3. Hopcroft J., Ullman J. D. — *Introduction to Automata Theory, Languages and Computation*, Addison Wesley, Reading Massachussetts, 1979.

4. Knuth D. E. — *Semantics of Context-free Languages*, Math. System Theory 2(1968) 127—145, 5(1971), 95—96.

5. Waite W. M., Goos G. — *Compiler Construction*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

Received 15.XI.1987

Faculty of Mathematics
University of Iaşi
Iaşi, R.S. Romania

ANALELE ŞTIINŢIFICE ALE UNIVERSITĂŢII „AL. I. CUZA" DIN IAŞI
Tomul XXXIII, s. I a. Matematică, 1987, fasc. 4

ON THE MYHILL-NERODE RELATIONS

BY

ALEXANDRU MATEESCU

1. **Definitions and notations.** As usually, Σ^* is the free monoid generated by the alphabet Σ and $P(\Sigma^*)$ is the set of all subsets of Σ^*, i.e. the set of all languages over Σ. The null element of Σ^* is λ. If $L\subseteq\Sigma^*$ then the relation $\sim_L({}_L\sim)$ (cf. [2], [4], [5], [8] etc.) is : $u\sim_L v(u\,{}_L\!\sim v)$ if and only if for every $x\in\Sigma^*$, $ux\in L\Leftrightarrow vx\in L(xu\in L\Leftrightarrow xv\in L)$. The relation $\approx_L$ is defined $u\approx_L v$ if and only if for every x, $y\in\Sigma^*$, $xuy\in L\Leftrightarrow xvy\in L$.

$\sim_L({}_L\sim)$ (see [1], [2], [5], [8]) is the greatest equivalence relations on Σ^*, right (left) invariant, such that L is closed with respect to $\sim_L({}_L\sim)$. Moreover, $\approx_L$ is the greatest congruence on Σ^* such that L is closed with respect to it.

The Myhill-Nerode theorem (1957—1958) says that a language L is regular if and only if $\approx_L$ is of finite index, respectively, if and only if $\sim_L$ $({}_L\sim)$ is of finite index.

Notations. If L, $L_1\subseteq\Sigma^*$ then, *left quotient of* L *by* L_1 is $L_1\backslash L$ and *right quotient of* L by L_1 is L/L_1, where :

$$L_1\backslash L=\{x\mid\exists y\in L_1,\ yx\in L\},\ L/L_1=\{x\mid\exists y\in L_1,\ xy\in L\}.$$

and let $\#$ be such that $\#\notin\Sigma$, the quotient of L by L_1 is :

$$\frac{L}{L_1}=\{x\#y\mid\exists t\in L_1,\ xty\in L\}.$$

Remark. $P(\Sigma^*)$ will be considered as a monoid with respect to the product of languages and the null element $\{\lambda\}$.

2. **Main results. Definition 1.** *If* $L\subseteq\Sigma^*$ *is a fixed language, then we define the relations* $\simeq_L$, ${}_L\simeq$, $\underset{L}{\simeq}$ *in* $P(\Sigma^*)$:

$$\text{for } L_1,\ L_2\subseteq\Sigma^*,\qquad L_1\simeq_L L_2\Leftrightarrow L_1\backslash L=L_2\backslash L,$$

$$\text{for } L_1,\ L_2\subseteq\Sigma^*,\qquad L_1\ {}_L\!\simeq L_2\Leftrightarrow L/L_1=L/L_2,$$

$$\text{for } L_1,\ L_2\subseteq\Sigma^*,\qquad L_1\underset{L}{\simeq}L_2\Leftrightarrow\frac{L}{L_1}=\frac{L}{L_2}.$$

Definition 2. *A language* $L\subseteq\Sigma^*$ *is weak closed with respect to the equivalence relation* $\sim$ *on* $P(\Sigma^*)$ *if and only if for any* L_1, $L_2\subseteq\Sigma^*$

$$\text{if } L_1\cap L\neq 0 \text{ and } L_1\sim L_2 \text{ then } L_2\cap L\neq 0 \text{ is true:}$$

Proposition 3. *For any language $L\subseteq\Sigma^*$ the following hold :*

(i) $\simeq_L$ *is the greatest equivalence relation on $P(\Sigma^*)$, right invariant, such that L is weak closed with respect to it.*

(ii) ${}_L\simeq$ *is the greatest equivalence relation on $P(\Sigma^*)$, left invariant, such that L is weak closed with respect to it.*

(iii) $\underset{L}{\simeq}$ *is the greatest congruence on $P(\Sigma^*)$ such that L is weak closed with respect to it.*

Proof. (i) Obviously $\simeq_L$ is an equivalence relation. This relation is right invariant, because if $A\in P(\Sigma^*)$ and if $L_1\simeq_L L_2$ then $L_1\setminus L=L_2\setminus L$ and therefore $(L_1A)\setminus L=(L_2A)\setminus L$, i.e. $L_1A\simeq_L L_2A$. L is weak closed with respect to $\simeq_L$. For that, let $L_1, L_2\in P(\Sigma^*)$ be such that $L_1\simeq_L L_2$ and $L_1\cap L\neq\varnothing$. Consider $x\in L_1\cap L$ and because $x\in L_1$ and $x\in L$, it results that $\lambda\in L_1\setminus L$ and therefore $\lambda\in L_2\setminus L$. There exists $y\in L_2$, such that $y\lambda\in L$, i.e. $y\in\alpha$. We conclude that $y\in L\cap L_2$ and therefore $L\cap L_2\neq\varnothing$.

We prove now that $\simeq_L$ is the greatest equivalence relation, right invariant, such that L is weak closed with respect to it. Let $\sim$ be another equivalence relation on $P(\Sigma^*)$, right invariant, such that L is weak closed with respect to it and consider $L_1, L_2\in P(\Sigma^*)$ such that $L_1\sim L_2$. We prove that $L_1\simeq_L L_2$, i.e. $\sim\subseteq\simeq_L$.

If L, L_1 and/or $L_i\setminus L$ are empty, $i=1, 2$ then obviously $L_1\simeq_L L_2$.

Assume that L, L_1, L_2, $L_1\setminus L$ and $L_2\setminus L$ are nonempty sets, $L_1\sim L_2$ and let x be in $L_1\setminus L$. It follows that there exists $t\in L_1$ such that $tx\in L$. Because $L_1\sim L_2$ and $\sim$ is a right invariant equivalence relation, it results that $L_1x\sim L_2x$. But $L_1x\cap L\neq\varnothing$ and L is weak closed with respect to $\sim$, so $L_2x\cap L\neq\varnothing$ and therefore $L_1\setminus L\subseteq L_2\setminus L$.

Analogously $L_2\setminus L\subseteq L_1\setminus L$ and, it results that $L_1\simeq_L L_2$; i.e $\sim\subseteq\simeq_L$.

ii) The proof is symmetrical.

iii) The proof is similar with i).

In the next theorem there are found some equivalent properties with respect to the property of a language to be regular.

Theorem 4. *Let L be a language over Σ. The following are equivalent :*

1) *L is regular*

2) *There exists a congruence R of finite index, on $P(\Sigma^*)$, such that L is weak closed with respect to it.*

3) *The congruence $\underset{L}{\simeq}$ is of finite index.*

4) *The equivalence relation $\simeq_L$ is of finite index.*

5) *There exists an equivalence relation $\sim$ on $P(\Sigma^*)$, right invariant, such that L is weak closed with respect to it.*

Proof. 1)$\Rightarrow$2). From L regular, it results that there exists a finite deterministic automaton $\mathcal{A}=(X, \Sigma, \delta, x_0, F)$ such that $L(\mathcal{A})=L$. Consider the finite monoid $M=(P(X)^{P(X)}, \cdot, 1_{P(X)})$ where $f\cdot g=g\circ f$.

Let φ be the function : $\varphi: P(\Sigma^*)\to P(X)^{P(X)}$ defined such that : $\varphi(A)(Y)=\bigcup\limits_{x\in Y}\delta(x, A)$.

It is easy to prove that φ is a homomorphism of monoids and therefore the kernel of φ, $R=\operatorname{Ker}\varphi$, is a congruence of finite index (M is a finite monoid). We prove now that L is weak closed with respect to R. Assume that L_1RL_2 and $L_1\cap L\neq\varnothing$. It results that $\varphi(L_1)=\varphi(L_2)$ and therefore $\varphi(L_1)(\{x_0\})=$

$=\varphi(L_2)(\{x_0\})$, i.e. $\delta(x_0, L_1)=\delta(x_0, L_2)$. But there exists $u\in L_1$, such that $\delta(x_0, u)\in F$. We conclude that there exists $v\in L_2$, such that $\delta(x_0, v)\in F$ and therefore $L_2\cap L\neq\varnothing$.

2) $\Rightarrow$3) results easy from proposition 3, iii).

3) $\Rightarrow$4) From proposition 3, i) it results that $\underset{L}{\simeq}\subseteq\simeq_L$ and therefore 4) is true.

4) $\Rightarrow$5) The relation $\simeq_L$ satisfies the properties of statement 5) (Proposition 3, i)).

5)$\Rightarrow$1) Define the finite deterministic automaton $\mathcal{A}=(P(\Sigma^*)/\sim, \Sigma, \delta, x_0, F)$ where $x_0=\hat{\lambda}$, $F=\{\hat{K}\mid K\subseteq\Sigma^*, K\cap L\neq\varnothing\}$. The transition mapping is : $\delta(\hat{L}_1, \sigma)=\widehat{L_1\delta}$. It is easy to prove that δ is well defined.

We prove now that $L(\mathcal{A})=L$. Let w be in $L(\mathcal{A})$ and therefore $\delta(\hat{\lambda}, w)\in F$, i.e. $\hat{w}\cap F\neq\varnothing$. There exists $K\subseteq\Sigma^*$ such that $K\cap L\neq\varnothing$ and $\{w\}\sim K$. But L is weak closed with respect to $\sim$ and therefore $\{w\}\cap L\neq\varnothing$. It results that $w\in$ and $L(\mathcal{A})\subseteq L$. For the contrary inclusion, let w be in L, $\{w\}\cap L\neq\varnothing$ and therefore $\delta(\hat{\lambda}, w)=\hat{w}$, i.e. $\delta(\hat{\lambda}, w)\in F$. Result that $w\in L(\mathcal{A})$ and $L\subseteq L(\mathcal{A})$. We conclude that $L=L(\mathcal{A})$ and therefore L is regular.

Remark. In theorem 4, the statement 4) may be replaced by „4') ${}_L\simeq$ is of finite index".

Remark. If $\sim$ is an equivalence relation on $P(\Sigma^*)$ then the set $P(\Sigma^*)/\sim$ can be a finite set, a denumerable infinite set or a nondenumerable set. We prove, in next the proposition, that the property of $\underset{L}{\simeq}(\simeq_L, {}_L\simeq)$ to be of denumerable index is *not* a property of the class of context free languages.

Proposition 5. (i) *There exists a context free language, 2-bounded, such that $\simeq_L$ is of nondenumerable index.*

(ii) *There exists a context free language, 2-bounded, such that $\simeq_L$ is of denumerable infinite index.*

Proof. (i) Consider the context free 2-bounded language $L=\{a^nb^n\mid n\geq 1\}$. Let $[A]_L$ be the set $\{B\mid B\subseteq\{a, b\}^*, B\simeq_L A\}$ and define $\varphi: P(a^*)\to P(\{a, b\}^*)/\simeq_L$, $\varphi(A)=[A]_L$. If $A_1, A_2\in P(a^*)$, $A_1\neq A_2$ then there exists a^s such that $a^s\in A_1$ and $a^s\notin A_2$. Therefore $A_1\setminus L\neq A_2\setminus L$, because $b^s\in A_1\setminus L$ and $b^s\notin A_2\setminus w$. It results that $[A_1]_L\neq[A_2]_L$ and we conclude that φ is an injective mapping.

But $P(a^*)$ is a nondenumerable set and therefore $P(\{a, b\}^*)/\simeq_L$ is a nondenumerable set.

(ii) An easy ; proof without theoretical difficulties shows that $\simeq_L$ is of denumerable infinite index.

Final remarks. The question which is the class of languages L with $\underset{L}{\simeq}(\simeq_L, {}_L\simeq)$ of nondenumerable index or of denumerable infinite index, remains open.

REFERENCES

1. Căzănescu V.E. — *Introducere in teoria limbajelor formale*, Ed. Academiei R.S.R. Bucureşti, 1983.

2. Ginsburg S. — *The Mathematical Theory of Context Free Languages*, Mc Graw Hill, New York, 1966.

3. Ginzburg A. — *Algebraic Theory of Automata*. Academic Press, New York, 1968.
4. Hopcroft J. E., Ullman J. D. — *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reeding Mass. 1979.
5. Livovschi L., Popovici C., Georgescu H., Țăndăreanu N. — *Bazele Informaticii*, Ed. Did. și Ped., 1981.
6. Marcus S. — *Gramatici și automate finite*. Ed. Academiei, București, 1964.
7. Mateescu A. — *O generalizare a teoremei Myhill-Nerode*, Lucrările Sesiunii Științifice a CCUB, București, 20—21 Februarie 1987.
8. Păun G. — *Probleme actuale în teoria limbajelor formale*. Ed. Șt. și Enc., București, 1984.
9. Rudeanu S. — *Curs de bazele informaticii*. Fac. de Matematică, Univ. din Buc., 1979—1980 (nepublicat).

Received 15.XI.1987

University of Bucharest
Faculty of Mathematics
Str. Academiei 14
70109 Bucuresti
România

ANALELE ȘTIINȚIFICE ALE UNIVERSITĂȚII „AL. I. CUZA" DIN IAȘI
Tomul XXXIII, s. I a. Matematică, 1987, fasc. 4

AN ALGEBRAIC APPROACH TO CONCURRENCY BY MUTUAL EXCLUSION

BY

GABRIEL CIOBANU

The paper represents a first attempt to introduce an algebraic formalism for expressing concurrent computation by means of the mechanism of mutual exclusion. It proposes the structure of lattice-ordered semigroups as a domain of mathematical (denotational) semantics of programming languages with concurrent computation facilities.

The basic notions are introduced; then the usefulness of algebraic lattices in denotational semantics of programming languages for the representation of nondeterminism is proved. A binary concurrent operation is then defined on an algebraic lattice and we assume that the structure of a lattice-ordered semigroup is obtained. The introduction of such an operation is justified by the established properties and rules. The paper is concerned with these properties; it emphasizes the expressive power of lattice-ordered semigroups in the approach to concurrent properties. The results prove that the proposed structure can be useful, offering a model for actual practice and is leading to a good understanding of the complex instances that are harder to master by people directly involved in experimental work. The results prove also that the monitors constitute the most elaborate real mechanism of mutual exclusion.

1. **Introduction.** Denotational semantics of programming languages has been given strong theoretical foundations by the work of D. Scott. The basic idea was to represent the data types as partially ordered sets called domains, and computations on such data as continuous functions on domains. Initially, Scott considered the continuous lattices as domains, but afterwards it was proved that the algebraic lattices could be considered as well [11]. The domains must be closed under sum $+$, product $\times$, the function space construct $\rightarrow$ etc. to be able to describe the usual programming constructs and it also has to allow for solutions of domain equations $D \simeq E(D)$, where E is an expression built up from $+$, $\times$, $\rightarrow$ etc.

Let $(D, \leq)$ be a partially ordered set (poset), $D \neq \emptyset$. $a \in D$ is a lower (upper) bound of $X \subseteq D$ and write $a \leq X (X \leq a)$ if $a \leq x (x \leq a)$ for every $x \in X$. Let $\underline{X}$ be the set of the lower bounds of X and X the set of the upper bounds of X. For any $X \subseteq D$, the least upper bound of X, when it exists, is denoted by $\bigvee X$ or sup X, and the greatest lower bound of X is denoted by $\bigwedge X$ or inf X(if exists). Particularly, $0 = \sup \emptyset$ is the least element of D and $I = \inf \emptyset$ is the greatest element of D, if they exist. If we have a partially