

G. GUAIANA

A. RESTIVO

S. SALEMI

Complete subgraphs of bipartite graphs and applications to trace languages

Informatique théorique et applications, tome 24, n° 4 (1990),
p. 409-417

http://www.numdam.org/item?id=ITA_1990__24_4_409_0

© AFCET, 1990, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

COMPLETE SUBGRAPHS OF BIPARTITE GRAPHS AND APPLICATIONS TO TRACE LANGUAGES (*)

by G. GUAIANA ⁽¹⁾, A. RESTIVO ⁽¹⁾ and S. SALEMI ⁽¹⁾

Communicated by J. E. PIN

Abstract. – Motivated by a concurrency problem, a lattice describing a certain concurrent behaviour is constructed, whose elements are complete subgraphs of a bipartite graph. As an application of this construction, one derives, extending a result of Cori and Perrin, an explicit computation for the product of trace languages.

Résumé. – On construit un treillis associé au comportement de processus concurrents. Les éléments de ce treillis sont les sous graphes bipartis complets d'un graphe biparti. Une application de cette construction est la détermination effective de l'automate reconnaissant le produit de deux langages de traces reconnaissables; cette construction précise un résultat de R. Cori et D. Perrin.

1. INTRODUCTION

Let us consider a process which is composed by an arbitrary sequence of actions from a given set A "followed" by an arbitrary sequence of actions from a given set B . The sequences of actions from A (resp. from B) are described by the words of the free monoid A^* (resp. B^*). The "serial" behaviour of this system is then described by the words of the set A^*B^* and can be represented as follows



(*) Received September 1988, revised in April 1989.

(1) Dipartimento di Matematica e Applicazioni, Università di Palermo, via Archirafi, 34, 90123 Palermo, Italy.

Let us now suppose that some actions from A are independent from some actions from B . The fact that the actions $a \in A$ and $b \in B$ are independent (or *concurrent*) means that the results on the system of the sequences ab and ba are the same: we say that a and b *commute*. We then define a symmetric relation $\Delta \subseteq A \times B$, called the *concurrency relation*, and consider the congruence of the free monoid Σ^* , where $\Sigma = A \cup B$, generated by the set of pairs (ab, ba) with $(a, b) \in \Delta$. We denote by $M(\Sigma, \Delta)$ the quotient of Σ^* by this congruence: it is called the *free partially commutative monoid* [3] generated by Σ with respect to the concurrency relation Δ . Subsets of $M(\Sigma, \Delta)$ may be identified with subsets of Σ^* closed with respect to this congruence relation. If X is a subset of Σ^* , its closure by this relation is denoted by $[X]_\Delta$ or simply by $[X]$.

In this formalism the concurrent behaviour of the system can be described by the set $[A^* B^*]$, in the sense that any word of this set is equivalent (produces the same result on the system) to a serial behaviour, *i.e.* a word of $A^* B^*$.

In this paper we give a lattice-theoretic characterization of this concurrent behaviour. We represent the concurrency relation Δ by a bipartite graph and we construct a lattice whose elements are the maximal complete subgraphs of the bipartite graph. This lattice provides three related descriptions of the concurrent behaviour:

- (1) A “partial order semantic” of the concurrent behaviour.
- (2) An automaton recognizing $[A^* B^*]$.
- (3) An explicit regular expression for the set $[A^* B^*]$.

In the second part of the paper an application of this construction is derived which gives a simple algorithm for the product of two trace languages. In particular a new proof and a generalization of a result of Cori and Perrin [5] is obtained.

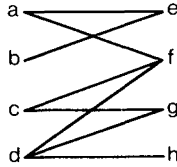
2. THE LATTICE OF THE CONCURRENT BEHAVIOUR

The concurrency relation $\Delta \subseteq A \times B$ can be represented by a *bipartite graph* (A, B, Δ) , *i.e.* a graph whose point set can be partitioned into two subsets A and B such that every line of the graph joins A with B .

A subgraph (A', B', Δ') of (A, B, Δ) is *complete* (as a bipartite graph) if $\Delta' = A' \times B'$ with $A' \subseteq A$ and $B' \subseteq B$. Remark that our definition includes also the case in which either A' or B' are empty. A complete subgraph of (A, B, Δ) is *maximal* if it is not a proper subgraph of any other complete

subgraph. For instance, if for all $b \in B$, $(A, \{b\}, A \times \{b\})$ is not a subgraph of (A, B, Δ) , then $(A, \emptyset, \emptyset)$ (\emptyset is the empty set) is a maximal complete subgraph of (A, B, Δ) . Remark that a maximal complete subgraph $(A', B', A' \times B')$ is uniquely specified by the set $\Sigma' = A' \cup B'$.

Example: Let $A = \{a, b, c, d\}$, $B = \{e, f, g, h\}$ and Δ given by the following bigraph:



The maximal complete subgraphs are given by the following subsets of $\Sigma = A \cup B$:

$$\begin{aligned} \Sigma_1 &= \{a, b, c, d\} = A \\ \Sigma_2 &= \{a, c, d, f\} \\ \Sigma_3 &= \{a, b, e\} \\ \Sigma_4 &= \{c, d, f, g\} \\ \Sigma_5 &= \{a, e, f\} \\ \Sigma_6 &= \{d, f, g, h\} \\ \Sigma_7 &= \{e, f, g, h\} = B. \end{aligned}$$

Let us denote by $S(\Delta) = \{\Sigma_1, \Sigma_2, \dots, \Sigma_n\}$ the set of maximal complete subgraphs of (A, B, Δ) . A partial order can be introduced in $S(\Delta)$ as follows:

$$\Sigma_i \leq \Sigma_j \quad \text{if} \quad \Sigma_i \cap B \subseteq \Sigma_j \cap B.$$

By the definitions it follows that if

$$\Sigma_i \cap B \subseteq \Sigma_j \cap B \quad \text{then} \quad \Sigma_i \cap A \subseteq \Sigma_j \cap A.$$

THEOREM 1: *The poset $(S(\Delta), \leq)$ is a lattice.*

Proof. — For any $\Sigma_r \in S(\Delta)$ introduce the following notation

$$\Sigma_r \cap A = \Sigma_{r,A}, \quad \Sigma_r \cap B = \Sigma_{r,B}.$$

We first prove that, for any pair of elements of $S(\Delta)$, there exists a least upper bound. Let us first remark that for any $\Sigma_i, \Sigma_j \in S(\Delta)$, the subgraph

$$\Delta_{ij} = (\Sigma_{iA} \cap \Sigma_{jA}) \times (\Sigma_{iB} \cup \Sigma_{jB})$$

is a complete subgraph of (A, B, Δ) . However it is not in general maximal.

It is then contained in a maximal complete subgraph $\Sigma_k \in S(\Delta)$. We prove that Σ_k is unique (*i.e.* there exists a unique maximal complete subgraph containing Δ_{ij}) and that Σ_k is the least upper bound of Σ_i and Σ_j . By definition

$$\Sigma_{iA} \cap \Sigma_{jA} \subseteq \Sigma_{kA} \quad \text{and} \quad \Sigma_{iB} \cup \Sigma_{jB} \subseteq \Sigma_{kB}$$

We first prove that $\Sigma_{kA} = \Sigma_{iA} \cap \Sigma_{jA}$. Indeed if there exists $a \in A$ such that

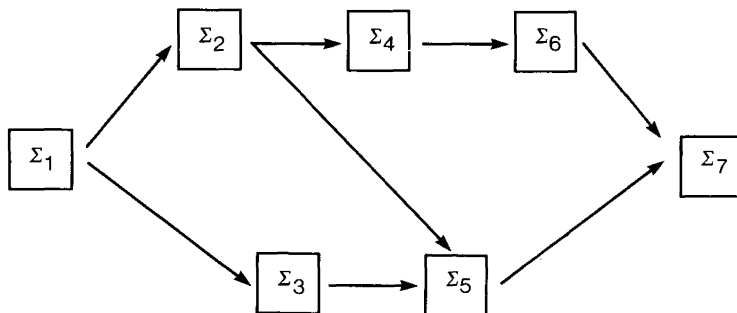
$$a \in \Sigma_{kA} \quad \text{and} \quad a \notin \Sigma_{iA} \cap \Sigma_{jA}$$

then

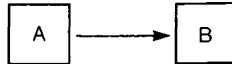
$$\Sigma_i \cup \{a\} \quad \text{and} \quad \Sigma_j \cup \{a\}$$

are both complete subgraphs, which contradicts the hypothesis that Σ_i and Σ_j are maximal complete subgraphs. By the remark that any element $\Sigma_r \in S(\Delta)$ is uniquely specified by the intersection $\Sigma_r \cap A$, one derives the unicity of Σ_k . Since $\Sigma_i \leq \Sigma_k$ and $\Sigma_j \leq \Sigma_k$, in order to prove that Σ_k is the least upper bound, let us consider an element $\Sigma_p \in S(\Delta)$ such that $\Sigma_i \leq \Sigma_p$ and $\Sigma_j \leq \Sigma_p$. These two inequalities imply that $\Sigma_{pA} \subseteq \Sigma_{iA} \cap \Sigma_{jA} = \Sigma_{kA}$. One then derives that $\Sigma_p \geq \Sigma_k$. In a symmetric way one proves that there exists a greatest lower bound. This concludes the proof.

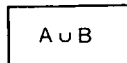
The lattice corresponding to previous example may be represented as follows



The lattice $S(\Delta)$ provides a “partial order semantic” of the concurrent behaviour: the ordering relation represents the relation of time successiveness of sequence of actions in possible executions of the system. In particular $\Sigma_i \leq \Sigma_j$ means that an arbitrary sequence of actions from Σ_i precedes an arbitrary sequences of actions from Σ_j in a correct run of the system. For instance in the special case $\Delta = \emptyset$, *i. e.* there is a complete dependence between the actions of A and B , the lattice $S(\Delta)$ is the lattice with two elements



and one obtains the “serial” behaviour. On the contrary, in the case $\Delta = A \times B$, *i. e.* all the elements of A are independent from all the elements of B , the lattice $S(\Delta)$ is the trivial lattice with only one element



and one obtains the maximal concurrency.

In order to give a language-theoretic formulation of these assertions let us introduce some notations.

Let $C = (\Sigma_1, \Sigma_2, \dots, \Sigma_k)$ be a maximal chain of $S(\Delta)$ and let $L(C)$ denote the following subset of Σ^* :

$$L(C) = \Sigma_1^* \Sigma_2^* \dots \Sigma_k^*.$$

THEOREM 2: *The set $[A^* B^*]$ is obtained as the union of the sets $L(C)$ with C ranging over all maximal chains of $S(\Delta)$.*

Proof: Let us first remark that, if $C = (\Sigma_1, \Sigma_2, \dots, \Sigma_k)$ is a maximal chain, one has:

$$\begin{aligned} \Sigma_{1A} \supset \Sigma_{2A} \supset \dots \supset \Sigma_{kA} \\ \Sigma_{1B} \subset \Sigma_{2B} \subset \dots \subset \Sigma_{kB}. \end{aligned}$$

If K denotes the union of the sets $L(C)$ with C ranging over all maximal chains, we first prove the inclusion $K \subseteq [A^* B^*]$.

If $u \in L(C)$ with $C = (\Sigma_1, \Sigma_2, \dots, \Sigma_k)$ a maximal chain, then

$$u = u_1 u_2 \dots u_k, \quad u_i \in \Sigma_i^*.$$

Let π_A be the projection from the free monoid Σ^* into A^* , π_B the projection from Σ^* into B^* and let \sim denote the congruence of Σ^* generated by Δ . Since for any i , Σ_i is a complete subgraph of (A, B, Δ) , the words $\pi_A(u_i)$ and

$\pi_B(u_i)$ commute. One has

$$u \sim \pi_A(u_1) \pi_B(u_1) \pi_A(u_2) \pi_B(u_2) \dots \pi_A(u_k) \pi_B(u_k).$$

By using previous chains of inclusions, one then derives:

$$u \sim \pi_A(u_1) \pi_A(u_2) \dots \pi_A(u_k) \pi_B(u_1) \pi_B(u_2) \dots \pi_B(u_k) \in [A^* B^*].$$

This proves that $K \subseteq [A^* B^*]$.

In order to prove the inclusion $[A^* B^*] \subseteq K$, since $A^* B^* \subseteq L(C)$ for any maximal chain C , we have only to prove that $[K] = K$. It suffices to show that for $a \in A$, $b \in B$ with $(a, b) \in \Delta$, for any $u, v \in \Sigma^*$ and for any maximal chain C , the following relations are verified:

(i) $uabv \in L(C) \Rightarrow ubav \in L(C')$ for some maximal chain C' ;

(ii) $ubav \in L(C) \Rightarrow uabv \in L(C')$ for some maximal chain C' .

In order to prove (i), suppose that $uabv \in \Sigma_1^* \Sigma_2^* \dots \Sigma_k^*$. If there exists an index i ($1 \leq i \leq k$) such that

$$u \in \Sigma_1^* \dots \Sigma_i^*, \quad v \in \Sigma_i^* \dots \Sigma_k^*, \quad a, b \in \Sigma_i$$

then, since Σ_i is a complete subgraph of (A, B, Δ) , the relation (i) is verified. Otherwise, one has the following situation:

$$u \in \Sigma_1^* \dots \Sigma_i^*, \quad v \in \Sigma_j^* \dots \Sigma_k^*, \quad a \in \Sigma_{iA}, \quad b \in \Sigma_{jB}, \quad i < j.$$

Since

$$a \in \Sigma_{iA}, \quad \Sigma_{jA} \subset \Sigma_{iA}, \quad a \notin \Sigma_{jA}$$

and

$$b \in \Sigma_{jB}, \quad \Sigma_{iB} \subset \Sigma_{jB}, \quad b \notin \Sigma_{iB}$$

one derives that

$$(\Sigma_{jA} \cup \{a\}) \times (\Sigma_{iB} \cup \{b\})$$

is a complete subgraph of (A, B, Δ) . It is then contained in a maximal complete subgraph $\Sigma_m \in \mathcal{S}(\Delta)$. By the relations

$$\Sigma_{jA} \cup \{a\} \cong \Sigma_m \subset \Sigma_{iA}, \quad \Sigma_{iB} \cup \{b\} \cong \Sigma_m \subset \Sigma_{jB}$$

one derives that

$$\Sigma_i < \Sigma_m < \Sigma_j.$$

There exists then in $S(\Delta)$ a maximal chain C' which begins with $\Sigma_1, \dots, \Sigma_i$, ends with $\Sigma_j, \dots, \Sigma_k$ and contains Σ_m . C' verifies the condition $ubav \in L(C')$, since $u \in \Sigma_1^* \dots \Sigma_i^*$, $ba \in \Sigma_m^*$, $v \in \Sigma_j^* \dots \Sigma_k^*$. This concludes the proof of (i). The proof of (ii) is simpler. Also here the only non trivial case is given by the following situation:

$$u \in \Sigma_1^* \dots \Sigma_i^*, \quad v \in \Sigma_j^* \dots \Sigma_k^*, \quad b \in \Sigma_{iB}, \quad a \in \Sigma_{jA}, \quad i < j.$$

Since $\Sigma_{iB} \subseteq \Sigma_{jB}$ (resp. $\Sigma_{jA} \subseteq \Sigma_{iA}$), then $b \in \Sigma_{jB}$ (resp. $a \in \Sigma_{iA}$). It follows that $uabv \in L(C)$. This concludes the proof of the theorem.

Remark 1: This theorem provides an explicit regular expression for the set $[A^* B^*]$ of words corresponding to correct runs of the system. For instance the maximal chains corresponding to previous example are: $(\Sigma_1, \Sigma_2, \Sigma_4, \Sigma_6, \Sigma_7)$, $(\Sigma_1, \Sigma_2, \Sigma_5, \Sigma_7)$, $(\Sigma_1, \Sigma_3, \Sigma_5, \Sigma_7)$ and the regular expression representing the set $[A^* B^*]$ is:

$$\begin{aligned} [A^* B^*] = & (a+b+c+d)^* (a+c+d+f)^* (c+d+f+g)^* (d+f+g+h)^* (e+f+g+h)^* \\ & + (a+b+c+d)^* (a+c+d+f)^* (a+e+f)^* (e+f+g+h)^* \\ & + (a+b+c+d)^* (a+b+e)^* (a+e+f)^* (e+f+g+h)^*. \end{aligned}$$

Remark 2: The lattice $S(\Delta)$ provides also a (nondeterministic) finite automaton (with ϵ -moves) which recognizes the set $[A^* B^*]$. The elements of $S(\Delta)$ correspond to the states of the automaton. The transitions of the automaton are defined as follows:

- If q is the state corresponding to the element Σ_i of $S(\Delta)$, for any $a \in \Sigma_i$, there is a transition from q back to q itself labelled by a .
- If $\Sigma_i \leq \Sigma_j$, there is an ϵ -move from Σ_i to Σ_j .

All states are start states and accepting states.

3. ON THE PRODUCT OF TRACE LANGUAGES

The problem in previous sections may be considered as a particular problem in *trace theory*. Trace theory was initiated by Mazurkiewicz [7] for describing the behaviour of concurrent systems. Its relationships with free partially commutative monoids and the study of different families of trace languages have been investigated in several papers (cf. [1], [2], [4], [5], [8], [9], [10]).

We call *trace* an element of $M(\Sigma, \Delta)$ and *trace language* any subset of $M(\Sigma, \Delta)$. A trace language may be identified with a subset of Σ^* closed with

respect to the congruence induced by Δ . A trace language is *recognizable* (resp. *context-free*) if it is recognizable (resp. context-free) as a subset of Σ^* .

Let A and B be two disjoint alphabets. Let $\Sigma = A \cup B$ and let $\Delta \subseteq A \times B$ be a symmetric relation. Let X be a language over the alphabet A and let Y be a language over the alphabet B . The shuffle $X \sqcup Y$ of the languages X and Y is defined as follows:

$$X \sqcup Y = \{u_1 v_1 u_2 v_2 \dots u_n v_n \mid n > 0, u_1 \dots u_n \in X, v_1 \dots v_n \in Y\}.$$

It is well known (cf. [6]) that the shuffle of two recognizable languages is recognizable and that the shuffle of a context-free language and a recognizable language is a context-free language. Next theorem provides an expression for the closure $[XY]$ of the product XY with respect to the congruence induced by Δ .

THEOREM 3: $[XY] = [A^* B^*] \cap (X \sqcup Y)$.

Proof: The inclusion $[XY] \subseteq [A^* B^*] \cap (X \sqcup Y)$ is trivially verified. In order to prove the converse inclusion, let us consider an arbitrary element $u \in [A^* B^*] \cap (X \sqcup Y)$. $u \in [A^* B^*]$ with $A \cap B = \emptyset$, implies $u \sim \pi_A(u) \pi_B(u)$. $u \in X \sqcup Y$ and $A \cap B = \emptyset$ implies $\pi_A(u) \in X$ and $\pi_B(u) \in Y$. One derives that $u \in [XY]$. This concludes the proof.

Let us now consider a single alphabet A and a symmetric relation $\Delta \subseteq A \times A$, and two languages X, Y over A closed with respect to the congruence induced by Δ . Let B be another alphabet (disjoint from A) such that $\text{Card}(A) = \text{Card}(B)$ and let $\theta: A \rightarrow B$ be a one-to-one mapping between A and B . θ can be extended to an isomorphism of the free monoids A^* and B^* . Let $\Delta_0 \subseteq A \times B$ be a symmetric relation defined as follows:

$$(a, b) \in \Delta_0 \Leftrightarrow (a, \theta^{-1}(b)) \in \Delta.$$

Let $\Sigma = A \cup B$ and let $\alpha: \Sigma^* \rightarrow A^*$ be the alphabetic morphism defined as follows:

$$\alpha(a) = \begin{cases} a & \text{if } a \in A \\ \theta^{-1}(a) & \text{if } a \in B. \end{cases}$$

The closure $[XY]_\Delta$ of the product $XY \subseteq A^*$ with respect to Δ can be expressed by this formula:

$$[XY]_\Delta = \alpha([X\theta(Y)]_{\Delta_0}) = \alpha([A^* B^*]_{\Delta_0} \cap (X \sqcup \theta(Y))).$$

One then derives the following corollary of theorem 3.

COROLLARY: *The product in $M(\Sigma, \Delta)$ of two recognizable trace languages is a recognizable trace language. The product of a recognizable trace language and a context-free trace language is a context-free trace language.*

This corollary is an extension of a result of Cori and Perrin [5]. Moreover it provides an explicit and simpler computation for the product of two trace languages. Let us further remark that this computation can be partitioned into two independent parts: one part depending only on the concurrency relation Δ and the other part depending only on the languages. By this partition one can derive interesting applications to the design of efficient algorithms for some special combinatorial problems on words over partially commutative alphabets.

REFERENCES

- [1] IJ. J. AALBERSBERG and G. ROZENBERG, *Trace Languages Defined by Regular String Languages*, RAIRO Informatique Théorique et Applications, Vol. 20, 1986, pp. 103-119.
- [2] A. BERTONI, G. MAURI and N. SABATINI, *Equivalence and Membership Problems for Regular Trace Languages*, Lecture Notes in Computer Science, Vol. 140, 1982, pp. 61-71.
- [3] P. CARTIER and D. FOATA, *Problèmes combinatoires de commutation et réarrangements*, Lecture Notes in Mathematics, Vol. 85, 1981.
- [4] R. CORI and Y. METIVIER, *Recognizable Subsets of Some Partially Abelian Monoids*, Theoretical Computer Science, Vol. 35, 1985, pp. 179-189.
- [5] R. CORI and D. PERRIN, *Automates et commutations partielles*, RAIRO Informatique Théorique, 19, 1985, pp. 21-32.
- [6] J. E. HOPCROFT and J. D. ULLMAN, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.
- [7] A. MAZURKIEWICZ, *Concurrent Program Schemes and Their Interpretations*, DAIMI Rep. PB-78, Aarhus Univ., Aarhus, 1977.
- [8] A. MAZURKIEWICZ, *Traces, Histories, Graphs: Instances of a Process Monoid*, Lecture Notes in Computer Science, Vol. 176, 1984, pp. 115-133.
- [9] D. PERRIN, *Words Over a Partially Commutative Alphabet*, NATO ASI Series F12, 1985, pp. 329-340.
- [10] J. SAKAROVITCH, *On Regular Trace Languages*, Theoretical Computer Science, Vol. 52, 1987, pp. 59-75.