Volume 11 Number 3

ACTA CYBERNETICA

Szeged, 1994

Information for authors: Acta Cybernetica publishes only original papers in English in the field of computer science. Review papers are accepted only exceptionally. Manuscripts should be sent in triplicate to one of the Editors. The manuscripts must be typed douple-spaced on one side of the paper only. For the form of references, see one of the articles previously published in the journal.

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

Mealy-automata in which the output-equivalence is a congruence*

I. Babcsányi† A. Nagy†

Dedicated to Professor A. ÁDÁM on his 60th birthday

Abstract

Every Mealy-automaton whose output equivalence is not the universal relation has a non-trivial simple state-homomorphic image. Thus the simple Mealy-automata play an importante role in the theory of Mealy-automata. It is very difficult to describe the structure of these automata. Contrary to the earlier investigations, in our present paper we concentrate our attention only to a special kind of simplicity, namely the strongly simplicity. Besides we give a construction for strongly simple Mealy-automata, we also describe the structure of all Mealy-automata which have strongly simple state-homomorphic image.

1 Preliminaries

By a *Mealy-automaton* we mean a system $\underline{A} = (A, X, Y, \delta, \lambda)$ consisting of a state set A, an input set X, an output set Y, a transition function $\delta : A \times X \to A$ and an output function $\lambda : A \times X \to Y$. In that case when $|A|, |X|, |Y|$ are finite, $\underline{A} = (A, X, Y, \delta, \lambda)$ is called finite ($|S|$ denotes the cardinality of a set S). A Mealy-automaton $\underline{A}$ is called a *Moore-automaton* if

$$\delta(a_1, x_1) = \delta(a_2, x_2) \Longrightarrow \lambda(a_1, x_1) = \lambda(a_2, x_2)$$

for all $a_1, a_2 \in A$ and $x_1, x_2 \in X$. It means that the function λ can be given in the form

$$\lambda(a, x) = \mu(\delta(a, x)) \quad (a \in A, x \in X),$$

where $\mu : A \to Y$ is a single-valued mapping. The function μ is said to be the *sign function* of $\underline{A}$.

*Research supported by project 11281 of the Academy of Finland, the Basic Research ASMICS II Working Group, and, in the case of the second author, also by the Alexander von Humboldt Foundation.

†Department of Mathematics, Transport Engineering Faculty, Technical University of Budapest, H-1111 Budapest, Műegyetem rkp. 9., Hungary

Let X^* and X^+ denote the free monoid and the free semigroup over a nonempty set X, respectively. We extend the functions δ and λ of $\underline{A}$ in the usual forms $\delta : A \times X^* \longrightarrow A^*$ and $\lambda : A \times X^* \longrightarrow Y^*$ as follows :

$$\delta(a,e) = a, \quad \delta(a,px) = \delta(a,p)\delta(ap,x),$$

$$\lambda(a,e) = e, \quad \lambda(a,px) = \lambda(a,p)\lambda(ap,x),$$

where $a \in A$, $p \in X^+$, $x \in X$, ap denotes the last letter of $\delta(a,p)$ and e denotes the empty word.

An equivalence relation τ of a state set A of a Mealy-automaton $\underline{A} = (A, X, Y, \delta, \lambda)$ is called a *congruence* on $\underline{A}$ if

$$(a,b) \in \tau \Longrightarrow (ap, bp) \in \tau \quad \text{and} \quad \overline{\lambda(a,p)} = \overline{\lambda(b,p)}$$

for all $a, b \in A$ and $p \in X^+$. (If $r \in Y^+$ then $\overline{r}$ denotes the last letter of r.)

Let ρ_{max} denote the relation on the state set A of a Mealy-automaton $\underline{A} = (A, X, Y, \delta, \lambda)$ defined by

$$(a,b) \in \rho_{max} \Longleftrightarrow \overline{\lambda(a,p)} = \overline{\lambda(b,p)} \quad \text{for all} \quad p \in X^+ \quad ([2]).$$

The ρ_{max}-class of $\underline{A}$ containing the state a of $\underline{A}$ is denoted by $\rho_{max}[a]$.

Denoting the identity relation of a Mealy-automaton $\underline{A}$ by ι, we say that $\underline{A}$ is *simple* if $\rho_{max} = \iota$.

It is easy to see that ρ_{max} is the greatest congruence of $\underline{A}$ and $\underline{A}/\rho_{max}$ is simple.

Let $\underline{A} = (A, X, Y, \delta, \lambda)$ and $\underline{A}' = (A', X, Y, \delta', \lambda')$ be arbitrary Mealy-automata. We say that a mapping $\alpha : A \longrightarrow A'$ is a *state-homomorphism* of $\underline{A}$ into $\underline{A}'$ if

$$\alpha(\delta(a,x)) = \delta'(\alpha(a), x), \quad \lambda(a,x) = \lambda'(\alpha(a), x)$$

for all $a \in A$ and $x \in X$. If α is surjective then $\underline{A}'$ is called a *state-homomorphic image* of $\underline{A}$. If α is bijective then α is called a *state-isomorphism* and the automata $\underline{A}$ and $\underline{A}'$ are said to be *state-isomorphic*.

Let $\underline{A} = (A, X, Y, \delta, \lambda)$ be a Mealy-automaton. By the *output-equivalence* of $\underline{A}$ we mean the equivalence ρ defined as

$$\rho = \{(a,b) \in A \times A : \quad (\forall x \in X) \quad \lambda(a,x) = \lambda(b,x)\} \quad ([3]).$$

It is evident that $\rho_{max} \subseteq \rho$. Moreover ρ is a congruence if and only if $\rho = \rho_{max}$. If ρ is the universal relation of A then, for every $a, b \in A$, $q \in X^*$ and $x \in X$,

$$\overline{\lambda(a,qx)} = \lambda(aq,x) = \lambda(bq,x) = \overline{\lambda(b,qx)}.$$

From this it follows that if ρ is the universal relation of A then $\rho = \rho_{max}$.

For notations and notions not defined here, we refer to [4] and [5].

2 Strongly simple Mealy-automata

Definition A Mealy-automaton will be called a *strongly simple* Mealy-automaton if $\rho = \iota$.

The next construction plays an importante role throughout this paper.

Construction 1 *Let $\underline{A} = (A, X, Y, \delta, \lambda)$ be a Mealy-automaton. To arbitrary states a of $\underline{A}$, we can associate mappings α_a of X into Y defined as follows:*

$$\alpha_a : x \rightarrow \lambda(a, x).$$

Consider the set $\mathcal{A} = \{\alpha_a; a \in A\}$ and, for every $a \in A$ and $x \in X$, let

$$\delta'(\alpha_a, x) = \alpha_{\delta(a,x)}, \quad \lambda'(\alpha_a, x) = \alpha_a(x).$$

Theorem 1 *For an arbitrary Mealy-automaton $\underline{A} = (A, X, Y, \delta, \lambda)$, the following four conditions are equivalent:*

(i) The quintuple $\underline{\mathcal{A}} = (\mathcal{A}, X, Y, \delta', \lambda')$, where $\mathcal{A}$, δ', λ' are defined as in Construction 1, is a Mealy-automaton;

(ii) $\rho = \rho_{max}$ in $\underline{A}$;

(iii) $\underline{\mathcal{A}}$ and $\underline{A}/\rho_{max}$ are state-isomorphic;

(iv) $\underline{A}/\rho_{max}$ is strongly simple.

Proof. Assume that $\underline{\mathcal{A}}$ is a Mealy-automaton. Then $\alpha_a = \alpha_b$ implies $\alpha_{\delta(a,x)} = \alpha_{\delta(b,x)}$ for every $a, b \in A$ and $x \in X$, because δ' is well-defined. We show that $\rho = \rho_{max}$ in $\underline{A}$. Consider two arbitrary elements a and b of A with $(a, b) \in \rho$. Then $\alpha_a = \alpha_b$ and so we get $\alpha_{\delta(a,x)} = \alpha_{\delta(b,x)}$ for every $x \in X$. Using this idea and the fact that δ is extended to $A \times X^*$, we get $\alpha_{ap} = \alpha_{bp}$ for every $p \in X^*$. Thus

$$\overline{\lambda(a, px)} = \lambda(ap, x) = \lambda(bp, x) = \overline{\lambda(b, px)}$$

for every $p \in X^*$ and $x \in X$. Consequently $(a, b) \in \rho_{max}$ which implies that $\rho = \rho_{max}$ in $\underline{A}$. Thus (i) implies (ii).

Assume that $\rho = \rho_{max}$ in a Mealy-automaton $\underline{A}$. To show that $\underline{\mathcal{A}}$ is a Mealy-automaton, it is sufficient to prove that δ' is well-defined. Let a and b be arbitrary elements of A with $\alpha_a = \alpha_b$. Then $(a, b) \in \rho = \rho_{max}$ from which we get $(\delta(a, x), \delta(b, x)) \in \rho = \rho_{max}$ for every $x \in X$. Thus $\alpha_{\delta(a,x)} = \alpha_{\delta(b,x)}$ $(x \in X)$ and so δ' is well-defined. Consequently, (ii) implies (i).

To show that (ii) implies (iii), assume $\rho = \rho_{max}$ in $\underline{A}$. Then $\alpha_a = \alpha_b$ if and only if $(a, b) \in \rho_{max}$ which implies that $\alpha_a \rightarrow \rho_{max}[a]$, $a \in A$ is a state-isomorphism of $\underline{\mathcal{A}}$ onto $\underline{A}/\rho_{max}$. Consequently, (iii) is satisfied.

Assume (iii). Then $\underline{\mathcal{A}}$ is a Mealy-automaton. Thus λ' is well-defined. From this it follows that $\underline{\mathcal{A}}$ and so $\underline{A}/\rho_{max}$ is strongly simple. Therefore, (iv) is true.

Condition (ii) follows from (iv) in a trivial way. □

Construction 2 *Let M be a non-empty subset of the set Y^X of all mappings of X into Y, where X and Y are arbitrary non-empty sets. Consider the Mealy-automaton $\underline{M} = (M, X, Y, \delta^*, \lambda^*)$, where δ^* is arbitrary and λ^* is defined as follows:*

$$\lambda^*(\alpha, x) = \alpha(x), \quad \alpha \in M, \ x \in X.$$

For non-empty sets X and Y, denote $\mathcal{M}[X,Y]$ the set of all Mealy-automata defined in Construction 2. It is evident that $\underline{\mathcal{A}} \in \mathcal{M}[X,Y]$ supposing that $\rho = \rho_{max}$ in the Mealy-automaton $\underline{A} = (A, X, Y, \delta, \lambda)$.

Theorem 2 *A Mealy-automaton is strongly simple if and only if it is state-isomorphic to a Mealy-automaton $\underline{M} = (M, X, Y, \delta^*, \lambda^*)$ defined in Construction 2 for some X, Y, δ^* and λ^*.*

Proof. It is trivial that Mealy-automata defined in Construction 2 are strongly simple.

Conversely, let $\underline{A} = (A, X, Y, \delta, \lambda)$ be an arbitrary strongly simple Mealy-automaton. For this Mealy-automaton consider $\underline{\mathcal{A}} = (\mathcal{A}, X, Y, \delta', \lambda')$ with $\mathcal{A}$, δ' λ' defined in Construction 1. By Theorem 1, $\underline{A}$ is isomorphic to $\underline{\mathcal{A}} \in \mathcal{M}[X,Y]$. □

Lemma 1 $\underline{M}_1$, $\underline{M}_2$ $\in$ $\mathcal{M}[X,Y]$ *are state-isomorphic if and only if* $\underline{M}_1 = \underline{M}_2$.

Proof. Assume that $\underline{M}_1$, $\underline{M}_2 \in \mathcal{M}[X,Y]$ are state-isomorphic. Let φ be a state-isomorphism of $\underline{M}_1$ onto $\underline{M}_2$. Then, for every $\alpha \in M_1$ and $x \in X$,

$$\alpha(x) = \lambda_1^*(\alpha, x) = \lambda_2^*(\varphi(\alpha), x) = \varphi(\alpha)(x)$$

and

$$\varphi(\delta_1^*(\alpha, x)) = \delta_2^*(\varphi(\alpha), x).$$

From the first expression we get that φ is identical and so $M_1 = M_2$, $\lambda_1^* = \lambda_2^*$. Then the second expression implies $\delta_1^* = \delta_2^*$. Consequently, $\underline{M}_1 = \underline{M}_2$. □

Corollary 1 *If X and Y are finite non-empty sets then*

$$|\mathcal{M}[X,Y]| = \sum_{k=1}^{|Y|^{|X|}} \binom{|Y|^{|X|}}{k} k^{k|X|}.$$

Proof. Let X and Y be arbitrary finite non-empty sets. Then $|Y^X| = |Y|^{|X|}$. Let $M \subseteq Y^X$ be arbitrary with $|M| = k$. By the Lemma, the number of all different Mealy-automata defined in Construction 2 with the state set M is $k^{k|X|}$, because we can choose $\delta^* : M \times X \to M$ in $k^{k|X|}$ different way. This implies our assertion. □

It is known that every Mealy-automaton is equivalent ([4]) to some Moore-automaton. Therefore, it is interesting for us to know how we can construct the strongly simple Moore-automata. We note that a Mealy-automaton $\underline{M}$ defined in Construction 2 is a Moore-automaton if and only if we choose δ^* such that

$$\alpha_1(x_1) \neq \alpha_2(x_2) \Longrightarrow \delta^*(\alpha_1, x_1) \neq \delta^*(\alpha_2, x_2)$$

for every $\alpha_1, \alpha_2 \in M$ and $x_1, x_2 \in X$. Moreover, the output function λ^* of $\underline{M}$ does not depend on the input signs if and only if all mappings $\alpha \in M$ are constant. In this case $\underline{M}$ can be considered as a special Moore-automaton ([1]) with the sign function λ^* and the output function λ defined by $\lambda(\alpha, x) = \lambda^*(\delta^*(\alpha, x))$. Thus the number of these special Moore-automata belonging to $\mathcal{M}[X,Y]$ is

$$\sum_{k=1}^{|Y|} \binom{|Y|}{k} k^{k|X|},$$

supposing that X and Y are finite.

Introduce a partially ordering "$\leq$" on $\mathcal{M}[X,Y]$ as follows: $\underline{M}_1 \leq \underline{M}_2$ if and only if $M_1 \subseteq M_2$ and δ_1 equals the restriction of δ_2 to $M_1 \times X$. Under this ordering an element of $\mathcal{M}[X,Y]$ is maximal if and only if its state set is Y^X. If X and Y are finite then the number of maximal elements of $\mathcal{M}[X,Y]$ is

$$|Y|^{(|Y|^{|X|})|X|^2}.$$

It can be easily verified that the number of maximal elements of $\mathcal{M}[X,Y]$ which are special Moore-automata (see above) is $|Y|^{|Y||X|}$.

3 Mealy-automata having a strongly simple state-homomorphic image

In this chapter we give a construction for Mealy-automata which has the property $\rho = \rho_{max}$.

Construction 3 *Let $\underline{M} = (M, X, Y, \delta^*, \lambda^*)$ be a strongly simple Mealy-automaton (defined in Construction 2). Consider a family of sets B_m, $m \in M$ such that $B_m \cap B_{m'} = \emptyset$ if $m \neq m'$. For all $x \in X$ and $m \in M$, let $\varphi_{m,x}$ be a mapping of B_m into $B_{\delta^*(m,x)}$. Let $B = \cup_{m \in M} B_m$. Define the functions $\delta^\circ : B \times X \to B$ and $\lambda^\circ : B \times X \to Y$ as follows. For arbitrary $b \in B_m$, let*

$$\delta^\circ(b,x) = \varphi_{m,x}(b) \quad \textit{and} \quad \lambda^\circ(b,x) = m(x).$$

It can be easily verified that δ° and λ° are well-defined and so $\underline{B} = (B, X, Y, \delta^\circ, \lambda^\circ)$ is a Mealy-automaton.

Theorem 3 *A Mealy-automaton has the property that $\rho = \rho_{max}$ if and only if it can be defined as in Construction 3.*

Proof. Let $\underline{B}$ be a Mealy-automaton defined in Construction 3. We prove that $\rho = \rho_{max}$. For all $m \in M$, $p \in X^*$ and $x \in X$ let $\varphi_{m,px} = \varphi_{mp,x} \circ \varphi_{m,p}$, where mp denotes the last letter of $\delta^*(m,p)$. It is clear that $\varphi_{m,p}(a) = ap$ for all $a \in B_m$ and $p \in X^*$, where ap denotes the last letter of $\delta^\circ(a,p)$. Assume $(a,b) \in \rho$ for some $a, b \in B$. Then $a, b \in B_m$ for some $m \in M$. For arbitrary $p \in X^*$ and $x \in X$,

$$\overline{\lambda^\circ(a,px)} = \lambda^\circ(ap,x) = \lambda^\circ(\varphi_{m,p}(a),x) = \lambda^\circ(\varphi_{m,p}(b),x) = \lambda^\circ(bp,x) = \overline{\lambda^\circ(b,px)}.$$

From this it follows that $(a,b) \in \rho_{max}$.

Conversely, assume that $\rho = \rho_{max}$ in a Mealy-automaton $\underline{A} = (A, X, Y, \delta, \lambda)$. By Theorem 1, $\underline{\mathcal{A}} = (\mathcal{A}, X, Y, \delta', \lambda')$ is a Mealy-automaton which is state-isomorphic with the strongly simple Mealy-automaton $\underline{A}/\rho_{max}$. Using Construction 3 for $\underline{M} = \underline{\mathcal{A}}$, consider the Mealy-automaton $\underline{B} = (B, X, Y, \delta^\circ, \lambda^\circ)$ such that $B_{\alpha_a} = \rho_{max}[a]$ and $\varphi_{\alpha_a,x}$ defined by $\varphi_{\alpha_a,x}(b) = \delta(b,x)$ for arbitrary $a \in A$, $b \in B_{\alpha_a}$, $x \in X$. It is easy to see that $A = B$, $\delta = \delta^\circ$ and $\lambda = \lambda^\circ$. Thus $\underline{A} = \underline{B}$. □

Remark. If the output equivalence ρ of a Mealy-automaton $\underline{A}$ is the universal relation of A then $\underline{A}$ is simple if and only if it is strongly simple if and only if it is trivial (it has only one state). Thus our problems are trivial in this case. We note that if $\underline{A} = (A, X, Y, \delta, \lambda)$ is a Mealy-automaton in which ρ is the universal relation then the congruences of $\underline{A}$ are the same as the congruences of the projection $\underline{A}_{pr} = (A, X, \delta)$ of $\underline{A}$. But the simplicity of automata without outputs is modified as follows: An automaton $\underline{B}$ without outputs is called simple if its every state-homomorphic image is trivial or isomorphic to $\underline{B}$. It is easy to see that this simplicity is different from the strongly simplicity. (Here the strongly simplicity means that the automaton is trivial.)

References

[1] Ádám, A., *On the question of description of the behaviour of finite automata*, Studia Sci. Math. Hungar., 13 (1978), 105-124.

[2] Babcsányi, I., *On output behaviour of Mealy-automata*, Periodica Polytechnica (Transportation Engineering), 19(1991), No 1-2, 15-21.

[3] Babcsányi, I., A. Nagy and F. Wettl, *Indistinguishable state pairs in strongly connected Moore-automata*, PU.M.A. Ser. A, 2 (1991), 15-24.

[4] Gécseg, F. and I. Peák, *Algebraic Theory of Automata*, Akadémiai Kiadó, Budapest, 1972.

[5] Szász, G., *Introduction to Lattice Theory*, Academic Press, New York- London, 1963.

Received December 21, 1993

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

Measure of Infinitary Codes

Nguyen Huong Lam * Do Long Van *

Abstract

An attempt to define a measure on the set A^N of infinite words over an alphabet A starting from any Bernoulli distribution on A is proposed. With respect to this measure, any recognizable (in the sense of Büchi-McNaughton) language is measurable and the Kraft-McMillan inequality holds for measurable infinitary codes. Nevertheless, we face some "anomalies" in contrast with ordinary codes.

1 Introduction

In this paper we need only very basic concepts and facts from the formal language theory and the theory of codes, for which we always refer to [Ei] and [Be-Pe]. Let A be a finite or countable alphabet and A^* be the set of (finite) words on A (that is A^* is the free monoid with base A) with the empty word (the unit of A^*) denoted by ϵ. The set of nonempty words is denoted by $A^+ = A^* - \epsilon$. The product of two words u and v is the concatenation uv of them.

A *factorization* of a word w on a given subset X of A^* is a sequence $u_1, \ldots, u_n$ of words of X such that $w = u_1 \ldots u_n$. A subset X of A^* is a *code* if every word of A^* has at most one factorization on X.

Intuitively, a code may not contain too many words and this idea has been stated mathematically in the remarkable Kraft-McMillan inequality. Let us mention it now.

A *Bernoulli distribution* on A is a function

$$p : A \rightarrow R_+$$

associating with each *letter* a nonnegative real number such that

$$\sum_{a \in A} p(a) = 1.$$

A distribution p is *positive* if $p(a) > 0$ for all $a \in A$. We extend p in a natural way to a word $u = a_1 \ldots a_n$ of A^* ($a_1, \ldots, a_n$ are letters) by

$$p(u) = \prod_{i=1}^{n} p(a_i)$$

*Institute of Mathematics, P. O. Box 631, 10 000 Hanoi, Vietnam.

and then to a subset X of A^* by

$$p(X) = \sum_{u \in X} p(u).$$

The value $p(X)$ is called the *measure* of X, which may be finite or infinite. If finite, the measure is the sum of an absolutely convergent numerical series, so the order of summation is not important and the definition is correct.

The well-known in the information theory Kraft-McMillan inequality ([Mc] or [Be-Pe]) says that:

> *For any Bernoulli distribution, the measure of any code does not exceed 1.*

The presentation that follows is an attempt to resolve a question, quite natural, in the mainstream of extensive studies on infinite words: how can one define a measure (in some sense) on the set of infinite words A^N so that this measure should be well compatible with the measure structure and properties of languages in A^*? Besides, we want this measure to satisfy our own demand: to prove something like the Kraft-McMillan inequality for infinitary codes, introduced in [Va]. To do this we come to the theory of measure, making use of its very basic concepts (Lebesgue extension of measures, infinite product of probability spaces) and we also exploit some techniques suggested by [Sm].

2 Measure Theory

2.1 Basic

We give a brief survey of facts for furthergoing treatment. For more details the reader is referred to [Ha]. Let X be any fixed set; we always deal with subsets of X, so in the sequel sets always mean subsets of this "base" set. Also we use the Euler fraktur alphabet to indicate classes (collections) of sets, for example, $\mathfrak{P}(X)$ is the class of all subsets of X (the power set). A class $\mathfrak{R}$ is called a (Boolean) *ring* of sets provided for any $E, F \in \mathfrak{R}$ the set-theoretic difference $E - F$ and union $E \cup F$ are also in $\mathfrak{R}$. A ring is called *σ-ring* if $\mathfrak{R}$ is closed under the formation of countable unions, i.e., $\cup_{i=1}^{\infty} E_i$ is in $\mathfrak{R}$ for any countable sequence of sets $E_1, E_2, \ldots$ of $\mathfrak{R}$. A ring (σ-ring) containing the base set X, is said to be an *algebra* (a *σ-algebra* resp.). Since $E \cap F = E \cup F - ((E - F) \cup (F - E))$ and $\cap_{i=1}^{\infty} E_i = X - \cup_{i=1}^{\infty}(X - E_i)$, we see that a ring is also closed under the formation of finite, and moreover if it is a σ-algebra, of countable intersections. Since the intersection of any number of rings (σ-rings) is also a ring (σ-ring), for any class $\mathfrak{E}$ there exists the smallest ring (σ-ring) containing it, which is called the ring (σ-ring) *generated* by $\mathfrak{E}$ and denoted by $R(\mathfrak{E})$ ($S(\mathfrak{E})$ resp.). We say that $\mathfrak{E}$ is a *hereditary class* if for every $E \in \mathfrak{E}$, $F \subseteq E$ implies $F \in \mathfrak{E}$. Clearly, the hereditarity of classes is preserved under any intersection therefore we can say of the smallest hereditary class $H(\mathfrak{E})$ containing a given class $\mathfrak{E}$.

Let $\mathfrak{E}$ be any class of sets. A *set function* on $\mathfrak{E}$ is a mapping

$$f : \mathfrak{E} \to R_+ \cup \infty$$

defined on $\mathfrak{E}$, taking real nonnegative values including infinity. A set function f is called

— *additive*, if for any disjoint sets E_1, E_2 of $\mathfrak{C}$ such that $E_1 \cup E_2 \in \mathfrak{C}$

$$f(E_1 \cup E_2) = f(E_1) + f(E_2);$$

— *countably additive*, or *σ-additive*, if for any countable sequence of mutually disjioint sets $E_1, E_2, \ldots$ of $\mathfrak{C}$ such that $\cup_{i=1}^{\infty} E_i \in \mathfrak{C}$

$$f(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} f(E_i).$$

A σ-additive set function μ on a ring $\mathfrak{R}$ is said to be a *measure* (on $\mathfrak{R}$). The value $\mu(E)$ is the *measure* of E. A measure μ is *finite* if every E of $\mathfrak{R}$ has finite measure and is *σ-finite* if every E of $\mathfrak{R}$ is a countable union of sets of $\mathfrak{R}$, all of them having finite measure.

2.2 Lebesgue Extension of Measures

Let μ_1, μ_2 be measures respectively on the rings $\mathfrak{R}_1$ and $\mathfrak{R}_2$ with $\mathfrak{R}_1 \subseteq \mathfrak{R}_2$, then μ_2 is an *extension* of μ_1 if restricted to $\mathfrak{R}_1$, μ_2 is equal to μ_1.

Provided the σ-additivity of the measure μ on some ring $\mathfrak{R}$, we can extend it considerably further to a σ-ring which is in some sense maximal as follows.

Let $H(\mathfrak{R})$ be the smallest hereditary σ-ring containing $\mathfrak{R}$. For any set $E \in H(\mathfrak{R})$, we define the *outer measure* of E

$$\mu^*(E) = \inf\{\sum_{i=1}^{\infty} \mu(E_i) : E \subseteq \bigcup_{i=1}^{\infty} E_i, E_i \in \mathfrak{R} \}.$$

Indeed, $\mu^*(E) = \mu(E)$ for $E \in \mathfrak{R}$. Following [Ko-Fo], a set $E \in H(\mathfrak{R})$ is called *measurable* if for any $\epsilon > 0$ there exist $E_0 \in \mathfrak{R}$ such that

$$\mu^*(E \triangle E_0) < \epsilon,$$

where $E \triangle E_0 = (E - E_0) \cup (E - E_0)$ is the symmetric difference of E and F.

It is proved that the class $\mathfrak{M}$ of all measurable sets is a σ-ring and the function μ^* is σ-additive on it and $S(\mathfrak{R}) \subseteq \mathfrak{M}$ [Ko-Fo].

Thus the measure μ on $\mathfrak{R}$ has been extended to the measure μ^* on the σ-ring $S(\mathfrak{R})$ generated by $\mathfrak{R}$ and certainly $\mu^*(E) = \mu(E)$ when $E \in \mathfrak{R}$. Usually, the triple $(X, \mathfrak{M}, \mu)$ consisting of the base set X, a σ-ring $\mathfrak{M}$ of subsets of X and a measure μ on $\mathfrak{M}$ is called a *measure space*; when $X \in \mathfrak{M}$ and $\mu(X) = 1$ the measure space is called a *probability space*.

We now make a remark that will be useful in the sequel. Sometimes, the starting point is not the ring $\mathfrak{R}$ itself, but some subclass $\mathfrak{S}$ such that it can generates $\mathfrak{R}$ and the latter is easily constructed from $\mathfrak{S}$. An example of such classes are semirings, considered in [Ko-Fo]: a class $\mathfrak{S}$ is a *semiring* provided, first, it is closed under the formation of finite intersections and, second, if $E, F \in \mathfrak{S}, E \subseteq F$ then F splits into a finite number of mutually disjoint subsets $E_0, E_1, \ldots, E_n$ of $\mathfrak{S}$ such that $E = E_0$: $F = \bigcup_{i=0}^{n} E_i$. If $\mathfrak{S}$ is a semiring, $R(\mathfrak{S})$ is then the class of all finite unions of subsets of $\mathfrak{S}$. It is easy to see also that if μ is σ-additive on $\mathfrak{S}$, so is in $R(\mathfrak{S})$.

2.3 Infinite Product Measure

Another fundamental construction we need here is the infinite product measure. More specifically, we treate only the countable product.

Let $(X_i, \mathfrak{M}_i, \mu_i), i = 1, 2, \ldots$ be a countable collection of probability spaces, i.e. measure spaces with $X_i \in \mathfrak{M}_i$ and $\mu_i(X_i) = 1$. Further, let $X = \prod_{i=1}^{\infty} X_i$ be the set-theoretic Cartesian product of the sets $X_1, X_2, \ldots$. A subset A of X of the form

$$A = \prod_{i=1}^{\infty} A_i, \qquad A_i \in \mathfrak{M}_i$$

and $A_i = X_i$ for almost all i, is called a *measurable rectangle*. The class of measurable rectangles is obviously a semiring and is denoted by $\mathfrak{A}$. Let us denote $\mathfrak{M} = S(\mathfrak{A})$ the σ-ring generated by the measurable rectangles. Theorem 2 of [Ha, Chapter VII, §38] states, in fact, that there exists uniquely a measure μ on $\mathfrak{M}$ such that if

$$A = A_1 \times \ldots \times A_n \times X_{n+1} \times X_{n+2} \times \ldots$$

is a measurable rectangle then

$$\mu(A) = \mu_1(A_1) \ldots \mu_n(A_n).$$

Since $\mu_i(X_i) = 1$ for all i, μ is well-defined on $\mathfrak{A}$ and $\mu(X) = 1$. Therefore, the triple $(X, \mathfrak{M}, \mu)$ is a probability space that is called the *product measure space* of spaces $(X_i, \mathfrak{M}_i, \mu_i)$ and the measure μ on $\mathfrak{M}$ is then called the *product measure* of measures μ_i.

This construction ensures the existence of a measure on the set of infinite words, which we shall consider in the next section.

3 Measure on A^N

An infinite word α on the alphabet A is an infinite sequense of letters indexed by natural numbers

$$\alpha = a_1 a_2 \ldots.$$

The set of all infinite words on A is denoted by A^N. We consider also the set $A^\infty = A^* \cup A^N$, on which we define the monoid structure as follows [Va]: for $\alpha, \beta \in A^\infty$, if $\alpha \in A^*$ then the product $\alpha \cdot \beta$ is the concatenation $\alpha\beta$ of α and β; otherwise, if $\alpha \in A^N, \alpha \cdot \beta$ is defined to be α. Naturally, the product of words can be extended for *languages*, i.e. subsets of A^∞: $XY = \{\alpha \cdot \beta | \alpha \in X \subseteq A^\infty, \beta \in Y \subseteq A^\infty\}$. Not to be too strict, in the folowing, we omit the dot in the product of words and when a set is a singleton we frequently identify it with its element.

Let now p be any Bernoulli distribution on A, as before extended to A^*; then $(A, \mathfrak{P}(A), p)$ actually forms a probability space, where $\mathfrak{P}(A)$ is the set of all subsets of A. Next, we can view A^N as the Cartesian product of ω (the cardinality of N) copies of A

$$A^N = \prod_{i \in N} A$$

and we can say of the class $\mathfrak{A}$ of measurable rectangles R

$$R = \prod_{i=1}^{\infty} A_i, \quad A_i \in \mathfrak{M}_i$$

with $A_i = A$ for almost all i, which is, needless to say, a semiring. We define a set function μ on $\mathfrak{A}$ by

$$\mu(R) = \prod_{i}^{\infty} p(A_i).$$

Clearly, by consideration of product measure in 2.3, μ is σ-additive on $\mathfrak{A}$ and thus is so on $\mathfrak{R} = R(\mathfrak{A})$. Now we can extend μ further to a σ-algebra $\mathfrak{M} = S(\mathfrak{R}) = S(\mathfrak{A})$ by measure extension procedure.

Beside measurable rectangles we also consider a subclass $\mathfrak{S}$ of measurable rectangles S of the special form

$$S = (a_1, \ldots, a_n, A, A, \ldots), \qquad a_i \in A,\ n \geq 1$$

which are nothing but the subset wA^N of A^N, where $w = a_1 \ldots a_n \in A^*$. Clearly, each measurable rectangle of $\mathfrak{A}$ is a union no more than countable of sets from $\mathfrak{S}$, and consequently $S(\mathfrak{S}) = S(\mathfrak{A}) = \mathfrak{M}$.

As an immediate consequence of the existence of the product measure on A^N, we have

Theorem 1 *If $X \subseteq A^*$ is a code of A^* such that $A^N = XA^N$, then X is a prefix code and for any Bernoulli distribution p on A, $p(X) = 1$, so X is a maximal code.*

Proof. Set $X' = X - XA^+$. Then X' is a prefix code and $A^N = XA^N = X'A^N = \cup_{w \in X'} wA^N$. The union is certainly countable and disjoint, therefore

$$1 = \mu(A^N) = \mu(\bigcup_{w\in X'} wA^N) = \sum_{w \in X'} \mu(wA^N) = \sum_{w\in X'} p(w) = p(X') \leq p(X).$$

But X is a code, by the Kraft-McMillan inequality, $p(X) \leq 1$, which implies $p(X') = p(X) = 1$ and $X = X'$ is a maximal prefix code. □

For any subset $X \subseteq A^N$, a *cover* of X is a finite or countable collection $\mathfrak{C}$ of sets from $\mathfrak{R}$ such that $X \subseteq \cup_{E \in \mathfrak{C}} E$. Since every set of $\mathfrak{R}$ is a finite or countable union of sets of $\mathfrak{S}$, so we can assume that a cover is always a countable collection of sets from $\mathfrak{S}$ and we write $\mathfrak{C} = \{w_i A^N : i \in I\}$, where $I \subseteq N$. From $\mathfrak{C}$ we discard the redundant subsets, that is, the subsets having no intersection with $X = \emptyset$ or containing another subset $\mathfrak{C}$ to obtain a subclass $\mathfrak{C}' = \{w'A^N : w' \in J \subseteq I\}$ which, evidently, is still a cover of X and besides $\{w' : w'A^N \in \mathfrak{C}'\}$ is a prefix subset of A^*. From now on, speaking of covers, we always mean covers with these properties. Obviously, the outer measure of X is

$$\mu^*(X) = \inf_{\mathfrak{C}} \sum_{wA^N \in \mathfrak{C}} \mu(wA^N) = \inf_{\mathfrak{C}} \sum_{wA^N \in \mathfrak{C}} p(w).$$

We prove now one simple property of the measure μ^*.

Proposition 2 *For any set $X \subseteq A^N$ and $w \in A^*, \mu^*(wX) = p(w)\mu^*(X)$.*

Proof. For any $\epsilon > 0$ let $\mathfrak{C} = \{w_i A^N : i \in I\}$ be a cover of X such that

$$\mu^*(X) \leq \sum_{i \in I} \mu(w_i A^N) = \sum_{i\in I} p(w_i) < \mu^*(X) + \epsilon$$

then $\mathfrak{C}\,' = \{ww_iA^N : i \in I\}$ is a cover of wX and

$$\begin{aligned}\mu^*(wX) \le \sum_{i\in I}\mu(ww_iA^N) &= \sum_{i\in I}p(ww_i)\\ = p(w)\sum_{i\in I}p(w_i) = p(w)\sum_{i\in I}\mu(w_iA^N) &< p(w)(\mu^*(X)+\epsilon)\end{aligned}$$

that means $\mu^*(wX) \le p(w)\mu^*(X)$.

For the reverse inequality, suppose that $\mathfrak{C} = \{w_iA^N : i \in I\}$ is a cover of wX,

$$wX \subseteq \bigcup_{i\in I} w_iA^N \tag{1}$$

such that

$$\mu^*(wX) \le \sum_{i\in I}\mu(w_iA^N) < \mu^*(wX)+\epsilon. \tag{2}$$

If $w = w_iw'$ for some i and $w' \in A^+$, then, in fact, $\mathfrak{C}$ must be a singleton class, $I = \{i\}$, hence

$$\mu^*(wX)+\epsilon > p(w_i) \ge p(w) \ge p(w)\mu^*(X).$$

If now for all i, w is a prefix of w_i, $w_i = ww_i'$, from (1) we have

$$X \subseteq \bigcup_{i\in I} w_i'A^N$$

that means $\mathfrak{C}' = \{w_i'A^N : i \in I\}$ is a cover, for which from (2) we get

$$\begin{aligned}p(w)\mu^*(X) \le p(w)\sum_{i\in I}\mu(w_i'A^N) &= \sum_{i\in I}\mu(ww_i'A^N)\\ = \sum_{i\in I}\mu(w_iA^N) &< \mu^*(wX)+\epsilon.\end{aligned}$$

That is, in both cases, ϵ abitrarily small, we have $p(w)\mu^*(X) \le \mu^*(wX)$ that concludes the proof. □

For any word $w \in A^\infty$ and any subset $E \subseteq A^\infty$ we define

$$\begin{aligned}w^{-1}E &= \{\beta \in A^\infty : (w\beta \in E)\&(w \in A^N) \Rightarrow \beta = \epsilon\};\\ Ew^{-1} &= \{\alpha \in A^\infty : (\alpha w \in E)\&(\alpha \in A^N) \Rightarrow w = \epsilon\}.\end{aligned}$$

The fisrt set is clear; the last one has the following meaning: empty word is the only one to be allowed to cut on the right of an infinite word in E. For any subset $F \subseteq A^\infty$, we write

$$F^{-1}E = \bigcup_{w\in F} w^{-1}E, \qquad EF^{-1} = \bigcup_{w\in F} Ew^{-1}.$$

Further on, p is assumed to be positive.

Proposition 3 *Let X be a subset of A^N and w a finite word of A^*. Then X is measurable if and only if wX is measurable and $\mu(wX) = p(w)\mu(X)$.*

Proof. It is easy to check that

$$w(X \triangle E) = (wX \triangle wE) \tag{3}$$

for any subset $E \subseteq A^N$. Set $E_1 = w^{-1}E$, we have

$$\begin{aligned} wX - wE_1 &= wX - E, \\ wE_1 - wX &\subseteq E - wX. \end{aligned}$$

Hence

$$w(X \triangle E_1) = (wX \triangle wE_1) \subseteq (wX \triangle E). \tag{4}$$

Proposition 2, monotonicity of μ^*, (3) and (4) imply that

$$\begin{aligned} p(w)\mu^*(X \triangle E) &= \mu^*(wX \triangle wE), \\ p(w)\mu^*(X \triangle E_1) &\leq \mu^*(wX \triangle E). \end{aligned}$$

Note that if $E \in \mathfrak{R}$ then $wE, w^{-1}E \in \mathfrak{R}$, so X is measurable iff wX is measurable. The second claim immediately follows from Proposition 2. □

Any language $X \subseteq A^\infty$ is a disjoint union of its finitary part $X_{\text{fin}} = X \cap A^*$ and its infinitaty part $X_{\text{inf}} = X \cap A^N$:

$$X = X_{\text{fin}} \cup X_{\text{inf}}.$$

For a langague of finite words $X \subseteq A^*$, commonly, X^* denotes its Kleene closure, that is $X^* = \{\epsilon\} \bigcup_{i=1}^{\infty} X^i$, or in other words, X^* is the smallest submonoid of A^* (thus of A^∞) containing X. We can extend this notion for any language X of A^∞, namely, X^* by definition is the smallest submonoid of A^∞ containing X, which, as one can easily verify, is $X^*_{\text{fin}} \cup X^*_{\text{fin}} X_{\text{inf}}$.

We recall now the concept of codes on A^∞ [Va]. Given any language X of A^∞ and a word $w \in A^\infty$, a *factorization* of w on X is a finite sequence of words $x_1, \ldots, x_{n-1}, x_n$ such that $x_1, \ldots, x_{n-1} \in X_{\text{fin}}, x_n \in X$ and $w = x_1 \ldots x_{n-1} x_n$. X is said to be an *infinitary code*, or *code* for short, if every word of A^∞ has at most one factorization on X. Clearly, if restricted to A^*, the infinitary codes are just the ordinary ones.

Naturally, we say that a subset $X \subseteq A^\infty$ is *measurable* if its infinitary part X_{inf} is measurable, and the measure $\mu(X)$ is defined to be

$$\mu(X) = p(X_{\text{fin}}) + \mu(X_{\text{inf}}).$$

Now we are in a position to prove the Kraft-McMillan inequality for infinitary codes.

Theorem 4 (Kraft-McMillan Inequality) *For any measurable code X of A^∞, $\mu(X) \leq 1$.*

Proof. Set $f = p(X_{\text{fin}}), i = \mu(X_{\text{inf}})$. We have $f \leq 1$ by Kraft-McMillan Inequality for ordinary codes. Since X is an infinitary code, the union

$$X^*_{\text{fin}} X_{\text{inf}} = \bigcup_{w \in X^*_{\text{fin}}} wX_{\text{inf}}$$

is disjoint. Therefore, by Proposition 2

$$\begin{aligned}\mu(X^*_{\text{fin}}X_{\text{inf}}) &= \sum_{w\in X^*_{\text{fin}}} \mu(wX_{\text{inf}}) = \sum_{w\in X^*_{\text{fin}}} p(w)\mu(X_{\text{inf}}) = \\ &= p(X^*_{\text{fin}})\mu(X_{\text{inf}}) \le 1 = \mu(A^N).\end{aligned}$$

If $f < 1$, then

$$p(X^*_{\text{fin}}) = 1 + f + f^2 + \cdots = \frac{1}{1-f}.$$

Consequently, $\frac{i}{1-f} \le 1$, i.e., $\mu(X) = i + f \le 1$. In the case $f = 1$, we show that $i = 0$. In fact, for all n, $p(X_{\text{fin}} \cup \cdots \cup X^n_{\text{fin}})\mu(X_{\text{inf}}) = ni$. Hence, if $i > 0$, $\mu(X^*_{\text{fin}}X_{\text{inf}}) = \lim_{n\to\infty} ni = \infty$, a contradiction. □

Example 5 A *prefix* of a word $\alpha \in A^\infty$ is a finite word w such that $\alpha = w\beta$ for some $\beta \in A^\infty - \epsilon$; a subset $X \subseteq A^\infty$ is called *prefix* if for any two words in X none of them is a prefix of the other i.e. $X_{\text{fin}}(A^\infty - \epsilon) \cap X = \emptyset$; X is *prefix-maximal* if for any prefix subset $Y, X \subseteq Y$ implies $Y = X$. Evidently, a prefix subset is a code. Every prefix-maximal subset P is measurable and $\mu(P) = 1$. Indeed, since P is prefix-maximal, every word not in P_{inf} has a prefix in P_{fin}, therefore

$$A^N = P_{\text{inf}} \bigcup_{w\in P_{\text{fin}}} wA^N$$

is a disjoint union. Consequently

$$1 = \mu(A^N) = \mu(P_{\text{inf}}) + \sum_{w\in P_{\text{fin}}} \mu(wA^N) = \mu(P_{\text{inf}}) + \sum_{w\in P_{\text{fin}}} p(P_{\text{fin}}) = \mu(P). \ \square$$

When A is a finite alphabet, any recognizable language is measurable, thus we have got a large class of measurable languages, which, by the way, are algorithmically constructible by finite means. Recall that a language $X \subseteq A^N$ is said to be *recognizable* if it is recognized by a finite Büchi automaton [Ei]. It has been well-known that the family Rec A^N of recognizable languages of A^N is the Boolean closure of the family Det A^N of deterministic recognizable ones (Büchi-McNaughton Theorem), i.e. the languages recognized by finite deterministic Büchi automata, which are the finite unions $\bigcup_{i=1}^n B_iC_i^\omega$, where B_i, C_i are (regular) prefix subsets of A^* and C_i^ω stands for the set of infinite words obtained by infinite concatenation of nonempty words of $C_i : C_i^\omega = \{x_1x_2\ldots : x_1, x_2, \ldots \in C_i\}$.

Proposition 6 *Every recognizable language X of A^N is measurable, i.e.* Rec $A^N \subseteq \mathfrak{M}$.

Proof. For any subset $B_iC_i^\omega$ with B_i, C_i prefix subsets of A^* we have

$$B_iC_i^\omega = \bigcap_{n=1}^{\infty} B_iC_i^nA^N.$$

By proposition 2, $B_iC_i^nA^N$ is measurable for all n. Since the σ-algebra $\mathfrak{M}$ of measurable subsets is closed under the formation of Boolean operations, moreover, of countable unions and intersections, $B_i C_i^\omega$ is measurable, hence $\mathrm{Det}A^N \subseteq \mathfrak{M}$ and thus $\mathrm{Rec}A^N \subseteq \mathfrak{M}$. $\square$

We now resume the assumption that A is finite or countable. A code is said to be *maximal* if it cannot be included properly in another code. The existence of a maximal code containing a given code X is easily verified by mean of the Zorn's lemma. A maximal code must has a "nonnegligible" fraction of words in A^N. More precisely, we have

Proposition 7 *For every maximal code X, the outer measure of $X_{\inf}$ is positive:* $\mu^*(X_{\inf}) > 0$.

Proof. Let

$$\mathrm{FD}\,(X_{\inf}) = \{\alpha \in A^N : \exists w \in A^* : w\alpha \in X_{\inf}\}.$$

be the subset of *suffixes* of $X_{\inf}$. Suppose that $\mu^*(X_{\inf}) = 0$, hence $\mu^*(\mathrm{FD}\,(X_{\inf})) = 0$. For any $w \in A^+, w(w^{-1}X_{\inf}) \subseteq X_{\inf}$, we have

$$0 \leq \mu^*(w(w^{-1}X_{\inf})) = p(w)\mu^*(w^{-1}X_{\inf}) \leq \mu^*(X_{\inf}) = 0,$$

hence $p(w)\mu^*(w^{-1}X_{\inf}) = 0$ and so $\mu^*(w^{-1}X_{\inf}) = 0$. Consequently

$$0 \leq \mu^*(\mathrm{FD}\,(X_{\inf})) = \mu^*(\bigcup_{w\in A^*} w^{-1}X_{\inf}) \leq \sum_{w\in A^*} \mu^*(w^{-1}X_{\inf}) = 0$$

(subadditivity of μ^*).

On the other hand, being a maximal code, X is *complete* [Va], i.e., $A^N = \mathrm{FD}\,(X_{\mathrm{fin}}^* X_{\inf})$. By $\mu^*(X_{\inf}) = 0$

$$0 \leq \mu^*(X_{\mathrm{fin}}^* X_{\inf}) \leq \sum_{w\in X_{\mathrm{fin}}^*} \mu^*(wX_{\inf}) = \sum_{w\in X_{\mathrm{fin}}^*} p(w)\mu^*(X_{\inf}) = 0,$$

that is $\mu^*(X_{\mathrm{fin}}^* X_{\inf}) = 0$, therefore

$$\mu^*(\mathrm{FD}\,(X_{\mathrm{fin}}^* X_{\inf})) = 0 = \mu(A^N) = 1,$$

a contradiction. $\square$

Example 8 (a non-measurable subset of A^N) A *suffix* of a word $\alpha \in A^\infty$ is a word β such that $\alpha = w\beta$ for some $w \in A^+$; $X \subseteq A^\infty$ is called a *suffix* subset if there are no words in X one of which is a suffix of the other, i.e. for every $w \in A^+ : X \cap wX = \emptyset$. A suffix set of A^N is called *suffix-maximal* if it is not contained properly in any other suffix subset of A^N. Let S be any suffix-maximal subset of A^N. Suppose that S is measurable; it is easy to see that $S \cup A$ is a code, so we have $\mu(S) = 0$. On the other hand, since $S \cup A$ is even a maximal code, the previous proposition shows that $\mu(S) = \mu^*(S) > 0$. This contradition means that S is not measurable.

In the propositions that follow we prove some properties of codes imposed with special conditions.

Proposition 9 *Let X be a measurable code of A^∞ with $\mu(X) = 1$ and $\mu(X_{\inf}) > 0$, then X_{fin} is a prefix code.*

Proof. We show that X_{fin}^* is *left unitary*, i.e., $X_{\text{fin}}^* = (X_{\text{fin}}^*)^{-1}X_{\text{fin}}^*$, whose base X_{fin} is then a prefix code. Always, $X_{\text{fin}}^* \subseteq (X_{\text{fin}}^*)^{-1}X_{\text{fin}}^*$. For the converse inclusion, we take any nonempty word $w \in (X_{\text{fin}}^*)^{-1}X_{\text{fin}}^*$, so there exist $u, v \in X_{\text{fin}}^*$ such that $uw = v$. Since $\mu(X) = 1, \mu(X_{\text{fin}}^* X_{\text{inf}}) = \frac{i}{1-f} = \frac{i}{i} = 1$, we have $wX_{\text{inf}} \cap X_{\text{fin}}^* X_{\text{inf}} \neq \emptyset$ otherwise

$$\mu(wX_{\text{inf}} \cup X_{\text{fin}}^* X_{\text{inf}}) = \mu(wX_{\text{inf}}) + \mu(X_{\text{fin}}^* X_{\text{inf}}) = p(w)i + 1 > 1$$

that is an obvious contradiction. So there exist $x \in X_{\text{fin}}^*, \alpha, \beta \in X_{\text{inf}}$ such that $w\alpha = x\beta$. Hence $v\alpha = ux\beta$, that implies $v = ux$, as X is a code. Thus $w = x \in X_{\text{fin}}^*$. □

Theorem 10 *If X is a measurable maximal code with $\mu(X) = 1$ then X_{fin} is a prefix code.*

Proof. By Proposition 7, $\mu(X_{\text{inf}}) > 0$ and by the previous proposition the result immediately follows. □

A language $X \subseteq A^\infty$ is called *finite-state* provided the collection $\{w^{-1}X : w \in A^*\}$ is finite. It is not difficult to prove that the family of finite-state languages is closed under the formation of finite unions, of finite intersections and the ω-product. It is noteworthy that Rec A^N is a subfamily of finite-state languages.

Proposition 11 *If X is a maximal code over A satisfying $(X_{\text{fin}}^*)^{-1}X_{\text{fin}}^* = A^*$, then X_{inf} is not a finite-state language if A consists of at least two elements.*

Proof. Under the assumption $(X_{\text{fin}}^*)^{-1}X_{\text{fin}}^* = A^*$, X is a (maximal) code iff X_{inf} is a suffix(-maximal) set. We show that a suffix-maximal language is not finite-state (the fact that it is not recognizable is shown in Example 8).

Fix $x \in A^*$, for any $r \in A^+$ we take a word

$$\alpha = (A^*(rx)^\omega \cup \mathrm{FD}\,(rx^\omega)) \cap X_{\text{inf}} \neq \emptyset.$$

This can be done, as X_{inf} is suffix-maximal. We write $\alpha = a(rx)^\omega$, where $a \in A^*$, hence $\alpha = arx(rx)^\omega$ and $(rx)^\omega \in (arx)^{-1}X_{\text{inf}}$. Thus for any x, there exists $u \in A^*$ such that $(ux)^{-1}X_{\text{inf}} \neq \emptyset$. Consequently, there exists an infinite sequence $v_1, v_2, \ldots$ such that v_i is a suffix of v_{i+1} and $v_i^{-1}X_{\text{inf}} \neq \emptyset$ for all i. As X_{inf} is a suffix set, $v_i^{-1}X_{\text{inf}} \neq v_j^{-1}X_{\text{inf}}$ for $i \neq j$. □

Proposition 12 *If X is a maximal code with X_{fin} a nonsingleton prefix code, then X_{inf} is not finite-state.*

Proof. Suppose on the contrary that X is finite-state. Consider the subset

$$Y_{\text{inf}} = X_{\text{inf}} \cap X_{\text{fin}}^\omega \subseteq X_{\text{fin}}^\omega \tag{5}$$

which is nonempty, since X is a maximal code. For every $w \in X_{\text{fin}}^*$ it is clear that

$$w^{-1}Y_{\text{inf}} = w^{-1}X_{\text{inf}} \cap X_{\text{fin}}^\omega \subseteq X_{\text{fin}}^\omega. \tag{6}$$

Let now c be a *coding morphism* for X_{fin}

$$c : B \to X_{\text{fin}},$$

where B is an alphabet of the same cardinality as X_{fin}. As X is a prefix code, we may correctly extend c to an injective morphism of monoids

$$c : B^{\infty} \rightarrow X_{\text{fin}}^{\infty},$$

where X_{fin}^{∞} denotes $X_{\text{fin}}^{*} \cup X_{\text{fin}}^{\omega}$. Therefore (5) and (6) and the fact that X is finite-state maximal code imply that $B \cup c^{-1}(Y_{\text{inf}})$ is also a finite-state maximal code on B^{∞} with $\text{Card}\, B \geq 2$ that contradicts Proposition 11. Thus X is not finite-state. □

Putting the propositions 6, 10 and 12 all together, we are lead to a situation quite opposite to the case of ordinary codes

Theorem 13 *Let X be a code on the finite alphabet A with X_{inf} a recognizable language of A^N, then the following two assertions are incompatible*

1. $\mu(X) = 1$
2. *X is a maximal code.*

References

[Sm] M. Smorodinsky, *On Infinite Decodable Codes,* Information and Control **11**(1968), 607–612.

[Ei] S. Eilenberg, *Automata, Languages and Machines,* Vol. **A**, Academic Press, New York, 1974.

[Be-Pe] J. Berstel, D. Perrin, *Theory of Codes,* Academic Press, New York, 1985.

[Va] Do Long Van, *Codes avec des mots infinis,* RAIRO Informatique théorique et applications **16**(1982), 371–386.

[Mc] B. McMillan, *Two Inequalities Implied By Unique Decipherability,* IRE Transactions on Information Theory **IT-2**(1956), 115–116.

[Ha] P. R. Halmos, *Measure Theory,* D. Van Nostrand, New York, 1950; Springer-Verlag, New York, 1974.

[Ko-Fo] A. N. Kolmogorov, S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis,* Nauka, Moscow, 1981. (in Russian)

Received January 30, 1993

Revised February 20, 1994

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

Heinz Faßbender † * Heiko Vogler †

Abstract

We formalize a universal unification algorithm for the class of equational theories which is induced by the class of canonical, totally-defined, not strictly subunifiable term rewriting systems (for short: *ctn-trs*). For a *ctn-trs* $\mathcal{R}$ and for two terms t and s, the algorithm computes a ground-complete set of $(E_\mathcal{R}, \Delta)$-unifiers of t and s, where $E_\mathcal{R}$ is the set of rewrite rules of $\mathcal{R}$ viewed as equations and Δ is the set of constructor symbols. The algorithm is based on the *unification-driven leftmost outermost narrowing relation* (for short: *ulo narrowing relation*) which is introduced in this paper. The ulo narrowing relation interleaves leftmost outermost narrowing steps with decomposition steps taken from the usual unification of terms. In its turn, every decomposition step involves a consistency check on constructor symbols combined with a particular form of the occur check. Since decomposition steps are performed as early as possible, some of the nonsuccessful derivations can be stopped earlier than in other universal unification algorithms for ctn-trs's. We give a proof that our algorithm really is a universal unification algorithm.

1 Introduction

The *unification problem* is to determine whether or not, for two given terms t and s, there exists a unifier φ of t and s, i.e., a substitution φ such that $\varphi(t) = \varphi(s)$. It is well-known that the unification problem for first-order terms is decidable [27].

The problem of unification generalizes to the problem of E-unification, if one considers the equality modulo a set E of equations, denoted by $=_E$, rather than the usual equality; $=_E$ is also called the equational theory induced by E. The *E-unification problem* is to determine whether or not, for two given terms t and s, there exists a substitution φ such that $\varphi(t) =_E \varphi(s)$; then φ is called an E-unifier of

*The work of this author has been supported by the Deutsche Forschungsgemeinschaft (DFG).

†Dept. of Theoretical Computer Science, University of Ulm, D-89069 Ulm, Germany, e-mail: {fassbend,vogler}@informatik.uni-ulm.de

t and s. Clearly, the decidability of the E-unification problem depends on the set E of equations. If, e.g., E is the empty set, then the E-unification problem coincides with the unification problem and therefore it is decidable. As another example, if E consists of the algebraic laws of associativity and distributivity, then the E-unification problem becomes undecidable; if the law of associativity is dropped, then it is not known whether the problem is decidable. Surveys about the problem of E-unification can be found in [28,20,18].

For a class $\mathcal{E}$ of equational theories, a *universal unification algorithm for* $\mathcal{E}$ (for short: uu-algorithm for $\mathcal{E}$) is a nondeterministic algorithm which takes as input an equational theory $=_E$ from the class $\mathcal{E}$ and two terms t and s, and which computes a complete set of E-unifiers of t and s (for the definition of complete set of E-unifiers cf., e.g., [28]). In this paper, we will concentrate on uu-algorithms for classes of equational theories which are induced by particular term rewriting systems (for short: trs's). A trs $\mathcal{R}$ induces the equational theory $=_{E_\mathcal{R}}$, where $E_\mathcal{R}$ is the set of rules of $\mathcal{R}$ viewed as equations.

Until now, a lot of research has been carried out to construct uu-algorithms for classes $\mathcal{E}$ of equational theories which are induced by trs's. There exist approaches which are extensions of the unification algorithm in [23] (cf. [19,12,18]). In these approaches there are additional transformation rules which perform the application of equations. Other approaches to construct uu-algorithms are based on the concept of narrowing [21]. More precisely, in every such investigation, a uu-algorithm is constructed for some particular class of trs's where the algorithm is based on a particular narrowing relation (plus some additional actions as, e.g., the usual unification of trees). Here we list some pairs (consisting of a class of trs's and a narrowing relation), for which uu-algorithms have been constructed.

- canonical trs's and narrowing [10,16]
- canonical trs's and basic narrowing [16,24]
- left-linear, non-overlapping trs's and D-narrowing [29]
- canonical, uniform trs's and leftmost outermost narrowing strategy [25]
- canonical, totally-defined, not strictly subunifiable trs's and any narrowing strategy [3].

We note that a narrowing strategy is a narrowing relation in which the narrowing occurrence is fixed. We also recall that a trs is *canonical*, if it is confluent and noetherian. A trs is *constructor-based*, if its ranked alphabet Ω is partitioned into sets F and Δ of function symbols and constructor symbols, respectively; moreover, the left-hand side of every rule is a linear term $f(t_1, \ldots, t_n)$ where f is a function symbol, $t_1, \ldots, t_n$ are terms over $\Delta \cup \mathcal{V}$ where $\mathcal{V}$ is the set of variables (cf. [30]). This particular structure of the left hand sides induces that every constructor term is irreducible. A trs is *totally-defined*, if it is constructor-based and every function symbol is completely defined over its domain or, equivalently: every normal form is a constructor term (cf., e.g., [3]). A trs which is *not strictly subunifiable* (cf. [3] and Subsection 3.1 of the present paper), satisfies a kind of local determinism, e.g., two rules cannot be applied at the same occurrence under the same substitution. In [25] totally-defined, not strictly subunifiable trs's are called *uniform* trs's.

In all mentioned narrowing-based approaches, the narrowing derivation results into two terms t' and s'; then, it has to be checked whether t' and s' are unifiable.

In [11] a uu-algorithm for totally-defined trs's is defined which interleaves unification with the narrowing derivation. More precisely, he considers any innermost narrowing strategy and interleaves decomposition steps without any occur check. Since the decomposition steps are performed as early as possible, it is clear that this can lead to a more efficient computation of $E_{\mathcal{R}}$-unifiers.

There exist some other narrowing relations as, e.g., *lazy narrowing* [26], *outer narrowing* [30] which were shown to be complete with respect to the unrestricted narrowing relation. It was not shown that a uu-algorithm which is based on one of the narrowing relations mentioned above, computes a complete set of $E_{\mathcal{R}}$-unifiers. However, for canonical trs's, this statement is clearly true (cf., e.g., [18] for a complete list of these narrowing relations).

In this paper we construct a uu-algorithm for the class of equational theories which are induced by canonical, totally-defined, not strictly subunifiable trs's (for short: ctn-trs's). This algorithm shall serve as a source for efficient implementations of $E_{\mathcal{R}}$-unification on deterministic abstract machines. Thus, we formalize our uu-algorithm in a way from which an operational approach can be derived easily. This is one of the reasons why we will introduce the uu-algorithm on the basis of a narrowing relation and not as a system of transition rules. The second reason for choosing the formalism of a narrowing relation is that we refine the uu-algorithm of [3] which, in its turn is based on a narrowing relation. The uu-algorithm in [3] improves the algorithm in [16] which is based on the unrestricted narrowing relation, by choosing an arbitrary narrowing strategy. For a particular narrowing strategy, our algorithm improves in its turn the uu-algorithm of [3] by following the idea of interleaving decomposition steps with the narrowing derivation as in [11]. However, we consider the leftmost outermost narrowing strategy and we implement a particular occur check. The relationships between the approaches of [16], [3], and [11], and our approach are illustrated in Figure 1.

More precisely, our uu-algorithm is based on the so-called *unification-driven leftmost outermost narrowing relation* (for short: ulo narrowing relation) which is introduced in this paper. For a trs $\mathcal{R}$, the ulo narrowing relation is denoted by $\overset{u}{\rightsquigarrow}_{\mathcal{R}}$. In $\overset{u}{\rightsquigarrow}_{\mathcal{R}}$ leftmost outermost narrowing is interleaved with the application of decomposition-rules (cf., e.g., [23]) which check the consistency of the root symbols of the terms to be unified. Moreover, the applicability of a decomposition-rule depends on a particular version of the occur check. Since decomposition-rules are applied as early as possible, the ulo narrowing relation is called 'unification-driven'.

Actually, for a ctn-trs $\mathcal{R}$ with some set Δ of constructors and two terms t and s, our uu-algorithm computes a *ground complete set of* $(E_{\mathcal{R}}, \Delta)$*-unifiers* of t and s. An $(E_{\mathcal{R}}, \Delta)$-unifier of t and s is an $E_{\mathcal{R}}$-unifier in which all the images are terms over $\Delta \cup \mathcal{V}$, where $\mathcal{V}$ is the set of variables; in particular, this means that we do not consider unifiers of the form $[z_1/f(t)]$ for some function symbol f. Roughly speaking, a set S of $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s is ground complete, if, for every ground $(E_{\mathcal{R}}, \Delta)$-unifier φ of t and s (i.e., the images of φ do not contain variables), there is a $\psi \in S$ which is more general than φ. This notion will be formalized in Section 3.

Let us give an example at which we can illustrate the ulo narrowing relation. In Figure 2 a set R_1 of rules of the ctn-trs $\mathcal{R}_1$ is shown where we assume to have a ranked alphabet $F_1 = \{sh^{(2)}, mi^{(1)}\}$ of function symbols and a ranked alphabet $\Delta_1 = \{\sigma^{(2)}, \alpha^{(0)}\}$ of constructor symbols. Intuitively, $\mathcal{R}_1$ defines two functions *shovel* and *mirror* with arity 2 and 1, respectively; *mirror* reflects terms over Δ at the vertical center line, and *shovel* accumulates in its second argument the

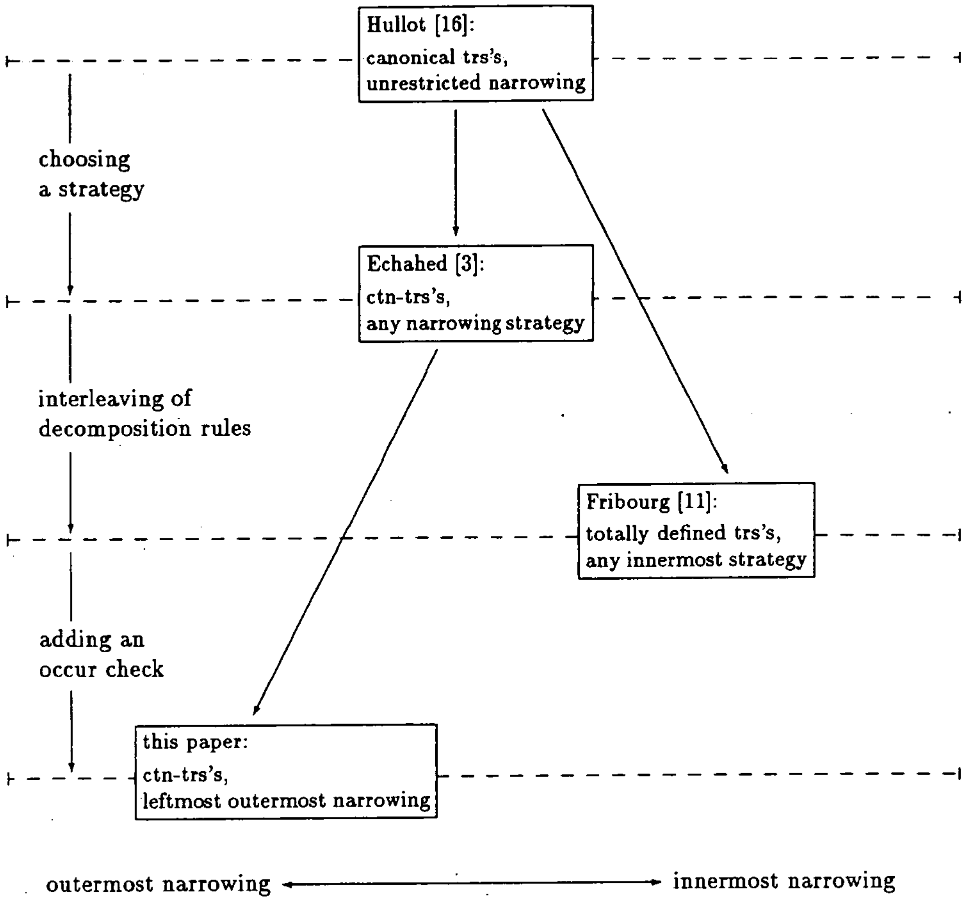

Figure 1: Relationship between some narrowing based approaches.

mirror-image of the second subterm of its first argument. If we consider, e.g., the term $t_1 = \sigma(\sigma(\alpha, s_1), s_2)$ for some terms s_1 and s_2, then for an arbitrary term t_2, $shovel(t_1, t_2)$ is the term $\sigma(mirror(s_1), \sigma(mirror(s_2), t_2))$.

$$\begin{array}{rcll}
sh(\alpha, y_1) & \rightarrow & y_1 & (1) \\
sh(\sigma(x_1, x_2), y_1) & \rightarrow & sh(x_1, \sigma(mi(x_2), y_1)) & (2) \\
mi(\alpha) & \rightarrow & \alpha & (3) \\
mi(\sigma(x_1, x_2)) & \rightarrow & \sigma(mi(x_2), mi(x_1)) & (4)
\end{array}$$

Figure 2: Set of rules of the ctn-trs $\mathcal{R}_1$.

Now we consider the $E_{\mathcal{R}_1}$-unification problem, where the set $E_{\mathcal{R}_1}$ of equations is obtained from R_1 by simply considering the rules as equations. In particular, we want to compute an $E_{\mathcal{R}_1}$-unifier for the terms $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$ in which z_1 and z_2 are free variables. Similar to Hullot in [16], we combine the two terms into one term $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$ with a new binary symbol equ (which is called H in [16]). Next we enrich R_1 by the set $R(\Delta)$ of *decomposition-rules of* Δ (cf. Figure 3). This enrichment yields the trs $\widehat{\mathcal{R}}_1$.

$$\begin{array}{rcll} equ(\alpha,\alpha) & \rightarrow & \alpha & (5) \\ equ(\sigma(x_1,x_2),\sigma(x_3,x_4)) & \rightarrow & \sigma(equ(x_1,x_3),equ(x_2,x_4)) & (6) \end{array}$$

Figure 3: Set of decomposition-rules of Δ_1.

Then a derivation by $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1}$ starting from $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$ may look as follows where we have attached to $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1}$ in every step the narrowing occurrence (in Dewey's notation), the applied rule, and the unifier as additional indices; $\varphi_\emptyset$ denotes the empty substitution; Λ denotes the empty word.

$$\begin{array}{lll}
 & & equ(sh(z_1,\alpha),mi(\sigma(z_2,\alpha))) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,1,(2),[z_1/\sigma(z_3,z_4)]} & equ(sh(z_3,\sigma(mi(z_4),\alpha)),mi(\sigma(z_2,\alpha))) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,1,(1),[z_3/\alpha]} & equ(\sigma(mi(z_4),\alpha),mi(\sigma(z_2,\alpha))) \\
* & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,2,(4),\varphi_\emptyset} & equ(\sigma(mi(z_4),\alpha),\sigma(mi(\alpha),mi(z_2))) \\
** & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,\Lambda,(6),\varphi_\emptyset} & \sigma(equ(mi(z_4),mi(\alpha)),equ(\alpha,mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,11,(3),[z_4/\alpha]} & \sigma(equ(\alpha,mi(\alpha)),equ(\alpha,mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,12,(3),\varphi_\emptyset} & \sigma(equ(\alpha,\alpha),equ(\alpha,mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,1,(5),\varphi_\emptyset} & \sigma(\alpha,equ(\alpha,mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,22,(3),[z_2/\alpha]} & \sigma(\alpha,equ(\alpha,\alpha)) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1,2,(5),\varphi_\emptyset} & \sigma(\alpha,\alpha)
\end{array}$$

If we compose the unifiers which are involved in the narrowing steps, then we obtain the substitution $\varphi = [z_1/\sigma(\alpha, \alpha), z_2/\alpha]$; in fact, φ is a ground $(E_{\mathcal{R}_1}, \Delta_1)$-unifier of $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$. Note that φ is not an $E_{\widehat{\mathcal{R}}_1}$-unifier, because the equational theory is generated by $E_{\mathcal{R}_1}$. The narrowing step at $*$ shows how the ulo narrowing relation deviates from the leftmost outermost narrowing relation. For the latter relation, 11 is the narrowing occurrence in the term $equ(\sigma(mi(z_4), \alpha), mi(\sigma(z_2, \alpha)))$, and then the subterm $mi(z_4)$ has to be narrowed. Note that, since $\mathcal{R}_1$ is constructor-based, every normal form s_1' of the first argument $s_1 = \sigma(mi(z_4), \alpha)$ of equ has the root label σ. Thus, s_1' is unifiable with a normal form s_2' of the second argument $s_2 = mi(\sigma(z_2, \alpha))$ of equ only if the constructors at the root of s_1' and s_2' are identical. Because of reasons of efficiency, it is important to check this consistency as soon as possible. And since the root of s_1 is already a constructor symbol (i.e., s_1 is evaluated *in constructor head normal form*), we narrow s_2 at step $*$ and try to get it also into head normal form. Actually, this form is reached as the result of the application of rule (4). Then, at step **, the consistency of root symbols is checked by applying the decomposition-rule (6).

This paper is organized in five sections where the second section contains preliminaries. In Section 3 we recall the definitions of the leftmost outermost narrowing relation and of ctn-trs's; we recall the uu-algorithm of [3]. In Section 4 we define the ulo narrowing relation and an algorithm of which we prove that it is a uu-algorithm, i.e., that it computes a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers for the class of equational theories $E_{\mathcal{R}}$ which are characterized by ctn-trs's. Finally, Section 5 contains some concluding remarks and indicates further research topics.

2 Preliminaries

We recall and collect some notations, basic definitions, and terminology which will be used in the rest of the paper. We try to be in accordance with the notations in [14] and [2] as much as possible.

2.1 General Notations

We denote the set of nonnegative integers by $\mathbb{N}$. The empty set is denoted by $\emptyset$. For $j \in \mathbb{N}$, $[j]$ denotes the set $\{1, \ldots, j\}$; thus $[0] = \emptyset$. For a finite set A, $\mathcal{P}(A)$ is the *set of subsets* of A and $card(A)$ denotes the *cardinality* of A. As usual for a set A, A^* denotes the set $\bigcup_{n\in\mathbb{N}}\{a_1a_2\ldots a_n \mid \text{for every } i \in [n] : a_i \in A\}$ that is called the set of *words over A; λ denotes the empty word.*

2.2 Ranked Alphabets, Variables, and Terms

A pair $(\Omega, rank_\Omega)$ is called *ranked alphabet,* if Ω is an alphabet and $rank_\Omega : \Omega \longrightarrow \mathbb{N}$ is a total function. For $f \in \Omega$, $rank_\Omega(f)$ is called *rank of f; maxrank*Ω denotes the maximal image of $rank_\Omega$. The subset $\Omega^{(m)}$ of Ω consists of all symbols of rank m $(m \geq 0)$. Note that, for $i \neq j$, $\Omega^{(i)}$ and $\Omega^{(j)}$ are disjoint. We can define a ranked alphabet $(\Omega, rank_\Omega)$ either by enumerating the finitely many subsets $\Omega^{(m)}$ that are not empty, or by giving a set of symbols that are indexed with their (unique) rank. For example, if $\Omega = \{a, b, c\}$ and $rank_\Omega : \Omega \longrightarrow \mathbb{N}$ with $rank_\Omega(a) = 0$, $rank_\Omega(b) = 2$, and $rank_\Omega(c) = 7$, then we can describe $(\Omega, rank_\Omega)$ either by $\Omega^{(0)} = \{a\}$, $\Omega^{(2)} = \{b\}$, and $\Omega^{(7)} = \{c\}$ or by $\{a^{(0)}, b^{(2)}, c^{(7)}\}$. If the ranks of the symbols are clear from the context, then we drop the function $rank_\Omega$ from the denotation of the ranked alphabet $(\Omega, rank_\Omega)$ and simply write Ω.

In the rest of the paper we let $\mathcal{V}$ denote a fixed enumerable set. Its elements are called *variables.* In the following we use the notations $x, x_1, x_2, \ldots, y, y_1, y_2, \ldots, z, z_1, z_2, \ldots$ for variables.

Let Ω be a ranked alphabet and let S be an arbitrary set (in the sequel S will be instantiated by sets of variables). Then the set of *terms over Ω indexed by S,* denoted by $T\langle\Omega\rangle(S)$, is defined inductively as follows: (i) $S \subseteq T\langle\Omega\rangle(S)$ and (ii) for every $f \in \Omega^{(k)}$ with $k \geq 0$ and $t_1, \ldots, t_k \in T\langle\Omega\rangle(S) : f(t_1, \ldots, t_k) \in T\langle\Omega\rangle(S)$. The set $T\langle\Omega\rangle(\emptyset)$, denoted by $T\langle\Omega\rangle$, is called the set of *ground terms over* Ω.

For a term $t \in T\langle\Omega\rangle(\mathcal{V})$, the set of *occurrences of t,* denoted by $O(t)$, is a subset of $\mathbb{N}^*$ and it is defined inductively on the structure of t as follows:

(i) If $t = x$ where $x \in \mathcal{V}$, then $O(t) = \{\Lambda\}$,

(ii) if $t = f$ where $f \in \Omega^{(0)}$, then $O(t) = \{\Lambda\}$, and

(iii) if $t = f(t_1, \ldots, t_n)$ where $f \in \Omega^{(n)}$ and $n > 0$, and for every $i \in [n] : t_i \in T\langle\Omega\rangle(\mathcal{V})$, then $O(t) = \{\Lambda\} \cup \bigcup_{i \in [n]} \{iu \mid u \in O(t_i)\}$.

The prefix order on $O(t)$ is denoted by $<$ and the lexicographical order on $O(t)$ is denoted by $<_{lex}$. The reflexive closures of $<$ and $<_{lex}$ are denoted by $\leq$ and $\leq_{lex}$, respectively. Clearly, $\leq \subseteq \leq_{lex}$. Note that $<_{lex}$ is a total order, whereas, in general, $<$ is a partial order. The minimal element with respect to $\leq_{lex}$ in a subset S of $O(t)$ is denoted by $min_{lex}(S)$. For a term $t \in T\langle\Omega\rangle(\mathcal{V})$ and an occurrence u of t, t/u denotes the *subterm of t at occurrence u*, and $t[u]$ denotes the *label of t at occurrence u*. We use $\mathcal{V}(t)$ to denote the set of variables occurring in t; that is, $x \in \mathcal{V}(t)$, if $x \in \mathcal{V}$ and there exists a $u \in O(t)$ such that $t/u = x$. Finally, we define $t[u \leftarrow s]$ as the term t in which we have replaced the subterm at occurrence u by the term s.

2.3 Algebras, Substitutions, and Congruences

Let $(\Omega, rank_\Omega)$ be a ranked alphabet. An *Ω-algebra* is a pair (A, int_A), where A is a set and int_A is a mapping such that $int_A(f) \in A$, if $rank_\Omega(f) = 0$, and $int_A(f) : A^n \to A$, if $rank_\Omega(f) = n$.

The Ω-algebra $(T\langle\Omega\rangle(\mathcal{V}), int_T)$, where for every $f \in \Omega^{(n)}$ and for every $t_i \in T\langle\Omega\rangle(\mathcal{V})$ with $i \in [n] : int_T(f)(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$, is called the *$\Omega$-term algebra*. It is a free Ω-algebra (cf. [15]).

If (A, int_A) and (B, int_B) are two Ω-algebras, we say that $h : A \to B$ is a *homomorphism*, if for every $f \in \Omega^{(n)}$ with $n \geq 0$ and for every $a_i \in A$ with $i \in [n]$, we have
$$h(int_A(f)(a_1, \ldots, a_n)) = int_B(f)(h(a_1), \ldots, h(a_n)).$$
A mapping $\nu : \mathcal{V} \to A$ is called an *A-assignment*.

The property that every A-assignment can be extended in a unique way to a homomorphism from $T\langle\Omega\rangle(\mathcal{V})$ to A is called the *universal property for the free Ω-algebras* in [15]. We use ν to denote both the A-assignment and its extension.

A *$(\mathcal{V}, \Omega)$-substitution* is a $T\langle\Omega\rangle(\mathcal{V})$-assignment φ, where the set $\{x \mid \varphi(x) \neq x, x \in \mathcal{V}\}$ is finite. The set $\{x \mid \varphi(x) \neq x\}$ is denoted by $\mathcal{D}(\varphi)$ and it is called the *domain of φ*. If $\mathcal{D}(\varphi) = \{x_1, \ldots, x_n\}$, then φ is represented by $[x_1/\varphi(x_1), \ldots, x_n/\varphi(x_n)]$. If $\mathcal{D}(\varphi) = \emptyset$, then φ is denoted by $\varphi_\emptyset$. We say that φ is *ground*, if for every $x \in \mathcal{D}(\varphi) : \mathcal{V}(\varphi(x)) = \emptyset$. The set $\bigcup_{x \in \mathcal{D}(\varphi)} \mathcal{V}(\varphi(x))$ is denoted by $\mathcal{I}(\varphi)$ and is called the set of *variables introduced by φ*. The set of $(\mathcal{V}, \Omega)$-substitutions and the set of ground $(\mathcal{V}, \Omega)$-substitutions are denoted by $Sub(\mathcal{V}, \Omega)$ and $gSub(\mathcal{V}, \Omega)$, respectively. The *composition* of two $(\mathcal{V}, \Omega)$-substitutions φ and ψ is the $T\langle\Omega\rangle(\mathcal{V})$-assignment which is defined by $\psi(\varphi(x))$ for every $x \in \mathcal{V}$. It is denoted by $\varphi \circ \psi$.

An equivalence relation $\sim$ on $T\langle\Omega\rangle(\mathcal{V})$ is called an *Ω-congruence over $T\langle\Omega\rangle(\mathcal{V})$*, if for every $f \in \Omega^{(n)}$ with $n > 0$ and for every $t_1, s_1, \ldots, t_n, s_n \in T\langle\Omega\rangle(\mathcal{V})$ with $t_1 \sim s_1, \ldots, t_n \sim s_n$, the relation $f(t_1, \ldots, t_n) \sim f(s_1, \ldots, s_n)$ holds.

2.4 E-Unification

An *equation over* Ω *and* $\mathcal{V}$ is a pair (t,s), where $t,s \in T\langle\Omega\rangle(\mathcal{V})$. As usual we denote an equation (t,s) by $t = s$. Thus, we consider an equation as an ordered pair. In the rest of the paper, we let E denote a finite set of equations over Ω and $\mathcal{V}$. The *E-equality,* denoted by $=_E$, is the finest (i.e., smallest) congruence relation over $T\langle\Omega\rangle(\mathcal{V})$ containing every pair $(\psi(t),\psi(s))$, where $t = s \in E$ and ψ is an arbitrary $(\mathcal{V},\Omega)$-substitution. If $t =_E s$, then t and s are called *E-equal* (cf. [15]). Two terms $t,s \in T\langle\Omega\rangle(\mathcal{V})$ are called *E-unifiable,* if there exists a $(\mathcal{V},\Omega)$-substitution φ such that $\varphi(t) =_E \varphi(s)$. The set $\{\varphi \mid \varphi(t) =_E \varphi(s)\}$ is called the *set of E-unifiers of t and s,* and it is denoted by $\mathcal{U}_E(t,s)$ (cf. [28]). Let V be a finite subset of $\mathcal{V}$. We define the preorder $\preceq_E (V)$ on $(\mathcal{V},\Omega)$-substitutions by $\varphi \preceq_E \varphi'$ (V), if there exists a $(\mathcal{V},\Omega)$-substitution ψ such that for every $x \in V : \psi(\varphi(x)) =_E \varphi'(x)$ (cf. [28]).

2.5 TRS, Reduction, Narrowing, and Narrowing Trees

A *term rewriting system*, denoted by $\mathcal{R}$, is a pair (Ω,R), where Ω is a ranked alphabet and R is a finite set of rules of the form $l \to r$ such that $l,r \in T\langle\Omega\rangle(\mathcal{V})$ and $\mathcal{V}(r) \subseteq \mathcal{V}(l)$ (cf. [14]). For every term rewriting system $\mathcal{R} = (\Omega,R)$, the *related set of equations,* denoted by $E_{\mathcal{R}}$, is the set $\{l = r \mid l \to r \in R\}$ (cf. [24]).

The *reduction relation associated with* $\mathcal{R}$, denoted by $\Longrightarrow_{\mathcal{R}}$, is defined as follows: for every $t,s \in T\langle\Omega\rangle(\mathcal{V}) : t \Longrightarrow_{\mathcal{R}} s$, if there exist $u \in O(t)$ with $t/u \notin \mathcal{V}, \varphi \in Sub(\mathcal{V},\Omega), l \to r \in R$ with $\varphi(l) = t/u$, and $s = t[u \leftarrow \varphi(r)]$ (cf. [14]). We use the standard notation $\Longrightarrow^*$ to denote the transitive-reflexive closure of $\Longrightarrow$.

A term rewriting system is *canonical,* if it is confluent and noetherian (cf. [15]). A term t is a *normal form of a term s*, if $s \Longrightarrow^*_{\mathcal{R}} t$ and t is irreducible, i.e., there does not exist any term t' such that $t \Longrightarrow_{\mathcal{R}} t'$. For a canonical term rewriting system $\mathcal{R}$, every term t has exactly one normal form (cf. [15]) which is denoted by $nf_{\mathcal{R}}(t)$. A $(\mathcal{V},\Omega)$-substitution φ is *in normal form* if for every $x \in \mathcal{D}(\varphi)$, $\varphi(x)$ is irreducible.

The *set of narrowing interfaces for* $\mathcal{R}$ *and* $t \in T\langle\Omega\rangle(\mathcal{V})$, denoted by $narI(\mathcal{R},t)$, is the set $\{(u,\varphi,l \to r,\rho) \mid u \in O(t), t/u \notin \mathcal{V}, l \to r \in R, \rho$ is a renaming of variables in l such that $\mathcal{V}(\rho(l)) \cap \mathcal{V}(t) = \emptyset, \varphi \in Sub(\mathcal{V},\Omega)$ is the most general unifier of $\rho(l)$ and $t/u\}$. The *set of narrowing occurrences for* $\mathcal{R}$ *and* $t \in T\langle\Omega\rangle(\mathcal{V})$, denoted by $narO(\mathcal{R},t)$, is the set $\{u \mid (u,\varphi,l \to r,\rho) \in narI(\mathcal{R},t)\}$. The *narrowing relation associated with* $\mathcal{R}$, denoted by $\rightsquigarrow_{\mathcal{R}}$, is defined as follows. For every $t,s \in T\langle\Omega\rangle(\mathcal{V})$ and $\psi,\psi' \in Sub(\mathcal{V},\Omega) : (t,\psi) \rightsquigarrow_{\mathcal{R}} (s,\psi')$, if the following three conditions hold:

1. There is a narrowing interface $(u,\varphi,l \to r,\rho) \in narI(\mathcal{R},t)$.
2. $s = \varphi(t[u \leftarrow \rho(r)])$.
3. $\psi' = \psi \circ (\varphi|_{\mathcal{V}(t)})$ (cf. [24]), where composition is read from left to right.

It is obvious that there are two types of nondeterminism involved in the narrowing relation. Starting from a term t, first, there may be more than one narrowing occurrence in t, and second, for a fixed narrowing occurrence, there may be more than one narrowing interface. As usual, for a given starting term t and for given orders on the set of occurrences of t and on the set R of rules, one can collect all the possible narrowing sequences which start from t, into one tree which is called *narrowing tree for t*.

3 $E_{\mathcal{R}}$-Unification by LO Narrowing and Unification

As starting point of our considerations we recall the uu-algorithm which is induced by Theorem 3 in [3]. Here we impose the leftmost outermost narrowing strategy on the narrowing relation of the algorithm.

Before we recall the approach of [3], let us first state that the approach of [25] is technically a bit too complicated for the present purpose although it would theoretically also be a possible starting point. In [25] a uu-algorithm for equational theories induced by canonical, uniform trs's, is presented, where only leftmost outermost narrowing steps are allowed; in fact, ctn-trs's are canonical, uniform trs's.

Furthermore, we note that for ctn-trs, outer narrowing [30] is the same as outermost narrowing. But, in [30], there is no uu-algorithm presented, only a universal matching algorithm.

3.1 The Leftmost Outermost Narrowing Relation and CTN-TRS's

In the leftmost outermost narrowing relation, a pair (t, ψ) derives to a pair (t', ψ') at the minimal element (with respect to $\leq_{lex}$) of the set of narrowing occurrences in t.

Definition 3.1 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system and let $t \in T\langle\Omega\rangle(\mathcal{V})$.

- The *leftmost outermost narrowing occurrence for $\mathcal{R}$ and t*, denoted by *lo-narO*$(\mathcal{R}, t)$, is the narrowing occurrence $min_{lex}(narO(\mathcal{R}, t))$.
- The *set of leftmost outermost narrowing interfaces for $\mathcal{R}$ and t*, denoted by *lo-narI*$(\mathcal{R}, t)$, is the set

$$\{(u, \varphi, l \to r, \rho) \mid (u, \varphi, l \to r, \rho) \in narI(\mathcal{R}, t) \text{ and } u = lo\text{-}narO(\mathcal{R}, t)\}.$$

- The *leftmost outermost narrowing relation associated with $\mathcal{R}$*, denoted by $\overset{lo}{\leadsto}_{\mathcal{R}}$, is defined as follows: for every $t, s \in T\langle\Omega\rangle(\mathcal{V})$ and $\psi, \psi' \in Sub(\mathcal{V}, \Omega)$: (t, ψ) *derives to* (s, ψ') *by* $\overset{lo}{\leadsto}_{\mathcal{R}}$, denoted by $(t, \psi) \overset{lo}{\leadsto}_{\mathcal{R}} (s, \psi')$, if the following three conditions hold:
  1. there is a leftmost outermost narrowing interface $(u, \varphi, l \to r, \rho) \in$ *lo-narI*$(\mathcal{R}, t)$
  2. $s = \varphi(t[u \leftarrow \rho(r)])$

3. $\psi' = \psi \circ (\varphi|_{\mathcal{V}(t)})$ $\oplus$

It is obvious, that $\overset{lo}{\rightsquigarrow}_{\mathcal{R}} \subseteq \rightsquigarrow_{\mathcal{R}}$.

In Example 1 of [3] it is shown that the uu-algorithm of [16] which is based on the unrestricted narrowing relation, is not complete if one imposes a strategy on the narrowing relation. In particular, this negative result holds for the leftmost outermost narrowing relation.

However, Echahed also proves a positive result: the uu-algorithm of [16] stays complete for an arbitrary strategy imposed on the narrowing relation if one restricts to canonical trs's that have the *property of free strategies*. We call these trs's *canonical, totally defined, not strictly sub-unifiable term rewriting systems,* for short: *ctn-trs's*.

A ctn-trs $\mathcal{R} = (\Omega, R)$ is a canonical trs, where Ω is divided into two disjoint ranked alphabets, denoted by F and Δ. F is called the set of function symbols and Δ is called the set of working symbols or constructors. The left hand sides of the rewrite rules in R are linear in $\mathcal{V}$; function symbols only occur at the root of a left hand side. Thus, ctn-trs's are particular constructor-based trs's (cf. [30]). Furthermore, every function symbol in F is totally defined over its domain (cf. Definition 12 in [3]), i.e., if a term is in normal form, then it is in $T\langle\Delta\rangle(\mathcal{V})$. Finally, the left hand sides of the rules in R must be pairwise not strictly sub-unifiable.

Definition 3.2 (cf. [3] Definition 10 and Definition 11). Let $t, t' \in T\langle\Omega\rangle(\mathcal{V})$.

- t and t' are *sub-unifiable,* if there exists an occurrence u in $O(t) \cap O(t')$ such that the following two conditions hold:
  1. t/u and $\rho(t'/u)$ are unifiable with most general unifier σ_u where ρ is a variable-renaming such that $\mathcal{V}(t/u) \cap \mathcal{V}(\rho(t'/u)) = \emptyset$.
  2. For all occurrences w with $w < u$, t/w and t'/w have the same label at the root.
- t and t' are *strictly sub-unifiable,* if there exists an occurrence u where t and t' are sub-unifiable and the corresponding most general unifier σ_u is neither a variable renaming nor the empty substitution. $\oplus$

Example 3.3 Let $\mathcal{R} = (\Omega, R)$ be a canonical trs where $\Omega = \{f^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and let R contain the following rules:

$$\begin{array}{rcll} f(\alpha,\alpha) & \rightarrow & \alpha & (1) \\ f(\gamma(x),\alpha) & \rightarrow & \gamma(\alpha) & (2) \\ f(x,\gamma(y)) & \rightarrow & \gamma(\gamma(\alpha)) & (3) \end{array}$$

- For the trs $\mathcal{R}$, the left hand sides of rule 1 and rule 3 are strictly sub-unifiable at occurrence 1; the same holds for rule 2 and rule 3.
- The left hand sides of rule 1 and rule 2 are sub-unifiable at occurrence 2 but not strictly sub-unifiable, because the most general unifier σ_2 is the empty substitution.

- Let $\mathcal{R}' = (\Omega, R')$ be a trs where R' contains rules 1 and 2 in R and additionally the following two rules:

$$\begin{array}{rcll} f(\alpha, \gamma(y)) & \rightarrow & \gamma(\gamma(\alpha)) & (3) \\ f(\gamma(x), \gamma(y)) & \rightarrow & \gamma(\gamma(\alpha)) & (4) \end{array}$$

The left hand sides of the rules in R' are pairwise not strictly sub-unifiable. Furthermore, the left hand sides of the rules 2 and 3 are not sub-unifiable and the left hand sides of the rules 1 and 4 are not sub-unifiable. $\oplus$

Now, we are able to define ctn-trs.

Definition 3.4 Let $\mathcal{R} = (\Omega, R)$ be a trs. $\mathcal{R}$ is a *canonical, totally defined, not strictly sub-unifiable term rewriting system,* for short *ctn-trs,* if the following conditions hold:

1. $\mathcal{R}$ is canonical.
2. $\Omega = F \cup \Delta$ and $F \cap \Delta = \emptyset$.
3. Every left hand side is linear in $\mathcal{V}$.
4. Every left hand side has the form $f(t_1, \ldots, t_n)$ where $f \in F^{(n)}$ and for every $i \in [n] : t_i \in T\langle\Delta\rangle(\mathcal{V})$.
5. For every $t \in T\langle\Omega\rangle(\mathcal{V}) : nf_{\mathcal{R}}(t) \in T\langle\Delta\rangle(\mathcal{V})$.
6. The left hand sides of the rewrite rules in R are pairwise not strictly sub-unifiable. $\oplus$

In the sequel we will denote a ctn-trs by the triple (F, Δ, R). In fact, the trs in Figure 2 is a ctn-trs. To give the reader an idea about the computational power of ctn-trs's, we mention that every primitive recursive tree function [17] can be described by a ctn-trs (which follows from [6]). But in fact, ctn-trs's are even more powerful.

In general, it is not decidable whether a trs is canonical (cf., e.g., [15]). However, if $\mathcal{R}$ is canonical, then the conditions (2)-(6) in Definition 3.4 are decidable.

3.2 The UU-Algorithm of Echahed

Here we recall the uu-algorithm of Echahed. This algorithm computes particular $E_{\mathcal{R}}$-unifiers which are called ground $(E_{\mathcal{R}}, \Delta)$-unifiers. The range of such a unifier is a subset of $T\langle\Delta\rangle$, i.e., function symbols and variables are not allowed. For a ctn-trs $\mathcal{R}$, this point of view is reasonable, because, in particular, $\mathcal{R}$ is totally defined and every function call can be evaluated into an element of $T\langle\Delta\rangle$. Thus, e.g., if we consider the ctn-trs $\mathcal{R}_1$ in Figure 2 and we want to compute $E_{\mathcal{R}_1}$-unifiers of the terms $mi(x)$ and z, then we are not interested in the minimal $E_{\mathcal{R}_1}$-unifier $[z/mi(x)]$; rather we should be able to compute the unifier $[z/\alpha, x/\alpha]$.

Definition 3.5 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs, let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let $\varphi \in \mathcal{U}_{E_{\mathcal{R}}}(t, s)$ be an $E_{\mathcal{R}}$-unifier of t and s.

- φ is an $(E_{\mathcal{R}}, \Delta)$-*unifier of* t *and* s, if $\varphi \in Sub(\mathcal{V}, \Delta)$.
- φ is a *ground* $(E_{\mathcal{R}}, \Delta)$-*unifier of* t *and* s, if $\varphi \in gSub(\mathcal{V}, \Delta)$.

The sets of $(E_{\mathcal{R}}, \Delta)$-unifiers and of ground $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s are denoted by $\mathcal{U}_{(E_{\mathcal{R}},\Delta)}(t,s)$ and $g\mathcal{U}_{(E_{\mathcal{R}},\Delta)}(t,s)$, respectively. $\oplus$

Similar to the situation of E-unifiers of two terms t and s, we do not have to compute the whole set $g\mathcal{U}_{(E_{\mathcal{R}},\Delta)}(t,s)$, but rather an approximation of it. It suffices to compute a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s.

Definition 3.6 (cf. [3] page 92) Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$ and let W be a finite set of variables containing $V = \mathcal{V}(t) \cup \mathcal{V}(s)$. A set S of $(\mathcal{V}, \Delta)$-substitutions is a *ground complete set of* $(E_{\mathcal{R}}, \Delta)$-*unifiers of* t *and* s *away from* W, if the following three conditions hold:

1. For every $\varphi \in S$: $\mathcal{D}(\varphi) \subseteq V$ and $\mathcal{I}(\varphi) \cap W = \emptyset$.
2. $S \subseteq \mathcal{U}_{(E_{\mathcal{R}},\Delta)}(t,s)$.
3. For every $\varphi \in g\mathcal{U}_{(E_{\mathcal{R}},\Delta)}(t,s)$ there is a $\psi \in S$ such that $\psi \preceq_{E_{\mathcal{R}}} \varphi\ (V)$. $\oplus$

For ctn-trs's, Theorem 3 of [3] shows a uu-algorithm which computes a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers based on an arbitrary narrowing strategy. We present an instance of this theorem where we choose the leftmost outermost strategy. We assume that $\overset{lo}{\leadsto}_{\mathcal{R}}$ is extended to objects of the form $(equ(t,s), \varphi)$ where equ is a new binary symbol, in the way as it is done in, e.g., [16] and [3].

Theorem 3.7 (cf. [3] Theorem 3) Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Delta)$-substitutions φ such that φ is in S iff there exists a derivation by $\overset{lo}{\leadsto}_{\mathcal{R}}$:

$$\begin{aligned}(equ(t,s), \varphi_\emptyset) &\overset{lo}{\leadsto}_{\mathcal{R}} (equ(t_1,s_1), \varphi_1)\\ &\overset{lo}{\leadsto}_{\mathcal{R}} (equ(t_2,s_2), \varphi_2) \overset{lo}{\leadsto}_{\mathcal{R}} \cdots \overset{lo}{\leadsto}_{\mathcal{R}} (equ(t_n,s_n), \varphi_n),\end{aligned}$$

where for every $i \in [n]$: φ_i is in normal form, t_n and s_n are in normal form and unifiable with most general unifier μ, and $\varphi = (\varphi_n \circ \mu)|_V$. Then S is a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s away from V. $\oplus$

Clearly, in the leftmost outermost narrowing relation only one type of nondeterminism occurs, i.e, for a fixed narrowing occurrence, there may be more than one rule applicable. Thus, the leftmost outermost narrowing tree for a term $equ(t,s)$ results from the narrowing tree for $equ(t,s)$ by deleting the branches which do not correspond to derivations by $\overset{lo}{\leadsto}_{\mathcal{R}}$. In Figure 4 we illustrate the leftmost outermost narrowing tree for the term $equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha)))$ and we compare it with the narrowing tree for $equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha)))$. The latter one consists of the shaded and the non-shaded areas, whereas the former one only contains the non-shaded areas. We note that, for the computation of the $E_{\mathcal{R}}$-unifier, it must be checked after the computations of the narrowing derivations, whether the two subtrees contained in the labels of the leaves are unifiable.

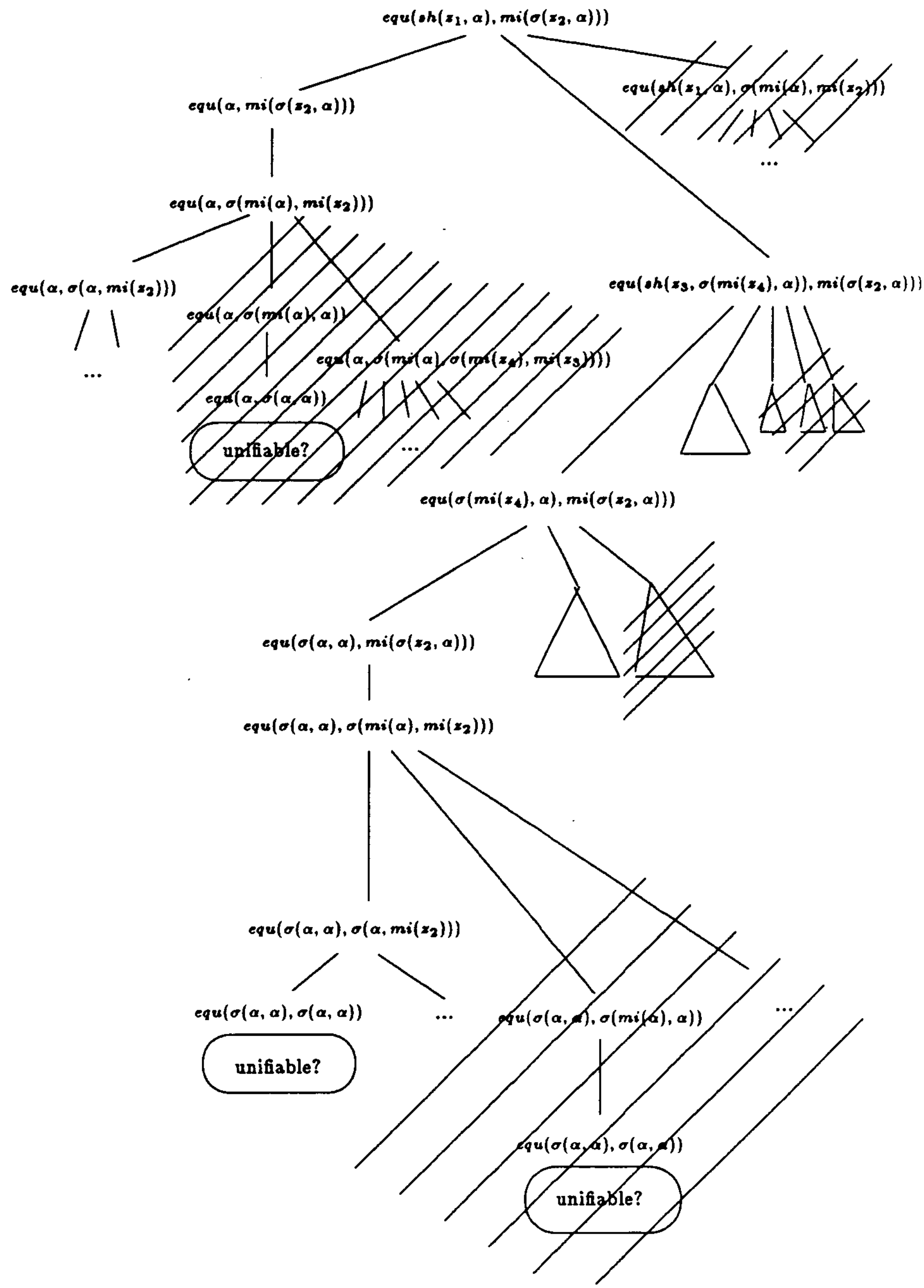

Figure 4: Leftmost outermost narrowing tree for $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$.

In general, by fixing one narrowing occurrence the breadth of the narrowing trees is reduced. Moreover, by choosing the leftmost outermost narrowing strategy, also the depth of narrowing trees is possibly reduced: arguments of functions are only evaluated on demand.

If we regard the shape of narrowing trees as a measure of the complexity of a uu-algorithm, then the uu-algorithm which is induced by Theorem 3.7, is as efficient as the uu-algorithm in [16] which is based on the unrestricted narrowing relation. But in some cases it is even more efficient. This is the reason for paying the price of a reduced expressiveness of ctn-trs's with respect to canonical trs's, because we want to introduce an efficient uu-algorithm.

4 $E_{\mathcal{R}}$-Unification by Unification-Driven LO-Narrowing

In this section we increase the efficiency of the uu-algorithm implied by Theorem 3.7 as follows. Consider a leaf n of some leftmost outermost narrowing tree. Now we view the unification which takes place at n, as a sequence of decomposition steps [23]. Next we split up this sequence and apply every decomposition step as early as possible. Moreover, whether a decomposition step is applicable or not depends on a particular occur check. By means of this technique, some of the derivations that do not yield unifiers, are blocked earlier than in the uu-algorithm of Echahed.

Every decomposition step is formalized as the application of one of the additional rules called *decomposition-rules*. The union of the decomposition-rules and $\mathcal{R}$ itself is called the *extension of* $\mathcal{R}$. Then the ulo narrowing relation is defined on the basis of the extension of $\mathcal{R}$.

We start this section with the definition of the ulo narrowing relation. As an intermediate result, we rephrase Theorem 3.7 by using the ulo narrowing relation (restricted to decomposition-rules) to unify two terms. Finally, based on the ulo narrowing relation, we present a uu-algorithm which computes a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers for every equational theory $=_{E_{\mathcal{R}}}$ where $\mathcal{R}$ is a ctn-trs.

4.1 The Unification-Driven Leftmost Outermost Narrowing Relation

Definition 4.1 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs.

- Let $\sigma \in \Delta^{(k)}$ with $k \geq 0$. The *decomposition-rule for* σ has the form
$$equ(\sigma(x_1, \ldots, x_k), \sigma(x_{k+1}, \ldots, x_{2k})) \rightarrow \sigma(equ(x_1, x_{k+1}), \ldots, equ(x_k, x_{2k})).$$
- The *decomposition-part of* $\mathcal{R}$, denoted by $\mathcal{R}(\Delta)$, is the triple $(\widehat{F}, \Delta, R(\Delta))$ where $\widehat{F} = F \cup \{equ\}$ and equ is a new binary symbol, and $R(\Delta)$ is the set of all decomposition-rules for elements in Δ.
- The *extension of* $\mathcal{R}$, denoted by $\widehat{\mathcal{R}}$, is the triple $(\widehat{F}, \Delta, \widehat{R})$ where $\widehat{R}$ is the set $R \cup R(\Delta)$. $\oplus$

$$
\begin{array}{rcll}
sh(\alpha, y_1) & \to & y_1 & (1) \\
sh(\sigma(x_1, x_2), y_1) & \to & sh(x_1, \sigma(mi(x_2), y_1)) & (2) \\
mi(\alpha) & \to & \alpha & (3) \\
mi(\sigma(x_1, x_2)) & \to & \sigma(mi(x_2), mi(x_1)) & (4) \\
equ(\alpha, \alpha) & \to & \alpha & (5) \\
equ(\sigma(x_1, x_2), \sigma(x_3, x_4)) & \to & \sigma(equ(x_1, x_3), equ(x_2, x_4)) & (6)
\end{array}
$$

Figure 5: Set of rules of an extension.

In Figure 5 the rules of the extension $\widehat{\mathcal{R}}_1 = (\widehat{F}_1, \Delta_1, \widehat{R}_1)$ of $\mathcal{R}_1$ (cf. Figures 2 and 3) are shown where $\widehat{F}_1 = \{sh^{(2)}, mi^{(1)}, equ^{(2)}\}$ and $\Delta_1 = \{\sigma^{(2)}, \alpha^{(0)}\}$.

Roughly speaking, the ulo narrowing relation is almost the same as the leftmost outermost narrowing relation associated with $\widehat{\mathcal{R}}$. But there are the following three differences between the two relations. Let (t, φ) be the current derivation form.

1. Consider the term $t = equ(\alpha, \sigma(mi(\alpha), mi(z_2)))$ at occurrence 11 in the leftmost outermost narrowing tree of Figure 4. The leftmost outermost narrowing occurrence of t is 21. However, it is clear that none of the branches starting from t will yield an $E_{\mathcal{R}_1}$-unifier, because the two direct subterms α and $\sigma(mi(\alpha), mi(z_2))$ of t have different root symbols which cannot be changed in further derivation steps (this is due to the fact that $\mathcal{R}_1$ is constructor-based); hence, the terms α and $\sigma(mi(\alpha), mi(z_2))$ cannot be $E_{\mathcal{R}_1}$-unified. Thus, we will define the ulo narrowing relation in such a way that it blocks at this point. We realize this property by requiring that rules may only be applied at the leftmost occurrence of equ in the current derivation form t. This occurrence of equ is called *important occurrence of* t, denoted by $impO(t)$, because the nonunifiability of the two subterms of t is recognized exactly here. In our concrete situation, $impO(t) = \Lambda$ and none of the decomposition rules is applicable at $impO(t)$; hence, the derivation blocks.

2. If $t/impO(t) = equ(z_i, t')$ or $t/impO(t) = equ(t', z_i)$ where t' is a term the root of which is labelled by a construtor symbol, e.g. σ, then we can apply the decomposition-rule for σ. Clearly, this leads to an instantiation of z_i. Since, in this situation, the algorithm for usual unification of terms [23] would apply the rule for 'elimination of variables' and since this elimination rule requires an occur check, we also have to restrict the applicability of the decomposition-rules by an occur check. However, we may only check whether z_i occurs in the $(\Delta \cup \mathcal{V})$-skeleton of t' (note that the $(\Delta \cup \mathcal{V})$-skeleton is called *shell* in [22]) or not. For instance, the $(\Delta \cup \mathcal{V})$-skeleton of the tree $\sigma(\sigma(\sigma(z_1, \alpha), z_2), \sigma(sh(\alpha, z_1), \alpha))$ is the pattern $\sigma(\sigma(\sigma(z_1, \alpha), z_2), \sigma(., \alpha))$. In general, our algorithm would be incomplete if we would check the whole term t', e.g., if we have the following situation: $t' = \sigma(f(\alpha, z_i), \alpha)$ where f is a new function symbol of rank 2, and there exists a rule $f(\alpha, y_1) \to \alpha$, then $[z_i/\sigma(\alpha, \alpha)]$ is an $E_{\mathcal{R}}$-unifier of z_i and t' which would not be computed if we would apply the occur check to the whole term t', because z_i occurs in t'.

3. If $t/impO(t) = equ(z_i, z_j)$ for two variables z_i and z_j, then, using the leftmost outermost narrowing relation associated with $\widehat{\mathcal{R}}$ in a naive way, (t, φ) derives

to $(\varphi_s(t[impO(t) \leftarrow s]), \varphi \circ \varphi_s)$ for every $s \in T\langle\Delta\rangle$ where $\varphi_s = [z_i/s, z_j/s]$. That means, φ_s would be computed as the most general unifier of z_i and z_j which is certainly wrong. The most general unifier of z_i and z_j is $[z_i/z_k, z_j/z_k]$ where $k-1$ is the maximal index of a free variable in use (cf. [23]). Thus, we define the ulo narrowing relation in such a way that a derivation form (t, φ) with $t/impO(t) = equ(z_i, z_j)$ derives as follows: $t/impO(t)$ is replaced by z_k, every occurrence of z_i and z_j in t is replaced by z_k, and φ is composed with the substitution $[z_i/z_k, z_j/z_k]$.

Before we introduce the ulo narrowing relation, we define some auxiliary notions.

Definition 4.2 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs and let $t \in T\langle\widehat{F} \cup \Delta\rangle(\mathcal{V})$.

- The *important occurrence in t,* denoted by $impO(t)$, is the occurrence $min_{lex}(\{u \in O(t) \mid t[u] = equ\})$.
- t is in *binding mode,* if $t[impO(t)1], t[impO(t)2] \in \mathcal{V}$.
- The $(\Delta \cup \mathcal{V})$*-skeleton of* t is the set

  $$\{u \in O(t) \mid \text{there does not exist any } v \in O(t), v \leq u \text{ and } t[v] \in F\}.$$

- The *occur check for t succeeds,* if the following conditions hold:
  1. t is not in binding mode.
  2. there is an $i \in [2]$ such that $t[impO(t)i] \in \mathcal{V}$ and $t[impO(t)(3-i)] \notin \mathcal{V}$ and there exists an occurrence u in the $(\Delta \cup \mathcal{V})$-skeleton of $t/(impO(t)(3-i))$ such that $t/(impO(t)(3-i))[u] = t[impO(t)i]$. $\bigoplus$

Definition 4.3 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. The *unification-driven leftmost outermost narrowing relation associated with* $\widehat{\mathcal{R}}$, denoted by $\overset{u}{\leadsto}_{\widehat{\mathcal{R}}}$, is defined as follows: for every $t, s \in T\langle\widehat{F} \cup \Delta\rangle(\mathcal{V})$ and $\psi, \psi' \in Sub(\mathcal{V}, \Delta)$: (t, ψ) *derives to* (s, ψ') *by* $\overset{u}{\leadsto}_{\widehat{\mathcal{R}}}$, denoted by $(t, \psi) \overset{u}{\leadsto}_{\widehat{\mathcal{R}}} (s, \psi')$, if $t/impO(t) = equ(t_1, t_2)$ where $t_1, t_2 \in T\langle F \cup \Delta\rangle(\mathcal{V})$ and one of the following four conditions holds:

1. $(t_1[\Lambda], t_2[\Lambda] \in \Delta$ and $t_1[\Lambda] = t_2[\Lambda])$ or $(((t_1[\Lambda] \in \Delta$ and $t_2[\Lambda] \in \mathcal{V})$ or $(t_1[\Lambda] \in \mathcal{V}$ and $t_2[\Lambda] \in \Delta))$ and the occur check fails for $t)$ and the following three conditions hold:
   (a) $(equ(t_1, t_2), \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}(\Delta)} (t', \varphi')$.
   (b) $s = \varphi'(t[impO(t) \leftarrow t'])$.
   (c) $\psi' = \psi \circ \varphi'$.
2. - $t_1, t_2 \in \mathcal{V}$, $t_1 \neq t_2$, and the following three conditions hold where $k = min\{i \mid z_i \in \mathcal{V} \backslash (\mathcal{V}(t) \cup \mathcal{D}(\psi) \cup \mathcal{I}(\psi))\}$:
     (a) $\varphi' = [t_1/z_k, t_2/z_k]$.
     (b) $s = \varphi'(t[impO(t) \leftarrow z_k])$.
     (c) $\psi' = \psi \circ \varphi'$.

- $t_1, t_2 \in \mathcal{V}$, $t_1 = t_2$, and the following two conditions hold:
  (a) $s = t[impO(t) \leftarrow t_1]$.
  (b) $\psi' = \psi$.

3. $t_1[\Lambda] \in F$ and the following three conditions hold:

   (a) $(t_1, \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (t', \varphi')$.
   (b) $s = \varphi'(t[impO(t)1 \leftarrow t'])$.
   (c) $\psi' = \psi \circ \varphi'$.

4. $t_1[\Lambda] \notin F$ and $t_2[\Lambda] \in F$ and the following three conditions hold:

   (a) $(t_2, \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (t', \varphi')$.
   (b) $s = \varphi'(t[impO(t)2 \leftarrow t'])$.
   (c) $\psi' = \psi \circ \varphi'$. $\oplus$

If a rule $l \rightarrow r \in \widehat{R}$ is applied, i.e., in cases 1, 3, and 4, we write $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}, l \rightarrow r}$. In case 2 we write $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}, bm}$ to indicate that the current term is in binding mode.

In the following example we show three derivations by the ulo narrowing relation which illustrate the involved occur check.

Example 4.4 Consider the ctn-trs $\mathcal{R}_1$ and its extension $\widehat{\mathcal{R}}_1$ (cf. Figure 5).
(a) Consider the terms $sh(z_1, \sigma(\alpha, z_2))$ and $\sigma(mi(z_1), \sigma(z_2, \alpha))$. A possible derivation by $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1}$ runs as follows:

$$\begin{array}{ll}
 & (equ(sh(z_1, \sigma(\alpha, z_2)), \sigma(mi(z_1), \sigma(z_2, \alpha))), \varphi_\emptyset) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, 1, (1)} & (equ(\sigma(\alpha, z_2), \sigma(mi(\alpha), \sigma(z_2, \alpha))), [z_1/\alpha]) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, \Lambda, (6)} & (\sigma(equ(\alpha, mi(\alpha)), equ(z_2, \sigma(z_2, \alpha))), [z_1/\alpha]) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, 12, (3)} & (\sigma(equ(\alpha, \alpha), equ(z_2, \sigma(z_2, \alpha))), [z_1/\alpha]) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, 1, (5)} & (\sigma(\alpha, \underline{equ(z_2, \sigma(z_2, \alpha))}), [z_1/\alpha])
\end{array}$$

Here the derivation stops, because the occur check succeeds.

(b) Consider the terms $sh(z_1, \sigma(\alpha, z_2))$ and $\sigma(mi(z_1), \sigma(z_3, \alpha))$. The first four derivation steps are analogous to those one in (a).

$$\begin{array}{ll}
 & (equ(sh(z_1, \sigma(\alpha, z_2)), \sigma(mi(z_1), \sigma(z_3, \alpha))), \varphi_\emptyset) \\
\overset{u}{\rightsquigarrow}{}^4_{\widehat{\mathcal{R}}_1} & (\sigma(\alpha, \underline{equ(z_2, \sigma(z_3, \alpha))}), [z_1/\alpha]) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, 2, (6)} & (\sigma(\alpha, \sigma(equ(z_4, z_3), equ(z_5, \alpha))), [z_1/\alpha, z_2/\sigma(z_4, z_5)]) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, 21, bm} & (\sigma(\alpha, \sigma(z_6, equ(z_5, \alpha))), [z_1/\alpha, z_2/\sigma(z_6, z_5), z_3/z_6]) \\
\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_1, 22, (5)} & (\sigma(\alpha, \sigma(z_6, \alpha)), [z_1/\alpha, z_2/\sigma(z_6, \alpha), z_3/z_6])
\end{array}$$

Here the derivation yields the $E_{\mathcal{R}_1}$-unifier $[z_1/\alpha, z_2/\sigma(z_6,\alpha), z_3/z_6]$.

(c) Now enrich $\mathcal{R}_1$ by the rules $sh(\beta, y) \to \beta$ (with number (7)) and $mi(\beta) \to \beta$ (with number (8)) where $\beta \in \Delta^{(0)}$. Denote this ctn-trs by $\mathcal{R}_2$ and its extension by $\widehat{\mathcal{R}}_2$ where the decomposition-rule for β has the number (9). Consider the terms z_1 and $\sigma(\alpha, sh(z_2, z_1))$.

$$
\begin{array}{lll}
 & & (equ(z_1, \sigma(\alpha, sh(z_2, z_1))), \varphi_\emptyset) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_2, \Lambda, (6)} & (\sigma(equ(z_3, \alpha), equ(z_4, sh(z_2, \sigma(z_3, z_4)))), [z_1/\sigma(z_3, z_4)]) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_2, 1, (5)} & (\sigma(\alpha, equ(z_4, sh(z_2, \sigma(\alpha, z_4)))), [z_1/\sigma(\alpha, z_4)]) \\
(*) & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_2, 22, (7)} & (\sigma(\alpha, equ(z_4, \beta)), [z_1/\sigma(\alpha, z_4), z_2/\beta]) \\
 & \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}_2, 2, (9)} & (\sigma(\alpha, \beta), [z_1/\sigma(\alpha, \beta), z_2/\beta])
\end{array}
$$

Hence, this derivation yields the $E_{\mathcal{R}_2}$-unifier $[z_1/\sigma(\alpha,\beta), z_2/\beta]$. Note that at (*) the occur check is only applied to the $(\Delta \cup \mathcal{V})$-skeleton of $sh(z_2, \sigma(\alpha, z_4))$. $\bigoplus$

4.2 Unification by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$

As an intermediate result between Theorem 3.7 and the intended uu-algorithm in Theorem 4.7 which is based on the ulo narrowing relation, we show in this subsection that the usual unification of two terms $t, s \in T\langle\Delta\rangle(\mathcal{V})$ can be realized by a derivation by the ulo narrowing relation associated with $\mathcal{R}(\Delta)$.

Lemma 4.5 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs and let $t, s \in T\langle\Delta\rangle(\mathcal{V})$. The terms t and s are unifiable with most general unifier φ iff there exists a derivation by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$ of the following form $(equ(t,s), \varphi_\emptyset) \overset{u}{\rightsquigarrow}{}^{*}_{\mathcal{R}(\Delta)} (t', \varphi)$ and $t' \in T\langle\Delta\rangle(\mathcal{V})$.

Proof: For the usual term unification, we consider the algorithm in [12] which transforms sets of unordered pairs. Let us briefly recall this algorithm. The unification of t and s starts with the set $P = \{\langle t, s\rangle\}$. Then, a finite number of transformations is applied step by step to this set. Every transformation is of one of the following three types:

1. If $\langle z_i, z_i\rangle \in P$, then P is transformed into the set $P \backslash \{\langle z_i, z_i\rangle\}$.
2. If $\langle \sigma(t_1, \ldots, t_k), \sigma(s_1, \ldots, s_k)\rangle \in P$, then P is transformed into the set $P \backslash \{\langle \sigma(t_1, \ldots, t_k), \sigma(s_1, \ldots, s_k)\rangle\} \cup \{\langle t_1, s_1\rangle, \ldots, \langle t_k, s_k\rangle\}$.
3. If $\langle z_i, s\rangle \in P$ such that z_i does not occur in s, then P is transformed into $\varphi(P \backslash \{\langle z_i, s\rangle\}) \cup \{\langle z_i, s\rangle\}$, where $\varphi = [z_i/s]$ and the φ-image of a set is defined as the set of the φ-images of its elements.

The algorithm stops, if P is in solved form, i.e., $P = \{\langle z_i, t_i\rangle \mid i \in [n]\}$ where for every $i, j \in [n] : z_i \neq z_j$ for $i \neq j$ and z_i does not occur in any t_j. Then, $[z_1/t_1, \dots, z_n/t_n]$ is the most general unifier of t and s.

Let us note that the algorithm computes the same unifier (modulo variable renaming) if a strategy is imposed on the order in which the transformation steps are applied. Thus, we can choose the order which corresponds to leftmost outermost narrowing by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$.

Each transformation of the unification algorithm corresponds to the following derivations by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$ where $(t, \varphi) \in T\langle\Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta)$.

1. A transformation of type 1 corresponds to the derivation step $(t, \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)} (t[impO(t) \leftarrow z_i], \varphi)$, because $t/impO(t) = equ(z_i, z_i)$. Then, the substitution φ is not changed.

2. A transformation of type 2 corresponds to the derivation step $(t, \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)} (t', \varphi)$, where $t' = t[impO(t) \leftarrow \sigma(equ(t_1, s_1), \dots, equ(t_k, s_k))]$ and φ is not changed, because $t/impO(t) = equ(\sigma(t_1, \dots, t_k), \sigma(s_1, \dots, s_k))$. Thus, an application of an decomposition-rule covers the transformation of type 2.

3. The correspondence of a transformation of type 3 is split up into two cases.

   Case 1: If $s \notin \mathcal{V}$, then the transformation corresponds to the derivation $(t, \varphi) \overset{u}{\rightsquigarrow}{}^{*}_{\mathcal{R}(\Delta)} (t', \varphi \circ [z_i/s])$, where t' is the term that results from t by replacing every occurrence of z_i by s. The length of this derivation is $size(s)$, because decomposition-rules are applied node by node in s. Note that the applicability of decomposition-rules is subjected to an occur check (cf. Definition 4.3 1.).

   Case 2: If $s = z_j$ with $j \neq i$, then t is in binding form and the transformation corresponds the derivation step $(t, \varphi) \overset{u}{\rightsquigarrow}{}^{*}_{\mathcal{R}(\Delta),bm} (t', \varphi \circ [z_i/z_k, z_i/z_k])$ where z_k is a new variable.

Conversely, in the definition of $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$, there occurs exactly one of the cases 1, 2, 3.1, and 3.2. In every of these cases, the derivation step by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$ corresponds to the transformation of the unification algorithm which is mentioned above. $\oplus$

The unification of the terms $t = \sigma(z_1, z_2)$ and $s = \sigma(\sigma(z_2, \alpha), \alpha)$ via a derivation by $\overset{u}{\rightsquigarrow}_{\mathcal{R}_1(\Delta_1)}$ is shown in Figure 6 (for $\mathcal{R}_1$ and Δ_1 cf. Figure 2). The most general unifier is $\theta = [z_1/\sigma(\alpha, \alpha), z_2/\alpha]$.

Now we rephrase Theorem 3.7 by replacing the unification by a derivation induced by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$.

Theorem 4.6 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Delta)$-substitutions φ such that φ is in S iff there exists a derivation by $\overset{lo}{\rightsquigarrow}_{\mathcal{R}}$:

$$\begin{aligned}(equ(t, s), \varphi_\emptyset) &\overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(t_1, s_1), \varphi_1) \\ &\overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(t_2, s_2), \varphi_2) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} \cdots \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(t_n, s_n), \varphi_n),\end{aligned}$$

$$
\begin{array}{ll}
 & (equ(\sigma(z_1, z_2), \sigma(\sigma(z_2, \alpha), \alpha)), \varphi_\emptyset) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(6)} & (\sigma(equ(z_1, \sigma(z_2, \alpha)), equ(z_2, \alpha)), \varphi_\emptyset) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(6)} & (\sigma(\sigma(equ(z_3, z_2), equ(z_4, \alpha)), equ(z_2, \alpha)), [z_1/\sigma(z_3, z_4)]) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),bm} & (\sigma(\sigma(z_5, equ(z_4, \alpha)), equ(z_5, \alpha)), [z_1/\sigma(z_5, z_4), z_2/z_5]) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(5)} & (\sigma(\sigma(z_5, \alpha), equ(z_5, \alpha)), [z_1/\sigma(z_5, \alpha), z_2/z_5]) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(5)} & (\sigma(\sigma(\alpha, \alpha), \alpha), [z_1/\sigma(\alpha, \alpha), z_2/\alpha])
\end{array}
$$

Figure 6: A unification by a derivation by $\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1)}$.

where for every $i \in [n]$: φ_i is in normal form, t_n and s_n are in normal form, and there exists a derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$:

$$(equ(t_n, s_n), \varphi_n) \overset{u}{\leadsto}{}^*_{\mathcal{R}(\Delta)} (t', \varphi'),$$

such that $t' \in T\langle\Delta\rangle(\mathcal{V})$ and $\varphi = \varphi'|_V$. Then S is a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers of t and s away from V.

Proof: The correctness of Theorem 4.6 immediately follows from Theorem 3.7 and from Lemma 4.5. $\oplus$

4.3 $E_\mathcal{R}$-Unification by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$

We finish this section by showing that we can compute a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers of two terms t and s by derivations induced by the ulo narrowing relation.

Theorem 4.7 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Delta)$-substitutions φ such that φ is in S iff there exists a derivation by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$:

$$(equ(t, s), \varphi_\emptyset) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (t_1, \varphi_1) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (t_2, \varphi_2) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} \cdots \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (t_n, \varphi_n),$$

where for every $i \in [n]$: φ_i is in normal form, $t_n \in T\langle\Delta\rangle(\mathcal{V})$, and $\varphi = \varphi_n|_V$. Then S is a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers of t and s away from V.

Proof: We show that there exists a derivation

$$(equ(t, s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^*_{\mathcal{R}} (equ(t', s'), \varphi') \overset{u}{\leadsto}{}^*_{\mathcal{R}(\Delta)} (t^*, \varphi^*), \qquad (1)$$

where $t', s', t^* \in T\langle\Delta\rangle(\mathcal{V})$ and $\varphi', \varphi^* \in Sub(\mathcal{V}, \Delta)$ iff there exists a derivation

$$(equ(t, s), \varphi_\emptyset) \overset{u}{\leadsto}{}^*_{\hat{\mathcal{R}}} (t^*, \varphi^*) \qquad (2)$$

Then from Theorem 4.6 the correctness of the present theorem follows.

Derivation 1 $\Longrightarrow$ Derivation 2

First, we show that for every derivation 1, there exists a derivation 2. For this purpose, we introduce the function *eqpos* : $T\langle F \cup \Delta\rangle(\mathcal{V}) \times T\langle F \cup \Delta\rangle(\mathcal{V}) \to \mathbb{N}$ that yields, for two terms t_1 and t_2, the maximal number of steps which can be performed by $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$ on the term $equ(t_1, t_2)$. In order to describe this function, we first have to find out the first occurrence $notequ(t_1, t_2)$ in t_1 or t_2 at which no decomposition-rule is applicable; $notequ(t_1, t_2)$ is defined by

$$\begin{array}{ll} min_{lex}(\{u \in O(t_1) \cup O(t_2) \mid & t_1[u] \in F \text{ or } t_2[u] \in F \text{ or} \\ & (t_1[u] \in \Delta \text{ and } t_2[u] \in \Delta \text{ and } t_1[u] \neq t_2[u]) \text{ or} \\ & \text{the occur check for } equ(t_1[u], t_2[u]) \text{ succeeds}\}) \end{array}$$

Then $eqpos(t_1, t_2)$ is defined by summing up the number of possible applications of decomposition-rules at occurrences which are common to t_1 and t_2.

$$\sum_{\{u \in O(t_1) \cap O(t_2) \mid u <_{lex} notequ(t_1, t_2)\}} equsteps(t_1, t_2, u)$$

$equsteps(t_1, t_2, u)$ is the number of possible applications of decomposition-rules at occurrence u. Let $t' = t_{3-i}/u$.

$$equsteps(t_1, t_2, u) = \begin{cases} 1 & \text{if } t_1[u], t_2[u] \in \Delta \text{ and } t_1[u] = t_2[u] \\ 1 & \text{if } t_1[u], t_2[u] \in \mathcal{V} \\ n & \text{if, for some } i \in [2] : t_i[u] \in \mathcal{V}, t' \in T\langle\Delta\rangle(\mathcal{V})\backslash\mathcal{V} \\ & \text{and } n = card(O(t')) \\ n & \text{if, for some } i \in [2] : t_i[u] \in \mathcal{V}, \\ & t' \in T\langle F \cup \Delta\rangle(\mathcal{V})\backslash T\langle\Delta\rangle(\mathcal{V}) \text{ and } n = \\ & card(\{w \in O(t') \mid w <_{lex} min_{lex}(\{v \mid t'[v] \in F\})\}) \end{cases}$$

To give an example, consider the following two terms $t_1 = \sigma(\sigma(\sigma(\alpha, \alpha), z_1), \sigma(f(\alpha), \alpha))$ and $t_2 = \sigma(\sigma(z_2, z_3), \sigma(\sigma(\alpha, \alpha), \alpha))$ (in Figure 7, the occurrences at which a decomposition-rule is applicable, are enclosed).

Obviously, $notequ(t_1, t_2) = 21$. Hence, $eqpos(t_1, t_2) = equsteps(t_1, t_2, \Lambda) + equsteps(t_1, t_2, 1) + equsteps(t_1, t_2, 11) + equsteps(t_1, t_2, 12) + equsteps(t_1, t_2, 2)$.

And $equsteps(t_1, t_2, \Lambda) = equsteps(t_1, t_2, 1) = equsteps(t_1, t_2, 12) = equsteps(t_1, t_2, 2) = 1$, and $equsteps(t_1, t_2, 11) = 3$. Thus, $eqpos(t_1, t_2) = 7$. This means that, starting from $equ(t_1, t_2)$, it is possible to perform exactly 7 applications of some decomposition-rule. The result after application of 7 decomposition-rules is the term

$$\sigma(\sigma(\sigma(\alpha, \alpha), z_4), \sigma(equ(f(\alpha), \sigma(\alpha, \alpha)), equ(\alpha, \alpha)))$$

which is shown in Figure 8, where z_4 results from the handling of the binding mode.

Furthermore, we prove the following Claim by induction on k.

Claim 1 For every $k \geq 0$, $\varsigma_t, \varsigma_s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, $\varsigma \in T\langle \widehat{F} \cup \Delta\rangle(\mathcal{V})$, and for every $\varphi, \psi \in Sub(\mathcal{V}, \Delta)$: If there exists a derivation

$$(equ(t, s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}{}^{k}_{\mathcal{R}} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{u}{\rightsquigarrow}{}^{eqpos(\varsigma_t, \varsigma_s)}_{\mathcal{R}(\Delta)} (\varsigma, \psi),$$

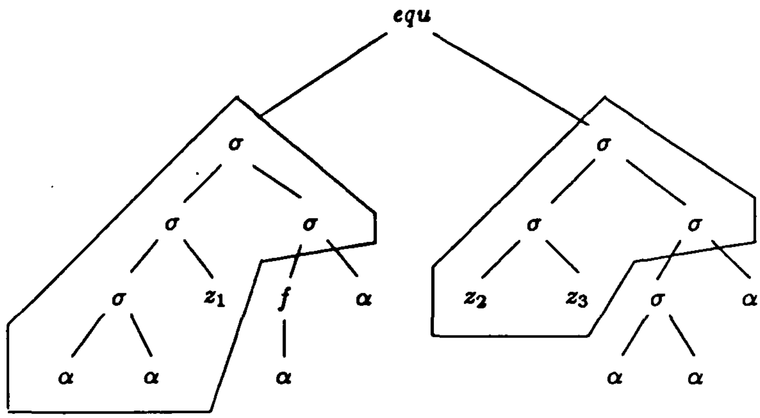

Figure 7: The term $equ(t_1, t_2)$.

then there exists a derivation

$$(equ(t,s),\varphi_\emptyset) \overset{u}{\rightsquigarrow}_{\widehat{R}}^{k+eqpos(\varsigma_t,\varsigma_s)} (\varsigma,\psi).$$

Induction on k:

<u>$k=0$:</u> $\varsigma_t = t$ and $\varsigma_s = s$. We have $(equ(t,s),\varphi_\emptyset) \overset{u}{\rightsquigarrow}_{R(\Delta)}^{eqpos(\varsigma_t,\varsigma_s)} (\varsigma,\psi)$.

From $R(\Delta) \subseteq \widehat{R}$ follows $(equ(t,s),\varphi_\emptyset) \overset{u}{\rightsquigarrow}_{\widehat{R}}^{eqpos(\varsigma_t,\varsigma_s)} (\varsigma,\psi)$.

<u>$k \to k+1$:</u> There exist $\varsigma_t', \varsigma_s' \in T\langle F \cup \Delta\rangle(\mathcal{V})$, $\varsigma' \in T\langle \widehat{F} \cup \Delta\rangle(\mathcal{V})$, $\varphi', \psi' \in Sub(\mathcal{V},\Delta)$, and there exists the following derivation:

$$(equ(t,s),\varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k} (equ(\varsigma_t,\varsigma_s),\varphi) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(\varsigma_t',\varsigma_s'),\varphi') \overset{u}{\rightsquigarrow}_{R(\Delta)}^{eqpos(\varsigma_t',\varsigma_s')} (\varsigma',\psi').$$

Now we split the derivation by $\overset{u}{\rightsquigarrow}_{R(\Delta)}$ into two derivations: There exist $\bar{\varsigma} \in T\langle \widehat{F} \cup \Delta\rangle(\mathcal{V})$, $\bar{\varphi} \in Sub(\mathcal{V},\Delta)$, and there exists the following derivation:

$$(equ(t,s),\varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k} (equ(\varsigma_t,\varsigma_s),\varphi) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(\varsigma_t',\varsigma_s'),\varphi') \overset{u}{\rightsquigarrow}_{R(\Delta)}^{eqpos(\varsigma_t,\varsigma_s)} (\bar{\varsigma},\bar{\varphi})$$
$$\overset{u}{\rightsquigarrow}_{R(\Delta)}^{eqpos(\varsigma_t',\varsigma_s')-eqpos(\varsigma_t,\varsigma_s)} (\varsigma',\psi').$$

There exist $\bar{\varsigma}' \in T\langle \widehat{F} \cup \Delta\rangle(\mathcal{V})$, $\bar{\varphi}' \in Sub(\mathcal{V},\Delta)$, and there exists the following derivation by changing the order of applications of rules in the previous derivation:

$$(equ(t,s),\varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k} (equ(\varsigma_t,\varsigma_s),\varphi) \overset{u}{\rightsquigarrow}_{R(\Delta)}^{eqpos(\varsigma_t,\varsigma_s)} (\bar{\varsigma}',\bar{\varphi}') \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}} (\bar{\varsigma},\bar{\varphi})$$
$$\overset{u}{\rightsquigarrow}_{R(\Delta)}^{eqpos(\varsigma_t',\varsigma_s')-eqpos(\varsigma_t,\varsigma_s)} (\varsigma',\psi').$$

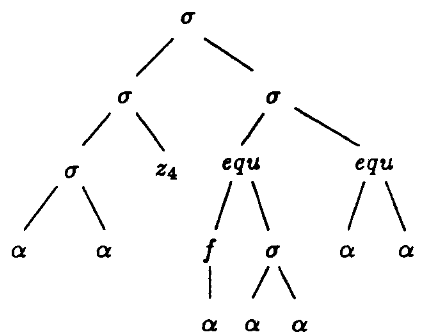

Figure 8: Resulting term t^* after application of seven decomposition-rules.

Changing the order of the derivation is correct, because in the derivation step $(equ(\varsigma_t, \varsigma_s), \varphi) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(\varsigma_t', \varsigma_s'), \varphi')$, a function f is applied at the leftmost outermost narrowing occurrence. From the definition of *eqpos* it follows that f is also the label of the leftmost outermost narrowing occurrence in $\bar{\varsigma}'$. Furthermore, in the case of a function application, the relations $\overset{lo}{\rightsquigarrow}_{\mathcal{R}}$ and $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}}$ yield the same result.

<u>*Example:*</u> Let $\varsigma_t = t_1$ and $\varsigma_s = t_2$ in Figure 7 and let $f(\alpha) \rightarrow \alpha$ be a rule in R. The leftmost outermost narrowing occurrence in $equ(t_1, t_2)$ in Figure 7 is occurrence 121 which is labelled by the function symbol f. After $eqpos(t_1, t_2) = 7$ applications of decomposition-rules we get the term t^* in Figure 8 which is denoted by $\bar{\varsigma}'$ in the proof. The leftmost outermost narrowing occurrence in t^* is the occurrence 211 which is also labelled by f. Furthermore, the next step in the derivation by $\overset{lo}{\rightsquigarrow}_{\mathcal{R}}$ starting with $equ(t_1, t_2)$ in Figure 7 is analogous to the next step in the derivation by $\overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}}$ starting with t^* in Figure 8. □

The existence of the following derivation follows from the induction hypothesis:

$$(equ(t, s), \varphi_\emptyset) \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}}^{k+eqpos(\varsigma_t, \varsigma_s)} (\bar{\varsigma}', \bar{\varphi}') \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}} (\bar{\varsigma}, \bar{\varphi}) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqpos(\varsigma_t', \varsigma_s') - eqpos(\varsigma_t, \varsigma_s)} (\varsigma', \psi').$$

The existence of the following derivation follows from $R(\Delta) \subseteq \widehat{R}$:

$$(equ(t, s), \varphi_\emptyset) \overset{u}{\rightsquigarrow}_{\widehat{\mathcal{R}}}^{k+1+eqpos(\varsigma_t', \varsigma_s')} (\varsigma', \psi').$$

This finishes the proof of Claim 1.

Especially, if k is equal to the length of the derivation by $\overset{lo}{\rightsquigarrow}_{\mathcal{R}}$ in derivation 1, it follows that for every derivation 1, there exists a derivation 2.

Derivation 2 $\Longrightarrow$ Derivation 1

Now we show that for every derivation 2, there exists a derivation 1. For this purpose, we introduce the function *eqapp* : $T\langle\widehat{F} \cup \Delta\rangle(\mathcal{V}) \to \mathbb{N}$ that yields, for a term t, the sum of applications of decomposition-rules and steps started by a term in binding mode, in the derivation by $\overset{u}{\leadsto}_{\widehat{\mathcal{R}}}$ up to t.

$$eqapp(t) = card(\{u \in O(t) \mid u <_{lex} impO(t)\})$$

Furthermore, we prove the following claim by induction on k.

Claim 2 For every $k \geq 0$, $\varsigma \in T\langle\widehat{F} \cup \Delta\rangle(\mathcal{V})$, and $\psi \in Sub(\mathcal{V}, \Delta)$: If there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}{}^{k}_{\widehat{\mathcal{R}}} (\varsigma, \psi),$$

then there exist $\varsigma_t, \varsigma_s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, $\varphi \in Sub(\mathcal{V}, \Delta)$, and there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k-eqapp(\varsigma)}_{\mathcal{R}} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{u}{\leadsto}{}^{eqapp(\varsigma)}_{\mathcal{R}(\Delta)} (\varsigma, \psi).$$

<u>*Induction on k:*</u>
<u>$k = 0$:</u> $\varsigma = equ(t,s)$, $\psi = \varphi_\emptyset$. Thus, $eqapp(\varsigma) = 0$. We have

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{0-0}_{\mathcal{R}} (equ(\varsigma_t, \varsigma_s), \varphi_\emptyset) \overset{u}{\leadsto}{}^{0}_{\mathcal{R}(\Delta)} (\varsigma, \psi).$$

<u>$k \to k+1$:</u> There exist $\varsigma' \in T\langle\widehat{F} \cup \Delta\rangle(\mathcal{V})$, $\psi' \in Sub(\mathcal{V}, \Delta)$, and there exists the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}{}^{k}_{\widehat{\mathcal{R}}} (\varsigma, \psi) \overset{u}{\leadsto}_{\widehat{\mathcal{R}}} (\varsigma', \psi').$$

From the induction hypothesis it follows that there exists the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k-eqapp(\varsigma)}_{\mathcal{R}} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{u}{\leadsto}{}^{eqapp(\varsigma)}_{\mathcal{R}(\Delta)} (\varsigma, \psi) \overset{u}{\leadsto}_{\widehat{\mathcal{R}}} (\varsigma', \psi').$$

Now we have to distinguish the following two cases:

<u>*Case 1:*</u> $eqapp(\varsigma') = eqapp(\varsigma)$. Then, the $k+1$th derivation step is a function application. The same function application can be applied to the term $equ(\varsigma_t, \varsigma_s)$ in a derivation step by $\overset{lo}{\leadsto}_{\mathcal{R}}$.

<u>*Example:*</u> Let $\varsigma_t = t_1$ and $\varsigma_s = t_2$ in Figure 7 and let ς be the term t^* in Figure 8; $eqapp(t^*) = 7$. The next derivation step in the derivation by $\overset{u}{\leadsto}_{\widehat{\mathcal{R}}}$ starting with t^* is the application of the rule $f(\alpha) \to \alpha$ which simply replaces the subterm $f(\alpha)$ in t^* by α. The resulting term is denoted by ς' in the proof. The next step in the derivation by $\overset{lo}{\leadsto}_{\mathcal{R}}$ starting with $equ(t_1, t_2)$ is the application of the same rule. □

Furthermore, the $eqapp(\varsigma)$ derivation steps by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$ work only on occurrences that are less with respect to $<_{lex}$ than the occurrence where the function is applied. Thus, there exist $\varsigma_t', \varsigma_s' \in T\langle F \cup \Delta\rangle(\mathcal{V})$, $\varphi' \in Sub(\mathcal{V}, \Delta)$, and there exists a derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k-eqapp(\varsigma)} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (equ(\varsigma_t', \varsigma_s'), \varphi') \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\varsigma')} (\varsigma', \psi').$$

Then we obtain the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k+1-eqapp(\varsigma')} (equ(\varsigma_t', \varsigma_s'), \varphi') \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\varsigma')} (\varsigma', \psi').$$

<u>*Case 2* :</u> $eqapp(\varsigma') = eqapp(\varsigma) + 1$. One of the cases 1 and 2 in Definition 4.3 is applied in the added step. In these cases $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}}$ exactly works as $\overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}$ (e.g., suppose that the subterm $f(\alpha)$ is replaced by $\sigma(\alpha, \alpha)$ in Figure 8, then the decomposition-rule for σ is applied in the added step.). We get the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k-eqapp(\varsigma)} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\varsigma)} (\varsigma, \psi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)} (\varsigma', \psi').$$

From $eqapp(\varsigma') = eqapp(\varsigma) + 1$ follows:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k-(eqapp(\varsigma')-1)} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\varsigma')} (\varsigma', \psi').$$

Here we obtain the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k+1-eqapp(\varsigma')} (equ(\varsigma_t, \varsigma_s), \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\varsigma')} (\varsigma', \psi').$$

Especially, if k is equal to the length of the derivation by $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}}$ in derivation 2, it follows that for every derivation 2, there exists a derivation 1. This finishes the proof of Claim 2. $\oplus$

The unification-driven leftmost outermost narrowing tree of $\mathcal{R}_1$ for the $E_{\mathcal{R}_1}$-unification of the terms $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$ is shown in Figure 9. At leaves which are labeled by 'clash!', the derivations are stopped by the ulo narrowing relation. Thus, in an intuitive sense, the uu-algorithm induced in Theorem 4.7 is more efficient than the uu-algorithm of Theorem 3.7. (Compare the ulo narrowing tree of Figure 9 with the leftmost outermost narrowing tree in Figure 4.)

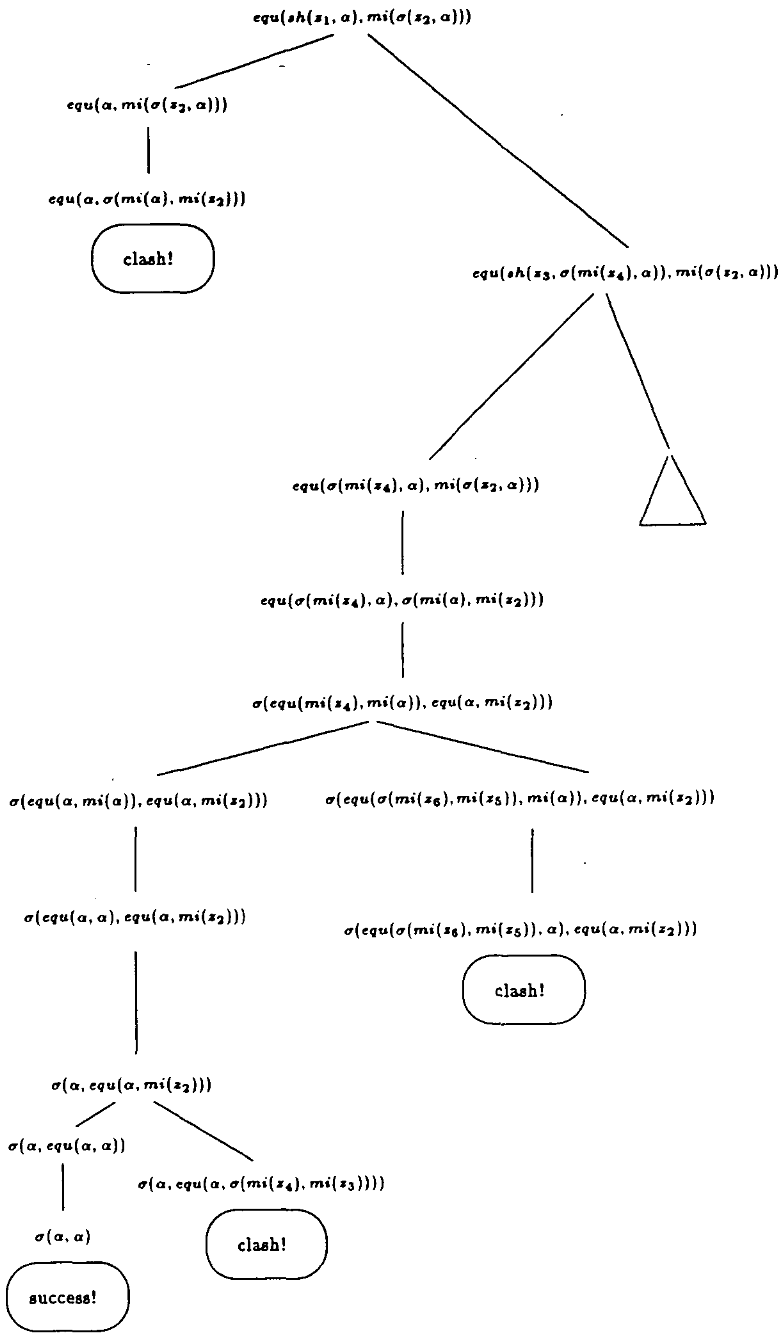

Figure 9: Ulo narrowing tree for $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$.

5 Conclusion

In this paper we have formalized a universal unification algorithm for equational theories which are characterized by ctn-trs's. This algorithm is at least as efficient as the algorithm which is implied by Theorem 3 in [3], but sometimes it is more efficient. The universal unification algorithm is based on the unification-driven leftmost outermost narrowing relation which is a combination of leftmost outermost narrowing and unification. It is inspired by the idea of the uu-algorithm in [11] which combines every innermost narrowing strategy with interleaving decomposition steps. The advantage of our uu-algorithm in comparison to the uu-algorithm in [11] is that arguments of function calls are only evaluated on demand which leads to a more efficient algorithm.

The conditions that the considered trs's are canonical and not strictly subunifiable, cannot be weakened, because the uu-algorithm would loose its completeness. Furthermore, the condition that the trs's are constructor-based, cannot be weakened. Otherwise, the decomposition-rules would make no sense. We are not sure, whether the condition that the trs's are totally-defined, can be weakened.

As mentioned in the introduction, there exist a lot of other uu-algorithms which are based on narrowing strategies. But none of them combines the narrowing strategy with interleaving decomposition steps. Thus, nonsuccessful derivations are computed up to the end, whereas they are immediately stopped in our algorithm.

Two implementations of leftmost outermost reduction for special ctn-trs's which are called macro tree transducers [1,4,5], are formalized in [8,13]. A nondeterministic implementation of the universal unification algorithm of the present paper which is an extension of the implementation in [8] by adding features for unification, is presented in [7]. In our current research [9] we construct a deterministic implementation of the universal unification algorithm by adding features for unification and backtracking to the implementation of leftmost outermost reduction shown in [13]. In the deterministic implementation a depth-first left-to-right traversal over the ulo narrowing tree is formalized. Clearly, this implementation does not produce a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers, because otherwise the $(E_{\mathcal{R}}, \Delta)$-unification problem would be decidable. Rather there are three possibilities:

- The machine stops and it has computed one $(E_{\mathcal{R}}, \Delta)$-unifier.
- The machine does not stop.
- The machine stops and it has computed no $(E_{\mathcal{R}}, \Delta)$-unifier. In fact, in this situation, the tree traversal has returned to the root and it is clear that there is no $(E_{\mathcal{R}}, \Delta)$-unifier at all.

As further research investigation, we will generalize the scope of this implementation from macro tree transducers to modular tree transducers [6]. Modular tree transducers are ctn-trs's which compute exactly the class of primitive recursive tree functions.

Acknowledgements

The authors would like to thank Alfons Geser for discussions on preliminary versions. The authors are grateful to the referee for communicating related work to us and for suggesting improvements of a previous version.

References

[1] B. Courcelle and P. Franchi-Zannettacci. Attribute grammars and recursive program schemes. *Theoret. Comput. Sci.*, 17:163–191 and 235–257, 1982.

[2] N. Dershowitz and J.P. Jouannaud. Notations for rewriting. *Bulletin of the EATCS*, 43:162–172, 1991.

[3] R. Echahed. On completeness of narrowing strategies. *CAAP 88, LNCS 299, 89-101*, 1988.

[4] J. Engelfriet. Some open questions and recent results on tree transducers and tree languages. In R.V. Book, editor, *Formal language theory; perspectives and open problems*. New York, Academic Press, 1980.

[5] J. Engelfriet and H. Vogler. Macro tree transducers. *J. Comput. System Sci.*, 31:71–146, 1985.

[6] J. Engelfriet and H. Vogler. Modular tree transducers. *Theoret. Comput. Sci.*, 78:267–304, 1991.

[7] H. Faßbender. Implementation of a universal unification algorithm for macro tree transducers. In *FCT'93*, pages 222–233. Springer-Verlag, 1993. LNCS 710.

[8] H. Faßbender and H. Vogler. An implementation of syntax directed functional programming on nested-stack machines. *Formal Aspects of Computing*, 4:341–375, 1992.

[9] H. Faßbender, H. Vogler, and A. Wedel. Implementation of a partial E–Unification algorithm for macro tree transducers. Technical report, University of Ulm, 1994. in preparation.

[10] M. Fay. First-order unification in an equational theory. In *Proceeding of the 4th workshop on automated deduction, Austin*, pages 161–167, 1979.

[11] L. Fribourg. Slog: A logic programming language interpreter based on clausual superposition and rewriting. In *Proceedings of the IEEE International Symposium on logic programming*, pages 172–184. IEEE Computer Society Press, 1985.

[12] J.H. Gallier and W. Snyder. A general complete E-unification procedure. In P. Lescanne, editor, *Rewriting techniques and applications RTA 87, LNCS 256*, pages 216–227, 1987.

[13] K. Gladitz, H. Faßbender, and H. Vogler. Compiler-based implementation of syntax directed functional programming. Technical Report 91-10, Technical University of Aachen, 1991.

[14] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comput. Mach.*, 27:797–821, 1980.

[15] G. Huet and D.C. Oppen. Equations and rewrite rules: a survey. In R. Book, editor, *Formal Language Theory: Perspectives and Open Problems*. Academic Press, New York, 1980.

[16] J.M. Hullot. Canonical forms and unification. In *Proceedings of the 5th conference on automated deduction, LNCS 87*, pages 318–334. Springer-Verlag, 1980.

[17] U. Hupbach. Rekursive Funktionen in mehrsortigen Algebren. *Elektron. Informationsverarb. Kybernetik*, 15:491–506, 1978.

[18] J. Jouannaud and H. Kirchner. Solving equations in abstract algebras: a rule-based survey of unification. In *Computational Logic. Essays in the honour of Alan Robinson*, pages 257–321. MIT Press, Cambridge, 1991.

[19] C. Kirchner. A new equational unification method: a generalisation of Martelli-Montanari's algorithm. In *Conference on Automated Deduction*, pages 224–247. Springer-Verlag, 1984. LNCS 170.

[20] K. Knight. Unification: a multidisciplinary survey. *ACM Computing Surveys*, 21:93–124, 1989.

[21] D.S. Lankford. Canonical inference. Technical Report ATP-32, Department of Mathematics and Computer Science, University of Texas, 1975.

[22] R. Loogen, F. Lopez-Fraguas, and M. Rodriguez-Artalejo. A demand driven computation strategy for lazy narrowing. In *PLILP'93*, pages 184–200, 1993. LNCS 714.

[23] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages Systems*, 4:258–282, 1982.

[24] A. Middeldorp and E. Hamoen. Counterexamples to completeness results for basic narrowing. In H. Kirchner and G. Levi, editors, *Algebraic and Logic Programming*, pages 244–258. Springer-Verlag, 1992. LNCS 632.

[25] P. Padawitz. Strategy-controlled reduction and narrowing. In P. Lescanne, editor, *Rewriting Techniques and Applications*, pages 242–255. Springer-Verlag, 1987. LNCS 256.

[26] U. S. Reddy. Narrowing as the operational semantics of functional languages. *IEEE Comp. Soc. Press 1985, Symp. Log. Progr.*, 1985.

[27] J.A. Robinson. A machine-oriented logic based on the resolution principle. *J. Assoc. Comput. Mach.*, 20:23–41, 1965.

[28] J. H. Siekmann. Unification theory. *J. Symbolic Computation*, 7:207–274, 1989.

[29] J.H. You. Solving equations in an equational language. In *Conference on algebraic and logic programming*, pages 245–254. Springer-Verlag, 1988. LNCS 343.

[30] J.H. You. Enumerating outer narrowing derivations for constructor-based term rewriting systems. *J. Symbolic Computation 7 (1989), 319-341*, 1989.

Received March 5, 1993

Revised September 1, 1993

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

Radical Theory for Group Semiautomata

Y. Fong* F.K. Huang* R. Wiegandt†

Abstract

A Kurosh-Amitsur radical theory is developed for group semiautomata. Radical theory stems from ring theory, it is apt for deriving structure theorems and for a comparative study of properties. Unlikely to conventional radical theories, the radical of a group semiautomaton need not be a subsemiautomaton, so the whole scene will take place in a suitably constructed category. The fundamental facts of the theory are described in § 2. A special feature of the theory, the existence of complementary radicals, is discussed in § 3. Restricting the theory to additive automata, which still comprise linear sequential machines, in § 4 stronger results will be achieved, and also a (sub)direct decomposition theorem for certain semisimple group semiautomata will be proved. Examples are given at appropriate places. The paper may serve also as a framework for future structural investigations of group semiautomata.

Key Words : Kurosh-Amitsur radical, group semiautomaton.

0 Introduction

The purpose of this paper is to develop a Kurosh-Amitsur radical theory for group semiautomata which may serve as a framework for future radical theoretical investigations and for describing the structure of semisimple group semiautomata.

In the variety of group semiautomata there is a one to one correspondence between homomorphisms and kernels, so it is meaningful to designate a kernel of a group semiautomaton as its radical. Doing so, however, there is an obstacle : a kernel is not always a subsemiautomaton, but only a normal subgroup subject to some additional requirement. This shortcoming can be overcome, if we work in an appropriately constructed category comprising group semiautomata and groups as objects. In this way kernels can be considered as subobjects.

The category suitable for a radical theory of group semiautomata will be constructed in § 1 analogously as done for semifields in [12]. Following the framework of [8], the fundamental notions of radical theory along with their characterizations, are given in § 2 in a self-contained way. A special feature of the radical theory of group semiautomata is the existence of complementary radical and semisimple classes which are discussed in § 3. Restricting the investigations to additive group

*Department of Mathematics, National Cheng-Kung University Tainan, 70101, Taiwan, R.O.C.

†Mathematical Institute, Hungarian Academy of Sciences P.O. Box 127, H-1364 Budapest, Hungary

semiautomata introduced in [4], we can get more explicit results. We shall see in § 4 that semisimple classes of additive group semiautomata are always hereditary, and we shall prove a subdirect decomposition theorem for additive group semiautomata which are semisimple with respect to a certain radical. Examples are supplied at appropriate places.

1 Preliminaries

A *group semiautomaton* (for short, a GS-*automaton*) is a quadruple $(\mathrm{A},+,\mathrm{X},\delta)$ consisting of an additive (not necessarily commutative) group $(\mathrm{A},+)$ as a set of states, of an input set $\mathrm{X}\neq\emptyset$ and a state transition function $\delta:\mathrm{A}\times\mathrm{X}\to\mathrm{A}$. The input set X, as usual, can be extended to the free monoid X^* over X, and then it is required that the transition function δ satisfies

$$\delta(a,xy)=\delta(\delta(a,x),y)$$

for all $x,y\in\mathrm{X}^*$.

The notion of GS-automaton is a generalization of that of linear sequential machine [3] or linear sequential automaton [1], and has been investigated, for instance, in [5], [6] (cf. also [9]).

In terms of universal algebra a GS-automaton is nothing but a universal algebra (A,Ω) with underlying set A and a set of operations $\Omega=\{+\}\cup\delta$ where $+$ is a binary operation making $(\mathrm{A},+)$ a group and δ consists of unary operations $f_x:a\to\delta(a,x)$, for all $a\in\mathrm{A}$ and $x\in\mathrm{X}$. Hence we know that GS-automata over a fixed input set X form a variety, and it is clear what a subsemiautomaton, a homomorphic image, an isomorphism, a direct or subdirect sum, a subdirectly irreducible GS-automaton, etc. means. Also the meaning of the homomorphism theorem and of the isomorphism theorems is obvious.

Throughout this paper the set X of inputs will be fixed, or equivalently, the set δ of unary operations will be a given one, and so a GS-automaton on the set A of states will be denoted by $(\mathrm{A},+,\delta)$, or sometimes briefly by A, if there is no fear of ambiguity. Moreover, for the clumsy notation $\delta(a,x)$ we shall write simply ax.

A congruence relation κ of a GS-automaton $(\mathrm{A},+,\delta)$ is a congruence on the group $(\mathrm{A},+)$, and therefore κ determines uniquely the coset K containing 0, which is a normal subgroup of $(\mathrm{A},+)$. Since κ is a congruence of the GS-automaton $(\mathrm{A},+,\delta)$, κ is compatible with the unary operations $f_x\in\delta, x\in\mathrm{X}$, that is, $f_x(a+k)$ is congruent to $f_x(a)$ modulo κ, that is,

$$(\star)\qquad (a+k)x-ax\in\mathrm{K}$$

for every $x\in\mathrm{X}, a\in\mathrm{A}$ and $k\in\mathrm{K}$. Conversely, if K is a normal subgroup of $(\mathrm{A},+)$ and satisfies condition $(\star)$, then the equivalence relation κ defined by K on the set A is a congruence on $(\mathrm{A},+,\delta)$. Thus by the homomorphism theorem every homomorphism

$$\varphi:(\mathrm{A},+,\delta)\to(\mathrm{B},+,\delta)$$

has a *kernel* K which is precisely a normal subgroup of $(\mathrm{A},+)$ subject to the requirement $(\star)$.

Let us observe a fact of importance for our investigations. *A kernel of a GS-automaton need not be a subsemiautomaton*, and a subsemiautomaton $(\mathrm{B},+,\delta)$ of a GS-automaton $(\mathrm{A},+,\delta)$ with normal subgroup $(\mathrm{B},+)$ in $(\mathrm{A},+)$, is not necessarily a kernel.

PROPOSITION 1.1. *A kernel* K *is a subsemiautomaton if and only if* $0X \subseteq K$. *If* K *contains a subsemiautomaton, then* K *itself is a subsemiautomaton.*

PROOF: Since

$$kx - 0x = (0 + k)x - 0x \in K$$

holds for arbitrary elements $k \in K$ and $x \in X$, the assertion follows. The second statement is now clear. □

EXAMPLE 1.2. A subsemiautomaton $(B, +, \delta)$ of a GS-automaton $(A, +, \delta)$ need not be a kernel even if $(B, +)$ is normal in $(A, +)$. Let us consider, namely, the Klein 4-group $(A, +) = \{0, a, b, c\}$ as the set of states and $X = \{x\}$ as the set of inputs. Define δ by the following graph

$$c \xrightarrow{x} b \xrightarrow{x} a \xrightarrow{x} 0 \xrightarrow{x} 0.$$

It can be easily seen that $\{0, a\}$, forms a subsemiautomaton (which is trivially a normal subgroup in A with $0X = 0 \in \{0, a\}$), but it is not a kernel, for

$$(b + a)x - bx = cx - bx = b - a = c \notin \{0, a\}.$$

The fact that there is a one-to-one correspondence between kernels and homomorphisms of GS-automata, but kernels are, in general, not subsemiautomata, adds a special flavor to the radical theory of GS-automata. A similar situation occurs also in the case of semifields [7], for which a radical theory has been developed in a category (universal class) comprising semifields and groups as objects [12]. In setting the scene we shall employ ideas of [12] and follow the framework of the Kurosh-Amitsur radical theory as developed in [8]. Thus we shall work in a universal class $\mathfrak{H}$ of GS-automata and groups, and it is our purpose in this note to develop a Kurosh-Amitsur radical theory in $\mathfrak{H}$ yielding specific results for GS-automata. Due to the high level of generality in [8], the adaptation of the results of [8] to our case is not quite straightforward, therefore for the sake of understandability and clarity we shall present the Kurosh-Amitsur radical theory of GS-automata in a self-contained way, though following the pattern of [8] and using ideas of [12].

Our investigations will take place within a suitable category $\mathfrak{C}$, the objects thereof are GS-automata and groups. Let $\mathfrak{A}$ denote the class of all GS-automata over a fixed input set X and $\mathfrak{G}$ the class of all groups, and we set $\mathrm{Ob}\,\mathfrak{C} = \mathfrak{A} \cup \mathfrak{G}$. For all $A, B \in \mathfrak{A} \cup \mathfrak{G}$ we consider the following three types of morphisms $\varphi\colon A \to B$:

1) All GS-automaton homomorphisms $\varphi\colon (A, +, \delta) \to (B, +, \delta)$ for $A, B \in \mathfrak{A}$.
2) All group homomorphisms $\varphi\colon (A, +) \to (B, +)$ for $A, B \in \mathfrak{G}$.
3) All group homomorphisms $\varphi\colon (A, +) \to (B, +, \delta)$ for $A \in \mathfrak{G}$ and $B \in \mathfrak{A}$ where one does not care about the transition function δ (or equivalently, about the unary operations $f_x \in \delta, x \in X$) defined on B.

The morphisms of types 1), 2) and 3) will consititute the morphisms of $\mathfrak{C}$. It is clear that $\mathfrak{C}$ has become a category. Designating the subclass

$$\mathcal{E} = \{\text{all surjective morphisms of types 1) and 2) in } \mathfrak{C}\}$$

and

$$\mathcal{M} = \{\text{all injective morphisms in } \mathfrak{C}\},$$

both $\mathcal{E}$ and $\mathcal{M}$, along with the objects of $\mathfrak{C}$, form obviously subcategories in $\mathfrak{C}$. Moreover, $\mathcal{E}$ and $\mathcal{M}$ consist of epimorphisms and monomorphisms, respectively, and $\mathcal{E} \cap \mathcal{M}$ is the class of all isomorphisms in $\mathfrak{C}$. Every morphism $\varphi: A \to B$ in $\mathfrak{C}$ factors as

$$A \xrightarrow{\varphi} B = A \xrightarrow{\varepsilon} C \xrightarrow{\mu} B$$

where $\varepsilon \in \mathcal{E}$ and $\mu \in \mathcal{M}$. Thus $\mathfrak{C}$ is endowed with a bicategory structure.

For developing a radical theory, it is sufficient and sometimes also useful to restrict the investigations to a certain subcategory of $\mathfrak{C}$. A non-empty subcategory $\mathfrak{H}$ of $\mathfrak{C}$ is called a *universal class*, if $\mathfrak{H}$ satisfies the following conditions :

(i) $\mathfrak{H}$ is closed with respect to all surjective morphisms $\varphi: A \to B$ of types 1) and 2).

(ii) $\mathfrak{H}$ is closed under taking kernels : for any morphisms $\varphi: A \to B$ in $\mathfrak{H}$ also $K = \ker \varphi$ is in $\mathfrak{H}$, (or equivalently, if K is a kernel in $A \in \mathfrak{H}$, then also $K \in \mathfrak{H}$).

(iii) $(A, +, \delta) \in \mathfrak{H}$ implies $(A, +) \in \mathfrak{H}$.

Concerning the universal class $\mathfrak{H}$ we shall work with, we make some observations.

1. The identical mapping ι of the set of states A induces a bijection $\iota: (A, +) \to (A, +, \delta)$ which is not an isomorphism, for its inverse does not exist in $\mathfrak{C}$ (in fact, it is not defined).

2. $\mathfrak{H}$ contains an initial object $(0, +)$ and a terminal object $(0, +, \delta)$ whenever $\mathfrak{H} \cap \mathfrak{A} \neq \emptyset$. We call $(0, +)$ and $(0, +, \delta)$ the *trivial objects* of $\mathfrak{H}$, and we shall write $\mathfrak{T}$ for the class of trivial objects. Since $(0, +)$ and $(0, +, \delta)$ are not isomorphic, in view of [11] we can predict a peculiar feature of the radical theory of GS-automata, and that is the existence of non-trivial complementary radical and semisimple classes (cf. § 3).

3. If $(A, +, \delta) \in \mathfrak{H}$ and $\varphi: A \to B$ is a morphism, then $K = \ker \varphi$ is either a subsemiautomaton $(K, +, \delta)$ (this is the case whenever K is a subsemiautomaton) or a normal subgroup $(K, +)$ (this is the case when K is not a subsemiautomaton). In the first case $(K, +)$ is a subobject of $(A, +, \delta)$ which is contained in the subobject $(K, +, \delta)$, but they are not equivalent subobjects.

4. The image of a kernel need not be a kernel. For instance, let $(K, +)$ be a kernel of a group $(A, +)$ and

$$\iota: (A, +) \to (A, +, \delta)$$

the identical embedding. Since $(K, +)$ is merely a normal subgroup of $(A, +)$,

$$\iota(K, +) = \begin{cases} (K, +) & \text{if K is not a subsemiautomaton,} \\ (K, +, \delta) & \text{if K is a subsemiautomaton,} \end{cases}$$

but $\iota(K, +)$ need not be a kernel of $\iota(A, +) = (A, +, \delta)$, regardless as whether it is a subsemiautomaton or not (cf. EXAMPLE 1.2).

5. We have to be careful in applying the second isomorphism theorem in $\mathfrak{H}$. Let $(L, +)$ be a subgroup of $(A, +)$ in a GS-automaton $(A, +, \delta)$. If K is a kernel of $(A, +, \delta)$, then $L/(L \cap K)$ is only a group, although $L + K$ may be a subsemiautomaton, for instance, if L is a kernel of $(A, +, \delta)$ and K is also a subsemiautomaton. In this case we have

$$(L/(L \cap K), +) \cong ((L + K)/K, +) \xrightarrow{\iota} ((L + K)/K, +, \delta)$$

and the left hand side is not isomorphic to the right hand side.

6. In the category $\mathfrak{C}$ (and therefore also in $\mathfrak{H}$) direct sums, in general, do not exist; more precisely, the (complete) direct sum $\sum^{\oplus} A_\alpha$ of objects $A_\alpha, \alpha \in \Lambda$, exists in $\mathfrak{C}$ if and only if either all A_α are GS-automata, or all of them are groups.

Kernels of an object A of $\mathfrak{H}$ form clearly a complete lattice isomorphic to the lattice of congruences of A. Unions and intersections in the lattice of kernels will be denoted by $\vee$ and $\wedge$, respectively. As usual, $\vee$ over the empty set and $\wedge$ over the empty set in the lattice of kernels of an object A, will mean the trivial kernel of A and A itself, respectively.

PROPOSITION 1.3. *If* K *and* L *are kernels of a* GS*-automaton* $(A, +, \delta)$, *then either* $K \vee L = (K + L, +)$ *or* $K \vee L = (K + L, +, \delta)$. *In particular, if* K *is a subsemiautomaton, then* $K \vee L = (K + L, +, \delta)$.

PROOF: $K + L$ is obviously a normal subgroup in A. Let $a \in A, k + l \in K + L$ and $x \in X$ be arbitrary elements. Then

$$(a + k + l)x - ax = (a + k + l)x - (a + k)x + (a + k)x - ax \in K + L$$

holds proving the first assertion. Hence in view of PROPOSITION 1.1 the second statement follows. □

2 Radical operator, radical class, semisimple class

In this section we fix a universal class $\mathfrak{H}$. Whenever we consider a subclass C of objects of $\mathfrak{H}$, we suppose that C is an abstract class (that is, C is closed under isomorphisms) and that $\mathfrak{T} \subseteq$ C. Moreover, we introduce the following notation :

A—▹B means a nonzero surjective morphism of type 1) or 2),
K ◃ A means that K is a nonzero kernel of A.

In the sequel we are going to give the fundamental definitions and characterizations of radical theory in a self-contained way for GS-automata. Further results can be proven in a similar way as in [12] or can be derived from [8].

An operator ϱ which assigns to each object $A \in \mathfrak{H}$ a kernel ϱA of A is called a *radical operator*, if ϱ satisfies the following set of conditions for all $A, B \in \mathfrak{H}$:

(ϱa) if $\varphi \colon A \to B$ is a surjective morphism, then $\varphi(\varrho A) \subseteq \varrho B$ holds,
(ϱb) $|\varrho(A/\varrho A)| = 1$,
(ϱc) if $\varrho B = B \triangleleft A$, then $B \subseteq \varrho A$,
(ϱd) $\varrho\varrho A = \varrho A$.

PROPOSITION 2.1. *Let* ϱ *be a radical operator. The class*

$$\mathbf{R}_\varrho = \{A \in \mathfrak{H} \mid \varrho A = A\}$$

fulfils the following conditions for all $A, B \in \mathfrak{H}$:

(Ra) *if* $A \in \mathbf{R}_\varrho$, *then for every* A—▹B *there exists a* $K \triangleleft B$ *with* $K \in \mathbf{R}_\varrho$,
(Rb) *if* $A \in \mathfrak{H}$ *and for every* A—▹B *there exists a* $K \triangleleft B$ *with* $K \in \mathbf{R}_\varrho$, *then*

$A \in \mathbf{R}_\varrho$,

(Rk) *if* $(A, +, \delta) \in \mathfrak{H}$ *and there exists a* $K \triangleleft (A, +)$ *such that* $K \in \mathbf{R}_\varrho$, *then there exists an* $L \triangleleft (A, +, \delta)$ *with* $L \in \mathbf{R}_\varrho$.

PROOF: Let $A \in \mathbf{R}_\varrho$ and $\varphi: A \twoheadrightarrow B$ be arbitrarily chosen. By (ϱa) we have

$$B = \varphi(A) = \varphi(\varrho A) \subseteq \varrho\varphi(A) = \varrho B \subseteq B,$$

and hence $B \in \mathbf{R}_\varrho$. Thus (Ra) is trivially satisfied.

Let $A \in \mathfrak{H} \setminus \mathfrak{T}$ be an object such that for each $A \twoheadrightarrow B$ there exists a $K \triangleleft B$ with $K \in \mathbf{R}_\varrho$. If $A \notin \mathbf{R}_\varrho$, then $\varrho A \neq A$, and so for $B = A/\varrho A$ we have $|B| > 1$. By the hypothesis there exists a $K \triangleleft B$ such that $\varrho K = K$, and hence (ϱc) yields $K \subseteq \varrho B$. Thus we have got

$$1 < |K| \leq |\varrho B| = |\varrho(A/\varrho A)|$$

contradicting (ϱb). Consequently $A \in \mathbf{R}_\varrho$, proving (Rb).

Finally, let us suppose that $(A, +, \delta) \in \mathfrak{H}$ is a GS-automaton such that $K \triangleleft (A, +)$ with some $K \in \mathbf{R}_\varrho$. Then (ϱc) yields $K \subseteq \varrho(A, +)$. Further, for the morphism $\iota: (A, +) \to (A, +, \delta)$ in view of (ϱa) we get

$$\iota(K) \subseteq \iota(\varrho(A, +)) \subseteq \varrho(\iota(A, +)) = \varrho(A, +, \delta),$$

and so

$$1 < |K| = |\iota(K)| \leq |\varrho(A, +, \delta)|$$

holds. Since by (ϱd) we have also $\varrho(A, +, \delta) \in \mathbf{R}_\varrho$, the validity of condition (Rk) has been established. $\square$

PROPOSITION 2.2. *If a subclass* $\mathbf{R}$ *of* $\mathfrak{H}$ *satisfies conditions* (Ra), (Rb), (Rk), *then* $\mathbf{R}$ *fulfils also the following ones :*

- (Rh) *the class* $\mathbf{R}$ *is homomorphically closed* : if $A \in \mathbf{R}$ and $\varphi: A \twoheadrightarrow B$, then $B \in \mathbf{R}$,
- (Rc) *if* $(A, +, \delta) \in \mathfrak{H}$ *and* $(A, +) \in \mathbf{R}$, *then* $(A, +, \delta) \in \mathbf{R}$,
- (Re) *the class* $\mathbf{R}$ *is closed under extensions* : if $K \triangleleft A \in \mathfrak{H}$, $K \in \mathbf{R}$ and $A/K \in \mathbf{R}$, then $A \in \mathbf{R}$,
- (Ri) *the class* $\mathbf{R}$ *has the inductive property* : if $K_1 \subseteq \cdots \subseteq K_\alpha \subseteq \cdots$ is any ascending chain of kernels of an object $A \in \mathfrak{H}$ such that $K_\alpha \in \mathbf{R}$ for each index α, then $\vee K_\alpha \in \mathbf{R}$,
- (Rt) $\mathfrak{T} \subseteq \mathbf{R}$.

PROOF: Let $A \in \mathbf{R}$ and $\varphi: A \twoheadrightarrow B$, and let us consider an arbitrary $\psi: B \twoheadrightarrow C$. Then also $\psi\varphi: A \twoheadrightarrow C$ holds, and so by (Ra) there exists a $K \triangleleft C$ with $K \in \mathbf{R}$. Hence (Rb) is applicable on B yielding $B \in \mathbf{R}$. This proves (Rh).

Let $(A, +, \delta)$ be a GS-automaton in $\mathfrak{H}$ such that $(A, +) \in \mathbf{R}$, and K be an arbitrary kernel of $(A, +, \delta)$ with $K \neq (A, +, \delta)$. Then we have

$$(A/K, +) \xrightarrow{\iota} (A/K, +, \delta)$$

and also $(A/K, +) \in \mathbf{R} \setminus \mathfrak{T}$ in view of (Rh). Hence (Rk) infers the existence of a kernel L of $(A/K, +, \delta)$ such that $L \in \mathbf{R} \setminus \mathfrak{T}$. Since the choice of K was arbitrary, by (Rb) we conclude $(A, +, \delta) \in \mathbf{R}$, proving the validity of (Rc).

For proving $(\mathrm{R}e)$, let L be an arbitrary nonzero kernel of A. We wish to apply $(\mathrm{R}b)$ on A. If $\mathrm{K} \subseteq \mathrm{L}$, then the isomorphism

$$\frac{\mathrm{A/K}}{\mathrm{L/K}} \cong \mathrm{A/L}$$

and the already demonstrated condition $(\mathrm{R}h)$ yield $\mathrm{A/L} \in \mathbf{R}$. If $\mathrm{K} \not\subseteq \mathrm{L}$, then $|\mathrm{K}/(\mathrm{L} \cap \mathrm{K})| > 1$ and again by $(\mathrm{R}h)$ also $\mathrm{K}/(\mathrm{L} \cap \mathrm{K}) \in \mathbf{R}$ is valid. Further, by PROPOSITION 1.3 we have

$$\mathrm{K}/(\mathrm{L} \cap \mathrm{K}) \cong (\mathrm{L} + \mathrm{K})/\mathrm{L} \xrightarrow{\iota} (\mathrm{L} \vee \mathrm{K})/\mathrm{L} \triangleleft \mathrm{A/L}$$

and so by $(\mathrm{R}c)$, if needed, also $(\mathrm{L} \vee \mathrm{K})/\mathrm{L} \in \mathbf{R}$ holds. Thus A/L possesses always a nonzero kernel in $\mathbf{R}$, and therefore $(\mathrm{R}b)$ infers $\mathrm{A} \in \mathbf{R}$. This proves $(\mathrm{R}e)$.

For demonstrating $(\mathrm{R}i)$, put $\mathrm{L} = \vee \mathrm{K}_\alpha$. If $\mathrm{L} \notin \mathbf{R}$, then in view of (Rb) there exists an $\mathrm{M} \triangleleft \mathrm{L}$ such that $|\mathrm{L/M}| > 1$ and L/M has no nonzero kernel in $\mathbf{R}$. Further, by $(\mathrm{R}h)$ we have $\mathrm{K}_\alpha/(\mathrm{M} \cap \mathrm{K}_\alpha) \in \mathbf{R}$ for each α. From PROPOSITION 1.3 we have

$$\mathrm{K}_\alpha/(\mathrm{M} \cap \mathrm{K}_\alpha) \cong (\mathrm{K}_\alpha + \mathrm{M})/\mathrm{M} \xrightarrow{\iota} (\mathrm{K}_\alpha \vee \mathrm{M})/\mathrm{M} \triangleleft \mathrm{L/M}$$

and so $(\mathrm{R}c)$ infers $(\mathrm{K}_\alpha \vee \mathrm{M})/\mathrm{M} \in \mathbf{R}$ for every α. Hence by the choice of M it follows $\mathrm{K}_\alpha \subseteq \mathrm{M}$ for every α, and so also $\mathrm{L} = \vee \mathrm{K}_\alpha \subseteq \mathrm{M}$, contradicting $|\mathrm{L/M}| > 1$.

$(\mathrm{R}t)$ is a trivial consequence of $(\mathrm{R}b)$. □

PROPOSITION 2.3. *Let a subclass* $\mathbf{R}$ *of* $\mathfrak{H}$ *satisfy conditions* $(\mathrm{R}h), (\mathrm{R}e), (\mathrm{R}i), (\mathrm{R}t), (\mathrm{R}k)$. *If the operator* ϱ *is defined as*

$$\varrho \mathrm{A} = \vee(\mathrm{K} \triangleleft \mathrm{A} \mid \mathrm{K} \in \mathbf{R}), \quad \forall \mathrm{A} \in \mathfrak{H},$$

then

i) $\varrho \mathrm{A} \in \mathbf{R}$, $\forall \mathrm{A} \in \mathfrak{H}$ *and* $\mathbf{R} = \{\mathrm{A} \in \mathfrak{H} \mid \varrho \mathrm{A} = \mathrm{A}\}$,

ii) ϱ *is a radical operator.*

PROOF: First we prove that $\mathbf{R}$ fulfils $(\mathrm{R}c)$. Suppose the contrary: there exists an automaton $(\mathrm{A}, +, \delta) \in \mathfrak{H} \setminus \mathbf{R}$ such that $(\mathrm{A}, +) \in \mathbf{R} \setminus \mathfrak{T}$. By $(\mathrm{R}i)$ and Zorn's Lemma there exists a kernel I of $(\mathrm{A}, +, \delta)$ such that $\mathrm{I} \in \mathbf{R}$ and I is maximal with respect to this property. Let us consider the automaton $\mathrm{A/I} = (\mathrm{A/I}, +, \delta)$. Since $\mathbf{R}$ has $(\mathrm{R}h)$, we have $(\mathrm{A/I}, +) \in \mathbf{R}$. Take any kernel L/I of $(\mathrm{A/I}, +\delta)$ such that $\mathrm{L/I} \in \mathbf{R}$. Then by $\mathrm{I} \in \mathbf{R}$ and $(\mathrm{R}e)$ we get $\mathrm{L} \in \mathbf{R}$. Hence the maximality of I gives us $\mathrm{L} = \mathrm{I}$. Thus there is no kernel L/K of $(\mathrm{A/I}, +, \delta)$ such that $|\mathrm{L/K}| > 1$ and $\mathrm{L/K} \in \mathbf{R}$. Applying $(\mathrm{R}k)$ we conclude that there is no kernel K/I of $(\mathrm{A/I}, +)$ such that $|\mathrm{K/I}| > 1$ and $\mathrm{K/I} \in \mathbf{R}$. This and $(\mathrm{A/I}, +) \in \mathbf{R}$ imply $\mathrm{A} = \mathrm{I} \in \mathbf{R}$, contradicting $\mathrm{A} \in \mathfrak{H} \setminus \mathbf{R}$. Thus (Rc) has been established.

Now we prove $\varrho \mathrm{A} \in \mathbf{R}$. By $(\mathrm{R}i)$ Zorn's Lemma is applicable yielding the existence of a kernel K of A being maximal with respect to $\mathrm{K} \in \mathbf{R}$. Let L be any other kernel of A with $\mathrm{L} \in \mathbf{R}$. By $(\mathrm{R}h)$ we have $\mathrm{L}/(\mathrm{L} \cap \mathrm{K}) \in \mathbf{R}$ and so in view of

$$\mathrm{L}/(\mathrm{L} \cap \mathrm{K}) \cong (\mathrm{L} + \mathrm{K})/\mathrm{K} \xrightarrow{\iota} (\mathrm{L} \vee \mathrm{K})/\mathrm{K}$$

condition $(\mathrm{R}c)$, if needed, yields $(\mathrm{L} \vee \mathrm{K})/\mathrm{K} \in \mathbf{R}$. Hence by condition $(\mathrm{R}e)$ we get $\mathrm{L} \vee \mathrm{K} \in \mathbf{R}$ which implies $\mathrm{L} \subseteq \mathrm{K}$ by the choice of K. Thus K is the unique kernel of A such that K is maximal with respect to $\mathrm{K} \in \mathbf{R}$. This means exactly $\varrho \mathrm{A} = \mathrm{K} \in \mathbf{R}$.

Now the assertion that $\mathbf{R} = \{\mathrm{A} \in \mathfrak{H} \mid \varrho \mathrm{A} = \mathrm{A}\}$ is obviously true.

For proving that ϱ is a radical operator, we notice that (ϱc) and (ϱd) are clearly satisfied, both by $\varrho A \in \mathbf{R}$.

Next we exhibit (ϱb). Let $L/\varrho A \triangleleft A/\varrho A$ and $L/\varrho A \in \mathbf{R}$. As we have already seen, $\varrho A \in \mathbf{R}$, hence condition (Re) implies $L \in \mathbf{R}$. Thus by the definition of ϱA we conclude $L \subseteq \varrho A$ which implies $|L/\varrho A| = 1$ as well as $|\varrho(A/\varrho A)| = 1$.

For demonstrating (ϱa) it suffices to exhibit its validity for morphisms $\psi\colon A \longrightarrow\!\!\triangleright B$ and $\iota\colon (A,+) \to (A,+,\delta)$ because every surjective morphism φ is a composition of such morphisms or $|A| = 1$, and this latter case is covered by condition (Rt). For any morphism $\psi\colon A \longrightarrow\!\!\triangleright B$ we have $\psi(K) \triangleleft B$ or $|\psi(K)| = 1$ whenever $K \triangleleft A$, in particular for $K = \varrho A$. Furthermore, also $\varrho A \in \mathbf{R}$ holds as we have seen, and so condition (Rh) infers $\psi(\varrho A) \in \mathbf{R}$. Thus by definition $\psi(\varrho A) \subseteq \varrho B$ holds. In the case $\iota\colon (A,+) \to (A,+,\delta)$, let us suppose that (ϱa) is not true, that is, $\iota(\varrho(A,+)) \not\subseteq \varrho(A,+,\delta)$. Then we have

$$|\varrho(A,+)/(\varrho(A,+) \cap \varrho(A,+,\delta))| > 1$$

and

$$\begin{aligned} \varrho(A,+)/(\varrho(A,+) \cap \varrho(A,+,\delta)) &\cong (\varrho(A,+) + \varrho(A,+,\delta))/\varrho(A,+,\delta) \\ &\triangleleft (A/\varrho(A,+,\delta),+). \end{aligned}$$

Moreover, condition (Rh) implies

$$\varrho(A,+)/(\varrho(A,+) \cap \varrho(A,+,\delta)) \in \mathbf{R}.$$

Hence condition (Rk) applies to $K = (\varrho(A,+) + \varrho(A,+,\delta))/\varrho(A,+,\delta)$ yielding the existence of an $L \triangleleft A/\varrho(A,+,\delta)$ with $L \in \mathbf{R}$. This, by $|L| > 1$, contradicts the already demostrated condition (ϱc). Thus $\iota(\varrho(A,+)) \subseteq \varrho(A,+,\delta)$ holds. □

A subclass $\mathbf{R}$ of $\mathfrak{H}$ is called a *radical class* if it satisfies condition (Ra), (Rb), (Rk). PROPOSITION 2.1, 2.2 and 2.3 can be summarized as follows

THEOREM 2.4. *Let ϱ be an operator assigning to each object $A \in \mathfrak{H}$ a kernel ϱA of A, and let $\mathbf{R}$ be a subclass of objects in $\mathfrak{H}$. Then the following three conditions are equivalent :*

1) *ϱ is a radical operator and $\mathbf{R}_\varrho = \mathbf{R}$,*
2) *$\mathbf{R}$ is a radical class and $\varrho A = \vee(K \triangleleft A \mid K \in \mathbf{R})$, $\forall A \in \mathfrak{H}$,*
3) *$\mathbf{R}$ satisfies conditions (Rh), (Re), (Ri), (Rk), (Rt) and $\varrho A = \vee(K \triangleleft A \mid K \in \mathbf{R})$, $\forall A \in \mathfrak{H}$.* □

Let ϱ be a radical operator. The class

$$\mathbf{S}_\varrho = \{A \in \mathfrak{H} \mid |\varrho A| = 1\}$$

is called the *semisimple* class of ϱ (or equivalently, of the radical class $\mathbf{R}_\varrho$). Obviously $\mathbf{R}_\varrho \cap \mathbf{S}_\varrho = \mathfrak{T}$ holds. It is useful introduce the *semisimple operator* $\mathcal{S}$ acting on subclasses $\mathbf{C}$ of objects of $\mathfrak{H}$ and defined by

$$\mathcal{S}\mathbf{C} = \{A \in \mathfrak{H} \mid K \triangleleft A \Rightarrow K \notin \mathbf{C}\}.$$

If ϱ is any radical operator and $\mathbf{R}_\varrho$ the corresponding radical class, then by THEOREM 2.4 we have

$$\mathbf{S}_\varrho = \mathcal{S}\mathbf{R}_\varrho$$

which justifies the terminology.

PROPOSITION 2.5. *If ϱ is a radical operator in $\mathfrak{H}$, then the semisimple class $\mathbf{S}_\varrho$ satisfies the following conditions :*

(S*a*) *if* $A \in \mathbf{S}_\varrho$, *then for every* $K \triangleleft A$ *there exists a* $K \longrightarrow\!\triangleright B$ *with* $B \in \mathbf{S}_\varrho$,

(S*b*) *if* $A \in \mathfrak{H}$ *and for every* $K \triangleleft A$ *there exists a* $K \longrightarrow\!\triangleright B$ *with* $B \in \mathbf{S}_\varrho$, *then* $A \in \mathbf{S}_\varrho$.

(S*c*) *if* $(A,+,\delta) \in \mathbf{S}_\varrho$, *then* $(A,+) \in \mathbf{S}_\varrho$.

PROOF: For exhibiting (S*a*), let us consider an object $A \in \mathbf{S}_\varrho = \mathcal{S}\mathbf{R}_\varrho$ and an arbitrary $K \triangleleft A$. Now we have $\varrho K \in \mathbf{R}_\varrho$, and so $|K/\varrho K| > 1$. Also $K/\varrho K \in \mathbf{S}_\varrho$ holds in view of (ϱb). Hence $K \longrightarrow\!\triangleright B \in \mathbf{S}_\varrho$ is satisfied with $B = K/\varrho K$.

Next, let us suppose that for every $K \triangleleft A$ there exists a $K \longrightarrow\!\triangleright B$ with $B \in \mathbf{S}_\varrho$, but $A \notin \mathbf{S}_\varrho$. Then $|\varrho A| > 1$. In particular, for $K = \varrho A$ there exists a $\varrho A \longrightarrow\!\triangleright C \in \mathbf{S}_\varrho$, and by ($\varrho a$) (or (R*h*)) we conclude also $\varrho C = C$ (or $C \in \mathbf{R}_\varrho$). Thus $C \in \mathbf{S}_\varrho \cap \mathbf{R}_\varrho = \mathfrak{T}$, contradicting $\varrho A \longrightarrow\!\triangleright C$. This proves the validity of (S*b*).

Finally, assume that $(A,+,\delta) \in \mathfrak{H}$ and $(A,+) \notin \mathbf{S}_\varrho$, that is, $\varrho(A,+) \triangleleft (A,+)$ and $\varrho(A,+) \in \mathbf{R}_\varrho$. By THEOREM 2.4 condition (R*k*) is applicable yielding the existence of an $L \triangleleft (A,+,\delta)$ with $L \in \mathbf{R}_\varrho$. Hence by (ϱb) it follows $L \subseteq \varrho(A,+,\delta)$ implying $(A,+,\delta) \notin \mathbf{S}_\varrho$. This proves (S*c*). □

For any subclass $\mathbf{C} \subseteq \mathfrak{H}$ we define an operator $\mathcal{U}$ as

$$\mathcal{U}\mathbf{C} = \{A \in \mathfrak{H} \mid A \longrightarrow\!\triangleright B \Rightarrow B \notin \mathbf{C}\}.$$

The operator $\mathcal{U}$, which is defined dually to the semisimple operator $\mathcal{S}$, is called the *upper radical operator.*

PROPOSITION 2.6. *If a subclass* $\mathbf{S} \subseteq \mathfrak{H}$ *satisfies conditions* (S*a*), (S*b*), (S*c*), *then* $\mathbf{R} = \mathcal{U}\mathbf{S}$ *is a radical class, and* $\mathbf{S} = \mathcal{S}\mathbf{R} = \mathbf{S}_\varrho$ *where ϱ denotes the radical operator correponding to the radical class* $\mathbf{R}$.

PROOF: Since the relation $\longrightarrow\!\triangleright$ is transitive, the class $\mathbf{R} = \mathcal{U}\mathbf{S}$ is homomorphically closed, that is, $\mathcal{U}\mathbf{S}$ satisfies (R*h*) and hence also the weaker condition (R*a*).

For demonstrating (R*b*), let us consider and object $A \in \mathfrak{H} \setminus \mathfrak{T}$ such that for every $A \longrightarrow\!\triangleright B$ there exists a $K \triangleleft B$ with $K \in \mathcal{U}\mathbf{S}$. If $A \notin \mathcal{U}\mathbf{S}$, then there exists an $A \longrightarrow\!\triangleright B$ with $B \in \mathbf{S}$ and by (S*a*) to every $K \triangleleft B$ there exists a $K \longrightarrow\!\triangleright C \in \mathbf{S}$, that is, $K \notin \mathcal{U}\mathbf{S}$. This contradicts the assumption on A, and so (R*b*) is satisfied. Let us notice that an object $A \in \mathfrak{T}$ trivially satisfies (R*b*).

Let $(A,+,\delta) \in \mathfrak{H}$ be an object such that $K \triangleleft (A,+)$ and $K \in \mathbf{R} = \mathcal{U}\mathbf{S}$ for some kernel of $(A,+)$. To prove (R*k*) we have to show that $(A,+,\delta) \notin \mathbf{S}$, because then by (S*b*) there exists an $L \triangleleft (A,+,\delta)$ such that $L \in \mathcal{U}\mathbf{S} = \mathbf{R}$, and this means exactly the validity of (R*k*). Suppose that $(A,+,\delta) \in \mathbf{S}$. Then by (S*c*) also $(A,+) \in \mathbf{S}$ is valid, and so by (S*a*) we have $K \longrightarrow\!\triangleright B \in \mathbf{S}$ for the kernel K of $(A,+)$ with an appropriate $B \in \mathfrak{H}$. This means $K \notin \mathcal{U}\mathbf{S}$, contradicting $K \in \mathcal{U}\mathbf{S}$. Thus (R*k*) has been established.

Since $\mathbf{R} = \mathcal{U}\mathbf{S}$ satisfies (R*a*), (R*b*) and (R*k*), by THEOREM 2.4 we conclude that $\mathbf{R}$ is a radical class.

As one readily checks, (S*a*) is equivalent to $\mathbf{S} \subseteq \mathcal{S}\mathcal{U}\mathbf{S}$ and (S*b*) is equivalent to $\mathcal{S}\mathcal{U}\mathbf{S} \subseteq \mathbf{S}$. Hence $\mathbf{S} = \mathcal{S}\mathbf{R}$ as well as $\mathbf{S} = \mathbf{S}_\varrho$ hold by the remark proceding PROPOSITION 2.5. □

PROPOSITIONS 2.5 and 2.6 infer immediately

COROLLARY 2.7. *A subclass* $\mathbf{S} \subseteq \mathfrak{H}$ *is the semisimple class of a radical class (or equivalently, of a radical operator) if and only if* $\mathbf{S}$ *satisfies conditions* (S*a*), (S*b*) *and* (S*c*).

For a subclass $\mathbf{S}$ of objects, let us define the operator η as

$$\eta A = \wedge(K \triangleleft A \mid A/K \in \mathbf{S})$$

which assigns to each $A \in \mathfrak{H}$ a kernel of A.

PROPOSITION 2.8. *If* $\mathbf{S}$ *is the semisimple class corresponding to a radical operator* ϱ, *then*

(Ss) $\mathbf{S}$ *is closed under subdirect sums :* $A = \sum_{\text{subdirect}} A_\alpha$ and $A_\alpha \in \mathbf{S}$ for all α imply $A \in \mathbf{S}$, or equivalently : $A/\eta A \in \mathbf{S}$,

(Sϱ) $\eta A = \varrho A$ *for all* $A \in \mathfrak{H}$,

(Sη) $\eta\eta A$ *is a kernel of* A *for all* $A \in \mathfrak{H}$,

(Se) $\mathbf{S}$ *is closed under extensions.*

PROOF: Firstly we prove (Ss). Let us consider an object $A \in \mathfrak{H}$ such that A is a subdirect sum of objects A_α, $\alpha \in \Lambda$, each in $\mathbf{S}$. Then there exists a set $\{K_\alpha \mid \alpha \in \Lambda\}$ of kernels of A such that $A/K_\alpha \cong A_\alpha \in \mathbf{S}$ and $|\wedge K_\alpha| = 1$. Let $L \triangleleft A$ be arbitrary. Now by $|L| > 1$ there exists an index α such that $L \not\subseteq K_\alpha$, and hence

$$(L \vee K_\alpha)/K_\alpha \triangleleft A/K_\alpha \in \mathbf{S}.$$

Thus, condition (Sa) infers the existence of an

$$(L \vee K_\alpha)/K_\alpha \twoheadrightarrow B \in \mathbf{S}.$$

Hence either

$$L \twoheadrightarrow L/(L \wedge K_\alpha) \cong (L \vee K_\alpha)/K_\alpha \twoheadrightarrow B \in \mathbf{S}$$

or by (Sc)

$$(L,+) \twoheadrightarrow (L/(L \wedge K_\alpha),+) \cong ((L + K_\alpha)/K_\alpha,+) \twoheadrightarrow (B,+) \in \mathbf{S}$$

yields $L \twoheadrightarrow C \in \mathbf{S}$ where $C = (B,+,\delta)$ or $(B,+)$. Hence by (Sb) we conclude that $A \in \mathbf{S}$, proving (Ss).

For demonstrating (Se), let us consider a $K \triangleleft A$ such that $K \in \mathbf{S}$ and $A/K \in \mathbf{S}$. Further, let $L \triangleleft A$ be arbitrary. If $L \subseteq K$, then by $L \triangleleft K$ and $K \in \mathbf{S}$ condition (Sa) implies the existence of an $L \twoheadrightarrow B \in \mathbf{S}$. If $L \not\subseteq K$, then we have

$$(L \vee K)/K \triangleleft A/K \in \mathbf{S},$$

and so by (Sa), $(L \vee K)/K \twoheadrightarrow B \in \mathbf{S}$ with an appropriate $B \in \mathfrak{H}$. The isomorphism

$$L/(L \wedge K) \cong (L + K)/K \xrightarrow{\iota} (L \vee K)/K$$

and condition (Sc), if necessary, infer $L \twoheadrightarrow C \in \mathbf{S}$ where either $C = (B,+,\delta)$ or $C = (B,+)$. Thus by (Sb) we obtain $A \in \mathbf{S}$ which proves (Sc).

Next, we are going to prove (Sϱ). By condition (ϱb) we have $|\varrho(A/\varrho A)| = 1$, and therefore $A/\varrho A \in \mathbf{S}_\varrho = \mathbf{S}$. Hence $\eta A \subseteq \varrho A$ holds by the definition of η. Suppose that $\eta A \neq \varrho A$. Then $\varrho A/\eta A \triangleleft A$ is valid as $|\varrho A/\eta A| > 1$. Moreover, by (ϱa) and (ϱd) we obtain

$$\varrho A/\eta A = \varrho\varrho A/\eta A \subseteq \varrho(\varrho A/\eta A),$$

yielding $\varrho A/\eta A \in \mathbf{R}_\varrho = \mathcal{U}\mathbf{S}$. Since $A/\eta A \in \mathbf{S}$ by (Ss), condition (Sa) applied to $\varrho A/\eta A \triangleleft A/\eta A$ yields the existence of a $\varrho A/\eta A \twoheadrightarrow C \in \mathbf{S}$, contradicting $\varrho A/\eta A \in \mathcal{U}\mathbf{S}$. Thus $\eta A = \varrho A$ has been proved.

Finally, condition (Sη) is a trivial consequence of $\eta A = \varrho A$ and condition (ϱd). □

PROPOSITION 2.9 . *Let* $\mathbf{S}$ *be a subclass of* $\mathfrak{H}$ *which fulfils conditions* (Sa), (Sc), (Se), (Ss), (St), (Sη). *Then* $\mathbf{S}$ *is a semisimple class.*

PROOF: In view of COROLLARY 2.7 all what we have to prove is the validity of condition (Sb). So, let us consider an object $A \in \mathfrak{H}$ such that for every $K \triangleleft A$ there exists a $K \longrightarrow\!\triangleright B$ with $B \in \mathbf{S}$. By way of contradiction, let us suppose that $A \notin \mathbf{S}$. Then by (Ss) we have $A/\eta A \in \mathbf{S}$, and so $|\eta A| > 1$, that is, $\eta A \triangleleft A$. By (Sη) also $\eta A/\eta\eta A \triangleleft A/\eta\eta A$ holds, and by (Ss) we have $\eta A/\eta\eta A \in \mathbf{S}$. Since

$$\frac{A/\eta\eta A}{\eta A/\eta\eta A} \cong A/\eta A \in \mathbf{S},$$

condition (Se) yields $A/\eta\eta A \in \mathbf{S}$ which implies $\eta A \subseteq \eta\eta A$. Thus by the definition of η, ηA has no non-zero isomorphic image in $\mathbf{S}$, contradicting (Sa). □

COROLLARY 2.10. *A subclass* $\mathbf{S}$ *of* $\mathfrak{H}$ *is a semisimple class if and only if* $\mathbf{S}$ *satisfies conditions* (Sa), (Sc), (Se), (Ss), (St) *and* (Sη). *Moreover, the operator* η *occuring in condition* (Sη) *is just the radical operator corresponding to the semisimple class* $\mathbf{S}$.

PROOF: Trivial by PROPOSITIONS 2.8 and 2.9. □

THEOREM 2.11. *The subclasses* $\mathbf{R}$ *and* $\mathbf{S}$ *are corresponding radical and semi-simple classes (that is,* $\mathbf{R} = \mathcal{U}\mathbf{S}$ *and* $\mathbf{S} = \mathcal{S}\mathbf{R}$*) if and only if*

a) $A \in \mathbf{R}$ *and* $A \longrightarrow\!\triangleright B$ *imply* $B \notin \mathbf{S}$, *that is,* $\mathbf{R} \subseteq \mathcal{U}\mathbf{S}$,
b) $A \in \mathbf{S}$ *and* $B \triangleleft A$ *imply* $B \notin \mathbf{R}$, *that is,* $\mathbf{S} \subseteq \mathcal{S}\mathbf{R}$,
c) for each $A \in \mathfrak{H}$ *there exists a kernel* K *of* A *such that* $K \in \mathbf{R}$ *and* $A/K \in \mathbf{S}$.
d) $\mathbf{S}$ *fulfils* (Sc) *or* $\mathbf{R}$ *satisfies* (Rk).

PROOF: We already know that these properties hold true for a radical class $\mathbf{R}$ with semisimple class $\mathbf{S} = \mathcal{S}\mathbf{R}$.

Conversely, we apply c) to each $A \in \mathcal{S}\mathbf{R}$. Since $A \in \mathcal{S}\mathbf{R}$ implies $B \notin \mathbf{R}$ for all $B \triangleleft A$, necessarily $|K| = 1$ and hence $A \in \mathbf{S}$, that is $\mathcal{S}\mathbf{R} \subseteq \mathbf{S}$. This together with b) yields $\mathbf{S} = \mathcal{S}\mathbf{R}$. Applying c) to each $A \in \mathcal{U}\mathbf{S}$, from $A/B \notin \mathbf{S}$ for all kernel $B \neq A$, we get $|A/K| = 1$, and so $A = K \in \mathbf{R}$, that is, $\mathcal{U}\mathbf{S} \subseteq \mathbf{R}$. This and a) gives us $\mathbf{R} = \mathcal{U}\mathbf{S}$. Thus $\mathbf{R} = \mathcal{U}\mathcal{S}\mathbf{R}$ and $\mathbf{S} = \mathcal{S}\mathcal{U}\mathbf{S}$ hold. As one easily sees, $\mathbf{R} = \mathcal{U}\mathcal{S}\mathbf{R}$ is equivalent to (Ra) and (Rb) and $\mathbf{S} = \mathcal{S}\mathcal{U}\mathbf{S}$ is equivalent to (Sa) and (Sb). This along with d) proves that $\mathbf{R}$ and $\mathbf{S}$ are corresponding radical and semisimple classes in view of COROLLARY 2.7 or by the definition of $\mathbf{R}$. □

Before giving explicit examples, let us notice that there are plenty of concrete radical classes, for instance, to every partition of simple GS-automata there is a radical class containing exactly one class of the partition (and the other class will be included in the corresponding semisimple class).

EXAMPLE 2.12. We say that a GS-automaton $(A, +, \delta)$ has the *relative* 0-*reset property*, if to every element $a \in A$ there exists an $x \in X^*$ depending on a, such that $ax = 0$. *The class*

$$\mathbf{R} = \{A \in \mathfrak{A} \mid A \textit{ has the relative } 0\textit{-reset property}\} \cup \{(0, +)\}$$

is a radical class. Conditions (Rh), (Ri), (Rk), (Rt) are trivially fulfilled. In view of THEOREM 2.4 we still have to show the validity of (Re). Let K be a kernel of

$A \in \mathfrak{H}$ such that $K \in \mathbf{R}$ and $A/K \in \mathbf{R}$. If $|K| = 1$, then we are done. So let $K \triangleleft A$. Since $K \in \mathbf{R}$ and $|K| > 1$, K is a subsemiautomaton, and therefore A has to be GS-automaton. Let $a \in A$ be an arbitrary element. Since $A/K \in \mathbf{R}$, there exists an $x \in X^*$ such that $(a + K)x \subseteq K$, that is,

$$(a + k)x \in K, \quad \forall k \in K.$$

K is a kernel of A, so also

$$(a + k)x - ax \in K$$

holds. These together yield

$$ax \in K \in \mathbf{R}.$$

Hence there exists a $y \in X^*$ such that $(ax)y = 0$, that is, $a(xy) = 0$ with $xy \in X^*$, proving that A has the relative 0-reset property. Thus $\mathbf{R}$ satisfies also condition (Re), and consequently $\mathbf{R}$ is a radical class.

EXAMPLE 2.13. In a GS-automaton A, 0 is a *reset* if there exists an $x \in X^*$ such that $Ax = 0$. Restricting the universal class to

$$H = \{\text{all finite GS} - \text{automata}\} \cup \{\text{all finite groups}\},$$

the class

$$\mathbf{R} = \{A \in \mathfrak{A} \cap \mathfrak{H} \mid 0 \textit{ is a reset in } A\} \cup \{(0, +)\}$$

is a radical class. Again, conditions (Rh), (Ri), (Rk), (Rt) are trivially satisfied. Notice that (Ri) would not be satisfied for infinite GS-automata. The same proof as in EXAMPLE 2.12 infers the validity of condition (Re), because there the element $x \in X^*$ may be chosen such that $Ax \subseteq K$ and $y \in X^*$ such that $Ky = 0$, whence $A(xy) = 0$.

EXAMPLE 2.14. A GS-automaton $(A, +, \delta)$ is said to be 0-*connected*, if for every $a \in A$ there exists an $x \in X$ such that $0x = a$. Then

$$\mathbf{R} = \{A \in \mathfrak{A} \mid A \text{ is 0-connected}\} \cup \{(0, +)\}$$

is a radical class. Conditions (Rh), (Rt), (Rk), (Ri) are trivially satisfied, only (Re) needs verification. So, let $K \triangleleft A$ such that $K \in \mathbf{R}$ and $A/K \in \mathbf{R}$. Now K has to be a subsemiautomaton, and therefore $KX \subseteq K$. Since $A/K \in \mathbf{R}$, for each $a \in A$ there exists an $x \in X$ such that $Kx \subseteq a + K$. Hence $KX \subseteq K$ implies $a \in K$, and by $K \in \mathbf{R}$ there exists a $y \in X$ with $0y = a$. Clearly, we have also

$$\mathbf{R} = \{A \in \mathfrak{A} \mid A = 0X\} \cup \{(0, +)\}.$$

3 Complementary radical and semisimple classes

We start this section with

EXAMPLE 3.1. The class

$$\mathbf{R} = \{A \in \mathfrak{A} \mid (0, +, \delta) \text{ is a subsemiautomaton in } A\} \cup \{(0, +)\}$$

is a radical class and

$$\mathbf{S} = \mathcal{S}\mathbf{R} = \{A \in \mathfrak{A} \mid 0 \text{ is not a subsemiautomaton in } A\} \cup \mathfrak{G}$$

is the corresponding semisimple class in the universal class $\mathfrak{C}$, as one readily checks. Moreover, $\mathbf{R} \cup \mathbf{S} = \mathfrak{C}$, though $\mathbf{R} \neq \mathfrak{C}$ and $\mathbf{S} \neq \mathfrak{C}$.

Motivated by this EXAMPLE we introduce the following definition.

Let ϱ be a radical operator in $\mathfrak{H}$ with corresponding radical class $\mathbf{R}$ and semisimple class $\mathbf{S}$. We say that ϱ is *complementary*, or that ($\mathbf{R}$ and $\mathbf{S}$) are *complementary*, if

$$\varrho \mathrm{A} = \mathrm{A} \quad or \quad |\varrho \mathrm{A}| = 1, \quad for\ all\ \ \mathrm{A} \in \mathfrak{H},$$

or equivalently,

$$\mathbf{R} \cup \mathbf{S} = \mathfrak{H}.$$

The existence of non-trivial complementary radical operators (here non-trivial means $\mathbf{R} \neq \mathfrak{T} \neq \mathbf{S}$) is a consequence of the fact that in the category $\mathfrak{H}$ the initial object $(0, +)$ is not equivalent to the terminal object $(0, +, \delta)$ (cf. [11]).

THEOREM 3.2. *Let ϱ be a radical operator in $\mathfrak{H}$ with radical class $\mathbf{R}$ and semisimple class $\mathbf{S}$. If*

1) $\mathbf{R}$ *contains at least one nonzero* GS*-automaton and all* GS*-automata of $\mathfrak{H}$ with one-element subsemiautomaton,*

and

2) $\mathbf{S}$ *contains all groups of $\mathfrak{H}$,*

then ϱ is a non-trivial complementary radical operator.

If $\mathfrak{H}$ is closed under forming finite direct sums (in the sense of 6 *of §1) and ϱ is a non-trivial complementary radical operator in $\mathfrak{H}$, then* $\mathbf{R}$ *fulfils* 1) *and* $\mathbf{S}$ *fulfils* 2).

PROOF: Assume that 1) and 2) are satisfied, and let $\mathrm{A} \in \mathfrak{H}$ be an arbitrary nonzero object. If $|\varrho \mathrm{A}| = 1$, then $\mathrm{A} \in \mathbf{S}$. Suppose that $|\varrho \mathrm{A}| > 1$. Since all groups are in $\mathbf{S}$, we conclude by $\varrho \mathrm{A} = \varrho\varrho \mathrm{A} \in \mathbf{R}$ that $\varrho \mathrm{A}$ is a GS-automaton with subsemiautomaton $(0, +, \delta)$, and hence so is A. Thus $\mathrm{A} \in \mathbf{R}$, proving that ϱ is complementary.

Next, suppose that $\mathfrak{H}$ has finite direct sums and ϱ is a non-trivial complementary radical operator. In virtue of (Sc) the semisimple class $\mathbf{S}$ contains at least one group $(\mathrm{A}, +) \notin \mathfrak{T}$. Let $(\mathrm{B}, +) \in \mathfrak{H} \setminus \mathfrak{T}$ be arbitrary. By the assumption on $\mathfrak{H}$ we have $(\mathrm{A}, +) \oplus (\mathrm{B}, +) \in \mathfrak{H}$. Now $(\mathrm{A}, +) \oplus (\mathrm{B}, +) \in \mathbf{R}$ is not possible because then $(\mathrm{A}, +) \oplus (\mathrm{B}, +) \twoheadrightarrow (\mathrm{A}, +)$ and (Rh) would imply $(\mathrm{A}, +) \in \mathbf{R}$. Thus $(\mathrm{A}, +) \oplus (\mathrm{B}, +) \in \mathbf{S}$, as ϱ is complementary. Since $\mathbf{S} \cap \mathfrak{G}$ is a semisimple class of groups, $\mathbf{S} \cap \mathfrak{G}$ is hereditary, and hence

$$(\mathrm{B}, +) \triangleleft (\mathrm{A}, +) \oplus (\mathrm{B}, +) \in \mathbf{S} \cap \mathfrak{G}$$

yields $(\mathrm{B}, +) \in \mathbf{S} \cap \mathfrak{G} \subseteq \mathbf{S}$, proving that $\mathbf{S}$ contains all groups of $\mathfrak{H}$.

Since $\mathbf{S}$ contains all groups and ϱ is non-trivial, $\mathbf{R}$ has to contain at least one nonzero GS-automaton. Assume that $\mathbf{R}$ does not contain all GS-automata of $\mathfrak{H}$ with one-element subsemiautomaton. Then there exists an $(\mathrm{A}, +, \delta_{\mathrm{A}}) \in \mathfrak{H}$ such that $(0, +, \delta_{\mathrm{A}})$ is a subsemiautomaton of $(\mathrm{A}, +, \delta_{\mathrm{A}})$ and $(\mathrm{A}, +, \delta_{\mathrm{A}}) \notin \mathbf{R}$, that is, $(\mathrm{A}, +, \delta_{\mathrm{A}}) \in \mathbf{S}$ by ϱ complementary. Let $(\mathrm{B}, +, \delta_{\mathrm{B}})$ be an arbitrary GS-automaton in $\mathfrak{H}$. By the assumption on $\mathfrak{H}$ the direct sum $(\mathrm{A}, +, \delta_{\mathrm{A}}) \oplus (\mathrm{B}, +, \delta_{\mathrm{B}})$ is in $\mathfrak{H}$. By (Rh) and $(\mathrm{A}, +, \delta_{\mathrm{A}}) \oplus (B, +, \delta_{\mathrm{B}}) \twoheadrightarrow (\mathrm{A}, +, \delta_{\mathrm{A}}) \in \mathbf{S}$ the relation $(\mathrm{A}, +, \delta_{\mathrm{A}}) \oplus (\mathrm{B}, +, \delta_{\mathrm{B}}) \in \mathbf{R}$ is not possible whence by ϱ complementary it follows $(\mathrm{A}, +, \delta_{\mathrm{A}}) \oplus (\mathrm{B}, +, \delta_{\mathrm{B}}) \in \mathbf{S}$. Thus by $\delta_{\mathrm{A}}(0, x) = 0$, $(\mathrm{B}, +, \delta_{\mathrm{B}}) \triangleleft (\mathrm{A}, +, \delta_{\mathrm{A}}) \oplus (\mathrm{B}, +, \delta_{\mathrm{B}})$ and hence by (Sa), there exists a $(\mathrm{B}, +, \delta_{\mathrm{B}}) \twoheadrightarrow (\mathrm{C}, +, \delta_{\mathrm{C}}) \in \mathbf{S}$. Thus by (R$h$) we get $(\mathrm{B}, +, \delta_{\mathrm{B}}) \notin \mathbf{R}$, and since ϱ is complementary, we conclude $(\mathrm{B}, +, \delta_{\mathrm{B}}) \in \mathbf{S}$. Hence $\mathbf{S} = \mathfrak{H}$ and $\mathbf{R} = \mathfrak{T}$

follows, contradicting the assumption that ϱ is non-trivial. Thus $\mathbf{R}$ contains all GS-automata of $\mathfrak{H}$ with one-element subsemiautomaton. □

COROLLARY 3.3. *The class*

$$\mathbf{R}_0 = \{\mathrm{A} \in \mathfrak{H} \cap \mathfrak{A} \mid (0,+,\delta) \text{ is a subsemiautomaton of } \mathrm{A}\} \cup \{(0,+)\}$$

is a complementary radical class in $\mathfrak{H}$. *If* $\mathfrak{H}$ *has finite direct sums and* $\mathbf{R} \neq \mathfrak{T}$ *is a complementary radical class, then* $\mathbf{R}_0 \subseteq \mathbf{R}$.

PROOF: The first statement follows from EXAMPLE 3.1 and the second one from THEOREM 3.2. □

THEOREM 3.4. $\mathbf{R}$ *is a complementary radical class in* $\mathfrak{H}$ *if and only if* $\mathbf{R}$ *satisfies* $(\mathrm{R}h), (\mathrm{R}c), (\mathrm{R}t)$ *and*

(C) $\mathrm{B} \in \mathbf{R}$ *and* $\mathrm{B} \triangleleft \mathrm{A} \in \mathfrak{H}$ *imply* $\mathrm{A} \in \mathbf{R}$.

$\mathbf{S}$ *is a complementary semisimple class in* $\mathfrak{H}$ *if and only if* $\mathbf{S}$ *satisfies* $(\mathrm{S}c)$,

$(\mathrm{S}t)$ $\mathfrak{T} \subseteq \mathbf{S}$,

$(\mathrm{S}h)$ $\mathrm{A} \in \mathbf{S}$ *and* $\mathrm{B} \triangleleft \mathrm{A}$ *imply* $\mathrm{B} \in \mathbf{S}$,

(D) $\mathrm{B} \in \mathbf{S}$, $\mathrm{A} \in \mathfrak{H}$ *and* $\mathrm{A} \longrightarrow\!\triangleright \mathrm{B}$ *imply* $\mathrm{A} \in \mathbf{S}$.

PROOF: Let $\mathbf{R}$ be a complementary radical class. If $\mathrm{A} \in \mathfrak{H} \setminus \mathbf{R}$, then $\mathrm{A} \in \mathcal{S}\mathbf{R}$ and hence $\mathrm{B} \notin \mathbf{R}$ for any $\mathrm{B} \triangleleft \mathrm{A}$, which implies (C).

Conversely, let us assume that $\mathbf{R}$ satisfies $(\mathrm{R}h), (\mathrm{R}c), (\mathrm{R}t)$ and (C). Condition (C) readily implies $(\mathrm{R}e)$ and $(\mathrm{R}i)$. To show $(\mathrm{R}k)$, let us consider a GS-automaton $(\mathrm{A},+,\delta)$ such that $\mathrm{K} \triangleleft (\mathrm{A},+)$ and $\mathrm{K} \in \mathbf{R}$ for some $\mathrm{K} \in \mathfrak{H}$. Now condition (C) implies $(\mathrm{A},+) \in \mathbf{R}$ and condition $(\mathrm{R}c)$ infers $(\mathrm{A},+,\delta) \in \mathbf{R}$, proving $(\mathrm{R}k)$. Thus by THEOREM 2.4 $\mathbf{R}$ is a radical class. Suppose that $\mathrm{A} \notin \mathbf{R}$ for some $\mathrm{A} \in \mathfrak{H}$. Then (C) yields $\mathrm{A} \in \mathbf{S}$, and hence $\mathbf{R}$ is complementary.

Assume that $\mathbf{S}$ is a complementary semisimple class. $(\mathrm{S}t)$ is always satisfied by $(\mathrm{S}a)$ and $(\mathrm{S}b)$ or $(\mathrm{S}\eta)$. If $\mathrm{B} \triangleleft \mathrm{A}$ and $\mathrm{B} \notin \mathbf{S}$, then $\mathrm{B} \in \mathbf{R}$ and hence $\mathrm{A} \notin \mathbf{S}$. This proves $(\mathrm{S}h)$. If $\mathrm{A} \in \mathfrak{H} \setminus \mathbf{S}$, then $\mathrm{A} \in \mathbf{R}$ and hence $\mathrm{A} \longrightarrow\!\triangleright \mathrm{B}$ implies $\mathrm{B} \in \mathbf{R}$ by $(\mathrm{R}h)$. This means that (D) is satisfied.

Conversely, let us suppose that $\mathbf{S}$ satisfies $(\mathrm{S}c), (\mathrm{S}t), (\mathrm{S}h)$ and (D). Condition $(\mathrm{S}h)$ implies trivially $(\mathrm{S}a)$. We want to see the validity of $(\mathrm{S}b)$. Assume that $\mathrm{A} \in \mathfrak{H}$ is such an object that for every $\mathrm{B} \triangleleft \mathrm{A}$ there exists a $\mathrm{B} \longrightarrow\!\triangleright \mathrm{C} \in \mathbf{S}$. From $\mathrm{B} \longrightarrow\!\triangleright \mathrm{C} \in \mathbf{S}$ and (D) we get $\mathrm{B} \in \mathbf{S}$ for every $\mathrm{B} \triangleleft \mathrm{A}$, in particular for $\mathrm{B} = \mathrm{A}$. If there is no $\mathrm{B} \triangleleft \mathrm{A}$, then $|\mathrm{A}| = 1$, and $(\mathrm{S}t)$ infers $\mathrm{A} \in \mathbf{S}$. Thus $(\mathrm{S}b)$ holds and so by COROLLARY 2.7 $\mathbf{S}$ is a semisimple class. We still have to see that $\mathbf{S}$ is complementary. If $\mathrm{A} \in \mathfrak{H} \setminus \mathcal{U}\mathbf{S}$, then there exists an $\mathrm{A} \longrightarrow\!\triangleright \mathrm{B} \in \mathbf{S}$ and so (D) yields $\mathrm{A} \in \mathbf{S}$. Thus $\mathbf{S}$ is complementary. □

4 Additive automata

An element $x_0 \in \mathrm{X}$ is called a *zero-input*, if $0x_0 = 0$. A GS-automaton $(\mathrm{A},+,\delta)$ is said to be *additive*, if there exists a zero-input $x_0 \in \mathrm{X}$ with the following properties :

i) *decomposition property* : $ax = ax_0 + 0x, \quad \forall a \in \mathrm{A}, \quad \forall x \in \mathrm{X}$,

ii) *zero-input additivity* : $(a+b)x_0 = ax_0 + bx_0, \quad \forall a,b \in \mathrm{A}$.

Obviously on every additive group $(\mathrm{A},+)$ of at least two elements one can define at least two non-isomorphic GS-automata. Concerning additive GS-automata the reader is refered to [4].

In the sequel we suppose that *all the* GS-*automata* in the universal class $\mathfrak{H}$ considered, *are additive ones.*

PROPOSITION 4.1. *Let* A *be an additive* GS-*automaton and* $C \lhd (B,+,\delta) \lhd (A,+,\delta)$. *If* $(C,+)$ *is a normal subgroup of* $(A,+)$, *then* C *is a kernel of* A.

PROOF: We have to show that

$$(a+k)x - ax \in C$$

holds for all $a \in A$, $k \in C$ and $x \in X$. Since A is additive, we have

$$(a+k)x - ax = (a+k)x_0 + 0x - 0x - ax_0 = ax_0 + kx_0 - ax_0.$$

Taking into account that C is a kernel of $(B,+,\delta)$, it follows

$$bx_0 + kx_0 - bx_0 = (b+k)x_0 + 0x - 0x - bx_0 = (b+k)x_0 - bx_0 \in C$$

for all $b \in B$. Since $(C,+)$ is a normal subgroup of $(A,+)$, we may conjugate by $(ax_0 - bx_0) \in A$ obtaining

$$ax_0 + kx_0 - ax_0 = (ax_0 - bx_0) + (bx_0 + kx_0 - bx_0) - (ax_0 - bx_0) \in C,$$

regardless as whether kx_0 is in C or not. Thus also

$$(a+k)x - ax \in C$$

holds proving the assertion. □

PROPOSITION 4.2. *Let* ϱ *be an operator assigning to each* $A \in \mathfrak{H}$ *a kernel* ϱA *of* A *and satisfying condition* (ϱa). *If* $(B,+,\delta) \lhd (A,+,\delta) \in \mathfrak{H}$, *then* $\varrho(B,+,\delta)$ *is a kernel of* A.

PROOF: In virtue of PROPOSITION 4.1 we have to prove that $(\varrho B,+)$ is a normal subgroup of $(A,+)$. Since $(B,+)$ is normal in $(A,+)$, for every element $a \in A$ the mapping

$$\varphi_a(b) = a + b - a, \qquad \forall b \in B,$$

is an isomorphism of $(B,+)$ onto itself. Hence condition (ϱa) yields

$$\varphi_a(\varrho B) \subseteq \varrho\varphi_a(B) = \varrho B,$$

proving that $(\varrho B,+)$ is a normal subgroup in $(A,+)$. □

THEOREM 4.3. *Every semisimple class* S *in* $\mathfrak{H}$ *is hereditary, that is,* S *satisfies* (Sh).

PROOF: Let ϱ be the radical operator corresponding to S. If $(B,+,\delta) \lhd (A,+,\delta) \in S$, then by PROPOSITION 4.2 $\varrho(B,+,\delta)$ is a kernel of $(A,+,\delta)$ and by (ϱc) and (ϱd) we have

$$\varrho(B,+,\delta) \subseteq \varrho(A,+,\delta) \in \mathfrak{T}.$$

Thus also $(B,+,\delta) \in S$ holds.

If $(B,+) \lhd (A,+,\delta)$, then also $(B,+) \lhd (A,+)$ is valid. Moreover, condition (Sc) infers $(A,+) \in S$. As is well-known, semisimple classes of groups are hereditary. Hence we conclude $(B,+) \in S$, and the Theorem is proved. □

From THEOREMS 2.11 and 4.3 we obtain immediately.

COROLLARY 4.4. *Subclasses* $\mathbf{R}$ *and* $\mathbf{S}$ *of* $\mathfrak{H}$ *are corresponding radical and semisimple classes if and only if*

a) $\mathbf{R} \cap \mathbf{S} \in \mathfrak{T}$,
b) $\mathbf{R}$ *is homomorphically closed, that is,* $(\mathrm{R}h)$ *is fulfilled,*
c) $\mathbf{S}$ *is strongly hereditary, that is,* $\mathbf{S}$ *satisfies* (Sc) *and* $(\mathrm{S}h)$,
d) *for each* $\mathrm{A} \in \mathfrak{H}$ *there exists a kernel* K *of* A *such that* $\mathrm{K} \in \mathbf{R}$ *and* $\mathrm{A}/\mathrm{K} \in \mathbf{S}$. □

In order to get more explicit results and derive a structure theorem (COROLLARY 4.7) for semisimple objects, we shall restrict our investigations to a universal class $\mathfrak{H}$ in which all the groups are commutative. This class still includes linear sequential machines.

PROPOSITION 4.5. *Let us suppose that* $\mathrm{L} \triangleleft \mathrm{K} \triangleleft (\mathrm{A},+,\delta) \in \mathfrak{H}$. *If* x_0 *is a 0-input for* A *then* $\mathrm{L} \triangleleft (\mathrm{A},+,\delta)$ *if and only if* $\mathrm{L}x_0 \subseteq \mathrm{L}$. *If* $\mathrm{L} \triangleleft (\mathrm{K},+,\delta) \triangleleft (\mathrm{A},+,\delta)$, *then* $\mathrm{L} \triangleleft (\mathrm{A},+,\delta)$.

PROOF:

$$\begin{aligned}
\mathrm{L} \triangleleft (\mathrm{A},+,\delta) \quad &\Leftrightarrow \quad (a+l)x - ax \in \mathrm{L} \text{ for all } a \in \mathrm{A}, l \in \mathrm{L} \text{ and } x \in \mathrm{X},\\
&\Leftrightarrow \quad lx_0 \in \mathrm{L} \text{ for all } l \in \mathrm{L},\\
&\Leftrightarrow \quad \mathrm{L}x_0 \subseteq \mathrm{L}.
\end{aligned}$$

For the second assertion, note that by $\mathrm{L} \triangleleft (\mathrm{K},+,\delta)$ it follows $(k+l)x - kx \in \mathrm{L}$ for all $k \in \mathrm{K}$, $l \in \mathrm{L}$ and $x \in \mathrm{X}$, and hence $lx_0 \in \mathrm{L}$ for all $l \in \mathrm{L}$. Thus the first statement yields $\mathrm{L} \triangleleft (\mathrm{A},+,\delta)$. □

A kernel K of an object $\mathrm{A} \in \mathfrak{H}$ is said to be *essential* in A, if for any other kernel $\mathrm{L} \triangleleft \mathrm{A}$ it follows $\mathrm{K} \wedge \mathrm{L} \notin \mathfrak{T}$. This fact will be denoted by $\mathrm{K} \triangleleft \circ \mathrm{A}$. A subclass $\mathbf{M}$ of $\mathfrak{H}$ is said to be *closed under essential extensions*, if $\mathrm{K} \triangleleft \circ \mathrm{A}$ and $\mathrm{K} \in \mathbf{M}$ imply $\mathrm{A} \in \mathbf{M}$.

THEOREM 4.6. *Let* $\mathbf{M}$ *be a subclass of* $\mathfrak{H} \cap \mathfrak{A}$ *such that* $\mathbf{M}$ *is hereditary, closed under essential extensions and satisfies condition*

(F) $\mathrm{L} \triangleleft \mathrm{K} \triangleleft \mathrm{A} \in \mathfrak{H}$ *and* $\mathrm{K}/\mathrm{L} \in \mathbf{M}$ *imply* $\mathrm{L} \triangleleft \mathrm{A}$.

If $\overline{\mathbf{M}}$ *denotes the subdirect closure of* $\mathbf{M}$ *that is*

$$\overline{\mathbf{M}} = \{\mathrm{A} \in \mathfrak{H} \mid \mathrm{A} \textit{ is a subdirect sum of objects from } \mathbf{M}\}$$

then the class $\mathbf{S} = \overline{\mathbf{M}} \cup (\mathfrak{H} \cap \mathfrak{G})$ *is a semisimple class.*

PROOF: First, we show that every kernel K of a GS-automaton $\mathrm{A} \in \overline{\mathbf{M}}$ is a subsemiautomaton. For this end it suffices to prove that $0\mathrm{X} = 0$. Since $\mathrm{A} \in \overline{\mathbf{M}}$, there are kernels I_α, $\alpha \in \Lambda$, of A such that $\mathrm{A}/\mathrm{I}_\alpha \in \mathbf{M}$ for each $\alpha \in \Lambda$ and $\wedge(\mathrm{I}_\alpha \mid \alpha \in \Lambda) \in \mathfrak{T}$. The class $\mathbf{M}$ consists of GS-automata, so by the hereditariness of $\mathbf{M}$ every kernel, in particular the trivial kernel of $\mathrm{A}/\mathrm{I}_\alpha$ is a subsemiautomaton, and therefore $\mathrm{I}_\alpha \mathrm{X} \subseteq \mathrm{I}_\alpha$ for each $\alpha \in \Lambda$. This implies

$$0\mathrm{X} = (\wedge \mathrm{I}_\alpha)\mathrm{X} \subseteq \wedge \mathrm{I}_\alpha = 0.$$

Next, we are going to prove that $\overline{\mathbf{M}}$ is hereditary. Let us consider an arbitrary kernel K of a GS-automaton $\mathrm{A} \in \overline{\mathbf{M}}$. If $\mathrm{K} \in \mathfrak{T}$, then by the previous statement

$K = (0, +, \delta)$ holds, and so $K \in \mathbf{M} \subseteq \overline{\mathbf{M}}$. So, let us assume that $K \triangleleft A$. Since K is a subsemiautomaton, we have

$$K/(K \wedge I_\alpha) \cong (K \vee I_\alpha)/I_\alpha \triangleleft A/I_\alpha \in \mathbf{M}$$

for all $\alpha \in \Lambda$. Thus the hereditariness of $\mathbf{M}$ yields $K/(K \wedge I_\alpha) \in \mathbf{M}$. Moreover, by $\bigwedge_\alpha (K \wedge I_\alpha) = (\bigwedge_\alpha I_\alpha) \wedge K = 0$ We conclude $K \in \overline{\mathbf{M}}$, proving that $\overline{\mathbf{M}}$ is, in fact, hereditary.

The hereditariness of $\overline{\mathbf{M}}$ readily yields that of the class $\mathbf{S} = \overline{\mathbf{M}} \cup (\mathfrak{H} \cap \mathfrak{G})$, and therefore $\mathbf{S}$ satisfies (S*a*) trivially.

To prove the validity of (S*b*), let us consider an object $A \in \mathfrak{H}$ such that every $K \triangleleft A$ has a nonzero homomorphic image K/L in $\mathbf{S} = \overline{\mathbf{M}} \cup (\mathfrak{H} \cup \mathfrak{G})$. If $A = (A, +) \in \mathfrak{H} \cap \mathfrak{G}$, then $A \in \mathbf{S}$. Hence we shall consider the case $A = (A, +, \delta)$. Let us suppose that $A \notin \mathbf{S}$. Since A is a GS-automaton, we get $A \notin \overline{\mathbf{M}}$. Hence

$$K = \wedge(K_\beta \triangleleft A \mid A/K_\beta \in \mathbf{M}) \neq 0.$$

Since $\mathbf{M}$ is hereditary and consists of GS-automata, the trivial kernel of A/K_β is a subsemiautomaton, which implies $K_\beta X \subseteq K_\beta$, and so each K_β is a subsemiautomaton. Hence so is K as well. By the hypothesis on A, K has a nonzero homomorphic image K/L in S and also $K/L \in \overline{\mathbf{M}}$ holds, for K is a subsemiautomaton. Hence there exists a kernel J/L of K/L such that

$$K/J \cong \frac{K/L}{J/L} \in \mathbf{M} \setminus \mathfrak{T}.$$

Using condition (F) we conclude that J is a kernel of A. Let us choose a kernel M of A being maximal with respect to the property $M \wedge K = J$. By Zorn's Lemma such a kernel M does exist. Now we have

$$K/J = K/(M \wedge K) \cong (K + M)/M \triangleleft A/M.$$

For any $Q/M \triangleleft A/M$ the choice of M yields $J \subsetneq Q \wedge K$, and therefore

$$(Q \wedge K)/J \triangleleft K/J \in \mathbf{M}.$$

Thus the hereditariness of $\mathbf{M}$ infers that $(Q \wedge K)/J$ is a GS-automaton, and so

$$0 \neq (Q \wedge K)/J \cong ((Q \wedge K) \vee M)/M \subseteq ((K \vee M)/M) \wedge (Q/M),$$

proving that $(K \vee M)/M$ is essential in A/M. Hence by

$$(K \vee M)/M \cong K/J \in \mathbf{M}$$

and by $\mathbf{M}$ being closed under essential extensions, we conclude $A/M \in \mathbf{M}$. This implies by the definition K that $K \subseteq M$, and so $J = M \wedge K = K$ holds, yielding $K/J \in \mathfrak{T}$, contradicting $K/J \notin \mathfrak{T}$. Thus $A \in \mathbf{S}$ has been proved, establishing the validity of condition (S*b*).

Since $\mathfrak{G} \cap \mathfrak{H} \subseteq \mathbf{S}$ by definition, the class $\mathbf{S}$ fulfils condition (S*c*), too. Thus in view of COROLLARY 2.7 $\mathbf{S}$ is a semisimple class. □

COROLLARY 4.7. *Let* $\mathbf{M}$ *be a subclass of* $\mathfrak{H} \cap \mathfrak{A}$ *such that* $\mathbf{M}$ *is hereditary, closed under essential extensions and satisfies condition* (F). *Then*

$$\mathbf{R} = \{A \in \mathfrak{H} \cap \mathfrak{A} \mid A \longrightarrow\!\!\triangleright B \Rightarrow B \notin \mathbf{M}\} \cup \{(0,+)\}$$

is a radical class. Denoting by ϱ *the corressponding radical operator,* $\varrho A = 0$ *for a* GS-*automaton* $A \in \mathfrak{H} \cap \mathfrak{A}$ *if and only if* $A = \sum_{\text{subdirect}} (A_\alpha \mid A_\alpha \in \mathbf{M})$. *In particular if* A *satisfies also the descending chain condition on kernels, then* A *is a finite direct sum of* GS-*automata from the class* $\mathbf{M}$.

PROOF: The assertions are immediate consequences of THEOREM 4.6, because $\mathbf{R} = \mathcal{U}\mathbf{S}$. The last assertions can be proved by standard reasoning (cf. [2] Corollary 5). □

PROPOSITION 4.8. *The radical class* $\mathbf{R}$ *of* COROLLARY 4.7 *has the following hereditary property :*

$$\textit{if } (K,+,\delta) \triangleleft (A,+,\delta) \in \mathbf{R}, \textit{ then } (K,+,\delta) \in \mathbf{R}.$$

PROOF: Suppose that $(K,+,\delta) \notin \mathbf{R}$. Then there exists a kernel L of K such that $K/L \in \mathbf{S} \setminus \mathfrak{T}$. Since K is a GS-automaton, necessarily $K/L \in \overline{\mathbf{M}}$ holds. Thus also $K/J \in \mathbf{M}$ holds with an appropriate $L \subseteq J \triangleleft K$. Applying condition (F) on $\mathbf{M}$, it follows $J \triangleleft A$. Let M be a kernel of A being maximal relative to the property $K \wedge M = J$. As we have seen in the proof of THEOREM 4.6,

$$K/J = K/(K \wedge M) \cong (K \vee M)/M \triangleleft \circ A/M.$$

Since $K/J \in \mathbf{M}$ and $\mathbf{M}$ is closed under essential extensions, we get $A/M \in \mathbf{M} \subseteq \mathbf{S}$. Thus $A/M \subseteq \mathbf{S} \cap \mathbf{R} = \mathfrak{T}$, which yields $A = M$, and also $J = K \wedge M = K$, as well as $K/J \in \mathfrak{T}$, a contradiction. Thus $K = (K,+,\delta) \in \mathbf{R}$ has been proved. □

Recall that an object $A \in \mathfrak{H}$ is said to be *subdirectly irreducible*, if $H = \wedge(K \triangleleft A) \notin \mathfrak{T}$. The kernel H of A is referred to as the *heart* of A.

In the sequel we give a concrete class $\mathbf{M}$ of GS-automata which satisfies the conditions required in THEOREM 4.6, COROLLARY 4.7 and PROPOSITION 4.8.

THEOREM 4.9. *The class*

$$\mathbf{M} = \{A = (A,+,\delta) \in \mathfrak{H} \mid A \textit{ is subdirectly irreducible and } 0X = 0\}$$

is hereditdary, closed under essential extensions and satisfies condition (F).

Furthermore, for the radical class

$$\mathbf{R} = \{A \in \mathfrak{H} \cap \mathfrak{A} \mid A \longrightarrow\!\!\triangleright B \Rightarrow B \notin \mathbf{M}\} \cup \{(0,+)\}$$

the following two conditions are equivalent :

(i) $A \in \mathbf{R} \setminus \mathfrak{T}$

(ii) $A \in \mathfrak{H} \cap \mathfrak{A}$ *and if* $K \triangleleft A$ *and* $K \longrightarrow\!\!\triangleright L$, *then* L *is not a simple* GS-*automaton with subsemiautomaton* 0.

In analogy with ring theory we may call this radical $\mathbf{R}$ the *antisimple radical* of commutative additive GS-automata.

PROOF: Since $0X = 0$, every kernel K of any $A \in \mathbf{M}$ is a subsemiautomaton. Hence by PROPOSITION 4.5 every kernel L of K is also a kernel of A. Thus the heart of A is contained in every $L \triangleleft K$, and therefore K is subdirectly irreducible, proving that **M** is hereditary.

For proving that **M** is closed under essential extensions, let us consider a $K \in \mathbf{M}$ and $K \triangleleft \circ A$. We have to show that A is subdirectly irreducible. Let $I \triangleleft A$ be arbitrary. Since $K \triangleleft \circ A$, it follows $K \wedge I \notin \mathfrak{T}$, and so the heart H of K is contained in $K \wedge I$ and also in I. Since I was arbitrary, also $H \subseteq \wedge(I \triangleleft A)$ holds, proving that A is subdirectly irreducible.

In order to show the validity of condition (F), let us suppose that $L \triangleleft K \triangleleft A \in \mathfrak{H}$ and that $K/L \in \mathbf{M}$. Since $K/L \in \mathbf{M}$, we have $LX \subseteq L$. Thus L is a subsemiautomaton of A, and consequently K as well as A are GS-automata. Hence PROPOSITION 4.5 yields $L \triangleleft A$.

By COROLLARY 4.7 **R** is a radical class. Assume that $K \triangleleft A \in \mathbf{R}$ and $K \to\!\!\triangleright L$. If K is merely a group, then so is L too, and condition (ii) is trivially fulfilled. So we may suppose that K is a subsemiautomaton. By PROPOSITION 4.8 it follows that $K \in \mathbf{R}$ which implies $L \notin \mathbf{M}$. Since simple GS-automata are subdirectly irreducible, by the definition of **M** we conclude that either L is not simple or 0 is not a subsemiautomaton of L or both, proving the validity of (ii).

Suppose that $A \notin \mathbf{R}$. Then either A is a group or $A/J \in \mathbf{M}$ with a suitable kernel J of A. In the second case, since the class **M** is hereditary, also the heart L of A/J is in **M** and in view of PROPOSITION 4.5 L has to be a simple GS-automaton. Since $L = K/J$ with an appropriate kernel K of A, we see that (ii) is not satisfied. □

As is well known [10] the subdirectly irreducible abelian groups are precisely the (quasi)-cyclic groups $C(p^n)$, $n = 1, 2, \ldots, \infty$ for all primes p. Obviously, on every subdirectly irreducible abelian group we may define an additive GS-automaton by assigning a homomorphism $x_0 \colon C(p^n) \to C(p^n)$, which will be a 0-input, and by defining $0X = 0$. There are, however, subdirectly irreducible additive GS-automata the additive group thereof is not subdirectly irreducible. Consider, for instance, the direct sum $C(p) \oplus C(p)$ of two copies of a simple cyclic group, the automorphism x_0 interchanging the components of $C(p) \oplus C(p)$. x_0 can be regarded as a 0-input of $C(p) \oplus C(p)$, further define $0X = 0$. Thus we have got a simple and hence subdirectly irreducible additive GS-automaton $(C(p) \oplus C(p), +, \delta)$, though $(C(p) \oplus C(p), +)$ is not a subdirectly irreducible group. Moreover, there are subdirectly irreducible additive GS-automata, which are not in **M**, for instance $(C(p), +, \delta)$ where the 0-input x_0 may be any homomorphism $x_0 \colon C(p) \to C(p)$, but $0X \neq 0$ for some $x \in X$. These observations demonstrate that COROLLARY 4.7 applied to the class **M** of THEOREM 4.9 provides a subdirect decomposition for some additive GS-automata only, and that the subdirectly irreducible components are not necessarily subdirectly irreducible groups.

The third author gratefully acknowledges the financial support of the National Science Council of the Republic of China and the kind hospitality of the National Cheng-Kung University, Tainan, Taiwan, R.O.C. 1991 Mathematics Subject Classification : Primary 68Q70, Secondary 18B20.

Acknowledgement : The third author gratefully acknowledges the financial support of the National Science Council of the Republic of China and the kind hospitality of the National Cheng-Kung University, Tainan, Taiwan, R.O.C.

References

[1] J. Adámek and V. Trnková, *Automata and Algebras in Categories*, Kluwer Acad. Publ., 1990.

[2] T. Anderson, K. Kaarli and R. Wiegandt, Radicals and subdirect decomposition, *Comm. in Algebra*, 13(1985), 479-494.

[3] S. Eilenberg, *Automata, languages and machines*, Vol.A, Academic Press, 1974.

[4] Y.Fong, F.K. Huang and W.F. Ke, Syntactic near-rings associated with group semiautomata, *PU.M.A.* (Hungary) Ser A, Vol. 2 (1991), pp. 187-204.

[5] G. Hofer, *Near-rings and group automata*, Ph.D. Thesis, J. Kepler-Universität, Linz, 1987.

[6] G. Hofer and G. Pilz, Group automata and near-rings, *Contr. to General Alg.* 2, Proc. Klagenfurt Conf., 1982, Hölder- Pichler-Tempsky, Wien and B.G. Teubner, Stuttagrt, 1983, 153-162.

[7] H.C. Hutchins and H.J. Weinert, Homomorphisms and kernels of semifields, *Period. Math. Hung.*, 21(1990), 113-152.

[8] L. Márki, R. Mlitz and R.Wiegandt, A general Kurosh-Amitsur radical theory, *Comm. in Algebra*, 16(1988), 249-305.

[9] G. Pilz, *Near-rings*, North-Holland, 1983.

[10] B.M. Schein, Homomorphisms and subdirect decompositions of semigroups, *Pacific J. Math.*, 17(1966), 529-549.

[11] S. Veldsman and R. Wiegandt, On the existence and non-existence of complementary radical and semisimple classes, *Quaest. Math.* 7(1984), 213-224.

[12] H. J. Weinert and R. Wiegandt, A Kurosh-Amitsur radical theory for proper semifields, *Comm. in Algebra*, 20(1992), 2419-2458.

Received May 21, 1993

Revised March 21, 1994

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

Structuring grammar systems by priorities and hierarchies*

Victor Mitrana † Gheorghe Păun‡ Grzegorz Rozenberg §

Abstract

A grammar system is a finite set of grammars that cooperate to generate a language. We consider two generalizations of grammar systems: (1) adding a priority relation between single grammar components, and (2) considering hierarchical components which by themselves are grammar systems. The generative power of these generalized grammar systems is investigated, and compared with the generative power of ordinary grammar systems and of some well-known types of grammars with regulated rewriting (such as matrix grammars). We prove that for many cooperating strategies the use of priority relation increases the generative capacity, however this is not the case for the maximal mode of derivation (an important case, because it gives a characterization of the ETOL languages). We also demonstrate that in many cases the use of hierarchical components does not increase the generative power.

1 Introduction

A cooperating grammar system (introduced in [7], and motivated by considerations related to two level grammars), is a set of usual Chomsky grammars which cooperate in rewriting sentential forms. In [7] a component that is currently rewriting a sentential form cannot quit until it introduces a symbol which it cannot rewrite (the current sentential form is not a sentential form of this component). Only one component at a time rewrites a sentential form. The set of terminal strings obtained in

*Research supported by project 11281 of the Academy of Finland, the Basic Research ASMICS II Working Group, and, in the case of the second author, also by the Alexander von Humboldt Foundation.

†University of Bucharest, Department of Mathematics Str. Academiei 14, 70109 Bucureşti, Romania

‡Institute of Mathematics of the Romanian Academy of Sciences P.O.Box 1 – 764, 70700 Bucureşti, Romania

§University of Leiden, Department of Computer Science Niels Bohrweg 1, 2333 CA Leiden, The Netherlands and Department of Computer Science, University of Colorado at Boulder Boulder, CO 80309, USA

this way is the language generated by the system. It is shown in [7] that this type of cooperating grammar systems (equiped with a control over the sequencing of the individual components) generates the family of programmed languages (which is equal to the family of languages generated by matrix grammars).

The cooperating grammar systems were rediscovered in [1], under the name of *modular grammars* (a term related to the time varying grammars). A rather intensive study of cooperating grammar systems has been initiated in [2], where the grammar systems were related to the notions from artificial intelligence, such as the blackboard model in problem solving [9]. (See also Chapter 1 of [3] for further links between grammar systems and topics in artificial intelligence, computer science, and cognitive psychology.) Within this framework, more conditions on enabling and disabling of individual components were considered. Two, quite basic, examples of this type are: the step limitations (a component must work exactly, or at least, or at most a given prescribed number of steps), and the maximal competence strategy (a component must work as long as it can) – this is similar in some extent to the stoping condition from [7]. The latter strategy is particularly interesting, because it yields a characterization of the family of ET0L languages.

A number of novel cooperating strategies has been considered recently – forming the *teams* of components, as in [6] and [9], is one of such strategies.

In this paper we consider two quite natural modifications of the basic model. The first of these is adding a *priority relation* between the components of a system. A component can become active only when no other component with a greater priority can rewrite the current string. The other modification consists of allowing components which by themselves are grammar systems, or systems of grammar systems, etc.

We demonstrate that neither of the two modifications increases the generative capacity when maximal competence strategy is used. For the other strategies, adding the priority relation strictly increases the generative power.

We end this section by pointing out that both modifications of grammar systems we consider in this paper, viz. priorities and hierarchies, are very natural. Adding priorities in rewriting systems in order to ensure the deterministic applicability of rules is a rather standard mechanism – e.g. it is used in regulated rewriting in context-free grammars and in term rewriting systems. Also, the way that a computation in a grammar system is defined on the base of computations of basic units (grammars) may be seen as just a specific cooperation mechanism. In order to understand its power, it is natural to consider the bootstrapping of this mechanism

- take grammar systems as basic units and obtain "grammar systems of depth 2" by organizing their work together by a given cooperation mechanism,

and proceeding inductively

- take grammar systems of depth $i \geq 2$ and organize their work together by a given cooperation mechanism obtaining "grammar systems of depth $i + 1$".

Then a way to understand a given cooperation mechanism as defined in grammar systems is to investigate the relationship between the generative power of grammar systems of different depth. This leads one then to hierarchical grammar systems.

2 Basic definitions

For an alphabet V, V^* denotes the free monoid generated by V; the empty string is denoted by λ, and $|x|$ denotes the length of $x \in V^*$. The families of context-free, context-sensitive and recursively enumerable languages are denoted by CF, CS, and RE, respectively; $ET0L$ denotes the family of ET0L languages.

A *matrix grammar* is a construct $G = (N, T, S, M, F)$, where N, T are disjoint alphabets, $S \in N, M$ is a finite set of sequences, called *matrices*, $(A_1 \to x_1, \ldots, A_n \to x_n), n \geq 1$, of context-free rules over $N \cup T$, and F is a set of occurrences of rules in matrices of M.

For $m = (A_1 \to x_1, \ldots, A_n \to x_n) \in M$, and $w, w' \in (N \cup T)^*$, we define $w \Longrightarrow_m w'$ iff there are $w_1, w_2, \ldots, w_{n+1}$ in $(N \cup T)^*$ such that $w = w_1, w' = w_{n+1}$, and for each $i, 1 \leq i \leq n$, either $w_i = w_i' A_i w_i'', w_{i+1} = w_i' x_i w_i''$, or A_i does not occur in $w_i, w_{i+1} = w_i$ and $A_i \to x_i$ appears in F.

If $F = \emptyset$, then the grammar is said to be without *appearance checking* (and the component F is omitted from the specification of G).

We denote by MAT_{ac} (respectively, MAT_{ac}^{λ}) the family of languages generated by λ-free (arbitrary) matrix grammars; when the appearance checking feature is not present we remove the subscript *ac*.

A (context-free) *ordered grammar* is a construct $G = (N, T, S, P, \succ)$, where N, T, S, P are as in a context-free grammar, and $\succ$ is a partial order relation over P. A rule $A \to x$ in P can be used for rewriting a string w only if no rule $B \to y$ in P with $B \to y \succ A \to x$ can rewrite the string w. The family of languages generated by λ-free ordered grammars is denoted by ORD, and ORD^{λ} is used for the case when λ-rules are allowed.

It is known that

$$CF \subset MAT \subset MAT_{ac} \subset CS,$$
$$MAT \subset MAT^{\lambda} \subset MAT_{ac}^{\lambda} = RE,$$
$$CF \subset ET0L \subset ORD \subset MAT_{ac}.$$

For the basic elements of formal language theory the reader is referred to [11]; for Lindenmayer systems we refer to [10] and for regulated rewriting to [4].

Definition 1 *A cooperating distributed (cd, for short) grammar system is a construct*

$$\Gamma = (N, T, S, P_1, P_2, \ldots, P_n),$$

where N, T are disjoint alphabets, $S \in N$, and $P_i, 1 \leq i \leq n$, are finite sets of context-free rules over $N \cup T$.

The sets P_i are called the *components* of Γ; we also say that Γ is a cd grammar system *of degree n*.

For a component P_i from a grammar system Γ as above, $dom(P_i) = \{A \in N \mid A \to x \in P_i\}$, and we define the derivation relation $\Longrightarrow_{P_i}$ in the usual way. Then we can consider derivations in P_i of exactly k successive steps, of at least k steps, at most k steps, and of an arbitrary number of steps; they are denoted by $\Longrightarrow_{P_i}^{=k}, \Longrightarrow_{P_i}^{\geq k}, \Longrightarrow_{P_i}^{\leq k}$, and $\Longrightarrow_{P_i}^{*}$, respectively. Another important relation is

$$x \Longrightarrow_{P_i}^{t} y \text{ iff } x \Longrightarrow_{P_i}^{*} y \text{ and there is no } z \in (N \cup T)^* \text{ such that } y \Longrightarrow_{P_i} z$$

(the derivation is maximal in the component P_i).

In this way we have specified stop conditions for the components, i.e. conditions under which an active component must/can become inactive.

For $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ the language generated by Γ in the f mode is defined by

$$L_f(\Gamma) = \{x \in T^* \mid S \Longrightarrow^f_{P_{i_1}} x_1 \Longrightarrow^f_{P_{i_2}} x_2 \Longrightarrow^f_{P_{i_3}} \dots \Longrightarrow^f_{P_{i_r}} x_r = x, \\ r \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq r\}.$$

The family of such languages, generated by systems with at most n components (all of them without λ-rules) is denoted by $CD_n(f)$ (if λ-rules are allowed, then we write $CD_n^\lambda(f)$). The union of the families $CD_n(f)$ for all n is denoted by $CD_\infty(f)$.

In [2] and [3] it is proved that:

$$\begin{aligned}
&CF = CD_\infty(=1) = CD_\infty(\geq 1) = CD_\infty(*) = CD_\infty(\leq k),\ k \geq 1, \\
&CF \subset CD_n(=k) \cap CD_n(\geq k),\ n \geq 2, k \geq 2, \\
&CD_\infty(=k) \subseteq MAT,\ CD_\infty(\geq k) \subseteq MAT,\ k \geq 1, \\
&CF = CD_1(t) = CD_2(t) \subset CD_n(t) = ET0L \\
&\qquad (\text{hence also } CD_n^\lambda(t) = ET0L),\ n \geq 3.
\end{aligned}$$

3 Introducing orderings and hierarchies into grammar systems

We introduce now new classes of grammar systems which will be investigated in this paper.

Definition 2 *A grammar system with priorities (pcd grammar system) is a construct $\Gamma = (N, T, S, P_1, \dots, P_n, \succ)$, where $N, T, S, P_1, \dots, P_n$ are as in a cd grammar system, and $\succ$ is a partial order relation over the set of components. For a derivation mode f, two strings $x, y \in (N \cup T)^*$, and a component P_i of Γ we write $x \Longrightarrow^{(f,\succ)}_{P_i} y$ if and only if $x \Longrightarrow^f_{P_i} y$ and for no component P_j with $P_j \succ P_i$ and no string $z \in (N \cup T)^*$, $x \Longrightarrow^f_{P_j} z$ holds.*

Note that if $x \Longrightarrow^f_{P_i} y$, then no P_j with $P_j \succ P_i$ can rewrite x in the f mode – but there may be P_j with $P_j \succ P_i$ that can rewrite x in some way (e.g. P_j can make only one rewriting step on x while $f =$ " ≥ 2").

We denote by $PCD_n(f)$ the family of languages generated by (λ-free) pcd grammar systems of degree at most n in the derivation mode f. Again, we add the superscript λ when also λ-rules may be used, and we replace n with ∞ when the degree is not bounded.

Here is an example of a pcd gramar system. Let

$$\begin{aligned}
\Gamma &= (\{S, A, B, A', B', A'', B''\}, \{a, b, c\}, S, P_1, P_2, P_3, P_4, P_5, \succ),\\
& P_1 = \{A \to aA'b,\ B \to cB'\},\\
& P_2 = \{A \to A'',\ B \to B''\},\\
& P_3 = \{A' \to A,\ B' \to B,\ A'' \to ab,\ B'' \to c\},\\
& P_4 = \{A' \to A',\ B' \to B',\ A'' \to A'',\ B'' \to B''\},\\
& P_5 = \{A \to A,\ B \to B,\ S \to AB\},\\
\text{and}\quad & P_4 \succ P_1,\ P_4 \succ P_2,\ P_5 \succ P_3.
\end{aligned}$$

Then

$$L_f(\Gamma) = \{a^n b^n c^n \mid n \geq 1\},$$

for all $f \in \{*, \geq 1\} \cup \{\leq k \mid k \geq 2\}$ (and also for $f \in \{= 2, \geq 2\}$).

Indeed, take a string $a^n A b^n c^n B, n \geq 0$; after using P_5, the component in which we must start any derivation, we have $n = 0$. We can apply either P_1 or P_2, using only one or both rules from each of these components. If we use only one rule, then we obtain either $a^{n+1}A'b^{n+1}c^n B$ or $a^n A b^n c^{n+1} B'$ when using P_1, and we obtain either $a^n A'' b^n c^n B$ or $a^n A b^n c^n B''$ when using P_2. In all cases, both P_4 and P_5 can be used afterwards (and one of them has to be used, because they have the priority over P_1, P_2, P_3). However, nothing changes then in the current string, and so the derivation is blocked. Consequently, when using P_1, P_2 we must use both rules from each of them, thus obtaining either $a^{n+1}A'b^{n+1}c^{n+1}B'$*or* $a^n A'' b^n c^n B''$. Now P_4 is applicable and it changes nothing, but it does not forbid the use of P_3 (P_5 is not applicable). If, using P_3, only one of A', B' in $a^{n+1}A'b^{n+1}c^{n+1}B'$ is replaced by A, B, respectively, then again the derivation is blocked in the components P_4, P_5, hence we must produce $a^{n+1}Ab^{n+1}c^{n+1}B$ – this is a string of the form that we have started with, hence the derivation can be iterated. If from $a^n A'' b^n c^n B''$ we produce either $a^{n+1}b^{n+1}c^n B''$ or $a^n A'' b^n c^{n+1}$, then the only applicable components are P_3 and P_4; P_4 changes nothing, hence we eventually will use P_3 again, and get in this way a terminal string $a^{n+1}b^{n+1}c^{n+1}$.

Definition 3 *A hierarchical grammar system (hcd grammar system) of depth $h, h \geq 0$, is*

1. *a context-free grammar $\Gamma = (N, T, S, P)$ if $h = 0$,*

2. *a construct $\Gamma = (N, T, S, \gamma_1, \gamma_2, \ldots, \gamma_m)$, $m \geq 1$, if $h \geq 1$, where $\Gamma_i = (N, T, S, \gamma_i), 1 \leq i \leq m$, are grammar systems of depth $h - 1$.*

Thus, at the bottom level of a hcd grammar system we have sets of context-free rules, on the next level it contains sets of such sets, then sets of sets of sets and so on. The systems $\gamma_1, \ldots, \gamma_m$ from the specification of Γ in point 2 of the above definition are called *components* or *subsystems of* Γ *of depth* $h - 1$.

Here is an example of a hcd grammar system of depth 2:

$$\begin{aligned}
\text{level two}\ &: \Gamma = (\{S, A, B, A', B'\}, \{a, b, c\}, S, \gamma_1, \gamma_2),\\
\text{level one}\ &: \gamma_1 = \{\gamma_{1,1}, \gamma_{1,2}\},\\
& \quad \gamma_2 = \{\gamma_{2,1}\},
\end{aligned}$$

$$\text{level zero}: \gamma_{1,1} = \{A \to aA'b,\ B \to cB'\},$$
$$\gamma_{1,2} = \{A' \to A,\ B' \to B\},$$
$$\gamma_{2,1} = \{S \to AB,\ A \to A,\ A \to ab,\ B \to c\}.$$

We known how to define a derivation step in a system of depth 0 (this is a usual derivation step in a context-free grammar), and we know how to define the derivation modes $\Longrightarrow_P^f$, for $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ in a set P of rules. Then, for a system of depth $h \geq 2$, $\Gamma = (N, T, S, \gamma_1, \ldots, \gamma_m)$ we define, for the component γ_j, $1 \leq j \leq m$,

$$x \Longrightarrow_{\gamma_j}^{=k} y \text{ iff } x \Longrightarrow_{\gamma_{j,i_1}}^{=k} x_1 \Longrightarrow_{\gamma_{j,i_2}}^{=k} \ldots \Longrightarrow_{\gamma_{j,i_k}}^{=k} x_k = y,$$
$$\gamma_{j,i_r}, 1 \leq r \leq k, \text{ are components of } \gamma_j;$$
$$x \Longrightarrow_{\gamma_j}^{\leq k} y \text{ iff } x \Longrightarrow_{\gamma_{j,i_1}}^{\leq k} \Longrightarrow_{\gamma_{j,i_2}}^{\leq k} \ldots \Longrightarrow_{\gamma_{j,i_s}}^{\leq k} x_s = y,$$
$$\gamma_{j,i_r}, 1 \leq r \leq s, \text{ are components of } \gamma_j, r \leq k,$$
$$x \Longrightarrow_{\gamma_j}^{\geq k} y \text{ iff } x \Longrightarrow_{\gamma_{j,i_1}}^{\geq k} \Longrightarrow_{\gamma_{j,i_2}}^{\geq k} \ldots \Longrightarrow_{\gamma_{j,i_s}}^{\geq k} x_s = y,$$
$$\gamma_{j,i_r}, 1 \leq r \leq s, \text{ are components of } \gamma_j, r \geq k,$$
$$x \Longrightarrow_{\gamma_j}^{*} y \text{ iff } x \Longrightarrow_{\gamma_{j,i_1}}^{*} \Longrightarrow_{\gamma_{j,i_2}}^{*} \ldots \Longrightarrow_{\gamma_{j,i_s}}^{*} x_s = y,$$
$$\gamma_{j,i_r}, 1 \leq r \leq s, \text{ are components of } \gamma_j, r \geq 0,$$
$$x \Longrightarrow_{\gamma_j}^{t} y \text{ iff } x \Longrightarrow_{\gamma_j}^{*} y \text{ and there is no } z \in (N \cup T)^*$$
$$\text{such that } y \Longrightarrow_{\gamma_j}^{=1} z.$$

Continuing the previous **example**, let us consider the $= 2$ derivation mode. Starting from S, we must use γ_2, which contains only one subsystem, hence

$$S \Longrightarrow_{\gamma_2}^{=2} x \text{ means } S \Longrightarrow_{\gamma_{2,1}}^{=2} x_1 \Longrightarrow_{\gamma_{2,1}}^{=2} x.$$

Hence after using $S \to AB$ and $A \to A$ (three times) we obtain $x = AB$. Now γ_1 must be applied, that is we must find a derivation

$$AB \Longrightarrow_{\gamma_{1,i}}^{=2} y_1 \Longrightarrow_{\gamma_{1,j}}^{=2} y_2,$$

for $i, j \in \{1, 2\}$. The only possibility is $i = 1, j = 2$, hence we get

$$AB \Longrightarrow_{\gamma_1}^{=2} aAbcB, \text{ because } AB \Longrightarrow_{\gamma_{1,1}}^{=2} aA'bcB' \Longrightarrow_{\gamma_{1,2}}^{=2} aAbcB.$$

This step can be iterated, obtaining $a^n Ab^n c^n B, n \geq 0$, and then γ_2 can be used for replacing A, B with ab, c, respectively. If the current string contains only one nonterminal, then γ_1 cannot be applied, hence after using γ_2 either a nonterminal string as above is produced or a terminal string must be obtained. It is easy to see that the generated language is

$$L_{=2}(\Gamma) = \{a^n b^n c^n \mid n \geq 1\}.$$

We denote by $H_hCD(f)$ the family of languages generated by grammar systems of depth at most $h, h \geq 1$, in the derivation mode f; we also set $H_0CD(f) = CF$, for all f.

4 The generative power of grammar systems with priorities

In this section we will consider the effect of adding a priority relation on the generative power of grammar systems.

The next results follow directly from the definitions.

Lemma 1 $CD_n(f) \subseteq PCD_n(f), PCD_n(f) \subseteq PCD_{n+1}(f), n \geq 1,$ *for all* $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$.

The example from the end of the previous section implies that $PCD_n(f) - CF \neq \emptyset$, for $n \geq 5, f \in \{*, \geq 1\} \cup \{\leq k \mid k \geq 2\}$. Since $CD_n(f) = CF$, for all $n \geq 1$, and f as above (see the end of Section 2), this demonstrate that adding priorities strictly increases the generative power. This result can be extended also to modes of derivation other than t.

Theorem 1 $PCD_n(f) - CD_\infty(f) \neq \emptyset,\ n \geq 10, f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$.

Proof. Consider the system

$$\Gamma = (\{S, A, A', B, B', B''\}, \{a\}, S, P_1, P_2, \ldots, P_{10}, \succ),$$

with the components and the priority relations given in the following figure, where the components P_i, P_j are in relation $P_i \succ P_j$ iff P_i is placed above P_j in one of the "composite boxes" below:

$P_1 : \begin{matrix} B' \to B' \\ B'' \to B'' \end{matrix}$
$P_2 : \begin{matrix} A \to A \\ A \to A'A' \end{matrix}$

$P_3 : B \to B$
$P_4 : \begin{matrix} A' \to A' \\ A' \to A \end{matrix}$

$P_5 : \begin{matrix} S \to AB \\ A \to A \end{matrix}$
$P_6 : \begin{matrix} B \to B \\ B \to B' \\ B'' \to B'' \\ B'' \to a \end{matrix}$

$P_7 : A' \to A'$
$P_8 : \begin{matrix} B' \to B' \\ B' \to B \\ B' \to B'' \end{matrix}$

$P_9 : \begin{matrix} B \to B \\ B' \to B' \end{matrix}$
$P_{10} : \begin{matrix} A \to A \\ A \to a \end{matrix}$

Notice first that the components P_1, P_3, P_7, P_9 consist of rules of the form $X \to X$ only, hence their application doės not change the current string. The same is true for P_5, except for the first step of a derivation, because S never appears later in a sentential form. Therefore, all components P_1, P_3, P_7, P_9 (as well as P_5 after the first step) check the appearance of the corresponding nonterminals and block the components P_2, P_4, P_8, P_{10} (and P_6), respectively. For this reason we will call P_1, P_3, P_5, P_7, P_9 the *control components* and $P_2, P_4, P_6, P_8, P_{10}$ the *rewriting components*.

The derivation starts in P_5 by producing the string AB (if we have a derivation mode $= k$ or $\geq k$ for $k \geq 2$, then we can use $k-1$ times the rule $A \rightarrow A$; this is true for all rewriting components, because they contain rules of the form $X \rightarrow X$, which do not modify the current string). Assume then that we have already generated a string $A^n B, n \geq 1$. The presence of the rules $A \rightarrow A$ and $B \rightarrow B$ in P_5 and P_9 forbides the use of components P_6 and P_{10}; P_4 and P_8 are not applicable to A and B. Thus P_2 is the only component which changes the current string. The obtained string will contain occurrences of both A and A' (and of B). Due to the presence of A we cannot use P_6, and due to the presence of B we cannot use P_4 and P_{10}; P_8 is not applicable. Therefore we must again use P_2 until all occurrences of A are replaced by A'. The so obtained string is of the form $A'^{2n}B$. Now the only applicable component which changes the string is P_6, and its use leads to a string of the form $A'^{2n}B'$, which allows the use of P_4 (and only of P_4, with the exception of control components like P_7 and P_9 which do not change the string under rewriting) which replaces occurrences of A' by A. As long as A, A' and B' are present, the only possibility is to continue to apply P_4 until each A' is replaced by A, obtaining in this way $A^{2n}B'$. Now one can apply P_8 (and only P_8 with the exception of P_1, P_5, P_9 which do not change the string under rewriting). If $A^{2n}B$, is obtained, then the above process can be iterated. If $A^{2n}B''$ is obtained, then the only applicable component (which changes the string) is P_{10}; it must be then used until each A is replaced by a. When A is not present anymore, one can use P_6, finishing the derivation by replacing B'' with a.

Consequently,

$$L_f(\Gamma) = \{a^{2^n+1} \mid n \geq 1\}.$$

Since $L_f(\Gamma)$ is not context-free, it is not in $CD_\infty(f)$, for $f \in \{*, =1, \geq 1\} \cup \{\leq k \mid k \geq 1\}$. Moreover, it is proved in [5] that the length set of every infinite language in $CD_\infty(f)$, for $f \in \{= k, \geq k \mid k \geq 1\}$, contains an infinite arithmetical progression. This implies that $L_f(\Gamma)$ is not in $CD_\infty(f)$, for $f \in \{= k, \geq k \mid k \geq 1\}$, which concludes the proof.

□

Our proof of the above theorem holds for $n \geq 10$. The question: "what is the smallest n for which Theorem 1 holds ?" remains *open*. Of course, the equalities $PCD_1(f) = CD_1(f) = CF$ are true for all f. Moreover, $PCD_2(=1) \subseteq CF$. Indeed, for $\Gamma = (N, T, S, P_1, P_2, \succ)$ with $P_1 \succ P_2$ (the same argument holds for $P_2 \succ P_1$) we may assume that $dom(P_1) \cap dom(P_2) = \emptyset$ (the rules $A \rightarrow x \in P_2$ with $A \in dom(P_1)$ can never be used, hence they can be eliminated). Thus $L_{=1}(\Gamma) = L(G)$ for $G = (N, T, S, P_1 \cup P_2)$ (the derivations in G and in Γ are the same up to a change of the order of using the rules).

The above language $\{a^{2^n+1} \mid n \geq 1\}$ is probably not in the family MAT (it is conjectured already in [11] that the one-letter matrix languages are regular). Since $CD_\infty(f) \subseteq MAT$ for all f as in Theorem 1 (and in some cases, $CD_\infty(f) = CF$), the increase in generative power by adding priorities is quite considerable for those derivation modes. Hence it is somewhat surprising that for the t mode of derivation adding a priority relation does not increase the generative power.

Theorem 2 $PCD_\infty(t) = CD_\infty(t)$.

Proof. We have to prove only the inclusion $\subseteq$.

For a pcd grammar system $\Gamma = (N, T, S, P_1, \ldots, P_n, \succ)$, we construct the cd grammar system Γ' as follows.

$$\Gamma' = (N', T, S', P_0, P_1', P_1'', P_2', P_2'', \ldots, P_n', P_n'', P_{n+1}),$$

$$N' = N \cup \{S, X, \#\} \cup \{X_i \mid 1 \leq i \leq n\},$$
$$P_0 = \{S' \to SX\} \cup \{X_i \to X \mid 1 \leq i \leq n\},$$
$$P'_i = P_i \cup \{X \to \#\} \cup \{X_j \to \# \mid 1 \leq j \leq n, j \neq i\}, 1 \leq i \leq n,$$
$$P''_i = \{X \to X_i\} \cup \{A \to \# \mid B \in dom(P_j), P_j \succ P_i, 1 \leq j \leq n\}, 1 \leq i \leq n,$$
$$P_{n+1} = \{X \to \lambda\} \cup \{A \to \# \mid A \in N\}.$$

Once introduced in a sentential form, the symbol $\#$ cannot be removed (it is a "trap-symbol"). The symbols $X_1, \dots, X_n$ identify the components $P_1, \dots, P_n$ of Γ. In the presence of X_i the component P_i will be simulated by P'_i and X_i can appear (introduced by P''_i) only when no component P_j with $P_j \succ P_i$ is applicable to the current string.

Let us see how these principles work in Γ' by examining in some detail a derivation. Consider a sentential form wX (initially we have $w = S$, obtained after using P_0, which is the only component which can be applied to S'). The component P_{n+1} can be used only if $w \in T^*$ – hence only as the final step of the derivation. A component P'_i introduces the trap-symbol $\#$. If P_i is maximal with respect to the relation $\succ$ among the components which can be applied to w, then P''_i can be used without blocking the derivation; it changes X into X_i, thus leading to wX_i. Now to a string wX_i we can apply either P_0, replacing again X_i with X (hence not achieving anything) or the component P'_i, which will simulate the application of P_i to w. The string $w'X_i$ obtained in this way can be rewritten only by P_0, which leads to $w'X$, and so the process can be iterated. In the presence of X_i, every component $P'_j, j \neq i$, will introduce the trap-symbol. Consequently, $L_t(\Gamma) = L_t(\Gamma')$. (Note that the λ-rule in P_{n+1} causes no problem, because $CD_\infty(t) = CD^\lambda_\infty(t) = ET0L$.) □

Let us return to families $PCD_\infty(f)$ for $f \neq t$. It is quite natural to compare these families with ORD, the family of languages generated by ordered grammars. Given an ordered grammar $G = (N, T, S, P, \succ)$, it is obvious that we have $L(G) = L_{=1}(\Gamma) = L_{\leq 1}(\Gamma)$ where Γ is a pcd grammar system obtained by considering each rule of P as a separate component and the relation $\succ$ defined as in G. Therefore $ORD \subseteq PCD_\infty(= 1) = PCD_\infty(\leq 1)$. This implies that the families $PCD_\infty(f)$, $f \in \{= 1, \leq 1\}$, strictly include $ET0L$ (and hence $CD_\infty(t)$).

A similar result is obtained for the $= k$ and $\leq k$ modes of derivation for all $k \geq 1$.

Theorem 3 $ORD \subseteq PCD_\infty(f), f \in \{\leq k, = k \mid k \geq 1\}$.

Proof. For $k = 1$ the statement follows by the argument as above. Consider $k \geq 2$. Let $G = (N, T, S, P, \succ)$ be an ordered grammar with

$$P = \{r_1, \dots r_n\},\ r_i : A_i \to x_i,\ 1 \leq i \leq n,\ n \geq 1.$$

We construct the pcd grammar system

$$\Gamma = (N', T, S, P_0, P_1, P_2, \dots, P_n, \succ),$$

where

$$N' = N \cup \{A_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k-1\},$$
$$P_0 = \{A_{i,j} \to A_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k-1\},$$
$$P_i = \{A_i \to A_{i,1}, A_{i,1} \to A_{i,2}, \dots, A_{i,k-2} \to A_{i,k-1}, A_{i,k-1} \to x_i\},\ 1 \leq i \leq n,$$

and

$$\begin{array}{ll} P_0 \succ P_i & \text{for all } 1 \leq i \leq n, \\ P_i \succ P_j & \text{iff } r_i \succ r_j \text{ in } G. \end{array}$$

Then $L(G) = L_{=k}(\Gamma) = L_{\leq k}(\Gamma)$.

Indeed, if we have a sentential form w to which P_0 can be applied, then on the one hand no other component of Γ can be used for rewriting w, while on the other hand the use of P_0 does not change the string w. Consequently, the derivation is blocked, and P_0 is a trap-component. As P_0 can be applied whenever any of the symbols in $N' - N$ is present, it follows that the components $P_i, 1 \leq i \leq n$, upon completing their derivations cannot produce strings containing symbols in $N' - N$. This implies that using a component $P_i, 1 \leq i \leq n$, in $\leq k$ or in $= k$ mode of derivation, means to use all the rules from P_i exactly once, hence to replace an occurrence of A_i first by $A_{i,1}$, then by $A_{i,2}, \ldots$, then by $A_{i,k-1}$, and finally by x_i. This is exactly the effect of using the rule $A_i \to x_i$. As the priority relation among the components P_i of Γ corresponds to the order relation among the rules of G, the equalities $L(G) = L_{=k}(\Gamma) = L_{\leq k}(\Gamma)$ follow.

□

It is an *open* question whether or not Theorem 3 holds also for the $\geq k$ mode of derivation.

We will now demonstrate that all the families $PCD_\infty(f)$ with $f \neq t$, are included in MAT_{ac}. In view of the strong generative power of matrix grammars with appearance checking this inclusion is somewhat expected, however is really cumbersome to write the detailed proof of this result. This is due to the fact that we have to check whether or not all the components greater than a given component (in the sense of the $\succ$ relation) are applicable to a given string in a specified mode of derivation. This is easy for modes $*, = 1, \geq 1, \leq k$, for all k, but much more difficult for the cases $= k, \geq k$, for $k \geq 2$, when all combinations of k rules in a component must be checked. For this reason the proof of the following theorem will be rather sketchy, but certainly containing enough information so that the interested reader may complete it to a detalied proof.

Theorem 4 $PCD_\infty(f) \subseteq MAT_{ac},\ f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$.

Proof. (1) For $f \in \{*\} \cup \{\leq k \mid k \geq 1\}$, consider a system $\Gamma = (N, T, S, P_1, \ldots, P_n, \succ)$, and construct the matrix grammar

$$\begin{aligned} G = {} & (N', T \cup \{c\}, S', M, F), \\ & N' = N \cup \{X, S', \#\} \cup \{[i,j] \mid 1 \leq i \leq n, 0 \leq j \leq k\}, \\ & M = \{(S' \to SX)\} \cup \\ & \quad \cup \{(X \to [i,0], A_1 \to \#, \ldots, A_s \to \#) \mid \{A_1, \ldots, A_s\} = \\ & \qquad \{A \in dom(P_j) \mid P_j \succ P_i, 1 \leq j \leq n\}, 1 \leq i \leq n\} \cup \\ & \quad \cup \{([i,j] \to [i,j+1], A \to x) \mid A \to x \in P_i, 1 \leq i \leq n, \\ & \qquad 0 \leq j \leq k-1\} \cup \\ & \quad \cup \{([i,j] \to X) \mid 1 \leq i \leq n, 0 \leq j \leq k\} \cup \\ & \quad \cup \{(X \to c)\}, \\ & F \text{ contains all rules } A \to \# \ (\# \text{ is a trap} - \text{symbol}). \end{aligned}$$

We have $L(G) = L_{\leq k}(\Gamma)\{c\}$. The first component of nonterminals $[i,j]$ specifies the simulated component, while the second one counts the used rules. The symbol X is replaced by $[i,0]$, starting the simulation of P_i, only when no component P_j with $P_j \succ P_i$ can use at least one of its rules for rewriting the current sentential form. After using j rules of P_i, for some $0 \leq j \leq k$, the symbol $[i,j]$ can be replaced by X and another component of Γ can be simulated.

If all symbols $[i,j]$ are replaced by $[i]$, and no reference is made to the number of used rules, then we obtain $L(G) = L_*(\Gamma)\{c\}$. As MAT_{ac} is closed under restricted morphisms, the new symbol c can be erased, and so $L_f(\Gamma) \in MAT_{ac}$, for $f \in \{*\} \cup \{\leq k \mid k \geq 1\}$.

(2) In the case of the derivation mode $= k$, starting from $\Gamma = (N, T, S, P_1, \ldots \ldots, P_n, \succ)$ with $N = \{A_1, \ldots, A_s\}$, we shall use again the closure of the family MAT_{ac} under restricted morphisms. We construct a matrix grammar G with appearance checking working as follows. The new axiom S' introduces a string SX, where S is the axiom of Γ and X a control symbol; X or its variants will be present during all derivation steps. Moreover, for each symbol $A \in N$ we have its *copy* A_c. In order to be able to check whether a component P_j can be applied to the current string w, we introduce a copy of each nonterminal appearing in w, obtaining in this way a string w_c scattered among the symbols of w; we try to use the rules of P_j on w_c so that the original string w is not destroyed.

Here is a "sub-routine" for such a copying, called for by the control symbol X_c (here and in the matrices below, $\#$ is a trap-symbol):

$$\begin{aligned}
&(X_c \to X_c, A \to A'A_c), \text{ for each } A \in N,\\
&(X_c \to X', A_1 \to \#, \ldots, A_s \to \#),\\
&(X' \to X', A' \to A), \text{ for each } A \in N,\\
&(X' \to X'', A'_1 \to \#, \ldots, A'_s \to \#).
\end{aligned}$$

(In the presence of X_c, each symbol $A \in N$ is replaced by $A'A_c$; when all symbols $A \in N$ have been so replaced, X_c can be replaced by X', and then in the presence of X' each A' is rewritten back to A; when this has been completed, X' is removed and the symbol X'' is introduced.)

Then, the control symbol X'' will guess a component, say P_i, to be used, by changing to X_i. Now all the components $P_j \succ P_i$ must be tested and if any of them can be used, then the derivation is blocked. This can be done as follows.

Having an ordered list $GR(P_i) = (P_{j_1}, \ldots, P_{j_{t_i}})$, of components that are "greater" than P_i, we inspect them in this order $P_{j_1}, \ldots, P_{j_{t_i}}$. If some P_{j_r} is applicable, then the derivation is blocked; if P_{j_r} is not applicable, then we pass to $P_{j_{r+1}}$. Finally when also $P_{j_{t_i}}$ is not applicable, the control symbol is changed to some Y_i, which leads to the simulation of P_i. This is done as in the $\leq k$ mode, introducing a counter which terminates the simulation of P_i when exactly k rules were used; then again the "general controller" X is introduced in order to start the simulation of another component. The derivation terminates (the control symbol, the copy symbols and their variants are replaced by the new terminal c) when no nonterminal from N is present in the current string.

Hence to complete the proof of the theorem for the $= k$ mode we have to show how to test whether or not a given component P_j is applicable in the $= k$ mode to the current string w (hence to the corresponding nonterminal string w_c containing copies of the nonterminals in w).

Consider the set P_j^k of all sequences of k rules in P_j,

$$P_j^k = \{m_1, m_2, \ldots, m_q\}, q = (card(P_j))^k,$$
$$m_l = (A_{l_1} \to x_{l_1}, \ldots, A_{l_k} \to x_{l_k}),\ A_{l_r} \to x_{l_r} \in P_j, 1 \le r \le k.$$

We have to check all these sequences – if at least one of them is applicable, then P_j is applicable. By appropriately modifying the control symbol, we check, one by one all the sequences $m_1, m_2, \ldots, m_q$. If some m_l is applicable, then the derivation is blocked by introducing the trap-symbol #; if m_l is not applicable, then we pass to m_{l+1}. Finally, when m_q is not applicable, then we conclude that P_j is not applicable.

We explain now the basic idea behind the checking whether or not some $m_l = (A_{l_1} \to x_{l_1}, \ldots \ldots, A_{l_k} \to x_{l_k})$ is applicable. Assume that the current string contains the control symbol $[i, j, l]$ (meaning: "for using P_i, we must be sure that $P_j \succ P_i$ is not applicable, and we will try the sequence m_l in P_j^k"). Consider the set of all sequences $C(m_l)$ associated with m_l as follows

$$C(m_l) = \{(A_{l_1} \to \alpha_{l_1}, \ldots, A_{l_k} \to \alpha_{l_k}) \mid \alpha_{l_r} \in \{x_{l_r}, \#\},$$
$$1 \le r \le k, \text{ and at least for one } r \text{ we have } \alpha_{l_r} = \#\}.$$

If m_l is applicable, then each sequence in $C(m_l)$, considered as a matrix with the rules $A \to \#$ used in the appearance checking manner, will introduce at least one occurrence of #. Conversely, if m_l is not applicable, then there is exactly one sequence in $C(m_l)$ which can be used without introducing the trap-symbol.

Indeed, take a rule $A_{l_r} \to x_{l_r}$. If it is applicable in m_l to the current string w, then it is also applicable in all sequences of $C(m_l)$, whether or not it is replaced by $A_{l_r} \to \#$. If it is applicable in m_l to a symbol not in w, but introduced by a previous rule $A_{l_p} \to x_{l_p}$, with x_{l_p} containing A_{l_r}, then we examine this rule, $A_{l_p} \to x_{l_p}$. If it remains unchanged in a sequence of $C(m_l)$, then it introduces A_{l_r}, hence also $A_{l_r} \to \alpha_{l_r}$ is applicable, introducing # when $\alpha_{l_r} = \#$. If it is replaced by $A_{l_p} \to \#$, then the above argument can be iterated again, considering two possible cases for A_{l_p}: either it appears in w or it is introduced by a previous rule. Since each sequence in $C(m_l)$ contains at least one rule $A_{l_g} \to \#$ whenever m_l is applicable, at least one # is introduced. When m_l is not applicable, at least one of its rules is not applicable. If we replace all not applicable rules by $A_{l_g} \to \#$, then we obtain a sequence in $C(m_l)$ which can be applied in the appearance checking mode without introducing the trap-symbol.

Consequently, for checking whether or not m_l is applicable it suffices to guess which sequence in $C(m_l)$ is applicable in the appearance checking mode (if the guessing is incorrect, then the derivation is blocked).

To this aim, the current control symbol $[i, j, l]$ is non-deterministicaly replaced by $[i, j, l; h]$, where h is the label of a sequence $(A_{l_1} \to \alpha_{l_1}, \ldots, A_{l_k} \to \alpha_{l_k})$ in $C(m_l)$. Here is the "sub-routine" for this step:

$$([i, j, l; h] \to [i, j, l; OK], (A_{l_1})_c \to (\alpha_{l_1})_c, \ldots, (A_{l_k})_c \to (\alpha_{l_k})_c),$$

where $(\alpha_{l_r})_c = \#$ if $\alpha_{l_r} = \#$ and it is obtained by replacing in x_{l_r} (whenever $\alpha_{l_r} = x_{l_r}$) all nonterminals $B \in N$ by their copies B_c and removing all the terminals; the terminal rules $B \to x$ are replaced by $B_c \to D$, where D is a special nonterminal (we do not introduce λ-rules).

Then, because the copy symbols have been altered, we replace all of them by D and for checking the next sequence in P_j^k (namely m_{l+1}) we produce a new series of copy symbols, using the following matrices:

$$
\begin{array}{l}
([i,j,l;OK] \to [i,j,l;OK], A_c \to D),\ A \in N, \\
([i,j,l;OK] \to [i,j,l;copy], (A_1)_c \to \#, \ldots, (A_s)_c \to \#), \\
([i,j,l;copy] \to [i,j,l;copy], A \to A', D \to A_c),\ A \in N, \\
([i,j,l;copy] \to [i,j,l;copy], A \to A'A_c),\ A \in N, \\
([i,j,l;copy] \to [i,j,l;copy'], A_1 \to \#, \ldots, A_s \to \#), \\
([i,j,l;copy'] \to [i,j,l;copy'], A' \to A),\ A \in N, \\
([i,j,l;copy'] \to [i,j,l+1], A'_1 \to \#, \ldots, A'_s \to \#).
\end{array}
$$

In this way the new copies of nonterminals in the current string of Γ as simulated in G use "the places" of the old copies (the order is not relevant if matrices are used only for testing their applicability); new places for copies of nonterminals are introduced only when we do not have enough occurrences of the "place holder" symbol D (this is important when we pass from the simulation of P_i, which can introduce new nonterminals, to the simulation of another component). Therefore the length of the string is not increased more than by a factor of three (more exactly, for a string $x \in L_{=k}(\Gamma)$ we can obtain in $L(G)$ a string with the length less than $2|x|+1$).

We believe that the description of G given above allows one to give a formal (quite tedious) construction of a matrix grammar G with appearance checking such that $L_{=k}(\Gamma) = h(L(G))$, where $h : (T \cup \{c\})^* \longrightarrow T^*$ is a 3-bounded morphism defined by $h(a) = a$ for $a \in T$, and $h(c) = \lambda$. Consequently, $L_{=k}(\Gamma) \in MAT_{ac}$.

The modifications for the $\geq k$ mode of derivation concern only the counting of rules used in P_i whenever the use of P_i is permitted. (A component $P_j \succ P_i$ is applicable in the $\geq k$ mode if and only if it is applicable in the $= k$ mode, hence the "checking part" of the construction from the above proof remains unchanged.) □

5 The power of hierarchical grammar systems

We begin by pointing out the relations which follow directly from definitions:

Lemma 2 $CF = H_0CD(f) \subseteq H_1CD(f) = CD_\infty(f) \subseteq H_2CD(f) \subseteq H_3CD(f) \subseteq \ldots, f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$.

For many derivation modes, this hierarchy is finite.

Theorem 5 $H_hCD(t) = H_1CD(t)$, *for each* $h \geq 1$.

Proof. We only have to prove the inclusion $H_hCD(t) \subseteq H_1CD(t)$, and to this aim it suffices to show that $H_2CD(t) \subseteq H_1CD(t)$ (by induction: having a system of arbitrary depth $h \geq 2$, if its subsystems of depth $h-1$ can be reduced to systems of depth 1, then we replace them by such systems and obtain in this way a system of depth 2 equivalent with the initial one; then again using the reduction from depth 2 to depth 1, we prove the theorem).

Hence consider a system $\Gamma = (N, T, S, \gamma_1, \ldots, \gamma_m)$ of depth 2, with $\gamma_i = \{\gamma_{i,1}, \ldots \ldots, \gamma_{i,r_i}\}, r_i \geq 1$, $1 \leq i \leq m$, where $\gamma_{i,j}$ are sets of context-free rules over $N \cup T$. We construct the system Γ', of depth 1, with the nonterminal alphabet

$$N' = \{S, \#\} \cup \{[A, i] \mid A \in N, 1 \leq i \leq m\},$$

the terminal alphabet T, the axiom S', and the following components:

$$\begin{aligned} P_1 &= \{S' \rightarrow [S,1],\ S' \rightarrow [S,2], \ldots, S' \rightarrow [S,m]\}, \\ P_{i,j} &= \{[A,i] \rightarrow h_i(x) \mid A \rightarrow x \in \gamma_{i,j}\},\ 1 \leq i \leq m, 1 \leq j \leq r_i, \\ P'_{i,j} &= \{[A,i] \rightarrow [A,j] \mid A \in N - (\cup_{s=1}^{r_i} dom(\gamma_{i,s})) \cup \\ &\qquad \cup \{[A,i] \rightarrow \# \mid A \in \cup_{s=1}^{r_i} dom(\gamma_{i,s})\},\ 1 \leq i,j \leq m, i \neq j, \end{aligned}$$

where, for each $1 \leq i \leq m$, $h_i : (N \cup T)^* \longrightarrow (N' \cup T)^*$ is the morphism defined by $h_i(A) = [A, i]$ for all $A \in N$, and $h_i(a) = a$ for all $a \in T$.

Each derivation in Γ' begins by a rule $S' \rightarrow [S, i]$, which selects a component γ_i of Γ which is simulated first. Assume we use now a component $P_{i,j}, 1 \leq j \leq r_i$. All the introduced nonterminals will be of the form $[A, i], A \in N$. The derivation will be maximal in $P_{i,j}$, hence it corresponds to a maximal derivation in $\gamma_{i,j}$. After finishing the derivation in $P_{i,j}$, another component $P_{i,s}$ for $1 \leq s \leq r_i$, can be used, and so on. At each moment, all the nonterminals present in the current string are of the form $[A, i]$, for the chosen i. When no component $P_{i,j}, 1 \leq j \leq r_i$, can be used (this corresponds to a maximal derivation in γ_i), a component $P'_{i,j}, j \neq i$, of Γ' can be used. It changes all nonterminals in the sentential form from $[A, i]$ to $[A, j]$. A component $P'_{i,j}, i \neq j$, can be used without blocking the derivation only when no derivation step in $P_{i,s}, 1 \leq s \leq r_i$, can be done, that is the corresponding derivation in γ_i is maximal (otherwise a rule $[A, i] \rightarrow \#, A \in dom(\gamma_{i,s})$, for some $1 \leq s \leq r_i$, can be used, which introduces the trap-symbol $\#$). Consequently, the terminal derivations in Γ' simulate derivations in Γ.

Conversely, it is obvious that each derivation in Γ can be simulated in Γ'.

Consequently, $L_t(\Gamma) = L_t(\Gamma')$, that is $H_2CD(t) \subseteq H_1CD(t) = CD_\infty(t)$, which concludes the proof.

□

Theorem 6 *$H_hCD(f) = H_1CD(f) = CF$, for $f \in \{*, = 1, \geq 1\} \cup \{\leq k \mid k \geq 1\}$, and $h \geq 1$.*

Proof. We proceed again as in the previous proof, reducing the problem to the inclusion $H_2CD(f) \subseteq H_1CD(f)$; because we know that $H_1CD(f) = CF$, for f as in the statement of the theorem, we shall prove the relation $H_2CD(f) \subseteq CF$.

Consider a system of depth 2, $\Gamma = (N, T, S, \gamma_1, \ldots, \gamma_m)$, with $\gamma_i = \{\gamma_{i,1}, \ldots \ldots, \gamma_{i,s_i}\}$, for each $1 \leq i \leq m$, where $\gamma_{i,j}$ is a set of context-free rules, $1 \leq j \leq s_i$. Let G be the context-free grammar $(N, T, S, \{A \rightarrow x \mid A \rightarrow x \in \gamma_{i,j}, 1 \leq i \leq m, 1 \leq j \leq s_i\})$.

Every derivation in Γ ammounts to the use of rules from sets $\gamma_{i,j}$, hence the inclusion $L_f(\Gamma) \subseteq L(G)$ is obvious (and actually holds for all modes of derivation, and not only for the modes f as in the statement of the theorem). Conversely, every derivation in G is correct with respect to the f mode in Γ, because we can reproduce all derivations in G as $= 1$ derivations in Γ. Consequently, $L(G) = L_f(\Gamma)$, that is $L_f(\Gamma) \in CF$. □

It is an *open* problem whether Theorem 6 can also be extended to the derivation modes $= k$ and $\geq k$, for $k \geq 2$. This question seems to be related to the unsolved problems about usual grammar systems concerning (1) the relations between families $CD_\infty(= k)$ and $CD_\infty(= j)$ for $k \neq j$, and (2) the strictness of the inclusions $CD_\infty(\geq k) \subseteq CD_\infty(\geq k+1)$ for $k \geq 2$ (weak inclusions are proved in [3]). In the example from Section 3 we have seen that a derivation in the $= 2$ mode at the level of the system corresponds, in some sense, to a derivation in the $= 4$ mode at the level of components: two rules from the first sub-component and two rules from the second sub-component are used.

We will demonstrate now that the result analogous to Theorem 3 holds for hcd grammar systems.

Theorem 7 *$H_hCD(f) \subseteq MAT$, for all $h \geq 0$ and for $f \in \{= k, \geq k \mid k \geq 2\}$.*

Proof. First of all notice that for each f as in the statement of the theorem, $H_0CD(f) = CF$, and $H_1CD(f) = CD(f)$ – thus (see also the end of Section 2) $H_0CD(f) \subseteq MAT$ and $H_1CD(f) \subseteq MAT$. Hence we may assume that $h \geq 2$.

Let Γ be a hcd grammar system of depth h, $\Gamma = (N, T, S, \gamma_1, \ldots, \gamma_m)$. Using a component γ_i in the $= k$ mode for $k \geq 2$, means to use k of its subsystems. This in turn means that k sub-subsystems are used, and so on until one reaches the level 0 (of sets of rules) where we use k rules from each set chosen by the previous steps. This means that from the sets P_j on the level 0 we use sequences in the sets P_j^k; then "concatenating" such sequences, we obtain sequences corresponding to the next level and so on. The so obtained sequences are matrices of rules, and so the work of Γ in the $= k$ mode can be simulated in a matrix grammar which is defined as follows.

For a sequence of matrices of context-free rules, $m_i = (r_{i,1}, \ldots, r_{i,s_i}), 1 \leq i \leq p$, we define $(m_1, \ldots, m_p) = (r_{i,1}, \ldots, r_{i,s_1}, r_{2,1}, \ldots, r_{2,s_2}, r_{3,1}, \ldots, r_{p,s_p})$, which is again a matrix of rules.

For a set P of context-free rules let $mat(P, k) = P^k$ (all matrices, in all orders and combinations, of k rules in P), and then, for a system $\delta = (N, T, S, \delta_1, \ldots$ $\ldots, \delta_q)$ of depth $h \geq 1$, we define recursively

$$mat(\delta, k) = \{mat(\delta_1, k), mat(\delta_2, k), \ldots, mat(\delta_s, k)\}^k.$$

The matrix grammar $G = (N, T, S, mat(\Gamma, k))$ has the property $L(G) = L_{=k}(\Gamma)$, which proves the inclusion $H_hCD(= k) \subseteq MAT$.

The inclusion $H_hCD(\geq k) \subseteq MAT$ can be obtained in the same way, using the observation that every derivation in a system Γ in the mode $\geq k$ can be decomposed into one or more derivations in the mode $= j$, for $k \leq j \leq 2k - 1$. Therefore, if we define now $mat'(P, k) = \cup_{j=k}^{2k-1} mat(P, j)$ and we modify in the same way the definition of $mat(\delta, k)$, then we obtain a matrix grammar G' generating the language $L_{\geq k}(\Gamma)$.

□

Note that in the above theorem we have dealt with matrix grammars without appearance checking.

References

[1] A. Atanasiu, V. Mitrana, Modular grammars, *Intern. J. Computer Math.*, 30 (1989), 101 - 122.

[2] E. Csuhaj-Varju, J. Dassow, On cooperating distributed grammar systems, *J. Inform. Process. Cybern., EIK*, 26 (1990), 49 – 63.

[3] E. Csuhaj-Varju, J. Dassow, J. Kelemen, Gh. Păun, *Grammar Systems*, Gordon and Breach, 1994.

[4] J. Dassow, Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, 1989.

[5] J. Dassow, Gh. Păun, S. Vicolov, On the generative capacity of certain classes of cooperating grammar systems, *Fundamenta Informaticae*, to appear.

[6] L. Kari, Al. Mateescu, Gh. Păun, A. Salomaa, Teams in cooperating grammar systems, *J. Experimental and Theoretical AI*, to appear.

[7] R. Meersman, G. Rozenberg, Cooperating grammar systems, *Proc. MFCS '78 Symp., LNCS 64*, Springer-Verlag, 1978, 364 – 374.

[8] P. H. Nii, Blackboard systems, in *The Handbook of AI*, vol. 4 (A. Barr, P. R. Cohen, E. A. Feigenbaum, eds.), Addison-Wesley, 1989.

[9] Gh. Păun, G. Rozenberg, Prescribed teams of grammars, *Acta Informatica*, to appear.

[10] G. Rozenberg, A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, 1980.

[11] A. Salomaa, *Formal Languages*, Academic Press, 1973.

Received October 6, 1993

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

Normal Forms and Minimal Keys in the Relational Datamodel*

J. Demetrovics † Vu Duc Thi‡

Abstract

The normalization of relations was introduced by E. F. Codd. The main purpose of normalization is to delete undesired redundancy and anormalies. The most desirable normal forms are second normal form (2NF), third normal form (3NF) and Boyce-Codd normal form (BCNF) that have been investigated in a lot of papers. The concepts of minimal key and prime attribute (recall that an attribute is prime if it belongs to a minimal key, and nonprime otherwise) directly concern 2NF, 3NF and BCNF. This paper investigates connections between these normal forms and sets of minimal keys. Lucchesi and Osborn showed [11] that the problem to decide if an arbitrary attribute is prime is NP-complete for relation scheme. We proved [9] that a set of all nonprime attributes is the intersection of all antikeys (maximal nonkeys) and this prime attribute problem can be solved by polynomial time algorithm for relation. From these results some problems are NP-complete for relation scheme, but for relation these problems are solved by polynomial time algorithms. It is known [5] that a set of all minimal keys of a relation scheme (and a relation) is a Sperner system (sometimes it is called an antichain) and for an arbitrary Sperner system there exists a relation scheme the set of all minimal keys of which is exactly this Sperner system. In this paper the following concepts are introduced.

A Sperner system K is in 2NF (3NF, BCNF, respectively) if for each relation scheme s such that $K_s = K$ then s is in 2NF (3NF, BCNF, respectively), where K_s is a set of all minimal keys of s. This paper gives necessary and sufficient conditions for an arbitrary Sperner system is in 2NF or 3NF or BCNF. We prove that problems of deciding whether K_s is in 2NF (3NF, respectively) are NP-complete. However, we show that if a relation scheme is changed to a relation then these problems are solved by polynomial time algorithms. We give a new characterization of relations and relation schemes that are uniquely determined by their minimal keys. From this characterization we give a polynomial time algorithm deciding whether an arbitrary

*Research supported by Hungarian Foundation for Scientific Research Grant 2575.

†Computer and Automation Institute Hungarian Academy of Sciences P.O.Box 63, Budapest, Hungary, H-1502

‡Centre for Systems and Management Research National Centre for Scientific Research of Vietnam P.O.B.626 Boho, Hanoi 10000 Vietnam

relation is uniquely determined by its set of all minimal keys. Osborn [14] gives a polynomial time algorithm testing BCNF property of a given relation scheme. This paper gives a polynomial time algorithm recognizing BCNF and finding a set of all minimal keys and a minimum cover if a given relation scheme is in BCNF.

Key Words and Phrases: database, relation, relational datamodel, functional dependency, relation scheme, second normal form, third normal form, Boyce-Codd normal form, closure, closed set, minimal generator, key, minimal key, antikey.

1 Introduction

Let us give some necessary definitions and results that are used in next section.

Definition 1.1 *Let $R = \{a_1, \ldots, a_n\}$ be a nonempty finite set of attributes, $r = \{h_1, \ldots, h_m\}$ be a relation over R, and $A, B \subseteq R$. Then we say that B functionally depends on A in r (denoted $A \xrightarrow[r]{f} B$) iff*

$$(\forall h_i, h_j \in r)((\forall a \in A)(h_i(a) = h_j(a)) \Longrightarrow (\forall b \in B)(h_i(b) = h_j(b))).$$

Let $F_r = \{(A, B) : A, B \subseteq R,\ A \xrightarrow[r]{f} B\}$. F_r is called the full family of functional dependencies of r. Where we write (A, B) or $A \to B$ for $A \xrightarrow[r]{f} B$ when r, f are clear from the context.

Definition 1.2 *A functional dependency over R is a statement of the form $A \to B$, where $A, B \subseteq R$. The FD $A \to B$ holds in a relation r if $A \xrightarrow[r]{f} B$. We also say that r satisfies the FD $A \to B$.*

Clearly, F_r is a set of all FDs that hold in r.

Definition 1.3 *Let R be a nonempty finite set, and denote $P(R)$ its power set. Let $y \subseteq P(R) \times P(R)$. We say that y is an f-family over R iff for all $A, B, C, D \subseteq R$*

1. $(A, A) \in y$,
2. $(A, B) \in y$, $(B, C) \in y \Longrightarrow (A, C) \in y$,
3. $(A, B) \in y$, $A \subseteq C$, $D \subseteq B \Longrightarrow (C, D) \in y$,
4. $(A, B) \in y$, $(C, D) \in y \Longrightarrow (A \cup C, B \cup D) \in y$. *Clearly, F_r is an f-family over R.*

It is known [1] that if y is an arbitrary f-family, then there is a relation r over R such that $F_r = y$.

Definition 1.4 *A relation scheme s is a pair $< R, F >$, where R is a set of attributes, and F is a set of FDs over R. Let F^+ be a set of all FDs that can be derived from F by the rules in Definition 1.3. Denote $A^+ = \{a : A \to \{a\} \in F^+\}$. A^+ is called the closure of A over s. It is clear that $A \to B \in F^+$ iff $B \subseteq A^+$.*

It is known [3] that there is a polynomial time algorithm which finds A^+ from A.

Clearly, if $s =< R, F >$ is a relation scheme, then there is a relation r over R such that $F_r = F^+$ (see, [1]). Such a relation is called an Armstrong relation of s. It is obvious that all FDs of s hold in r.

Definition 1.5 *Let r be a relation, $s =< R, F >$ be a relation scheme, y be an f-family over R and $A \subseteq R$. Then A is a key of r (a key of s, a key of y) if $A \xrightarrow[r]{f} R$ ($A \to R \in F^+, (A, R) \in y$). A is a minimal key of $r(s, y)$ if A is a key of $r(s, y)$, and any proper subset of A is not a key of $r(s, y)$. Denote $K_r, (K_s, K_y)$ the set of all minimal keys of $r(s, y)$. Clearly, K_r, K_s, K_y are Sperner systems over R.*

Definition 1.6 *Let K be a Sperner system over R. We define the set of antikeys of K, denoted by K^{-1}, as follows:*

$$K^{-1} = \{A \subset R : (B \in K) \Longrightarrow (B \not\subseteq A) \, and \, (A \subset C) \Longrightarrow (\exists B \in K)(B \subseteq C)\}$$

It is easy to see that K^{-1} is also a Sperner system over R.

It is known [5] that if K is an arbitrary Sperner system over R then there is a relation scheme s such that $K_s = K$.

In this paper we always assume that if a Sperner system plays the role of the set of minimal keys (antikeys), then this Sperner system is not empty (doesn't contain R). We consider the comparison of two attributes as an elementary step of algorithms. Thus, if we assume that subsets of R are represented as sorted lists of attributes, then a Boolean operation on two subsets of requires at most $|R|$ elementary steps.

Definition 1.7 *Let $I \subseteq P(R)$, $R \in I$, and $A, B \in I \Longrightarrow A \cap B \in I$. Let $M \subseteq P(R)$. Denote $M^+ = \{\cap M' : M' \subseteq M\}$. We say that M is a generator of I iff $M^+ = I$. Note that $R \in M^+$ but not in M, since it is the intersection of the empty collection of sets.*

Denote $N = \{A \in I : A \neq \cap\{A' \in I : A \subset A'\}\}$.

In [6] it is proved that N is the unique minimal generator of I. Thus, for any generator N' of I we obtain $N \subseteq N'$.

Definition 1.8 *Let r be a relation over R, and E_r the equality set of r, i.e. $E_r = \{E_{ij} : 1 \leq i < j \leq |r|\}$, where $E_{ij} = \{a \in R : h_i(a) = h_j(a)\}$. Let $T_r = \{A \in P(R) : \exists E_{ij} = A, \nexists E_{pq} : A \subset E_{pq}\}$. Then T_r is called the maximal equality system of r.*

Definition 1.9 *Let r be a relation, and K a Sperner system over R. We say that r represents K iff $K_r = K$. The following theorem is known ([8]).*

Definition 1.10 *Let K be a non-empty Sperner system and r a relation over R. Then r represents K iff $K^{-1} = T_r$, where T_r is the maximal equality system of r.*

Definition 1.11 *Let $s =< R, F >$ be a relation scheme over R. We say that an attribute a is prime if it belongs to a minimal key of s, and nonprime otherwise. $s =< R, F >$ is in*

1. *2NF if $A \to \{a\} \notin F^+$ for each $K \in K_s$, $A \subset K$, $a \notin A$, and a is nonprime.*

2. 3NF if $A \to \{a\} \notin F^+$ for $A^+ \neq R$, $a \notin A$, a is nonprime.

3. BCNF if $A \to \{a\} \notin F^+$ for $A^+ \neq R$, $a \notin A$.

Clearly, if s is in BCNF (3NF, respectively) then s is in 3NF (2NF, respectively).

If a relation scheme is changed to a relation we have the definition of 2NF, 3NF and BCNF for relation.

Definition 1.12 [4] *Let P be a set of all f-families over R. An ordering over P is defined as follows:*

For $F, F' \in P$ let $F \leq F'$ iff for all $A \subseteq R$, $H_{F'}(A) \subseteq H_F(A)$. where $H_F(A) = \{a \in R: (A, \{a\}) \in F\}$.

Theorem 1.13 [7] *Let K be a Sperner system over R. Let*

$$L(A) = \begin{cases} \bigcap_{A \subseteq B} & B \\ if \exists B \in K^{-1} : A \subseteq B & R \text{ otherwise} \end{cases}$$

and $F = \{(C, D) : D \subseteq L(C)\}$.

Then F is an f-family over R, $H_F = L$, and $K_F = K$. If F' is an arbitrary f-family over R such that $K_{F'} = K$ then $F \leq F'$ holds.

2 Results

In this section we give some results related to 2NF, 3NF, BCNF and sets of minimal keys.

Definition 2.1 *Let K be a Sperner system over R. We say that K is in 2NF (3NF, BCNF, respectively) if for every relation scheme $s = < R, F >$ such that $K_s = K$ then s is in 2NF (3NF, BCNF, respectively).*

Now we give a necessary and sufficient condition for an arbitrary Sperner system is in 2NF.

Let K be a Sperner system over R. Denote $K_p = \{a \in R: \exists A \in K: a \in A\}$, and $K_n = R - K_p$. K_p (K_n) is called the set of prime (nonprime) attributes of K.

Given a relation scheme $s = < R, F >$, we say that a functional dependency $A \to B \in F$ is redundant if either $A = B$ or there is $C \to D \in F$ such that $C \subsetneq A$.

Theorem 2.2 *Let K be a Sperner system over R. Then K is in 2NF if and only if $K_n = \emptyset$.*

Proof. According to definitions of 2NF relation, 2NF Sperner system and K_n we can see that if $K_n = \emptyset$ then K is in 2NF.

Now, assume that K is in 2NF. Denote K^{-1} the set of all antikeys of K. From K, K^{-1} we construct the following relation scheme.

For each $A \subset R$ there is $B \in K^{-1}$ such that $A \subseteq B$. Denote $C = \cap\{B \in K^{-1}: A \subseteq B\}$. We set $A \to C$. Denote T the set of all such functional dependencies. Set $F = \{E \to R: E \in K\} \cup (T - Q)$, where $Q = \{X \to Y \in T: X \to Y$ is a redundant functional dependency $\}$. From Theorem 1.13 and definition of Sperner system we obtain $K_s = K$. Clearly, for each arbitrary relation scheme $s' = < R, F' >$ such that $K_{s'} = K$ and $A \subseteq R$ we have $A^+_{s'} \subseteq A^+_s$, where $A^+_{s'} = \{a: A \to \{a\} \in F'^+\}$. We showed [9] that K_n is the intersection of all antikeys of K. Based on the construction of $s =< R, F >$ and according to definition of 2NF Sperner system we obtain $K_n = \emptyset$. Our proof is complete.

It is easy to see that a 3NF relation scheme is in 2NF and if a set of all non-prime of arbitrary relation scheme is empty then this relation scheme is in 3NF. Consequently, Theorem 2.2 immediately implies the following corollary.

Corollary 2.3 *Let K be a Sperner system over R. Then K is in 3NF if and only if $K_n = \emptyset$.*

Definition 2.4 *Let K be a Sperner system over R. We say that K is unique if K uniquely determines the relation scheme $s =< R, F >$, i. e. for every relation scheme $s' =< R, F' >$ such that $K_{s'} = K$ we have $F^+ = F'^+$.*

From definition of BCNF Sperner system and Definition 2.4 we obtain

Proposition 2.5 *K is in BCNF iff K is unique.*

Now we introduce the following problem.

Theorem 2.6 *The following problem is NP-complete:*

Given a relation scheme s, decide whether K_s is in 2NF.

Proof. For each $a \in R$ we nondeterministically choose a subset B of R such that $a \in B$. By an algorithm finding the closure of B over s (see [3]) and based on definition of minimal key we decide whether B is a minimal key of s. From this we can decide whether a is prime of s. According to Theorem 2.2 if for every $a \in R$ a is prime then K_s is in 2NF, in the converse case K isn't in 2NF. It is obvious that this algorithm is nondeterministic polynomial. Thus, our problem lies in NP.

Now we shall show that our problem is NP-hard. It is known [11] that the prime attribute problem for relation scheme is NP-complete. Now we prove that this problem is polynomially reducible to our problem.

Let $s' =< P, F' >$ be a relation scheme over P, and $a \in P$. Without loss of generality we assume that P is not a minimal key of s', i.e. if $A \in K_{s'}$ then $A \subset P$. By a polynomial time algorithm finding a minimal key of relation scheme (see [11]) we can find a minimal key Q of s' from P and F'. Denote $T = \{l : l \in P - Q,\ \{l\} \rightarrow P \notin F'^+\}$. Assume that $T = \{a_1, \ldots, a_t\}$. Now we construct the relation scheme $s =< R, F >$ as follows:

$R = P \cup \{b, c, d, e_1, \ldots, e_{t-1}\}$, where $b, c, d, e_1, \ldots, e_{t-1} \notin P$ and F contains F' and the following functional dependencies:

- $\{b\} \rightarrow \{a\}$,
- $\{c, d\} \rightarrow \{b\}$,
- $Q \cup \{c\} \rightarrow R$,
- $Q \cup \{b\} \rightarrow Q \cup \{c\}$,
- $\{\{a_i, a_{i+1}, e_i\} \rightarrow R : 1 \leq i \leq t-1\}$.

It can be seen that s is constructed in polynomial time in the sizes of P and F'. According to the construction of $s =< R, F >$ and definition of minimal key and by $Q \cup \{c\} \rightarrow R$, for all $A \in K_{s'}$ we have $A \cup \{c\} \in K_s(1)$. Based on $Q \cup \{b\} \rightarrow Q \cup \{c\}$ and $\{b\} \rightarrow \{a\}$ if $A \in K_{s'}$ we obtain $(A - a) \cup \{b\} \in K_s(2)$. By $\{\{a_i, a_{i+1}, e_i\} \rightarrow R : 1 \leq i \leq t-1\}$ we have $\{a_i, a_{i+1}, e_i\}(1 \leq i \leq t-1) \in K_s$. From this and (1) $\forall a' \in P, b, c, e_1, \ldots, e_{t-1}$ are prime attributes of s. According to the construction of s and definition of 3NF relation scheme we can see that s is in 3NF. Now we prove that K_s is in 2NF iff a is prime attribute of s'.

Assume that K_s is in 2NF. According to Theorem 2.2 we can see that d is prime attribute of s. Consequently, there is a minimal key B of s such that $d \in B$.

It can be seen that $a, b, e_1, \ldots, e_{t-1} \notin B$. Since there is only functional dependency $\{c, d\} \rightarrow \{b\}$ the left side of which contains d we obtain $c \in B$. According to $\{b\} \rightarrow \{a\}$, $\{c, d\} \rightarrow \{b\}$ and (2) it is easy to see that $(B \cup a) - \{c, d\} \in K_{s'}$. Thus, a is prime attribute of s'.

Now we assume that K_s is not in 2NF. By Theorem 2.2 d is nonprime attribute of s. If $a \in A : A \in K_{s'}$ then by $\{c, d\} \rightarrow \{a\} \in F^+$ and from (2) $\{c, d\} \cup (A - a) \in K_s$ holds. This conflicts with the fact that d is nonprime attribute of s. Consequently, a is nonprime attribute of s'. The theorem is proved.

Theorem 2.6 immediately implies the following corollary

Corollary 2.7 *The problem of deciding whether K_s is in 3NF is NP-complete for given a relation scheme s.*

It is known [8] that there is a polynomial time algorithm which from a given relation r finds the maximal equality system T_r. Based on Theorem 1.10 and because the set of all nonprime attributes is the intersection of all antikeys we have the following proposition.

Proposition 2.8 *There is an algorithm that for a given relation r decides if K_r is in 2NF or 3NF. The time complexity of this algorithm is polynomial in the sizes of R and r.*

From Theorem 2.2 we immediately obtain the following corollary.

Corollary 2.9 *There is a polynomial time algorithm that decides whether a given Sperner system is in 2NF or 3NF. Let $s = < R, F >$ be a relation scheme over R, K_s is a set of all minimal keys of s. Denote K_s^{-1} the set of all antikeys of s. From Theorem 1.10 we obtain the following corollary.*

Corollary 2.10 *Let $s = < R, F >$ be a relation scheme and r a relation over R. We say that r represents s if $K_r = K_s$. Then r represents s iff $K_s^{-1} = T_r$, where T_r is the maximal equality system of r. In [7] we proved the following theorem.*

Theorem 2.11 *Let $r = \{h_1, \ldots, h_m\}$ be a relation , and F an f-family over R. Then $F_r = F$ iff for every $A \in P(R)$*

$$H_F(A) = \begin{cases} \bigcap_{A \subseteq E_{ij}} & E_{ij} \\ if\ \exists E_{ij} \in E_r : A \subseteq E_{ij} & R \text{ otherwise} \end{cases}$$

where $H_F(A) = \{a \in R: (A, \{a\}) \in F\}$ and E_r is the equality set of r.

Let $s = < R, F >$ be a relation scheme over R. From s we construct $Z(s) = \{X^+ : X \subseteq R\}$, and compute the minimal generator N_s of $Z(s)$. We put

$$T_s = \{A \in N_s : \nexists B \in N_s : A \subset B\}$$

It is known [1] that for a given relation scheme s there is a relation r such that r is an Armstrong relation of s. On the other hand, by Corollary 2.10 and Theorem 2.11 the following proposition is clear

Proposition 2.12 *Let $s = < R, F >$ be a relation scheme over R. Then*

$$K_s^{-1} = T_s.$$

It is known [5] that for given a Sperner system K there exists a relation scheme s (a relation r, respectively) such that $K_s = K$ ($K_r = K$, respectively). We say that s (r, respectively) is unique if K_s (K_r, respectively) uniquely determines s (r, respectively), i.e. K_s (K_r, respectively) is unique.

Now we give a necessary and sufficient condition for given a relation scheme is unique.

Theorem 2.13 *Let $s =< R, F >$ be a relation scheme over R. Then s is unique iff for all $a \in A$, $A \in K_s^{-1}$: $A - a = \cap\{B \in K_s^{-1} : (A - a) \subset B\}$ holds.*

Proof. It is known [4] that a Sperner system K is unique iff for all $B \subseteq A$, $A \in K^{-1}$, B is an intersection of antikeys. Denote $P_s = \{A - a : A \in K_s^{-1}, a \in A\}$.

It can be seen that if $s =< R, F >$ is unique then $B \in P_s$ implies B is an intersection of antikeys, i.e. $B = \cap\{A \in K_s^{-1} : B \subseteq A\}$.

Conversely, assume that for every $B \in P_s$ we have $B = \cap\{A \in K_s^{-1} : B \subseteq A\}(*)$. Now we shall prove the following result : $s =< R, F >$ is in BCNF iff for all $B \in P_s$, $B^+ = B(1)$ holds.

It is easy to see that if s is in BCNF then we obtain (1). Now, we assume that for each $B \in P_s$, $B^+ = B$. Suppose $C \rightarrow \{d\} \in F^+$ and $d \notin C(2)$. If $C^+ \neq R$ then by definition of antikey and Proposition 2.12 there exists an $A \in K_s^{-1}$ such that $C^+ \subseteq A$ and by (2) $d \in A$ holds. Clearly, $C \subseteq A - d$ holds. It is easy to see that $(A - d)^+ \rightarrow \{d\}$ holds. By $A - d \in P_s$ we have $(A - d)^+ \neq A - d$. This conflicts with the fact that $(A - d)^+ = A - d$. Hence, $C^+ = R$ holds, i.e. s is in BCNF.

From this result and according to Proposition 2.12 we have $N_s \subseteq (P_s \cup K_s^{-1})$. It can be seen that s is in BCNF. Based on definition of N_s and Proposition 2.12 $K_s^{-1} \subseteq N_s$ holds. According to (*) we obtain $K_s^{-1} = N_s$. Because s is in BCNF we can see that for all $B \subseteq A$, $A \in K_s^{-1}$: $B^+ = B$ holds. Thus, B is an intersection of antikeys of s. The proof is complete.

According to definition of BCNF Sperner system and based on Theorem 2.13 and Proposition 2.5 we give a necessary and sufficient condition for an arbitrary Sperner system is in BCNF.

Theorem 2.14 *Let K be a Sperner system over R. Then K is in BCNF iff for all $a \in A$, $A \in K^{-1}$: $A - a = \cap\{B \in K^{-1} : (A - a) \subset B\}$ holds.*

By a polynomial time algorithm finding a set of all antikeys of a given relation and according to Theorem 2.13 we obtain the following proposition.

Proposition 2.15 *There exists an algorithm deciding whether a given relation r is unique. The time complexity of this algorithm is polynomial in the sizes of R and r.*

Theorem 2.14 and Proposition 2.15 immediately imply the following

Proposition 2.16 *There exists a polynomial time algorithm deciding whether a set of all minimal keys of a given relation is in BCNF.*

Theorem 2.13 immediately implies the next corolllary.

Corollary 2.17 *Let K be a Sperner system over R. Then there exists a polynomial time algorithm deciding whether a Sperner system H is unique, where $H^{-1} = K$.*

Now we introduce the following problems : Given a relation scheme s (a Sperner system K, respectively), decide whether s (K, respectively) is unique.

It is obvious that these problems are equivalent to the next problems: Given a relation scheme s (a Sperner system K, respectively), decide whether K_s (K, respectively) is in BCNF.

It is unknown that these problems have polynomial time complexity. We consider these problem as interesting open problems.

Osborn [14] gives a polynomial time algorithm deciding whether a relation scheme is in BCNF. It is known [10, 12] that a relation scheme $s =< R, F >$ is in BCNF iff its minimum cover contains functional dependencies $\{K_1 \rightarrow R, \ldots, K_t \rightarrow$

$R\}$, where $K_i (1 \leq i \leq t)$ are minimal keys of s. From this the BCNF property of relation scheme also is recognized in polynomial time.

Let $s = < R, F >$ be a relation scheme over R. From rules (3) and (4) of Definition 1.3 we can see that the functional dependency $A \to \{a_1, \ldots, a_t\}$ is equivalent to the set of functional dependencies $\{A \to \{a_1\}, \ldots, A \to \{a_t\}\}$. Thus, we can assume that F only contains the functional dependencies form $A \to \{a\}$.

Definition 2.18 *Let $s = < R, F >$ be a relation scheme. We say that s is an a-relation scheme over R if $F = \{A \to \{b\}: A \neq b,\ \nexists B : (B \to \{b\})(B \subset A)\}$, where $b \in R$.*

Definition 2.19 *Let $s = < R, F >$ be a relation scheme, $b \in R$. Denote $K_b = \{A \subseteq R: A \to \{b\}, \nexists B: (B \to \{b\})(B \subset A)\}$. K_b is called the family of minimal sets of the attribute b. Clearly, $R \notin K_b$, $\{b\} \in K_b$ and K_b is a Sperner system over R.*

Algorithm 2.20 *(Finding a minimal set of the attribute b)*
Input: Let $s = < R, F >$ be a relation scheme, $A = \{a_1, \ldots, a_t\} \to \{b\}$.
Output: $A' \in K_b$
Step 0: We set $L(0) = A$
Step i+1: Set

$$L(i+1) = \begin{cases} L(i) - a_{i+1} & \text{if } L(i) - a_{i+1} \\ \to \{b\} & L(i) \text{ otherwise.} \end{cases}$$

Then we set $A' = L(t)$.

Lemma 2.21 $L(t) \in K_b$

Proof. By the induction it can be seen that $L(t) \to \{b\}$, and $L(t) \subseteq \ldots \subseteq L(0)$ (1). If $L(t) = b$, then by the definition of the minimal set of attribute b we obtain $L(t) \in K_b$. Now we suppose that there is a B such that $B \subset L(t)$ and $B \neq \emptyset$. Thus, there exists a_j such that $a_j \notin B$, $a_j \in L(t)$. According to the construction of algorithm we have $L(j-1) - a_j \not\to \{b\}$. It is obvious that by (1) we obtain $L(t) - a_j \subseteq L(j-1) - a_j (2)$. It is clear that $B \subseteq L(t) - a_j$. From (1),(2) we have $B \not\to \{b\}$. The lemma is proved.

Clearly, by the linear-time membership algorithm in [3] the time complexity of Algorithm 2.20 is $O(|R|^2|F|)$.

Algorithm 2.22 *(Finding an a-relation scheme)*
Input: Let $s = < R, F >$ be a relation scheme.
Output: an a-relation scheme $s' = < R, F' >$ such that $F'^+ = F^+$.
Step 1: By rules (3) and (4) of Definition 1.3 from s we construct $s" = < R, F" = \{A \to \{b\} : b \in R\} >$ such that $F"^+ = F^+$.
Step 2: For each $A \to \{b\} \in F"$ we use algorithm 2.20 to find a minimal set A' of attribute b over $s"$. Set $F^* = \{A' \to b : \forall b \in R\}$.
Step 3: Set $s' = < R, F' = F^* - Q) >$, where $Q = \{X \to Y \in F^*: X \to Y$ is a redundant functional dependency $\}$.
According to definition of a-relation scheme, based on Definition 2.19 and Lemma 2.21 we can see that s' is an a-relation scheme and $F'^+ = F^+$.
It can be seen that the time complexity of Algorithm 2.22 is polynomial in the sizez of R and F.

Theorem 2.23 *Let $s =< R, F >$ be a relation scheme. Then s is in BCNF if and only if there exists an a-relation scheme $s' =< R, F' >$ such that $F'^{+} = F^{+}$ and for every $A \rightarrow \{b\} \in F'$ $A \in K_{s'}$ holds.*

Proof. Assume that s is in BCNF. By Algorithm 2.22 we can construct an a-relation scheme $s' =< R, F' >$ such that $F'^{+} = F^{+}$. By Step 3 of this algorithm for each $A \rightarrow \{b\} \in F'$ $b \notin A$ holds. Since s' is in BCNF we have $A^{+} = R$. Clearly, if there is a $C \subset A$ such that $C^{+} = R$ then $C \rightarrow \{b\}$ holds. This is a contradiction. Thus, $A \in K_{s'}$ holds.

Conversely, we assume that there is an a-relation scheme $s' =< R, F' >$ such that $F^{+} = F'^{+}$ and for every $A \rightarrow \{b\} \in F'$ $A \in K_{s'}$ holds. By Lemma 3 in [14] s' is in BCNF. Thus, s is in BCNF. Our theorem is proved.

In Theorem 2.23 we set $K = \{A: A \rightarrow \{b\} \in F'\}$. We have the following.

Proposition 2.24 $K = K_s$.

Proof. By definition of BCNF relation scheme $K_{s'} = K_s$ holds. From Theorem 2.23 $K \subseteq K_{s'}$ holds. Suppose $B \in K_{s'}, B \subset R$ and $B \notin K$. Because $K_{s'}$ is a Sperner system over R we can see that $K \cup B$ also is a Sperner system over R. It can be seen that according to definition of a-relation scheme $B^{+} = B$ over s'. This conflicts with the fact that B is a minimal key of s'. The proof is complete.

Theorem 2.23 immediately implies the following.

Proposition 2.25 *Let $s =< R, F >$ be a relation scheme. Then s is in BCNF if and only if there exists an a-relation scheme $s' =< R, F' >$ such that $F'^{+} = F^{+}$ and for every $A \rightarrow \{b\} \in F'$ A is a key of s'.*

It can be seen that based on definition of a-relation scheme, in Proposition 2.25 if $A \rightarrow \{b\} \in F'$ then A is a minimal key of s'.

Clearly, the time complexity of Algorithm 2.22 (finding an a-relation scheme) is polynomial and deciding whether a set of attributes is a key also takes polynomial time, It is known [10, 12] that a relation scheme $s =< R, F >$ is in BCNF iff its minimum cover contains functional dependencies $\{K_1 \rightarrow R, \ldots, K_t \rightarrow R\}$, where $K_i (1 \leq i \leq t)$ are minimal keys of s. We can give a polynomial time algorithm recognizing the BCNF property of arbitrary relation scheme s, and if relation scheme s is in BCNF then this algorithm finds a minimum cover and a set of all minimal keys of s.

Algorithm 2.26 *Input: Let $s =< R, F >$ be a relation scheme.*

Output: Deciding whether s is in BCNF, if s is in BCNF then finding K_s, and an a-relation scheme $s' =< R, F' >$ such that s' is a minimum cover of s.

Step 1: Use Algorithm 2.22 we construct an a-relation scheme $s" =< R, F" = \{A \rightarrow \{b\} : b \in R\} >$ such that $F"^{+} = F^{+}$.

Step 2: If there is an $A \rightarrow \{b\} \in F"$ such that A is not a key of $s"$ then s isn't in BCNF and stop. In the converse case go to the following step.

Step 3: Set $K_s = \{A: A \rightarrow \{b\} \in F"\}$.

Step 4: Denote elements of K_s by $A_1, \ldots, A_t$. Set $F' = \{A_i \rightarrow R : 1 \leq i \leq t\}$.

It can be seen that $s' =< R, F' >$ is a minimum cover of s.

3 Conclusion

Our further research will be devoted to the following problems:
Given a relation scheme s.

1. What is the time complexity of deciding whether s is in unique?

Given a Sperner system K over R.

1. What is the time complexity of deciding whether K is unique?

Acknowledgments

The authors are grateful to Dr. Uhrin Bela for useful comments to the first version of the manuscript.

References

[1] Armstrong W.W. Dependency Structures of Database Relationships. *Information Processing 74,* Holland Publ. Co. (1974) pp. 580-583.

[2] Beeri C., Dowd M., Fagin R., Staman R. On the Structure of Armstrong relations for Functional Dependencies. *J. ACM,* 31 (1984) pp. 30-46.

[3] Beeri C., Bernstein P.A. Computational problems related to the design of normal form relational schemas. *ACM Trans. on Database Syst.* 4 (1979) pp. 30-59.

[4] Burosch G., Demetrovics J., Katona G.O.H. The poset of closures as a model of changing databases, *Order* 4 (1987) pp. 127-142.

[5] Demetrovics J. On the equivalence of candidate keys with Sperner systems. *Acta Cybernetica* 4 (1979) pp. 247-252.

[6] Demetrovics J. Logical and structural Investigation of Relational Datamodel (in Hungarian). *MTA-SZTAKI Tanulmányok,* Budapest, 114 (1980) pp. 1-97.

[7] Demetrovics J., Thi V.D. Some results about functional dependencies. *Acta Cybernetica* 8 (1988) pp. 273-278.

[8] Demetrovics J., Thi V.D. Relations and minimal keys. *Acta Cybernetica* 8 (1988) pp. 279-285.

[9] Demetrovics J., Thi V.D. On Keys in the Relational Datamodel. *EIK* 24 (1988) 10, pp. 515-519.

[10] Gottlob G. , Libkin L. Investigations on Armstrong relations, dependency inference, and excluded functional dependencies. *Acta Cybernetica* 9 (1990) pp. 385-402.

[11] Lucchesi C.L., Osborn S.L. Candidate keys for relations. *J.Comput.Syst. Scien.* 17 (1978) pp. 270-279.

[12] Maier D. Minimum cover in the relational database model. *J.ACM* 27 (1980) pp. 664-674.

[13] Mannila H., Raiha K.J. Design by Example: An Application of Armstrong relations. *J. Comput. Syst. Scien.* 33 (1986) pp. 126-141.

[14] Osborn S.L. Testing for existence of a covering Boyce-Codd normal form. *Inform. Proc. Lett.* 8 (1979) pp. 11-14.

[15] Thi V.D. Investigation on Combinatorial Characterzations Related to Functional Dependency in the Relational Datamodel (in Hungarian). *MTA-SZTAKI Tanulmányok,* Budapest, 191 (1986) pp. 1-157. Ph.D. Dissertation.

[16] Thi V.D. Minimal keys and Antikeys. *Acta Cybernetica* 7 (1986) pp. 361-371.

[17] Thi V.D. On Antikeys in the Relational Datamodel (in Hungarian). *Alkalmazott Matematikai Lapok* 12 (1986) pp. 111-124.

Received March 3, 1993

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

On Strong-Generalized Positive Boolean Dependencies*

Le Thi Thanh †

Abstract

Strong-Generalized Positive Boolean Dependencies are introduced.

Key Words and Phrases: *relation, data base, functional dependency, Boolean dependency, positive Boolean dependency, generalized positive Boolean dependency, Armstrong relation, strong generalized positive Boolean dependency.*

1 Introduction

In the theory of relational databases, connections between functional and multivalued dependencies and a certain fragment of propositional logic have been investigated in several papers.

The full family and the possible mathematical structure of functional dependencies was first axiomatized by W.W.Armstrong [1]. Different kinds of functional dependencies have also been investigated. The full family of strong dependencies has been introduced and axiomatized [5,7,8,9,14,15].

The family of Boolean dependencies is introduced [13]. In [2,3], the large subclass of positive Boolean dependencies, that is, Boolean combinations of attributes and the logical constant TRUE in which neither negation nor FALSE occur are studied. In [4], the class of equational dependencies is introduced. This class includes the class of functional dependencies as well as the Boolean dependencies, the positive Boolean dependencies and the classes of dependencies considered in [6,10].

In the papers mentioned above, the connection between dependencies and the fragment of propositional logic is built on the set of truth assignments T_R of a given relation R. Namely, for each pair of distinct tuples of R, the set T_R contains the truth assignment that maps an attribute A to TRUE if the two tuples are equal on A and to FALSE if the two tuples have different values for A.

In [11] a large class of mappings for constructing the truth assignments of relations was introduced. This class includes the equality mappings mentioned above. The class of Generalized Positive Boolean dependencies is introduced on these mappings.

In this paper we introduce a class of strong-Generalized Positive Boolean dependencies. We present a characterization of Armstrong relations for a given set of strong Generalized Positive Boolean dependencies.

*Research supported by Hungarian Foundation for Scientific Research Grant 2575.

†Computer and Automation Institute, Hungarian Academy of Sciences, H-1111 Budapest, Lágymányosi u. 11. Hungary.

The paper is structured as follows. In Section 2 we give some basic definitions. The concept of strong Generalized Positive Boolean dependencies is introduced in Section 3. In Section 4 we investigate connections between full families of strong Generalized Positive Boolean dependencies, s-semillatice and strong operations. Armstrong relation, the update problem and membership problem for strong Generalized Positive Boolean dependencies are studied in Section 5, Section 6 and Section 7.

2 Basic Definitions

We assume that the reader is familiar with the relational model of database systems and with the basic concepts of relational database theory [12,16]. In this paper we use the following notation.

Let $\mathcal{U} = \{A_1, \dots, A_n\}$ be a set of *attributes*. Corresponding to each attribute A_i is a set d_i, $1 \le i \le n$, called the *domain* of A_i. We assume that every d_i contains at least two elements.

A *relation* R over $\mathcal{U}$ is a subset of $d_1 \times \dots \times d_n$. Elements of R are called *tuples* and we usually denote them by u, v or t. The class of all relations over $\mathcal{U}$ is denoted by $\mathcal{R}$. For $k \ge 0$, $\mathcal{R}_k$ denotes those relations in $\mathcal{R}$ that have at most k tuples. If $R \in \mathcal{R}$, $t \in R$, $A \in \mathcal{U}$ and $X \subseteq \mathcal{U}$, then we denote by $t[A]$ the value of t for the attribute A, and by $t[X]$ the set $\{t[A] \mid A \in X\}$.

By $\mathcal{F}$ we denote the set of all *formulas* that can be constructed from $\mathcal{U}$ using the logical connectives $\wedge, \vee, \rightarrow, \neg$, and logical constants 1 (TRUE) and 0 (FALSE).

For $X = \{A_{i_1}, \dots, A_{i_k}\} \subseteq \mathcal{U}$, $\wedge X$ denotes the formula $A_{i_1} \wedge \dots \wedge A_{i_k}$, and $\vee X$ denotes the formula $A_{i_1} \vee \dots \vee A_{i_k}$.

Let $\mathcal{B} = \{0, 1\}$. A *valuation* is any function $x : \mathcal{U} \rightarrow \mathcal{B}$. The notation $x = (x_1, \dots, x_n) \in \mathcal{B}^n$ means that $x(A_i) = x_i$, $A_i \in \mathcal{U}$, $1 \le i \le n$.

If $f \in \mathcal{F}$ and $x \in \mathcal{B}^n$, then $f(x)$ denotes the truth value of f on the valuation x. For a finite subset Σ of $\mathcal{F}$ and for a valuation x in $\mathcal{B}^n$, we denote $\Sigma(x) = \wedge\{f(x) \mid f \in \Sigma\}$.

Let f be a formula in $\mathcal{F}$. We denote $T_f = \{x \in \mathcal{B}^n \mid f(x) = 1\}$. For a subset Σ of $\mathcal{F}$, we denote $T_\Sigma = \cap\{T_f \mid f \in \Sigma\}$. Then $x \in T_\Sigma$ if and only if $(\forall f \in \Sigma)\,(f(x) = 1))$.

Definition 2.1 *Let f and g be two formulas. f implies g, written $f \vdash g$, if $T_f \subseteq T_g$. f and g are equivalent, $f \equiv g$, if $T_f = T_g$. For $\Sigma, \Gamma \subseteq \mathcal{F}$, $\Sigma \vdash \Gamma$ if $T_\Sigma \subseteq T_\Gamma$, and $\Sigma \equiv \Gamma$ if $T_\Sigma = T_\Gamma$.*

Let $e = (1, \dots, 1)$ be the valuation that consists of all 1. A formula f in $\mathcal{F}$ is *positive* if $f(e) = 1$. Let $\mathcal{F}_p$ denote all positive formulas on $\mathcal{U}$. We know that $\mathcal{F}_p$ is equivalent to the set of all formulas that can be built using the connectives $\wedge, \vee, \rightarrow$ and constant 1 [10].

For each domain d_i, $1 \le i \le n$, we consider a mapping $\alpha_i : {d_i}^2 \rightarrow \mathcal{B}$. We assume that the mappings α_i satisfy the following properties.

(i) $(\forall a \in d_i)\,(\alpha_i(a, a) = 1)$,

(ii) $(\forall a, b \in d_i)\,(\alpha_i(a, b) = \alpha_i(b, a))$, and

(iii) $(\exists a, b \in d_i)\,(\alpha_i(a, b) = 0)$.

Example 2.2 *It is easy to see that the equality mappings on d_i,*

$$\alpha_i(a, b) = \begin{cases} 1 & if\ a = b \\ 0 & otherwise \end{cases}$$

$$a, b \in d_i,\ 1 \leq i \leq n$$

satisfy the properties (i) - (iii).

Example 2.3 *Let $\mathcal{U} = \{A, B, C\}$, where d_A is the set of positive integers, d_B is the set of real numbers and a null-value $\perp$, and d_C is the set of words w on a nonempty alphabet P, where the length of w is not greater than k, $k \geq 1$. We define the mappings α_A, α_B, and α_C as follows.*

$$\alpha_A(a, b) = \begin{cases} 1 & \text{if both } a \text{ and } b \text{ are simultaneously odd or even numbers} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_B(a, b) = \begin{cases} 1 & \text{if both } a \text{ and } b \text{ are simultaneously real or } \perp \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_C(a, b) = \begin{cases} 1 & \text{if both } a \text{ and } b \text{ have the same length} \\ 0 & \text{otherwise} \end{cases}$$

It is not hard to verify that the mappings α_A, α_B, and α_C satisfy the properties (i) - (iii).

Let $R \in \mathcal{R}$. For $u, v \in R$ we denote by $\alpha(u, v)$ the valuation

$$(\alpha_1(u[A_1], v[A_1]), \ldots, \alpha_n(u[A_n], v[A_n])).$$

Now for $R \in \mathcal{R}$ we denote $T_R = \{\alpha(u, v) \mid u, v \in R\}$. Note that for every u in R, $\alpha(u, u) = e$, so e is in T_R.

Definition 2.4 *Elements of $\mathcal{F}_p$ are called generalized positive Boolean dependencies (GPBD).*

Definition 2.5 *For $R \in \mathcal{R}$ and $f \in \mathcal{F}_p$, we say that R satisfies the GPBD f, written $R(f)$, if $T_R \subseteq T_f$.*

Definition 2.6 *Let $R \in \mathcal{R}$ and $\Sigma \subseteq \mathcal{F}_p$, we say that R satisfies the set of GPBDs Σ, written $R(\Sigma)$, if $R(f)$ for all $f \in \Sigma$. This is equivalent to $T_R \subseteq T_\Sigma$.*

For $\Sigma \subseteq \mathcal{F}_p$ and $f \in \mathcal{F}_p$, $\Sigma \models f$ means that, for all $R \in \mathcal{R}$, if $R(\Sigma)$ then $R(f)$. $\Sigma \models_2 f$ means that, for all $R \in \mathcal{R}_2$, if $R(\Sigma)$ then $R(f)$. In other words, $\Sigma \models f$ if and only if for all $R \in \mathcal{R}$, $T_R \subseteq T_\Sigma$ implies $T_R \subseteq T_f$.

For the equality mappings mentioned in Example 2.2 several classes of Boolean dependencies were investigated. Boolean dependencies were introduced in [13]. Positive Boolean dependencies are studied in [2,3]. Equational dependencies were introduced in [4]. Boolean dependencies of a special form are studied in [6,10]. These papers consider dependencies equivalent to the Boolean dependencies $\wedge X \rightarrow \wedge Y$ (functional dependency), $\wedge X \rightarrow \vee Y$ (weak dependency), $\vee X \rightarrow \wedge Y$ (strong dependency), and $\vee X \rightarrow \vee Y$ (dual dependency). In [3], the authors shown that the consequence relation for positive Boolean dependencies is the same as the consequence relation for propositional logic.

3 Strong-Generalized Positive Boolean Dependencies

Definition 3.1 *Let $R = \{t_1, \ldots, t_m\}$ be a relation over the finite set of attributes $\mathcal{U}$, and $X, Y \subseteq \mathcal{U}$. We say that GPBD $\vee X \rightarrow \wedge Y$ is strong-GPBD (for short s-GPBD) in R denoted $f_R^s(X,Y) = \vee X \xrightarrow[R]{s} \wedge Y$ or $X \xrightarrow[R]{s} Y$ or $X \xrightarrow{s} Y$ if*

$$(\forall t_i, t_j \in R)(\exists A \in X)(\alpha_A(t_i[A], t_j[A]) = 1) \longrightarrow$$
$$(\forall B \in Y)(\alpha_B(t_i[B], t_j[B]) = 1).$$

Let $S_R = \{X \xrightarrow[R]{s} Y\}$. S_R is called a full family of s-GPBDs of R.

Definition 3.2 *A s-GPBD over $\mathcal{U}$ is a statement of the form $X \xrightarrow{s} Y$, where $X, Y \subseteq \mathcal{U}$. The s-GPBD $X \xrightarrow{s} Y$ holds in a relation R if $X \xrightarrow[R]{s} Y$. We also say that R satisfies the $X \xrightarrow{s} Y$.*

We now introduce five inference s-axioms for s-GPBDs. Let $\mathcal{U}$ be a finite set of attributes, and denote by $P(\mathcal{U})$ its power set. Let $G \subseteq P(\mathcal{U}) \times P(\mathcal{U})$. We say that G is a full family of s-GPBDs over $\mathcal{U}$, if for all $X, Y, Z, W \subseteq \mathcal{U}$, and $A \in \mathcal{U}$

(S1.) $f^s(A, A) \in G$

(S2.) $f^s(X,Y) \in G, f^s(Y,Z) \in G, Y \neq \emptyset \longrightarrow f^s(X,Z) \in G$

(S3.) $f^s(X,Y) \in G, Z \subseteq X, W \subseteq Y \longrightarrow f^s(Z,W) \in G$

(S4.) $f^s(X,Y) \in G, f^s(Z,W) \in G \longrightarrow f^s(X \cup Z, Y \cap W) \in G$

(S5.) $f^s(X,Y) \in G, f^s(Z,W) \in G \longrightarrow f^s(X \cap Z, Y \cup W) \in G$

Let Σ_s be a set of s-GPBDs over $\mathcal{U}$. The closure of Σ_s, written Σ_s^+, is the smallest set containing Σ_s such that s-axioms cannot be applied to the set to yield an s-GPBD not in the set. Since Σ_s^+ must be finite, we can compute it by staring with Σ_s, applying S1, S2 and S5 and adding the derived s-GPBDs to Σ_s until no new s-GPBDs can be derived.

It can be seen [11] that there is a relation R over $\mathcal{U}$ such that $S_R = \Sigma_s^+$. Such a relation is called Armstrong relation for Σ_s.

Definition 3.3 *$X \xrightarrow{s} Y$ is a s-GPBD over $\mathcal{U}$ if X and Y are both subsets of $\mathcal{U}$. Σ_s is a set of s-GPBDs over $\mathcal{U}$ if every s-GPBD in Σ_s is s-GPBD over $\mathcal{U}$.*

Definition 3.4 *If Σ_s is a set of s-GPBDs over $\mathcal{U}$ and G is the set of all possible s-GPBDs over $\mathcal{U}$, then $\Sigma_s^- = G - \Sigma_s^+$. Σ_s^- is the exterior of Σ_s.*

If Σ_s is a set of s-GPBDs over $\mathcal{U}$ and X is a subset of $\mathcal{U}$, then there is s-GPBD $X \xrightarrow{s} Y$ in Σ_s^+ such that Y is maximal: for any other s-GPBD $X \xrightarrow{s} Z$ in Σ_s^+, $Y \supseteq Z$. This result follows from S5. Y is called the closure of X, and is denoted by X^+.

Definition 3.5 *Let Σ_s be a set of s-GPBDs over $\mathcal{U}$. $X \subseteq \mathcal{U}$, $A \in \mathcal{U}$. Then $\{A\}^+ = \{B \in \mathcal{U} \mid \{A\} \xrightarrow{s} \{B\} \in \Sigma_s^+\}$, $X^+ = \{B \in \mathcal{U} \mid X \xrightarrow{s} \{B\} \in \Sigma_s^+\}$.*

$\{A\}^+$ is called the closure of $\{A\}$.

Theorem 3.6 *Inference axioms S1 to S5 are complete.*

Proof: Given a set Σ_s of s-GPBDs over $\mathcal{U}$, for any s-GPBD $X \xrightarrow{s} Y$ in Σ_s^-. We shall axhibit a relation R that satisfies Σ_s^+ but not $X \xrightarrow{s} Y$. Hence, we can see that there are no s-GPBDs implied by Σ_s that are not derived by Σ_s. Relation R will satisfy most of the s-GPBDs in Σ_s^+, for a s-GPBD $(W \xrightarrow{s} Z)$ in Σ_s^+.

Let $\mathcal{U} = \{A_1, A_2, \ldots, A_n\}$ and let a_i, b_i, c_i be distinct elements of $dom(A_i)$, $1 \leq i \leq n$. There will be only two tuples in R, t_1 and t_2. Tuple t_1 will be $< a_1 a_2 \ldots a_n >$. Tuple t_2 is defined as

$$\forall A_i \in X^+, \alpha_{A_i}(t_1[A_i], t_2[A_i]) = 1$$

and

$$\forall A_i \notin X^+, \alpha_{A_i}(t_1[A_i], t_2[A_i]) = 0.$$

First we show that R does not satisfy $X \xrightarrow{s} Y$. From the definition of R, $\exists B \in X$ that $\alpha_B(t_1[B], t_2[B]) = 1$. Suppose $\alpha_C(t_1[C], t_2[C]) = 1$ for all $C \in Y$, and hence $Y \subseteq X^+$.

But since $(X \xrightarrow{s} X^+) \in \Sigma_s^+$, by S3, we obtain that $X \xrightarrow{s} Y$ is in Σ_s^+, a contradiction to $X \xrightarrow{s} Y$ is in Σ_s^-.

Now we show that R satisfies all the s-GPBD in Σ_s^+. Let $\{B\} \in X^+$, hence by Definition 3.5. we obtain that $\{B\}^+ = X^+$. By the definition of s-GPBDs, we have $(W \xrightarrow{s} X^+) \in \Sigma_s^+$. Since $(W \xrightarrow{s} Z) \in \Sigma_s^+$, and by S5, we obtain $(W \xrightarrow{s} (X^+ \cup Z)) \in \Sigma_s^+$, so $(X^+ \cup Z) \in W^+$. Hence $Z \subseteq X^+$, and $\alpha_C(t_1[C], t_2[C]) = 1$ for all $C \in Z$. □

4 Strong-Generalized Positive Boolean Dependencies and s-semilattice

Definition 4.1 *Let $I \subseteq P(\mathcal{U})$. We say that I is a $\cap$-semilattice over $\mathcal{U}$ if $\mathcal{U} \in I$, and $X, Y \in I \longrightarrow X \cap Y \in I$. Let $M \subseteq P(\mathcal{U})$. Denote by M^+ the set $\{\cap M' \mid M' \subseteq M\}$. Then we say M generates I if $M^+ = I$.*

Theorem 4.2 [4] *Let $I \subseteq P(\mathcal{U})$ be a $\cap$-semilattice over $\mathcal{U}$. Let $N = \{X \in I : \forall Z, W \in I : X = Z \cap W \longrightarrow X = Z$ or $X = W\}$. Then N generates I and if N' generates I, then $N \subseteq N'$. N is called the minimal generator of I (It is obvious that $\mathcal{U} \in N$).*

Definition 4.3 [15] *Let $I \subseteq P(\mathcal{U})$. We say that I is an s-semilattice over $\mathcal{U}$ if I satisfies*

(1.) I is a $\cap$-semilattice,

(2.) for all $X \subseteq N \setminus \mathcal{U}$

$$((\exists A \in X)(\forall Z \in N \setminus \mathcal{U})(X \not\subset Z) \longrightarrow (A \notin Z),$$

where N is the minimal generator of I.

Definition 4.4 [15] *The mapping $F : P(\mathcal{U}) \to P(\mathcal{U})$ is called a strong operation over $\mathcal{U}$ if for every $A, B \in \mathcal{U}$ and $X \in P(\mathcal{U})$, the following properties hold:*

(1.) $F(\emptyset) = \mathcal{U}$,

(2.) $A \in F(\{A\})$,

(3.) $B \in F(\{A\}) \longrightarrow F(\{B\}) \subseteq F(\{A\})$,

(4.) $F(X) = \bigcap\limits_{A \in X} F(\{A\})$.

Theorem 4.5 [15] *Let F be a strong operation over $\mathcal{U}$. Let $I_F = \{F(X) \mid X \in P(\mathcal{U})\}$. Then I_F is an s-semilattice over $\mathcal{U}$. Conversely, if I is an s-semilattice over $\mathcal{U}$, then there is exactly one strong operation F such that $I_F = I$, where $F(\emptyset) = \mathcal{U}$, and for all $A \in \mathcal{U}$*

$$F(\{A\}) = \begin{cases} \bigcap\limits_{\substack{A \in W \\ W \in N \setminus \mathcal{U}}} W & \text{if } \exists W : A \in W \ (N \text{ the minimal generator of } I), \\ U & \text{otherwise.} \end{cases}$$

Theorem 4.6 *Let $G \subseteq P(\mathcal{U}) \times P(\mathcal{U})$. G is a full family of s-GPBDs over $\mathcal{U}$. Let $(X, Y) \in P(\mathcal{U}) \times P(\mathcal{U}) \setminus G$. There is an $A \in X$, and an $E_A \subseteq \mathcal{U}$ such that*

(i.) $A \in E_A$,

(ii.) $(\{A\} \xrightarrow{s} E_A) \in G$,

(iii.) $E' \supset E_A$ *implies that* $(\{A\} \xrightarrow{s} E') \notin G$.

Proof: If for any $A \in X$ we have $(\{A\}, Y) \in G$. By S5 we have $(X, Y) \in G$. Hence there is an $A \in X$ such that $(\{A\} \xrightarrow{s} Y) \notin G$. If for every $B \in Y$, $(\{A\} \xrightarrow{s} \{B\}) \in G$ holds, then by S4 $(\{A\} \xrightarrow{s} Y) \in G$.

Thus there is a $B \in Y$ such that $(\{A\} \xrightarrow{s} \{B\}) \notin G$. By S1 and S3 there is an $E_A \subseteq \mathcal{U}$ such that $A \in E_A$, $(\{A\} \xrightarrow{s} E_A) \in G$ and E_A is maximal to this property. □

Theorem 4.7 *Let $G \subseteq P(\mathcal{U}) \times P(\mathcal{U})$. G is a full family of s-GPBDs over $\mathcal{U}$ if and only if there is a family $\{E_i : i = 1, \ldots, l;\ \bigcup\limits_{i=1}^{l} E_i = \mathcal{U}\}$ of subsets of $\mathcal{U}$ such that*

(i.) for all $X \subseteq \mathcal{U}$, $(\emptyset \xrightarrow{s} X) \in G$,

(ii.) for any $X, Y \subseteq \bigcap\limits_{E_i \cap X \neq \emptyset} \longrightarrow (X \xrightarrow{s} Y) \in G$,

(iii.) $(Z \xrightarrow{s} W) \in G$, $Z \cap E_i \neq \emptyset \longrightarrow W \subseteq E_i$.

Proof: Only if: Assume that G is a full family of s-GPBDs over $\mathcal{U}$. Then by Theorem 4.6, S1, S3, and S5 for each $A \in \mathcal{U}$ we can construct an $E_i (E_i \subseteq \mathcal{U})$ such that $(\{A\} \xrightarrow{s} E_i) \in G$, and $\forall E' \mid E_i \subset E'$ implies $(\{A\} \xrightarrow{s} E') \notin G$. By Theorem 4.6, it is obvious that $A \in E_i$ and we have n such E_i-s, where $n = \mid \mathcal{U} \mid$. Thus, we have the set $E = \{E_i : i = 1, \ldots, n; \bigcup_{i=1}^{n} E_i = \mathcal{U}\}$. Assume $X = \{A_1 A_2 \ldots A_k : A_j \in \mathcal{U}, j = 1, \ldots, k\} \neq \emptyset$ and Y_1 is a set such that $(X \xrightarrow{s} Y_1) \in G$, $\forall Y_2 : Y_1 \subset Y_2$ implies $(X \xrightarrow{s} Y_2) \notin G$. By the construction of E, we have that for each A_j there is an $E_j \in E$ such that $(\{A\} \xrightarrow{s} E_j) \in G$. By S4 we have $(\bigcap_{j=1}^{k} A_j \xrightarrow{s} \bigcap_{j=1}^{k} E_j) = (X \xrightarrow{s} \bigcap_{j=1}^{k} E_j) \in G$. By Theorem 4.6 and the definition of Y_1 we have $\bigcap_{j=1}^{k} E_j \subseteq Y_1$. By $(X \xrightarrow{s} Y_1) \in G$ and by S3, we have $(\{A_j\} \xrightarrow{s} Y_1) \in G$ for all $j (j = 1, \ldots, k)$. Thus, $Y_1 \subseteq \bigcap_{j=1}^{k} E_j$ holds. Hence, $Y_1 = \bigcap_{j=1}^{k} E_j$. It is obvious that

$$\bigcap_{E_i \cap X \neq \emptyset} E_i \subseteq \bigcap_{j=1}^{k} E_j.$$

Thus, for all

$$Y (Y \subseteq \bigcap_{E_i \cap X \neq \emptyset} E_i) : Y \subseteq Y_1.$$

Hence $(X \xrightarrow{s} Y) \in G$ holds.

If $(Z \xrightarrow{s} W) \in G$, $Z \cap E_i \neq \emptyset$. Let $A_1 \in Z \cap E_i$. Suppose that $W \cap (\mathcal{U} \setminus E_i) \neq \emptyset$. Let $D_1 \in W \cap (\mathcal{U} \setminus E_i)$.

By S3 we have $(\{A_1\} \xrightarrow{s} \{D_1\}) \in G$, and by S1 we have $(\{A_1\} \xrightarrow{s} \{A_1\}) \in G$. Let $A \in E_i$, then $(\{A\} \xrightarrow{s} E_i) \in G$ implies that $(\{A, A_1\} \xrightarrow{s} \{A_1\}) \in G$ by S5. Hence by S3 we have $(\{A\} \xrightarrow{s} \{A_1\}) \in G$. Since $(\{A\} \xrightarrow{s} \{A_1\}) \in G$, $(\{A_1\} \xrightarrow{s} \{D_1\}) \in G$ and by S2 we have $(\{A\} \xrightarrow{s} \{D_1\}) \in G$. Thus, by S4 we have $(\{A\} \xrightarrow{s} E_i \cup \{D_1\}) \in G$.

On the other hand, by Theorem 4.6 we have $(\{A\} \xrightarrow{s} E_i) \in G$ and $\forall E' : E_i \subset E'$ implies $(\{A\} \xrightarrow{s} E') \in G$. Hence $W \subseteq E_i$.

If : Assume that there is a family $\{E_i : i = 1, \ldots, l : \bigcup_{i=1}^{l} E_i = \mathcal{U}\}$ such that satisfies $(i), (ii)$ and (iii).

By Theorem 4.6 we can construct an $E_i (E_i \subseteq \mathcal{U})$ so that $\forall A \in \mathcal{U}$,

$$(\{A\} \xrightarrow{s} E_i) \in G,$$

and $\forall E' : E_i \subset E'$ implies $(\{A\} \xrightarrow{s} E') \notin G$.

It is obvious that $A \in E_i$, and easy to see that $l = n$, where $n = \mid \mathcal{U} \mid$.

Then, from (ii), easy to see that $\forall A \in \mathcal{U}$, we have $(\{A\} \xrightarrow{s} \{A\}) \in G$. Assume S5 does not hold, that is if $(X \xrightarrow{s} Y) \in G$ and $(Z \xrightarrow{s} W) \in G$ then

$$((X \cap Z) \xrightarrow{s} \cup W)) \in G. \tag{4.7.1}$$

Suppose $X \cap Z = \emptyset$ and $Y \cup W = \mathcal{U}$. From (4.7.1), we have $(\emptyset \xrightarrow{s} \mathcal{U}) \notin G$. This contradiction to (i), so S5 holds.

Assume S4 does not hold, that is if $(X \xrightarrow{s} Y) \in G$ and $(Z \xrightarrow{s} W) \in G$, then

$$((X \cap Z) \xrightarrow{s} (Y \cup W)) \notin G. \tag{4.7.2}$$

Suppose $X \cup Z = Z'$,

$$Y \cap W = W' \subseteq \bigcap_{E_i \cap X \neq \emptyset} E_i.$$

From (4.7.2), we have $(Z' \xrightarrow{s} W') \notin G$. this contradiction to (ii), so S4 holds.
From $(ii), (iii)$ it is easy to see that S2, S3 hold too. □

Theorem 4.8 *Let G be a full family of s-GPBDs over $\mathcal{U}$. We define the mapping $F_G : P(\mathcal{U}) \times P(\mathcal{U})$ as follow:*

$$F_G(X) = \{A \in \mathcal{U} \mid (X \xrightarrow{s} \{A\}) \in G\}.$$

Then F_G is a strong operation over $\mathcal{U}$. Conversely, if F is an arbitrary strong operation over $\mathcal{U}$, then there is exactly one full family of s-GPBDs G such that $F_G = F$, where

$$G = \{(X \xrightarrow{s} Y) \mid X, Y \in P(\mathcal{U}) : Y \subseteq F(X)\}.$$

Proof: 1. Assume G is a full family of s-GPBDs over $\mathcal{U}$. We show that F_G is a strong operation. Since $F_G(X) = \{A \in \mathcal{U} \mid (X \xrightarrow{s} \{A\}) \in G\}$, so

$$F_G(\{A\}) = \{B \in \mathcal{U} \mid (\{A\} \xrightarrow{s} \{B\}) \in G\}. \tag{4.8.1}$$

By S1, we have that $\forall A \in \mathcal{U}, A \in F_G(\{A\})$. By (i) in Theorem 4.7,

$$\forall C \subseteq \mathcal{U}, (\emptyset \xrightarrow{s} C) \in G.$$

So we have $F_G(\emptyset) = \mathcal{U}$. By Theorem 4.6, and by (4.8.1), we obtain that for $A \in \mathcal{U}$, $F_G(\{A\}) = E_A$. So, by (ii) in Theorem 4.6, we have for $B \in \mathcal{U}$, $(\{B\} \xrightarrow{s} F_G(\{B\})) \in G$. Thus, assume $B \in F_G(\{A\})$, and by (iii) in Theorem 4.7, we have $F_G(\{B\}) \subseteq F_G(\{A\})$.

On the other hand, from (4.8.1) and Theorem 4.6, we have for $A \in \mathcal{U}$, $(\{A\} \xrightarrow{s} F_G(\{A\})) \in G$.

Let $A \in X \subseteq \mathcal{U}$, then by S5 we obtain

$$(X \xrightarrow{s} \bigcap_{A \in X} F(\{A\})) \in G.$$

That is

$$\bigcap_{A \in X} F(\{A\}) \subseteq F_G(X).$$

By the definition of $F_G(X)$, we have $(X \xrightarrow{s} F_G(X)) \in G$. Since for $\forall A \in X, X \cap F_G(\{A\}) \neq \emptyset$, by Theorem 4.7, we obtain $F_G(X) \subseteq F_G(\{A\})$. So

$$F_G(X) \subseteq \bigcap_{A \in X} F(\{A\}).$$

Hence

$$F(X) = \bigcap_{A \in X} F(\{A\}).$$

2. Assume that F is a strong operation over $\mathcal{U}$, and $G = \{(X \xrightarrow{s} Y) \mid Y \subseteq F(X)\}$. We have to show that G is a full family of s-GPBDs. That is, we show that it satisfies $(i), (ii)$ and (iii) in Theorem 4.7.

By Theorem 4.6 and Theorem 4.7, we set

$$E = \{F(\{A\}) : A \in \mathcal{U}, n = \mid \mathcal{U} \mid\}.$$

Assume

$$\bigcap_{F(\{A\}) \cap X \neq \emptyset} F(\{A\}) \subseteq F(X).$$

Since $G = \{(X \xrightarrow{s} Y) \mid Y \subseteq F(X)\}$. So if

$$Y \subseteq \bigcap_{F(\{A\}) \cap X \neq \emptyset} F(\{A\}),$$

then it satisfies (ii) in Theorem 4.7

Assume $(V, W) \in G$, and $V \cap F(\{A\}) \neq \emptyset$. Let $B \in V \cap F(\{A\})$, so $B \in V$ and $B \in F(\{A\})$. Thus, by (iii) in the definition of strong operation $B \in F(\{A\})$ implies $F(\{B\}) \subseteq F(\{A\})$. By the definition of G, we have $W \subseteq F(V)$. By (iii) in the definition of strong operation, we have

$$F(V) = \bigcap_{D \in V} F(\{D\}).$$

Since $B \in V$, so

$$\bigcap_{D \in V} F(\{D\}) \subseteq F(\{B\}).$$

Hence $D \subseteq F(\{A\})$, i.e. it satisfies (iii) in Theorem 4.7. It is clear that $\forall A \in \mathcal{U}$, $(\emptyset \xrightarrow{s} \{A\}) \in G$. □

5 Armstrong relation for s-GPBDs

Definition 5.1 *Let Σ_s be a set of s-GPBDs on $\mathcal{U}$, and let R be a relation on $\mathcal{U}$. R exactly represents Σ_s if $S_R = \Sigma_s^+$. If R exactly represents Σ_s then we also say that R is an Armstrong relation for Σ_s.*

Definition 5.2 *Let $R = \{t_1, \ldots, t_m\}$ be a relation over $\mathcal{U}$. We set $E_{ij} = \{A \in \mathcal{U} \mid \alpha_A(t_i[A], t_j[A]) = 1\}$, and $E_R = \{E_{ij}, 1 \leq i, j \leq m\}$. We denote $E(A) = \bigcap_{A \in E_{ij}} E_{ij}$ if there is a such E_{ij}, in the converse case set $E(A) = \mathcal{U}$, where $A \in \mathcal{U}$. Denote $E_R^* = \{E(A) \mid A \in \mathcal{U}\}$. E_R^* is called the α-attribute-equality set of R.*

A strong relation scheme is a pair $(\mathcal{U}, \Sigma_s)$, where $\mathcal{U}$ is a set of attributes and Σ_s is a set of s-GPBGs on $\mathcal{U}$.

Definition 5.3 *Let $H =< \mathcal{U}, \Sigma_s >$ be a strong relation scheme, $X \subseteq \mathcal{U}$. We set $X^+ = \{A \in \mathcal{U} \mid (X \xrightarrow{s} \{A\}) \in \Sigma_s^+\}$. X^+ is called the closure of X. Denote $I(H) = \{X^+ \mid X \in P(\mathcal{U})\}$. It can be seen that $I(H)$ (for short $I(\Sigma_s)$) is a s-semilattice over $\mathcal{U}$. Denote by $N(H)$ (for short $N(\Sigma_s)$) the minimal generator of $I(H)$.*

It is easy to see that $N(H)$ satisfies (2) in Definition 4.3 and $X^+ \cap Y^+ = (X \cup Y)^+, X^+ = \bigcap_{A \in X} \{A\}^+$.

Theorem 5.4 *Let G be a full family of s-GPBDs, and $R = \{t_1, \ldots, t_m\}$ be a relation over $\mathcal{U}$. Then R represents G iff for each $A \in \mathcal{U}$*

$$F_G(\{A\}) = \begin{cases} \bigcap_{A \in E_{ij}} E_{ij} & \text{if } \exists E_{ij} : A \in E_{ij}, \\ \mathcal{U} & \text{otherwise.} \end{cases}$$

Where $F_G(X) = \{A \in \mathcal{U} \mid (X \xrightarrow{s} \{A\}) \in G\}$, and E_{ij} is the equality set of R.

Proof: Only if: By Theorem 4.8 $S_R = G$ if and only if $F_{S_R} = F$, where F is strong operation over $\mathcal{U}$. We have show that $F_{S_R}(\{a\}) = F_G(\{A\})$ for all $A \in \mathcal{U}$. Clearly,

$$F_{S_R}(\{A\}) = \{B \in \mathcal{U} : (\{A\} \xrightarrow{s} \{B\})\}. \tag{5.4.1}$$

According to the definition of s-GPBDs we know that for any $A \in \mathcal{U}$, and $A \neq \emptyset$ $(\{A\} \xrightarrow{s} Y)$ iff

$$(\forall t_1, t_2 \in R)\alpha_A(t_1[A], t_2[A]) = 1 \longrightarrow (\forall B \in Y)\alpha_B(t_1[B], t_2[B]) = 1.$$

Let $T = \{E_{ij} \mid A \in E_{ij}\}$. It is easy to see that if $T = \emptyset$, then $F_{S_R}(\{A\}) = \mathcal{U}$ holds. If $T \neq \emptyset$. Let

$$X = \bigcap_{A \in E_{ij}} E_{ij}.$$

If $T = E$ (E is the set of all α-attribute equality sets of R), then $(\{A\} \xrightarrow{s} X)$. If $T \subset E$, then for all $E_{ij} \in T$, we have $\alpha_A(t_1[A], t_2[A]) \neq 1$. By (5.4.1), we obtain

$$F_{S_R}(\{A\}) = \bigcap_{A \in E_{ij}} E_{ij}.$$

If: If F_G holds to (5.4.1), then we have $F_G(\{A\}) = F_{S_R}(\{A\})$. By Theorem 4.8, we obtain $F_G = F_{S_R}$.

□

Definition 5.5 *Let R be a relation, an F a strong operation over $\mathcal{U}$. We say that the relation R exectly represents F iff $F_{S_R} = F$.*

Lemma 5.6 [15] *Let F be a strong operation and R a relations over $\mathcal{U}$. Then R represents F iff for all $A \in \mathcal{U}$,*

$$F(\{A\}) = \begin{cases} \bigcap_{A \in E_{ij}} E_{ij} & \text{if } \exists E_{ij} : A \in E_{ij}, \\ U & \text{otherwise.} \end{cases}$$

Theorem 5.7 *Let Σ_s be a set of s-GPBDs on $\mathcal{U}$, and let R be a nonempty relation on $\mathcal{U}$. Then R is an Armstrong relation for Σ_s if and only if*

$$N(\Sigma_s) \subseteq E_R^* \subseteq I(\Sigma_s).$$

Proof: Only if: If R is an Armstrong relation for Σ_s, then by Definition 5.1 $S_R = \Sigma_s^+$. We set $F_{\Sigma_s^+} = X^+$ for all $X \in P(\mathcal{U})$ and

$$F_{S_R}(X) = \{A \in \mathcal{U} \mid (X \xrightarrow{s} \{A\})\}.$$

By Theorem 4.8, $S_R = \Sigma_s^+$ if and only if $F_{S_R} = F$, where F is a strong operation over $\mathcal{U}$. It follows that $F_{\Sigma_s^+} = F_{S_R}$.

By Theorem 4.5 and Definition 5.3, $I(\Sigma_s) = I_{F_{S_R}}$ and $N(\Sigma_s) = N$, where N is the minimal generator of $I_{F_{S_R}}$. In other hand, since

$$F_{S_R}(X) = \bigcap_{A \in X} F_{S_R}(\{A\})$$

for all $X \in P(\mathcal{U})$, so we have to show that $F_{S_R}(\{A\}) = E(A)$ for each $A \in \mathcal{U}$.

Clearly, $F_{S_R}(\{A\}) = \{B \in \mathcal{U} \mid (\{A\} \xrightarrow{s} \{B\})\}$. By the definition of s-GPBD, we know that for any $A \in \mathcal{U}$, $A \neq \emptyset$, $(\{A\} \xrightarrow{s} Y)$ iff

$$(\forall t_i, t_j \in R)(\alpha_A(t_i[A], t_j[A]) = 1) \longrightarrow ((\forall B \in Y)(\alpha_B(t_i[B], t_j[B]) = 1)).$$

Assume $Q = \{E_{ij} \mid A \in E_{ij}\}$. It is obvious that if $Q = \emptyset$ then $F_R(\{A\}) = \mathcal{U}$. If $Q = \emptyset$, then assume that

$$X = \bigcap_{A \in E_{ij}} E_{ij},$$

then it is obvious that $(\{A\} \xrightarrow{s} X)$ and for all $E_{ij} : E_{ij} \notin Q$,

$$(\alpha_A(t_i[A], t_j[A]) \neq 1).$$

Hence,

$$F_{S_R}(\{A\}) = \bigcap_{A \in E_{ij}} E_{ij} = E(A)$$

for all $A \in \mathcal{U}$. Therefore, by Definition 5.3, $E_R^* \subseteq I_{F_R}$.

Now we show that $N(\Sigma_s) \subseteq E_R^*$. By Definition 4.3, Theorem 4.2, and Theorem 4.5, clearly to see that $N(\Sigma_s) \subseteq E_R^*$.

If: Assume that $N(\Sigma_s) \subseteq E_R^* \subseteq I(\Sigma_s)$. Since $E_R^* \subseteq I(\Sigma_s)$, and $I(\Sigma_s) = \{X^+ : X \in P(\mathcal{U})\}$,

$$X^+ = \{A \in \mathcal{U} \mid (X \xrightarrow{s} \{A\}) \in \Sigma_s^+.$$

Thus we obtain $E_R^* = \{F_{\Sigma_s^+}(\{A\}) : A \in \mathcal{U}\}$. By above proof for each $A \in \mathcal{U}$, we have that $E(A) = F_{S_R}(\{A\})$. Hence,

$$\{F_{\Sigma_s^+}(\{A\}) : A \in \mathcal{U}\} = \{F_{S_R}(\{A\}) : A \in \mathcal{U}\}.$$

Suppose $A \in \mathcal{U}$ that $F_{\Sigma_s^+}(\{A\}) \neq F_{S_R}(\{A\})$. By Definition 4.4 and Theorem 4.5 we assume that $F_{\Sigma_s^+} = Y$, where $Y \in N(\Sigma_s)$. Since $N(\Sigma_s) \subseteq E_R^*$, so $F_{\Sigma_s^+} \in E_R^*$. Clearly to see that $F_{\Sigma_s^+}(\{A\}) = E(A)$. This is a contradiction. Therefore, we obtain that $F_{\Sigma_s^+}(\{A\}) = F_{S_R}(\{A\})$ for each $a \in \mathcal{U}$. Thus, $F_{\Sigma_s^+} = F_{S_R}$, and by Theorem 4.8, $S_R = \Sigma^+$.

□

Algorithm 5.8 *(Finding Σ_s)*

(Input :) *Given relation $R = \{t_1, \ldots, t_m\}$ over $\mathcal{U}$.*

(Output :) *Construct Σ_s, such that $S_R = \Sigma_s^+$.*

(Step 1 :) *From R we compute E_R.*

(Step 2 :) *From E_R we construct $E_R^* = \{E(A) : A \in \mathcal{U}\}$.*

(Step 3 :) *Set $\Sigma_s = (\{A\} \xrightarrow{s} E(A)) \mid A \in \mathcal{U}\}$*

Clearly, the time complexity of this algorithm is polynomial in the size of R.

Algorithm 5.9 *(Finding $\{A\}$)*

(Input :) *Given $\Sigma_s = \{(A_i \xrightarrow{s} B_i) \mid i = 1, \ldots, m\}$* and $A \in \mathcal{U}$.

(Output :) *Compute $\{A\}^+$*

(Step 1 :) *$A \in \mathcal{U}$, let $L_0 = \{A\}$*

(Step i+1 :) *If there is an $(A_i \xrightarrow{s} B_i) \in \Sigma_s$*

so that $A_j \cap X^{(i)} \neq \emptyset$ and $B \not\subseteq X^{(i)}$ then

$$X^{(i+1)} = X^i \cup \Big(\bigcup_{A_j \cap X^{(i)}} B_j \Big).$$

In the converse case we set $\{A\}^+ = X^{(t)}$.

It can be seen that the time complexity of this algorithm is polynomial in the sizes of Σ_s and $\mathcal{U}$.

6 Update Problem

In [11], the update problem is introduced for a set of GPBDs Σ. Let R be a relation that satisfies a set of GPBDs Σ and t be a tuple $d_1 \times \ldots \times d_n$. We say that t can be added to R if $R \cup \{t\}$ satisfies Σ.

Theorem 6.1 [11] *Let R be a relation satisfying a set of GPBDs Σ, and let t be a tuple in $d_1 \times \ldots \times d_n$. Then t can be added to R if and only if $(\forall u \in R)(\alpha(t,u) \in T_\Sigma)$.*

Let Σ_s be a set of s-GPBDs, $\Sigma_s = \{X_i \xrightarrow{s} Y_i\}$, where $X_i, Y_i \subseteq \mathcal{U}$. Let $M = \cup X_i, N = \cup Y_i$. By Theorem 6.1 and definition of s-GPBDs, we get the following result.

Theorem 6.2 *Let R be a relation satisfying a set of s-GPBDs Σ_s, $\Sigma_s = \{X_i \xrightarrow{s} Y_i\}$, and let t be a tuple in $d_1 \times \ldots \times d_n$. Then t can be added to R if and only if $(\forall u \in R)(\forall A \in N)(\alpha_A(t[A], u[A]) = 1)$.*

It is easy to see that, if $(\forall u \in R)(\forall A \in M)(\alpha_A(t[A], u[A]) = 0)$. Then t is added to R too.

7 Membership Problem for s-GPBDs

In [11], the membership problem for GPBDs is introduced. Given a set of GPBDs Σ and a GPBD f, decide whether $\Sigma \models f$.

From Algorithms 5.8, 5.9 and $X^+ = \cup\{A\}^+ \; A \in X$. We have the following.

Proposition 7.1 *Let Σ_s be a set of s-GPBDs on $\mathcal{U}$ and $X, Y \subseteq \mathcal{U}$. Then, there is an algorithm deciding whether that $X \xrightarrow{s} Y \in \Sigma_s^+$.*

The time complexity of this algorithm is polynomial in the sizes of Σ_s and $\mathcal{U}$.

Theorem 7.2 [11] *Let Σ be a set of GPBDs on $\mathcal{U}$, and $X, Y, Z \subseteq \mathcal{U}$. Then*

1. $\Sigma \models \wedge X \to \wedge Y \Leftrightarrow$
   $(\forall x \in T_\Sigma)(((\exists A \in X)\,(x(A) = 0)) \vee ((\forall B \in Y)\,(x(B) = 1)))$.
2. $\Sigma \models \wedge X \to \vee Y \Leftrightarrow$
   $(\forall x \in T_\Sigma)(((\exists A \in X)\,(x(A) = 0)) \vee ((\exists B \in Y)\,(x(B) = 1)))$.
3. $\Sigma \models \vee X \to \wedge Y \Leftrightarrow$
   $(\forall x \in T_\Sigma)(((\forall A \in X)\,(x(A) = 0)) \vee ((\forall B \in Y)\,(x(B) = 1)))$.
4. $\Sigma \models \vee X \to \vee Y \Leftrightarrow$
   $(\forall x \in T_\Sigma)(((\forall A \in X)\,(x(A) = 0)) \vee ((\exists B \in Y)\,(x(B) = 1)))$.
5. $\Sigma \models \wedge X \to (\wedge Y \vee \wedge Z) \Leftrightarrow (\forall x \in T_\Sigma)(((\exists A \in X)\,(x(A) = 0)) \vee$
   $(((\forall B \in Y)\,(x(B) = 1)) \vee ((\forall C \in Z)\,(x(C) = 1))))$.

Theorem 7.3 *Let Σ_s be a set of s-GPBDs on U, and $X, Y \subseteq U$. Then*

$$
\begin{tikzcd}
 & \Sigma_s \models \vee X \to \wedge Y \arrow[dl, dash] \arrow[dr, dash] & \\
\Sigma_s \models \vee X \to \vee Y \arrow[dr, dash] & & \Sigma_s \models \wedge X \to \wedge Y \arrow[dl, dash] \\
 & \Sigma_s \models \wedge X \to \vee Y &
\end{tikzcd}
$$

Proof:
By Theorem 7.2 and definition of s-GPBDs. It is easy to see that Theorem 7.3 holds. □

References

[1] Armstrong W.W., Dependency structures of database relationships. *Information Processing 74, Holland Publ.Co.* (1974), 580-583.

[2] Berman J., Blok W.J., Generalized Boolean Dependencies. *Abstracts of AMS*, **6** (1985), 163.

[3] Berman J., Blok W.J., Positive Boolean Dependencies. *Inf. Processing Letters*, **27** (1988), 147-150.

[4] Berman J., Blok W.J., Equational Dependencies. *Manuscript*, (1990).

[5] Czédli G., Függőségek relációs adatbázis modellben. *Alkalmazott Matematikai Lapok*, **6** (1980), 131-143.

[6] Czédli G., On dependencies in the relational model of data. *J.EIK*, **17** (1981), 103-112.

[7] Demetrovics J., Relációs adatmodell logikai és structurális vizsgálata. *MTA-SZTAKI Tanulmányok, Budapest*, **114** (1980), 1-97.

[8] Demetrovics J.,Gyepesi Gy., On the functional dependencies and some generalizations of it. *Acta Cybernetica*, **5** (1981), 295-305.

[9] Demetrovics J.,Gyepesi Gy., Logical dependencies in relational database. *MTA-SZTAKI Tanulmányok, Budapest*, **133**, (1982), 59-78.

[10] Demetrovics J.,Gyepesi Gy., Some generalized type functional dependencies formalized as equality set on matrics. *Discrete Applied Mathematics*, **6** (1983), 35-47.

[11] Huy N.X., Thanh L.T., Generalized Positive Boolean Dependencies. *J.EIK*, **28** (1992), 363-370.

[12] Maier D. *The Theorem of Relational Databases.* Computer Science Press , (1983).

[13] Sagiv Y., Delobel C., Parker D.S., and Fagin R. An Equivalence Between Relational Database Dependencies and a Fragment of Propositional Logic. *J.ACM*, **28** (1981), 435-453.

[14] Thi V.D. Logical dependencies and irredundant relations. *Computers and Artificial Intelligence*, **7** (1988), 165-184.

[15] Thi V.D. Strong dependencies and s-semilatties. *Acta Cybernetica*, **8** (1987), 195-202.

[16] Ullman J.D. *Principles of Database Systems.* (Second Edition.) Computer Science Press, Potomac, Md., 1982.

Received July 30, 1993

Acta Cybernetica, Vol. 11, No. 3, Szeged, 1994

Partitioning Graphs into Two Trees*

Ulrich Pferschy † Gerhard J. Woeginger‡ En-Yu Yao§

Abstract

We investigate the problem of partitioning the edges of a graph into two trees of equal size. We prove that this problem is NP-complete in general, but can be solved in polynomial time on series-parallel graphs.

1 Introduction

In this note, we will examine the partitioning problem PG2T defined as follows.

PARTITIONING GRAPHS INTO TWO TREES (PG2T)

Input. A graph $G = (V, E)$.

Question. Does there exist a partition of $E = E_1 \cup E_2$ with $|E_1| = |E_2|$, $V_1, V_2 \subseteq V$, such that the two edge-induced subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ of G both are trees ?

If the trees G_1 and G_2 are required to be *spanning* trees of G, the problem can be solved in polynomial time by matroid partitioning techniques, see Lawler [4]. In contrast to this polynomial time result, we will show that detecting a partitioning into two arbitrary (not necessarily spanning) equal-sized trees is NP-complete. Our reduction is done from the Hamiltonian Path problem in cubic graphs (Garey and Johnson [2]). To simplify the presentation, we will introduce an intermediate problem TCT (defined below) and prove that it is also NP-complete.

On the positive side, we will show that PG2T is polynomial time solvable for the class of series-parallel graphs.

The paper is organized as follows: Section 2 presents the NP-completeness result, Section 3 gives the polynomial time algorithm for series-parallel graphs and Section 4 finishes with the discussion.

*This research was partially supported by the Christian Doppler Laboratorium für Diskrete Optimierung and by the Fonds zur Förderung der wissenschaftlichen Forschung, Project P8971-PHY.

†TU Graz, Institut für Mathematik B, Kopernikusgasse 24, A-8010 Graz, Austria

‡TU Graz, Institut für Theoretische Informatik, Klosterwiesgasse 32/II, A-8010 Graz, Austria

§Mathematical Department, Zhejiang University, Hangzhou, People's Republic of China

2 Why the problem is NP-complete

In this section, we prove that PG2T is NP-complete. The proof is done by a two-step reduction from the following special case of the Hamiltonian Path problem.

> HAMILTONIAN PATH IN CUBIC GRAPHS (HP3)
>
> **Input.** A graph $G' = (V', E')$ such that all vertices in V' are of degree three with the exception of the degree one vertices s and t.
>
> **Question.** Does there exist a Hamiltonian Path of G' that starts in s and ends in t ?

> TWO COVERING TREES PROBLEM (TCT)
>
> **Input.** A graph $G'' = (V'', E'')$, two disjoint subsets F_1 and F_2 of E''.
>
> **Question.** Do there exist two edge-disjoint trees T_1 and T_2 in G'' such that T_i contains all edges in F_i, $i = 1, 2$?

To be precise, we will show that HP3 is polynomial time reducible to TCT, and then that TCT is polynomial time reducible to PG2T. This clearly establishes the NP-completeness result claimed above.

Hence, let us consider some instance $G' = (V', E')$, $s, t \in V'$ of HP3. We will construct a corresponding instance of TCT that is solvable if and only if HP3 is solvable. This is done in three steps as follows.

(i) First, we subdivide every edge $e = (u, v) \in E'$ into two subedges $(u, e(m))$ and $(e(m), v)$ by introducing a new vertex $e(m)$. Furthermore we introduce a single new vertex c. Vertex c is connected to all vertices $e(m)$ by an edge which is put into F_2.

(ii) We perform the following construction for every $v \in V'$ of degree three: Two new vertices v^* and $\tilde{v}$ are introduced together with the two edges (v, v^*) and $(v, \tilde{v})$. The edge (v, v^*) is put into F_1, the edge $(v, \tilde{v})$ into F_2.

(iii) Finally, we introduce two new vertices s^* and t^* and two edges (s, s^*) and (t, t^*) that are both put into F_1.

We claim that the designed instance of TCT is solvable if and only if G' has a Hamiltonian Path.

(If): Assume, a Hamiltonian Path exists. Our tree T_1 simply consists of all edges in F_1 together with all subdivided edges of the Hamiltonian Path (i.e. if the edge $e = (u, v)$ is in the Hamiltonian Path, we put the two edges $(u, e(m))$, $(e(m), v)$ into the tree). It is trivial to check that this edge set is connected, without cycles and contains all edges in F_1.

Hence, it remains to show that the set E^* of remaining edges also forms a tree. First, we will argue that E^* is connected. Consider some vertex v of V' and the three incident subedges $(v, e_1(m))$, $(v, e_2(m))$ and $(v, e_3(m))$. The Hamiltonian Path uses exactly two of the edges e_1, e_2 and e_3. Therefore, the edge $(v, \tilde{v})$ is connected to vertex c via the unused edge.

E^* contains all edges $(c, e_i(m))$ incident to c. Some of the vertices $e_i(m)$ are of degree one in E^*, some of them are incident to two edges $(u, e_i(m))$ and $(v, e_i(m))$.

Finally, there are the corresponding edges (u, u^*) and (v, v^*) appended to u respectively v. Hence, E^* is a tree of radius three with center c and the proof of the (If)-part is complete.

(Only if): Now we assume that the TCT-instance is solvable. Consider T_1 and call an edge $e = (u, v)$ in E' *complete* iff both subedges $(u, e(m))$ and $(e(m), v)$ are in T_1. We claim that the complete edges constitute a Hamiltonian Path in G'.

Every degree three vertex v in G' is incident to at most two complete edges (otherwise, the edge $(v, \tilde{v})$ in F_2 would be separated from T_2). Vertices s and t are incident to exactly one complete edge.

We remove from T_1 all edges that are neither in F_1 nor subedge of a complete edge. It is easy to check that these removals cannot disconnect T_1. Then we remove all edges in F_1 and replace the remaining subedges by the corresponding complete edges. Since each vertex is of degree at most two and since s and t are of degree one, the resulting graph is a path spanning all vertices in V'. This completes the proof of the (Only If)-part.

What we proved till now suffices to establish the NP-completeness of TCT. However, we are interested in proving the NP-completeness of PG2T, and to this end we need the following lemma.

Lemma 2.1 *Given an instance of* HP3*, we can compute in polynomial time an instance* $G'' = (V'', E'')$*,* F_1*,* F_2 *of* TCT*, such that* HP3 *is solvable iff* TCT *is solvable and such that the following four conditions hold.*

(C1) TCT *is solvable if and only if there exist two edge-disjoint connected subgraphs* S_1 *and* S_2 *such that* S_i *contains all edges in* F_i*,* $i = 1, 2$.

(C2) If TCT *is solvable, then there exists a solution that uses all edges in* E''.

(C3) $|F_1|$ *and* $|F_2|$ *are two distinct prime numbers.*

(C4) F_1 *and* F_2 *both contain at least one edge with one endvertex of degree one.*

Proof. To see (C1), we just have to check that in the proof of the (Only If)-part above, we did not exploit the fact that T_1 is a tree but only the connectedness of T_1. (C2) follows from the proof of the (If)-part.

To ensure that (C3) and (C4) hold, we first compute a prime p_1 with $|F_1| < p_1 < 2|F_1|$. Such a prime exists by Chebyshev's theorem. The prime can be computed in polynomial time, since $|F_1|$ is unary encoded by enumerating its elements. By similar arguments, we can find another prime $p_2 \neq p_1$ with $|F_2| < p_2 < 4|F_2|$.

Then for $i = 1, 2$, we take an edge $e_i = (v_i, u_i) \in F_i$, create $p_i - |F_i|$ new vertices for V'' and connect all these new vertices to v_i by new edges that are added to F_i. Obviously, this new instance of TCT fulfills (C3) and (C4), it is solvable if and only if the original instance was solvable, and conditions (C1) and (C2) still hold. □

Now we consider an instance $G'' = (V'', E'')$, $F_1, F_2 \subseteq E$ of TCT as described in the statement of Lemma 2.1. We construct an instance of PG2T that is solvable exactly if TCT is solvable. Our construction is as follows (let $n = 2|E''|$, $p_1 = |F_1|$, $p_2 = |F_2|$).

(i) We subdivide every edge e in E'' by a new vertex $v(e)$. If $e \in F_1$, we append to $v(e)$ a path of length $p_2 n^2$. Similarly, if $e \in F_2$ then we append to its subdividing vertex a path of length $p_1 n^2$.

(ii) Let $e_i = (u_i, v_i) \in F_i$ with degree of v_i equal to one, $i = 1, 2$, denote the two edges that exist by (C4). We connect v_1 and v_2 by a path of length $2|E''|$.

Clearly, the size of the new graph $G = (V, E)$ is polynomial in the size of G'', and the construction can be performed in polynomial time. The total number $|E|$ of edges in the new graph is $2p_1p_2n^2 + 4|E''|$. We claim that the designed instance of PG2T is solvable if and only if TCT has a solution.

(If): Let T_1 and T_2 constitute a solution of TCT that uses all edges in E''. Let n_1 denote twice the number of edges in T_1, $0 < n_1 < 2|E''|$. We put into E_1 all the edges corresponding to T_1, i.e. subdivided edges of G'' and the corresponding appended paths. Moreover, we put into E_1 the $2|E''| - n_1$ edges of the path defined in (ii) that are nearest to v_1.

Thus, E_1 contains p_1 appended paths with p_2n^2 edges per path, together with a number n_1 of subdivided edges from G'', together with $2|E''| - n_1$ edges from the path defined in (ii). This gives a total number of $p_1p_2n^2 + 2|E''| = |E|/2$ edges in E_1. It is easy to see that E_1 is cyclefree and connected, since T_1 is cyclefree and connected. The same holds for $E - E_1$.

(Only If): Assume, the PG2T-instance has a solution E_1, E_2. Each of the appended paths defined in (i) is contained as a whole either in E_1 or in E_2.

We claim that all p_1 paths of length p_2n^2 are in one of the E_i, and all p_2 paths of length p_1n^2 are in E_{3-i}. Suppose otherwise: Let E_1 contain x_1 paths of length p_1n^2 $(0 < x_1 \leq p_2)$ and x_2 paths of length p_2n^2 $(0 < x_2 \leq p_1)$. Then the facts $0 < x_1 \leq p_2$ and $0 < x_2 \leq p_1$ imply $x_1p_1n^2 + x_2p_2n^2 \neq p_1p_2n^2$. W.l.o.g. E_1 contains at least as many edges of the appended paths as E_2. This yields a contradiction, since

$$|E_1| \geq p_1p_2n^2 + n^2 > p_1p_2n^2 + 2|E''| = |E|/2.$$

To complete the proof, we show that there exists a connected subgraph S_1 of G'' that covers F_1. Using a symmetric argument for F_2 and condition (C1) of Lemma 2.1, this implies the existence of a solution to the TCT-instance. W.l.o.g. let E_1 connect all appended paths corresponding to edges in F_1. We define S_1 to contain all edges in F_1 together with all edges in E'' for which *both* subedges are in E_1 (if only one of the subedges is in E_1, it cannot contribute to the connectivity of E_1). It is easy to check that S_1 is connected.

Summarizing, we have proved the following theorem.

Theorem 2.2 *The problem* PG2T *is NP-complete.* □

3 Series-parallel graphs are easy to treat

The class of series-parallel graphs is a well-known model of series-parallel electrical networks. Many difficult combinatorial problems for graphs become easy when restricted to series-parallel graphs, see e.g. Tamkamizawa, Nishizeki and Saito [5]. In this section, we show that the same holds for the partitioning problem of a graph into two trees, i.e. we will give a polynomial time algorithm for this problem on series-parallel graphs.

One possibility to define series-parallel graphs is via two-terminal graphs, cf. Duffin [1]. A *two-terminal graph* $G = (V, E)$ is a graph with two special vertices

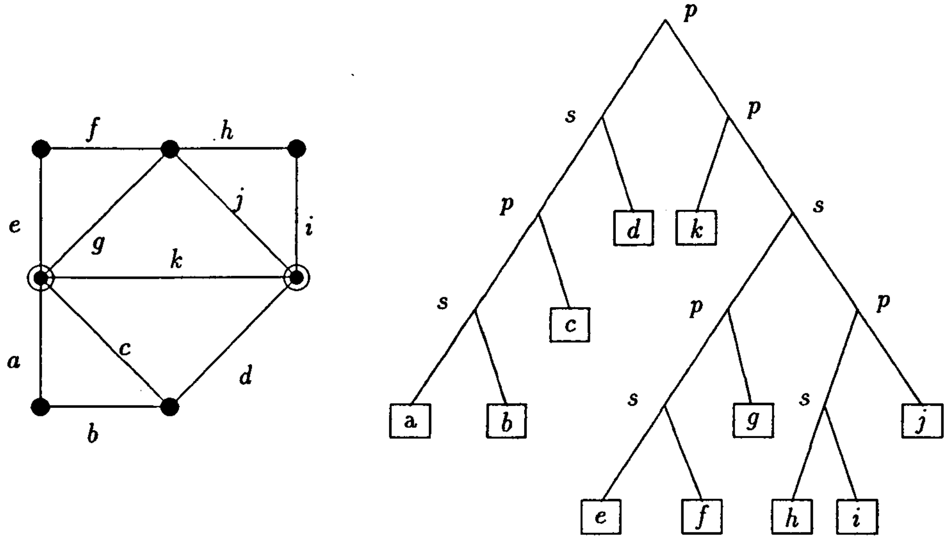

Figure 1: A TTSP graph and its binary decomposition tree

that are called the *left terminal* t_l and the *right terminal* t_r. For two-terminal graphs $G_i = (V_i, E_i)$ with terminals t_l^i and t_r^i, $1 \le i \le 2$, we define the following two operations.

- The *series connection* $G_s = G_1 * G_2$ of G_1 and G_2 results from identifying the right terminal of G_1 with the left terminal of G_2. The obtained graph G_s is regarded as a two-terminal graph with with left terminal t_l^1 and right terminal t_r^2.
- The *parallel connection* $G_p = G_1//G_2$ of G_1 and G_2 results from identifying both right terminals with each other and both left terminals with each other. The terminal vertices of G_p simply are the identified terminals.

Now a *two-terminal series-parallel graph* (TTSP) is defined as follows:

(i) The graph consisting of two terminals connected by a single edge is a TTSP.

(ii) If G_1 and G_2 are TTSPs, then $G_1 * G_2$ and $G_1//G_2$ are TTSPs.

(iii) No other graphs than those defined by (i) and (ii) are TTSPs.

Finally, a graph is a *series-parallel graph* iff it is the underlying graph of a TTSP (i.e. the terminals are considered as ordinary vertices).

It is well-known that decomposing a series-parallel graph into its atomic parts according to the series and parallel operations can be done in linear time. Essentially, such a decomposition corresponds to a binary tree where all interior vertices are labeled by s or p (series or parallel connection) and where all leaves correspond to edges of the graph (see Figure 1 for an illustration). We associate with every interior vertex v of the decomposition tree the series-parallel graph $G(v)$ defined by the subtree rooted in v.

The usual way to deal with problems on series-parallel graphs is dynamic programming via the decomposition tree, and this approach also works in our case.

Let us consider a TTSP graph $G = (V, E)$, and one of the TTSP components $G(v)$ of G associated with one of the vertices v of the decomposition tree of G, and let t_l and t_r denote the terminals of $G(v)$. Let T be a subtree of G, and let T' denote the edge-induced subgraph of T induced by the edges in $T \cap G(v)$. We distinguish five combinatorial types for T'.

(T1) T' consists of two connected components, one containing terminal t_l and the other one containing t_r.

(T2) T' is connected and contains both terminals t_l and t_r.

(T3) T' is connected and contains only terminal t_l but not t_r.

(T4) T' is connected and contains only terminal t_r but not t_l.

(T5) T' is connected and contains neither t_r nor t_l.

Clearly, type (T1) covers the only possibility of not connected T' (In this case, T can only be connected via some path going from t_l to t_r outside of $G(v)$). The remaining four types (T2), (T3), (T4), and (T5) cover all possibilities for connected graphs T'. Note that a T' of type (T1) consists of exactly two trees, and a T' of one of the other types is a tree itself.

We introduce twenty-five two-dimensional boolean arrays $A_{ij}[v, m]$, $1 \le i, j \le 5$. The first index v runs through all vertices of the binary decomposition tree, the second index m runs from 1 to $|V|$. $A_{ij}[v, m]$ will be set to TRUE if and only if there exists a partition of $G(v)$ into two edge-disjoint subgraphs T_1' and T_2' such that T_1' is of type (Ti) and T_2' is of type (Tj) with respect to $G(v)$, and such that T_1' has exactly m edges.

If we compute the truthvalues of all entries of all arrays $A_{i,j}[*,*]$, we solve the PG2T-problem as a by-product: The root r of the decomposition tree corresponds to the graph $G=(V,E)$ itself. The problem PG2T has a solution if and only if $|E|$ is even and at least one of the sixteen entries $A_{ij}[r,|E|/2]$ with $2 \le i,j \le 5$ is set to true.

Hence, our goal is to compute all entries of the array. This is done in a bottom-up fashion according to the decomposition tree: We start with the entries corresponding to leaves of the decomposition tree, and move up towards the root. The entries corresponding to some vertex v of the decomposition tree are calculated only if all entries corresponding to both sons have already been computed.

The initialization step is trivial, since the leaves of the decomposition tree correspond to TTSPs consisting of a single edge.

The computation of entries corresponding to interior vertices v of the decomposition tree is a little bit more complicated and depends on whether v is labeled s or labeled p. We just sketch two of the 50 possible cases and leave the other cases to the reader as an exercise. (Some combinations like $A_{55}[*,*]$ will only have entries set to FALSE).

(1) Computation of $A_{11}[v,m]$ if v is labeled s: Let v_1 and v_2 denote the right and left son of v. In this case, T_1 may consist of (i) a not-connected part of type (T1) in $G(v_1)$ and a connected part of type (T2) in $G(v_2)$ (or the symmetric possibility with $G(v_1)$ and $G(v_2)$ exchanged), or (ii) of a part of type (T2) or (T3) in $G(v_1)$ and a part of type (T4) in $G(v_2)$ (or again some symmetric possibilities). The same possibilities hold for T_2.

We just check whether there exist corresponding true entries $A_{ij}[v_1,m_1]$ and $A_{kl}[v_2,m_2]$, where m_1, m_2 denote two non-negative integers with $m_1+m_2=m$ and i,j,k,l correspond to appropriate types as explained above.

(2) Computation of $A_{52}[v,m]$ if v is labeled p: Again, let v_1 and v_2 denote the right and left son of v. In this case, T_1 must consist of a part of type (T5) in $G(v_1)$ and $G(v_2)$ and of an empty part in the other subgraph. T_2 must consist of a part of type (T2) in $G(v_1)$ and a part that is not of type (T5) in $G(v_2)$ (or vice versa). Similarly as above, $A_{25}[v,m]$ can be computed by investigating appropriate entries $A_{ij}[v_1,m_1]$ and $A_{kl}[v_2,m_2]$, with numbers $\{m_1,m_2\}=\{0,m\}$.

Since all the operations used in the computations of the $A_{ij}[v,m]$ can be performed in polynomial time, we may formulate the following summarizing theorem.

Theorem 3.1 *The problem* PG2T *is solvable in polynomial time if the graph under consideration is series-parallel.* □

4 Discussion

In this paper, we proved that the problem of partitioning a graph into two trees is NP-complete in general, and that the problem is polynomial time solvable for the class of series-parallel graphs.

A similar but simpler version of the dynamic programming approach used for series-parallel graphs in Section 3 succeeds to show that the problem can be solved in polynomial time for trees.

The problem is also polynomial time solvable on the classes of interval graphs, cographs, circular arc graphs, chordal graphs and split graphs (see Johnson [3] for definitions). These results are rather easy to see: The graphs in these graph classes tend to be rather dense and to contain large cliques, whereas a graph $G = (V, E)$ that is partitionable into two trees must fulfill $|E| \leq 2|V| - 2$ and cannot contain cliques of size greater or equal to five. Consequently, most of the graphs in these classes may be a priori disregarded, whereas the remaining 'reasonable' graphs possess a rather rigid and primitive structure. (E.g. a 'reasonable' split graph consists of a clique C with at most four vertices, an independent set I and some edges between C and I).

We do not elaborate on these questions. The surprising part of our results is not that the problem is easy on specially structured graphs, but that the problem is hard in general.

References

[1] R.J.Duffin, Topology of series-parallel networks, *J. Math. Applic.* **10**, 1965, 303–318.

[2] M.R.Garey and D.S.Johnson, *Computers and Intractability, A guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[3] D.S.Johnson, The NP-completeness column: an ongoing guide, *J. Algorithms* **6**, 1985, 434–451.

[4] E.Lawler, *Combinatorial Optimization, Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[5] K.Tamkamizawa, T.Nishizeki and N.Saito, Linear-time computability of combinatorial problems on series-parallel graphs, *J. Assoc. Comput. Mach.* **29**, 1982, 623–641.

Received August 30, 1993

Subscription information anl mailing address for editorial correspondence:

Acta Cybernetica
Árpád tér 2.
Szeged
H-6720 Hungary

CONTENTS

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1994. szeptember
Terjedelem: 7,12 (B/5) ív