered as a partial mapping from A^* to $(T, A)^*$, which only maps words of the length 1, card (T), card $(2T)$, card $(3T)$, ... (This is no real striction because any nonfitting word can be filled up to the next fitting length.) For any formal language $S(\subseteq A)^*$ let $\hat{h}(S)$ be the $T-A$-language, defined by

$$\hat{h}(S) := \{\hat{h}(p)/p \in S \text{ and } \hat{h}(p) \text{ is defined}\}.$$

Now, with these notions, the "real-time recognition of formal languages by off-line ca" reduces to the problem:

Let S be a formal language, T a template and h a T-wrap. Is the $T-A$-language $\hat{h}(S)$ T-recognizable or not?

In this paper we give a partial answer to this question concerning the T-recognizability of regular and context-free formal languages.

First we restrict our considerations to the family of Moore templates, M^d and their capabilities to recognize regular languages. Smith [11] has shown that, in case $d=1$, for any regular language, R, there is a ca using template M^1 which M^1-recognizes $\hat{h}(R)$, where h is the straightforward inscription from left to right. Theorem 10 states that this inscription technique can not be generalized for $d>1$. There it is shown that no T-wrap which divides the admitted inscription areas (i.e.: kM^d into parallel rows and fills these individual rows strictly from left to right or right to left each can generally be used to recognize regular languages. In Theorem 11, however, it is shown that, for any d-dimensional template T, there exists a nontrivial T-wrap which makes possible the T-recognition of any regular language.

In case of context-free languages we obtained only negative answers which are the sharper the more extreme points the template contains. Ruling out the trivial

[5] Note that $k\neq k' \Rightarrow \mathrm{card}\,(kT)\neq \mathrm{card}\,(k'T)$ $(k, k'=0, 1, 2, 3, \ldots)$ (cp. [6, 7]).

case where T contains exactly one extreme point (and, consequently, kT is a singleton) we proved the fcllowing:

If T contains exactly 2 extreme points (which implies that each kT is a (possible sparsed) line), then the simple T-wrap along this line does not fit for all context-free languages (Theorem 12). It is, however, an open problem whether some other T-wraps will do it.

If T contains exactly 3 extreme points (and therefore any kT with $k \geqq 1$ contains also exactly 3 extreme points), then kT can be considered to consist of a series of lines which are parallel to one of the 3 extreme egdes of kT. In Theorem 13 we prove that there is no inscription rule which fills first the starting extreme egde and then the remaining lines in a strictly removing manner and which fits for all context-free languages. This is true even if it would be allowed to vary the internal order within each particular row arbitraryly and depending on the particular language.

Finally, if T contains 4 extreme points or more, then there is a context-free language, C, for which there is no T-wrap h at all such that $\hat{h}(C)$ is T-recognizable (Theorem 14).

3. The proofs of the results

First, we give some appropriate notations and some basic statements concerning them which have been developed previously ([6], [7]). Essentially, the notions mentioned in section 2 are treated in a more systematical way and some of them will be redefined or generalized.

Let $\mathbf{N}=\{0, 1, 2, \ldots\}$ denote the set of natural numbers. Let d $(d \geqq 1)$ be a dimension, M a finite subset of $\mathbf{Z}^d$ and A an alphabet. Then any function $w: M \to A$ is called a *(d-dimensional) word (over the alphabet A)* and M is called the *support* or *domain* of w, denoted as dom (w). The set of all d-dimensional words over alphabet A is denoted as $(d, A)^*$ and equals

$$\bigcup_{M \subseteq \mathbf{Z}^d:\ \mathrm{card}\,(M) < \infty} A^M.$$

d-dimensional words may be *displaced* and *restricted:* for any word $w \in (d, A)^*$ and any vector $x \in \mathbf{Z}^d$ let the word $w \oplus x$, the *x-displacement* of w, be defined by

$$\mathrm{dom}\,(w \oplus x) := \mathrm{dom}\,(w) + x$$

and

$$(w \oplus x) \quad (y) := w(y-x) \quad (y \in \mathrm{dom}\,(w \oplus x)).$$

Instead of $w \oplus (-x)$ we write $w \ominus x$, too. (Clearly, $w \oplus (x+y) = (w \oplus x) \oplus y$.) For any word w with $\mathrm{dom}\,(w) = M$ and any subset N of M let $w|_N$ be the wellknown *restriction* of w to N. Note that $\mathrm{dom}\,(w|_N) = N$.

Given any template, T $(2 \leqq \mathrm{card}\,(T) < \infty)$, then we get, for $k, k' \in \mathbf{N}$ with $k \neq k'$ $kT \neq k'T$. This is proved in [7] using the strictly increasing (Euclidean) diameter of the sets kT. Thus, $(T, A)^*$, as defined in section 2, is the disjoint union of its constituting subsets A^{kT} $(k \in \mathbf{N})$.

Therefore, for any $w \in (T, A)^*$, the natural number k with $w \in A^{kT}$ is uniquely determined and is denoted by $D(w)$ or $D_T(w)$ and named: the *(T-)diameter* of w. Note that, because of

$$(k+m)T = \bigcup_{x \in kT} x + mT \quad (k, m \in \mathbf{N});$$

for any word $w \in (T, A)^*$, any i with $0 \leqq i \leqq D(w)$ and any $x \in iT$, the word $(w \ominus x)|_{(D(w)-i)T} (= w|_{x+(D(w)-i)T} \ominus x)$ is also a word in $(T, A)^*$ (with diameter $D(w)-i$). Furthermore, let $(T, A)^+$ be defined as the set $\{w / w \in (T, A)^* \text{ and } D_T(w) \geqq 1\}$.

Now let us formalize the notion of T-recognition. A T-recognizing cellular automaton (Trca) **A** is a quintuple $\mathbf{A} = (T, A, Z, f, F)$ where T is a template, Z is an alphabet, called the *state alphabet*, $A (\subseteq Z)$ is another alphabet, called the *input alphabet*, f is some function $f\colon Z^T \to Z$ and called the *local transition function* and $F (\subseteq Z)$ is called the *set of accepting states* or, shortly, *accepting set*. Now, identifying the set $Z^{\{0\}}$ with state alphabet, Z, we may extend domain and range of f, widening f to be a function $f\colon (T, Z)^+ \to (T, Z)^*$ such that, for any $w \in (T, Z)^+$

$$\operatorname{dom}(f(w)) := (D(w)-1)T$$

with

$$f(w)(x) := f((w \ominus x)|_T) \quad (x \in (D(w)-1)T).$$

(This is possible because of $(k+1)T = \bigcup_{x \in kT} x + T \; (k \in \mathbf{N})$.) Then, clearly, the function $f^*\colon (T, A)^* \to Z^{\{0\}} (= Z)$, defined by

$$f^*(w) := f^{D(w)}(w) \quad (w \in (T, A)^*),$$

is well defined. Now, given the Trca $\mathbf{A} = (T, A, Z, f, F)$ and the $T-A$-language $L (\subseteq (T, A)^*)$, we say that **A** T-recognizes L iff

$$\forall w \in (T, A)^*\colon w \in L \Leftrightarrow f^*(w) \in F.$$

The $T-A$-language $L (\subseteq (T, A)^*)$ is said to be T-recognizable iff there exist Z, f, G such that the Trca $\mathbf{A} = (T, A, Z, f, G)$ T-recognizes L.

Now we give some basic notions and results concerning T-recognizability. Because of their importance within this section, we will cite them as explicit definitions and theorems. They are presented here as in [7].

Definition 1. Let A and Z be two (arbitrary) alphabets, T a template and g some function $g\colon (T, A)^* \to Z$. Then let the function $\bar{\bar{g}}\colon (T, A)^+ \to Z^T$ be defined by

$$\bar{\bar{g}}(w)(x) := g((w \ominus x)|_{(D(w)-1)T}) \quad (x \in T,\ w \in (T, A)^+).$$

Using this notion we get

Theorem 2. The T-language $L (\subseteq (T, A)^*)$ is T-recognizable iff there is an alphabet Z, a function $g\colon (T, A)^* \to Z$, a function $f\colon Z^T \to Z$ and a subset F of Z such that

$$\forall w \in (T, A)^+\colon f(\bar{\bar{g}}(w)) = g(w)$$

and

$$\forall w \in (T, A)^*\colon (w \in L \Leftrightarrow g(w) \in F)$$

hold.

Now we introduce a new notion of equivalence relation.

Definition 3. Let A be an alphabet, T a template and L a $T-A$-language. Then, for any $k \in \mathbf{N}$, any two words, $w, w' \in (T, A)^*$ with $D(w) \geqq k$ and $D(w') \geqq k$ are said to be $k-L$-*equivalent* iff

$$\forall i (0 \leqq i \leqq k): \forall x \in iT: \big((w \ominus x)|_{(D(w)-i)T} \in L \Leftrightarrow (w' \ominus x)|_{(D(w')-i)T} \in L\big).$$

Let E_{k-L} denote the number of equivalence classes the set $\{w/w \in (T, A)^*$ and $D(w) \geqq k\}$ is divided into by this relation.

With respect to E_{k-L}, two sequences of numbers turn out to be important:

Definition 4. Let T be a template; then the sequences $(d_{k,T})_{k \in \mathbf{N}}$ and $(e_{k,T})_{k \in \mathbf{N}}$ are defined by

$$d_{k,T} := \operatorname{card}(kT) \quad (k \in \mathbf{N})$$

and

$$e_{k,T} := \sum_{i=0}^{k} d_{i,T} \quad \left(= \sum_{i=0}^{k} \operatorname{card}(iT)\right) \quad (k \in \mathbf{N}).$$

Theorem 5. Let A be an alphabet, T a template and L a $T-A$-language. Then, generally, it holds

$$E_{k-L} \leqq 2^{(e_{k,T})} \quad (k \in \mathbf{N}).$$

If L is T-recognizable, then it holds

$$E_{k-L} \leqq C^{(d_{k,T})} \quad (k \in \mathbf{N})$$

for some appropriate positive constant C.

The following theorem serves as a widely applicable general information compression argument which is proved in full details in [7]. Essentially, it states the following:

Let M and N be two disjoint subsets of kT and i an integer with $0 \leqq i \leqq k$ such that $M-iT$ and $N-iT$ are disjoint on $(k-i)T$. Moreover, let $(w_{m,n})$ be some family of words of A^{kT} where m and n range over some index sets $\underset{\sim}{M}$ and $\underset{\sim}{N}$ respectively such that all $w_{m,n}$ are identical outside $M \cup N$, words with the same index m are identical on M and words with the same index n identical on N. Let L be a $T-A$-language such that, for any pair (n, n') with $n \neq n'$, there exists an m such that $w_{m,n}$ and $w_{m,n'}$ are separated by L, then any Trca which T-recognizes L contains at least C states where

$$C^{\operatorname{card}((k-i)T \cap (N-iT))} \geqq \operatorname{card}(\underset{\sim}{N}).$$

This is true because, after starting the Trca with a word $w_{m,n}$ as input and running it exactly i-times, the information about the index n must be preserved in the field $(k-i)T \cap (N-iT)$. For the area outside $N-iT$ can not be influenced from input information on N, the area inside $(N-iT) \cap (k-i)T$ can not be influenced from information on M (about m) and information outside $(k-i)T$ can not influence the deciding cell at origin within the remaining $k-i$ steps.

The theorem as presented below is a more applicable reformulation of this elementary fact, using the T-diameter k as running index and i as an additional free parameter which, in typical applications, is chosen as an appropriate function on k.

Theorem 6. If the $T-A$-language L is T-recognizable, then there is a (positive) constant C such that the following assertion holds:

Let $k\in\mathbf{N}$, M_k and N_k two sets with $N_k, M_k\subseteq kT$ and $M_k\cap N_k=\emptyset$, $\underset{\sim}{M}_k$ and $\underset{\sim}{N}_k$ two non empty finite sets of indices and $(w^k_{m,n})_{n\in\underset{\sim}{N}_k, m\in\underset{\sim}{M}_k}$ a family of words such that

$$\forall m\in\underset{\sim}{M}_k\colon \forall n\in\underset{\sim}{N}_k\colon w^k_{m,n}\in A^{kT}, \tag{1}$$

$$\forall m\in\underset{\sim}{M}_k\colon \forall n, n'\in\underset{\sim}{N}_k\colon w^k_{m,n|M_k} = w^k_{m,n'|M_k} \tag{2}$$

$$\forall n\in\underset{\sim}{N}_k\colon \forall m, m'\in\underset{\sim}{M}_k\colon w^k_{m,n|N_k} = w^k_{m',n|N_k} \tag{3}$$

$$\forall n, n'\in\underset{\sim}{N}_k\colon \forall m, m'\in\underset{\sim}{M}_k\colon w_{m,n|kT\setminus(M_k\cup N_k)} = w_{m',n'|kT\setminus(M_k\cup N_k)} \tag{4}$$

$$\forall n, n'\in\underset{\sim}{N}_k(n\neq n')\colon \exists m\in\underset{\sim}{M}_k\colon w^k_{m,n}\in L \quad\text{and}\quad w^k_{m,n'}\notin L \quad\text{or}$$
$$w^k_{m,n}\notin L \quad\text{and}\quad w^k_{m,n'}\in L. \tag{5}$$

Then we have, for any i $(0\leqq i\leqq k)$ for which additionally holds

$$(M_k-iT)\cap(N_k-iT)\cap(k-i)T=\emptyset, \tag{6}$$

the necessary inequality

$$\operatorname{card}(\underset{\sim}{N}_k)\leqq C^{\operatorname{card}((N_k-iT)\cap(k-i)T)}.$$

Because the topic of this paper is the treatment of T-languages which are the T-wraps of some (conventional) string languages, we have to provide for some tools to construct T-wraps or to compose complicated ones from simpler ones. To compose them it serves

Notation 7. Let G be any set with the (partially defined) associative binary operation $\square$ and identity element λ, let I be any finite set of indices with $\operatorname{card}(I)=n$ and let $\lessdot$ be a total linear ordering of I. Then, for any family $(g_i)_{i\in T}$ of elements of G, we define the abbreviation

$$\underset{i\in I}{\overset{\lessdot}{\square}} g_i := \lambda\square g_{i_1}\square g_{i_2}\square\dots\square g_{i_n}$$

where $I=\{i_j|1\leqq j\leqq n\}$ and $i_1\lessdot i_2\lessdot i_3\dots\lessdot i_n$.

Three such associative operations play some role, two of which are wellknown from automata theory and another one which is introduced in [6].

Definition 8. For any alphabet A let $\circ$ denote the usual *concatenation* of words of A^*; the *empty word* ε serves as identity element. For any finite automaton (with input alphabet A) let σ be the set of all its transition functions σ_p where p ranges over A^*. Let $\hat{\circ}$ denote their product with $\sigma_p\hat{\circ}\sigma_{p'}=\sigma_{p\circ p'}$. For any state s of the finite automaton let $s\cdot\sigma_p$ denote the state assumed after p is input into the automaton starting with state s. Let $d\geqq 1$, M and N be two disjoint finite subsets of $\mathbf{Z}^d$ and let us assume that $h'\colon M\to\{1, 2, \dots, \operatorname{card}(M)\}$ and $h''\colon N\to\{1, 2, \dots, \operatorname{card}(N)\}$ both are bijections, then we denote by $h'\triangle h''$ the bijection $h\colon M\cup N\to\{1, 2, \dots, \operatorname{card}(M)+\operatorname{card}(N)\}$ defined by

$$h(x) := \begin{cases} h'(x) & \text{if}\quad x\in M \\ \operatorname{card}(M)+h''(x) & \text{if}\quad x\in N. \end{cases}$$

Note that $h\colon\emptyset\to\emptyset$ serves as identity element and that $\triangle$ is not commutative.

The following theorem ([6], [8]) will deliver our T-wraps:

Theorem 9. Let T be a template. Then there is an alphabet Q, a function $t\colon Q\to 2^{T\times Q}$[6] and a family $(M_q^k)_{k\in\mathbf{N},q\in Q}$ of sets with

$$M_q^k \subseteqq kT \quad (k\in\mathbf{N},\ q\in Q), \tag{7}$$

$$kT = \dot{\bigcup_{q\in Q}} M_q^k \quad (k\in\mathbf{N}) \tag{8}$$

and

$$M_q^{k+1} = \dot{\bigcup_{(x,r)\in t(q)}} x+M_r^k \quad (k\in\mathbf{N},\ q\in Q) \tag{9}$$

where $\dot{\bigcup}$ means that all participating sets are disjoint.

Using these notions and results we turn to prove our claims of section 2.

Theorem 10. Let $T=M^d$. (Then kT represents a cube with side length $2k+1$ and the origin as centre.) Let $h_d=(h_d^k)_{k\geqq 0}$ be some T-wrap such that h_d^k fills the cube kT row by row, where all rows are parallel to each other and the order within each row is strictly from left to right or right to left, but the order of rows may be chosen arbitrarily. Then there exists a regular language R_1 which can be chosen independently of d, such that, for any $d\geqq 2$, $\hat{h}_d(R_1)$ is not M^d-recognizable.

Proof. Let $A:=\{a,b,c\}$ and consider the regular language $R_1=(ab^+a)^*\cdot\cdot(cb^+c)^*(ab^+a)^*$. Now let $d\geqq 2$ be any dimension and T the d-dimensional Moore template, M^d. Furthermore, let $h=(h^k)_{k\in\mathbf{N}}$ be any T-wrap where h^k maps any two row neighbours onto two successive natural numbers. We claim that the $T-A$-language $L:=\hat{h}(R_1)$ is not T-recognizable. To show this, we assume without loss of generality that any row of kT consists of the points $(n_1,n_2,\ldots,n_{d-1},j)$ where $-k\leqq\leqq j\leqq k$. Thus any row is entirely characterized by some row address $n=(n_1,n_2,\ldots\ldots,n_{d-1})$ $(\in kM^{d-1})$. Let us call this row the *n-row*.

Now let us assume that L is T-recognizable. Then we may apply Theorem 6. Let C be chosen such that the assertion cited in that theorem holds. Now take any $k\geqq 1$ and consider, for any two row-addresses n and m $(\in kM^{d-1})$, the word $w_{m,n}^k(\in A^{kT})$ defined by

$$w_{m,n}^k(x_1,x_2,\ldots,x_d) := \begin{cases} b & \text{iff} \quad -(k-1)\leqq x_d\leqq (k-1),\\ c & \begin{cases}\text{if} & (x_1,x_2,\ldots,x_{d-1})=m\wedge x_d=-k\\ \text{or} & (x_1,x_2,\ldots,x_{d-1})=n\wedge x_d=k,\end{cases}\\ a & \text{else.}\end{cases}$$

In short: the two sides of $w_{m,n}^k$ which mark row ends (or row beginnings) are entirely filled with a's except the leftmost element of the m-row and the rightmost element of the n-row which, in turn, exhibit two c's. The residue of $w_{m,n}^k$, i.e., the entire space between these two sides, is filled with b's. (For the case $d=3$, this is visualized in Fig. 3.)

Now, as one verifies easily, independently upon whether the rows are T-wrapped from left to right or from right to left and whether the wrapping direction alternates

[6] For any set M let 2^M denote the set of all subsets of M.

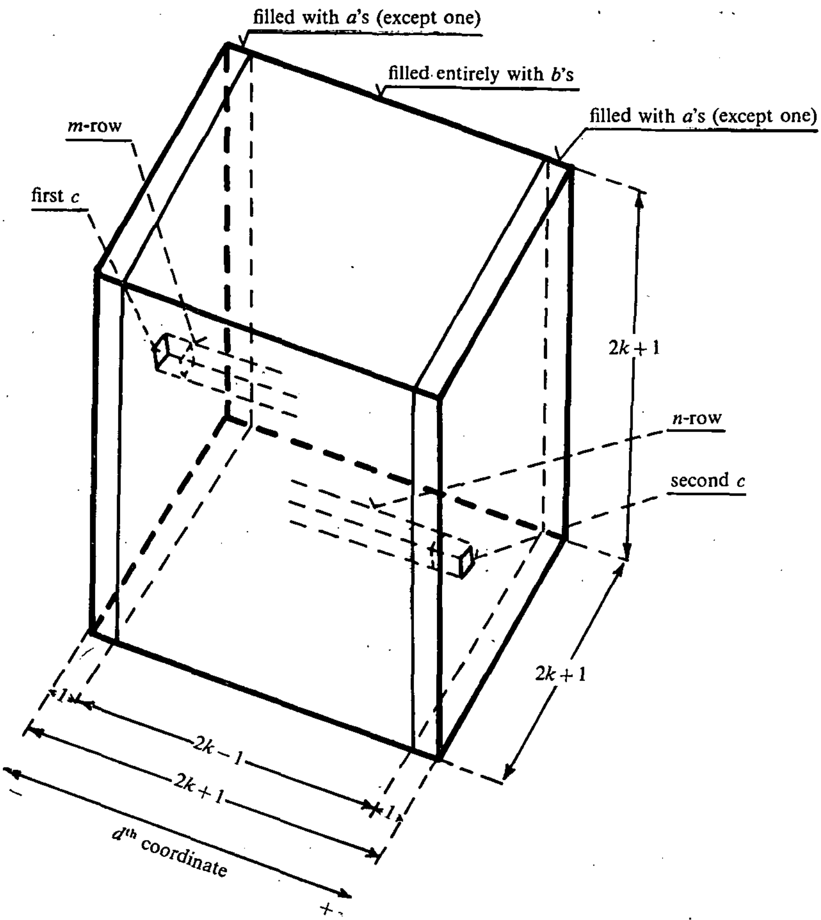


Fig.3
The words $w^k_{m,n}$ used in the proof of Theorem 10 (case $d=3$)

between some rows or not and independently upon which row is wrapped first, which second and so on, the following holds: $w^k_{m,n}$ is a word in L iff $n=m$, i.e.: iff the two exhibited c's are set vis-à-vis. Now, to continue applying Theorem 6, we set $M_k:=(kM^{d-1})\times\{-k\}$, $N_k:=(kM^{d-1})\times\{k\}$ (i.e.: the left and right side of kT resp.) and $\tilde{N}_k:=\tilde{M}_k:=kM^{d-1}$ (i.e.: $\tilde{N}_k$ and $\tilde{M}_k$ represent all possible row addresses in kT). We verify (1)—(5) step by step: Clearly, $w^k_{m,n}$ is an element of A^{kT} (1). $w^k_{m,n}$ and $w^k_{m,n'}$ differ not on M_k and $w^k_{m,n}$ and $w^k_{m',n}$ differ not upon N_k for arbitrary row addresses m, n, m' and n' ((2), (3)). On $kT\setminus(M_k\cup N_k)$, all admitted words $w^k_{m,n}$ exhibit only b's (4). For two distinct row addresses n and n', set $m=n$; then, clearly, $w^k_{m,n}$ is a member of L whereas $w^k_{m,n'}$ is not (5).

Now, let $i:=k-1$. Then M_k-iT contains only points whose d^{th} coordinates are less than 0 and N_k-iT contains only points whose d^{th} coordinates are greater than 0. Thus $(M_k-iT)\cap(N_k-iT)=\emptyset$ which, clearly, implies $(M_k-iT)\cap(N_k-iT)\cap\cap(k-i)T=\emptyset$. Thus (6) is verified, too, and according to Theorem 6, we get (with $k-i=1$)

$$(2k+1)^{d-1} = \text{card}(kM^{d-1}) = \text{card}\,(N_k) \leqq C^{\text{card}((N_k-iT)\cap T)} \leqq C^{\text{card}(T)} = C^{(d^3)}$$

independently of k ($k\geqq 1$). But for $d\geqq 2$ this is a contradiction. Thus, for such d's, $L=\hat{h}(R_1)$ is not T-recognizable, which proves Theorem 10. □

Theorem 11. For any dimension d and any template $T(\subseteq\mathbf{Z}^d)$ there is some T-wrap, h_T, such that, for any regular language R, $\hat{h}_T(R)$ is T-recognizable.

Proof. Let A be an alphabet, R a regular language ($\subseteq A^*$) and $T(\subseteq\mathbf{Z}^d)$ a template. We will derive a T-wrap which does not depend on R such that $\hat{h}(R)$ is T-recognizable. To do this, we rely on Theorem 9 according to which there exists an alphabet Q, a function $t\colon Q\to 2^{T\times Q}$ and a family $(M_q^k)_{k\in\mathbf{N},q\in Q}$ of sets with (7), (8) and (9). Because all unions encountered in Theorem 9 are disjoint, we can define a T-wrap $h=(h^k)_{k\in\mathbf{N}}$ in the following recursive way:

First some notational simplification: for any finite subset U of $\mathbf{Z}^d$, a bijection $u\colon U\to\{1,2,\ldots,\text{card}\,(U)\}$ and vector $x\in\mathbf{Z}^d$ let $(u\oplus x)\colon (U+x)\to\{1,2,\ldots,\text{card}\,(U+x)\}$ $(=\{1,2,\ldots,\text{card}\,(U)\})$ be defined by

$$(u\oplus x)\quad (y) := u(y-x)\quad (y\in U+x).$$

Now, let $h_q^k\colon M_q^k\to\{1,2,\ldots,\text{card}\,(M_q^k)\}$ be recursively defined by

$$h_q^0\colon \emptyset\to\emptyset\ \text{ if }\ M_q^0=\emptyset\quad (q\in Q)$$

$$h_q^0(0)=1\quad\text{if}\quad M_q^0=\{0\}\quad (q\in Q)$$

(note that $\{0\}=0T=\dot{\bigcup\limits_{q\in Q}}M_q^0$) and, for $k\geqq 0$,

$$h_q^{k+1} := \underset{(x,r)\in t(q)}{\widehat{\lessdot}}\ h_r^k\oplus x\quad (k\in\mathbf{N},\ q\in Q)$$

where $\lessdot$ is any (fixed) total ordering of $T\times Q$ (which does not vary with k). Now, let $h^k\colon kT\to\{1,2,\ldots,d_{k,T}\}$ be defined by

$$h^k := \underset{q\in Q}{\widehat{<}}\ h_q^k\quad (k\in\mathbf{N})$$

where $<$ is some fixed total ordering of Q (not depending on k). Clearly, $h=(h^k)_{k\in\mathbf{N}}$ is a T-wrap.

For two disjoint finite subsets M and N of $\mathbf{Z}^d$, two bijections $m\colon M\to\{1,2,\ldots,\text{card}\,(M)\}$, $n\colon N\to\{1,2,\ldots,\text{card}\,(N)\}$ and two words $w\colon M\to A$, $v\colon N\to A$, let $\bar{m}(w)$ be the word $p=a_1a_2\ldots a_l$ ($\in A^*$) with $l=\text{card}\,(M)$ and $a_j:=w(m^{-1}(j))$ ($1\leqq j\leqq l$) and $\bar{n}(v)$ defined similarly. Then we have for any word $u\colon M\cup N\to A$

$$\overline{m\triangle n}(u) = \bar{m}(u|_M)\circ\bar{n}(u|_N)$$

and for any word $w\colon M\to A$ and any point $x\in\mathbf{Z}^d$

$$\overline{m\oplus x}(w\oplus x) = \bar{m}(w).$$

Using these notations we get

$$\overline{h^k}(w) = \prod^{<}_{q\in Q} \overline{h^k_q}(w|_{M^k_q}) \quad (k\in \mathbf{N},\ w\in A^{kT},\ q\in Q)$$

and

$$\overline{h^{k+1}_q}(w|_{M^{k+1}_q}) = \prod^{\lessdot}_{(x,r)\in t(q)} \overline{h^k_r}((w\ominus x)|_{kT|M^k_r}) \quad (k\in \mathbf{N},\ w\in A^{(k+1)T},\ q\in Q).$$

The first formula is trivial, whereas the second one is derived in the following way:

$$\overline{h^{k+1}_q}(w|_{M^{k+1}_q}) = \overline{\prod^{\lessdot}_{(x,r)\in t(q)} (h^k_r\oplus x)}(w|_{M^{k+1}_q}) = \prod^{\lessdot}_{(x,r)\in t(q)} \overline{(h^k_r\oplus x)}(w|_{x+M^k_r}) =$$

$$= \prod^{\lessdot}_{(x,r)\in t(q)} \overline{(h^k_r\oplus x)}(w|_{x+kT|x+M^k_r}) = \prod^{\lessdot}_{(x,r)\in t(q)} \overline{(h^k_r\oplus x)}((w\ominus x)|_{kT|M^k_r}\oplus x) =$$

$$= \prod^{\lessdot}_{(x,r)\in t(q)} \overline{h^k_r}((w\ominus x)|_{kT|M^k_r}).$$

Now, let σ be the finite semigroup of transition functions of some finite deterministic automaton which recognizes the language R. Let s_0 be its initial state and G its set of accepting states. As it is well known, we have $R=\{p/p\in A^*$ and $s_0\cdot\sigma_p\in G\}$. Thus we get

$$\hat{h}(R) = \{w/w\in(T, A)^* \quad \text{and} \quad s_0\cdot\sigma_{(\overline{h^k}(w))}\in G\}.$$

Now we apply Theorem 2 to show T-recognizability of $\hat{h}(R)$. To do this, we choose $Z:=\sigma^Q$ and function $g\colon (T, A)^*\to Z$ such that

$$g(w)(q) := \sigma_{(\overline{h^k_q}(w|_{M^k_q}))} \quad (k\in \mathbf{N},\ w\in A^{kT},\ q\in Q).$$

Then we have for any $k\in\mathbf{N}$ and $w\in A^{kT}$:

$$s_0\cdot\sigma_{(\overline{h^k}(w))} = s_0\cdot\sigma_{\left(\prod^{<}_{q\in Q} \overline{h^k_q}(w|_{M^k_q})\right)} = s_0\cdot \prod^{<}_{q\in Q} \sigma_{(\overline{h^k_q}(w|_{M^k_q}))} = s_0\cdot \prod^{<}_{q\in Q} g(w)(q).$$

Thus, we get

$$\forall w\in(T, A)^*\colon\ w\in\hat{h}(R) \Leftrightarrow \Big(s_0\cdot \prod^{<}_{q\in Q} g(w)(q)\Big)\in G$$

or, equivalently,

$$\forall w\in(T, A)^*\colon\ w\in\hat{h}(R) \Leftrightarrow g(w)\in F$$

where $F(\subseteq Z)$ is defined to be

$$F := \Big\{z/z\in Z \quad \text{and} \quad s_0\cdot \prod^{<}_{q\in Q} z(q)\in G\Big\}.$$

Moreover, let $f\colon Z^T\to Z$ be defined by

$$f(v)(q) := \prod^{\lessdot}_{(x,r)\in t(q)} v(x)(r) \quad (v\in Z^T,\ q\in Q).$$

Then, we have for any $k \in \mathbf{N}$, any $w \in A^{(k+1)T}$ and any $q \in Q$:

$$g(w)(q) = \sigma_{\overline{(h_q^{k+1}}(w|_{M_q^{k+1}}))} = \sigma_{\left(\widehat{\circledast}_{(x,r)\in t(q)} \overline{h_r^k}((w\ominus x)|_{kT|M_r^k})\right)} = \widehat{\circledast}_{(x,r)\in t(q)} \sigma_{\overline{(h_r^k}((w\ominus x)|_{kT|M_r^k}))} =$$

$$= \widehat{\circledast}_{(x,r)\in t(q)} \bar{\bar{g}}(w)(x)(r) = f(\bar{\bar{g}}(w))(q).$$

Thus, the entities Z, F, g and f fulfill the conditions of Theorem 2 and, therefore, the T-recognizability of $\hat{h}(R)$ is shown. Because the construction of h has not depended on R, Theorem 11 is proved. □

Theorem 12. There is a context-free language, C_2, with the following property: Let T be any template with exactly 2 extreme points,[7] e_1 and e_2. Let $h_T = (h_T^k)$ be the T-wrap such that h_T^k begins at point ke_1, moves strictly toward ke_2 and ends there (ke_1 and ke_2 are the extreme points of $kT(k \geqq 0)$). Then $\hat{h}_T(C_2)$ is not T-recognizable.

Proof. Let T be a template with exactly two extreme points. We will give some context-free language C_2 such that any T-wrap, h, beginning with the one extreme point of kT and moving strictly toward the other one yields a non-T-recognizable T-language. Without loss of generality we may assume that $\mathbf{0} \in T$ and, therefore, that $T \subseteqq \mathbf{Z}$. Moreover, let $\mathbf{0}$ be the left extreme point of T, i.e.: we take $T = \{0 = x_1, x_2, \ldots, x_s = m\}$ with $x_1 < x_2 < x_3 < \ldots < x_s$. Furthermore, let us assume that $\gcd(x_2, x_3, \ldots, x_s) = 1$[8]. (These restrictions are without loss of generality, because they correspond to certain affine transformations.) Then, according to [5], there exist two natural numbers, 1 and r, such that for all $k \geqq k_0$ ($k_0 := m^2 \cdot s$) it holds

$$kT = \underline{\underline{M}} \cup [l, km-r] \cup (km - \overline{\overline{M}}) \tag{10}$$

where $\underline{\underline{M}} \subseteqq [0, l-2]$, $\overline{\overline{M}} \subseteqq [0, r-2]$ and $[i, j]$ denotes the set of all integers between and including i and j.

Now, define the context-free language C_2 ($\subseteqq A^*$ with $A = \{a, b, c, d, \S\}$) by

$$C_2 := \bigcup_{i,j \in \mathbf{N}} aa^i \S (a, b, c, d)^* ca^i db^j c(a, b, c, d)^* \S b^j b.$$

Because C_2 is quasi-symmetric we restrict our considerations to T-wraps h from left to right. In the sequel let, for any word $p \in A^*$, $\bar{p}$ denote $\hat{h}(p)$ (if it is defined for that p). The proof that $\hat{h}(C_2)$ is not T-recognizable is carried out using Theorem 5.

[7] For the reader who is not familiar with convex sets we recapitulate the notion of convex hulls and extreme points: Let $\mathbf{R}$ denote the set of all real numbers and $\mathbf{R}^d$ the set of all d-tuples of real numbers. For any finite, not empty set $M \subseteqq \mathbf{Z}^d$ ($\subseteqq \mathbf{R}^d$) let $\tilde{M}$ denote the *convex hull* of M, defined by

$$\tilde{M} := \{\Sigma_{y\in M} a_y \cdot y / 0 \leqq a_y \leqq 1 (y \in M) \quad \text{and} \quad \Sigma_{y \in M} a_y = 1\}.$$

A point $x \in M$ is called an *extreme point* of M if any representation $x = \Sigma_{y\in M} a_y \cdot y$ with $0 \leqq a_y \leqq 1$ ($y \in M$) and $\Sigma_{y\in M} a_y = 1$ implies $a_x = 1$ and $a_y = 0 (y \neq x)$. It is matter of triviality that a point e is an extreme point of template T iff ke is an extreme point of kT ($k \geqq 1$).

[8] gcd = greatest common divisor.

In order to do this, set $N_k := \{(i,j)/0 \leqq i, j \leqq k\}$ $(k \in \mathbf{N})$. For any $k \in \mathbf{N}$ and any function $f: \mathbf{N}^2 \to \{0, 1\}$ let w_f^k denote the word

$$w_f^k := \overline{aa^k \S (\underset{(i,j)\in N_k}{\lesssim} v_{f,i,j}^k)(u_f^k) \S b^k b}$$

where

$$v_{f,i,j}^k := \begin{cases} ca^i db^j c & \text{iff } f(i,j) = 1 \\ cc & \text{iff } f(i,j) = 0 \end{cases} \quad (i,j) \in N_k$$

and u_f^k is chosen as a sequence of c's such that w_f^k fits into some KT (e.g.: the next smallest) with $(K-k) \geqq k_0$ $(K = K(k,f))$; let $<$ be any fixed total ordering of set $\mathbf{N}^2$.

For any $k \in \mathbf{N}$ with $km \geqq \max(l, r)$, we represent the words w_f^{2km} as

$$w_f^{2km} = \overline{aa^{2km} \S q_f^{2km} \S b^{2km} b}$$

where q_f^{2km} is appropriately chosen from A^*. Then, for any i $(0 \leqq i \leqq k)$ and any $x \in iT$ (which implies $0 \leqq x \leqq im$), the word ${}_{i,x}w_f^{2km}$, defined by

$${}_{i,x}w_f^{2km} := (w_f^{2km} \ominus x|_{(D(w_f^{2km})-i)T}),$$

has the form

$${}_{i,x}w_f^{2km} = \overline{aa^{F(k,i,x)} \S q_f^{2km} \S b^{G(k,i,x)} b}$$

where

$$F(k, i, x) = 2km - x$$

and

$$G(k, i, x) = 2km - (im - x) = (2k - i)m + x.$$

(For an illustration see Fig. 4; note that we have $K - i \geqq k_0$ $(K = K(2km, f))$ which, in turn, implies that ${}_{i,x}w_f^{2km}$ has domain $\underline{\underline{M}} \cup [l, (K-i)m - r] \cup (K-i)m - \overline{\overline{M}}$. Thus, ${}_{i,x}w_f^{2km}$ is taken from w_f^{2km} by only removing x a's from left and $im - x$ b's from right.)

Thus, we have ${}_{i,x}w_f^{2km} \in \hat{h}(C_2)$ iff $ca^{F(k,i,x)} db^{G(k,i,x)} c$ is contained in q_f^{2km} which, in turn, holds iff $f(F(k, i, x), G(k, i, x)) = 1$. Therefore, for any two functions f, f': $\mathbf{N}^2 \to \{0, 1\}$ which differ on at least one point of $\{(F(k, i, x), G(k, i, x))/ /0 \leqq i \leqq k, x \in iT\}$ $(=: R_k)$, we get that w_f^{2km} and $w_{f'}^{2km}$ are not $k - \hat{h}(C_2)$-equivalent. Now, clearly, $0 \leqq i, i' \leqq k$, $x \in iT$, and $x' \in i'T$ with $(i, x) \neq (i', x')$ implies that $(F(k, i, x), G(k, i, x)) \neq (F(k, i', x'), G(k, i', x'))$ which, in turn, yields $\operatorname{card}(R_k) = \operatorname{card}(\{(i, x)/0 \leqq i \leqq k, x \in iT\})$. Therefore we get at least $2^{(e_{k,T})}$ $k - \hat{h}(C_2)$-equivalence classes. Furthermore, from (10) we get that $d_{k,T}$ equals asymptotically km and $e_{k,T}$ equals asymptotically $k^2m/2$. Thus $E_{k-\hat{h}(C_2)}$ cannot be bounded by any $C^{(d_{k,T})}$ which proves our claim that $\hat{h}(C_2)$ cannot be T-recognized. □

Theorem 13. There is a context-free language, C_3, with the following property: Let T be any template with exactly 3 extreme points, e_1, e_2 and e_3. (This implies that kT has also exactly 3 extreme points, namely ke_1, ke_2 and ke_3.) Let $h = (h^k)_{k \geqq 0}$ be any T-wrap such that h^k begins with the (possibly sparsely filled) "line" $ke_1 \to ke_2$, fills that row completely (in any order), moves then to the next parallel row, fills

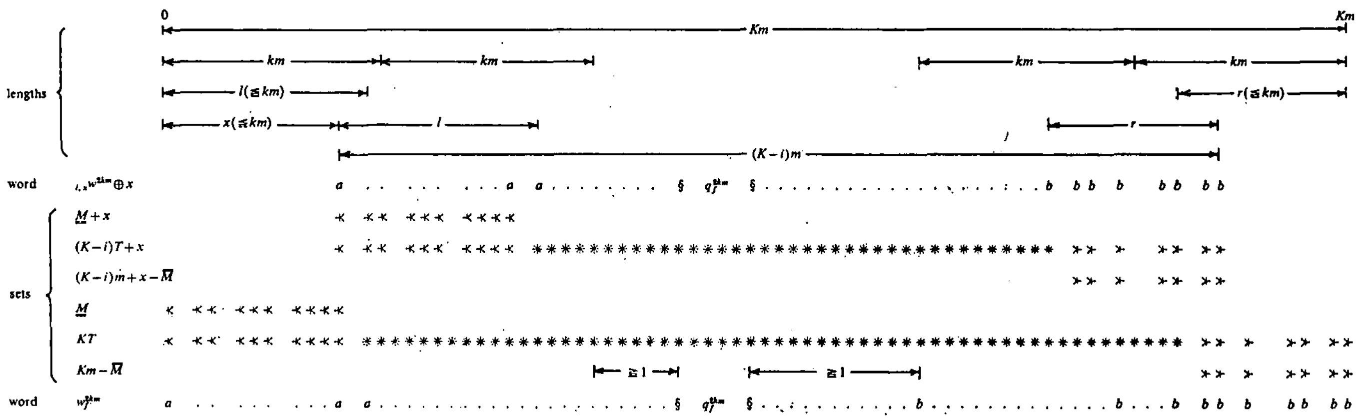


Fig.4

Illustration of the words w_f^{2km} and ${}_{i,x}w_f^{2km}$ $(0\leqq i\leqq k,\ x\in iT\ (\Rightarrow 0\leqq x\leqq km))$. Note that $(K-i)T+x\subseteqq KT$ and $K=K(2km,f)$. $\ast\ast\ldots\ast$ $(\ast\ast\ldots\ast)$: possible pattern of set $\underline{\underline{M}}\,(-\overline{\overline{M}})$

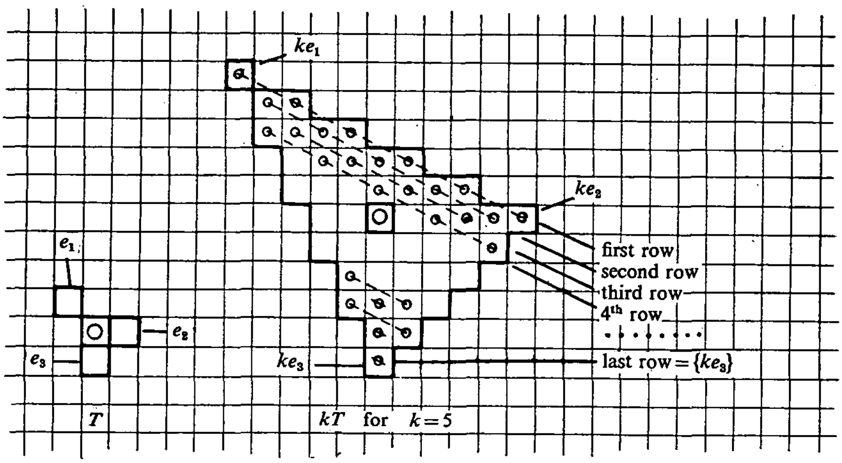


Fig.5
Illustration to help understanding the T-wraps mentioned in Theorem 13 (for some 2-dimensional template T with exactly 3 extreme points e_1, e_2 and e_3)

this row (in any order), goes then to the next row etc. until it reaches the third extreme point, ke_3 (sc. Fig. 5). Then $\hat{h}(C_3)$ is not T-recognizable.

Proof. Let $T \subseteqq \mathbf{Z}^d$ be a neighbourhood template with exactly 3 extreme points e_1, e_2 and e_3. Let $\hat{h}=(h^k)_{k\in\mathbf{N}}$ be a T-wrap as described in Theorem 13. We will show that, for the context-free language $C_3 := \{w/w\in\{0, 1\}^*$ and $w=w^R\}$, i.e.: the set of all palindromes over alphabet $A=\{0, 1\}$, $\hat{h}(C_3)$ is not T-recognizable.

To pursue the proof we assume that $\hat{h}(C_3)$ is T-recognizable and apply Theorem 6 to get a contradiction. Without loss of generality we may presuppose that the origin $\mathbf{0}$ is one of the extreme points, e_3, say, and that dimension $d=2$. As one easily verifies, we get that $\mathbf{0}$, ke_1 and ke_2 are the (only) extreme points of kT and

$$kT \subseteqq \overline{\{\mathbf{0}, ke_1, ke_2\}}. \tag{11}$$

Now we look for entities which fulfill the conditions of Theorem 6. For any $k\in\mathbf{N}$, $0\leqq j\leqq k$, we define the sets N_k, $M_{k,j}$ and M_k and the number $\bar{k}$ in the following way:

$$\begin{aligned} N_k &:= \overline{\{ke_1, ke_2\}}\cap kT, \\ M_{k,j} &:= \overline{\{\mathbf{0}, je_1, je_2\}}\cap kT \\ \bar{k} &:= \min\,\{j/0 \leqq j \leqq k,\ \operatorname{card}(M_{k,j}) \geqq \operatorname{card}(N_k)\} \\ M_k &:= M_{k,\bar{k}}. \end{aligned} \tag{12}$$

Clearly, N_k and M_k are subsets of kT $(k\in\mathbf{N})$. Because N_k contains at least the points $je_1+(k-j)e_2$ $(0\leqq j\leqq k)$ which are all different and because $\overline{\{ke_1, ke_2\}}$ contains at most $k\sqrt{(e_1-e_2)^2}+1$ elements of $\mathbf{Z}^d$ there is a positive constant C_1 such that

$$k+1 \leqq \operatorname{card}(N_k) \leqq C_1\cdot k \quad (k \geqq 1).$$

On the other hand, $M_{k,j}$ contains at least the points $0+i_1e_1+i_2e_2$ $(0\leqq i_1,\ i_2\leqq j,\ i_1+i_2\leqq j)$ which are all different. Thus,

$$\operatorname{card}(M_{k,j}) \geqq j^2/2 \quad (0\leqq j\leqq k).$$

Therefore we get $\bar{k}\leqq\lceil\sqrt{2C_1k}\rceil$ which implies that there is a constant C_2 with

$$\bar{k}\leqq C_2\sqrt{k} \quad (k\geqq 1). \tag{13}$$

Particularly, this means that there is a $k_0\in\mathbf{N}$ such that $k>\bar{k}$ $(k\geqq k_0)$.

Now, for $k>k_0$, set $i(=i(k)):=k-\bar{k}-1$. Then we have

$$(N_k-iT)\cap(k-i)T\subseteqq\overline{\{(\bar{k}+1)e_1,\ (\bar{k}+1)e_2\}} \quad (k>k_0) \tag{14}$$

and

$$(M_k-iT)\cap(N_k-iT)\cap(k-i)T=\emptyset \quad (k>k_0). \tag{15}$$

This is shown in the following way: Obviously, according to (11) and (12), we have (for $k>k_0$)

$$(N_k-iT)\cap(k-i)T\subseteqq\left(\overline{\{ke_1,\ ke_2\}}-\overline{\{0,\ ie_1,\ ie_2\}}\right)\cap\overline{\{0,\ (k-i)e_1,\ (k-i)e_2\}}.$$

Furthermore, any element x of the right set has two representations $x=a_1ke_1+a_2ke_2-b_1ie_1-b_2ie_2$ and $x=c_1(k-i)e_1+c_2(k-i)e_2$ with $0\leqq a_1, a_2, b_1, b_2, c_1, c_2\leqq 1$ and $a:=a_1+a_2=1$, $0\leqq b:=b_1+b_2\leqq 1$ and $0\leqq c:=c_1+c_2\leqq 1$. Because e_1 and e_2 are linearly independent, we have $(a_1k-b_1i)=c_1(k-i)$ and $(a_2k-b_2i)=c_2(k-i)$ and, summing up both sides, $k-bi=c(k-i)$. Evaluating i yields $(1-b)k+b(\bar{k}+1)=c(\bar{k}+1)$. Because $\bar{k}+1\leqq k$, this is possible only if $c\geqq 1$, which yields $c_1+c_2=1$ and shows that x is a member of the right set of (14). Because $k-i=\bar{k}+1$, (14) is proved.

To prove (15) we use (14): Let $x\in M_k-iT$; then, using (11) and (12), we get that x is a member of $\overline{\{0,\bar{k}e_1,\bar{k}e_2\}}-\overline{\{0,\ ie_1,\ ie_2\}}$, too and, therefore, has representation $x=a_1\bar{k}e_1+a_2\bar{k}e_2-b_1ie_2-b_2ie_2$ with $0\leqq a_1, a_2, b_1, b_2\leqq 1$, $0\leqq a:=a_1+a_2\leqq 1$ and $0\leqq b:=b_1+b_2\leqq 1$. Because e_1 and e_2 are linearly independent, we get, evaluating i, that $x=d_1(\bar{k}+1)e_1+d_2(\bar{k}+1)e_2$ where $d_j=\left(a_j\bar{k}-b_j(k-\bar{k}-1)\right)/(\bar{k}+1)$ are uniquely determined $(j=1,2)$. Now we have $d_1+d_2=\left((a+b)\bar{k}+b(1-k)\right)/(\bar{k}+1)=$ $=b+a\bar{k}/(\bar{k}+1)-bk/(\bar{k}+1)\leqq a\bar{k}/(\bar{k}+1)<1$ (because $\bar{k}<k$). Thus x is not a member of $\overline{\{(\bar{k}+1)e_1,(k+1)e_2\}}$ and, because of (14), not a member of $(N_k-iT)\cap(k-i)T$. This proves (15).

Furthermore, because of $M_k\subseteqq\overline{\{0,\bar{k}e_1,\bar{k}e_2\}}$, $N_k\subseteqq\overline{\{ke_1,\ ke_2\}}$, e_1 and e_2 linearly independent and $k>\bar{k}$ for $k>k_0$ we conclude that

$$M_k\cap N_k=\emptyset \quad (k>k_0).$$

Now, for any $k>k_0$ and any word $p\in A^*$ of length $\operatorname{card}(kT)$, $\hat{h}(p)$ is constructed by filling N_k with the first $\operatorname{card}(N_k)$ symbols of p and filling M_k with the last $\operatorname{card}(M_k)$ $(\geqq\operatorname{card}(N_k))$ symbols of p. Let $g_k\colon N_k\to M_k$ be defined by

$$g_k(x)=x' \quad\text{iff}\quad h^k(x)=\operatorname{card}(kT)-h^k(x')+1 \quad (k>k_0,\ x\in N_k).$$

Clearly, g_k is injective. (Informally, for any $x\in N_k$, if the first j^{th} symbol of p is placed at point x, then the last j^{ht} symbol is placed at $g_k(x)$ $(1\leqq j\leqq\operatorname{card}(N_k))$.)

Let $\underset{\sim}{M}_k$ and $\underset{\sim}{N}_k$ denote the set of all functions $n: N_k \to \{0, 1\}$ or $m: M_k \to \{0, 1\}$, resp. Furthermore, for any $k > k_0$, any $m \in \underset{\sim}{M}_k$ and $n \in \underset{\sim}{N}_k$ define the word $w^k_{m,n}$ $(\in \{0, 1\}^{kT})$ as follows:

$$w^k_{m,n|N_k} := n$$

$$w^k_{m,n|M_k} := m$$

$$w^k_{m,n|kT \setminus (M_k \cup N_k)} :\equiv 0.$$

Thus we have from the construction of $w^k_{m,n}$ that (1)—(4) are fulfilled. Now, $w^k_{m,n}$ is an element of $\hat{h}(C_3)$ iff $n(x) = m(g_k(x))$ for all $x \in N_k$ and $m|_{M_k \setminus g_k(N_k)} \equiv 0$. Therefore, for any two $n, n' \in \underset{\sim}{N}_k$ with $n \neq n'$ choose $m \in \underset{\sim}{M}_k$ such that $m(g_k(x)) = n(x)$ for $x \in N_k$ and $m|_{M_k \setminus g_k(N_k)} \equiv 0$. Then, clearly, we have

$$w^k_{m,n} \in \hat{h}(C_3) \quad \text{and} \quad w^k_{m,n'} \notin \hat{h}(C_3)$$

which establishes (5). Moreover, setting $i = i(k) := k - \bar{k} - 1$, we have $0 \leqq i(k) \leqq k$ and (15) which resembles (6).

Because we have assumed that $\hat{h}(C_3)$ is T-recognizable, Theorem 6 allows us to conclude that there is some constant C which does not depend on k such that

$$\text{card}(\underset{\sim}{N}_k) \leqq C^{\text{card}((N_k - iT) \cap (k-i)T)} \quad (k > k_0).$$

Clearly, $\text{card}(\underset{\sim}{N}_k) \geqq 2^{k+1}$. On the other hand, because of (14) we have $\text{card}((N_k - iT) \cap (k-i)T) \leqq (\bar{k}+1)\sqrt{(e_1 - e_2)^2} + 1 \overset{(13)}{\leqq} C_2' \sqrt{k}$ (with appropriate constant C_2'). Thus, we would get

$$2^{k+1} \leqq C^{C_2' \cdot \sqrt{k}} \quad (k > k_0)$$

which, clearly, is impossible. This proves that $\hat{h}(C_3)$ is not T-recognizable. □

Theorem 14. There is a context-free language, C_4, such that, for any dimension d and any template $T (\subseteqq \mathbf{Z}^d)$ which contains more than 3 extreme points, there is no T-wrap at all such that $\hat{h}(C_3)$ is T-recognizable.

Proof. Let T be a neighbourhood template with 4 or more extreme points and $h = (h^k)_{k \in \mathbf{N}}$ be any T-wrap. We will show that, for the context-free language C_4 which contains all words p over the alphabet $A = \{a, b, c\}$ which exhibit (any number of c's and) exactly as many b's as a's, the language $\hat{h}(C_4)$ is not T-recognizable.

To carry out the proof, we assume that $\hat{h}(C_4)$ is T-recognizable and apply Theorem 6 to get a contradiction.

Let $E = \{e_1, e_2, e_3, \ldots, e_n\}$ $(n \geqq 4)$ be the set of all extreme points of T. Then, clearly, $\{ke_1, ke_2, \ldots, ke_n\}$ is the set of all extreme points of kT $(k \in \mathbf{N})$. Furthermore, we can choose two pairs of extreme points (e_1, e_2) and (e_3, e_4), say, such that $\overline{\{e_1, e_2\}}$ and $\overline{\{e_3, e_4\}}$ constitute disjoint extreme edges of $\bar{T}$, i.e.: for any point $x \in \overline{\{e_1, e_2\}}$ and any representation $x = \Sigma_{y \in T} a_y \cdot y$ with $0 \leqq a_y \leqq 1$ $(y \in T)$ and $\Sigma_{y \in T} a_y = 1$ we get $a_y = 0$ $(y \notin \{e_1, e_2\})$ and for any point $x \in \overline{\{e_3, e_4\}}$ and any representation $x = \Sigma_{y \in T} a_y \cdot y$ (with $0 \leqq a_y \leqq 1$ $(y \in T)$ and $\Sigma_{y \in T} a_y = 1$) we get $a_y = 0$ $(y \notin \{e_3, e_4\})$.

Moreover, $\overline{\{e_1, e_2\}} \cap \overline{\{e_3, e_4\}} = \emptyset$. This implies that $\overline{\{ke_1, ke_2\}}$ and $\overline{\{ke_3, ke_4\}}$ are the corresponding (disjoint) extreme edges of $\overline{kT}$ $(k \geqq 1)$ [9].

Now for any $k \geqq 1$, $0 \leqq j \leqq k$ let $y_j^k := je_1 + (k-j)e_2$ and $x_j^k := je_3 + (k-j)e_4$. Clearly, x_j^k, $y_j^k \in kT$ and all points x_j^k and y_j^k, are different. Let $N_k := \{y_j^k / 0 \leqq j \leqq k\}$ and $M_k := \{x_j^k / 0 \leqq j \leqq k\}$. Set $\underset{\sim}{M}_k := \underset{\sim}{N}_k := \{0, 1, \ldots, k\}$ and for $m \in \underset{\sim}{M}_k$, $n \in \underset{\sim}{N}_k$ define $w_{m,n}^k$ $(\in A^{kT})$ by

$$w_{m,n}^k(x_j^k) := \begin{cases} a & \text{if} \quad 0 \leqq j < m \\ c & \text{if} \quad m \leqq j \leqq k \end{cases}$$

$$w_{m,n}^k(y_j^k) := \begin{cases} b & \text{if} \quad 0 \leqq j < n \\ c & \text{if} \quad n \leqq j \leqq k \end{cases}$$

$$w_{m,n|kT \setminus (N_k \cup M_k)}^k :\equiv c.$$

Thus $w_{m,n}^k$ depends on n only at N_k and on m only at M_k which implies conditions (1)—(4) of Theorem 6. Furthermore, for $n, n' \in \underset{\sim}{N}_k$ with $n \neq n'$, we get $w_{n,n}^k \in \hat{h}(C_4)$ whereas $w_{n,n'}^k \notin \hat{h}(C_4)$ because in any word $\tilde{w}_{m,n}^k$ the number of occurring a's differs from the number of occurring b's by exactly $|m-n|$. Hence, (5) is fulfilled, too.

Now, let $i (= i(k) := k-1)$ $(k \geqq 1\,(!))$. We have to ensure that

$$(N_k - iT) \cap (M_k - iT) \cap T = \emptyset. \tag{16}$$

This is true, because otherwise there would exist a point $x \in T$ with representations

$$x = je_1 + (k-j)e_2 - \Sigma_{y \in T} l_y \cdot y \;\; (0 \leqq j \leqq k,\; l_y \in \mathrm{N},\; \Sigma_{y \in T} l_y = k-1)$$

and

$$x = j'e_3 + (k-j')e_4 - \Sigma_{y \in T} l'_y \cdot y \;\; (0 \leqq j' \leqq k,\; l'_y \in \mathrm{N},\; \Sigma_{y \in T} l'_y = k-1).$$

This fact would imply that $(x + \Sigma_{y \in T} l_y \cdot y)/k = \frac{j}{k} e_1 + \frac{k-j}{k} e_2$ $(\in \overline{\{e_1, e_2\}}$ and $(x + \Sigma_{y \in T} l'_y \cdot y)/k = \frac{j'}{k} e_3 + \frac{k-j'}{k} e_4$ $(\in \overline{\{e_3, e_4\}})$. However, because $\overline{\{e_1, e_2\}}$ and $\overline{\{e_3, e_4\}}$ are extreme edges of $\overline{T}$, we might conclude that $x \in \overline{\{e_1, e_2\}}$ and $x \in \overline{\{e_3, e_4\}}$, which, obviously, is a contradiction to the assumption that these two extreme edges are disjoint. Thus, (16) and therefore (6) is fulfilled $(k \geqq 1)$.

Theorem 6 tells us that, in this case, there is a constant C such that card $(\underset{\sim}{N}_k) \leqq C^{\mathrm{card}((N_k - iT) \cap (k-i)T)}$ $(k \geqq 1,\; i = i(k) = k-1)$. However, card $(\underset{\sim}{N}_k) = k+1$ whereas card $((N_k - iT) \cap (k-i)T) \leqq$ card (T). Therefore the inequality just now mentioned can not be true. Thus $\hat{h}(C_4)$ is not T-recognizable. □

4. Conclusion and summary

Using new notions of (d-dimensional) languages and their recognition which seem to be more adequate to the phenomena occuring in d-dimensional cellular automata, we could generalize and improve the results of Seiferas [12] concerning the recognition speed of regular languages in such structures. Smith [11] raised the ques-

[9] This is an elementary fact which is easily proved using basic properties of convex set (cp. [4]).

tion whether contextfree languages, inscribed in a "natural" way into one-dimensional cellular automata, can be recognized in real-time or not. In our sense, this question is answered in the negative in a special case of dimension one as well as in a very general way for arbitrary dimensions ($d>1$). Thus we have found a further property, in which regular languages differ essentially from context-free ones.

Acknowledgements. The author would like to thank Prof. Dr. R. Vollmar and W. O. Höllerer for many stimulating discussions about the topic as well as about editorial questions. Thanks also to Dr. Brandenburg for editorial remarks.

Abstract

This is a new approach to recognize formal languages by deterministic d-dimensional off-line cellular automata. It allows to exploit the parallelism inherent in such devices in a higher rate than this is done by two other approaches already known. Although the proposed notion of recognition turns out to be the strongest one known to date, the known results concerning real-time recognition of regular languages can be improved (for all dimensions). On the other hand, the strength of this notion allows us to show — under some very general assumptions — the non-recognizability of context-free languages in real-time (for all dimensions).

LEHRSTUHL C FÜR INFORMATIK
TECHNISCHEN UNIVERSITÄT BRAUNSCHWEIG
GAUSS STRASSE 11
D—3300 BRAUNSCHWEIG, FRG

Literature

[1] CODD, E. F., *Cellular automata,* New York, London, 1968.
[2] COLE, S. N., Real-time computation by n-dimensional iterative arrays of finite-state machines, IEEE Conference Record of Seventh Annual Symposium on Switching and Automata Theory, 1966, pp. 53—77.
[3] GARDNER, M., The fantastic combination of John Conway's new solitaire game "life", *Scientific American,* v. 224, 1970, pp. 120.
[4] GRÜNBAUM, B., *Convex polytopes,* London, New York, Sidney, 1967.
[5] NATHANSON, M. B., Sums of finite set of integers, *Amer. Math. Monthly,* v. 79, 1972, pp. 1010—1012.
[6] PECHT, J. G., Ein weiterer Ansatz zur Mustertransformation und -erkennung in zellularen Räumen, Dissertation, TU Braunschweig, 1980.
[7] PECHT, J. G., T-recognition of T-languages: a new approach to describe and exploit the parallel capabilities of d-dimensional tessellation structures, submitted for publication, 1980.
[8] PECHT, J. G., A recursive decomposition property of sets of finite sums of d-tuples of integers, to submit, 1981.
[9] SALOMAA, A., *Formal languages,* New York, 1973.
[10] SMITH III, A. R., Cellular automata theory, Technical Report no. 2, Stanford University, Stanford, Cal., 1969.
[11] SMITH III, A. R., Real-time language recognition by one-dimensional cellular automata, *J. Comput. System Sci.,* v. 6, 1972, pp. 233—253.
[12] SEIFERAS, J. I., Linear-time computation by nondeterministic multi-dimensional iterative arrays, *SIAM J. Comput.,* v. 6, 1977, pp. 487—504.
[13] VOLLMAR, R., *Algorithmen in Zellularautomaten,* Stuttgart, 1979.
[14] YAMADA, H. and S. AMOROSO, Tessellation automata, *Inform. and Control,* v. 14, 1969, pp. 299—317.

(Received April 13, 1981)

Locally synchronous cellular automata

By I. WACHSMUTH

1. Cellular automaton and the synchronization problem*

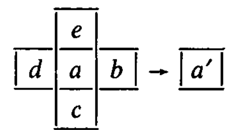
The concept of cellular automaton was evolved by John von Neumann when he dealt with questions concerning the capabilities of machines, in particular the feasibility of a mechanical self-reproduction [6], see also [2]. A cellular automaton is a "macro automaton" or "polyautomaton" composed in a uniform manner out of a not a priori bounded number of "micro automata" called *cells*. These are practically thought of as being arranged at the integer lattice points of the Euclidian plane (or in general the Euclidian n-space). The cells are interconnected in a uniform local scheme called the *neighbourhood connection*, and the cells directly connected to a given cell are called its *neighbours*. Each of the cells can be in one of a finite number of *states* which can be changed at certain times. The next state of a cell depends on its actual state and those of its neighbours. This dependency is described by a finite set of *local transition rules* which is assumed to be the same for each individual cell.

A cellular automaton is thus characterized by a tupel $Z=(\mathbf{Z}^2, N, Q, \Sigma)$ where $\mathbf{Z}^2$ is the universe, N is the neighbourhood set given by a finite number of vectors, Q is the finite set of cell states having at least two elements, and Σ is the finite set of local rules. The basic idea is that in the simultaneous interplay "the whole is more than the sum of the parts", as was shown, for example, in universal computation, self-reproduction (of patterns figured by cell states), pattern recognition and -transformation. An essential topic which seems to be considered still in its first steps is *parallel computing*, particularly with regard to so-called myopic algorithms, the elementary operations of which take reference only to bounded, well-defined subsets of data, for example *cellular algorithms* as introduced by LEGENDI [4].

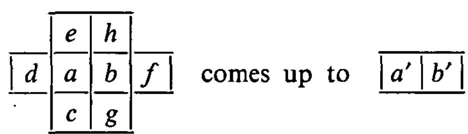
Such formal computation procedures can be conceived as *deductions* in a special type of a "more-dimensional" substitution calculus which in the following shall be called *cellular calculus:* The alphabet consists of the cells' states symbols, the dimension being given by the underlying space which can be thought of as a frame set of

* To avoid confusion it should be mentioned that quite a different type of "synchronization problems" is known in the literature such as the "firing squad synchronization problem" introduced by John Myhill.

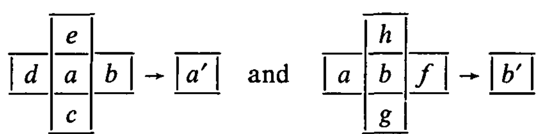
symbol fields, and the basic substitution rules are given by the local transition rules. In the von Neumann two-dimensional cellular automaton and related cases with neighbourhood set $\{(0, 0), (1, 0), (0, -1), (-1, 0), (0, 1)\}$ these rules are of the following normed shape (a, b, c, d, e, a' denote state symbols):



The cellular calculus operates on two-dimensional words over the state symbol alphabet which are called *patterns*. To determine the manner of rule application designating a deduction step in the calculus, the specification of a *meta rule* is needed: Here in the von Neumann concept, in each single step, the *simultaneous* substitution of the whole set of rule-shaped subwords is considered, according to the basic substitution rules. Wherever at adjacent symbol fields the rule premises are overlapping they must refer mutually to the state symbols given *before* application of a substitution rule, i.e.,



by the rules



Cellular calculus and cellular automaton can be regarded as being in the relationship "rules of a game" versus "game"; and to perform an *adequate* execution (according to the meta and basic rules — throughout this paper the term "adequate" shall be fixed for this notation) of the "game" by the automaton, it is a usual assumption that a *global clock* gives rise to a *synchronous* switching of state transitions of all automata cells. From this synchronity assumption an organization problem arises which is here referred to as the *synchronization problem:* synchronizing a not a priori bounded number of cells, for a cellular automaton has to be considered as an unbounded, potentially infinite automaton (for detail see [9]).

Physically motivated objections against such a synchronity assumption led to the development of so-called *asynchronous cellular automata* introduced by NAKAMURA [5], PRIESE [7] and GOLZE [3], where besides other things, put briefly, the meta rule of the underlying calculus is changed: The simultaneous application of the basic rules (and thereby the "grade of parallelism") is more or less restricted. A synchronization problem as in the case of a "simultaneous cellular calculus" does not appear in these cases, see [8].

The approach we take in this paper is shown in the following questions.

— Cannot the original concept of simultaneous cellular calculus as a model for highly parallel information processing be kept, and the execution of such a calculus by a cellular automaton be organized in such a manner that, for building a synchronization scheme, no reference to the whole number of cells is required?

— Can a cellular calculus in the sense of John von Neumann be executed in an *adequate* manner only by a synchronous automaton?

In other words:

— Is the *sufficient* condition of synchronity also *necessary*?

In the following paragraphs firstly a weaker assumption of "local synchronity" will be introduced and discussed and, secondly, a scheme will be designed which could lead to a "locally synchronous performance" of a cellular automaton by which a simultaneous cellular calculus can be adequately executed.

2. Synchronous and locally synchronous working

The main difference between regarding a calculus as an ideal system, and an automaton executing that calculus, seen as a physical device, is that the latter does while the former does not submit to certain physical restrictions; so that in the automaton case we have to consider physical limitations such as bounded signal velocity, delay, and bounded exactness of properties of materials — from this point the discussion of asynchronous automata has arisen.

Let us assume that clock signals are used each to initiate one deduction step in a simultaneous cellular calculus being executed by an appropriate cellular automaton. If accepted that signal transmission cannot instantaneously reach cells in a certain spatial distance from the signal outspring there arises at once the problem of time:

— At what time will a state transition of a given cell be initiated?

— Are all cells reached by the signal in a time interval small enough to perform an adequate execution of one deduction step as indicated above?

— At what delay after initiating the execution of one deduction step could the execution of another step be initiated?

All these problems are ignored when considering a discrete time scale $\vec{t}$ by saying, as is usual, that time passes by in discrete steps $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow \dots$; each time a time step has elapsed each cell must have executed one state transition. From this it is possible to consider the *configuration* (whole or global state) c of a cellular automaton which is simply defined by the infinite cartesian product of all cell states and can be formally denoted by a mapping $c\colon \mathbf{Z}^2 \rightarrow Q$, where Q is the set of state symbols.

Furthermore, a *global transformation* $\bar{\Sigma}$ — corresponding to a deduction step in the calculus given by the simultaneous application of the local rules in Σ to all neighbourhood-shaped patterns — can be considered which leads from a given configuration c to one out of a finite set of possible successor configurations, c'. If the rule set Σ is deterministic which means that no two rules with identical premises leading to different conclusions exist, $\bar{\Sigma}$ can be thought of as a *global transition function*.

Having the concept of "configuration", and supposing that a synchronous state transition switching of all cells of a cellular automaton is given, it is possible to say that each time a time step has elapsed the cellular automaton has changed its global state (configuration) once; and a "run" of a cellular automaton is seen as a sequence of configurations, starting from an initial configuration.

As seen from this, however, a *concept of synchronity* is needed when speaking of a configuration and a global transition of a cellular automaton. To discuss such a concept in relation to the above restrictions, we now introduce a continuous time scale to be thought of as an observation time. This refinement of the point of view will be seen later as an intermediate step towards the tool of explaining a locally synchronous performance.

Let us consider a totally ordered *time scale* $\mathscr{t}=(\mathbf{R}, \leqq, t_1)$, bounded from below and referring to the point of signal source, say the origin of $\mathbf{Z}^2$. We call *instants* the elements of $\mathscr{t}$ and *durations* the lengths of intervals out of $\mathscr{t}$. We denote the lattice points by $\alpha, \beta, \gamma, \ldots$, thinking of them as being the cells. At certain instants $t_1, t_2, t_3, \ldots$ clock signals $T_1, T_2, T_3, \ldots$ shall be propagated from the origin.

We write $\langle t_k^\alpha \rangle$ to fix the *time-spatial event* that at cell $\alpha (\alpha \in \mathbf{Z}^2)$ at time t_k^α $(t_k^\alpha \in \mathscr{t})$ a unique clock pulse appears initiating a k-th state transition of α, assuming that, at this time, α is in a well-defined state q $(q \in Q)$. We shall refer to an event $\langle t_k^\alpha \rangle$ as the k-th *clock pulse at* α.

As indicated above, there may be objections when assuming that an *unbounded* number of cells could be synchronized while this would be certainly possible for any given fixed number of cells. Furthermore, an *absolutely* synchronous cell state switching would not be possible in a physical device and therefore a small amount of phase variations not leading to mishaps should be admissible. From that it is carried out the following definition.

Definition 1. A finite set of clock-pulsed cells is said to be *synchronously working,* if for any given clock signal T_k, in the *whole* set of cells the pairwise difference in time of k-th clock pulses of cells is at most ε s.t.

(i) the correct execution of local transitions is not affected,

(ii) ε is not exceeded after an arbitrary number of clock steps.

Since it will take a nonzero but bounded duration s until the state transition of any given cell is executed, say

$$0 < s_{\min} \leqq s \leqq s_{\max} < \infty \tag{1}$$

in all cases, the minimum signal distance in a synchronously working cellular automaton, leading with certainty to a one-to-one execution of each global transition by reason of one signal, is given by

$$\varepsilon + s_{\max} \tag{2}$$

In opposition to this, *synchronity* of clock pulse events would usually be defined, as an *equivalence relation,* in the following way.

Definition 2. Two events $\langle t_k^\alpha \rangle$ and $\langle t_l^\beta \rangle$, where t_k^α, $t_l^\beta \in \mathscr{t}$, are said to be *synchronous* iff $t_k^\alpha = t_l^\beta$. To denote that these two events are in the synchronity relation we shall write $\langle t_k^\alpha \rangle \operatorname{syn} \langle t_l^\beta \rangle$.

Surely, in a synchronously working set of cells, if $\langle t_k^\alpha \rangle \operatorname{syn} \langle t_l^\beta \rangle$ then $k=l$. Consider, for the moment, the case that all cells of an unbounded cellular automaton Z undergo synchronous state transitions by reason of clock pulses occurring synchronously at each individual cell, without discussion whether or not this is possible. In this case, for a given k, and with 0 denoting the origin of $\mathbf{Z}^2$, the set $\{\alpha \mid \langle t_k^\alpha \rangle \operatorname{syn} \langle t_k^0 \rangle\}$ consists of the whole set of cells of Z and thus at time t_k^0 a well-defined configuration of Z, resp. the initiation of the k-th global transition of Z, can be considered ($k=1, 2, 3, \ldots$).

The relation between the continuous time scale $\ell=(\mathbf{R}, \leqq, t_1)$ and a discrete time scale $\vec{\ell}$ is worked out in the following. Denote the times of input of clock signals $T_1, T_2, \ldots$ at the origin 0 of a cellular automaton Z by $t_1^0, t_2^0, \ldots$, and the events arising from this by $\langle t_1^0\rangle, \langle t_2^0\rangle, \ldots$. Regarding (2), let, for all k, $|t_{k+1}^0 - t_k^0| \geqq \varepsilon + s_{\max}$. Thus, if two clock pulse events are synchronous, they are in particular localized within the same time interval $[t_k^0, t_{k+1}^0)$ out of ℓ. Let $t_1 = t_1^0$. We can now define a discrete time scale by a totally ordered set of instants which is bounded from below:

$$\vec{\ell} = (\{t_k^0\}_{k\in\mathbf{N}}, \leqq, t_1^0).$$

Now, when saying that time passes by in steps $t_1^0 \to t_2^0 \to \ldots$, we can assume that each time a time step has elapsed Z has once changed its global state. Because of the synchronity of t_k^α, for all α, we could instead consider $\{t_k^\alpha\}_{k\in\mathbf{N}}$ as set of instants, for any α; indeed, the simultaneous consideration of all t_k^α, $\alpha\in\mathbf{Z}^2$, given by the equivalence relation of synchronity, leads to a globally applicable discrete time concept.

We now return to the questions formulated in the first paragraph and firstly look at the necessity of synchronity for adequate execution of a simultaneous cellular calculus.

In a synchronous cellular automaton, at any individual global transition there takes place only a *local* information processing at each cell, namely, the state of a cell is requested by its *finite* number of neighbouring cells for their computation of a successor cell state, and vice versa. Only by way of a state changing, can inputs to the cell effect its output which will, however, be only requested by the neighbours at the next global transition. (We agree that cells are Moore-type automata; if considering non-deterministic rule sets Σ, this case can easily be generalized, see [8].)

Hence, the effect of the information processing of an individual cell during a global transition is restricted to the region of neighbourhood of the cell. Thus, for the correct execution of the state transitions of each cell, it is only necessary that every two neighbouring cells work synchronously. This leads us to the following definition.

Definition 3. A set of clock-pulsed cells is said to be *locally synchronously working*, if for any given clock signal T_k, the pairwise difference in time of k-th clock pulses of *neighbouring* cells is at most ε s.t.

(i) the correct execution of local transitions is not affected,
(ii) ε is not exceeded after an arbitrary number of clock steps.

Obviously, synchronous working includes locally synchronous working, the opposite of which is not true since, for example, three adjacent cells working locally synchronously allow the clock pulses of the outer cells to differ in time by 2ε. Thus locally synchronous working is a weaker concept than synchronous working of a set of cells, and note that a reference to a boundation of the number of cells is not required in this case.

As it is easy to see, in a cellular automaton Z working locally synchronously the same successor cell states are (or, in case of an indeterministic rule set, could be) generated as when working synchronously. That means, if it is assumed that all cells have had one but not more clock pulses and have finished execution of the induced state transition, in each case the same result would (resp. could) be obtained, namely, the result of one deduction step in the basing cellular calculus.

To get an adequate execution of a simultaneous cellular calculus by way of a locally synchronous performance, it must be guaranteed that by each command to the automaton to execute one deduction step each cell will perform, at some time, one and only one corresponding state transition whereby neighbouring cells do this synchronously. Again, this gives rise to an *organization problem:* to *achieve* such a locally synchronous performance. It appears that two subproblems are to be solved:

(P1) How to perform the adequate execution of one individual deduction step?

(P2) If (P1) is solved, how to proceed on sequences of deduction steps?

3. The concept of *T*-net

For the solution of these problems we now introduce an organization scheme called *T-net.* A *T*-net is a device to be added to a cellular automaton to effect an appropriate distribution of clock signals such that a locally synchronous working can be performed. It consists of uniform type components which are thought to be integrated each to one automaton cell and hence allow the building-up of the automaton simply by the proper arranging of the cells and interconnecting them neighbour-to-neighbour. Thus an automaton in realization can be extended, *including* the organization scheme, by simply subjoining the required number of cells.

Signal transmission in a *T*-net will intentionally be considered as a propagation in a plane lattice such that each lattice point will receive exactly one signal offspring. Since in a physical device a loss of energy has to be supposed it is assumed that a signal regeneration occurs in the *T*-net components. The delay arising between the appearance of the signal (as a clock pulse) at a given cell and its neighbours is considered as being small enough to allow proper working of the cells, according to definition 3.

A *T*-net component is an automata network able to handle a number of clock pulses running in parallel. For the description of this, first a special type of automaton is needed.

An *asynchronous parallel automaton* (APA) A is a system $A=(S_A, I_A, O_A, R_A)$ of pairwise disjoint finite sets S_A, I_A, and O_A, and a subset R_A of $(S_A\times\not{p}(I_A))\times\times(S_A\times\not{p}(O_A))$ where $\not{p}(M)$ denotes the set of all subsets of a given set M. The elements of S_A, I_A, and O_A are called *states, inputs* resp. *outputs of A*. R_A is called the *transition relation of A*.

This concept was introduced by Priese [7]. It allows the description of the behaviour of automata under the simultaneous occurrence of several inputs or outputs, indeterminacy, and, in addition, the feasibility of state transitions independent of inputs or outputs. Concerning the sets I_A and O_A, note that, instead of considering distinct input and output signals on *one* channel, it is here assumed that one type of signals appears at *distinct* channels. Thus we shall call the elements of I_A and O_A input resp. output *places* or likewise *entrances* resp. *exits*. Since the sets are disjoint no confusion will arise. An APA can hence be named a *directed* automaton.

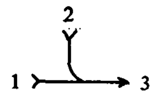
An APA network, shortly *net*, is simply the result of any junction of several APA in such a way that in no case is one output place connected to more than one input place, and vice versa. Input and output places remaining inconnected in such a process will be called *input* resp. *output places of the net*. Again, the junction of nets in the indicated way gives a net. We will imagine in the sequel that signals are very short pulses distinct from a quiescent state which are "running along wires through the net".

A *T-net component* is figured out of four types of APA called K, A, P, I. The elements of their relations are given by transitions

$$(s, \{i_1, \ldots, i_j\}) \to (s', \{\sigma_1, \ldots, \sigma_k\})$$

to be read as follows: "in state s under input at places $i_1, \ldots, i_j$ go over to state s' under output at $\sigma_1, \ldots, \sigma_k$".

The module K gets the symbol



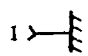
and is defined by

$$K := (\{0\}, \{1, 2\}, \{3\}, R_K)$$

$$R_K := \{(0, \{1\}) \to (0, \{3\}), \quad (0, \{2\}) \to (0, \{3\}), \quad (0, \{1, 2\}) \to (0, \{3\})\}$$

which means that signal inputs appearing at place 1 or 2 will come out at place 3; if two inputs synchronously appear at 1 and 2 a "united" output signal will appear at place 3.

The module A gets the symbol

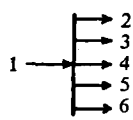


and is defined by

$$A := (\{0,\} \{1\}, \emptyset, \{(0, \{1\}) \to (0, \emptyset)\}).$$

It describes the total absorbing of signals.

The module P gets the symbol

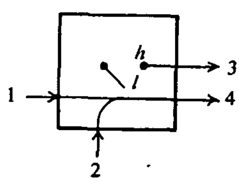


and is defined by

$$P := (\{0\}, \{1\}, \{2, 3, 4, 5, 6\}, \{(0, \{1\}) \to (0, \{2, 3, 4, 5, 6\})\}).$$

A signal input at 1 generates five parallel outputs.

The behaviour of the following "pulse injecting" module I depends on its internal state. I gets the symbol



and is defined by

$$I := (\{l, h, \uparrow, \downarrow\}, \{1, 2\}, \{3, 4\}, R_I)$$

$$R_I := \{(l, \{1\}) \to (\uparrow, \{4\}), (\uparrow, \{1\}) \to (\uparrow, \{4\}), (\uparrow, \emptyset) \to (h, \{3\}),$$
$$(\uparrow, \{1\}) \to (h, \{3, 4\}), (h, \{1\}) \to (h, \{4\}), (h, \{2\}) \to (\downarrow, \{4\}), (\downarrow, \emptyset) \to (l, \emptyset)\}$$

I has two stable states, l and h, and two unstable states, $\uparrow$ and $\downarrow$. At the moment when I reaches state h ("high") a newly created signal is put out at 3; causal for a switching from l to h, and hence for this output, is an input signal reaching the module in state l. For the duration of this switching a "transition state", denoted by $\uparrow$ (resp. $\downarrow$ for the switching from h to l), is considered[1]; in the relational description it is expressed that h can be reached, not influenced by an occurence of further inputs at 1. State-input combinations not listed above will not appear in the nets considered in the sequel.

The indeterministic description of the I-module contains the stated behaviour only in principle in the first instance. To make sure that it will actually happen within a given time period we fix in addition layers for the *duration of switching*, denoted by the delay that will occur between the initiation by a matching input and the moment of the occupation of the new state:

(i) from l to h: $\underline{v}$ $(0 < \underline{v}_{\min} \leqq \underline{v} \leqq \underline{v}_{\max} < \infty)$

(ii) from h to l: $\bar{v}$ $(0 < \bar{v}_{\min} \leqq \bar{v} \leqq \bar{v}_{\max} < \infty)$.

In the case under consideration (two-dimensional cellular space with von Neumann neighbourhood) a *T-net component* is compounded from three K-modules and always one module I, A, and P, which are interconnected to a net as indicated in Fig. 1. Such a net has five input places and five output places: always four *external* places for the reception and emission of clock signals and one *internal* input resp. output place for the junction of always one *T-net* component and one cell. The signal transmission channels will be called *wires* for short. A heuristic description of the functioning of a T-net component follows now.

Consider a T-net component with no signals running on any wire, in the starting position, and with the I-module being in state l. Signals appearing at the external input places of the net pass the K-gates and reach place 1 of the I-module as a *sequence of signals,* the number of which may have diminished by the possible case of synchronous checking into the two input places of a K-module. The signals leave the I-module at 4 and become absorbed at the A-module. Only the first signal of the sequence initiates a switching of I from the stable state l to the stable state h whereby, after the delay $\underline{v}$, a (newly created) signal occurs at place 3. This reaches the P-module whereby five signals run in parallel:

(i) to all external output places of the T-net component

(ii) to the internal output place CLOCK

and leave the net. Incoming signals are thus separated by the I-module: When I is in state l only the first signal of a sequence is able to influence the net. Only when a

[1] When assuming a *duration* is going along with a switching process it is possible, as in the considered case of an APA, that during a switching further signal inputs occur. It is thus necessary, for a complete description, to take account of such an unstable transition state.

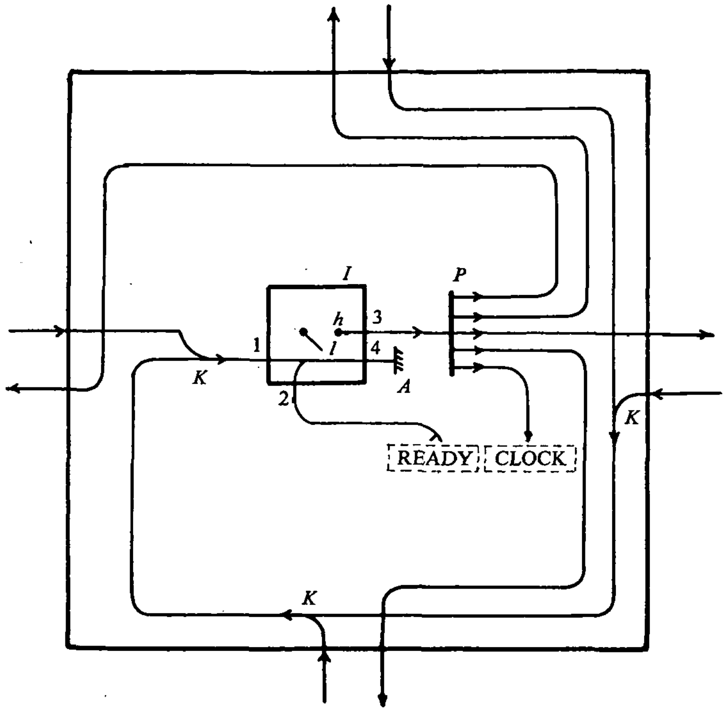


Fig. 1

transition of I back to state l is initiated and fullfilled, by way of a signal incoming at the internal READY entrance, will a signal reaching place 1 of the I-module cause a new signal output at 3.

In respect of the internal exit CLOCK that means: Signals coming from outside into a T-net component can effect a signal output at CLOCK *only once in a certain time period.* This period contains in particular the duration of the event lying between a signal output to CLOCK and a signal input from READY.

Concerning the total time period taken by a process in a T-net component the delay v will be significant which appears between the earliest input of a signal, causally for an output at place 3 of the I-module, and the resulting output of five parallel signals at the exits of the T-net component ($\underline{v}<v$). Considering a number of T-net components, as is done in the following, it is assumed that the following condition holds:

$$0 < v_{\min} \leqq v \leqq v_{\max} < \infty \tag{3}$$

However, concerning repetitions of the described process, we focus attention throughout this paper on the case that

component delays do not vary in time (4)

It will be subject of a future paper [9] to examine the implications given be the assumption of time-variant component delays.

We call *T-net* an APA net originating when a set of copies of the described *T*-net component is compounded at neighbouring integer lattice points of the Euclidian plane and interconnected, in a canonical way, by identifying corresponding output and input places. With respect to its connection scheme and its components a *T*-net is thus *homogeneously* compounded and can, assuming that it consists of a finite number of components, be *boundlessly* extended by repetition of the same process: attaching another *T*-net component at a free lattice point adjacent to the border of the hitherto existing net; admitted are rotations by $k \cdot 90°$, $k \in \{1, 2, 3\}$, in the plane. A section of a *T*-net is shown in Fig. 2.

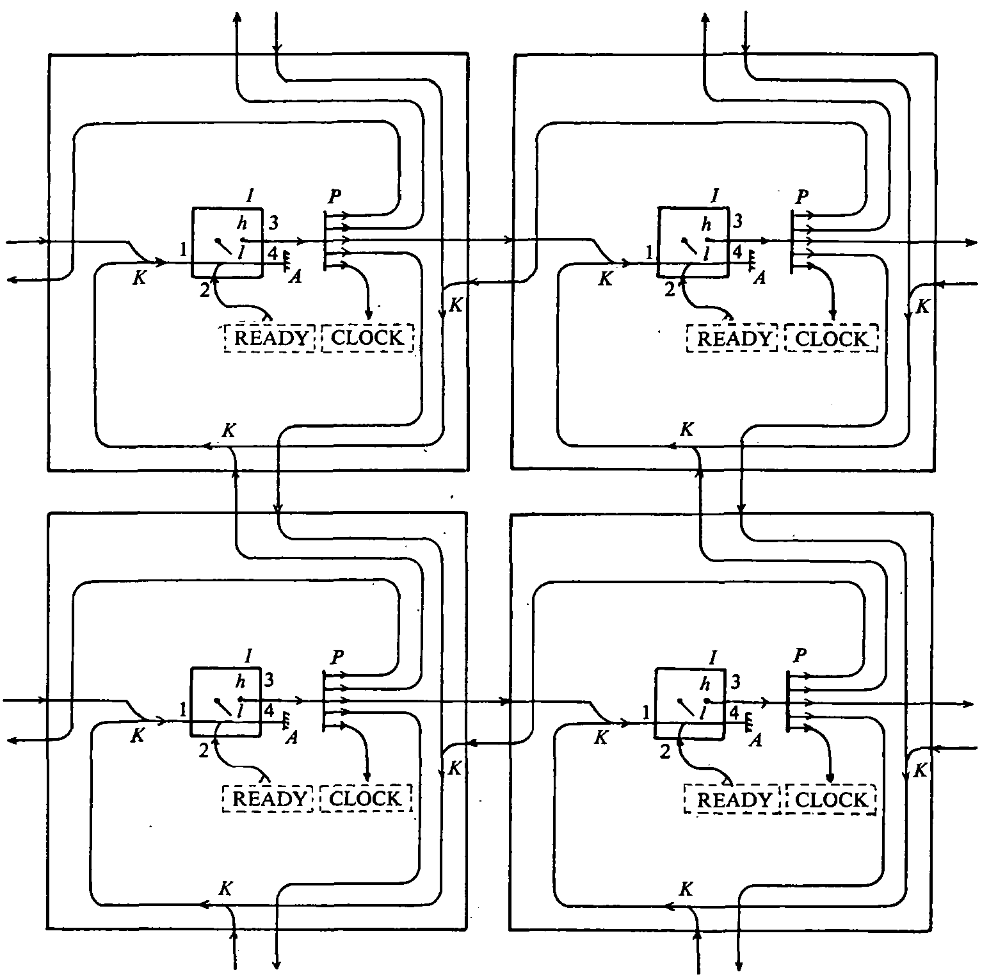


Fig. 2

This completes the description of the construction of a T-net. As pointed out, the concept of APA net here is extended by a time concept to be applied locally as will be worked out later. This gives the possibility of referring to the durations of subprocesses in the T-net. To integrate, for the moment, the starting and ending points of such processes in a T-net component into the time scale t (to be thought of as an observation time), we introduce the following notations. For cells α which contain a T-net component we say that:

at time t, α *is in T-state* s iff the I-module of α is in state s at that time, $s \in \{l, \uparrow, h, \downarrow\}$;

at time t, α *has a T-output* iff at one or several of the external exits of the T-net component a signal is appearing at that time;

at time t, α *has a T-input* iff at one or several of the external entrances of the T-net component a signal is appearing at that time;

at time t, α *has a causal T-input* iff α has a T-input at that time and the earliest signal subsequently appearing at place 1 of the I-module happens to reach the I-module being in state l.

A causal T-input will thus give rise to the switching from l to h whereby, in particular, at a time t' $(t' > t)$ a signal will appear at the CLOCK exit; as stated above, the delay occuring between t and t' is v.

In the following, the interplay between the T-net components and the cells of a cellular automaton will be considered. A clock pulse signal, leaving a certain T-net component by the CLOCK exit, will initiate a state transition of the cell which, after the delay s (see (1)), will terminate with a "ready"-signal entering the T-net component by the READY entrance whereby the I-module is re-enabled, i.e. switched into state l. To distinguish some characteristic steps in such a process we mark by indices the corresponding instants. For cell α, denote by

$\underline{t}_k^\alpha$ the instant of the k-th causal T-input at α

t_k^α the instant of the subsequent k-th CLOCK output at α

$\underline{r}_k^\alpha$ the instant of the k-th READY input at α

r_k^α the instant of the subsequent k-th re-enabling of the I-module of α.

The events $\langle \underline{t}_k^\alpha \rangle$, $\langle t_k^\alpha \rangle$, $\langle \underline{r}_k^\alpha \rangle$, and $\langle r_k^\alpha \rangle$ occurring at those instants are ordered in time in the manner shown; each is represented by the appearance of a signal at a certain section place in the net, except for the last event in such a chain, $\langle r_k^\alpha \rangle$, which is represented by the transition from T-state $\downarrow$ to T-state l. Note that a causal T-input could be represented by several signals entering the T-net component at *different* times (and places) but resulting in one signal appearing at place 1 of the I-module. Thus we fix, in addition, $\langle \underline{t}_k^\alpha \rangle$ to be represented by the *earliest* signal in question.

Together with the CLOCK signal, appearing with an event $\langle \underline{t}_k^\alpha \rangle$, four signals appear, in parallel, at the external output places of the T-net component, possibly but not necessarily at the same time, say at the instants $\bar{t}_{1,k}^\alpha, \bar{t}_{2,k}^\alpha, \bar{t}_{3,k}^\alpha, \bar{t}_{4,k}^\alpha$. Since the construction of the T-net is done by identifying corresponding external input/output places of neighbouring T-net components, each of the associated events $\langle \bar{t}_{1,k}^\alpha \rangle, \ldots$ represents a T-input for a neighbour cell of α. It is then possible that such a T-output, say $\langle \bar{t}_{1,k}^\alpha \rangle$, is a *causal* T-input for some neighbour cell β of α: $\langle \underline{t}_l^\beta \rangle$. This *identity* of such two events is denoted by $\langle \underline{t}_l^\alpha \rangle \equiv \langle \bar{t}_{1,k}^\alpha \rangle$, i.e. in particular $\underline{t}_l^\alpha = \bar{t}_{1,k}^\alpha$.

Is is assumed in the sequel that, in each case, the appearance of a CLOCK signal is the event latest in that parallelism, i.e.

$$t_k^\alpha = \max \{t_k^\alpha, \bar{t}_{1,k}^\alpha, \dots, \bar{t}_{4,k}^\alpha\} \tag{5}$$

and, in addition,

$$\max_i |t_k^\alpha - \bar{t}_{i,k}^\alpha| < v_{\min} \tag{6}$$

If no confusion arises the k-index will be suppressed in the following. To distinguish the notations for a cellular calculus and the corresponding device we write Z in the first case and in the second case $\hat{Z}$.

4. The structure ($\hat{Z}$, T)

By use of the concept of T-net we want to show how to perform adequate execution of deduction steps in simultaneous cellular calculi.

Assume a cellular automaton $\hat{Z}=(\mathbf{Z}^2, N, Q, \Sigma)$ with von Neumann neighbourhood is given, executing such a calculus in case all cells undergo synchronous state transitions. A state transition of a cell α will be initiated by a clock pulse signal $\langle t^\alpha \rangle$, and a "ready"-signal shall appear at the moment of its termination. Following Arbib [1, p. 375] we will assume here that a state transition of cell α is completed in three phases (see Fig. 3):

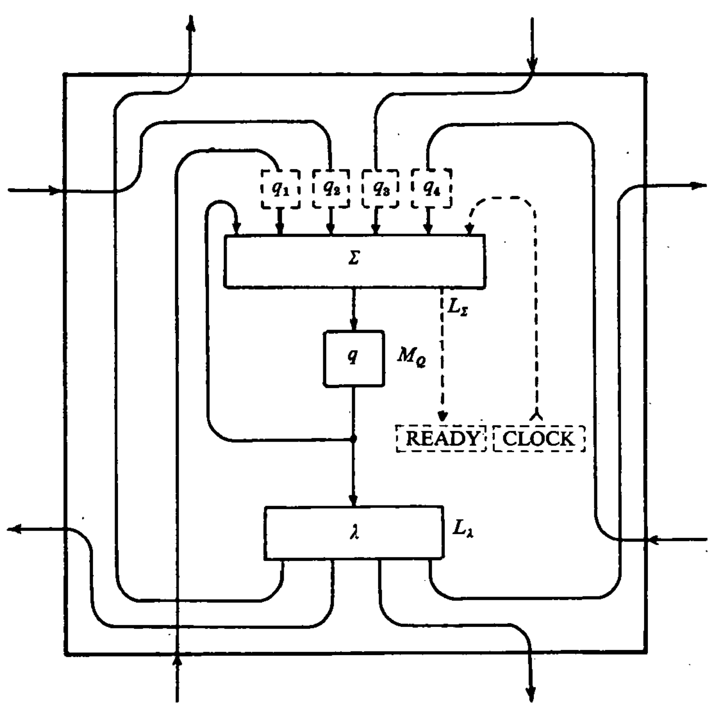


Fig. 3

(i) 'POOL': The relevant neighbourhood information given at time t^α: q_1, q_2, q_3, q_4, is pooled in suitable registers;

(ii) 'EXECUTE': Based on these and its own state q an applicable rule is executed, by a logic L_Σ;

(iii) 'MOVE': The resulting successor state symbol q' is moved to a memory register, M_Q, whereat the predecessing state symbol q is deleted.

From this it follows that the "old" state symbol of the cell is displayed to its neighbours, by an output logic L_λ, till the end of phase (ii), i.e., always from the end of phase (iii) of a state transition to the end of phase (ii) of the *next* transition, a cell is in a *well-defined* state.

Consider now identical copies of a cell of this type and the T-net component, as introduced above, being one-to-one compounded by identification of the internal places CLOCK and READY, and then, as usual, being arranged at the lattice points of the plane. While denoting the cellular automaton, bare of the T-net components, by $\hat{Z}$, the new structure originating from this process is named $(\hat{Z}, T)$. It is of *homogeneous* compound, as sketched in Fig. 4 (dotted lines indicate the junctions of T-net components).

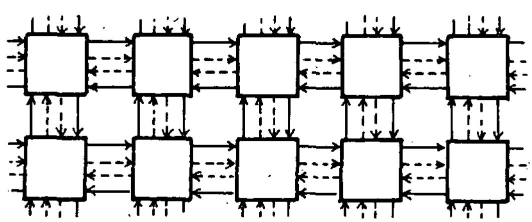


Fig. 4

The interplay of a T-net component and the associated cell is now explained. From a causal T-input $\langle \underline{t}^\alpha \rangle$ at a cell α being in T-state l a clock pulse signal output $\langle t^\alpha \rangle$ will result which, by way of the CLOCK junction, effects the initiation and execution of a state transition of α, whereby the T-net component is in T-state h. On the termination of the state transition, by way of the READY junction, a READY input $\langle \underline{r}^\alpha \rangle$ effects the removing to the T-state l, occurring as event $\langle r^\alpha \rangle$, whereat the cell is enabled for a new causal T-input.

In a schematic way, taking into account the durations s, v, and $\bar{v}$ of subprocesses, as introduced earlier, the mode of operation of a cell in the $(\hat{Z}, T)$-structure is outlined below in Fig. 5.

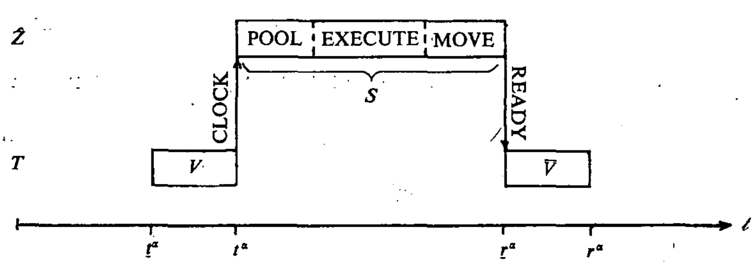


Fig. 5

On considering the $\hat{Z}$-structure, we see that the operation mode of a cell is characterized by two time values:

a) the bounded switching duration[2] s

b) ε, the maximum amount admissible for clock pulses to differ in time, according to definition 3.

For illustration of this, Fig. 6 is given where α and β are neighbouring cells; in phase (i) cell β sees α in a well-defined state, and vice versa.

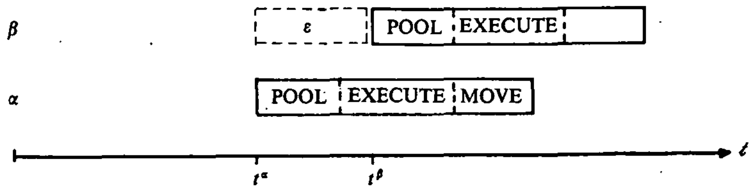


Fig. 6

On considering the T-net structure, we note that the values characterizing a T-net component are

c) the bounded delay v occurring between a causal T-input and the clock pulse (i.e. the CLOCK output) effected by this;

d) the bounded delay $\bar{v}$ which occurs with the re-enabling of the T-net component, i.e. the switching from T-state h to l.

These four time values s, ε, v, and $\bar{v}$ have to be considered as a whole in the $(\hat{Z}, T)$-structure. For the following, it is required that

$$v_{max} \leqq \varepsilon \tag{7}$$

which means that the maximum delay occurring in a T-net component does not exceed the admissable difference of the clock pulse instants of neighbouring cells, and

$$v_{max} \leqq s_{min} + \bar{v}_{min} \tag{8}$$

which is in particular satisfied by (7).

We now investigate the distribution of clock signals in a given polyautomaton $(\hat{Z}, T)$ underlying the assumptions stated above, with a given initial configuration of $\hat{Z}$, i.e. all cells are in an initial state which is stored in M_Q. It is assumed that the T-net is clear of signals, and that all cells are in T-state l.

At first, $(\hat{Z}, T)$ will be considered under *input of a single clock signal*. If an *adequate* execution of the corresponding simultaneous calculus Z is expected, it must be shown that, from this single signal, *one* deduction step in the calculus is put into execution (P1).

At time $t_1 = 0$ (referring to the origin) a solitary T-input shall be given to the origin cell, from an external clock. This T-input is causal for a clock pulse occurring after a delay v by which a state transition of the origin cell is initiated, while all

[2] In the non-deterministic case it is assumed that an applicable rule is executed within that duration.

neighbours (being in the T-state l) have previously had a T-input, which, at each cell, again gives rise to a clock pulse etc. This process can be interpreted as the propagation of a clock signal, introduced into the T-net at the origin, by signal offsprings which appear as an individual clock pulse for each cell α, at a certain time t^α. Whether a cell at which has already occurred a clock pulse event can have a repeated *causal* T-input, is to be elucidated.

By way of the T-net components an original signal is thus propagated in all directions of the plane, whereby in the cells passed, state transitions are initiated. At individual cells different delay, lying between v_{min} and v_{max}, can thereby appear (3). The actual course of a signal frontage (of first occurrence of a clock pulse) at a given time t is located within bounds determined by v_{min}, v_{max}, and t, as follows. If defining the *distance* between two cells α and β with coordinates (a_1, a_2) resp. (b_1, b_2) by

$$d(\alpha, \beta) = |a_1-b_1|+|a_2-b_2|$$

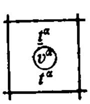
then the maximum distance from the origin 0 of cells α which had a clock pulse $\langle t^\alpha \rangle$ until time $t(t^\alpha \leqq t)$ lies between

$$d^t_{max}(\alpha, 0) = \left[\frac{t}{v_{min}}\right]-1 \quad (t \geqq v_{min})$$

$$d^t_{min}(\alpha, 0) = \left[\frac{t}{v_{max}}\right]-1 \quad (t \geqq v_{max})$$

where [ ] denotes the greatest integer z: $z<\frac{t}{v}$. In the following example an actual signal distribution possible under the above assumptions will be demonstrated.

Consider the T-net of a $(\hat{Z}, T)$-structure for which it is assumed that $v_{min}=1$ and $v_{max}=3$. A causal T-input $\langle \underline{t}^\alpha \rangle$ for a cell α is given by the earliest T-output $\langle \bar{t}^\beta_v \rangle$ of a neighbour β of α. We assume here, for simplicity, that, for all β, $t^\beta=\bar{t}^\beta_1=\dots$ $\dots=\bar{t}^\beta_4$. Thus we have, in the above case, $\bar{t}^\beta_v=t^\beta=\underline{t}^\alpha$. In Fig. 7 a section of the cellular plane is shown where the numbers in a field for a cell α represent the following values:



instant of a causal T-input at α delay occuring

betveen $\underline{t}^\alpha$ and t^α instant of the clock pulse at α

From a T-input at time 0 at the origin, and by occurrence of the specified delays, the process displayed in Fig. 7 will arise. The sequence of the appearances of signal offsprings, represented by clock pulses at the individual cells, can be read along the clock pulse instants.

We denote by F_t the set of all cells that have a clock pulse up to time t; $F_t = \{\alpha | \exists t^\alpha (t^\alpha \leqq t \& \langle t^\alpha \rangle)\}$. From the way of signal distribution it is clear that, for any t, F_t — conceived as a subset of the cellular plane — is connected and that, for $t_1, t_2 \in \mathcal{t}$, $t_1 < t_2$ implies $F_{t_1} \subseteq F_{t_2}$. From the above-stated example, secondly, it follows immediately that F_t can be of a "genus" greater than zero (consider F_7 in the example).

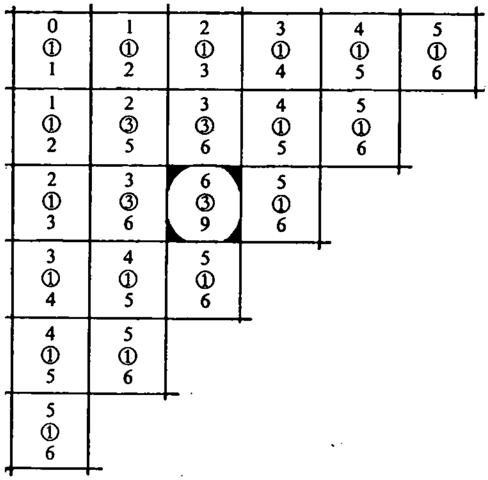


Fig. 7

5. Locally synchronous performance

We now proceed to state the main result of this paper.

Theorem. For a given simultaneous cellular calculus $Z=(\mathbf{Z}^2, N, Q, \Sigma)$ with von Neumann neighbourhood, let $(\hat{Z}, T)$ be the corresponding polyautomaton, and let all cells of $(\hat{Z}, T)$ be in T-state l. At time 0 of the observation time $\not{t}=(\mathbf{R}, \leqq, 0)$ a clock signal is introduced as T-input into the origin cell. Then it follows that:

(a) For each cell of $(\hat{Z}, T)$ there exists an instant t where it has a clock pulse.

(b) The earliest clock pulses (within $\not{t}$) of every two neighbouring cells of $(\hat{Z}, T)$ differ in time at most by $v_{\max}$.

(c) No cell of $(\hat{Z}, T)$ has more than one clock pulse.

Proof. (a) From the foregoing discussion it follows that a cell α with coordinates (x, y) has a clock pulse at the latest at time $t=(|x|+|y|+1)\cdot v_{\max}$, where $0<v_{\max}<\infty$.

(b) Sketch: Of any two neighbouring cells, each of which still has not had a clock pulse, the one which has a clock pulse first gives a T-input to the other which will be causal (and thus will lead, within the delay $v_{\max}$, to a clock pulse of the second cell), unless this one had an earlier causal T-input from elsewhere (which would then lead to an earlier occurring clock pulse). The clock pulses differ in time then at most by $v_{\max}$.

Let now α and β be any two neighbouring cells in T-state l, none of which has already had a clock pulse. From (a) there exists, for each cell, a clock pulse instant.

1st case. α and β have clock pulses at the same time. Then the difference is $0<v_{\max}$.

2nd case. Without loss of generality, let α have a clock pulse first, at time t^α. Thus, by (5), at the latest at the same time as $\langle t^\alpha\rangle$ a T-output of α will appear which is a T-input for β. In the sequel, we denote such a T-output of α for β by $\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$. According to (6), $|t^{\alpha} - \bar{t}^{\alpha}_{j(\beta)}| < v_{\min}$.

Hence, β has a clock pulse, at a time t^{β} $(t^{\alpha} < t^{\beta})$, for which either (i) the T-output $\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$, linked with the event $\langle t^{\alpha} \rangle$, is causal, i.e. $\langle \underline{t}^{\beta} \rangle \equiv \langle \bar{t}^{\alpha}_{j(\beta)} \rangle$, or (ii) a T-output of another neighbour cell of β, having occurred earlier than $\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$, or (iii) several synchronous T-outputs of neighbours of β other than α having occurred earlier than $\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$, or (iv) $\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$ *and* one ore more T-outputs of neighbours of β appearing at the same time as $\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$; see the representation of an event $\langle \underline{t}^{\alpha}_{k} \rangle$ as pointed out earlier.

The clock pulse of β will occur at the latest after the delay $v_{\max}$ following a causal T-input, thus $t^{\beta} \leqq \underline{t}^{\beta} + v_{\max}$.

Cases (i) and (iv) apply to: $\underline{t}^{\beta} = \bar{t}^{\alpha}_{j(\beta)} (\bar{t}^{\alpha}_{j(\beta)} \leqq t^{\alpha})$,
then $\underline{t}^{\beta} + v_{\max} \leqq t^{\alpha} + v_{\max}$,
and thus $t^{\beta} = t^{\alpha} + v_{\max}$,
equivalent to $t^{\beta} - t^{\alpha} \leqq v_{\max}$.
Hence, by $t^{\alpha} < t^{\beta}$, $|t^{\beta} - t^{\alpha}| \leqq v_{\max}$.

Cases (ii) and (iii) apply to: $\underline{t}^{\beta} < \bar{t}^{\alpha}_{j(\beta)} (\bar{t}^{\alpha}_{j(\beta)} \leqq t^{\alpha})$.
Here we have $\underline{t}^{\beta} + v_{\max} < t^{\alpha} + v_{\max}$
and it follows analogously: $|t^{\beta} - t^{\alpha}| < v_{\max}$.

Hence, in each case the earliest clock pulse instants of every two neighbouring cells differ at most by $v_{\max}$, what had to be shown for (b).

Before proving (c) we formulate with the results obtained up to now the *rules for the distribution of clock signals by a T-net:*

(T0) $\exists t^{0} (0 + v_{\min} \leqq t^{0} \leqq 0 + v_{\max} \;\&\; \langle t^{0} \rangle)$

(T1) For $t^{\alpha} > t^{0}$:

$$\langle t^{\alpha} \rangle \Rightarrow \exists \beta (\beta \in N(\alpha) \setminus \{\alpha\} \;\&$$

$$\exists t^{\beta} (t^{\beta} < t^{\alpha} \;\&\; \langle t^{\beta} \rangle \;\&\; 0 < |t^{\alpha} - t^{\beta}| \leqq v_{\max} \;\&\; \langle \bar{t}^{\beta}_{j(\alpha)} \rangle \equiv \langle \underline{t}^{\alpha} \rangle))$$

(T2) $\langle t^{\alpha} \rangle \Rightarrow \forall \beta (\beta \in N(\alpha) \setminus \{\alpha\} \Rightarrow \exists j(\beta) (j(\beta) \in \{1, \ldots, 4\} \;\&$

$\langle \bar{t}^{\alpha}_{j(\beta)} \rangle$ is T-input for β & (β in T-state l at $\bar{t}^{\alpha}_{j(\beta)} \Rightarrow$

$$\exists t^{\beta} (t^{\alpha} < t^{\beta} \;\&\; \langle t^{\beta} \rangle \;\&\; 0 < |t^{\alpha} - t^{\beta}| \leqq v_{\max}))))$$

(T3) Only by (T0), (T1), and (T2) can there occur clock pulse events.

(T0) means that the origin cell 0 has, at time t^{0}, a clock pulse event (by hypothesis of the theorem that there is given an external T-input to the origin at time 0, to be interpreted as the input of a clock signal).

(T1) concerns the *cause* of the appearance of a clock signal at a given cell α and means that, for the occurrence of a clock pulse event at α (with exception of $\alpha = 0$), a foregoing (at most by $v_{\max}$) clock pulse event of a neighbour cell different from α is necessary which is linked with a causal T-input for α.

(T2) concerns the propagation of a clock signal which has appeared at α, and means that, succeeding a clock pulse event $\langle t^{\alpha} \rangle$ there occurs, at the latest by time $v_{\max}$, a clock pulse event $\langle t^{\beta} \rangle$ at all cells β being in the T-state l at time $\bar{t}^{\alpha}_{j(\beta)}$ (whereby not necessarily $\langle \bar{t}^{\alpha}_{j(\beta)}\rangle \equiv \langle \underline{t}^{\beta}\rangle$, see (ii) in the second case above, it being allowed that $\underline{t}^{\beta} < \bar{t}^{\alpha}_{j(\beta)}$).

By way of the construction of T-net and the assumptions made, only in accordance with (T0), (T1), and (T2) can there occur clock pulses. The rules are therefore completed by (T3).

We now proceed to prove (c). For the occurrence of a clock pulse at a cell α a foregoing causal T-input is necessary which, by (T1), appears as T-output of a distinct neighbour cell of α (except for $\langle t^0\rangle$). It will be shown for (c) that no cell having had a first clock pulse can receive a new causal T-input.

Consider, for arbitrary $t \in \mathcal{t}$, the set G_t of cells which have *exactly* one clock pulse until time t:

$$G_t = \{\alpha \mid \exists t^{\alpha}(t^{\alpha} \leqq t \,\&\, \langle t^{\alpha}\rangle \,\&\, \neg\, \exists t_1^{\alpha}(t_1^{\alpha} \leqq t \,\&\, t_1^{\alpha} \neq t^{\alpha} \,\&\, \langle t_1^{\alpha}\rangle))\}.$$

First we show:

Lemma 1. No cell β having a first clock pulse at time t^{β} can thereby yield a causal T-input for a cell $\alpha \in G_{t^{\beta}}$.

For proving this we distinguish three cases, for any $\beta \in Z^2$ with the first clock pulse at t^{β}.

1st case. $\alpha \in G_{t^{\beta}}$ & $\alpha \notin N(\beta)$
No T-input for α is linked with $\langle t^{\beta}\rangle$, see (T2), (T3).

2nd case. $\alpha \in G_{t^{\beta}}$ & $\alpha = \beta$
The case that β yields a T-input for itself is excluded, see (T2), (T3), particularly in case $\beta = \alpha = 0$.

3rd case. $\alpha \in G_{t^{\beta}}$ & $\alpha \in N(\beta) \setminus \{\beta\}$
By (T2) at the latest with $\langle t^{\beta}\rangle$ a T-input for α has appeared, as T-output of β: $\langle \bar{t}^{\beta}_{j(\alpha)}\rangle$. As $\alpha \in G_{t^{\beta}}$, there exists exactly one t^{α}: $t^{\alpha} \leqq t$ & $\langle t^{\alpha}\rangle$. We show that $\langle \bar{t}^{\beta}_{j(\alpha)}\rangle$ is neither causal (i) for $\langle t^{\alpha}\rangle$ nor (ii) for clock pulse events possibly occurring later.

(i) By $t^{\alpha} \leqq t^{\beta}$: $\quad t^{\alpha} - v_{\min} \leqq t^{\beta} - v_{\min}$
from (6) it follows $\quad t^{\beta} - v_{\min} < \bar{t}^{\beta}_{j(\alpha)}$
and from (3) $\quad t^{\alpha} - v_{\max} \leqq \underline{t}^{\alpha} \leqq t^{\alpha} - v_{\min}$
Hence $\quad \underline{t}^{\alpha} < \bar{t}^{\beta}_{j(\alpha)}$
and thus $\langle \bar{t}^{\beta}_{j(\alpha)}\rangle \not\equiv \langle \underline{t}^{\alpha}\rangle$, i.e. $\langle \bar{t}^{\beta}_{j(\alpha)}\rangle$ is not causal for $\langle t^{\alpha}\rangle$.

Remark. A clock pulse $\langle t^{\beta}\rangle$ cannot therefore be causal for a clock pulse $\langle t^{\alpha}\rangle$ occurring either at the same or an earlier time. If conversely $\langle \bar{t}^{\beta}_{j(\alpha)}\rangle \equiv \langle \underline{t}^{\alpha}\rangle$ holds, it follows that $t^{\alpha} - t^{\beta} > 0$. This fact, ensured by (6), is already considered in the above formulation or rules (T1), (T2).

For (ii) we show that $\langle \bar{t}^{\beta}_{j(\alpha)}\rangle$ is not causal for a $\langle t_1^{\alpha}\rangle$, $t^{\alpha} < t_1^{\alpha}$. As assumed, $\langle t^{\alpha}\rangle$ and $\langle t^{\beta}\rangle$ are the earliest clock pulses of the neighbouring cells α and β. From $t^{\alpha} \leqq t^{\beta}$ it follows by (b):

$$0 \leqq t^{\beta} - t^{\alpha} \leqq v_{\max}$$

and upon (8) $\quad 0 \leqq t^{\beta} - t^{\alpha} < s_{\min} + \bar{v}_{\min}$
i.e. $\quad t^{\alpha} \leqq t^{\beta} < t^{\alpha} + s_{\min} + \bar{v}_{\min}$

However, α is *not* in the T-state l up to the re-enabling of its I-module, at the least until time $t^{\alpha} + s_{\min} + \bar{v}_{\min}$ and, hence, at time t^{β} (see Fig. 5); i.e. the T-input for α: $\langle \bar{t}^{\beta}_{j(\alpha)} \rangle$ occurring at the latest with $\langle t^{\beta} \rangle$ is not causal for a new clock pulse event at α. This ends the proof of Lemma 1.

In Lemma 1, it is referred to sets $G_{t^{\beta}}$. Certainly, for all t, $G_t \subseteqq F_t$. However, for certain t, $G_t \subset F_t$ could be, i.e. nothing can be said about whether the Lemma regards, in every case, all cells which had a clock pulse up to time t^{β} generally. It is thus to be shown that no cell of F_t can have more than one causal T-input. By hypothesis of the theorem, only one external T-input is given to the origin cell; i.e. for any cell α only such possibly causal T-inputs need to be considered which appear in the neighbourhood of α as T-outputs, linked with clock pulse events, see (T0), (T1). In the following, we denote by $N(S)$ the *extended neighbourhood set* of a finite subset S of $\mathbf{Z}^2$: $N(S) = \bigcup_{\alpha \in S} N(\alpha)$, where $N(\alpha)$ is the set of neighbours of a single cell α as given above (no confusion will arise by using the same symbol for both).

Lemma 2. For every t: $F_t = G_t$

Proof. We define, within $\mathcal{t}$, special instants $t_1, t_2, t_3, \ldots$ as follows:

$$t_1 = t^0$$

$$t_{n+1} = \min \{t^{\alpha} | t_n < t^{\alpha} \ \& \ \exists \alpha \langle t^{\alpha} \rangle\}$$

Each instant t_n is thus a clock pulse instant of one or several cells in $\mathbf{Z}^2$, and, for all n, $F_{t_n} \subseteqq F_{t_{n+1}}$. The above minimum always exists arising from the following fact: In each case, only a finite number of cells α comes into question for the next clock pulse (within $\mathcal{t}$) caused by the cells β of F_{t_n}, namely, the cells α: $\alpha \in N(F_{t_n})$. Thereby F_{t_n} and thus $N(F_{t_n})$ are finite sets, at each instant, because from each clock pulse of any cell only clock pulse events of its finite number of neighbour cells can be caused, see (T2). So $(t_n)_{n \in \mathbf{N}}$ is well-ordered within $\mathcal{t}$: $t_1 < t_2 < \ldots$, and $(F_{t_n})_{n \in \mathbf{N}}$ is well-ordered concerning '$\subseteqq$'; and the following statement holds:

$$F_{t_1} = \{0\}; \quad \bigcup_n F_{t_n} = \mathbf{Z}^2.$$

We now claim

$$\forall n: \ F_{t_n} = G_{t_n} \quad (n \in \mathbf{N}).$$

Proof by induction on n. Obviously, $F_{t_1} = G_{t_1} = \{0\}$ (base of induction).
Induction hypothesis: Let $F_{t_n} = G_{t_n}$.
Induction step. As $G_{t_{n+1}} \subseteqq F_{t_{n+1}}$, it remains to be shown that

$$F_{t_{n+1}} \setminus G_{t_{n+1}} = \emptyset.$$

Supposing there exists α: $\alpha \in F_{t_{n+1}} \setminus G_{t_{n+1}}$, then α has exactly one clock pulse $\langle t^{\alpha} \rangle$ up to time t_n, for it is $F_{t_n} \subseteqq F_{t_{n+1}}$ and, by induction hypothesis, $F_{t_n} = G_{t_n}$, i.e. $\alpha \in G_{t_n}$; and α has a second clock pulse $\langle t^{\alpha}_1 \rangle$ at time t_{n+1}, for by construction of (t_n) a clock pulse cannot exist between t_n and t_{n+1}. Hence, by (T1),

$$\exists \beta (\beta \in N(\alpha) \setminus \{\alpha\} \ \& \ \exists t^{\beta} (\langle t^{\beta} \rangle \ \& \ t^{\beta} \leqq t_n < t^{\alpha}_1 \ \&$$

$$|t^{\alpha}_1 - t^{\beta}| \leqq v_{\max} \ \& \ \langle \bar{t}^{\beta}_{j(\alpha)} \rangle \equiv \langle t^{\alpha}_1 \rangle)).$$

Thus α must have received a causal T-input before $t_{n+1} (t_{n+1} = t^{\alpha}_1)$, linked with a clock pulse of a neighbour cell β.

1st case. $\langle t^\beta \rangle$ is *not* the first clock pulse of $\beta(t^\beta \leqq t_n)$.
That would mean: $\beta \in F_{t_n} \setminus G_{t_n}$ which is a contradiction of the induction hypothesis.

2nd case. $\langle t^\beta \rangle$ *is* the first clock pulse of $\beta(t^\beta \leqq t_n)$.

(i) $t^\beta = t_n$. From Lemma 1 it follows that no cell of G_{t^β}, $G_{t^\beta} = G_{t_n}$, can get a causal T-input by $\langle t^\beta \rangle$. But from the above assumption it follows that $\langle \bar{t}^\beta_{j(\alpha)} \rangle \equiv \equiv \langle \underline{t}^\alpha_1 \rangle$ & $\alpha \in G_{t_n}$ which is impossible by Lemma 1 and hence results in a contradiction.

(ii) $t^\beta < t_n$. Then $\alpha \in G_{t^\beta}$ which is shown as follows. Between two clock pulses of α, $\langle t^\alpha \rangle$ and $\langle t^\alpha_1 \rangle$, there are located the events $\langle \underline{r}^\alpha \rangle$, $\langle r^\alpha \rangle$, and $\langle \underline{t}^\alpha_1 \rangle$, as was pointed out in the third section. Thus, it is valid for the distance in time of the two clock pulses:

$$|t^\alpha - t^\alpha_1| > s_{\min} + \bar{v}_{\min} + v_{\min}$$

for it must be $r^\alpha < \underline{t}^\alpha_1$; furthermore:

$$|t^\beta - t^\alpha_1| \leqq v_{\max}$$

and by $t^\beta < t^\alpha_1$:

$$t^\alpha_1 - t^\beta \leqq v_{\max}$$

resp.

$$-v_{\max} \leqq t^\beta - t^\alpha_1$$

Thus, by $t^\alpha < t^\alpha_1$:

$$s_{\min} + \bar{v}_{\min} + v_{\min} < t^\alpha_1 - t^\alpha$$

Addition of the last two inequalities yields

$$s_{\min} + \bar{v}_{\min} + v_{\min} - v_{\max} < t^\beta - t^\alpha$$

and by (8):

$$0 < t^\beta - t^\alpha$$

thus $t^\alpha < t^\beta$ whence $\alpha \in G_{t^\beta}$. From Lemma 1 then again a contradiction results. Hence, $F_{t_{n+1}} = G_{t_{n+1}}$. From this, the above assertion is proved: For all n, $F_{t_n} = G_{t_n}$ is valid.

The proof of Lemma 2 is now completed as follows. By construction of (t_n):

$$\forall n \forall t (t_n \leqq t < t_{n+1} \Rightarrow F_t = F_{t_n} \,\&\, G_t = G_{t_n})$$

It holds that: $\bigcup_n [t_n, t_{n+1}) = \{t \mid t \in \mathscr{t} \;\&\; t_1 \leqq t\}$. But there is no clock pulse before t_1, it is $F_t = G_t = \emptyset$ for $t \in [0, t_1)$. Thus, for all $t \in \mathscr{t}$, $F_t = G_t$. This ends the proof of Lemma 2.

Altogether we have now:

$$\bigcup_{t \in \mathscr{t}} G_t = \bigcup_{n \in \mathbf{N}} G_{t_n} = \bigcup_{n \in \mathbf{N}} F_{t_n} = \mathbf{Z}^2$$

i.e. every cell $\alpha \in \mathbf{Z}^2$ has exactly one clock pulse, which concludes the proof of the theorem.

From this, the above stated organization problem (P1) is solved, and it is now to be examined how to proceed on sequences of deduction steps to perform an adequate execution of a simultaneous cellular calculus by a $(\hat{Z}, T)$-structure (P2). Thus, $(\hat{Z}, T)$ will now be considered under *input of a sequence of clock signals.*

As was assumed here (4), the delays appearing in the T-net components do not vary in time (but of course it is allowed that the individual delays of different components vary within $[v_{min}, v_{max}]$). In this case the distribution of any subsequent clock signal will occur in exactly the same manner as in case of the first signal, provided that each cell that receives a signal offspring, generated by a signal T_{k+1} following T_k, has already re-entered T-state l. This is, however, obviously satisfied under the above assumption, if only the input of a successor signal T_{k+1}, given as a T-input $\langle t^0_{k+1}\rangle$ to the origin cell, takes place *after* the k-th re-enabling of the origin cell by the event $\langle r^0_k\rangle$. To get from the event $\langle t^0_k\rangle$ to $\langle r^0_k\rangle$ will take the minimum duration $D_{min} := v_{max} + s_{max} + \tilde{v}_{max}$ (see Fig. 5), so that the above conditions are satisfied if

$$t^0_{k+1} \geqq t^0_k + D_{min} \quad (k = 1, 2, 3, \ldots) \tag{9}$$

In this case, by each clock signal T_k the k-th deduction step in the calculus is put into execution which can be seen from the following facts.

Upon occurrence of a clock pulse $\langle t^\alpha_2\rangle$, in correspondence with a second clock signal T_2, all neighbours of a cell α will have executed their first state transition, including re-entry of T-state l, and be in a welldefined state, keeping ready for a new causal T-input. That means, even though not every cell of $(\hat{Z}, T)$ has executed a state transition at the time of input of T_2, this actually has happened *locally* (within one neighbourhood) wherever a clock pulse event caused by T_2 occurs.

The proof of the theorem relied exclusively on *local arguments* (in each case only neighbouring cells have been considered), so that the hypotheses of the theorem have only to be valid locally, i.e. the statements proven in the theorem for a single clock signal input are implied in the same manner for a second signal input, etc. Therefore, from statement (b) of the theorem, for any k the clock pulse events of neighbouring cells differ in time at most by v_{max} ($k=1, 2, 3, \ldots$). Under these assumptions, hence, the conditions of definition 3 are satisfied, and we get the

Corollary. Provided that the component delays appearing in the T-net are *time-invariant*, it follows that, for given $\hat{Z}$, the set of all cells in structure $(\hat{Z}, T)$ works locally synchronous.

But that means: With each clock signal, $(\hat{Z}, T)$ puts into execution one deduction step in the corresponding simultaneous calculus, in a locally synchronous performance. Thereby, it is no obstacle that, as was shown above, at certain times the region in $\mathbf{Z}^2$ being reached by clock signal offsprings can possibly be of "genus" greater than zero: As far as at time t clock pulse events, generated by T_k but not T_{k+1}, have occurred, and state transitions initiated by this are executed, there is present the partial result of the k-th deduction step in the simultaneous calculus that arises from an initial pattern.

Thus it is proved: The (sufficient) condition of synchronity is not necessary; an adequate execution of cellular calculi in the sense of John von Neumann is possible by way of locally synchronous cellular automata.

6. T-synchronity

The execution of a simultaneous cellular calculus in a structure $(\hat{Z}, T)$ is by no means attained through synchronity of the local substitutions, referring to a global or discrete time scale as introduced in the second section. This seems to make impossible the view of a global state of a locally synchronous cellular automaton: Compared with the global time t referring to the origin, $(\hat{Z}, T)$ does not perform the computation of a successor "configuration" at once — precisely: between two subsequent instants of the discrete time scale $\vec{t}$ — but instead does this successively from the inner to the outer region of $\mathbf{Z}^2$. However, concurrently with the foregoing computation, the computation of another successor can be started after short delay $D \geqq D_{\min}$ as given by (9), *before* the first computation is completed.

Thus at no time (of t) a global state, as the total result of a deduction step, appears in a structure $(\hat{Z}, T)$. Note that the definition of a configuration, as well as of a global transformation, was based on a concept of synchronity, allowing the simultaneous consideration of all cells of a cellular automaton at "global" instants, more precisely, at simultaneously considered local instants; see the discussion of these matters in the second section.

But, as was shown by the main result, a simultaneous cellular calculus is actually put into execution in structure $(\hat{Z}, T)$. Furthermore, under the above assumption of time-invariant component delays in the T-net, the execution of a next deduction step can, at any given case, be initialized after a delay nearly as small as in the case of an ideal synchronous cellular automaton which is seen from the following.

Consider a cellular automaton $\hat{Z}$ working synchronously, in a hypothetic instance, by clock pulses occurring instantaneously at each cell whenever a clock signal is given to the automaton. (This could be visualized as a $(\hat{Z}, T)$-structure with T-inputs occurring synchronously at each cell.) In this case, the minimum signal distance would be $\varepsilon + s_{\max}$, see (2).

In a locally synchronous cellular automaton $(\hat{Z}, T)$, the minimum time distance of clock signals given to the origin cell is $D_{\min}$, as seen from (9). As $D_{\min} = v_{\max} + \bar{v}_{\max} + s_{\max}$, we have, by $\varepsilon' := v_{\max} + \bar{v}_{\max}$, a computation speed of the $(\hat{Z}, T)$ automaton which relates to the above case of instantaneous signal transmission! (Of course, in a finite run of a cellular automaton, corresponding to a sequence of a certain number of deduction steps in the calculus, special considerations have to be made of the time by which the *total* result of such a computation can be brought out, referring to t. The discussion of this is deferred to [9]; at the moment we are considering infinite runs.)

To attain a concept of a "configuration of a locally synchronous cellular automaton", we now introduce a *new concept of synchronity* which is based on local observation times: Corresponding to each clock signal T_k, we consider local time scales t_k^α for each individual cell α in a locally synchronous cellular automaton: $t_k^\alpha = (\mathbf{R}, \leqq, t_k^\alpha)$, where $t_k^\alpha \in t$ $(k = 1, 2, \ldots)$. Thus for each cell α, with each causal T-input $\langle t_k^\alpha \rangle$ a new observation time is beginning ("$\langle t_k^\alpha \rangle$ brings along the time"; compared with t, the starting points of the local scales of neighbouring cells differ, at any given time, at most by the maximum component delay, $v_{\max}$). Then, for each k, each cell α is in a well-defined state at time t_k^α (see section 4), before the k-th clock signal, represented by the k-th clock pulse $\langle t_k^\alpha \rangle$, $t_k^\alpha < t_k^\alpha$, appears at α.

In definition 3, the synchronity of clock pulse events was defined by coincidence of the times of their occurrences, with respect to t. (This could as well be applied to define the synchronity of causal T-inputs.) Here, we define the "synchronity" of events of causal T-inputs occurring in a T-net by covering the starting points of the local time scales that correspond to the same clock signal:

Definition 4. In a $(\hat{Z}, T)$-structure, two events $\langle t_k^\alpha \rangle$ and $\langle t_l^\beta \rangle$, where t_k^α, $t_l^\beta \in t$, are said to be *T-synchronous* iff $k = l$. To denote this we shall write $\langle t_k^\alpha \rangle$ T-syn $\langle t_l^\beta \rangle$.

The concept of T-synchrony is well-defined in a locally synchronous cellular automaton since by each clock signal T_k there occurs exactly one clock pulse event at each cell α and, hence, exactly one T-input $\langle \underline{t}_k^\alpha \rangle$ causal for this. T-synchronity gives an equivalence relation, in a canonical way, dividing the set of causal T-inputs into a countable number of classes, each containing all causal T-inputs generated by the same clock signal. We define special sets $\underline{t}_k$ called (k-th) *time cuts*, by "cutting through the local times":

$$\underline{t}_k := \{\underline{t}_l^\gamma | \langle \underline{t}_l^\gamma \rangle \ T\text{-syn} \ \langle \underline{t}_k^0 \rangle\}$$

and order them by means of their natural sequence, i.e. along k. This concept makes possible a simultaneous consideration of the local time scales $\mathscr{t}_k^\alpha$, for each k (refer to the remarks given on synchronity, in section 2): While in the above hypothetic case there were considered *synchronous* events of clock pulse inputs at all cells, we consider now, in structure $(\hat{Z}, T)$, the case of *T-synchronous* T-inputs at all cells. A time cut $\underline{t}_k$ contains exactly the instants of all events of k-th causal T-inputs, namely, the starting points of the local times $\mathscr{t}_k^\alpha$. To set up the relation to the global observation time $\mathscr{t}$, we can also express the $\underline{t}_k$ by the following:

$$\underline{t}_k = \{\underline{t}_k^\alpha | \underline{t}_k^\alpha \in \mathscr{t} \ \& \ \exists \alpha \langle \underline{t}_k^\alpha \rangle\}$$

An element of $\underline{t}_k$ can, of course, be the starting point of several local time scales $\mathscr{t}_k^\alpha, \mathscr{t}_k^\beta, \ldots$ (with fixed k), in case that several of these T-synchronous events occur synchronously, in the original sense. Each time cut thus contains at most as many instants as the number of cells under consideration because, in locally synchronous working, no cell can get more than one k-th causal T-input ($k=1, 2, \ldots$). On the other hand, each causal T-input lies in exactly one time cut.

By use of the concepts of T-synchronity and time cut we now proceed to explain the terms "configuration" and "global state transition of a locally synchronous cellular automaton".

With reference to any given time cut $\underline{t}_k$, all cells of such an automaton are T-synchronously in well-defined states, and thus it makes sense to speak of a global state, or a configuration, of the locally synchronous cellular automaton *"at time cut* $\underline{t}_k$". The relation between the continuous time scale $\mathscr{t}$ and a discrete time scale to take as a basis for the consideration of global steps is now to be worked out. The times of input of clock signals $T_1, T_2, \ldots$ at the origin 0 were denoted by $\underline{t}_1^0, \underline{t}_2^0, \ldots$, and from (9) it is, for all k, $|\underline{t}_{k+1}^0 - \underline{t}_k^0| \geqq D_{\min} = \varepsilon' + s_{\max}$. So we could say: *Locally*, i.e. at each individual cell α, time passes by in discrete steps $\underline{t}_1^\alpha \to \underline{t}_2^\alpha \to \ldots$; each time a step $\underline{t}_k^\alpha \to \underline{t}_{k+1}^\alpha$ has elapsed (by occurrence of $\langle \underline{t}_{k+1}^\alpha \rangle$), α has once changed its state.

For a locally synchronous cellular automaton, we now define a discrete time scale the "instants" of which are *time cuts:*

$$\vec{\vec{\mathscr{t}}} = (\{\underline{t}_k\}_{k \in \mathrm{N}}, \leqq, \underline{t}_1)$$

In section 2, we introduced a continuous time scale to explain locally synchronous working; now we have returned to a discrete time concept, by means of the equivalence relation of T-synchronity. While, in the synchronous case, saying that between each two subsequent instants $\underline{t}_1^0$, $\underline{t}_2^0$ of $\vec{\mathscr{t}}$ there takes place a global state transition of $\hat{Z}$ leading from a configuration c to a successor configuration c', we could say that, in a locally synchronous cellular automaton, between each two subsequent time cuts t_k, t_{k+1} of $\vec{t}$ there takes place a global state transition of $(\hat{Z}, T)$ from c to a c'.

We conclude this discussion by a remark added to clarify the above ideas. The concept of T-synchronity might seem to be a bit rough and only of technical value to re-open the view of a global state. But think of an imaginary observer gliding across the cellular plane, at the same bounded speed as is actually given by the component delays appearing in the T-net, along his course. Then, at any region of the plane, he would get the impression of a *static* global state from the fact that, during his observation, all cells passed are T-synchronously in well-defined states, corresponding to the so far obtained result of the execution of a deduction step in the simultaneous cellular calculus.

Abstract

It is shown that, for a cellular automaton in the sense of John von Neumann, the assumption of synchronous state transitions of all cells is not necessary but can be weakened to an assumption of locally synchronous working. An organization scheme is described which achieves locally synchronous performance in any cellular automaton of the von Neumann type. A special concept of synchronity is introduced which makes possible the consideration of configurations and global state transitions of locally synchronous cellular automata.

FB MATHEMATIK
UNIVERSITÄT OSNABRÜCK
ALBRECHTSTR. 28
D—4500 OSNABRÜCK, W. GERMANY

References

[1] ARBIB, M. A., *Theories of abstract automata,* Englewood Cliffs, N. J., 1969.

[2] BURKS, W. ed., *Essays on cellular automata,* Urbana, Ill., 1970.

[3] GOLZE, U., *A*-synchronous non-deterministic cell spaces simulating each other, *J. Comput. System Sci.,* v. 17, No. 2, 1978, pp. 176—193.

[4] LEGENDI, T., Programming of cellular processors; GOLZE, U., VOLLMAR, R. eds., *Beiträge zur Theorie der Polyautomaten,* Braunschweig, 1977, pp. 53—66.

[5] NAKAMURA, K., Asynchronous cellular automata and their computational ability, *Systems Comput. Controls,* v. 5, No. 5, 1974, pp. 58—66.

[6] VON NEUMANN, J., Theory of automata: Construction, Reproduction, Homogeneity; *The Theory of Self-Reproducing Automata,* herausgeg. und vervollst. von A. W. Burks, Urbana, Ill., 1966.

[7] PRIESE, L., A note on asynchronous cellular automata, *J. Comput. System Sci.,* v. 17, No. 2, 1978, pp. 237—251.

[8] WACHSMUTH, I., Simultane zellulare Kalküle und lokal-synchrone Zellularautomaten, Dissertation, Universität Hannover, 1980.

[9] WACHSMUTH, I., Nets with time-variant component delays, in preparation.

(Received April 19, 1981)

О сложности реализации булевых функций с данным вектором активностей

А. В. Петросян, В. А. Варданян

Рассмотрим задачу реализации булевых функций с данным вектором активностей схемами из функциональных элементов в базисе $\{\vee, \&, -\}$.

Введем следующие обозначения:

B^n — n — мерный единичный куб;

N_f — множество наборов из B^n — на которых булевая функция $f(x_1, x_2, \ldots, x_n)$ принимает значение 1;

$|A|$ — число элементов множества A.

Вектор $\omega^f=(\omega_1^f, \ldots, \omega_n^f)$ называется вектором активностей функции $f(x_1, \ldots, x_n)$, а число ω_i^f называется активностью функции $f(x_1, \ldots, x_n)$ по i-й координате, если

$$\omega_i^f = \frac{1}{2^n} \sum_{(\alpha_1,\ldots,\alpha_n)\in B^n} \left(f(\alpha_1, \ldots, \alpha_i, \ldots, \alpha_n) \oplus f(\alpha_1, \ldots, \bar{\alpha}_i, \ldots, \alpha_n)\right), \quad 1 \leqq i \leqq n.$$

Подробно об активностях можно найти в работе [4].

Пусть $\bar{\omega}=(\omega_1, \ldots, \omega_n)$-набор n вещественных чисел, для которых $\omega_i = \frac{\eta_i(n)}{2^{n-1}}$, $\eta_i(n)$-целочисленные параметры, $0<\eta_i(n)\leqq 2^{n-1}$, $1\leqq i\leqq n$. Через $\Phi_n(\bar{\omega})$ обозначим класс функций $f(x_1, \ldots, x_n)$, для которых $\omega_i^f\leqq\omega_i$, $1\leqq i\leqq n$.

Пусть $L(f)$ — наименьшая из сложностей схем реализующих функцию f, и $L(\Phi_n(\bar{\omega}))=\max_{f\in\Phi_n(\bar{\omega})} L(f)$.

В настоящей работе устанавливаются верхние и нижние оценки для сложности $L(\Phi_n(\bar{\omega}))$.

Пусть $f(x_1, \ldots, x_n)$ — произвольная функция из класса $\Phi_n(\bar{\omega})$. Рассмотрим ее разложение по переменной x_1:

$$f(x_1, \ldots x_n) = x_1 f(1, x_2, \ldots, x_n) \vee \bar{x}_1 f(0, x_2, \ldots, x_n). \qquad (1)$$

Введем следующие обозначения:

$$f_0(x_1, \dots, x_n) = f(x_1, \dots, x_n)$$

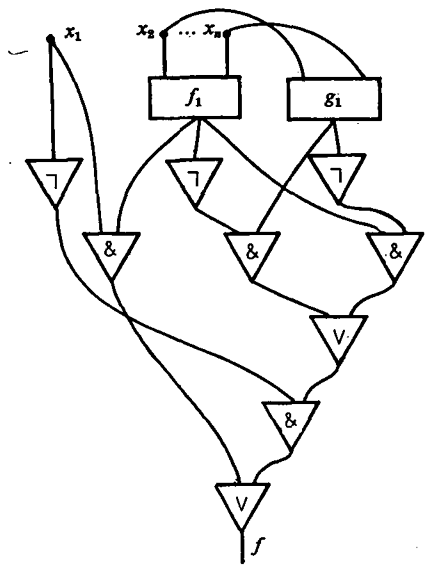
$$f_1(x_2, \dots, x_n) = f(1, x_2, \dots, x_n)$$

$$g_1(x_2, \dots, x_n) = f_0(1, x_2, \dots, x_n) \oplus f_0(0, x_2, \dots, x_n)$$

Заметим, что $|N_{g_1}| = \omega_1^{f_0} 2^{n-1} = \omega_1^{f} 2^{n-1}$. Легко видеть, что разложение (1) может быть представлено в виде

$$f_0(x_1, \dots, x_n) = x_1 f_1 \vee \bar{x}_1 (f_1 \oplus g_1). \tag{2}$$

Следовательно (см. рис.)



$$L(f) = L(f_0) \leqq L(f_1) + L(g_1) + 9 \tag{3}$$

Рассмотрим последовательные разложения функций

$$f_j(x_{j+1}, \dots, x_n) \quad \text{при} \quad j = 1, 2, \dots, k-1:$$

$$f_j(x_{j+1}, \dots, x_n) = x_{j+1} f_{j+1}(x_{j+2}, \dots, x_n) \vee$$

$$\vee \bar{x}_{j+1} (f_{j+1}(x_{j+2}, \dots, x_n) \oplus g_{j+1}(x_{j+2}, \dots, x_n))$$

где

$$f_{j+1}(x_{j+2}, \dots, x_n) = f_j(1, x_{j+2}, \dots, x_n)$$

$$g_{j+1}(x_{j+2}, \dots, x_n) = f_j(1, x_{j+2}, \dots, x_n) \oplus f_j(0, x_{j+2}, \dots, x_n).$$

Тогда ясно, что

$$L(f_j) \leqq L(f_{j+1}) + L(g_{j+1}) + 9, \quad 1 \leqq j \leqq k-1. \tag{4}$$

Складывая соответственно левые и правые части неравенств (3) и (4) при $1 \leqq j \leqq k-1$, получаем, что

$$L(f) \leqq L(f_k) + \sum_{i=1}^{k} L(g_i) + 9k \tag{5}$$

По известной теореме Лупанова [1, 2] функцию $f_k(x_{k+1}, \ldots, x_n)$ можно реализовать схемой сложности не более $\frac{2^{n-k}}{n-k}\left(1+O\left(\frac{\log(n-k)}{n-k}\right)\right)$. Другими словами,

$$L(f_k) \leqq \frac{2^{n-k}}{n-k}(1+o(1)).$$

Положим $k=[n-2\log n]$, где $[x]$—целая часть чусла x. Тогда $L(f_{[n-2\log n]}) \leqq \leqq \frac{n^2}{\log n}(1+o(1))$ и из неравенства (4) следует, что

$$\begin{aligned} L(f) &\leqq \frac{n^2}{\log n}(1+o(1)) + \sum_{i=1}^{[n-2\log n]} L(g_i) + 9(n-2\log n) \leqq \\ &\leqq \frac{n^2}{\log n}(1+o(1)) + \sum_{i=1}^{[n-2\log n]} L(g_i). \end{aligned} \tag{6}$$

Нетрудно заметить, что

$$|Ng_i| = \omega_i^{f_{i-1}} 2^{n-i} \leqq \omega_i^{f} 2^{n-1} \leqq \omega_i 2^{n-1}, \quad 1 \leqq i \leqq [n-2\log n] \tag{7}$$

Теорема 1. Если $\omega_i = O\left(\frac{\varphi(n)}{2^{n-1}}\right)$, $1 \leqq i \leqq [n-2\log n]$, где $\varphi(n)$ — целочисленный параметр, $1 \leqq \varphi(n) \leqq 2^{n-1}$, $\varphi(n) \to \infty$ при $n \to \infty$, то

$$L(\Phi_n(\bar{\omega})) \lesssim n2^{n-1} \sum_{i=1}^{[n-2\log n]} \omega_i = O(n^2\varphi(n)).$$

Доказательство. Для реализации функций g_i, $1 \leqq i \leqq [n-2\log n]$ используем метод синтеза, основанный на совершенной дизъюнтивной нормальной форме [3]. Тогда, учитывая неравенства (7), получаем, что

$$L(g_i) \leqq (n-i)(\omega_i^{f_{i-1}} 2^{n-i}+1)-1 \leqq n(\omega_i 2^{n-1}+1)-1.$$

Таким образом из (6) следует, что

$$L(f) \leqq \frac{n^2}{\log n}(1+o(1)) + n(n-2\log n) + n2^{n-1} \sum_{i=1}^{[n-2\log n]} \omega_i. \tag{8}$$

Если $\omega_i = O\left(\frac{\varphi(n)}{2^{n-1}}\right)$, $1 \leqq i \leqq [n-2\log n]$, где $\varphi(n) \to \infty$ при $n \to \infty$, то главной частью в выражении (8) будет $n2^{n-1} \sum_{i=1}^{[n-2\log n]} \omega_i$, другими словами

$$L(f) \lesssim n2^{n-1} \sum_{i=1}^{[n-2\log n]} \omega_i = O(n^2\varphi(n))$$

Так как правая часть последнего неравенства не зависит от конкретной функции, то получаем утверждение теоремы. Теорема доказана.

Верхняя оценка, полученная в теореме 1, в некоторых случаях можно улучшить, если при реализации функций g_i используем методы Финикова и Лупанова [1] для реализации функций с данным числом единиц.

Теорема 2. 1) Если $\omega_i = O\left(\frac{\log n}{2^{n-1}}\right)$, $1 \leqq i \leqq [n-2\log n]$, то $L(\Phi_n(\bar\omega)) \lesssim cn^2$, где $c=\text{const}$.

2) Если $\omega_i \leqq \frac{1}{2}$ и $\frac{\omega_i 2^{n-1}}{\log n} \to \infty$, при $n\to\infty$, $1\leqq i\leqq [n-2\log n]$ то

$$L(\Phi_n(\bar\omega)) \lesssim \sum_{i=1}^{[n-2\log n]} \frac{\log C_{2^{n-1}}^{\omega_i 2^{n-1}}}{\log\log C_{2^{n-1}}^{\omega_i 2^{n-1}}}.$$

Доказательство. Если $\omega_i = O\left(\frac{\log n}{2^{n-1}}\right)$, $1\leqq i\leqq [n-2\log n]$, то учитывая неравенства (7) и результат Финикова [1], получаем, что $L(g_i)\lesssim n$, $1\leqq i\leqq [n-2\log n]$. Тогда из (6) следует утверждение 1).

Если $\frac{\omega_i 2^{n-1}}{\log n}\to\infty$, то из неравенства (7) на основании результата Лупанова [1] имеем, что

$$L(g_i) \lesssim \frac{\log C_{2^{n-i}}^{\omega_i^{f_{i-1}} 2^{n-i}}}{\log\log C_{2^{n-i}}^{\omega_i^{f_{i-1}} 2^{n-i}}} \leqq \frac{\log C_{2^{n-1}}^{\omega_i^{f} 2^{n-1}}}{\log\log C_{2^{n-1}}^{\omega_i^{f} 2^{n-1}}}.$$

Следовательно, из (8) получаем, что

$$L(f) \lesssim \sum_{i=1}^{[n-2\log n]} \frac{\log C_{2^{n-1}}^{\omega_i^{f} 2^{n-1}}}{\log\log C_{2^{n-1}}^{\omega_i^{f} 2^{n-1}}}.$$

В силу того, что $\omega_i\leqq 1/2$, $1\leqq i\leqq [n-2\log n]$, из последнего неравенства следует, что

$$L(f) \lesssim \sum_{i=1}^{[n-2\log n]} \frac{\log C_{2^{n-1}}^{\omega_i 2^{n-1}}}{\log\log C_{2^{n-1}}^{\omega_i 2^{n-1}}}.$$

Правая часть последнего неравенства не зависит от функции f, откуда и вытекает утверждение 2).

Замечание. Используя таблицу сложностей классов функций для некоторых значений числа единиц, приведенную в [1], находим соответствующие асимптотики для верхней оценки $L(\Phi_{n+1}(\bar\omega))$.

Нижние оценки для $L(\Phi_n(\bar\omega))$ получаются из «мощностных соображений» [1, 2], основанные на следующем утверждении.

$\omega_i,\ 1 \leq i \leq [n+1-2\log(n+1)]$	$L(\Phi_{n+1}(\bar{\omega}))$
$\frac{(\log n)^c}{2^n},\quad c>1$	$n^2(\log n)^{c-1}$
$\frac{n^c}{2^n},\quad c>0$	$\frac{n^{c+2}}{(c+1)\log n}$
$2^{n^c-n},\quad 0<c<1$	$n^{2-c}2^{n^c}$
$2^{(c-1)n},\quad 0<c<1$	$\frac{1-c}{c}\,n2^{cn}$
$\frac{1}{\psi(n)}$, где $\psi(n)\to\infty,\quad \frac{\log\psi(n)}{n}\to 0$ в частности $\frac{1}{n^c},\quad c>0$	$\frac{2^n\log\psi(n)}{\psi(n)}$ $\frac{c2^n\log n}{n^c}$
$\alpha,\quad 0<\alpha\leq\frac{1}{2}$	$\beta 2^n$, где $2^{-\beta}=\alpha^\alpha(1-\alpha)^{1-\alpha}$

Лемма

$$|\Phi_n(\bar{\omega})| > \left(\frac{2}{\omega_i}\right)^{\omega_i 2^{n-1}},\quad \omega_i=\min_{1\leq j\leq n}\omega_j$$

Доказательство. Через Φ_{ω_i} обозначим класс функций $f(x_1,\dots,x_n)$, для которых $\omega_i^f=\omega_i$, и из равенства $f(\alpha_1,\dots,\alpha_i,\dots,\alpha_n)=f(\alpha_1,\dots,\bar{\alpha}_i,\dots,\alpha_n)$ следует, что $f(\alpha_1,\dots,\alpha_i,\dots,\alpha_n)=f(\alpha_1,\dots,\bar{\alpha}_i,\dots,\alpha_n)=0$.

Нетрудно проверить, что если $f\in\Phi_{\omega_i}$, то $\omega_j^f\leq\omega_i^f$, $1\leq j\leq n$. Следовательно, $\Phi_{\omega_i}\subset\Phi_n(\bar{\omega})$, $|\Phi_{\omega_i}|<|\Phi_n(\bar{\omega})|$.

Легко видеть, что $|\Phi_{\omega_i}|=C_{2^{n-1}}^{\omega_i 2^{n-1}}\cdot 2^{\omega_i 2^{n-1}}\geq$ (использовали неравенство $C_n^k>\left(\frac{n}{k}\right)^k$) $\geq\left(\frac{2^{n-1}}{\omega_i 2^{n-1}}\right)^{\omega_i 2^{n-1}}\cdot 2^{\omega_i 2^{n-1}}$ откуда и получаем утверждение леммы.

Теорема 3. Если, $\omega_i=\min\limits_{1\leq j\leq n}\omega_j$, $\frac{\omega_i 2^{n-1}}{\log n}\to\infty$ при $n\to\infty$, то

$$L(\Phi_n(\bar{\omega}))\gtrsim\frac{\omega_i 2^{n-1}(1-\log\omega_i)}{\log(\omega_i n2^{n-1})}.$$

Доказательство. Достаточно показать, что для любого $\varepsilon>0$ при $k(n)=(1-\varepsilon)\frac{\omega_i 2^{n-1}(1-\log\omega_i)}{\log(\omega_i n2^{n-1})}$ и $n\to\infty$ справедливо соотношение $\frac{N(n,k(n))}{|\Phi_n(\bar{\omega})|}\to 0$ $(n\to\infty)$, где $N(n,k)$ — число различных неприводимых схем сложности не более k (см. [1, 2]).

Известно (см. [1, 2]), что $N(n, k(n)) \leqq (16(n+k(n)))^{n+k(n)+3}$. Из этого неравенства и леммы имеем

$$\log \frac{N(n, k(n))}{|\Phi_n(\bar{\omega})|} = \log N\left(n, \frac{(1-\varepsilon)\omega_i 2^{n-1}(1-\log\omega_i)}{\log(\omega_i n 2^{n-1})}\right) - \log|\Phi_n(\bar{\omega})| \leqq$$

$$\leqq \left[(1-\varepsilon)\frac{\omega_i 2^{n-1}(1-\log\omega_i)}{\log(\omega_i n 2^{n-1})} + n + 3\right]\left[4 + \log\left(n + \frac{(1-\varepsilon)\omega_i 2^{n-1}(1-\log\omega_i)}{\log(\omega_i n 2^{n-1})}\right)\right] -$$

$$-\omega_i 2^{n-1}(1-\log\omega_i) \to -\infty,$$

при $n \to \infty$, если $\frac{\omega_i 2^{n-1}}{\log n} \to \infty$. Тем самым теорема доказана.

Следствие 1. Если $\omega_j \leqq 1/2$, $1 \leqq j \leqq [n - 2\log n]$ и $\frac{\omega_i 2^{n-1}}{\log n} \to \infty$ при $n \to \infty$, $\omega_i = \min_{1 \leqq j \leqq n} \omega_j$, то

$$\frac{\omega_i 2^{n-1}(1-\log\omega_i)}{\log(\omega_i n 2^{n-1})} \lesssim L(\Phi_n(\bar{\omega})) \lesssim \sum_{i=1}^{[n-2\log n]} \frac{\log C_{2^{n-1}}^{\omega_i 2^{n-1}}}{\log\log C_{2^{n-1}}^{\omega_i 2^{n-1}}}.$$

Следствие 2. Если $\omega_i = \frac{n^s}{2^{n-1}}$, $s > 0$, $1 \leqq i \leqq n$, то

$$\frac{n^{s+1}}{(s+1)\log n} \lesssim L(\Phi_n(\bar{\omega})) \lesssim \frac{n^{s+2}}{(s+1)\log n}.$$

Summary

It is shown in this work that if Boolean functions with restricted activities of arguments are considered then more simple realizing schemes can be constructed. In many partial cases the upper estimations for Shannon function have been recevied and in some cases the lower ones are given as well.

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР
АКАДЕМИИ НАУК АРМЯНСКОЙ ССР
УЛ. П. СЕВАКА, 1.
375044 ЕРЕВАН-44, СССР

Литература

[1] Лупанов, О. Б., Об одном подходе к синтезу управляющих систем — принципе локального кодирования, *Проблемы кибернетики*, вып. 14, М., *Наука*, 1965.

[2] Лупанов, О. Б., О синтезе некоторых классов управляющих систем, *Проблемы кибернетики*, вып. 10, М., Физматгиз, 1963.

[3] Яблонский, С. В., Введение в дискретную математику, М., *Наука*, 1979.

[4] Петросян, А. В., Некоторые дифференциальные свойства булевых функций, Совместный сборник ИВТиА Венгерской АН и ВЦ АН Арм. ССР *Некоторые задачи автоматизации проектирования*, Будапешт, 1982.

(Поступило 19-ого января 1982 г.)

On homomorphically α_i-complete systems of automata

By P. Dömösi

In [2] there is introduced a family of semi-cascade products named α_i-products, where the index i is a nonnegative integer, which denotes the maximal admissible length of feedbacks. By results of F. Gécseg (see for example [3]) it can be seen that there exist no finite homomorphically complete systems with respect to the α_0- and α_1-products.

From [1] it follows that every automaton having n states can be represented (in a certain sense) by an α_0-product of automata, such that all components of this product are either two-state reset automata, or special n-state automata, named "standard automata". Using results of [4] we get that these "standard automata" can be embedded state-isomorphically into an α_2-product of two-state automata. Therefore, taking into consideration the fact that an α_0-product of α_2-products is an α_2-product, every automaton can be represented (in a certain sense) by an α_2-product of two-state automata.

In this paper we present a direct proof of this statement. By this result we receive that for every $i \geqq 2$ there exists a finite homomorphically complete system of automata with respect to the α_i-product. For the notions and notations that will not be defined here, we refer to the book [3].

By an *automaton* $\mathbf{A}=(X, A, Y, \delta, \lambda)$ we mean a finite Mealy-type automaton, where X, A and Y are the finite input, state and output sets, respectively; furthermore $\delta\colon A\times X\to A$ denotes the transition and $\lambda\colon A\times X\to Y$ is the output function.

Let $\mathbf{A}_t=(X_t, A_t, Y_t, \delta_t, \lambda_t)$ $(t=1, \ldots, n)$ be a system of automata. Moreover, let X and Y be finite nonvoid sets and

$$\varphi\colon A_1\times\ldots\times A_n\times X \to X_1\times\ldots\times X_n, \quad \psi\colon A_1\times\ldots\times A_n\times X\to Y$$

mappings. We say that the automaton $\mathbf{A}=(A, X, Y, \delta, \lambda)$ with $A=A_1\times\ldots\times A_n$, $\lambda((a_1,\ldots, a_n), x)=\psi(a_1, \ldots, a_n, x)$

$$\delta((a_1, \ldots, a_n), x) = (\delta_1(a_1, \varphi_1(a_1, \ldots, a_n, x)), \ldots, \delta_n(a_n, \varphi_n(a_1, \ldots, a_n, x))),$$

is the α_i-*product* of $\mathbf{A}_t$ $(t=1, \ldots, n)$ with respect to X Y, φ, ψ if φ can be given in the form $\varphi(a_1, \ldots, a_n, x)=(\varphi_1(a_1, \ldots, a_n, x), \ldots, \varphi_n(a_1, \ldots, a_n, x))$, such that $\varphi_j(1\leqq j\leqq n)$ is independent of states having indices greater than or equal to $j+i$, where i is a fixed nonnegative integer. For this product we shall use the short notation

$\mathbf{A}=\prod_{t=1}^{n}\mathbf{A}_t[X, Y, \varphi, \psi]$. The mappings φ and ψ are called *feedback function* and *output function*, respectively.

Let **A**, **B** be a pair of automata. We say that **A** *can be embedded state-isomorphically* into **B** if **B** has an A-subautomaton $\mathbf{B}'$, such that $\mathbf{B}'$ is A-isomorphic to **A**. If **B** has an A-subautomaton $\mathbf{B}'$, such that $\mathbf{B}'$ can be mapped A-homomorphically onto **A** then it is said that **A** *can be strongly covered* (or *can be represented*) by **B**.

Take a nonnegative integer i. Any system Σ of automata is *homomorphically complete with respect to the* α_i*-product,* or briefly, Σ is *homomorphically* α_i*-complete* if every automaton can be strongly covered by an appropriate α_i-product of components from Σ. Moreover, the system Σ is *finite* if it has finite-many elements.

Consider an automaton $\mathbf{A}=(X, A, Y, \delta, \lambda)$ with n states. For an arbitrary positive integer $m\leqq n$ we say that **A** is *m-husked* if there exists an arrangement $a_1, \ldots, a_n$ of states in **A**, such that for $a_l\in A$, $x\in X$, $l<m$ we have $\delta(a_l, x)\in\langle a_1, \ldots, a_{l+1}\rangle$. (Obviously, for $m=1$ this is a formal requirement. Therefore, all automata are 1-husked.)

If an automaton **A** with n states is n-husked then it is said to be *right-husked.* (We note that all $(n-1)$-husked automata with $n>1$ states need necessarily be right-husked.)

The following holds.

Lemma 1. Every m-husked automaton **A** having $n>m$ states can be strongly covered by a suitable α_0-product $\mathbf{M}=\prod_{t=1}^{2}\mathbf{A}_t[X, Y, \varphi, \psi]$ whose components satisfy the following conditions:

(i) $\mathbf{A}_1$ has $n-m$ states;

(ii) $\mathbf{A}_2$ is an $(m+1)$-husked automaton the number of states of which is equal to n.

Proof. Take an m-husked automaton $\mathbf{A}=(X, A, Y, \delta, \lambda)$ with $n>m$ number of states and let $a_1, \ldots, a_n$ be an arrangement of states in **A**, such that for $a_l\in A$, $x\in X$, $l<m$ it holds that $\delta(a_l, x)\in\langle a_1, \ldots, a_{l+1}\rangle$. For any triplet $u, v, w\in\langle 1, \ldots, n\rangle$ we introduce the notation

$$a_{(u,v,w)}=\begin{cases} a_u & \text{if } u\notin\langle v, w\rangle,\\ a_v & \text{if } u=w,\\ a_w & \text{if } u=v.\end{cases}$$

Construct the automata $\mathbf{A}_1=(X, B, B\times X, \delta_1, \lambda_1)$ and $\mathbf{A}_2=(B\times X, A, Y, \delta_2, \lambda_2)$ in the following way. $B=\langle m+1, \ldots, n\rangle$, furthermore, for every triplet $v\in B$, $a_l\in A$, $x\in X$

$$\delta_1(v, x)=\begin{cases} v & \text{if } \delta(a_m, x)\in\langle a_1, \ldots, a_m\rangle,\\ w & \text{if } \delta(a_m, x)\notin\langle a_1, \ldots, a_m\rangle \text{ and } \delta(a_m, x)=a_w,\end{cases}$$

$$\delta_2(a_l, (v, x))=\begin{cases} a_{(z,m+1,v)} & \text{if } \delta(a_m, x)\in\langle a_1, \ldots, a_m\rangle \text{ and } \delta(a_{(l,m+1,v)}, x)=a_z,\\ a_{(z,m+1,w)} & \text{if } \delta(a_m, x)=a_w\notin\langle a_1, \ldots, a_m\rangle\\ & \qquad\text{and } \delta(a_{(l,m+1,v)}, x)=a_z,\end{cases}$$

$$\lambda_1(v, x)=(v, x),\quad \lambda_2(a_l, (v, x))=\lambda(a_{(l,m+1,v)}, x).$$

Define the α_0-product $\mathbf{M}=\sum_{t=1}^{2}\mathbf{A}_t[X, Y, \varphi, \psi]$, where in case of every pair $(v, a_l)\in B\times A$, $x\in X$

$$\varphi(v, a_l, x) = (x, (v, x)),$$

$$\psi(v, a_l, x) = \lambda_2(a_l, (v, x)).$$

By an elementary computation we obtain that the mapping μ: $B\times A\to A$ with $\mu(v, a_l)=a_{(l, m+1, v)}$ is an A-homomorphism of $\mathbf{M}$ onto $\mathbf{A}$. By the definitions of $\mathbf{A}_1$ and $\mathbf{A}_2$ this completes the proof of Lemma 1.

The following statement is trivial.

Lemma 2. Let $\langle \mathbf{A}_1, ..., \mathbf{A}_n\rangle$ and $\langle \mathbf{B}_1, ..., \mathbf{B}_m\rangle$ be arbitrary finite systems of automata. If any automaton $\mathbf{A}$ can be strongly covered by an α_0-product of components from $\langle \mathbf{A}_1, ..., \mathbf{A}_n\rangle$, moreover, an element $\mathbf{A}_t$ of $\langle \mathbf{A}_1, ..., \mathbf{A}_n\rangle$ can be strongly covered by an α_0-product of components from $\langle \mathbf{B}_1, ..., \mathbf{B}_m\rangle$ then $\mathbf{A}$ can be strongly covered by an α_0-product of components from $\langle \mathbf{A}_1, ..., \mathbf{A}_{t-1}, \mathbf{B}_1, ..., \mathbf{B}_m, \mathbf{A}_{t+1}, ..., A_n\rangle$.

Using Lemma 1 and Lemma 2 by an induction we get the following

Lemma 3. Every automaton $\mathbf{A}$ can be strongly covered by an α_0-product of right-husked automata having not more states than $\mathbf{A}$.

Lemma 4. Every right-husked automaton can be embedded state-isomorphically into and α_2-product of two-state automata.

Proof. Let $\mathbf{A}=(X, A, Y, \delta, \lambda)$ be an arbitrary right-husked automaton and take an arrangement $a_1, ..., a_n$ of its states with $\delta(a_t, x)\in\langle a_1, ..., a_{t+1}\rangle$ $(t=1, ..., n-1$, $x\in X)$. Consider the automaton $\mathbf{B}=(\langle u, v\rangle, \langle 0, 1\rangle, \langle z\rangle, \delta_\mathbf{B}, \lambda_\mathbf{B})$ where $\delta_\mathbf{B}(0, u)= =\delta_\mathbf{B}(1, v)=0$, $\delta_\mathbf{B}(0, v)=\delta_\mathbf{B}(1, u)=1$ and $\lambda_\mathbf{B}(j, x)=z$ for any $j\in\langle 0, 1\rangle, x\in\langle u, v\rangle$. Construct the α_2-product $\mathbf{C}=(X, C, Y, \delta_\mathbf{C}, \lambda_\mathbf{C})=\prod_{t=1}^{n}\mathbf{B}_t\ [X, Y, \varphi, \psi]$ with $\mathbf{B}_1= =...=\mathbf{B}_n=\mathbf{B}$ as follows. For any $1\leqq s\leqq n$, $(d_1, ..., d_n)\in\prod_{t=1}^{n}B_t$ and $x\in X$

$$\varphi_s(d_1, ..., d_n, x) = \begin{cases} v & \text{if } d_j = 1,\ \ \delta(a_{n-j+1}, x) = a_{n-s+1} \text{ for some} \\ & \qquad j\in\langle 1, ..., s-1, s+1\rangle\cap\langle 1, ..., n\rangle \text{ or} \\ & \qquad d_s = 1,\ \ \delta(a_{n-s+1}, x) \neq a_{n-s+1}, \\ u & \text{otherwise,} \end{cases}$$

$$\psi(d_1, ..., d_n, x) = \begin{cases} \lambda(a_{n-j+1}, x) & \text{if } d_j = 1 \text{ for some } 1\leqq j\leqq n \\ & \qquad \text{and } \sum_{t=1}^{n} d_t = 1, \\ \text{arbitrary fixed element of } Y \text{ otherwise.} \end{cases}$$

Denote C' the set of all elements $(d_1, ..., d_n)\in\prod_{t=1}^{n}B_t$ for which $\sum_{t=1}^{n}d_t=1$. It is clear that $\mathbf{C}'=(X, C', Y, \delta_{\mathbf{C}|C'\times X}, \lambda_{\mathbf{C}|C'\times X})$ is an A-subautomaton of $\mathbf{C}$. Now consider the mapping $\nu\colon (d_1, \ldots, d_n) \to a_{\sum_{t=1}^{n} d_t \cdot (n-t+1)}$ $((d_1, \ldots, d_n) \in C')$. It can be seen that ν is an A-isomorphism of $\mathbf{C}'$ onto $\mathbf{A}$. This ends the proof of Lemma 4.

It is evident that any α_0-product of α_2-products also is an α_2-product. Therefore, by Lemma 3 and Lemma 4, the following result is shown.

Theorem. Every automaton can be strongly covered by an α_2-product of two-state automata.

We know, by definition, that for every $i > 2$ the concept of α_i-product is a generalization of α_2-product. (In [4] it is shown that this generalization is proper.) Thus, the above Theorem and our remark about α_0-product and α_1-product jointly imply the following result.

Corollary. For every nonnegative integer i there exists a finite homomorphically α_i-complete system if and only if $i \geqq 2$.

DEPT. OF MATHEMATICS
KARL MARX UNIVERSITY OF ECONOMICS
DIMITROV TÉR 8.
BUDAPEST, HUNGARY
H—1093

References

[1] Евтушенко, Н. В., К реализации автоматов каскадным соединением стандардных автоматов, *Автоматика и Вычислительная Техника*, 1979, № 2, pp. 50—53.
[2] Gécseg, F., Composition of automata, Proceedings of the 2nd Colloquium on Automata, Languages and Programming, v. 14, 1974, pp. 351—363.
[3] Gécseg, F. and I. Peák, *Algebraic theory of automata*, Akadémiai Kiadó, Budapest, 1972.
[4] Imreh, B., On α_i-products of automata, *Acta Cybernet.*, v. 3, 1978, pp. 301—307.

(Received Dec. 19, 1981)

Одна задача об ориентированных графах

А. В. Косточка

В настоящей заметке слово «граф» обозначает ориентированный граф без петель и кратных дуг, слово «цикл» — контур без самопересечений, слово «путь» — ориентированный путь. Мы говорим что два цикла пересекаются, если они имеют общие вершины. Если G-граф, то через $V(G)$ обозначается множество вершин G, через $E(G)$ — множество дуг G. Если G-граф, $V_0 \subset V(G)$, то через $G(V_0)$ обозначим подграф графа G, порожденный множеством вершин V_0.

Сильно связный граф назовем (ξ)-графом, если любые два его цикла пересекаются. Сильно связный граф G назовем (ε)-графом, если существует такая вершина $x \in V(G)$, что для любой дуги $\vec{e} \in E(G)$ найдется цикл, проходящий через x и $\vec{e}$. Остальные обозначения и термины, использованные в заметке, общеприняты.

В [1] А. Адам поставил следующую задачу (см. проблему 3):

Существует ли (ξ)-граф, не являющийся (ε)-графом?

В настоящей работе мы отвечаем на этот вопрос отрицательно.

Утверждение. Каждый (ξ)-граф является (ε)-графом.

Доказательство проведем индукцией по числу дуг. Для (ξ)-графов с двумя и тремя дугами утверждение очевидно.

Пусть G — наименьший по числу дуг (ξ)-граф, не являющийся (ε)-графом.

Случай 1. Для любой дуги $\vec{e} \in E(G)$ $G \setminus \vec{e}$ сильно связен. Тогда

$$|E(G)| \geqq |V(G)| + 1. \tag{1}$$

Кроме того, в этом случае для каждой дуги $\vec{e} \in E(G)$ $G \setminus \vec{e}$ (ξ)-граф и, по индукционному предположению, (ε)-граф, т.е. найдется такая вершина $x_{\vec{e}} \in V(G)$, что для любой дуги $\vec{e}' \in E(G) \setminus \vec{e}$ существует цикл, проходящий через $x_{\vec{e}}$ и $\vec{e}'$. Если бы для какой-либо дуги $\vec{e}_0 \in E(G)$ нашелся цикл C_0, проходящий через $x_{\vec{e}_0}$ и $\vec{e}_0$, то утверждение справедливо. Допустим, что такой дуги нет. Тогда для любых двух дуг $\hat{e}_1, \hat{e}_2 \in E(G)$ $(\hat{e}_1 \neq \hat{e}_2)$

$$x_{\hat{e}_1} \neq x_{\hat{e}_2},$$

т.е. $|V(G)| \geqq |E(G)|$, в противоречие с (1).

Случай 2. Существует такая дуга $\overrightarrow{(a, b)} \in E(G)$, что $G \setminus \overrightarrow{(a, b)}$ не сильно связен.

Обозначим через V_1 множество вершин графа $G \setminus \overrightarrow{(a, b)}$, достижимых из a, $V_2 = V(G) \setminus V_1$. Так как $a \in V_1, b \in V_2$, то $V_1 \neq \emptyset$, $V_2 \neq \emptyset$. В силу того, что в G каждые два цикла пересекаются, либо в $G(V_1)$, либо в $G(V_2)$ нет циклов. Без ограничения общности считаем, что циклов нет в $G(V_2)$ (иначе изменим ориентацию всех дуг). Обозначим $A = \{v \in V_1 | \exists w \in V_2 \colon \overrightarrow{(w, v)} \in E(G)\}$. Допустим, что $a \in A$, т.е. существует такая вершина $v \in V_2$, что $\overrightarrow{(v, a)} \in E(G)$. Так как G сильно связен, то найдется цикл C, проходящий через $\overrightarrow{(v, a)}$. Понятно, что $\overrightarrow{(a, b)} \in E(C)$. Следовательно,

$$V(C) \cap V_1 = \{a\}. \tag{2}$$

Так как $G - (\xi)$-граф, то каждый цикл в $G(V_1)$, в силу (2), должен проходить через a. Но любой цикл в G проходящий хотя бы через одну вершину из V_2, необходимо проходит через $\overrightarrow{(a, b)}$. Т.е. a — панциклическая вершина. Противоречие. Таким образом, $a \notin A$.

Построим G_1 по следующим правилам:

$$V(G_1) = V_1;\ E(G_1) = E(G(V_1)) \cup \{\overrightarrow{(a, v)} | v \in A \ \&\ \overrightarrow{(a, v)} \notin E(G(V_1))\}.$$

Докажем два свойства графа G_1.

Свойство 1. G_1 сильно связен.

Действительно, выберем произвольную упорядоченную пару $\langle v_1, v_2 \rangle$ вершин в G_1. Если в графе G из v_1 в v_2 вёл путь, не проходящий через $\overrightarrow{(a, b)}$, то этот же путь ведет из v_1 в v_2 и в графе G_1. Если же в G из v_1 в v_2 вёл путь, проходящий через $\overrightarrow{(a, b)}$ и заходящий из V_2 в V_1 по дуге $\overrightarrow{(b', a')}$, то часть этого пути, задевающую вершины из V_2, заменим в G_1 дугой $\overrightarrow{(a, a')}$.

Свойство 2. Каждые два цикла в G_1 пересекаются.

Допустим, что в G_1 есть два непересекающихся цикла C_1 и C_2.

Если $E(C_1) \cup E(C_2) \subset E(G) \cap E(G_1)$, то они должны пересечься, т.к. $G - (\xi)$-граф. Следовательно, один из этих циклов проходит через вершину a, а другой нет. Пусть, для определённости, $E(C_1) \subset E(G_1) \cap E(G)$, $E(C_2) \setminus E(G) = \{\overrightarrow{(a, a')}\}$, где $a' \in A$. Тогда найдется такая вершина $b' \in V_2$, что $\overrightarrow{(b', a')} \in E(G)$. В G найдется путь P, ведущий из a в b'. Понятно, что $\overrightarrow{(a, b)} \in E(P)$. Тогда в G существует цикл $\tilde{C}_2$, такой что $E(\tilde{C}_2) = (E(C_2) \setminus \{\overrightarrow{(a, a')}\}) \cup E(P) \cup \{\overrightarrow{(b', a')}\}$, не пересекающийся с C_1. Противоречие.

Таким образом, согласно свойствам 1 и 2, граф G_1 суть (ξ)-граф и, по индукционному предположению, (ε)-граф. Т.е. найдется такая вершина $x \in V(G_1)$, что для любой дуги $\overrightarrow{(v_1, v_2)} \in E(G_1)$ в G_1 существует цикл C, проходящий через x и $\overrightarrow{(v_1, v_2)}$. Покажем, что x обладает тем же свойством и в G.

Подслучай 2.1. $\overrightarrow{(v_1, v_2)} \in E(G) \cap E(G_1)$. Если цикл C в G_1, проходящий через x и $\overrightarrow{(v_1, v_2)}$, содержит дуги лишь из $E(G) \cap E(G_1)$, то этот цикл нам и нужен. Допустим, что C содержит дугу $\overrightarrow{(a, w)} \in E(G_1) \setminus E(G)$. Так как $w \in A$, то в G найдется путь P, ведущий из b в w, все вершины которого, за исключением w, принадлежат V_2. Тогда цикл C', такой что $E(C') = (E(C) \setminus \{\overrightarrow{(a, w)}\}) \cup \{\overrightarrow{(a, b)}\} \cup E(P)$, будет искомым для x и $\overrightarrow{(v_1, v_2)}$.

Подслучай 2.2. $\overrightarrow{(v_1, v_2)} \in E(G) \setminus E(G_1)$. Так как G сильно связен, то существует цикл C, проходящий через $\overrightarrow{(v_1, v_2)}$ и $\overrightarrow{(a, b)}$. Обозначим дугу этого цикла, ведущую из V_2 в V_1, через $\overrightarrow{(w, u)}$, а отрезок этого цикла, ведущий из a в u, через P. По определению вершины x, в G_1 найдется цикл C_1, проходящий через x и $\overrightarrow{(a, u)}$. Тогда цикл C_2, такой что $E(C_2) = (E(C_1) \setminus \{\overrightarrow{(a, u)}\}) \cup E(P)$ — искомый цикл для x и $\overrightarrow{(v_1, v_2)}$.

Abstract

It is proved that, if each pair of (directed) cycles of a strongly connected directed graph intersects each other (i.e., the cycles have at least one vertex in common), then a vertex x exists such that, for an arbitrary edge $\vec{e}$, there is a cycle in the graph which contains both x and $\vec{e}$

ИНСТИТУТ МАТЕМАТИКИ СО АН СССР
УНИВЕРСИТЕТСКИЙ ПРОСПЕКТ 4
630090 НОВОСИБИРСК, СССР

Литература

[1] Ádám, A., On some open problems of applied automaton theory and graph theory (suggested by the mathematical modelling of certain neuronal networks), *Acta Cybernet.*, v. 3, 1977, pp. 187—214.

(Поступило 21-ого января 1982 г.)

Probability model for non-homogeneous multiprogramming computer system

By J. SZTRIK

1. Introduction

In this paper we deal with a special queueing problem, which is of considerable practical importance in the field of computer applications. A new mathematical model for FIFO multiprogramming system is given in the following way.

A number of n jobs are permitted simultaneous acces to the resources of the system in such a way that the Central Processor Unit (CPU) is busy processing one job while various input-output (I/O) peripheral units (e.g. a rotating disk memory, a swapping drum, a magnetic tape unit, a data cell, a card reader and so on) are processing some of the others. In such a multiprogrammed enviroment jobs do indeed circulate among these devices in such a way that they require the attention of the CPU followed by the need for some peripheral units, and such a cycle is repeated several times.

We assume that the CPU services the jobs according to FIFO (first in-first out) discipline. In addition the system is supposed to have enough peripheral device, so no queueing for I/O operation occurs and a waiting line can be formed at the CPU only.

The jobs are assumed to be stochastically different; the i-th programm is characterised by exponentially distributed I/O time with rate λ_i and CPU time with an arbitrary distribution function $F_i(x)$. The I/O and compute times for all jobs are mutually independent. Similar models with less complexity were discussed by GAVER [5], TOMKÓ [3].

The interested reader is referred to KLEINROCK [1, 2] for further models and a good bibliography on this subject.

The purpose of the present paper is to give the main characteristics of the system in steady state, namely CPU productivity, expected CPU busy period, mean response time of the jobs. Finally numerical example illustrates the problem in question.

2. Description of the model

Let the random variable (abbreviated by r.v.) $\nu(t)$ denote the number of jobs at the CPU at time t and let $(\alpha_1(t), \ldots, \alpha_{\nu(t)}(t))$ indicate their indices in the order of their arrival.

Introduce the process

$$\mathbf{Y}(t) = (\nu(t), \alpha_1(t), \ldots, \alpha_{\nu(t)}(t)).$$

The stochastis process $(\mathbf{Y}(t), t \geqq 0)$ is not Markovian unless the distribution functions $F_i(x)$ are exponential, $i = 1, 2, \ldots, n$.

Let the r.v. ξ_t denote the attained compute time the job under service has got till time t.

Putting

$$\mathbf{X}(t) = (\nu(t), \alpha_1(t), \ldots, \alpha_{\nu(t)}(t); \xi_t) \tag{1}$$

the process $(\mathbf{X}(t), t \geqq 0)$ has already the Markov property.

Let V_k^n denote the set of all variations of order k of the integers $1, 2, \ldots, n$ ordered lexicographically. The points $(i_1, \ldots, i_k; x)$ form the state space of process (1), where $(i_1, \ldots, i_k) \in V_k^n$, $x \in \mathbf{R}_+$, $1 \leqq k \leqq n$. The process is in state $(i_1, \ldots, i_k; x)$ if k jobs need the CPU their indices in the order of arrival are $(i_1, \ldots, i_k)$ and the attained service time of job i_1 is x. The state when the CPU is idle is denoted by $\{0\}$.

In order to derive the Chapman—Kolmogorov equations we should consider the transitions that can occur in an arbitrary time interval $(t, t+\Delta t)$. The transition probabilities are the following.

$$P\{\mathbf{x}(t+\Delta t) = (i_1, \ldots, i_k; x+\Delta t)/\mathbf{x}(t) = (i_1, \ldots, i_k; x)\} =$$

$$= (1 - \sum_{j \neq i_1, \ldots, i_k} \lambda_j \Delta t) \frac{1 - F_{i_1}(x+\Delta t)}{1 - F_{i_1}(x)} + o(\Delta t), \tag{i}$$

$$P\{\mathbf{x}(t+\Delta t) = (i_1, \ldots, i_{k+1}; x+\Delta t)/\mathbf{x}(t) = (i_1, \ldots, i_k; x)\} =$$

$$= \lambda_{i_{k+1}} \Delta t \frac{1 - F_{i_1}(x+\Delta t)}{1 - F_{i_1}(x)} + o(\Delta t), \tag{ii}$$

$$P\{\mathbf{x}(t+\Delta t) = (i_2, \ldots, i_k; 0)/\mathbf{x}(t) = (i_1, \ldots, i_k; x)\} =$$

$$= \frac{F_{i_1}(x+\Delta t) - F_{i_1}(x)}{1 - F_{i_1}(x)} + o(\Delta t). \tag{iii}$$

Let us introduce some further notations.

$$\Lambda_{i_1, \ldots, i_k} = \sum_{j \neq i_1, \ldots, i_k} \lambda_j, \quad \Lambda = \sum_{j=1}^{n} \lambda_j,$$

$$S_{i_1, \ldots, i_k} = \sum_{j=1}^{k} \lambda_{i_j}, \quad \beta_i = \int_0^\infty x \, dF_i(x), \quad \Phi(s; i) = \int_0^\infty e^{-sx} \, dF_i(x),$$

$$\Pi_l^{(i_1, \ldots, i_k)} = \prod_{r=l+1}^{k} \lambda_{i_r} \Big/ \prod_{q=l+1}^{k} S_{i_{l+1}, \ldots, i_q}, \quad 1 \leqq l \leqq k, \quad 1 \leqq k \leqq n.$$

For the distribution of $\mathbf{x}(t)$ consider the functions given below.

$$P_0(t) = P(\nu(t) = 0),$$

$$P_{i_1,\dots,i_k}(x, t) = P(\nu(t) = k,\ \alpha_1(t) = i_1,\ \dots,\ \alpha_k(t) = i_k;\ \xi_t \leqq x),$$

$$(i_1, \dots, i_k) \in V_k^n, \quad k = 1, 2, \dots, n.$$

Theorem 1. If $\beta_i < \infty$ $(i=1, 2, \dots, n)$ then the limits

$$P_0 = \lim_{t\to\infty} P_0(t)$$

$$P_{i_1,\dots,i_k}(x) = \lim_{t\to\infty} P_{i_1,\dots,i_k}(x;\ t).$$

exist and satisfy the equation

$$P_0 + \sum_{k=1}^{n} \Sigma_{V_k^n} \lim_{x\to\infty} P_{i_1,\dots,i_k}(x) = 1$$

Proof. Note that $(\mathbf{x}(t), t \geqq 0)$ is a linear Markov process treated in GNEDENKO—KOVALENKO [7] in details. Our statement follows from a theorem on page 211 of this monograph.

Our next task is to give a procedure to determine the ergodic probabilities

$$(P_0, P_{i_1,\dots,i_k}), \quad (i_1, \dots, i_k) \in V_k^n, \quad k = 1, 2, \dots, n.$$

To do so, first of all we show that the ergodic functions

$$P_{i_1,\dots,i_k}(x), \quad (i_1, \dots, i_k) \in V_k^n, \quad x \in \mathbf{R}_+, \quad k = 1, 2, \dots, n$$

are differentiable at common continuity points of $F_i(x)$.

Then we introduce the so-called normed density functions:

$$p^*_{i_1,\dots,i_k}(x) = \frac{\dfrac{d}{dx} P_{i_1,\dots,i_k}(x)}{1 - F_{i_1}(x)}. \tag{2}$$

We derive a system of integro-differential equations for these functions, and by the help of its solution we can give an algorithm for calculating the stationary distribution.

Let $V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(\tau)$ denote the probability of the event that at an arbitrary instant jobs with indices $(i_1, \dots, i_r)$ are in compute period and from this epoch during a period of time τ additional jobs $(i_{r+1}, \dots, i_k)$ finish their I/O operation in this order. One can readily verify that

$$V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(\tau) = e^{-\Lambda_{i_1,\dots,i_k}\tau}.$$

$$\int\limits_{0<z_1<z_2<\dots<z_{k-r}} \lambda_{i_{r+1}} e^{-\lambda_{i_{r+1}} z_1} \dots \lambda_{i_k} e^{-\lambda_{i_k} z_{k-r}} \, dz_1 \dots dz_{k-r},$$

which can be expressed by the help of exponential functions. In the homogeneous case $(\lambda_1 \equiv \lambda)$

$$V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(\tau) = \frac{1}{(k-r)!}(1-e^{-\lambda\tau})^{k-r} e^{-(n-k)\lambda\tau}.$$

Now we prove the following theorem.

Theorem 2. The ergodic distribution function $P_{i_1,\dots,i_k}(x)$ possesses density function $p_{i_1,\dots,i_k}(x)$, $(i_1, \dots, i_k) \in V_k^n$, $1 \leqq k \leqq n$, and at almost every $x \in \mathbf{R}_+$. In addition, the normed d.f.

$$p^*_{i_1,\dots,i_k}(x) = \frac{p_{i_1,\dots,i_k}(x)}{1-F_{i_1}(x)}$$

is differentiable at every $x \in \mathbf{R}_+$.

Proof. We first prove the existance of densities $p_{i_1,\dots,i_k}(x)$. Let the $(\mathbf{x}(t), t \geqq 0)$ be in state $(i_1, \dots, i_k;\ x)$ at an arbitrary t. The process is in this state iff some epoch u, $t-x<u<t$, the CPU completes a computation and immediately starts servicing job i_1. If the indices of tasks in compute period are $(i_1, \dots, i_r)$ at time u then the unexpired service time of i must excess $t-u$ and during the time interval (u, t) jobs $(i_{r+1}, \dots, i_k)$ should arrive in this order. For the sake of easier understanding we notice that the process $\mathbf{x}(t)$ is of regenerative type.

The regenerative periods can be defined in several ways. Let us consider for example the epochs when the CPU completes a computation and starts to serve the programm i_1 while the others with indices $(i_2, \dots, i_r)$ are already waiting for their turn. If the initial state $(j_1, \dots, j_s;\ z)$ differs from $(i_1, \dots, i_r;\ 0)$ then the renewal process in question is a so-called delayed one.

Let us denote by $H_{j_1,\dots,j_s,z}^{i_1,\dots,i_r}(t)$ the renewal function of the process considered above.

Denote by

$$(R_0, R_{j_1,\dots,j_s}(z);\ z \geqq 0,\ (j_1, \dots, j_s) \in V_s^n,\ s = 1, 2, \dots, n)$$

the initial distribution of $(\mathbf{x}(t), t \geqq 0)$. Keeping in mind the behaviour of the recurrent process, by using the theorem of total probability, we get

$$P_{i_1,\dots,i_k}(x, t) = \left(\sum_{s=1}^{n} \Sigma_{V_s^n} \int_0^t dR_{j_1,\dots,j_s}(z) + R_0\right) \cdot$$

$$\cdot \sum_{r=1}^{k} \int_{t-x}^{t} V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(t-u)[1-F_{i_1}(t-u)]\, dH_{j_1,\dots,j_s;z}^{i_1,\dots,i_r}(u).$$

Applying the key renewal theorem of Smith we have

$$\lim_{t\to\infty} P_{i_1,\dots,i_k}(x, t) = P_{i_1,\dots,i_k}(x) = \sum_{r=1}^{n} \frac{1}{m_{i_1,\dots,i_r}} \int_0^x [1-F_{i_1}(u)] V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(u)\, du,$$

where $m_{i_1,\dots,i_r}$ denotes the expected recurrence time into state $(i_1, \dots, i_r, 0)$ which is finite since the process is ergodic. The functions $V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(\tau)$ can be expressed with the aid of exponential ones hence the d.f. $P_{i_1,\dots,i_k}(x)$ is differentiable at every continuity point of $F_{i_1}(x)$.

This implies that the density f. $p_{i_1,\dots,i_k}(x)$ is defined at almost everywhere and

$$p_{i_1,\dots,i_k}(x) = \sum_{r=1}^{k} \frac{1}{m_{i_1,\dots,i_r}} V_{i_1,\dots,i_r}^{i_1,\dots,i_k}(x)[1-F_{i_1}(x)].$$

Therefore the normed functions are differentiable at $x \in \mathbf{R}_+$.

Theorem 3. The stationary density f of the process $(\mathbf{X}(t), t \geqq 0)$ satisfies the following system of differential equations

$$\begin{aligned} &\frac{dp_{i_1}^*(x)}{dx} + \Lambda_{i_1} p_{i_1}^*(x) = 0, \\ &\vdots \\ &\frac{dp_{i_1,\dots,i_k}^*(x)}{dx} + \Lambda_{i_1,\dots,i_k} p_{i_1,\dots,i_k}^*(x) = \lambda_{i_k} p_{i_1,\dots,i_{k-1}}^*(x), \\ &\vdots \\ &\frac{dp_{i_1,\dots,i_n}^*(x)}{dx} = \lambda_{i_n} p_{i_1,\dots,i_{n-1}}^*(x). \end{aligned} \tag{3}$$

The boundary conditions are

$$\begin{aligned} &\Lambda P_0 = \sum_{j=1}^{n} \int_0^\infty p_j^*(x)\, dF_j(x), \\ &\vdots \\ &p_{i_1,\dots,i_k}(0) = \sum_{j \neq i_1,\dots,i_k}' \int_0^\infty p_{j,i_1,\dots,i_k}^*(x)\, dF_j(x), \quad p_{i_1,\dots,i_n}(0) = 0. \end{aligned} \tag{4}$$

Proof. Considering the transition probabilities, using the equations of Chapman—Kolmogorov the first part of the theorem can easily be verified. The second part can be proved by the theorem of total probability. The details of the proof is to be found in SZTRIK (6).

It is quite easy to see that the solution of (3)—(4) is

$$p_{i_1,\dots,i_k}^*(x) = \sum_{l=1}^{k} (-1)^{k-l} c_{i_1,\dots,i_l} e^{-\Lambda_{i_1,\dots,i_l} x} \Pi_l^{(i_1,\dots,i_k)},$$

$$(i_1, \dots, i_k) \in V_k^n, \quad k = 1, 2, \dots, n,$$

where the constants $C_{i_1,\dots,i_l}$ are to be determined from the boundary condition (4).

In the following we describe an iterative method to calculate these coefficients. Let $\mathbf{c}_k$ denote the vector

$$\begin{vmatrix} c_{1,2,\dots,k} \\ \vdots \\ c_{i_1,\dots,i_k} \\ \vdots \\ c_{n,\dots,n-k+1} \end{vmatrix}$$

of dimension $\binom{n}{k}k!$. The components of $\mathbf{c}_k$ are listed in the lexicographic order of their indices, $k=1,\dots,n$. Notice, that boundary condition $p_{i_1,\dots,i_n}(0)=0$ is equivalent to equation

$$\mathbf{c}_n = A_{n-1}^{(n)}\mathbf{c}_{n-1}+\dots+A_1^{(n)}\mathbf{c}_1,$$

with a suitably chosen matrix $A_k^{(n)}$ of order $n!x\binom{n}{k}k!$. The k-th boundary condition, where $2\leqq k\leqq n-1$, gives the relation

$$\sum_{l=1}^{k}(-1)^{k-l}c_{i_1,\dots,i_l}\Pi_l^{(i_1,\dots,i_k)} = \sum_{j\neq i_1,\dots,i_k}$$
$$\sum_{l=0}^{k}(-1)^{k-l}C_{j,i_1,\dots,i_l}\Pi_l^{(i_1,\dots,i_k)}\int_0^\infty e^{-A_{j,i_1,\dots,i_l}x}\,dF_j(x).$$

In term of the Laplace—Stieltjes transform this becomes

$$\sum_{l=1}^{k}(-1)^{k-l}c_{i_1,\dots,i_l}\Pi_l^{(i_1,\dots,i_k)} = \sum_{j\neq i_1,\dots,i_k}$$
$$\sum_{l=0}^{k}(-1)^{k-l}c_{j,i_1,\dots,i_l}\Pi_l^{(i_1,\dots,i_k)}\Phi(A_{j,i_1,\dots,i_l};j).$$

More succintly

$$\mathbf{c}_k = A_{k+1}^{(k)}\mathbf{c}_{k+1}+\dots+A_1^{(k)}\mathbf{c}_1.$$

Now we are ready to define our algorithm. We have

$$\mathbf{c}_n = \sum_{j=1}^{n-1}A_j^{(n)}\mathbf{c}_j,$$
$$\mathbf{c}_{n-1} = \sum_{j=1}^{n-2}B_j^{(n-1)}\mathbf{c}_j, \tag{$*$}$$

where the matrix $B_j^{(n-1)}$ is defined by

$$B_j^{(n-1)} = (1-A_n^{(n-1)}A_{n-1}^{(n)}-A_{n-1}^{(n-1)})^{-1}(A_n^{(n-1)}A_j^{(n)}+A_j^{(n-1)}),\quad 1\leqq j\leqq n-2.$$

Similarly

$$\mathbf{c}_k = \sum_{j=1}^{k-1}B_j^{(k)}\mathbf{c}_j, \tag{$**$}$$

where the matrix $B_j^{(k)}$ is given by

$$B_j^{(k)} = (1-A_{k+1}^{(k)}B_k^{(k+1)}-A_k^{(k)})^{-1}(A_{k+1}^{(k)}B_j^{(k+1)}+A_j^{(k)})$$
$$2\leqq k\leqq n-1,\quad 1\leqq j\leqq k-1.$$

For $\mathbf{c}_1$ we have the equation

$$\mathbf{c}_1 = A_2^{(1)}\mathbf{c}_2+A_1^{(1)}\mathbf{c}_1+P_0\begin{pmatrix}\lambda_1\\ \vdots\\ \lambda_n\end{pmatrix}.$$

Using the formula for C_2 we obtain

$$(1-A_2^{(1)}B_1^{(2)})\mathbf{c}_1 = P_0\begin{pmatrix}\lambda_1\\ \vdots\\ \lambda_n\end{pmatrix}.$$

Hence

$$\mathbf{c}_1 = (1-A_2^{(1)}B_1^{(2)})^{-1}\begin{pmatrix}\lambda_1\\ \vdots\\ \lambda_n\end{pmatrix}P_0. \qquad (***)$$

Starting with an arbitrary P_0 and using the relations $(*), (**), (***)$ we can determine the vectors $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n$ (in this order). Following this procedure we obtain all constants and also the density functions $p_{i_1,\ldots,i_k}(x)$.

Let us denote by $P_{i_1,\ldots,i_k}$ the stationary probability that jobs with indices $(i_1, \ldots, i_k)$ are in compute period. Apparently,

$$P_{i_1,\ldots,i_k} = \int_0^\infty p_{i_1,\ldots,i_k}(x)\,dx = \int_0^\infty p^*_{i_1,\ldots,i_k}(x)[1-F_{i_1}(x)]\,dx.$$

Denote by P_k the steady state probability that k programm is at the CPU. We have

$$P_k = \Sigma_{V_k^n} P_{i_1,\ldots,i_k}.$$

The value of P_0 can be calculated from the normalising condition

$$P_0 + \sum_k P_k = 1.$$

3. Utility investigations

(i) CPU utilization.

It is easy to see that the CPU's activity can be divided into two periods, viz. idle and busy ones. Together they form a cycle. The durations of these cycles are independent and identically distributed random variables.

By the virtue of a renewal consideration it follows that

$$P_0 = \frac{1/\Lambda}{1/\Lambda + M\delta}$$

where $M\delta$ denotes the mean CPU busy period and $1/\Lambda$ is the average idle period length.

If U_{CPU} denotes the CPU productivity, which is the long-run fraction of time the CPU is busy, then

$$U_{\text{CPU}} = \frac{M\delta}{1/\Lambda + M\delta}$$

Consequently

$$M\delta = (1-P_0)/(\Lambda P_0).$$

(ii) Mean waiting times.

During the execution a job waits for the CPU, occupies it and takes I/O operations. If one considers these periods as a cycle, then in equilibrium for a fixed job these cycles have identical distribution, but they are not independent.

Let $P^{(i)}$ denote the steady state probability that job i is in compute period and let the average period lengths designated by W_i, β_i, $1/\lambda_i$, respectively.

Consider now the semi-Markov process $(\mathbf{Y}(t), t \geqq 0)$, with state space $\bigcup_{k=1}^{n} V_k^n + \{0\}$.

Let H_i be the event that the programm i does not take I/O operations. Introduce the function

$$Z_{H_i}(t) = \begin{cases} 1 & \text{if } \mathbf{Y}(t) \in H_i, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 4.

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T Z_{H_i}(t)\,dt = \frac{W_i+\beta_i}{1/\lambda_i + W_i + \beta_i} = P^{(i)}.$$

The statement is a special case of a theorem concerning mean sojourn time for semi-Markov, processes, see Tomkó [4] on page 297. Since the probability $P^{(i)}$ can be calculated from the distribution $P_{i_1,\ldots,i_k}$, by

$$P^{(i)} = \sum_{k=1}^{n} \Sigma_{V_k^n,\, i \in (i_1,\ldots,i_k) \in V_k^n} P_{i_1,\ldots,i_k},$$

the expected waiting time of job i is

$$W_i = P^{(i)}/[\lambda_i(1-P^{(i)})] - \beta_i.$$

4. Numerical results

We shall deal with only the case $n=4$ because the size of matrices involved in the iteration grows rapidly by increasing n. We assume that the I/O times are identically distributed with parameter $\lambda=1.2$ and the compute times are mixtures of Erlangian distributions, namely

$$F(1, x) = E(1, 1.2, x),$$
$$F(2, x) = 0.8E(2, 3.3, x)+0.2E(2, 0.9, x),$$
$$F(3, x) = 0.2E(2, 3.5, x)+0.6E(3, 2.6, x)+0.2E(1, 0.2, x),$$
$$F(4, x) = 0.8E(2, 3.3, x)+0.2E(2, 0.9, x).$$

$E(k, \lambda, x)$ indicates the k-stage Erlangian distribution with parameter λ. So the stationary probability that the i-th job is at the CPU, $i=1, 2, 3, 4$, and that the CPU is idle are the following:

$P^{(1)}=0.8110664463$, $P^{(2)}=0.8130187511$, $P^{(3)}=0.8209813440$,
$P^{(4)}=0.8130187511$, $P_0=0.005945$.

Finally, the main characteristics are:
CPU utilization$=0.994055$,
mean waiting times:

$W(1)=3.5773884090$, $W(2)=3.623441546$, $W(3)=3.821674988$, $W(4)=3.623441546$

Acknowledgment. I am very grateful to Dr. J. Tomkó for his helpful discussion.

Abstract

The aim of the present paper is to give a more adequate model for FIFO multiprogramming computer systems.

The jobs are stochastically different, programm i is characterized by exponentially distributed I/O time with rate λ_i and CPU time with an arbitrary distribution function $F_i(x)$.

In stationary case we deal with CPU utilization, mean actual waiting times of the jobs.

Finally numerical examples illustrate the problem in question.

KEYWORDS: I/O time, CPU time, utilization, mean response time.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF DEBRECEN
PF. 12., DEBRECEN, HUNGARY
H—4010

References

[1] KLEINROCK, L., *Sorbanállás-kiszolgálás*, Műszaki Könyvkiadó, Budapest, 1979.
[2] KLEINROCK, L., *Queueing systems*, vol. 2, Wiley-Interscience, New York, 1976.
[3] TOMKÓ, J., Processor utilization study, *Alkalmaz. Mat. Lapok*, v. 1, 1975, pp. 319—331; v. 3, 1977, pp. 83—96.
[4] TOMKÓ, J., On sojourn times for semi-Markov processes, 14-th European Meeting of Statisticians, Wroclaw, Poland, 1981.
[5] GAVER, D. P., Probability models for multiprogramming computer systems, *J. Assoc. Comput. Mach.*, v. 3, 1967, pp. 423—438.
[6] SZTRIK, J., On machine interference, Ph. D. Thesis, University of Debrecen, 1980.
[7] GNEDENKO, B. V., I. N. KOVALENKO, Lectures on Queueing Systems, Kiev, 1963, (in Russian).

(Received Dec. 3, 1981)

Das Prinzip der maximalen Entropie und seine Anwendung in der Codierungs- und Suchtheorie

Von TH. FISCHER

1. Einführung

Das von E. T. Jaynes ursprünglich für die statistische Mechanik begründete Prinzip der maximalen Entropie hat zwar vielfältige Anwendungen gefunden, ist aber bis heute nicht unumstritten. Einen Überblick sowohl über die verschiedenen Anwendungsgebiete als auch über die Schwierigkeiten, die sich bei der Begründung dieses Prinzips ergeben, findet man in [13]. In dieser Arbeit wird nun eine Ungleichung bewiesen, durch die die Verwendung der entropiemaximierenden Verteilung für gewisse Anwendungsbereiche eine neue Rechtfertigung erfährt.

Es sei p eine Wahrscheinlichkeitsverteilung über der endlichen Menge X. Mit

$$H(p) =_{\mathrm{Df}} \sum_{x \in X} p(x) \log p(x)$$

sei die Entropie der Verteilung p bezeichnet. Aus der Codierungstheorie ist gut bekannt, daß für jede weitere Verteilung q über X die Funktion

$$H(p, q) =_{\mathrm{Df}} \sum_{x \in X} p(x) \log q(x)$$

stets größer oder gleich $H(p)$ ist. Im Gegensatz zur Entropie besitzt die Funktion $H(p, q)$ auch keine obere Schranke. Wenn man jedoch eine konvexe und kompakte Menge P von Verteilungen über X fest vorgibt, läßt sich stets eine Verteilung q so finden, daß für alle $p \in P$ die Abschätzung

$$H(p, q) \leqq \max_{p \in P} H(p) \tag{1}$$

gilt.

Diese Ungleichung besitzt nicht nur theoretische Bedeutung. In der Codierungstheorie interessiert man sich für die Existenz von Codes, die nicht nur für eine spezielle Verteilung, sondern für eine ganze Klasse von Verteilungen gleichermaßen geeignet sind, vgl. [2]. Die Ungleichung (1) eröffnet eine besonders einfache Möglichkeit, derartige universelle Codes zu erhalten. Wendet man nämlich bestimmte Konstruktionsalgorithmen nicht auf die Quellverteilung p, sondern auf eine andere

Verteilung q an, dann läßt sich die Güte des erhaltenen Codes mit Hilfe der Funktion $H(p, q)$ charakterisieren [3]. Durch Anwendung bekannter Konstruktionsalgorithmen auf eine entropiemaximierende Verteilung lassen sich also gute universelle Codes gewinnen. Ähnliches gilt für die Konstruktion universeller Suchbäume. Dabei ist besonders wichtig, daß dies auch für kleine Blocklängen gilt, für große Blocklängen liefern dagegen asymptotische Methoden bessere Ergebnisse [2].

Im nachfolgenden Abschnitt wird zunächst eine Verallgemeinerung der Ungleichung (1) für die von A. Rényi eingehend untersuchte Entropie der Ordnung α bewiesen. Anschließend werden dann die schon angedeuteten Anwendungen dieses Ergebnisses auf verschiedene Probleme der Codierungs- und Suchtheorie eingehend dargelegt.

2. Eine Ungleichung für die α-Entropie

Es sei X eine endliche nichtleere Menge und p eine Wahrscheinlichkeitsverteilung über X. Für jeden reellen Parameter α, $0<\alpha\leqq 1$, und jede Verteilung p sei die Entropie der Ordnung α (α-Entropie) als

$$H_\alpha(p) =_{\mathrm{Df}} \begin{cases} H(p) & \text{für} \quad \alpha = 1 \\ \dfrac{1}{1-\alpha} \log \sum_{x\in X} p(x)^\alpha & \text{für} \quad \alpha\in (0,1) \end{cases}$$

erklärt. Als Logarithmenbasis ist dabei jede von 1 verschiedene positive reelle Zahl zugelassen. Wie üblich wird $0 \log 0=0$ gesetzt. Die Eigenschaften der α-Entropie wurden in [12] eingehend untersucht.

Von E.G. Nath stammt folgende Verallgemeinerung der α-Entropie für zwei Verteilungen p und q:

$$H_\alpha(p,q) =_{\mathrm{Df}} \begin{cases} H(p,q) & \text{für} \quad \alpha = 1 \\ \dfrac{1}{1-\alpha} \log \sum_{x\in X} p(x) q(x)^{\alpha-1} & \text{für} \quad \alpha\in (0,1). \end{cases}$$

Offensichtlich ist die Funktion $H_\alpha(p, q)$ ebenso wie die α-Entropie stets nichtnegativ, im Gegensatz zu dieser aber unbeschränkt nach oben. Setzt man jedoch voraus, daß mit $q(x)$ auch stets $p(x)$ verschwindet, dann ist $H_\alpha(p, q)$ endlich und stetig sowohl in p als auch in q. Weiterhin gilt stets $H_\alpha(p, q)\geqq H_\alpha(p)$, vgl. [3, 11], und für $q=p$ erhält man $H_\alpha(p,p)=H_\alpha(p)$.

Es sei jetzt P eine konvexe und kompakte Menge von Wahrscheinlichkeitsverteilungen über X und α sei beliebig aber fest gewählt. Ferner sei p_0 eine Verteilung, die die α-Entropie in P maximiert, d. h. es gilt $H_\alpha(p_0)=\max_{p\in P} H_\alpha(p)$.

Satz 1. Für alle $p\in P$ gilt $H_\alpha(p, p_0)\leqq H_\alpha(p_0)$.

Zum Beweis von Satz 1 wird der folgende Hilfssatz benötigt (R bezeichne dabei den Körper der reellen Zahlen):

Hilfssatz. G sei ein konvexes Gebiet im R^n und g sei eine in G definierte reellwertige Funktion. Dann ist $g(p)$ dann und nur dann eine konvexe Funktion des Variablenvektors $p\in G$, wenn die reelle Funktion $f(\lambda)=_{\mathrm{Df}} g(\lambda p'+(1-\lambda)p'')$, $0\leqq\lambda\leqq 1$, für alle $p', p''\in G$ konvex bezüglich λ ist.

Beweis. Es seien $\lambda_1, \lambda_2 \in [0, 1]$ und für beliebige $p', p'' \in G$ sei $p_i =_{\mathrm{Df}} \lambda_i p' + (1-\lambda_i)p''$, $i=1, 2$. Mit p' und p'' liegen auch p_1 und p_2 in G. Wenn g konvex in G ist, erhält man für jedes $\varrho \in [0, 1]$ unter Beachtung der Gleichung $1 = \varrho + (1-\varrho)$

$$\begin{aligned} f(\varrho\lambda_1 + (1-\varrho)\lambda_2) &= g\big((\varrho\lambda_1 + (1-\varrho)\lambda_2)p' + (1-\varrho\lambda_1 - (1-\varrho)\lambda_2)p''\big) = \\ &= g(\varrho p_1 + (1-\varrho)p_2) \leqq \varrho g(p_1) + (1-\varrho)g(p_2) = \\ &= \varrho g(\lambda_1 p' + (1-\lambda_1)p'') + (1-\varrho)g(\lambda_2 p' + (1-\lambda_2)p'') = \\ &= \varrho f(\lambda_1) + (1-\varrho)f(\lambda_2), \end{aligned}$$

d. h. f ist konvex. Wenn hingegen f als konvex vorausgesetzt wird, erhält man mit $\lambda_1 = 1$, $\lambda_2 = 0$ die Ungleichung $f(\varrho) \leqq \varrho f(1) + (1-\varrho)f(0)$, woraus sich unter Beachtung der Definition von f unmittelbar die Konvexität von g ergibt.

Beweis von Satz 1. Für beliebige Verteilungen $p \in P$ sei

$$f(\lambda) =_{\mathrm{Df}} H_\alpha(\lambda p + (1-\lambda)p_0).$$

Da die α-Entropie eine konkave Funktion des Variablenvektors p ist, ist auch f konkav bezüglich λ. Da überdies $f(0) = H_\alpha(p_0) = \max_{p \in P} H_\alpha(p)$ gilt und alle Verteilungen $\lambda p + (1-\lambda)p_0$ wegen der Konvexität von P zu P gehören, erhält man für alle $\lambda \in [0, 1]$ die Ungleichung $f(\lambda) \leqq f(0)$. Da f konkav ist, ergibt sich daraus, daß f eine monoton fallende Funktion von λ ist. Für alle $\lambda \in [0, 1]$ gilt also $f'(\lambda) \leqq 0$.

Für $\alpha \in (0, 1)$ erhält man

$$f'(\lambda) = \frac{\alpha}{1-\alpha} \sum_{x \in X} \frac{(\lambda p(x) + (1-\lambda)p_0(x))^{\alpha-1}(p(x) - p_0(x)) \log e}{\sum_{x' \in X} (\lambda p(x') + (1-\lambda)p_0(x'))} \leqq 0.$$

Setzt man in dieser Ungleichung $\lambda = 0$, erhält man

$$\frac{\alpha}{1-\alpha} \sum_{x \in X} \frac{p_0(x)^{\alpha-1}}{\sum_{x' \in X} p_0(x')^\alpha} (p(x) - p_0(x)) \leqq 0,$$

wegen $\alpha/(1-\alpha) > 0$ and $\sum_{x' \in X} p_0(x')^\alpha > 0$ ergibt sich daraus

$$\sum_{x \in X} p(x) p_0(x)^{\alpha-1} \leqq \sum_{x \in X} p_0(x)^\alpha$$

und durch Logarithmierung und anschließende Multiplikation mit dem Faktor $1/(1-\alpha)$ erhält man die Behauptung von Satz 1.

Zum Beweis für den Fall $\alpha = 1$ bildet man ebenfalls die Ableitung von f. Es gilt

$$f'(\lambda) = -\sum_{x \in X} (p(x) - p_0(x)) \log((1-\lambda)p_0(x) + \lambda p(x)) \leqq 0.$$

Für $\lambda = 0$ folgt daraus $H(p, p_0) \leqq H(p_0)$.

Es bleibt noch zu zeigen, daß die Ableitung an der Stelle 0 auch tatsächlich existiert. Man sieht leicht ein, daß $f'(\lambda)$ für $\lambda = 0$ nur dann nicht definiert ist, wenn es ein $\bar{x}$ mit $p(\bar{x}) > 0$ und $p_0(\bar{x}) = 0$ gibt. O. B. d. A. sei $\bar{x}$ das einzige Element mit dieser Eigenschaft. Während alle übrigen Summanden einen endlichen Wert besitzen, strebt dann der in $f'(\lambda)$ vorkommende Ausdruck $-p(\bar{x})\log \lambda p(\bar{x})$ für $\lambda \to 0$ gegen ∞. Es gilt daher $\lim_{\lambda \to 0} f'(\lambda) = \infty$. Dies widerspricht aber der Tatsache, daß die Ableitung von f nicht positiv ist.

Wie in früheren Arbeiten gezeigt wurde, spielt auch die Funktion

$$H'_\alpha(p, q) =_{\mathrm{Df}} \alpha H_\alpha(p, q) + (1-\alpha) H_\alpha(q)$$

im Zusammenhang mit Codierungsproblemen eine wichtige Rolle [3, 4].

Folgerung 1. Für alle $p \in P$ gilt $H'_\alpha(p, p_0) \leqq H_\alpha(p_0)$.

Der Beweis dieser Ungleichung folgt unmittelbar aus Satz 1 und der Definition von $H'_\alpha(p, q)$.

3. Universelle längenvariable Codierungen

Mit $[X, p]$ sei jetzt eine diskrete gedächtnislose Quelle bezeichnet und c_n: $X^n \to W(Y)$ sei eine Codierung aller Wörter der Länge n in Wörter über einem anderen Alphabet Y. Die Länge eines Codeworts $c_n(u)$ wird mit $l(u)$ bezeichnet. In [1] wurde als Maß für den Aufwand einer Codierung c_n die (exponentielle) mittlere Codewortlänge

$$L_p^t(c_n) =_{\mathrm{Df}} \frac{1}{t} \log_b \sum_{u \in X^n} p(u) b^{tl(u)} \tag{2}$$

vorgeschlagen. Dabei bezeichnet b die Kardinalzahl von Y und t ist eine beliebige positive reelle Zahl. Für $t \to 0$ geht (2) in die lineare mittlere Codewortlänge

$$L_p(c_n) =_{\mathrm{Df}} \sum_{u \in X^n} p(u) l(u)$$

über, es ist daher gerechtfertigt $L_p^0(c_n) =_{\mathrm{Df}} L_p(c_n)$ zu setzen.

Es sei jetzt $t \geqq 0$ fest gewählt und $\alpha =_{\mathrm{Df}} 1/(1+t)$. Weiterhin sei eine beliebige Verteilung q über X vorgegeben, die durch die Festlegung $q(u) =_{\mathrm{Df}} q(x_1) \cdot \ldots \cdot q(x_n)$ für alle $u = x_1 \ldots x_n \in X^n$ zu einer Wahrscheinlichkeitsverteilung über X^n erweitert wird. Mit q_α wird die durch $q_\alpha(x) =_{\mathrm{Df}} q(x)^\alpha / \sum_{x \in X} q(x)^\alpha$, $x \in X$, definierte Hilfsverteilung bezeichnet, die sich ganz analog auf X^n ausdehnen läßt. Bekanntlich läßt sich stets eine eindeutig decodierbare Codierung c_n: $X^n \to W(Y)$ so finden, daß für alle Codewörter $c_n(u)$ die Beziehung

$$l(u) \leqq -\log_b q_\alpha(u) + 1 \tag{3}$$

erfüllt ist. Dies kann zum Beispiel durch Anwendung des bekannten Shannonschen Algorithmus auf die Verteilung q_α erreicht werden. Für die mittlere Codewortlänge (gemittelt mit der Quellverteilung p) erhält man dann

$$L_p^t(c_n) \leqq nH'_\alpha(p, q) + 1, \tag{4}$$

vgl. [5]. Für $t=0$ gilt offenbar $q_\alpha = q$ und man erhält

$$L_p(c_n) \leqq nH(p, q) + 1.$$

Es sei jetzt P eine konvexe und kompakte Menge von Wahrscheinlichkeitsverteilungen über X. Durch P wird eine ganze Klasse diskreter gedächtnisloser Quellen mit dem gemeinsamen Alphabet X festgelegt. Ferner sei eine Verteilung $p_0 \in P$ so gewählt, daß

$$H_\alpha(p_0) = \max_{p \in P} H_\alpha(p)$$

gilt. Ersetzt man nun die Verteilung q durch p_0, dann gilt wegen (4) für jede Verteilung p

$$L_p^t(c_n) \leqq nH'_\alpha(p, p_0)+1$$

und für alle $p \in P$ ergibt sich aus der Folgerung 1

$$L_p^t(c_n) \leqq nH_\alpha(p_0)+1.$$

Satz 2. Für jede konvexe und kompakte Klasse P von diskreten gedächtnislosen Quellen $[X, p]$ und jedes $t \geqq 0$ läßt sich eine Folge eindeutig decodierbarer Codierungen c_n: $X^n \to W(Y)$ derart konstruieren, daß für jede Verteilung $p \in P$ die Ungleichung

$$L_p^t(c_n) \leqq n \max_{p \in P} H_\alpha(p)+1$$

erfüllt ist.

Da für die Konstruktion der Codierung c_n nur die Kenntnis von P benutzt wurde, läßt sich c_n als universelle Codierung für P auffassen. Im Unterschied zu den üblicherweise betrachteten asymptotisch universellen Codierungen [2] gilt diese Eigenschaft für jedes n, insbesondere also auch für $n=1$. Dies ist wichtig, weil mit wachsendem n auch der Umfang des Codebuchs exponentiell anwächst. Überdies lassen sich die erwähnten asymptotischen Methoden auch nicht für die Konstruktion von Suchbäumen nutzen.

4. Universelle binäre Suchbäume

Als Maß für den Suchaufwand in einem binären Suchbaum B wird gewöhnlich die mittlere Weglänge

$$L_p(B) =_{\mathrm{Df}} \sum_{x \in X} p(x)\,l(x)$$

betrachtet. Dabei ist X eine den Endknoten von B zugeordnete endliche linear geordnete Menge, $l(x)$ bezeichnet die Weglänge (Anzahl der Kanten) von der Wurzel des Baums bis zu dem mit x markierten Endknoten und p ist eine Wahrscheinlichkeitsverteilung über X. Bekanntlich läßt sich zu jeder Zugriffsverteilung p ein binärer Suchbaum so konstruieren, daß für alle $x \in X$ die Ungleichung

$$l(x) \leqq -\log_2 p(x)+2 \tag{5}$$

gilt. Dies kann zum Beispiel mit den Algorithmen von Gilbert und Moore [9] oder Mehlhorn [10] geschehen.

Es sei nun wieder angenommen, daß von der konkret vorliegenden Zugriffsverteilung nur ihre Zugehörigkeit zu einer konvexen und kompakten Klasse P bekannt ist. Ferner sei p_0 so gewählt, daß $H(p_0) = \max_{p \in P} H(p)$ gilt. Wendet man die oben genannten Verfahren auf die Verteilung p_0 an, dann erhält man wegen [5] einen

Suchbaum B mit

$$L_p(B) \leqq H(p, p_0)+2.$$

Durch Anwendung von Satz 1 für $\alpha=1$ erhält man daraus unmittelbar das folgende Ergebnis:

Satz 3. Zu jeder konvexen und kompakten Klasse P von Zugriffsverteilungen läßt sich ein binärer Suchbaum B so konstruieren, daß für alle $p \in P$

$$L_p(B) \leqq \max_{p \in P} H(p)+2$$

gilt.

Es soll hier nur kurz erwähnt werden, daß Satz 3 auch für den Fall verallgemeinert werden kann, daß mit positiver Wahrscheinlichkeit nach Informationen gesucht wird, die nicht im Suchbaum abgespeichert sind. In diesem Fall kann dann nur der ohnehin günstigere Algorithmus von Mehlhorn verwendet werden, vgl. [5].

5. Universelle Blockcodierungen

In der klassischen Informationstheorie werden als Blockcodes Mengen der Form

$$C_n(p) =_{\mathrm{Df}} \{u/u \in X^n \wedge p(u) > \varepsilon\}$$

betrachtet, vgl. [8]. Dabei ist $\varepsilon > 0$ eine vorgegebene Schranke. Für konvexe und kompakte Klassen P von diskreten gedächtnislosen Quellen $[X, p]$ läßt sich ein für P universell geeigneter Blockcode ebenfalls mit Hilfe einer entropiemaximierenden Verteilung bestimmen. Es sei p_0 also wieder so gewählt, daß $H(p_0)=\max_{p \in P} H(p)$ gilt. Für jede reelle Zahl $R>0$ sei dann

$$C_n(p_0) =_{\mathrm{Df}} \{u/u \in X^n \wedge p_0(u) > 2^{-nR}\}.$$

Mit der durch

$$\varphi(u) =_{\mathrm{Df}} \begin{cases} 0 & \text{für} \quad u \in C_n(p_0) \\ 1 & \text{für} \quad u \in X^n \setminus C_n(p_0) \end{cases}$$

definierten Funktion φ läßt sich die Fehlerwahrscheinlichkeit bei Verwendung des Codes $C_n(p_0)$ durch

$$P_n =_{\mathrm{Df}} \sum_{u \in X^n} p(u)\varphi(u)$$

ausdrücken. Offenbar gilt für alle $\alpha \in (0, 1)$ die Abschätzung

$$\varphi(u) \leqq \left[\frac{p_0(u)}{2^{-nR}}\right]^{\alpha-1}.$$

Damit erhält man

$$P_n \leqq 2^{-n(1-\alpha)R} \sum_{u \in X^n} p(u) p_0(u)^{\alpha-1} = \exp\{-n(1-\alpha)R+\log \sum_{u \in X^n} p(u) p_0(u)^{\alpha-1}\} =$$

$$= \exp\{-n(1-\alpha)[R-H_\alpha(p, p_0)]\} \leqq \exp\{-n \sup_{\alpha \in (0,1)} (1-\alpha)[R-H_\alpha(p, p_0)]\}.$$

Ähnlich wie in [8] läßt sich zeigen, daß der Exponent

$$E(R, p, p_0) = \sup_{\alpha \in (0,1)} (1-\alpha)[R - H_\alpha(p, p_0)]$$

positiv ist, wenn der Parameter R die Bedingung

$$R > H(p, p_0)$$

erfüllt. Wegen $H(p_0) \geqq H(p, p_0)$ für alle $p \in P$ ist dann für $R > H(p_0)$ erst recht $E(R, p, p_0) > 0$. Damit ist der folgende Satz bewiesen:

Satz 4. Für jede konvexe und kompakte Klasse P von diskreten gedächtnislosen Quellen $[X, p]$ und jedes n läßt sich ein Blockcode $C_n \subseteqq X^n$ derart bestimmen, daß für die bei der Verwendung von C_n entstehende Fehlerwahrscheinlichkeit P_n

$$P_n \leqq 2^{-nE(R, p, p_0)}$$

gilt. Wenn $R > \max_{p \in P} H(p)$ ist, gilt für alle $p \in P$

$$\lim_{n \to \infty} P_n = 0.$$

Damit ist eine Folge von Blockcodes gefunden, deren Fehlerwahrscheinlichkeit für alle Quellen $[X, p]$ mit $p \in P$ verschwindet. Die Folge $(C_n)_{n=1,2,\ldots}$ kann daher ebenso wie die einzelnen Blockcodes als universell geeignet für P angesehen werden.

6. Abschließende Bemerkungen

1. Im Zusammenhang mit den Sätzen 2 und 3 muß auf das folgende offene Problem hingewiesen werden: Es ist nicht bekannt, was die optimalen Algorithmen bei der Anwendung auf eine entropiemaximierende Verteilung leisten. Die weiter oben benutzte Methode gibt hierüber keinen Aufschluß, da die optimalen Algorithmen nicht die für den Beweis der Sätze 2 und 3 wesentlichen Ungleichungen (3) bzw. (5) erfüllen. Ähnlich verhält es sich auch mit einigen anderen Konstruktionsverfahren, vgl. z. B. [6].

2. Die Fehlerabschätzung im Satz 4 ist etwas schwächer als die von Jelinek für den Fall einer bekannten Verteilung p angegebene. Eine entsprechende Abschätzung, die den Fehlerexponenten von Jelinek als Spezialfall enthält, wurde in einer früheren Arbeit bewiesen [3].

3. Abschließend sei noch darauf verwiesen, daß das Prinzip der maximalen Entropie hier gegenüber dem üblichen Gebrauch etwas erweitert wurde. Einerseits wurde es auf die Entropie der Ordnung α bezogen, andererseits wurde es durch die Einbeziehung der Ungleichung von Satz 1 ergänzt. Durch Satz 1 erhält das Prinzip der maximalen Entropie hier zugleich seine eigentliche Rechtfertigung.

Zusammenfassung

In dieser Arbeit werden universelle Codier- und Suchverfahren beschrieben, die auf der Verwendung des Prinzips der maximalen Entropie beruhen. Hierzu wird eine Ungleichung für verallgemeinerte Entropiemaße bewiesen, durch die das Prinzip der maximalen Entropie zugleich eine neue Rechtfertigung erhält.

AKADEMIE DER WISSENSHAFTEN DER DDR
INSTITUT FÜR MATHEMATIK
MOHRENSTRAßE 39, PF: 1304
DDR 1086 BERLIN

Literatur

[1] Campbell, L. L., A coding theorem and Rényi's entropy, *Inform. and Control*, v. 8, 1965, pp. 423—429.
[2] Davisson, L. D., Universal noiseless coding, *IEEE Trans. Inform. Theory*, IT—19, 1973, pp. 783—795.
[3] Fischer, Th., Über Verallgemeinerungen der Bongard-Entropie und Kodierungssätze für Quellen mit unbekanntem statistischen Verhalten, *Elektron. Informationsverarb. Kybernet.*, v. 13, 1977, pp. 125—135.
[4] Fischer, Th., Some remarks on the role of inaccuracy in Shannon's theory of information transmission, Trans. Eighth Prague Conf. on Information Theory etc., Academia, Prague, 1978, Vol. A, pp. 211—226.
[5] Fischer, Th., On the weighted path length of binary search trees for unknown access probabilities, *Lecture Notes in Computer Science*, v. 74, 1979, pp. 284—291.
[6] Horibe, Y., An improved bound for weight-balanced tree, *Inform. and Control*, v. 34, 1977, pp. 148—151.
[7] Jaynes, E. T., Information theory and statistical mechanics, *Phys. Rev.*, v. 106, 1957, pp. 620—630; v. 108, 1957, pp. 171—190.
[8] Jelinek, F., *Probabilistic information theory*, McGraw Hill, New York, 1968.
[9] Knuth, D. E., The art of computer programming, Vol. 3; *Sorting and searching*, Addison — Wesley, Reading, 1973.
[10] Mehlhorn, K., A best possible bound for weighted path length of binary search trees, *SIAM J. Comput.*, v. 6., 1977, pp. 235—239.
[11] Nath, P., Inaccuracy and coding theory, *Metrika*, v. 13, 1968, pp. 123—135.
[12] Rényi, A., Wahrscheinlichkeitsrechnung, Mit einem Anhang über Informationstheorie, DVW, Berlin, 1963.
[13] Shore, J. E., R. W. Johnson, Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy, *IEEE Trans. Inform. Theory*, IT—26, 1980, pp. 26—37.

(Eingegangen am 14. Sept. 1981)

Epis of some categories of Z-continuous partial algebras

By A. PASZTOR

§ 1. Introductory remarks on the connections with Computer Science

Let $\perp\omega\,\mathrm{Alg}_\Sigma$ denote the category of ω-continuous Σ-algebras with bottom and bottom-preserving ω-continuous homomorphisms between them in the sense e.g. of p. 132 of [13]. The structures $\mathfrak{A}\in\mathrm{Ob}\perp\omega\,\mathrm{Alg}_\Sigma$ are simply algebraic systems in the sense of [6] and the morphisms $h\colon \mathfrak{A}\to\mathfrak{B}$ of $\perp\omega\,\mathrm{Alg}_\Sigma$ are homomorphisms in the sense of [6]. What is special about $\perp\omega\,\mathrm{Alg}_\Sigma$ is that these algebraic systems and homomorphisms have to satisfy certain conditions. The present paper investigates $\perp\omega\,\mathrm{Alg}_\Sigma$ and certain strongly related categories.

Nowadays a very large part of Theoretical Computer Science (TCS) is based on $\perp\omega\,\mathrm{Alg}_\Sigma$ see e.g. [13] or [4] or [8]. We do not give here more references but it is very easy to find them in any recent publication on "Algebraic Semantics of Programming" or in the recent volumes of MFCS or FCT. Just for referential purposes we note that the French school of TCS uses the word "*complete magma*" for an algebraic system $\mathfrak{A}\in\mathrm{Ob}\,(\perp\omega\,\mathrm{Alg}_\Sigma)$. The importance of $\perp\omega\,\mathrm{Alg}_\Sigma$ for computer science was perhaps first discovered by Dana Scott and his co-workers during their pioneering work a long time ago but of course at that time the tool they found did not have its present polished form. Among others, the fixed point semantics of programming is based mostly on $\perp\omega\,\mathrm{Alg}_\Sigma$ (though this may not be explicit in some of the papers on the subject).

In computer science one has to deal with recursion (or iteration). In $\perp\omega\,\mathrm{Alg}_\Sigma$ recursion is treated as the supremum of an ω-chain where the members of that ω-chain are the finite approximations of the recursion in question.

Since $\perp\omega\,\mathrm{Alg}_\Sigma$ is the foundation for a large part of TCS, we think it is important for TCS — and what is more, it is indispensable for TCS — to investigate the *basic* properties of $\perp\omega\,\mathrm{Alg}_\Sigma$. Such basic questions are to characterize the epimorphisms of $\perp\omega\,\mathrm{Alg}_\Sigma$ and to know e.g. whether or not it is co-well-powered. The present paper investigates these questions. We note that these questions are indeed basic, e.g. in algebraic logic the epimorphism problem is equivalent to the problem of the connections between explicit definitions and implicit definitions in the logic under (algebraic) investigation.

§ 2. Introduction

In [12], when characterizing the epis of POS (Z) — the category of Z-complete posets with bottom and Z-continuous bottom-preserving maps — I just solved the first problem which arised on the way of characterizing the epis in $\perp\mathrm{Alg}_\Sigma(Z)$ — the category of Z-continuous Σ-algebras with bottom and of Z-continuous, bottom-preserving homomorphisms. The present paper solves some other problems which seem to play an important role in solving the main problem.

Instead of going on in an abstract manner, I will give first the *basic definitions.* A *subset system* is a map Z which assigns to each poset A a collection $Z(A)$ of its subsets such that for each monotonic map $f: A\to B$, if $X\in Z(A)$, then $f(X):=$ $:=\{f(x): x\in X\}\in Z(B)$.
A poset A is *Z-complete* if every element of $Z(A)$ has a l.u.b. (or $\sup_{\leqq_A}$) in A.

A map $f: A\to B$ is *Z-continuous* if it is monotonic and whenever $X\in Z(A)$ and $\sup_{\leqq_A} X$ exists, then $\sup_{\leqq_B} f(X)$ also exists and equals $f(\sup_{\leqq_A} X)$.
Let Σ be a similarity type or signature, i.e. a set of function symbols. For any $\sigma\in\Sigma$, $r(\sigma)$ denotes the arity of σ, which is an *arbitrary ordinal number.*
A *partial Σ-algebra* $\mathfrak{A}$ consists of a set A and of a family $\langle\sigma^A: \mathrm{dom}\,\sigma^A\to A\rangle_{\sigma\in\Sigma}$ of partial operations on A, i.e. for each $\sigma\in\Sigma$, $\mathrm{dom}\,\sigma^A\subseteqq A^{r(\sigma)}$. Given two partial Σ-algebras $\mathfrak{A}$ and $\mathfrak{B}$, a *homomorphism* $f: \mathfrak{A}\to\mathfrak{B}$ is a map $f: A\to B$ with the property that for any $\sigma\in\Sigma$, whenever $\mathbf{a}\in\mathrm{dom}\,\sigma^A$, $f\circ\mathbf{a}\in\mathrm{dom}\,\sigma^B$ and $f(\sigma^A(\mathbf{a}))=$ $=\sigma^B(f\circ\mathbf{a})$.
A partial Σ-algebra $\mathfrak{A}$ is *total,* if for any $\sigma\in\Sigma$, $\mathrm{dom}\,\sigma^A=A^{r(\sigma)}$.

For more about subset systems Z see [1], [9], [7]. For more about the theory of partial Σ-algebras see [2], [11], [10], [3].

The frame category of the present paper will be $\perp ZP\,\mathrm{Alg}_\Sigma$ defined as follows. $\mathfrak{A}\in\mathrm{Ob}\perp ZP\,\mathrm{Alg}_\Sigma$ iff $\mathfrak{A}$ is a partial Σ-algebra, A is partially ordered by $\leqq_A$ with least element and all the operations of $\mathfrak{A}$ are monotonic. $f: \mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ iff f is a Z-continuous bottom-preserving homomorphism.

The present paper gives a characterization of the epis in $\perp ZP\,\mathrm{Alg}_\Sigma$, for any Z and for any Σ.

Actually, we are more interested in some full subcategories of $\perp ZP\,\mathrm{Alg}_\Sigma$, which we define below.
$\perp Z\,\mathrm{Alg}_\Sigma$ denotes the full subcategory of $\perp ZP\,\mathrm{Alg}_\Sigma$ defined as $\mathrm{Ob}\perp Z\,\mathrm{Alg}_\Sigma=$ $=\{\mathfrak{A}\in\mathrm{Ob}\perp ZP\,\mathrm{Alg}_\Sigma: \mathfrak{A}$ is total$\}$.
$\perp P\,\mathrm{Alg}_\Sigma(Z)$ denotes the full subcategory of $\perp ZP\,\mathrm{Alg}_\Sigma$ with objects $\mathfrak{A}$ which are Z-complete and in which the operations are Z-continuous, i.e. for any $\sigma\in\Sigma$, if $X\in Z(\mathrm{dom}\,\sigma^A)$ and if $\sup_{\leqq_{A^{r(\sigma)}}} X\in\mathrm{dom}\,\sigma^A$, then $\sup_{\leqq_A}\{\sigma^A(\mathbf{x}): \mathbf{x}\in X\}=\sigma^A(\sup_{\leqq_{A^{r(\sigma)}}} X)$.
The objects of $\perp P\,\mathrm{Alg}_\Sigma(Z)$ are called *Z-continuous partial Σ-algebras.*
$\perp\mathrm{Alg}_\Sigma(Z)$ is the full subcategory of $\perp P\,\mathrm{Alg}_\Sigma(Z)$ with objects in which all operations are total.
$\perp P\,\mathrm{Alg}_{\Sigma,Z}$ is the full subcategory of $\perp P\,\mathrm{Alg}_\Sigma(Z)$ in which the objects are such that the domains of the operations are Z-complete.

In §3 we define the closure operator CL_Σ (see Definition 6) and, in Theorem 1, we prove that a morphism $f: \mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ is an epi iff $\mathrm{CL}_\Sigma(f(A))=B$.

This is a characterization of epis in $\perp ZP\,\mathrm{Alg}_\Sigma$. In Theorem 2 we extend this characterization to many other categories. At the end of §3 we show the connection of CL_Σ with CL of [12].

In §4 we use the above characterization of epis to show co-well-poweredness, assuming that the subset system Z is bounded. This assumption cannot be omitted.

Acknowledgement. This paper came into being due to Hajnal Andréka, István Németi, and Ildikó Sain, who pushed and helped me with many exciting discussions to write down these results.

§3. Characterization of epis

Throughout the paper, let a signature Σ and a subset system Z be fixed. Throughout this section, let $\mathfrak{A}\in\mathrm{Ob}\perp ZP\,\mathrm{Alg}_\Sigma$, $X\subseteq A$ and $a, b, c, d\in A$.

Definition 0. cl (X) is the least subset Y of A such that $X\subseteq Y$ and whenever $V\in Z(Y)$, then $\sup_{\leqq_A} V\in Y$.

Definition 1. We define a to be *X-greater or equal than b* $\left(a\xrightarrow{X}b\right)$ iff there is an ordinal α such that a is α, X-greater than b $\left(a\xrightarrow{\alpha, X}b\right)$ and the latter is defined as follows: $a\xrightarrow{0, X}b$ iff $b\leqq_A x\leqq_A a$ for some $x\in X$. Let $\alpha>0$. Then $a\xrightarrow{\alpha, X}b$ iff there is a term-function symbol t of type Σ such that $b\leqq_A t^{\mathfrak{A}}(\mathbf{b})$ and $a\ {}_A{\geqq}\ t^{\mathfrak{A}}(\mathbf{a})$ for some $\mathbf{b}, \mathbf{a}\in\mathrm{dom}\ t^{\mathfrak{A}}$ and for any $i<r(t)$, $\mathbf{b}(i)\in\mathrm{cl}\,(Y^i)$ for some $Y^i\subseteq A$ and for each $y\in Y^i$ there is an ordinal $\alpha_y<\alpha$ with $\mathbf{a}(i)\xrightarrow{\alpha_y, X}y$.

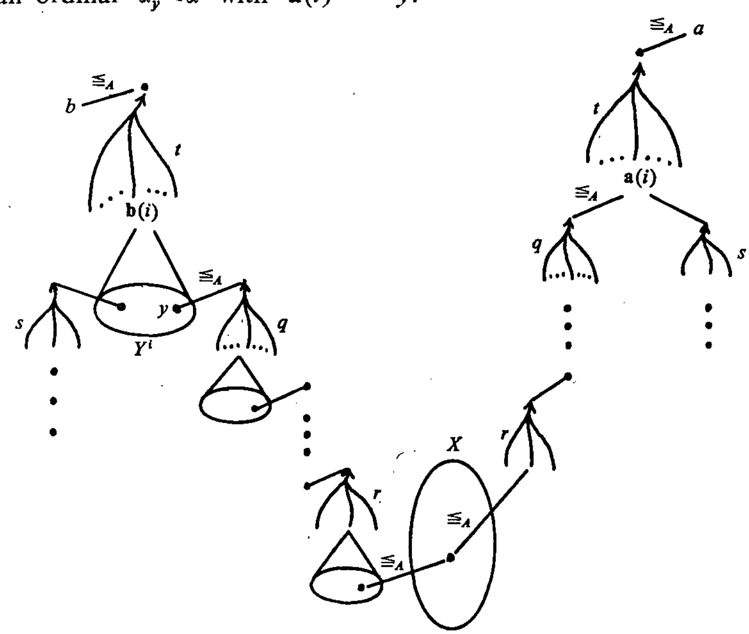


Fig. 1

Remark 2. Comparing this definition with Definition 1 of [12], notice that $a \overset{\alpha,X}{\longmapsto} b$ implies $a \xrightarrow{\alpha,X} b$ (just take for t the identity). If the operations of $\mathfrak{A}$ have all empty domains, then $a \xrightarrow{X} b$ iff ($a \overset{\alpha,X}{\longmapsto} b$ for some ordinal α).

Lemma 3. Suppose $a \xrightarrow{X} b$ and let $f, g\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$. Then $f{\restriction}X = g{\restriction}X$ implies $f(a) \,_B{\geqq}\, g(b)$.

Proof. $a \xrightarrow{X} b$ means $a \xrightarrow{\alpha,X} b$ for some ordinal α.

If $\alpha=0$ then there is an $x \in X$ with $b \leqq_A x \leqq_A a$. Hence by the monotonity of f and g we have $g(b) \leqq_B g(x) = f(x) \leqq_B f(a)$.

Suppose $\alpha>0$. Then $b \leqq_A t^{\mathfrak{A}}(\mathbf{b})$ and $t^{\mathfrak{A}}(\mathbf{a}) \leqq_A a$ for some termfunction t of type Σ and some $\mathbf{b}, \mathbf{a} \in \mathrm{dom}\, t^{\mathfrak{A}}$, and for any $i<r(t)$, $\mathbf{b}(i) \in \mathrm{cl}\,(Y^i)$ for some $Y^i \subseteq A$ and for each $y \in Y^i$ there is an ordinal $\alpha_y < \alpha$ with $\mathbf{a}(i) \xrightarrow{\alpha_y,X} y$. By the induction hypothesis, for any $i<r(t)$ and for any $y \in Y^i$ we have $g(y) \leqq_B f(\mathbf{a}(i))$. But, since by the Z-continuity of g we have $g(\mathbf{b}(i)) \in \mathrm{cl}\,(g(Y^i))$, also $g(\mathbf{b}(i)) \leqq_B f(\mathbf{a}(i))$ must hold for any $i<r(t)$. By the monotonity of the operations we get then $g(b) \leqq_B g(t^{\mathfrak{A}}(\mathbf{b})) = t^{\mathfrak{B}}(g \circ \mathbf{b}) \leqq_B t^{\mathfrak{B}}(f \circ \mathbf{a}) = f(t^{\mathfrak{A}}(\mathbf{a})) \leqq_B f(a)$. □

Corollary 4. $a \xrightarrow{X} b$ implies $a \,_A{\geqq}\, b$.

Corollary 5. Let $f, g\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ with $f{\restriction}X = g{\restriction}X$. Then $a \xrightarrow{X} a$ implies $f(a) = g(a)$.

Definition 6. $\mathrm{CL}_\Sigma(X) := \{a \in A\colon a \xrightarrow{X} a\}$.

Corollary 7. If for an $f\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ we have $\mathrm{CL}_\Sigma(f(A)) = B$, then f is an epi!

Now we are going to prove the converse of Corollary 7.

Lemma 8. $a \,_A{\geqq}\, b \xrightarrow{\alpha,X} c \,_A{\geqq}\, d$ imply $a \xrightarrow{\alpha,X} d$.

Proof. Immediate by Definition 1. □

Lemma 9. $a \xrightarrow{\mathrm{CL}_\Sigma(X)} b$ implies $a \xrightarrow{X} b$.

Proof. Suppose $a \xrightarrow{\alpha,\mathrm{CL}_\Sigma(X)} b$. We prove by transfinit induction on α that $a \xrightarrow{X} b$.

First let $\alpha=0$. Then there is an $x \in \mathrm{CL}_\Sigma(X)$ such that $b \leqq_A x \leqq_A a$. But since $x \xrightarrow{X} x$, it follows from Lemma 8 that $a \xrightarrow{X} b$.

Now suppose $\alpha>0$. Then $b \leqq_A t^{\mathfrak{A}}(\mathbf{b})$, $t^{\mathfrak{A}}(\mathbf{a}) \leqq_A a$ for some termfunction t of type Σ and for some $\mathbf{b}, \mathbf{a} \in \mathrm{dom}\,(t^{\mathfrak{A}})$ and for each $i<r(t)$, $\mathbf{b}(i) \in \mathrm{cl}\,(Y^i)$ for some $Y^i \subseteq A$ and for any $y \in Y^i$ there is an ordinal $\alpha_y < \alpha$ with $\mathbf{a}(i) \xrightarrow{\alpha_y,\mathrm{CL}_\Sigma(X)} y$ and hence by the induction hypothesis there is another ordinal β_y with $\mathbf{a}(i) \xrightarrow{\beta_y,X} y$. Applying Definition 1 we get then $a \xrightarrow{\beta,X} b$ for e.g. $\beta = \Sigma\{(\beta_y+1)\colon y \in Y^i,\ i<r(t)\}$ (see Fig. 2). □

Corollary 10. The operator $\mathrm{CL}_\Sigma\colon \mathscr{P}(A) \to \mathscr{P}(A)$ which assigns $\mathrm{CL}_\Sigma(X)$ to each $X \subseteq A$ is a closure operator.

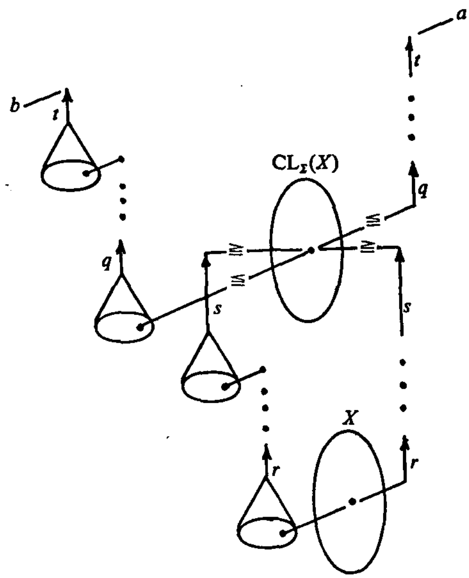


Fig. 2

Proof. 1) It follows immediately from Definition 1 that $a \xrightarrow{X} b$ for some $a, b \in A$ and $X \subseteqq Y \subseteqq A$ imply $a \xrightarrow{Y} b$. Hence $X \subseteqq Y \subseteqq A$ implies $\mathrm{CL}_\Sigma(X) \subseteqq \mathrm{CL}_\Sigma(Y)$.

2) $X \subseteqq \mathrm{CL}_\Sigma(X)$ follows from Definition 1.

3) $\mathrm{CL}_\Sigma(\mathrm{CL}_\Sigma(X)) \subseteqq \mathrm{CL}_\Sigma(X)$ follows from Lemma 9. □

Remark 11. Note that, by Lemma 9, if we suppose that either a or b is in $\mathrm{CL}_\Sigma(X)$, then $a \xrightarrow{X} b$ iff $a \,{}_A{\geqq}\, b$.

Lemma 12. The operations of $\mathfrak{A}$ are monotonic w.r.t. the relation "X-greater than or equal to" $\xrightarrow{X}$.

Proof. Let $\sigma \in \Sigma$ be arbitrary and suppose that for any $i < r(\sigma)$, $a_i \xrightarrow{\alpha_i, X} b_i$. Then by Definition 1, $\sigma^{\mathfrak{A}}(a_i \colon i < r(\sigma)) \xrightarrow{\alpha, X} \sigma^{\mathfrak{A}}(b_i \colon i < r(\sigma))$, where e.g. $\alpha := \Sigma\, \{(\alpha_i + 1) \colon i < r(\sigma)\}$ (just let $t = \sigma$ and $Y^i = \{b_i\}$). □

Corollary 13. $\mathrm{CL}_\Sigma(X)$ is closed w.r.t. all operations of $\mathfrak{A}$.

Lemma 14. Let $Y \subseteqq A$. If $a \xrightarrow{X} y$ for every $y \in Y$ then $a \xrightarrow{X} b$ for every $b \in \mathrm{cl}\,(Y)$.

Proof. Let $b \in \mathrm{cl}\,(Y)$. For every $y \in Y$ let α_y be such that $a \xrightarrow{\alpha_y, X} y$. Let $\alpha := \Sigma\{(\alpha_y + 1) \colon y \in Y\}$. Then $a \xrightarrow{\alpha, X} b$ by Definition 1 (just take for t the identity termfunction), i.e. $a \xrightarrow{X} b$. □

Corollary 15. $\mathrm{cl}\,(\mathrm{CL}_\Sigma(X)) = \mathrm{CL}_\Sigma(X)$.

Remark 16. In general, $\mathrm{CL}_\Sigma(X)$ is greater than the least subset $Y \subseteqq A$ such that $X \subseteq Y$ and Y is closed under the operations and $\mathrm{cl}(Y)=Y$. This follows from the fact that $\mathrm{cl} \neq \mathrm{CL}$ in POS which is proved in LEHMANN—PASZTOR [5].

Having arrived at this point we formulate the main result of the present paper.

THEOREM 1. If $f\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ is an epi then $\mathrm{CL}_\Sigma(f(A))=B$.

Proof. Denote $\mathrm{CL}_\Sigma(f(A))$ by B_0 and suppose that $B-B_0 \neq 0$. We will construct $\varphi, \psi\colon \mathfrak{B} \to \mathfrak{C} \in \mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ with $f\cdot\varphi=f\cdot\psi$ but $\varphi\neq\psi$, which contradicts the epiness of f.

Let $\varrho\colon B-B_0 \to B_1$ be a set isomorphism, where B_1 is disjoint from B. Let $C:=B\cup B_1$ (the second copower of B with amalgam B_0), $\varphi:=\mathrm{id}_B$ and $\psi:=\mathrm{id}_{B_0}\cup\varrho$ (the injections), where id_B and id_{B_0} denote the identity maps on B and B_0 respectively. Let $\delta:=\varphi\cup\varrho^{-1}$. Then $\delta\colon C\to B$.

Definition 17. We define on C the relation $\leqq_C$ as follows. For any $a, b\in C$

$$a \leqq_C b \quad \text{iff} \quad \begin{cases} \delta(a) \leqq_B \delta(b) & \text{if } a, b\in B \text{ or } a, b\in B_1 \\ \delta(a) \overset{B_0}{\longleftarrow} \delta(b) & \text{otherwise.} \end{cases}$$

ASSERTION 1. $\leqq_C$ is a partial order on C.

Proof. 1) $\leqq_C$ is reflexive since $\leqq_B$ is reflexive.

2) Suppose $a\leqq_C b\leqq_C c$. Assume $a, c\in B$ or $a, c\in B_1$. Then $\delta(a)\leqq_B\delta(b)\leqq_B\delta(c)$ by Definition 17 and Corollary 4, hence $\delta(a)\leqq_B\delta(c)$ by transitivity of $\leqq_B$, i.e. $a\leqq_C c$ by Definition 17. Assume that one of a and c is in B and the other one is in B_1. Then either $\delta(a)\overset{B_0}{\longleftarrow}\delta(b)$ or $\delta(b)\overset{B_0}{\longleftarrow}\delta(c)$, by Definition 17. Then $\delta(a)\overset{B_0}{\longleftarrow}\delta(c)$ by Lemma 8 and Corollary 4, i.e. $a\leqq_C c$ by Definition 17.

3) Let $a\leqq_C b$ and $b\leqq_C a$ for some $a, b\in C$. If $a, b\in B$ or $a, b\in B_1$ then $a=b$ by antisymmetry of $\leqq_B$ and since δ is one to one on B_1. Suppose one of a, b is in B and the other one is in B_1. Then $\delta(a)\overset{B_0}{\longleftarrow}\delta(b)\overset{B_0}{\longleftarrow}\delta(a)$ by Definition 17 and hence $\delta(a)=\delta(b)\in B_0$ by Corollary 4 and Lemma 8. Then $a=\delta(a)=\delta(b)=b$ (contradicting our hypothesis). □

ASSERTION 2. $\delta\colon C\to B$ is monotonic and $\varphi\cdot\delta=\psi\cdot\delta=\mathrm{id}_B$.

Proof. Immediate by Corollary 4 and by the definitions. □

ASSERTION 3. $\varphi, \psi\colon B\to C$ are Z-continuous.

Proof. 1) Clearly φ is monotonic. Let $a, b\in B$ be such that $a\leqq_B b$. If $a\in B_0$ or $b\in B_0$ then $a\overset{B_0}{\longleftarrow}b$ by Remark 11 and hence $\psi(a)\leqq_C\psi(b)$. If $a, b\in B-B_0$ then $\psi(a)\leqq_C\psi(b)$ by $\psi\cdot\delta=\mathrm{id}_B$ and Definition 17. Thus ψ is monotonic.

2) Let $Y\in Z(B)$ and assume that $b:=\sup\limits_{\leqq_B} Y$ exists. By Definition 17, $y\leqq_C b$ for any $y\in Y$, i.e. $b=\varphi(b)$ is an upper bound of $Y=\varphi(Y)$ in C. Now let $c\in C$ be another upper bound of Y. If $c\in B$ then $b\leqq_C c$ by Definition 17 and since $b=\sup\limits_{\leqq_B} Y$. Suppose $c\in B_1$. Then $\delta(c)\overset{B_0}{\longrightarrow}\delta(y)=y$ for every $y\in Y$, by Definition 17. Thus $\delta(c)\overset{B_0}{\longrightarrow}b$ by Lemma 14, i.e. $b\leqq_C c$. Thus $b=\sup\limits_{\leqq_C} Y$.

Since ψ is monotonic, $\psi(b)$ is an upper bound of $\psi(Y)$. Let $c\in C$ be another upper bound of $\psi(Y)$. Suppose $c\in B_0\cup B_1=\psi(B)$. Then $\delta(c)$ is an upper bound of Y in B since δ is monotonic and $\psi\cdot\delta=\mathrm{id}_B$, therefore $b\leqq_B\delta(c)$ by $b=\sup_{\leqq_B} Y$. Then $\psi(b)\leqq_C\psi\delta c=c$ by monotonity of ψ. Suppose $c\in B-B_0$. Then $c\xrightarrow{B_0}y$ for every $y\in Y$ by Definition 17 and Remark 11, therefore $c\xrightarrow{B_0}b$ by Lemma 14, i.e. $c=\delta(c)\xrightarrow{B_0}\delta\psi b$, hence $\psi(b)\leqq_C c$ by Definition 17. Thus $\psi(b)=\sup_{\leqq_C}\psi(Y)$.

Now we take on C the structure inherited from $\mathfrak{B}$, i.e. for any $\sigma\in\Sigma$, $\sigma^C:=\sigma^B\cup\psi\circ\sigma^B$. Since by Corollary 13 B_0 is closed under the operations of $\mathfrak{B}$, σ^C is a partial operation on C.

Remarks 18. 1) If $\mathfrak{B}$ is a *total* Σ-algebra and if Σ contains at most unary operation symbols, then $\mathfrak{C}:=(C,\sigma^C)_{\sigma\in\Sigma}$ is also a *total* Σ-algebra.

2) $\mathfrak{C}$ is the second copower of $\mathfrak{B}$ with amalgam B_0 in the category of all partial Σ-algebras.

3) $\delta\colon\mathfrak{C}\to\mathfrak{B}$ is a homomorphism. □

By its definition and by Lemma 12, σ^C is monotonic. Let $\mathfrak{C}:=(C,\sigma^C)_{\sigma\in\Sigma}$ with partial order $\leqq_C$. Then $\mathfrak{C}\in\mathrm{Ob}\perp ZP\,\mathrm{Alg}_\Sigma$. Clearly, $\varphi,\psi\colon\mathfrak{B}\to\mathfrak{C}$ are homomorphisms, therefore $\varphi,\psi\in\mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$, by Assertion 3. By $B-B_0\neq 0$ we have $\varphi\neq\psi$ and by $f(B)\subseteqq B_0$ we have $f\cdot\varphi=f\cdot\psi$. Thus f is not an epi. □

To prove Theorem 2, we shall need Lemma 19.

Lemma 19. Let P be any poset. Then conditions (i) and (ii) below are *equivalent.*

(i) P is directed.

(ii) For any $X\subseteqq P$ either X is cofinal in P or $P-X$ is cofinal in P.

Proof. Suppose that $X\subseteqq P$ is such that neither X nor $P-X$ is cofinal in P. Then there are $x,p\in P$ such that $x\leqq a$ implies $a\in X$ and $p\leqq a$ implies $a\notin X$. Then $\{x,p\}$ cannot have an upper bound.

Suppose that $\{x,p\}$ does not have an upper bound. Then neither $\{a\in P\colon a\geqq x\}$ nor $\{a\in P\colon a\ngeqq x\}$ is cofinal in P. □

Notation. $Z\subseteqq\Delta$ denotes the fact that X is directed for any poset P and $X\in Z(P)$.

Theorem 2. 1) For any Z and for any type Σ we have $f\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ is an epi iff $\mathrm{CL}_\Sigma(f(A))=B$.

2) Suppose that $Z\subseteqq\Delta$. Then a)—c) below hold.

a) For any type Σ,
$f\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp P\,\mathrm{Alg}_\Sigma(Z)$ is an epi iff $\mathrm{CL}_\Sigma(f(A))=B$.

b) For any type Σ,
$f\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp P\,\mathrm{Alg}_{\Sigma,Z}$ is an epi iff $\mathrm{CL}_\Sigma(f(A))=B$.

c) If the type Σ contains only 0- or 1-ary operation symbols then
$f\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp\mathrm{Alg}_\Sigma(Z)$ is an epi iff $\mathrm{CL}_\Sigma(f(A))=B$.

Proof. A) By Corollary 7, if for $f\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$ we have $\mathrm{CL}_\Sigma(f(A))=B$ then f is an epi. Further on, for any category $\mathscr{C}$ and any subcategory $\mathscr{B}$ of $\mathscr{C}$, if $f\in\mathrm{Mor}\,\mathscr{B}$ is an epi in $\mathscr{C}$, then it is an epi also in $\mathscr{B}$.

B) Now 1) follows from Theorem 1. Suppose that $Z \subseteq \Delta$. To prove 2) we shall use the construction in Theorem 1, i.e. we shall use $\varphi, \psi, \mathfrak{B}, \delta$, and $\mathfrak{C}$.

ASSERTION 4. If $\mathfrak{B}$ is Z-complete then so is $\mathfrak{C}$.

Proof. Let $X \in Z(C)$. Then $\delta(X) \in Z(B)$ since δ is monotonic, hence $b := \sup_{\leqq_B} \delta(X)$ exists. Then $\varphi(b) = \sup_{\leqq_C} \varphi\delta X$ and $\psi(b) = \sup_{\leqq_C} \psi\delta X$ by Assertion 3. Suppose that $B \cap X$ is cofinal in X. Then $\varphi\delta(B \cap X) = B \cap X$ is cofinal in $\varphi\delta X$, hence $\varphi(b) = \sup_{\leqq_C}(B \cap X) = \sup_{\leqq_C} X$. If $B \cap X$ is not cofinal in X then $B_1 \cap X$ is cofinal in X by $Z \subseteq \Delta$ and Lemma 19. Then, similarly as before, $\sup_{\leqq_C} X = \psi(b)$. □

ASSERTION 5. If $\mathfrak{B}$ is Z-continuous then so is $\mathfrak{C}$ and if $\mathfrak{B} \in \mathrm{Ob}\perp P\,\mathrm{Alg}_{\Sigma,Z}$ then $\mathfrak{C} \in \mathrm{Ob}\perp P\,\mathrm{Alg}_{\Sigma,Z}$.

Proof. $\mathfrak{C}$ is Z-complete by Assertion 4. Let $\sigma \in \Sigma$ and $X \in Z(\mathrm{dom}\,\sigma^C)$. Let us denote $B^{r(\sigma)}$, $C^{r(\sigma)}$, $\varphi^{r(\sigma)}$, $\psi^{r(\sigma)}$ and $\delta^{r(\sigma)}$ by $\bar{B}$, $\bar{C}$, $\bar{\varphi}$, $\bar{\psi}$ and $\bar{\delta}$ respectively. By the definition of $\mathfrak{C}$ we have $\mathrm{dom}\,\sigma^C \subseteq \bar{\varphi}(\bar{B}) \cup \bar{\psi}(\bar{B})$. Therefore either $X_\varphi := X \cap \bar{\varphi}(\bar{B})$ or $X_\psi := X \cap \bar{\psi}(\bar{B})$ is cofinal in X, by Lemma 19.

Suppose X_φ is cofinal in X. Then $\sigma^C(X_\varphi)$ is cofinal in $\sigma^C(X)$ by monotonity of σ^C, hence $\sup_{\leqq_C} X = \sup_{\leqq_C} X_\varphi$ and $\sup_{\leqq_C} \sigma^C(X) = \sup_{\leqq_C} \sigma^C(X_\varphi)$. By $X_\varphi \subseteq \bar{\varphi}(\bar{B})$ and $\varphi \cdot \delta = \mathrm{id}_B$ we have $X_\varphi = \bar{\varphi}\bar{\delta}X_\varphi$. Since $\delta \colon \mathfrak{C} \to \mathfrak{B}$ is a monotonic homomorphism, we have $\bar{\delta}(X_\varphi) \in Z(\mathrm{dom}\,\sigma^B)$ and thus $\sigma^B(\bar{\delta}X_\varphi) \in Z(B)$. Now, since φ is a homomorphism, we have $\sigma^C(\bar{\varphi}\bar{\delta}X_\varphi) = \varphi\sigma^B(\bar{\delta}X_\varphi)$, and then $\sup_{\leqq_C} \varphi\sigma^B(\bar{\delta}X_\varphi) = \varphi \sup_{\leqq_B} \sigma^B(\bar{\delta}X_\varphi)$ by Z-continuity of φ. By Z-completeness of $\mathfrak{B}$ and $\mathfrak{C}$ and by Z-continuity of φ we have that $\bar{\varphi}(\sup_{\leqq_B} Y) = \sup_{\leqq_C} \bar{\varphi}(Y)$ for any $Y \in Z(\bar{B})$ (because $\varphi[(\sup_{\leqq_B} Y)(i)] = \varphi \sup_{\leqq_B} \{\mathrm{y}(i) \colon \mathrm{y} \in Y\} = \sup_{\leqq_C} \{\varphi \mathrm{y}(i) \colon \mathrm{y} \in Y\} = (\sup_{\leqq_C} \bar{\varphi}Y)(i)$ for any $i < r(\sigma)$). Thus $\bar{\varphi} \sup_{\leqq_B} \bar{\delta}X_\varphi = \sup_{\leqq_C} X_\varphi$ and $\sup_{\leqq_B} \bar{\delta}(X_\varphi) = \bar{\delta}(\sup_{\leqq_C} X_\varphi)$ by $\varphi \cdot \delta = \mathrm{id}_B$, therefore $\sup_{\leqq_C} X_\varphi \in \mathrm{dom}\,\sigma^C$ iff $\sup_{\leqq_B} \bar{\delta}(X_\varphi) \in \mathrm{dom}\,\sigma^B$ since φ and δ are homomorphisms. Suppose $\sup_{\leqq_B} \bar{\delta}(X_\varphi) \in \mathrm{dom}\,\sigma^B$. Then $\sup_{\leqq_B} \sigma^B(\bar{\delta}X_\varphi) = \sigma^B(\sup_{\leqq_B} \bar{\delta}X_\varphi)$ by Z-continuity of $\mathfrak{B}$ and $\varphi\sigma^B(\sup_{\leqq_B} \bar{\delta}X_\varphi) = \sigma^C(\bar{\varphi} \sup_{\leqq_B} \bar{\delta}X_\varphi)$ since φ is a homomorphism.

Summing up:

$$\sup_{\leqq_C} \sigma^C(X) = \sup_{\leqq_C} \sigma^C(X_\varphi) = \sup_{\leqq_C} \sigma^C(\bar{\varphi}\bar{\delta}X_\varphi) = \sup_{\leqq_C} \varphi\sigma^B(\bar{\delta}X_\varphi) = \varphi \sup_{\leqq_B} \sigma^B(\bar{\delta}X_\varphi) =$$

$$= \varphi\sigma^B(\sup_{\leqq_B} \bar{\delta}X_\varphi) = \sigma^C(\bar{\varphi} \sup_{\leqq_B} \bar{\delta}X) = \sigma^C(\sup_{\leqq_C} \bar{\varphi}\bar{\delta}X_\varphi) = \sigma^C(\sup_{\leqq_C} X_\varphi) = \sigma^C(\sup_{\leqq_C} X).$$

If X_ψ is cofinal in X then the proof is the same as above, only φ has to be replaced by ψ everywhere. □

Now 2) follows from Assertion 5, Remarks 18 and from the proof of Theorem 1. □

In part 2 of [12] we defined the closure operator CL on posets. Now we are going to prove that in some cases CL_Σ on Z-continuous Σ-algebras equals CL.

Lemma 20. Let $Z \subseteqq \Delta$ and let Σ be finitary. Then for any $\mathfrak{A} \in \mathrm{Ob}\,\bot\mathrm{Alg}_\Sigma(Z)$, if $X \subseteqq A$ is a subalgebra of $\mathfrak{A}$, then for any $\sigma \in \Sigma$ $a_i \overset{\alpha_i, X}{\longmapsto} b_i$, $i < r(\sigma)$ implies $a := \sigma^A(a_i\colon i<r(\sigma)) \overset{\alpha, X}{\longmapsto} \sigma^A(b_i\colon i<r(\sigma)) =: b$ for $\alpha := \sup_{i<r(\sigma)} \alpha_i$.

Proof. If $\alpha = 0$ then for each $i < r(\sigma)$, $b_i \in \mathrm{cl}(Y^i)$ for some $Y^i \subseteqq A$ and for each $y \in Y^i$ there is an $x_y \in X$ such that $y \leqq x_y \leqq a_i$. Let $Y := \{\sigma^A(y)\colon y \in \times_{i<r(\sigma)} Y_i\}$.

By NELSON [9], for $Z \subseteqq \Delta$ and finitary Σ, $\sigma^A(\mathrm{cl}(Y_1), \ldots, \mathrm{cl}(Y_n)) = \mathrm{cl}(\sigma^A(Y_1, \ldots, Y_n))$, hence $b \in \mathrm{cl}(Y)$ and by the monotonity of σ for any $y \in Y$, since $y = \sigma^A(\mathbf{y})$ for some $\mathbf{y} \in \times_{i<r(\sigma)} Y_i$, there is an $x_y := \sigma^A(x_{\mathbf{y}(i)}\colon i<r(\sigma)) \in X$ (X is closed w.r.t. σ), such that $y \leqq_A x_y \leqq_A a$. Hence $a \overset{0, X}{\longmapsto} b$.

Let $\alpha > 0$ and suppose that whenever $\sup_{i<r(\sigma)} \alpha_i < \alpha$ the statement holds. Then for any $i < r(\sigma)$ $b_i \in \mathrm{cl}(Y_i)$ for some $Y_i \subseteqq A$ and for any $y^i \in Y_i$ there is a $b_{y^i} {}_A\!\geqq y^i$ and an ordinal $\beta_{y^i} < \alpha_i$ such that $a_i \overset{\beta_{y^i}, X}{\longmapsto} b_{y^i}$. Let $Y := \{\sigma^A(y)\colon y \in \times_{i<r(\sigma)} Y_i\}$. Then for any $y \in Y$, $y = \sigma^A(\mathbf{y})$ for some $\mathbf{y} \in \times_{i<r(\sigma)} Y_i$ and by the monotonity of σ $y \leqq_A \sigma^A(b_{\mathbf{y}(i)}\colon i<r(\sigma)) =: b_y$. By the induction hypothesis, $a \overset{\beta_y, X}{\longmapsto} b_y$, where $\beta_y := \sup_{i<r(\sigma)} \beta_{\mathbf{y}(i)}$. Since by the assumption $b \in \mathrm{cl}(Y)$ and since for any $y \in Y$, $\beta_y < \alpha$ (because $r(\sigma) \in \omega$), $a \overset{\alpha, X}{\longmapsto} b$. $\square$

Corollary 21. If $Z \subseteqq \Delta$ and Σ is finitary, then for any $f\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\,\bot\mathrm{Alg}_\Sigma(Z)$, $\mathrm{CL}(f(A))$ is a subalgebra of $\mathfrak{B}$.

Lemma 22. Let $Z \subseteqq \Delta$ and let Σ be finitary. Then for any $\mathfrak{A} \in \mathrm{Ob}\,\bot\mathrm{Alg}_\Sigma(Z)$, for any $a, b \in A$ and for any subalgebra X of $\mathfrak{A}$, $a \overset{X}{\longrightarrow} b$ implies $a \overset{\alpha, X}{\longmapsto} b$ for some ordinal α.

Proof. Suppose $a \overset{0, X}{\longrightarrow} b$. Then $b \leqq_A x \leqq_A a$ for some $x \in X$, which implies $a \overset{0, X}{\longmapsto} b$.

Let $a \overset{\alpha, X}{\longrightarrow} b$ and suppose that for any $\beta < \alpha$, $a \overset{\beta, X}{\longrightarrow} b$ already implies $a \overset{\gamma, X}{\longmapsto} b$ for some ordinal γ. Then $a \overset{\alpha, X}{\longrightarrow} b$ means $b \leqq_A t^{\mathfrak{A}}(\mathbf{b})$ and $t^{\mathfrak{A}}(\mathbf{a}) \leqq_A a$ for some term-function symbol t and some $\mathbf{a}, \mathbf{b} \in A^{r(t)}$ and for any $i < r(t)$, $\mathbf{b}(i) \in \mathrm{cl}(Y^i)$ for some $Y^i \subseteqq A$ and for each $y \in Y^i$ there is an ordinal $\alpha_y < \alpha$ with $\mathbf{a}(i) \overset{\alpha_y, X}{\longrightarrow} y$. By the induction hypothesis for each $i < r(t)$ and for each $y \in Y^i$, $\mathbf{a}(i) \overset{\beta_y, X}{\longmapsto} y$, for some ordinal β_y.

Let $Y := \{t^{\mathfrak{A}}(y)\colon y \in \times_{i<r(t)} Y^i\}$. Then $t^{\mathfrak{A}}(\mathbf{b}) \in \mathrm{cl}(Y)$, since by NELSON [9] for finitary Σ and for $Z \subseteqq \Delta$, $t^{\mathfrak{A}}(\mathrm{cl}(Y^0), \ldots, \mathrm{cl}(Y^{r(t)-1})) = \mathrm{cl}(t^{\mathfrak{A}}(Y^0, \ldots, Y^{r(t)-1}))$. For any $y \in Y$, $y = t^{\mathfrak{A}}(\mathbf{y})$ for some $\mathbf{y} \in \times_{i<r(t)} Y^i$, hence by Lemma 20, $t^{\mathfrak{A}}(\mathbf{a}) \overset{\beta_y, X}{\longmapsto} y$,

where $\beta_y = \sup_{i<r(t)} \beta_{y(i)}$. Then $t^{\mathfrak{A}}(\mathbf{a}) \overset{\beta, X}{\longmapsto} t^{\mathfrak{A}}(\mathbf{b})$ for some β greater than each β_y, $y \in Y$. By Lemmas 3 and 4 in part 2 of [12], $a \overset{\beta+1, X}{\longmapsto} b$. □

Corollary 23. If $Z \subseteq \Delta$ and Σ is finitary, then for any $f\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp\mathrm{Alg}_\Sigma(Z)$ we have $\mathrm{CL}_\Sigma(f(A)) \subseteq \mathrm{CL}(f(A))$.

Corollary 24. If $Z \subseteq \Delta$ and Σ contains at most unary operations, then 1) and 2) below hold.

1) $f\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp\mathrm{Alg}_\Sigma(Z)$ is an epi iff $\mathrm{CL}(f(A)) = B$.

2) $\perp\mathrm{Alg}_\Sigma(Z)$ is co-well-powered.

Proof. 1) By Corollary 23 and Corollary 21, $\mathrm{CL}_\Sigma(f(A)) = \mathrm{CL}(f(A))$. By Theorem 2, $f\colon \mathfrak{A} \to \mathfrak{B} \in \mathrm{Mor}\perp\mathrm{Alg}_\Sigma(Z)$ is an epi iff $\mathrm{CL}_\Sigma(f(A)) = B$.

2) In Corollary 2 of part 4 in [12] I proved $\mathrm{CL}(X) \subseteq \{\sup_{\leqq_A} S\colon S \subseteq X\}$ for any $A \in \mathrm{POS}(Z)$ (Z arbitrary) and $X \subseteq A$. □

§ 4. Co-well-poweredness

Suppose that Z is bounded, i.e. there is a cardinal $\delta(Z)$ such that for any poset A, if $X \in Z(A)$ then $|X| < \delta(Z)$.

In what follows our aim is to prove that for such Z-s those categories for which we have proved $[f\colon \mathfrak{A} \to \mathfrak{B}$ epi $\Leftrightarrow \mathrm{CL}_\Sigma(f(A)) = B]$ (see Theorem 2) are co-well-powered.

Let $\delta(\Sigma)$ denote the ordinal dimension of the type Σ, i.e. the least regular ordinal δ such that $|\delta| < |r(\sigma)|$ for any $\delta \in \Sigma$.

Denote by $\delta := \delta(\Sigma, Z)$ the least regular ordinal greater than $\max\{\delta(Z), \delta(\Sigma)\}$.

Notice that for any poset A, if $a \in A$ and $Y \subseteq A$, then $a \in \mathrm{cl}(Y)$ implies that there is an $Y' \subseteq Y$ with $|Y'| < \delta(Z)$ and $a \in \mathrm{cl}(Y')$. In the following we will suppose immodiately $|Y| < \delta(Z)$ when writing $a \in \mathrm{cl}(Y)$.

In the following let $\mathfrak{A} \in \mathrm{Ob}\perp ZP\,\mathrm{Alg}_\Sigma$, $X \subseteq A$ and $a, b, c, d \in A$.

Lemma 25. Suppose that Z is bounded by $\delta(Z)$ and let $\delta(\Sigma, Z)$ be as above. Then $a \overset{X}{\longrightarrow} b$ implies $a \overset{\beta, X}{\longrightarrow} b$ for some $\beta < \delta(\Sigma, Z)$.

Proof. Let $a \overset{X}{\longrightarrow} b$. Then $a \overset{\alpha, X}{\longrightarrow} b$ for some α.

If $\alpha = 0$ then the statement is true by $\delta > 0$.

Let $\alpha > 0$ and suppose that for every $\beta < \alpha$ the statement holds. $a \overset{\alpha, X}{\longrightarrow} b$ means that $b \leqq_A t^{\mathfrak{A}}(\mathbf{b})$, $t^{\mathfrak{A}}(\mathbf{a}) \leqq_A a$ for some termfunction symbol t and some $\mathbf{b}, \mathbf{a} \in \mathrm{dom}\, t^{\mathfrak{A}}$ and that for any $i < r(t)$, $\mathbf{b}(i) \in \mathrm{cl}(Y^i)$ for some $Y^i \subseteq A$, $\mathrm{card}(Y^i) < \delta(Z)$, and for any $y \in Y^i$ there is an ordinal $\alpha_y < \alpha$ such that $\mathbf{a}(i) \overset{\alpha_y, X}{\longrightarrow} y$. By the induction hypothesis for any $i < r(t)$ and for any $y \in Y^i$ there is an ordinal $\beta_y < \delta(\Sigma, Z)$ with $\mathbf{a}(i) \overset{\beta_y, X}{\longrightarrow} y$. Let $\beta := \Sigma\{(\beta_y + 1)\colon y \in Y^i,\ i < r(t)\}$. By the definition of $\delta(\Sigma, Z)$ we have $\beta < \delta(\Sigma, Z)$ and by Definition 1, $a \overset{\beta, X}{\longrightarrow} b$. □

Definition 26. For every $a, b \in A$ such that $a \overset{X}{\longrightarrow} b$ we define $R_{a,b}$ as follows. Let α be the least ordinal for which $a \overset{\alpha, X}{\longrightarrow} b$.

If $\alpha=0$ then $a \xrightarrow{0,X} b$ means that there is an $x \in X$ with $b \leqq_A x \leqq_A a$. Let us fix one $x \in X$ with this property. Then $R_{a,b} := \langle \{x\}, 0 \rangle$.

If $\alpha>0$ then $a \xrightarrow{\alpha,X} b$ means that $b \leqq_A t^{\mathfrak{A}}(\mathbf{b})$ and $t^{\mathfrak{A}}(\mathbf{a}) \leqq a$ for some termfunction symbol t and some $\mathbf{b}, \mathbf{a} \in A^{r(t)}$ and that for each $i<r(t)$, $\mathbf{b}(i) \in \mathrm{cl}(Y^i)$ for some $Y^i \subseteq A$ and for any $y \in Y^i$ there is an ordinal $\alpha_y < \alpha$ with $\mathbf{a}(i) \xrightarrow{\alpha_y, X} y$. Let us fix t, $\mathbf{a}$, $\mathbf{b}$ and the Y^i-s for any $i<r(t)$. Then

$$R_{a,b} := \langle t, \langle \{R_{\mathbf{a}(i),y} \colon y \in Y^i\} \rangle_{i<r(t)}, \alpha \rangle.$$

Lemma 27. If $a \xrightarrow{X} b$, $c \xrightarrow{X} d$, and $R_{a,b}=R_{c,d}$ then $d \leqq_A a$ (and $b \leqq_A c$).

Proof. By transfinit induction on α of $R_{a,b}$.

Let $R_{a,b}=R_{c,d}=\langle \{x\}, 0 \rangle$. Then $b \leqq_A x \leqq_A a$ and $d \leqq_A x \leqq_A c$, hence $d \leqq_A x \leqq_A a$, i.e. $d \leqq_A a$.

Let $R_{a,b}=R_{c,d}=\langle t, \langle \{R_{\mathbf{a}(i),y} \colon y \in Y^i\} \rangle_{i<r(t)}, \alpha \rangle$ where $\alpha>0$. Then $b \leqq_A t^{\mathfrak{A}}(\mathbf{b})$, $t^{\mathfrak{A}}(\mathbf{a}) \leqq_A a$, $d \leqq_A t^{\mathfrak{A}}(\mathbf{d})$ and $t^{\mathfrak{A}}(\mathbf{c}) \leqq_A c$ for some $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d} \in A^{r(t)}$ and for any $i<r(t)$, $\mathbf{b}(i) \in \mathrm{cl}(Y^i)$ and $\mathbf{d}(i) \in \mathrm{cl}(Z^i)$ for some $Y^i, Z^i \subseteq A$ and for any $y \in Y^i$ there is a $z \in Z^i$ such that $R_{\mathbf{a}(i),y}=R_{\mathbf{c}(i),z}$ (and since $R_{a,b}=R_{c,d}$ of course for any $z \in Z^i$ there is a $y \in Y^i$ with $R_{\mathbf{c}(i),z}=R_{\mathbf{a}(i),y}$). By the induction hypothesis this implies $z \leqq_A \mathbf{a}(i)$ for any $z \in Z^i$. Since $\mathbf{d}(i) \in \mathrm{cl}(Z^i)$, $\mathbf{d}(i) \leqq_A \mathbf{a}(i)$. Then by the monotonity of t, $d \leqq_A t^{\mathfrak{A}}(\mathbf{d}) \leqq_A t^{\mathfrak{A}}(\mathbf{a}) \leqq_A a$, i.e. $d \leqq_A a$. □

Let $\mathrm{Term}(\Sigma)$ denote the class of all termfunction symbols of type Σ. It is easy to show that $\mathrm{Term}(\Sigma)$ is a set. Let $\gamma(X,Z)$ be the least regular ordinal greater than $(\max\{|X|, |\mathrm{Term}(\Sigma)|, \delta(\Sigma,Z)\})^{\delta(\Sigma,Z)}$.

Let H_0 be the set of all $R_{a,b}$-s of form $\langle \{x\}, 0 \rangle$. Then $|H_0|=|X|<\gamma(X,Z)$.

Let $0<\alpha<\delta(\Sigma,Z)$. Then we define H_α to be the set of all $R_{a,b}$-s of form $\langle t, \langle \{R_{\mathbf{a}(i),y} \colon y \in Y^i\} \rangle_{i<r(t)}, \alpha \rangle$. Then $|H_\alpha| < |\mathrm{Term}(\Sigma)| \cdot |(\bigcup \{H_\beta \colon \beta<\alpha\})^{\delta(Z)\cdot\delta(\Sigma)}| < \gamma(X,Z)$.

By Lemma 25, if $a \xrightarrow{X} b$, then there is an ordinal $\beta<\delta(\Sigma,Z)$ such that $R_{a,b} \in H_\beta$. By the definition of $\gamma(X,Z)$, $|\bigcup\{H_\beta \colon \beta<\delta(\Sigma,Z)\}|<\gamma(X,Z)$.

By Lemma 26, we know that for any $a, b \in \mathrm{CL}_\Sigma(X)$, if $R_{a,a}=R_{b,b}$ then $a=b$. Hence we can immediately see that we have proved

Corollary 28. $|\mathrm{CL}_\Sigma(X)|<\gamma(X,Z)$.

Corollary 29. 1) Let Z be bounded. Then for any similarity type Σ, $\perp ZP\,\mathrm{Alg}_\Sigma$ is co-well-powered.

2) Suppose that Z is bounded and $Z \subseteq \Delta$. Then for any type Σ, $\perp P\,\mathrm{Alg}_\Sigma(Z)$ and also $\perp P\,\mathrm{Alg}_{\Sigma,Z}$ are co-well-powered. If Σ contains only 0- or 1-ary operation symbols then $\perp \mathrm{Alg}_\Sigma(Z)$ is co-well-powered. □

Next we prove that in Corollary 29 the condition that Z is bounded cannot be omitted.

Proposition 30. Let Σ be a signature with at least one $f \in \Sigma$ such that $r(f)>0$. Then there is a subset system $Z \subseteq \Delta$ such that *both* $\perp ZP\,\mathrm{Alg}_\Sigma$ and $\perp Z\,\mathrm{Alg}_\Sigma$ are not co-well-powered.

Proof. For simplicity we assume $\Sigma=\{f\}$ with $r(f)=1$. It is obvious how to extend the present proof for the general case (in the formulation of the present Proposition).

We define Z as follows. For every poset $\langle A, \leqq_A\rangle$ let $Z(\langle A, \leqq_A\rangle):=\{Y\colon Y\subseteq A$ and $(\exists\alpha\in\mathrm{Ord})\langle\alpha, \in\rangle\cong\langle Y, \leqq_A\rangle\}$. Clearly, Z is a subset system and $Z\subseteq\Delta$.

ω denotes the set of natural numbers and Id_S denotes the identity function on S, for any set S.

Let $\mathfrak{A}:=\langle\omega, \perp_A, \leqq_A, f^A\rangle$ such that $\perp_A=0$, $\leqq_A=\{0\}\times\omega\cup\mathrm{Id}_\omega$ and $f^A=\mathrm{Id}_\omega$. Then $\mathfrak{A}\in\mathrm{Od}\perp Z\,\mathrm{Alg}_\Sigma$.

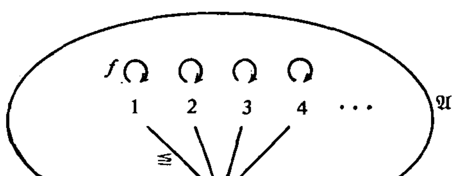


Fig. 3

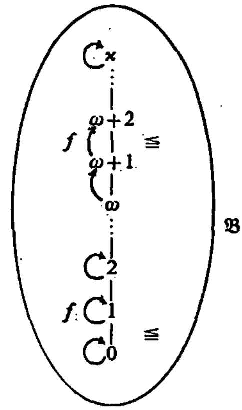


Fig. 4

Let $\varkappa\in\mathrm{Ord}$ be arbitrary but such that $\varkappa\geqq\omega$. Let $\mathfrak{B}:=\langle\varkappa+1, \perp_B, \leqq_B, f^B\rangle$ such that $\perp_B=0$, $\leqq_B=\in\cap(B\times B)\cup\mathrm{Id}_B$ and $f^B=f^A\cup\{\langle\alpha, \alpha+1\rangle\colon \omega\leqq\alpha+1\in B\}\cup\bigcup\{\langle\varkappa, \varkappa\rangle\}$. E.g. if $\omega+1\in B$ then $f^B(\omega)=\omega+1$ (see Fig. 4). Now $\mathfrak{B}\in\mathrm{Ob}\perp Z\,\mathrm{Alg}_\Sigma$ since $f^B\colon\langle B, \leqq_B\rangle\to\langle B, \leqq_B\rangle$ is an endomorphism that is $f^B\colon B\to B$ is monotonic.

Let $h:=\mathrm{Id}_\omega$, i.e. $h\colon A\to B$ is the identical embedding of ω into $\varkappa+1$. Then $h\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp Z\,\mathrm{Alg}_\Sigma$ since h is a bottom-preserving homomorphism and h is Z-continuous. Since $\perp Z\,\mathrm{Alg}_\Sigma\subseteq\perp ZP\,\mathrm{Alg}_\Sigma$ we have that $h\colon\mathfrak{A}\to\mathfrak{B}\in\mathrm{Mor}\perp ZP\,\mathrm{Alg}_\Sigma$, too.

ASSERTION 6. $h\colon\mathfrak{A}\to\mathfrak{B}$ is an epi in $\perp ZP\,\mathrm{Alg}_\Sigma$ as well as in $\perp Z\,\mathrm{Alg}_\Sigma$.

Proof. Let $X=h(A)$. Then $X=\omega\subseteq B$. Let $\gamma\in B$. Assume $\gamma\in\mathrm{Cl}_\Sigma(X)$. If $\gamma\in X$ then $\gamma+1\in X\subseteq\mathrm{CL}_\Sigma(X)$ obviously. Assume $\gamma\notin X$. Then $\gamma\geqq\omega$, and $f^B(\gamma)=\gamma+1$. Hence $\gamma+1\in\mathrm{CL}_\Sigma(X)$ by Corollary 13. Let $\alpha\in B$ be a limit ordinal and assume $\alpha\subseteq\mathrm{CL}_\Sigma(X)$. Then by $\alpha=\sup\limits_{\leqq_B}\alpha$ and $\alpha\in Z(B)$ we conclude $\alpha\in\mathrm{cl}\,(\mathrm{CL}_\Sigma(X))\subseteq\mathrm{CL}_\Sigma(X)$ by Corollary 15. Thus by induction we proved $B=\varkappa+1=\mathrm{CL}_\Sigma(X)$. Hence by Lemma 7 we have checked that h is an epi both in $\perp ZP\,\mathrm{Alg}_\Sigma$ and in $\perp Z\,\mathrm{Alg}_\Sigma$. □

By Assertion 6 and the definition of $\mathfrak{B}$ we proved that $\mathfrak{A}$ is such that $(\forall\varkappa\in\mathrm{Ord})\cdot(\exists\mathfrak{B})(\exists\mathfrak{A}\twoheadrightarrow\mathfrak{B})\,|B|\geqq|\varkappa|$, which means that *the epimorphic images of* $\mathfrak{A}$ are not isomorphic to any sub*set* of $\mathrm{Ob}\perp ZP\,\mathrm{Alg}_\Sigma$. Thus $\perp ZP\,\mathrm{Alg}_\Sigma$ is not co-well-powered.

Since h is an epi in $\perp Z\ \mathrm{Alg}_\Sigma$ as well, we have that $\perp Z\ \mathrm{Alg}_\Sigma$ is not co-well-powered either. □

Problem 31. Is $\perp ZP\ \mathrm{Alg}_\Sigma$ co-well-powered for some *un*bounded Z? More precisely, is it true that for all Σ there is some unbounded Z such that $\perp ZP\ \mathrm{Alg}_\Sigma$ is co-well-powered?

INSTITUT FÜR INFORMATIK
UNIVERSITÄT STUTTGART
7 STUTTGART 1 AZENBERGSTR. 12.

References

[1] ADÁMEK, J., E. NELSON, J. REITERMAN, Tree construction of free continuous algebras, *J. Comput. System Sci.*, v. 24, 1982, pp. 114—146.
[2] ANDRÉKA, H., I. NÉMETI, Generalization of the concept of variety and quasivariety to partial algebras through category theory, Dissertationes Mathematicae (Rozprawy Math.) No 204, Polish Scientific Publishers, Warszawa, 1981, p. 101.
[3] BURMEISTER, P., Partial algebras — Survey of a unifying approach towards a two-valued model theory for partial algebras, *Algebra Universalis*, To appear.
[4] GUESSARIAN, I., Algebraic semantics, Lecture Notes in Computer Science, v. 99, Springer-Verlag, Berlin, 1981, p. 158.
[5] LEHMANN, D., A. PASZTOR, Epis need not be dense, *Theoret. Comput. Sci.*, v. 17, 1982, pp. 151—161.
[6] MALCEV, A. I., *Algebraic systems*, Springer-Verlag, Berlin, 1973.
[7] MESEGUER, J., Completions, Factorizations and Colimits for ω-Posets, Colloq. Math. Soc. J. Bolyai XXVI.
[8] MESEGUER, J., A Birkhoff-like theorem for algebraic classes of interpretations of program schemes, *Formalization of Programming Concepts* (Proc. Peniscola, Spain, 1981), Lecture Notes in Computer Science v. 107, Springer-Verlag, Berlin, 1981.
[9] NELSON, E., Free Z-continuous algebras, *Proc., Workshop on Continuous Lattices*. LNM 871, 1981, pp. 315—334.
[10] NÉMETI, I., From hereditary classes to varieties in abstract model theory and partial algebras, *Beiträge Algebra Geom.*, v. 7, 1978, pp. 69—78.
[11] NÉMETI, I., I. SAIN, Cone-implicational subcategories and some Birkhoff-type theorems, *Contributions to Universal Algebra* (Proc. Coll. Esztergom, 1977), Colloq. Math. Soc. J. Bolyai, Vol 29, North-Holland, 1981, pp. 535—578.
[12] PASZTOR, A., The epis of POS(Z), *Comment. Math. Univ. Carolin.*, v. 23, 1982, pp. 285—299.
[13] THATCHER, J. W., E. G. WAGNER, J. B. WRIGHT, Notes on algebraic fundamentals for theoretical computer science, *Foundations of Computer Science* III. Part 2: Languages, Logic, Semantics, J. W. DEBAKKER ed. J. VAN LEEUWEN ed, Mathematical Centre Tract, v. 109, pp. 83—164.

(Received Dec. 3, 1981)

INDEX — TARTALOM

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1982. június 30
Megjelenés: 1983. április
Példányszám: 500. Terjedelem: 10,85 (A/5) ív
Készült monószedéssel, íves magasnyomással
az MSZ 5601 és az MSZ 5602—55 szabvány szerint
82-3047 — Szegedi Nyomda — F. v.: Dobó József igazgató