Printing Requisition/Graphic Services

Dept. No. 86023

Title or Description
Developments in the Theory of Regular Languages CS-80-26

1. Please complete unshaded areas on form as applicable. (4-part no carbon required).
2. Distribute copies as follows: White, Canary and Pink—Printing, Arts Library or applicable Copy Centre Goldenrod—**Retain.**
3. On completion of order, pink copy will be returned with printed material. Canary copy will be costed and returned to requisitioner, **Retain as a record of your charges.**
4. Please direct enquiries, quoting requisition number, to Printing/Graphic Services, Extension 3451.

Date	Date Required	Account
May 7/80	ASAP	126-6240-41

Signature	Signing Authority
Brad Lagrion	P.A. Bizzozero IBS.

Department	Room	Phone	Delivery
Computer Science	5100	3293	☑ Mail ☐ Via Stores ☐ Pick-up ☐ Other

Reproduction Requirements
☑ Offset ☐ Signs/Repro's ☐ Xerox

Number of Pages 18

Number of Copies 50

Type of Paper Stock
☑ Bond ☐ Book ☐ Cover ☐ Bristol ☐ Supplied

Paper Size
☑ 8½ x 11 ☐ 8½ x 14 ☐ 11 x 17

Paper Colour
☑ White ☐ Other

Ink
☐ Black

Printing
☑ 1 Side ☐ 2 Sides

Numbering to

Binding/Finishing Operations
☑ Collating ☐ Corner Stitching ☐ 3 Ring ☐ Tape ☐ Plastic Ring ☐ Perforating

Folding
Finished Size 3 STAPLES

Cutting
Finished Size

Special Instructions

3 staples please; covers supplied & Backs

Cost: Time/Materials		Fun.	Prod.Un.	Prod.Opr. Cl. No.	Mins.	Total
Signs/Repro's	1					
Camera	2					
Correcting & Masking Negatives	3					
Platemaking	4					
Printing	5			32	20	
Bindery	6			21	40	
Sub. Total Time						
Sub. Total Materials						
Prov. Tax						
Total						

Film		Plates		
Qty	Size	Qty	Size & Type	
Paper		**Plastic Rings**		
Qty	Size	Qty	Size	
Outside Services				

10M.

Graphic Services May 68 482-2

DEVELOPMENTS IN THE THEORY OF REGULAR LANGUAGES

by

J.A. Brzozowski

RESEARCH REPORT CS-80-26

University of Waterloo
Department of Computer Science
Waterloo, Ontario, Canada

May 1980

Invited Paper for IFIP-80, Tokyo and Melbourne, October 1980

DEVELOPMENTS IN THE THEORY OF REGULAR LANGUAGES*

Janusz A. Brzozowski
Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada

Although the number of researchers working on problems related to regular languages is presently quite small, significant developments have taken place during the past five years. Powerful new tools for studying families of regular languages and the corresponding finite monoids have been provided by Eilenberg in his theory of varieties. The variety approach has been used by Straubing and Thérien to characterize families of languages whose syntactic monoids contain only solvable groups. Other examples of recently studied monoid varieties include R-trivial and related monoids. Some recent results on closure properties of varieties and on the connection between codes and monoid varieties will be discussed. Several relatively old problems that were solved recently will also be treated.

1. INTRODUCTION

In the preface of their monograph published in 1971 McNaughton and Papert [28] state:

> "Until recently the significant classification of automata divided finite from infinite machines ... . There were, of course, definitions of subclasses; for example, definite-event machines. But the significant theorems took no account of these definitions.
>
> This situation has changed. In this monograph we shall study a particular class of finite automata."

The class they studied was that of counter-free automata and the corresponding star-free languages. Now the situation has changed again, quite significantly. Many new classes of finite automata and regular languages have been characterized. Moreover, the study of these classes has been systematized through the use of the theory of varieties of languages and semigroups.

The connection between finite automata and regular languages was established by Kleene in 1951 [20]. In 1957 Myhill [29] characterized recognizable languages by congruences of finite index. A systematic treatment of these topics may be found in Rabin and Scott's paper [40] written in 1957. Finite monoids and semigroups have been associated with finite automata and regular languages by many authors; a clear exposition of the concept of syntactic monoids was written in 1966 by McNaughton and Papert [27].

The first subclass of regular languages to be introduced was that of definite languages defined by Kleene [20]. This subclass was later studied by several authors: Perles, Rabin and Shamir [33] in 1961, Brzozowski [1] in 1962, Ginzburg [15] in 1966, and Steinby [48] in 1969. Two classes closely related to the class of definite languages were also considered: reverse definite languages [1,15], and generalized definite languages [15].

A much larger and more significant subclass, that of star-free languages, was characterized by Schützenberger in 1964 [42]. In 1968 a systematic framework for the study of subclasses of the class of star-free languages was provided by the dot-depth hierarchy of Cohen and Brzozowski [9,10]. In that framework the class of finite/cofinite languages emerged as a basic building block.

In the early 1970's characterizations of several more classes were found. It became clearer that natural characterizations of classes of regular languages and classes of finite semigroups or monoids correspond. Thus we have the semigroup characterization of locally testable languages by Brzozowski and Simon [8], McNaughton [26] and Zalcstein [56, 57,58], and of piecewise testable languages by Simon [45,46]. The study of depth-one languages [8,45] showed that finite/cofinite, definite, reverse definite, generalized definite, and locally testable languages are the simplest languages at the very "bottom" of an infinite hierarchy approaching the depth-one languages, i.e. languages requiring at most one level of concatenation. Also, the piecewise testable languages were shown to be very special cases of depth-one languages. The class of depth-one languages is, in turn, at the bottom of the infinite dot-depth hierarchy leading to the star-free languages. Thus the results mentioned must be viewed as a very modest beginning, since they were confined to subclasses of depth-one languages.

A rather major breakthrough in the study of classes of regular languages came with the publication of Eilenberg's work on varieties [12]. The concepts of varieties have been used previously in the theory of automata and languages, notably by Schützenberger (e.g. in [41]), but the first systematic treatment appears in [12]. During the last five years numerous varieties of languages and semigroups have been characterized, and the theory of subclasses of regular languages is now very

*Research supported by NSERC, Canada, Grant No. A-1617.

rich indeed. Moreover, many concepts of classical mathematics have been successfully applied to the study of these languages.

Sections 3 through 7 of this paper survey recent contributions to the theory of varieties in general, and also to characterizations of particular varieties. The last section summarizes some recent contributions to the theory of regular languages that do not fit this general pattern.

2. SEMIGROUPS, CONGRUENCES, LANGUAGES AND AUTOMATA

We begin with a brief review of some basic concepts; for more detail see [11], [12], and [25].

A semigroup $(S,\cdot)$ is a set S with a binary operation $\cdot$ (called multiplication or product) that satisfies the associative law: $r\cdot(s\cdot t) = (r\cdot s)\cdot t$ for all $r,s,t \in S$. An idempotent of S is any element e of S satisfying $e\cdot e = e$. A zero of S is an element of S, usually denoted by 0, such that $0\cdot s = s\cdot 0 = 0$ for all $s \in S$. Note that S can have at most one zero, and that 0 is an idempotent of S. A unit or identity of S is an element of S, usually denoted by 1, such that $1\cdot s = s\cdot 1 = s$ for all $s \in S$. Note that S can have at most one unit, and that 1 is an idempotent of S. A monoid is a semigroup with a unit element. A group is a monoid in which every element s has an inverse s^{-1} satisfying: $ss^{-1} = s^{-1}s = 1$.

A subsemigroup of a semigroup $(S,\cdot)$ is a subset T of S such that $t,t' \in T$ implies $t\cdot t' \in T$; i.e. $(T,\cdot)$ is itself a semigroup. A submonoid of a monoid $(M,\cdot,1)$ is a subset T of M such that $(T,\cdot,1)$ is a monoid. Let $(S,\cdot)$ be a semigroup and $T \subset S$ a subsemigroup of S. If T has a unit element e, then $(T,\cdot,e)$ is a monoid in S. If a monoid in S is a group then it is called a group in S.

Given subsets of a semigroup S one defines the usual boolean operations: if $T,R \subset S$ then $T \cup R$, $T \cap R$ and $\overline{T} = S - T$ denote union, intersection and complement, respectively. We also extend the multiplication in S to subsets, i.e. $TR = \{s \mid s = tr,\ t \in T,\ r \in R\}$. (The dot for the product is frequently omitted for convenience.) If $T \subset S$ then T^+ denotes the subsemigroup of S generated by T: $T^+ = \bigcup_{n\geq 1} T^n$, where $T^{n+1} = T^nT$, $n \geq 1$. If S is a monoid we define T^*, the submonoid generated by T, as $T^* = \bigcup_{n\geq 0} T^n$, where $T^0 = \{1\}$ for all T.

Let S be a semigroup, $T \subset S$, and $s \in S$. We define the left and right quotients of T by s as follows: $s^{-1}T = \{s' \in S \mid ss' \in T\}$, and $Ts^{-1} = \{s' \in s \mid s's \in T\}$.

The direct product of two semigroups $(S,\cdot)$ and $(T,\circ)$ is the semigroup $(S\times T,\square)$ where $(s,t) \square (s',t') = (s\cdot s', t\circ t')$ for all $s,s' \in S$, $t,t' \in T$.

A morphism φ from a semigroup $(S,\cdot)$ to a semigroup $(T,\circ)$ is a mapping $\varphi : S \to T$ such that $(s\cdot s')\varphi = (s\varphi) \circ (s'\varphi)$. φ is surjective if $T = S\varphi$; then T is said to be a morphic image of S. If φ is surjective and injective (1-1) then it is a semigroup isomorphism from S to T and we write $S \simeq T$. If $(S,\cdot,1)$ and $(T,\circ,I)$ are monoids, then $\varphi : S \to T$ is a monoid morphism if it is a semigroup morphism and $1\varphi = I$. Usually the unit elements of both S and T are denoted by 1. A semigroup T divides a semigroup S, written $T < S$, iff T is a morphic image of a subsemigroup of S.

Let S be a semigroup and α an equivalence relation on S; α is a congruence on S iff $s\ \alpha\ s'$ and $t\ \alpha\ t'$ implies $st\ \alpha\ s't'$. If α and β are congruences on S and $\beta \subset \alpha$ then α is a superset of β. Let S and T be semigroups and let $\varphi : S \to T$ be a semigroup morphism. If β is a congruence on T we define a congruence $\alpha = \beta\varphi^{-1}$ on S by $s(\beta\varphi^{-1})s'$ iff $(s\varphi)\beta(s'\varphi)$. We say that $\alpha = \beta\varphi^{-1}$ is obtained from β by an inverse morphism.

Let A be a finite non-empty alphabet and A^* the free monoid generated by A with unit 1. An element $w \in A^*$ is called a word and 1 is called the empty word. The product in A^* is frequently called concatenation. For $w \in A^*$, $|w|$ denotes the length of w. Any subset of A^* is called a language over A.

If $L \subset A^*$ and α is an equivalence relation on A^*, then L is an α-language iff L is a union of equivalence classes of α, i.e. $x\ \alpha\ y$ implies $x \in L$ iff $y \in L$. Given a language $L \subset A^*$ we define $\equiv_L$, the syntactic congruence of L, as follows. For $x,y \in A^*$, $x \equiv_L y$ iff $(uxv \in L$ iff $uyv \in L)$, for all $u,v \in A^*$. The quotient monoid $M_L = A^*/\equiv_L$ is called the syntactic monoid of L. The natural morphism $\varphi_L : A^* \to M_L$, mapping a word $w \in A^*$ into the equivalence class $[w]$ of $\equiv_L$ containing L, is called the syntactic morphism of L.

In some cases we must restrict the definition of a language to be a subset of A^+, the free semigroup generated by A. The concept of α-language is then defined with respect to an equivalence relation on A^+. The syntactic congruence is defined as follows. For $x,y \in A^+$, $x \equiv_L y$ iff $(uxv \in L$ iff $uyv \in L)$ for all $u,v \in A^*$ and not for all $u,v \in A^+$. The quotient semigroup $S_L = A^+/\equiv_L$ is called the syntactic semigroup of L, and $\varphi_L : A^+ \to S_L$ is the syntactic morphism. If it is necessary to distinguish between subsets of A^+ and A^* we will call them +-languages and *-languages, respectively.

We assume that the reader is familiar with the basic concepts of the theory of finite automata. A semiautomaton (Q,A,τ) consists of a finite, non-empty set Q of states, an alphabet A and a transition function τ which gives, for each state $q \in Q$ and input $a \in A$, the next state, usually denoted by qa, $qa \in Q$. A finite automaton (Q,A,τ,q_0,F) is a semiautomaton in which $q_0 \in Q$ is a designated initial state and $F \subset Q$ is the set of final states. The language accepted by

an automaton is $L = \{w \mid q_0 w \in F\}$. Every semiautomaton defines a finite monoid, its transformation monoid, whose elements are transformations of Q into itself associated with input words $w \in A^*$; $q \in Q$ is transformed to qw by w. If a finite automaton is reduced, its transformation monoid is isomorphic to the syntactic monoid of its language. Thus one can view a finite automaton as a generator of the syntactic monoid.

Most results about language families discussed in this paper have elegant statements in terms of monoids. For brevity we will avoid giving similar statements about automata. Note that each semigroup $(S,\cdot)$ can be viewed as a semiautomaton $(S,S,\cdot)$.

3. VARIETIES

3.1 S-varieties and +-varieties

Our objective is to study families of semigroups (or monoids), congruences, and languages; we begin with semigroups. It is known from universal algebra that the concept of a family of semigroups closed under division and direct products leads to important useful results. Since the semigroups corresponding to regular languages are always finite, only finite direct products are used, and a restricted notion of variety.

An S-variety V is a family of finite, non-empty semigroups satisfying the conditions:

$$S \in V \text{ and } T < S \text{ imply } T \in V, \tag{3.1}$$

$$S,S' \in V \text{ implies } S \times S' \in V. \tag{3.2}$$

Further motivation for requiring closure under division and direct product is provided from the finite automata point of view. We can think of any semiautomaton as having a certain computational power. It is natural to include in any family containing a semiautomaton S any semiautomaton T that can be realized by S. Thus, if T is a subsemiautomaton of S, then S can do anything that T can do, or S has more power than T. This also holds if T is a morphic image of S, and we are led to (3.1). Similarly, if we have semiautomata S and S' we can operate them in parallel and obtain the power of $S \times S'$. This justifies (3.2).

To avoid the degenerate case of the empty S-variety we usually assume that each S-variety V contains at least one non-empty semigroup S. In that case V also contains the one-element semigroup 1 (which is also a monoid), since $1 < S$ for any non-empty S. In terms of semiautomata this means that the one-state semiautomaton is always included. Note that $\{1\}$ is an S-variety - the trivial one.

For any family X of semigroups we denote by $(X)_S$ the least S-variety containing X.

We next turn to languages. Observe that two languages over two distinct alphabets can have isomorphic syntactic semigroups; therefore, we consider all finite alphabets. A +-class $\mathcal{V} = \{A^+V\}$ of languages consists of families A^+V of regular subsets of A^+ defined for each finite non-empty alphabet A. A +-variety $\mathcal{V}$ of languages is a +-class $\mathcal{V} = \{A^+V\}$ of languages satisfying:

$$L,L' \in A^+V \text{ implies } \bar{L} \in A^+V$$

$$\text{and } L \cup L' \in A^+V; \tag{3.3}$$

$$L \in A^+V,\ x \in A^+ \text{ implies}$$

$$x^{-1}L,\ Lx^{-1} \in A^+V; \tag{3.4}$$

$$\text{if } \varphi : B^+ \to A^+ \text{ is a semigroup morphism then}$$

$$L \in A^+V \text{ implies } L\varphi^{-1} \in B^+V. \tag{3.5}$$

Thus a +-variety is closed under boolean operations, quotients and inverse morphisms. To avoid the empty +-variety, we add $\emptyset \in A^+V$, for all A. This implies that $A^+ = \bar{\emptyset}$ is also in V. Note that the family $A^+V = \{\emptyset, A^+\}$ defines a +-variety; this is the trivial +-variety.

S-varieties and +-varieties are closely related. If $\mathcal{V}$ is a +-class let $V = (S_L \mid L \in A^+ \text{ for some } A)_S$ be the S-variety generated by the syntactic semigroups S_L of languages L from $\mathcal{V}$. We write $\mathcal{V} \to V$ if V is obtained from $\mathcal{V}$ in this way. Conversely, given any S-variety V, we define the +-class $\mathcal{V}$ by $\mathcal{V} = \{A^+V\}$ and $A^+V = \{L \mid L \subset A^+ \text{ and } S_L \in V\}$, and we write $V \to \mathcal{V}$. Here $\mathcal{V}$ is the class of all languages whose syntactic monoids are in V. It can be shown that for every S-variety V, $V \to \mathcal{V}$ implies $\mathcal{V} \to V$ and that $\mathcal{V}$ is a +-variety. Conversely, if $\mathcal{V}$ is a +-variety then $\mathcal{V} \to V$ implies $V \to \mathcal{V}$. This is summarized by Eilenberg's variety theorem [12]:

Theorem S S-varieties and +-varieties are in 1-1 correspondence.

A third point of view has been added to the variety theory by Thérien [54]. It involves varieties of congruences, which one can view as a concept falling between S-varieties and +-varieties of languages.

A +-class $\Delta = \{A^+\Delta\}$ of congruences consists of families $A^+\Delta$ of congruences of finite index on A^+ defined for each finite alphabet A. A +-variety Δ of congruences is a +-class $\Delta = \{A^+\Delta\}$ of congruences satisfying

$$\alpha \in A^+\Delta \text{ and } \beta \supset \alpha \text{ imply } \beta \in A^+\Delta; \tag{3.6}$$

$$\alpha,\alpha' \in A^+\Delta \text{ implies } \alpha \cap \alpha' \in A^+\Delta; \tag{3.7}$$

$$\text{if } \varphi : B^+ \to A^+ \text{ is a semigroup morphism then}$$

$$\alpha \in A^+\Delta \text{ implies } \alpha\varphi^{-1} \in B^+\Delta, \tag{3.8}$$

where $u\ \alpha\varphi^{-1}\ v$ holds in B^+ iff $(u\varphi)\alpha(v\varphi)$ holds in A^+. Thus a +-variety of congruences is closed under superset-taking, intersection, and inverse morphisms in the sense of (3.8). To avoid the empty +-variety we add $\omega_A \in A^+\Delta$ for each A, where ω_A is the universal congruence: $u\ \omega_A\ v$ for all $u,v \in A^+$. Note that the family $A^+\Delta = \{\omega_A\}$ defines a +-variety -- the trivial +-variety of congruences.

To establish the correspondence between congruences and semigroups define

$\Delta \to V$ if $V = \{S \mid S \simeq A^+/\alpha, \alpha \in A^+\Delta \text{ for some } A\}$

and

$V \to \Delta$ if $\Delta = \{\alpha \mid A^+/\alpha \simeq S \text{ for some } S \in V\}$.

One verifies that

$\Delta \to V$ implies $V \to \Delta$,

$V \to \Delta$ implies $\Delta \to V$,

and that +-varieties of congruences define S-varieties and vice versa. From Theorem S it follows that +-varieties of congruences and +-varieties of languages correspond.

Example 3.1 Consider the trivial S-variety $\{1\}$, the trivial +-variety $\mathcal{V} = \{A^+\mathcal{V}\}$ of languages where $A^+\mathcal{V} = \{\emptyset, A^+\}$, and the trivial +-variety $\Delta = \{\omega_A\}$ of congruences. These three concepts correspond.

Example 3.2 Let $A^+\mathcal{V} = \{L \subset A^+ \mid \text{either } L \text{ or } \bar{L} \text{ is finite}\}$. Then $\mathcal{V} = \{A^+\mathcal{V}\}$ is a +-variety of finite/cofinite languages. A semigroup S is nilpotent iff S has a zero 0 and there exists an integer $n \geq 1$ such that $S^n = 0$. One verifies that the family $\mathbf{V}$ of nilpotent semigroups is an S-variety. Let γ_k, $k \geq 1$ be the congruence on A^+ defined by $u \gamma_k v$ iff $(|u| < k$ and $u = v)$ or $(|u|, |v| \geq k)$. Thus all words of length $\geq k$ are in the same congruence class. Now let $A^+\Gamma = \{\gamma_k \mid k \geq 1\}$ and let Δ be the least +-variety of congruences containing $\{A^+\Gamma\}$. Then one can show that $\mathcal{V}$, $\mathbf{V}$ and Δ correspond; i.e. the following are equivalent:

(a) $L \subset A^+$ is finite/cofinite,

(b) S_L is nilpotent,

(c) L is a γ_k-language for some $k \geq 1$.

3.2 M-varieties and *-varieties

It would appear that the distinction between semigroups and monoids is very minor and perhaps unnecessary. In fact it is essential. Consider the language $L = a$ over the alphabet $\{a,b\}$; L is finite and its syntactic semigroup S_L is nilpotent. Its syntactic monoid is $M_L = \{1,m,0\}$ with $m^2 = 0$. The syntactic monoid of the language $L' = b^*ab^*$ is isomorphic to M_L. This shows that L and L' are not distinguishable by their syntactic monoids, though they are distinguishable by their semigroups since $S_{L'} \simeq M_{L'}$ is not nilpotent. Since many concepts appear more naturally in monoids, we consider also varieties of monoids.

A M-variety $\mathbf{V}$ is a family of finite monoids closed under division and finite direct products. The corresponding language and congruence concepts use A^* instead of A^+. Thus a *-class $\mathcal{V}$ of languages is $\mathcal{V} = \{A^*\mathcal{V}\}$, where $A^*\mathcal{V}$ is a family of regular *-languages, i.e. subsets of A^*. A *-variety of languages is a *-class closed under boolean operations (where $\bar{L} = A^* - L$), quotients (where $x^{-1}L = \{w \in A^* \mid xw \in L\}$), and inverse monoid morphisms. A *-variety of congruences is a *-class closed under superset-taking, intersection and inverse monoid morphisms. We obtain:

Theorem M M-varieties and *-varieties are in 1-1 correspondence.

Example 3.3 The trivial M-variety $\mathbf{V} = \{1\}$, the *-variety $\mathcal{V} = \{A^*\mathcal{V}\}$, where $A^*\mathcal{V} = \{\emptyset, A^*\}$, and the *-variety $\Delta = \{\omega_A\}$ correspond.

Example 3.4 The M-variety of all finite monoids, the *-variety of all regular languages, and the *-variety of all congruences of finite index correspond.

Example 3.5 Let $\mathbf{V}$ be the class of idempotent and commutative monoids, i.e. monoids M satisfying

$$m^2 = m \quad \text{and} \quad mm' = m'm$$

for all $m,m' \in M$. One verifies that $\mathbf{V}$ is an M-variety. Let $A^*\mathcal{V}$ be the boolean closure of languages $(A\text{-}a)^*$ for all $a \in A$. For instance, for $A = \{a,b\}$, $A^*\mathcal{V}$ is the set of all unions of the languages $\{1, a^+, b^+, A^*aA^* \cap A^*bA^*\}$, as one can easily verify. Finally, let the congruence α be defined by $x \alpha y$ iff x and y contain exactly the same letters, and let Δ be the *-variety generated by α. Then one can show that $\mathbf{V}$, $\mathcal{V}$ and Δ correspond.

When the meaning is clear from the context we will say variety instead of +-variety, S-variety, etc.

3.3 Closure properties

Some recent contributions to the theory of varieties deal with closure properties. A *-variety $\mathcal{V}$ is closed under star if for every alphabet A, $L \in A^*\mathcal{V}$ implies $L^* \in A^*\mathcal{V}$. Similarly $\mathcal{V}$ is closed under concatenation if $A^*\mathcal{V}$ is closed under concatenation for each alphabet A. Perrot [34] has shown that the only *-variety closed under the star operation is the *-variety of all regular languages. The proof utilizes a theorem of Pin [37]. Perrot also observed that the smallest nontrivial *-variety closed under concatenation is the *-variety of star-free languages. This follows from Schützenberger's theorem [42]. For results concerning closure under the shuffle operation, under literal morphisms and inverse substitution see the survey by Perrot and Pin [36]. Straubing found a characterization of all varieties closed under concatenation [51]. He showed that, if $\tilde{\mathcal{V}}$ is the closure of nontrivial variety $\mathcal{V}$ under boolean operations and concatenation, then $\tilde{\mathbf{V}}$ (the M-variety corresponding to $\tilde{\mathcal{V}}$) is the smallest M-variety which contains $\mathbf{V}$ (corresponding to $\mathcal{V}$) and is closed under inverse images of aperiodic morphisms.

Straubing has studied language varieties corresponding to closure under the operation of taking power sets [53]. If M is a finite monoid, then so is $P(M)$, the power set of M. If $\mathbf{V}$ is an M-variety, let $P(\mathbf{V})$ be the M-variety generated by $\{P(M) \mid M \in \mathbf{V}\}$. $\mathbf{V}$ is said to be closed under power sets if $\mathbf{V} = P(\mathbf{V})$. For example, if $\mathbf{V} = \{1\}$ is the trivial M-variety, then $P(\mathbf{V})$ is the M-variety J_1 of

idempotent and commutative monoids. Next $P(J_1)$ turns out to be the M-variety of commutative aperiodic monoids. This last variety is closed under power sets. The closure of the S-variety of aperiodic semigroups is the S-variety of all finite semigroups. Straubing asked whether the sequence $V \subset P(V) \subset P^2(V) \subset \ldots$ can ever be infinite. The question was answered negatively by Pin who showed that $P^4(V) = P^5(V)$ [38]. However, the question whether $P^3(V) = P^4(V)$ is still open. The operation on language varieties corresponding to the operation $V \to P(V)$ on M-varieties is the operation of literal morphism [53] (called very fine morphism by Eilenberg).

4. SUBWORD-COUNTING LANGUAGES

4.1 Counting congruences

The largest proper subfamily of the family of finite monoids that has been studied is the M-variety M_{SOL} of monoids whose groups are solvable. This family was first characterized by Straubing [49,50] in 1978. It was also studied in a somewhat more general setting by Thérien in 1979 [54,55]. We begin with Thérien's approach which uses congruences.

Let $N = \{0,1,\ldots\}$ be the set of all non-negative integers. For $t \geq 0$, $m \geq 1$ we define the congruence $\theta_{t,m}$ on N as follows: $a\ \theta_{t,m}\ b$ iff ($a < t$ and $b = a$) or ($a \geq t$, $b \geq t$, and $a \equiv b \pmod m$). We say that $\theta_{t,m}$ counts modulo m with threshold t.

Let A be a finite, non-empty alphabet and $x,u \in A^*$ with $u = a_1 \ldots a_n$, where $a_i \in A$ for $i = 1,\ldots,n$, and $n \geq 1$. The binomial coefficient $|x|_u$ is defined as the number of factorizations: $x = v_0 a_1 v_1 a_2 \ldots v_{n-1} a_n v_n$ with $v_0,\ldots,v_n \in A^*$. In other words $|x|_u$ gives the number of ways that u appears as a subword of x. Also we adopt the convention that $|x|_1 = 1$ for all $x \in A^*$. We say that u appears in x in context $(v_0,\ldots,v_n)$ in such a factorization.

We now define an equivalence relation $\alpha_{t,m,\ell}$ on A^* that counts the number of times that each subword of length $\leq \ell$ appears in a given word x, where the counting is done modulo m with threshold t, in the following sense. For $t \geq 0$, $m \geq 1$, $\ell \geq 0$ and for $x,y \in A^*$,

$$x\ \alpha_{t,m,\ell}\ y \quad \text{iff} \quad |x|_u\ \theta_{t,m}\ |y|_u$$

for all $u \in A^*$ with $|u| \leq \ell$. One verifies that $\alpha_{t,m,\ell}$ is a congruence of finite index on A^*.

Counting in context with respect to a congruence is defined next. Suppose β is a congruence of finite index on A^*. Let $u = a_1 \ldots a_n$, and let $W = (w_0,\ldots,w_n)$ be an $(n+1)$-tuple of words that we will interpret as a context for u. If $n = 0$, $|x|_{1,W_\beta} = 1$ if $x\ \beta\ w_0$ and $|x|_{1,W_\beta} = 0$ otherwise. For $n > 0$, $|x|_{u,W_\beta}$ is the number of factorizations $x = v_0 a_1 \ldots a_n v_n$ with $v_j\ \beta\ w_j$ for $j = 1,\ldots,n$.

We will use the notation $\beta\alpha_{t,m,\ell}$ for the congruence that counts subwords of length $\leq \ell$ modulo m with threshold t in context with respect to β. (Note that this is not the usual composition of relations.) More precisely for $x,y \in A^*$

$$x\ \beta\alpha_{t,m,\ell}\ y \quad \text{iff} \quad |x|_{u,W_\beta}\ \theta_{t,m}\ |y|_{u,W_\beta},$$

for all $u \in A^*$ with $|u| \leq \ell$ and for all $W = (w_0,\ldots,w_n)$ with $w_j \in A^*$ for $j = 1,\ldots,n$. Note that, although the number of distinct contexts is infinite for each n, the number of equivalence classes of contexts with respect to the congruence β is finite since β is assumed to be of finite index. Thus $\beta\alpha_{t,m,\ell}$ is also of finite index, and one verifies that it is a congruence on A^*.

Starting with the universal congruence ω on A^* we construct a sequence of congruences of finite index, where each successive congruence counts in context with respect to increasingly more complex congruences. This is done by induction on i, the number of iterations, as follows:

$$\alpha^0_{t,m,\ell} = \omega,$$

and $\alpha^{i+1}_{t,m,\ell} = \alpha^i_{t,m,\ell}\ \alpha_{t,m,\ell}$, for $i \geq 0$.

Note that $\alpha^1_{t,m,\ell} = \alpha^0_{t,m,\ell}\ \alpha_{t,m,\ell} = \omega\alpha_{t,m,\ell} = \alpha_{t,m,\ell}$, since counting in any context W with respect to ω is equivalent to ignoring the context. $\alpha^{i+1}_{t,m,\ell}$ is the congruence that counts subwords of length $\leq \ell$ modulo m with threshold t in context with respect to the congruence $\alpha^1_{t,m,\ell}$.

4.2 Thérien's hierarchies

It turns out that a family of congruences derived from $\alpha^1_{t,m,\ell}$ is sufficient to characterize all finite monoids whose groups are solvable. Furthermore, by varying the four parameters (threshold, modulus, length, and iteration depth) one obtains hierarchies of monoids and languages that correspond to some well-known mathematical concepts. Each congruence $\alpha^i_{t,m,\ell}$ generates a *-variety of congruences that we denote by $\Delta^i_{t,m,\ell}$ as follows:

$$A^*\Delta^i_{t,m,\ell} = \{\alpha \mid \alpha \supset \alpha^i_{t,m,\ell}\}.$$

The notation * in place of a parameter $p \in \{t,m,\ell,i\}$ denotes the union taken over all values of p. Thus $\Delta^i_{*,m,\ell} = \bigcup_{t \geq 0} \Delta^i_{t,m,\ell}$, etc.

We now discuss Thérien's results about hierarchies defined by subword counting.

(i) Solvable groups

For concepts from group theory see Hall [16].

The left part of fig. 1 corresponds to varieties of congruences where the parameter t (the threshold) is set to 0, i.e. we concentrate exclusively on modulo counting. With this restriction, counting letters without context yields the variety $\Delta^1_{0,*,1} = \Delta_{0,*,1}$ which corresponds to the variety G_{ARFI} of abelian

groups. Counting words without context yields $\Delta^1_{0,*,*}$ which corresponds to the variety G_{NIL} of nilpotent groups. Starting with abelian groups, as we increase the parameter i (representing the complexity of contexts), we obtain a hierarchy defined by $\Delta^1_{0,*,1}$ which corresponds precisely to the variety $G_{DER\leq i}$ of solvable groups of derived length $\leq i$. Similarly, starting with nilpotent groups and increasing i, we obtain a hierarchy $\Delta^i_{0,*,*}$ corresponding to the variety $G_{FIT\leq i}$ of solvable groups of fitting length $\leq i$. Finally, $\Delta^*_{0,*,*}$ corresponds to the variety G_{SOL} of all solvable groups.

In summary, the congruence approach provides a new point of view for solvable groups and permits us to capture many classical concepts from group theory in one framework.

(ii) Aperiodic monoids

The right part of fig. 1 shows hierarchies of aperiodic monoids, i.e. monoids whose groups are trivial, one-element groups. To obtain this we set the parameter m (the modulus) to 1. Thus modulo counting is impossible, and we concentrate on threshold counting only. The variety analogous to abelian groups is the variety $M_{AP,COM}$ of aperiodic commutative monoids defined by $\Delta^1_{*,1,1} = \Delta_{*,1,1}$. This corresponds to counting letters to a threshold without context. Counting words to a threshold without context leads to $\Delta^1_{*,1,*}$ which corresponds to the variety of J-trivial monoids discussed in the next two sections. Note that $M_{J\text{-}TRIV}$ is analogous to G_{NIL}; both count subwords without context, in one case to a threshold t, in the other modulo m.

Note that varieties analogous to those defined by derived length and fitting length for groups exist also in the case of aperiodic monoids but, to the best of our knowledge, these varieties have not been studied. Fig. 1 suggests that their study may be very fruitful [54].

(iii) Monoids with solvable groups

Combining threshold counting with modulo counting leads to $\Delta^*_{*,*,*}$ which corresponds to M_{SOL}.

Some of the parameters of $\Delta^i_{t,m,\ell}$ can be eliminated in certain cases [54]. For example, one can reach solvable groups by counting letters only, instead of arbitrary subwords since $\Delta^*_{0,*,*} = \Delta^*_{0,*,1}$. However, a higher iteration index is required to define the same families. Similarly, counting letters with threshold 1 is sufficient since $\Delta^*_{*,*,*} = \Delta^*_{1,*,1}$. Also, the concept of context can be made one-sided as described below.

4.3 Straubing's counting

Given a language $L \subset A^*$ and a word $w \in A^*$ let w_L be the number of non-empty prefixes of w that belong to L. For example let $L = (a \cup ba^*b)^*$. Then $(abb)_L = 2$ and $(baa)_L = 0$. Also, for any $w \in A^*$, w_{A^*} is precisely the

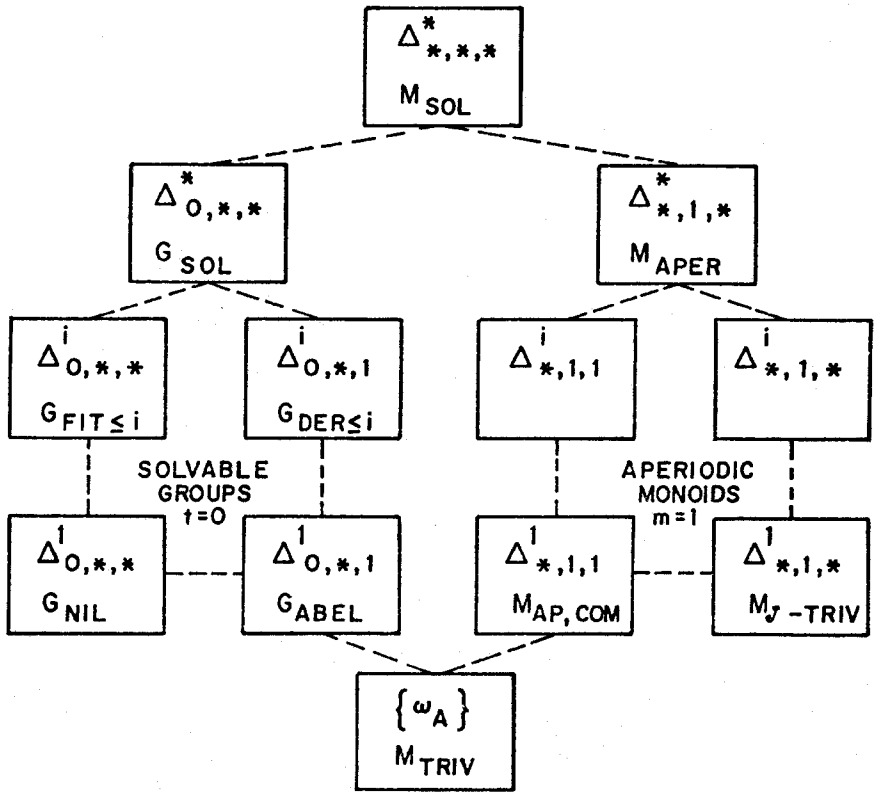

Fig. 1. Thérien's hierarchies.

length of w. Now let $L \subset A^*$, $m \geq 1$ and $0 \leq r < m$. Define the language

$$< L,r,m > = \{w \in A^* \mid w_L \equiv r \pmod m\}.$$

In particular, the language $< La,r,m >$ with $a \in A$ plays a key role in Straubing's work. It is the set of all words w such that the number of factorizations $w = xay$ with $x \in L$ is congruent to r modulo m. One can view this as a sort of counting of letters in left context defined by L.

Denote by G_{SOL} the variety of languages corresponding to G_{SOL}; i.e. $L \in A^*G_{SOL}$ iff $M_L \in G_{SOL}$. Similarly, we define M_{SOL}. Straubing's characterization of G_{SOL} is given by the following theorem [50]:

Theorem G_{SOL} (Straubing)

The family A^*G_{SOL} of languages over A whose syntactic monoids are solvable groups, is the smallest family F of languages such that

(a) $\emptyset \in F$,

(b) $L,L' \in F$ implies $\bar{L} \in F$ and $L \cup L' \in F$,

(c) $L \in F$ implies $< La,r,m > \in F$, for all $a \in A$, $m \geq 1$ and $0 \leq r < m$.

Thus A^*G_{SOL} is characterized as the smallest family containing the empty language and closed under boolean operations and Straubing counting.

Recall that Kleene's theorem characterizes A^*M, the variety of languages over A whose syntactic monoids are finite, as the smallest family containing letters and closed under boolean operations, concatenation and star. Thus we have:

Theorem M (Kleene)

The family A^*M of languages over A whose syntactic monoids are finite is the smallest class F of languages such that

(a) $\{a\} \in F$ for all $a \in A$,

(b) $L,L' \in F$ implies $\bar{L} \in F$ and $L \cup L' \in F$,

(c) $L,L' \in F$ implies $LL' \in F$,

(d) $L \in F$ implies $L^* \in F$.

Straubing's theorem for M_{SOL} is analogous to Kleene's theorem, with the star operator replaced by counting. Thus we have [50]:

Theorem M_{SOL} (Straubing)

The family A^*M_{SOL} of languages over A whose syntactic monoids have solvable groups is the smallest family F of languages satisfying (a), (b) and (c) of Theorem M and

(d) $L \in F$ implies $< La,r,m > \in F$ for all $a \in A$, $m \geq 1$ and $0 \leq r < m$.

A particular case of Straubing's theorem with the parameter $m = 1$ yields Schützenberger's theorem characterizing A^*M_{APER} as the smallest family of languages containing letters and closed under boolean operations and concatenation.

Straubing has also found two hierarchies related to his counting operation. For any family F of languages over an alphabet A let FB denote the boolean closure of F. Define $A_0 = \{\emptyset\}$, and for $i \geq 0$, $a \in A$, $m \geq 1$, $0 \leq r < m$,

$$\mathcal{B}_i = A_iB$$

and $$A_{i+1} = \mathcal{B}_i \cup \{< La,r,m > \mid L \in \mathcal{B}_i\}.$$

Then Straubing shows that $L \in \mathcal{B}_i$ iff M_L is a solvable group of derived length i; i.e. derived length is directly related to the number of nested counting operations. Similarly let $C_0 = \{\{a\} \mid a \in A\}$ and for $i \geq 0$, $a \in A$, $m \geq 1$, $0 \leq r < m$, let $\mathcal{D}_i$ be the closure of C_i under boolean operations and concatenation and $C_{i+1} = \mathcal{D}_i \cup \{< La,r,m > \mid L \in \mathcal{D}_i\}$.

Then $L \in \mathcal{D}_i$ iff the abelian group complexity of M_L is $\leq i$. For details see [50].

5. STAR-FREE LANGUAGES

In this section we describe recent results concerning star-free languages, i.e. the variety of languages whose semigroups are aperiodic. As we have seen, this is the case where only threshold counting is permitted.

5.1 Dot-depth hierarchy

For any family F of +-languages (subsets of A^+) let FS be the semigroup closure of F, i.e. the smallest family containing F and closed under concatenation. Define the following hierarchy:

$$\mathcal{B}_0 = \{\{a\} \mid a \in A\}B$$

and for $i \geq 0$

$$\mathcal{B}_{i+1} = \mathcal{B}_iSB = \mathcal{B}_0(SB)^{i+1}.$$

One can interpret $\mathcal{B}_i$ as the set of all languages that can be constructed from the letters of the alphabet by using no more than i levels of concatenation (i.e. "dot" operation). Hence the sequence $\mathcal{B}_0 \subset \mathcal{B}_1 \subset \mathcal{B}_2 \subset \ldots$ is called the dot-depth hierarchy [9,10]. Each family $\mathcal{B}_i$ defines a +-variety of languages and $\bigcup_{i \geq 0} \mathcal{B}_i$ defines the +-variety of +-languages whose syntactic semigroups are aperiodic. The question whether the dot-depth hierarchy was finite was posed in 1968; it was shown to be infinite by Brzozowski and Knast in 1977 [6]. The proof of this result is not easy and uses congruences of finite index related to decompositions of words. A somewhat different proof was found by Straubing [52]. The problem of characterizing $\mathcal{B}_i$ for $i \geq 2$ remains open, and it is not known if one can decide whether a given star-free language is in $\mathcal{B}_i$.

5.2 Depth-one languages

Languages in $\mathcal{B}_1$ were characterized by Simon in terms of certain congruences [45]. Let $W = (w_1,\ldots,w_n)$ be an n-tuple of words of length ℓ. We say that W occurs in a given word x iff there exist words $u_1,\ldots,u_n,v_1,\ldots,v_n$ such that $|u_1| < |u_2| < \ldots < |u_n|$ and $x = u_i w_i v_i$, for $i = 1,\ldots,n$. For example, let $x = ababb$. Then the following pairs (2-tuples) of words of length 2 appear in x: (ab,ba),(ab,ab),(ab,bb),(ba,ab),(ba,bb). Note that the pair (ba,ab) occurs only with "overlap", as $x = a(ba)bb = ab(ab)b$.

For $x \in A^*$, define the front f_ℓ and the tail t_ℓ of length ℓ of x for $\ell \geq 0$:

$xf_\ell = xt_\ell = x$, if $|x| < \ell$;

xf_ℓ is the prefix of length ℓ of x, if $|x| \geq \ell$;

xt_ℓ is the suffix of length ℓ of x, if $|x| \geq \ell$.

Also for each $n \geq 0$, $\ell \geq 1$, $x \in A^*$ define

$$xU_{n,\ell} = \{W=(w_1,\ldots,w_i) \mid |w_1| = \ldots = |w_i| = j,\ 0 \leq i < n,\ 1 \leq j \leq \ell \text{ and } W \text{ occurs in } x\},$$

where $xU_{0,j} = \{1\}$ for all $x \in A^*$ and $j \geq 1$, by convention. We now define the congruence $\tilde{n}_\ell$ which turns out to characterize depth-one languages. For $x,y \in A^+$, $n \geq 0$, $\ell \geq 1$,

$$x \tilde{n}_\ell y \text{ iff } xf_{\ell-1} = yf_{\ell-1},\ xt_{\ell-1} = yt_{\ell-1} \text{ and } xU_{n,\ell} = yU_{n,\ell}.$$

Simon has shown that a +-language L is of depth-one iff it is a $\tilde{n}_\ell$ language for some $n \geq 0$, $\ell \geq 1$. However, this characterization is not constructive for it is not known how to find n and ℓ.

The various families defined by $\tilde{n}_\ell$ are shown in fig. 2. We briefly discuss some special cases. For the time being consider only the central rectangle of languages defined by various values of n and ℓ.

(i) $n = 0$, $\ell = 1$: Trivial

This corresponds to the trivial +-variety $\{\emptyset, A^+\}$ and the trivial S-variety $\{1\}$. Note that $\{1\}$ is also an M-variety.

(ii) $n = 0$, $\ell = *$: Generalized-definite

Here we are testing only the front and tail of length $\ell - 1$ for each word. This defines the +-variety corresponding to the S-variety satisfying

$$eSe = e$$

for all idempotents $e \in S$. [8,31,56]

(iii) $n = 1$, $\ell = 1$: Letter-testable

Here the test for front and tail is trivial, and two words are congruent iff they contain the same letters. This corresponds to the M-variety of idempotent and commutative monoids discussed in Example 3.5.

(iv) $n = 1$, $\ell = *$: Locally testable

For a given ℓ, the congruence tests the front and tail of length $\ell - 1$ and the set of segments of length ℓ that appear in a given word. The corresponding S-variety satisfies the condition:

eSe is idempotent and commutative

for all idempotents $e \in S$. [8,12,26,56,57,58]

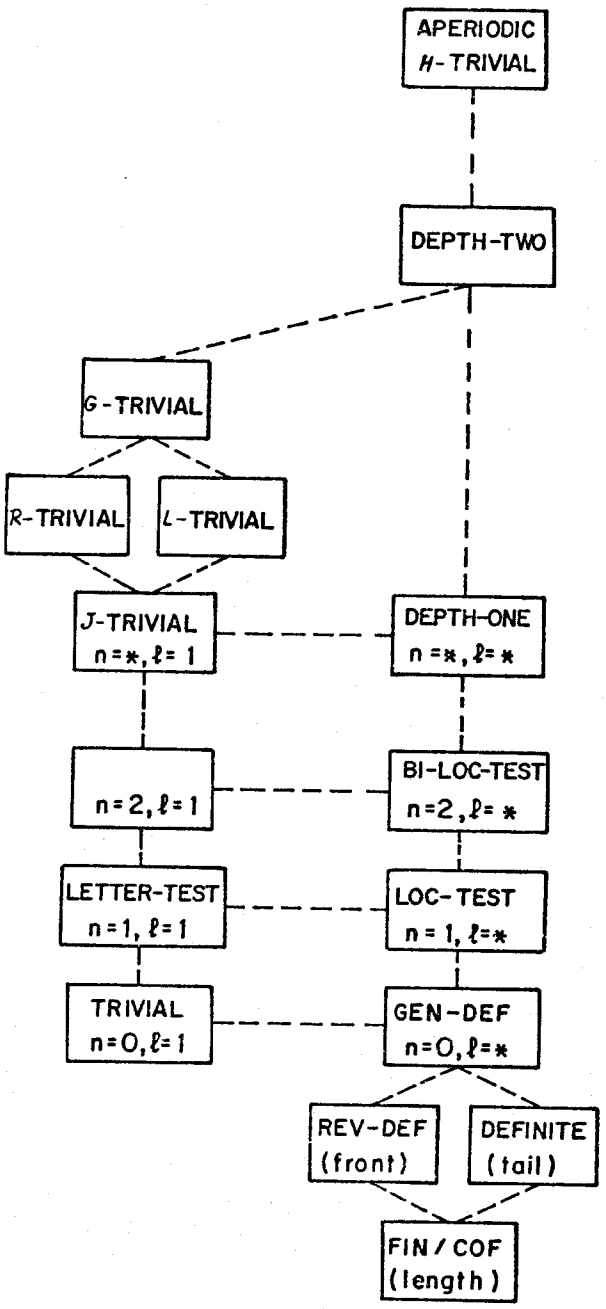

Fig. 2. Simon's hierarchies.

(v) $n = 2$, $\ell = *$: Bi-locally-testable

The test consists of front, tail, and pairs of segments appearing in a given word. A semigroup characterization of this family of languages was found by Knast in 1979 [21]. The characterization is rather involved.

No characterizations are as yet known for the families defined by $\underset{n}{\sim}_{\ell}$ for $n > 2$.

(vi) $n = *$, $\ell = 1$: Piecewise-testable

For a given n, the congruence tests the set of all n-tuples of letters appearing in a given word. The front and tail tests are degenerate. This family defines the M-variety of J-trivial monoids, i.e. monoids M satisfying

$$MmM = Mm'M \text{ implies } m = m'$$

for all m, $m' \in M$. This remarkable correspondence was found by Simon in 1972 [45,46,12].

(vii) $n = *$, $\ell = *$: Depth-one

Consider, on the one hand, the M-varieties defined by $\underset{n}{\sim}_{1}$ in the left-most column of the rectangle in fig. 2, and, on the other hand, the corresponding S-varieties defined by $\underset{n}{\sim}_{*}$ in the right-most column of the rectangle. For $n = 0$ and 1 we have the following pattern:

$\ell = 1$. L is in the family iff its syntactic monoid M_L has property P_n.

$\ell = *$. L is in the family iff, for each idempotent e in S_L, the monoid eS_Le has property P_n.

We also have for $n = *$, $\ell = 1$: L is in the family iff M_L is J-trivial. It is natural to ask whether this pattern extends to case $n = *$, $\ell = *$, i.e. whether the condition eS_Le is J-trivial for each idempotent $e \in S_L$ characterizes the family $\mathcal{B}_1$ of depth-one languages. Simon showed that the condition is necessary [45], but the question of sufficiency remained open for many years. In 1978 Knast [22] showed that the condition is not sufficient. A counter-example is provided by the language L over $\{a,b,c,d\}$:

$$L = (ab^+ \cup ac^+)^*ab^+d(c^+d \cup b^+d)^*.$$

Although eS_Le is J-trivial for each $e \in S$, the language is not of depth-one.

Knast also proposed a characterization of $\mathcal{B}_1$ in 1978 [22] as follows: A language L is in $\mathcal{B}_1$ iff there exists an integer $n > 0$ such that for all idempotents $e_1, e_2 \in S_L$ and and for all $a,b,c,d \in S_L$

$$(e_1ae_2b)^ne_1ae_2de_1(ce_2de_1)^n = (e_1ae_2b)^ne_1(ce_2de_1)^n.$$

A key point in the proof is a difficult theorem on certain congruences on graphs [23]. The details of the proof of this result on graphs still need to be supplied.

The families of languages in the "diamonds" outside the central square of fig. 2 will be discussed shortly.

The notion of local testability can be extended as follows. Define the congruence ℓ/m on A^+ as follows. For all $x,y \in A^+$

$$x \; \ell/m \; y \text{ iff } xf_{\ell-1} = yf_{\ell-1},\ xt_{\ell-1} = yt_{\ell-1}$$

$$\text{and } |x|_u \equiv |y|_u \mod m$$

for all u with $|u| = \ell$. A language is strictly periodically locally testable iff it is an ℓ/m-language for some $\ell \geq 1$, $m \geq 1$. Knast has shown that L is strictly periodically locally testable iff S_L is finite and for each idempotent $e \in S_L$, the monoid eS_Le is an abelian group [24]. Of course these languages are no longer star-free, in general.

6. R-TRIVIAL AND RELATED LANGUAGES

Before discussing R-trivial languages, let us consider the "diamond" in the lower right-hand corner of fig. 2. It consists of the generalized definite languages (where the front and tail are tested), the definite languages (tail only), the reverse definite languages (front only) and the finite/cofinite languages (length only). The conditions for membership in these families can be restated as follows:

(i) $eS = e = Se$ - Every idempotent is a zero.

(ii) $eS = e$ - Every idempotent is a left zero.

(iii) $Se = e$ - Every idempotent is a right zero.

(iv) $eSe = e$ - Every idempotent is what one might call a "half-zero" (two e's are needed around $s \in S$ to yield e).

The concepts of the four types of zeros can be generalized as follows. In any monoid M, define the subset P_m of M for each $m \in M$ by:

$$P_m = \{m' \in M \mid m \in Mm'M\}.$$

One can view P_m as the alphabet of m in M, i.e. as the set of all elements in M with which m can be written. Let $M_m = P_m^*$ be the submonoid of M generated by P_m. The concepts of local zero, local left zero, etc. are defined with respect to sets of the type M_m. Thus we have:

(i)' $eM_e = e = M_ee$ - Every idempotent is a local zero. This turns out to be a characterization of the M-variety of J-trivial monoids [2]. The corresponding languages are the piecewise-testable languages, where membership of a word in a language is determined by the n-tuples of letters occurring in the word.

(ii)' $eM_e = e$ - Every idempotent is a local left zero. This characterizes the M-variety of R-trivial monoids, i.e. those satisfying $mM = m'M$ implies $m = m'$, for all m, $m' \in M$.

The corresponding languages have been studied by Eilenberg [12] and Brzozowski and Fich [5]. Membership in such a language is determined not only by the set of n-tuples of letters appearing in a given word, but also by their order of appearance, from left to right. The corresponding automata are precisely the partially ordered automata, where the partial order on the states is defined by reachability.

(iii)' $\underline{M_e e = e}$ - Every idempotent is a local right zero. This defines L-trivial monoids where $Mm = Mm'$ implies $m = m'$. The languages are those where membership is determined by n-tuples of letters and their order of appearance from the right.

(iv)' $\underline{eM_e e = e}$ - Every idempotent is a local half-zero. These monoids are called G-trivial. The languages are analogous to generalized-definite languages, and the order of appearance of n-tuples of letters is considered both from the left and from the right [13,14]. We will return to this family shortly.

As seen from fig. 2 the generalized definite diamond constitutes the beginning of the depth-one finite/cofinite hierarchy. The analogy above suggests that the G-trivial diamond is at the bottom of a similar hierarchy. Thus the next family of monoids to be studied would be the family of monoids M satisfying: $eM_e e$ is idempotent and commutative for each idempotent e in M. The language family is a generalization of the locally-testable family, but it has not been studied.

In general, R, L and G-trivial languages are not of depth-one, but they are all of depth 2. Also, G-trivial languages do not contain B_1.

We have introduced the G-trivial family of monoids through the language hierarchies. However, they had been previously studied in 1976 by Schützenberger [43] from a different point of view. He has shown some remarkable properties of this family. The concatenation LL' of two languages L and L' is called unambiguous iff each $w \in A^*$ admits at most one factorization $w = uu'$ with $u \in L$, $u' \in L'$. Define the family G of languages over A as the smallest family satisfying

$\{a\} \in G$ for all $a \in A$,

$B^* \in G$ for all $B \in A$,

G is closed under disjoint union and unambiguous concatenation.

Theorem G (Schützenberger)

A language L is in the family G iff its syntactic monoid M_L is G-trivial.

In a similar fashion Schützenberger found characterization of the languages corresponding to R-trivial, L-trivial and J-trivial monoids by restricting unambiguous concatenation to deterministic, reverse deterministic and bi-deterministic (unambiguous) concatenation, respectively. (A product LL' is deterministic iff either L is prefix-free or LL' is prefix-free and $L' \in A$. L is prefix-free if no word of L is a prefix of any other word of L.)

7. CODES

A code C over alphabet A is a subset of A^+ such that for all $u_i, v_i \in C$

$$u_i \ldots u_n = v_1 \ldots v_m$$

implies $n = m$ and $u_i = v_i$ for $i = 1,\ldots,n$. If C is a code and $w \in C^*$ then w is called a message. The definition above implies that every message in uniquely decipherable. A prefix code is a code in which no word is the prefix of any other word. A prefix code C is full iff for each $w \in A^*$

$$w A^* \cap C A^* \neq \emptyset;$$

i.e. each w is either a prefix of some word in C or has a prefix which is in C.

For a general survey of fundamentals of the theory of codes, the reader is referred to [35]. We are interested here in the role that codes play in variety theory as discussed in [36].

At first glance, the study of languages of the form C^* where C is a full finite prefix code appears to be very restrictive, since one would expect such languages to have very simple properties. In fact the opposite is true. In discussing the closure properties of varieties in Section 3 we mentioned a theorem of Pin. The complete statement of that theorem is [37]:

Theorem C (Pin)

For every finite monoid M there exists a full finite prefix code C such that M divides the syntactic monoid M_{C^*} of C^*.

This arbitrary complexity that can be represented by prefix codes provides sufficient motivation for their study. Also the star operation is rather poorly understood [4]. The study of particular languages like codes should shed some light on this. Note that the star operation on codes is unambiguous.

Further evidence that codes are important in variety theory is provided by the following results of Pin [39,36]. We say that a +-variety V is generated by a class C of prefix codes if V is the least variety V' such that for any alphabet A^+, A^+V' contains all semigroups C^+ with $C \subset A^+$ and $C \in C$. We have the following results [39]. A language $L \subset A^*$ is pure iff for any $u \in A^*$ and $n > 0$, $u^n \in L^*$ implies $u \in L$. L is very pure iff for all $u,v \in A^*$ $u,v \in A^*$ $uv \in L^*$ and $vu \in L^*$ imply $u \in A^*$ and $v \in A^*$.

1. The +-variety of regular languages is generated by the class of all finite prefix codes.
2. The +-variety of star-free languages is generated by the class of all pure finite prefix codes.
3. The +-variety of locally testable languages is generated by all very pure finite prefix codes.

For further recent results regarding finite codes see also [19,30,32,44].

8. OTHER RESULTS

We conclude the paper by mentioning two recent results about regular languages that are not related to variety theory.

8.1 Limited languages

A language $L \subset A^*$ is limited iff there exists an integer n such that $(L \cup 1)^n = L^*$. The star operation on a limited language is weak in the sense that it can be replaced by a finite power. The question of deciding whether a given regular language is limited was posed by the author in 1966. The question was answered positively in 1977 by Hashiguchi [17] and, independently, by Simon [47]. The first proof is combinatorial in nature, whereas the second is more algebraic and uses semigroup theory.

8.2 Regular equations

Let A be a finite alphabet. Systems of equations of the form shown below are well-known. Let

$$X_i = \bigcup_{a \in A} a X_{i,a} \cup \delta_i, \quad i = 1,\ldots,n, \tag{8.1}$$

where $X_{i,a} \in \{X_1,\ldots,X_n\}$, for all $i \in \{i,\ldots,n\}$, and for all $a \in A$ and $\delta_i \in \{\emptyset,\{1\}\}$. Then the system (8.1) is in 1-1 correspondence with a deterministic finite automaton. It is also well-known that non-deterministic finite automata correspond to systems of the form

$$X_i = \bigcup_{a \in A} a F_{i,a} \cup \delta_i, \quad i = 1,\ldots,n, \tag{8.2}$$

where each $F_{i,a}$ is a finite (possibly empty) union of elements from the set $\{X_1,\ldots,X_n\}$. It is natural to consider a more general system than (8.2), where the $F_{i,a}$ are replaced by arbitrary boolean functions in the variables $X_1,\ldots,X_n$. For example, we might have

$$\begin{aligned} X_1 &= a(\bar{X}_1 \cup \bar{X}_2) \cup bX_2 \\ X_2 &= a\bar{X}_1 \cup b((\bar{X}_1 \cap \bar{X}_2) \cup (X_1 \cap X_2)). \end{aligned} \tag{8.3}$$

Such systems of equations were studied by Brzozowski and Leiss [7]. It was shown that they correspond to boolean automata, and have unique regular solutions. In some cases these systems provide very concise descriptions of regular languages. For example, the two equations (8.3) define a language whose reduced deterministic automaton has 16 states.

9. CONCLUDING REMARKS

Due to time and space restrictions we have been unable to include all the recent results about regular languages. However, we have attempted to present the highlights of the work done during the last five years. We firmly believe that many of the results discussed are very fundamental and quite significant.

We refer the interested reader to [4] where several old open problems (including the star height problems) are discussed in some detail. For a characterization of star height preserving morphisms see [18].

REFERENCES

[1] J.A. Brzozowski, Canonical regular expressions and minimal state graphs for definite events, Mathematical Theory of Automata, Polytechnic Press, Brooklyn, 1962, 529-561.

[2] J.A. Brzozowski, A generalization of finiteness, Semigroup Forum, vol. 13, 1977, 239-251.

[3] J.A. Brzozowski, Hierarchies of aperiodic languages, R.A.I.R.O., Informatique Théorique, vol. 10, 1976, 33-49.

[4] J.A. Brzozowski, Open problems about regular languages, Proc. Formal Language Theory Symposium, Santa Barbara, December 1979, to appear. (Also University of Waterloo Report CS-80-03, 1980.)

[5] J.A. Brzozowski and F.E. Fich, Languages of R-trivial monoids, J. Computer and System Sciences, vol. 20, 1980.

[6] J.A. Brzozowski and R. Knast, The dot-depth hierarchy of star-free languages is infinite, J. Computer and System Sciences, vol. 16, 1978, 37-55.

[7] J.A. Brzozowski and E. Leiss, On equations for regular languages, finite automata, and sequential networks, Theoretical Computer Science, vol. 10, 1980, 19-35.

[8] J.A. Brzozowski and I. Simon, Characterizations of locally testable events, Discrete Mathematics, vol. 4, 1973, 243-271.

[9] R.S. Cohen and J.A. Brzozowski, On star-free events, Proc. Hawaii Internat. Conf. Syst. Sci., Honolulu, 1968, 1-4.

[10] R.S. Cohen and J.A. Brzozowski, Dot-depth of star-free events, J. Computer and System Sciences, vol. 5, 1971, 1-15.

[11] Eilenberg, Automata, languages and machines, vol. A, Academic Press, New York, 1974.

[12] Eilenberg, Automata, languages and machines, vol. B, Academic Press, New York, 1976.

[13] F.E. Fich, Languages of R-trivial and related monoids, M.Math Thesis, University of Waterloo, 1978.

[14] F.E. Fich and J.A. Brzozowski, A characterization of a dot-depth two analogue of generalized definite languages, Proc. 6th ICALP, Lecture Notes in Computer Science, vol. 71, Springer-Verlag, Berlin, 1979, 230-244.

[15] A. Ginzburg, About some properties of definite, reverse definite and related automata, IEEE Trans. Electronic Computers, vol. EC-15, 1966, 806-810.

[16] Hall, Theory of Groups, Chelsea, New York, 1976.

[17] K. Hashiguchi, A decision procedure for the order of regular events, Theoretical Computer Science, vol. 8, 1979, 69-72.

[18] K. Hashiguchi and N. Honda, Homomorphisms that preserve star height, Information and Control, vol. 30, 1976, 247-266.

[19] K. Hashiguchi and N. Honda, Properties of code events and homomorphisms over regular

events, J. Computer and System Sciences, vol. 12, 1976,352-367.
[20] S.C. Kleene, Representation of events in nerve nets and finite automata, Automata Studies, (Shannon and McCarthy, eds.,) Princeton University Press, Princeton, 1954, 3-41.
[21] R. Knast, Bilocally testable languages, 1979, (unpublished).
[22] R. Knast, A semigroup characterization of dot-depth one languages, 1979, (unpublished).
[23] R. Knast, Some theorems on graph congruences, 1979, (unpublished).
[24] R. Knast, Periodic local testability, 1979, (unpublished).
[25] Lallement, Semigroups and combinatorial applications, Wiley, New York, 1979.
[26] R. McNaughton, Algebraic decision procedures for local testability, Math. Systems Theory, vol. 8, 1974, 60-76.
[27] R. McNaughton and S. Papert, The syntactic monoid of a regular event, Algebraic Theory of Machines, Languages, and Semigroups (Arbib, ed.), Academic Press, New York, 1968, 297-312.
[28] McNaughton and Papert, Counter-free automata, The MIT Press, Cambridge, 1971.
[29] J. Myhill, Finite automata and representation of events, WADC Tech. Rept. 57-624, 1957.
[30] D. Perrin, Codes bipréfixes et groupes de permutations,Thèse de Doctorat d'Etat, Université Paris VII, 1975.
[31] D. Perrin, Sur certains semigroupes syntaxiques, Seminaires de l'IRIA, Logiques et Automates, 1971, 169-177.
[32] D. Perrin, La transitivité du groupe d'un code bipréfixe fini, Mathematische Zeitschrift, 1977, 283-287.
[33] M. Perles, M.O. Rabin and E. Shamir, The theory of definite automata, IEEE Trans. Electronic Computers, vol. EC-12, 1963, 233-243.
[34] J.F. Perrot, Variétés de langages et opérations, Theoretical Computer Science, vol. 7, 1978, 197-210.
[35] J.F. Perrot, Informatique et algèbre: la théorie des codes à longueur variable, Proc. 3rd GI Conference, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1977.
[36] J.F. Perrot and J.E. Pin, Finite syntactic monoids, Proc. FCT 1979, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1979.
[37] J.E. Pin, Sur le monoide syntactique de L^* lorsque L est un langage fini, Theoretical Computer Science, vol. 7, 1978, 211-215.
[38] J.E. Pin, Variétés de langages et monoide des parties, Semigroup Forum, (to appear).
[39] J.E. Pin, On varieties of rational languages and variable length codes, (to appear).
[40] M.O. Rabin and D. Scott, Finite automata and their decision problems, IBM J. Res. and Dev., vol. 3, 1959, 114-125.
[41] M.P. Schützenberger, Sur certaines varietés de monoides finis, Automata Theory (Caianiello, ed.), Academic Press, New York, 1966, 314-319.
[42] M.P. Schützenberger, On finite monoids having only trivial subgroups, Inform. and Control, vol. 8, 1965, 190-194.
[43] M.P. Schützenberger, Sur le produit de concatenation non ambigu, Semigroup Forum, vol. 13, 1976, 47-75.
[44] M.P. Schützenberger, A property of finitely generated submonoids of free monoids, Algebraic Theory of Semigroups (Pollak, ed.), North-Holland, Amsterdam, 1979.
[45] I. Simon, Hierarchies of events with dot-depth one, Ph.D. Thesis, University of Waterloo, 1972.
[46] I. Simon, Piecewise testable events, Proc. 2nd GI Conf., Lecture Notes in Computer Science, vol. 33, Springer-Verlag, 1975, 214-222.
[47] I. Simon, Limited subsets of a free monoid, Proc. 19th Annual Symposium on Foundations of Computer Science, 1978, 143-150.
[48] M. Steinby, On definite automata and related systems, Ann. Acad. Sc. Fennicae, series A.I., No. 444, 1969.
[49] H. Straubing, Varieties of recognizable sets whose syntactic monoids contain solvable groups, Ph.D. Thesis, University of California-Berkeley, 1978.
[50] H. Straubing, Families of recognizable sets corresponding to certain varieties of finite monoids, J. Pure and Applied Algebra, vol. 15, 1979, 305-318.
[51] H. Straubing, Aperiodic homomorphisms and the concatenation produce of recognizable sets, J. Pure and Applied Algebra, vol. 15, 1979, 319-327.
[52] H. Straubing, A generalization of the Schützenberger product of finite monoids, Theoretical Computer Science, (to appear).
[53] H. Straubing, Recognizable sets and power sets of finite semigroups, Semigroup Forum, vol. 18, 1979, 331-340.
[54] D. Thérien, Classification of regular languages by congruences, Ph.D. Thesis, University of Waterloo, 1980.
[55] D. Thérien, Languages of nilpotent and solvable groups, Proc. 6th ICALP, Lecture Notes in Computer Science, no. 71, Springer-Verlag, Berlin, 1979, 616-632.
[56] Y. Zalcstein, Locally testable languages, J. Computer and System Sciences, vol. 6, 1972, 151-167.
[57] Y. Zalcstein, Locally testable semigroups, Semigroup Forum, vol. 5, 1973, 216-227.
[58] Y. Zalcstein, Syntactic semigroups of some classes of star-free languages, Automata, Languages and Programming (m. Nivat, ed.), North-Holland Publishing Company, Amsterdam, 1973, 135-144.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF WATERLOO
TECHNICAL REPORTS 1979

Report No.	Author	Title
CS-79-01	E.A. Ashcroft W.W. Wadge	Generality Considered Harmful - A Critique of Descriptive Semantics
CS-79-02	T.S.E. Maibaum	Abstract Data Types and a Semantics for the ANSI/SPARC Architecture
CS-79-03	D.R. McIntyre	A Maximum Column Partition for Sparse Positive Definite Linear Systems Ordered by the Minimum Degree Ordering Algorithm
CS-79-04	K. Culik II A. Salomaa	Test Sets and Checking Words for Homomorphism Equivalence
CS-79-05*	T.S.E. Maibaum	The Semantics of Sharing in Parallel Processing
CS-79-06	C.J. Colbourn K.S. Booth	Linear Time Automorphism Algorithms for Trees, Interval Graphs, and Planar Graphs
CS-79-07	K. Culik, II N.D. Diamond	A Homomorphic Characterization of Time and Space Complexity Classes of Languages
CS-79-08	M.R. Levy T.S.E. Maibaum	Continuous Data Types
CS-79-09	K.O. Geddes	Non-Truncated Power Series Solution of Linear ODE's in ALTRAN
CS-79-10	D.J. Taylor J.P. Black D.E. Morgan	Robust Implementations of Compound Data Structures
CS-79-11	G.H. Gonnet	Open Addressing Hashing with Unequal-Probability Keys
CS-79-12	M.O. Afolabi	The Design and Implementation of a Package for Symbolic Series Solution of Ordinary Differential Equations
CS-79-13	W.M. Chan J.A. George	A Linear Time Implementation of the Reverse Cuthill-McKee Algorithm
CS-79-14	D.E. Morgan	Analysis of Closed Queueing Networks with Periodic Servers
CS-79-15	M.H. van Emden G.J. de Lucena	Predicate Logic as a Language for Parallel Programming
CS-79-16	J. Karhumäki I. Simon	A Note on Elementary Homorphisms and the Regularity of Equality Sets
CS-79-17	K. Culik II J. Karhumäki	On the Equality Sets for Homomorphisms on Free Monoids with two Generators
CS-79-18	F.E. Fich	Languages of R-Trivial and Related Monoids

* Out of print - contact author

CS-79-19	D.R. Cheriton	Multi-Process Structuring and the Thoth Operating System
CS-79-20	E.A. Ashcroft W.W. Wadge	A Logical Programming Language
CS-79-21	E.A. Ashcroft W.W. Wadge	Structured LUCID
CS-79-22	G.B. Bonkowski W.M. Gentleman M.A. Malcolm	Porting the Zed Compiler
CS-79-23	K.L. Clark M.H. van Emden	Consequence Verification of Flow-charts
CS-79-24	D. Dobkin J.I. Munro	Optimal Time Minimal Space Selection Algorithms
CS-79-25	P.R.F. Cunha C.J. Lucena T.S.E. Maibaum	On the Design and Specification of Message Oriented Programs
CS-79-26	T.S.E. Maibaum	Non-Termination, Implicit Definitions and Abstract Data Types
CS-79-27	D. Dobkin J.I. Munro	Determining the Mode
CS-79-28	T.A. Cargill	A View of Source Text for Diversely Configurable Software
CS-79-29	R.J. Ramirez F.W. Tompa J.I. Munro	Optimum Reorganization Points for Arbitrary Database Costs
CS-79-30	A. Pereda R.L. Carvalho C.J. Lucena T.S.E. Maibaum	Data Specification Methods
CS-79-31	J.I. Munro H. Suwanda	Implicit Data Structures for Fast Search and Update
CS-79-32	D. Rotem J. Urrutia	Circular Permutation Graphs
CS-79-33*	M.S. Brader	PHOTON/532/Set - A Text Formatter
CS-79-34	D.J. Taylor D.E. Morgan J.P. Black	Redundancy in Data Structures: Improving Software Fault Tolerance
CS-79-35	D.J. Taylor D.E. Morgan J.P. Black	Redundancy in Data Structures: Some Theoretical Results
CS-79-36	J.C. Beatty	On the Relationship between the LL(1) and LR(1) Grammars
CS-79-37	E.A. Ashcroft W.W. Wadge	R_x for Semantics

* Out of print - contact author

CS-79-38	E.A. Ashcroft W.W. Wadge	Some Common Misconceptions about LUCID
CS-79-39	J. Albert K. Culik II	Test Sets for Homomorphism Equivalence on Context Free Languages
CS-79-40	F.W. Tompa R.J. Ramirez	Selection of Efficient Storage Structures
CS-79-41*	P.T. Cox T. Pietrzykowski	Deduction Plans: A Basis for Intelligent Backtracking
CS-79-42	R.C. Read D. Rotem J. Urrutia	Orientations of Circle Graphs

* Out of print - contact author

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF WATERLOO
TECHNICAL REPORTS 1980

Report No.	Author	Title
CS-80-01	P.T. Cox T. Pietrzykowski	On Reverse Skolemization
CS-80-02	K. Culik II	Homomorphisms: Decidability, Equality and Test Sets
CS-80-03	J. Brzozowski	Open Problems About Regular Languages
CS-80-04	H. Suwanda	Implicit Data Structures for the Dictionary Problem
CS-80-05	M.H. van Emden	Chess-Endgame Advice: A Case Study in Computer Utilization of Knowledge
CS-80-06	Y. Kobuchi K. Culik II	Simulation Relation of Dynamical Systems
CS-80-07	G.H. Gonnet J.I. Munro H. Suwanda	Exegesis of Self-Organizing Linear Search
CS-80-08	J.P. Black D.J. Taylor D.E. Morgan	An Introduction to Robust Data Structures
CS-80-09	J.Ll. Morris	The Extrapolation of First Order Methods for Parabolic Partial Differential Equations II
CS-80-10*	N. Santoro H. Suwanda	Entropy of the Self-Organizing Linear Lists
CS-80-11	T.S.E. Maibaum C.S. dos Santos A.L. Furtado	A Uniform Logical Treatment of Queries and Updates
CS-80-12	K.R. Apt M.H. van Emden	Contributions to the Theory of Logic Programming
CS-80-13	J.A. George M.T. Heath	Solution of Sparse Linear Least Squares Problems Using Givens Rotations
CS-80-14	T.S.E. Maibaum	Data Base Instances, Abstract Data Types and Data Base Specification
CS-80-15	J.P. Black D.J. Taylor D.E. Morgan	A Robust B-Tree Implementation
CS-80-16	K.O. Geddes	Block Structure in the Chebyshev-Padé Table
CS-80-17	P. Calamai A.R. Conn	A Stable Algorithm for Solving the Multi-facility Location Problem Involving Euclidean Distances

* In preparation

CS-80-18	R.J. Ramirez	Efficient Algorithms for Selecting Efficient Data Storage Structures
CS-80-19	D. Therien	Classification of Regular Languages by Congruences
CS-80-20	J. Buccino	A Reliable Typesetting System for Waterloo
CS-80-21	N. Santoro	Efficient Abstract Implementations for Relational Data Structures
CS-80-22	R.L. de Carvalho T.S.E. Maibaum T.H.C. Pequeno A.A. Pereda P.A.S. Veloso	A Model Theoretic Approach to the Theory of Abstract Data Types and Data Structures
CS-80-23	G.H. Gonnet	A Handbook on Algorithms and Data Structures
CS-80-24	J.P. Black D.J. Taylor D.E. Morgan	A Case Study in Fault Tolerant Software
CS-80-25	N. Santoro	Four O(n**2) Multiplication Methods for Sparse and Dense Boolean Matrices
CS-80-26	J.A. Brzozowski	Development in the Theory of Regular Languages