G. Guaiana
A. Restivo
S. Salemi

Complete subgraphs of bipartite graphs and applications to trace languages


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

\[ \text{A} \xrightarrow{} \text{B} \]

(*) Received September 1988, revised in April 1989.
(1) Dipartimento di Matematica e Applicazioni, Universita’ 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 $\Sigma^*$, 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 $\Sigma^*$ by this congruence: it is called the free partially commutative monoid [3] generated by $\Sigma$ with respect to the concurrency relation $\Delta$. Subsets of $M(\Sigma, \Delta)$ may be identified with subsets of $\Sigma^*$ closed with respect to this congruence relation. If $X$ is a subset of $\Sigma^*$, 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 $\Delta$ 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.

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 $(A, B, \Delta)$, 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 $(A', B', \Delta')$ of $(A, B, \Delta)$ 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 $(A, B, \Delta)$ 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 $(A, B, \Delta)$, then $(A, \emptyset, \emptyset)$ ($\emptyset$ is the empty set) is a maximal complete subgraph of $(A, B, \Delta)$. 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 $\Delta$ given by the following bigraph:

![Diagram](image)

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

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

Let us denote by $S(\Delta) = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_n\}$ the set of maximal complete subgraphs of $(A, B, \Delta)$. 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

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

**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 $S(A)$, there exists a least upper bound. Let us first remark that for any $\Sigma_i, \Sigma_j \in S(\Delta)$, the subgraph

$$\Delta_{ij} = (\Sigma_i \cap \Sigma_j, A) \times (\Sigma_i \cup \Sigma_j, B)$$

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

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

\[
\Sigma_i \cap \Sigma_j \subseteq \Sigma_k \quad \text{and} \quad \Sigma_i \cup \Sigma_j \subseteq \Sigma_k.
\]

We first prove that $\Sigma_k = \Sigma_i \cap \Sigma_j$. Indeed if there exists $a \in A$ such that

\[
a \in \Sigma_k \quad \text{and} \quad a \notin \Sigma_i \cap \Sigma_j
\]

then

\[
\Sigma_i \cup \{a\} \quad \text{and} \quad \Sigma_j \cup \{a\}
\]

are both complete subgraphs, which contradicts the hypothesis that $\Sigma_i$ and $\Sigma_j$ 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 $\Sigma_k$. Since $\Sigma_i \subseteq \Sigma_k$ and $\Sigma_j \subseteq \Sigma_k$, in order to prove that $\Sigma_k$ is the least upper bound, let us consider an element $\Sigma_p \in S(\Delta)$ such that $\Sigma_i \subseteq \Sigma_p$ and $\Sigma_j \subseteq \Sigma_p$. These two inequalities imply that $\Sigma_p \subseteq \Sigma_i \cap \Sigma_j = \Sigma_k$. One then derives that $\Sigma_p \subseteq \Sigma_k$. 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.

```
Informatique théorique et Applications/Theoretical Informatics and Applications
```
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 $\Sigma_i$ precedes an arbitrary sequences of actions from $\Sigma_j$ 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

$$\begin{array}{c}
A \\
\longrightarrow
\end{array}$$

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

$$A \cup B$$

and one obtains the maximal concurrency.

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

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

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

**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 $C = (\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$ is a maximal chain, one has:

$$\Sigma_1 A \supset \Sigma_2 A \supset \ldots \supset \Sigma_k A \quad \text{and} \quad \Sigma_1 B \subset \Sigma_2 B \subset \ldots \subset \Sigma_k B.$$ 

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 $C = (\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$ a maximal chain, then

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

Let $\pi_A$ be the projection from the free monoid $\Sigma^*$ into $A^*$, $\pi_B$ the projection from $\Sigma^*$ into $B^*$ and let $\sim$ denote the congruence of $\Sigma^*$ generated by $\Delta$. Since for any $i$, $\Sigma_i$ is a complete subgraph of $(A, B, \Delta)$, the words $\pi_A(u_i)$ and
π_B(u_i) commute. One has

\[ u \sim π_A(u_1) π_B(u_1) π_A(u_2) π_B(u_2) \cdots π_A(u_k) π_B(u_k). \]

By using previous chains of inclusions, one then derives:

\[ u \sim π_A(u_1) π_A(u_2) \cdots π_A(u_k) π_B(u_1) π_B(u_2) \cdots π_B(u_k) ∈ [A^* B^*]. \]

This proves that \( K ⊆ [A^* B^*] \).

In order to prove the inclusion \([A^* B^*] ⊆ K\), 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 \( a ∈ A, b ∈ B \) with \((a, b) ∈ Δ\), for any \( u, v ∈ Σ^* \) and for any maximal chain \( C \), the following relations are verified:

(i) \( uabv ∈ L(C) ⇒ ubav ∈ L(C') \) for some maximal chain \( C' \);
(ii) \( uabv ∈ L(C) ⇒ uabv ∈ L(C') \) for some maximal chain \( C' \).

In order to prove (i), suppose that \( uabv ∈ Σ_1^* Σ_2^* \cdots Σ_k^* \). If there exists an index \( i (1 ≤ i ≤ k) \) such that

\[ u ∈ Σ_1^* \cdots Σ_i^*, \quad v ∈ Σ_i^* \cdots Σ_k^*, \quad a, b ∈ Σ_i \]

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

\[ u ∈ Σ_1^* \cdots Σ_i^*, \quad v ∈ Σ_j^* \cdots Σ_k^*, \quad a ∈ Σ_i A, \quad b ∈ Σ_j B, \quad i < j. \]

Since

\[ a ∈ Σ_i A, \quad Σ_j A ⊆ Σ_i A, \quad a ≠ Σ_j A \]

and

\[ b ∈ Σ_j B, \quad Σ_i B ⊆ Σ_j B, \quad b ≠ Σ_i B \]

one derives that

\[ (Σ_j A \cup \{a\}) × (Σ_i B \cup \{b\}) \]

is a complete subgraph of \((A, B, Δ)\). It is then contained in a maximal complete subgraph \( Σ_m ∈ S(Δ) \). By the relations

\[ Σ_j A \cup \{a\} ⊆ Σ_m A \subseteq Σ_i A, \quad Σ_i B \cup \{b\} ⊆ Σ_m B \subseteq Σ_j B \]

one derives that

\[ Σ_i < Σ_m < Σ_j. \]
There exists then in $S(\Delta)$ a maximal chain $C'$ which begins with $\Sigma_1, \ldots, \Sigma_i$, ends with $\Sigma_j, \ldots, \Sigma_k$ and contains $\Sigma_m$. $C'$ verifies the condition $uabv \in L(C')$, since $u \in \Sigma^*_i \ldots \Sigma^*_k$, $ba \in \Sigma^*_m$, $v \in \Sigma^*_k \ldots \Sigma^*_s$. 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:

$$u \in \Sigma^*_i \ldots \Sigma^*_s, \quad v \in \Sigma^*_j \ldots \Sigma^*_k, \quad b \in \Sigma^*_i, \quad a \in \Sigma^*_j, \quad i < j.$$  

Since $\Sigma^*_i \Sigma^*_j \subseteq \Sigma^*_j \Sigma^*_i$ (resp. $\Sigma^*_i \Sigma^*_j \subseteq \Sigma^*_i \Sigma^*_j$), then $b \in \Sigma^*_j \Sigma^*_k$ (resp. $a \in \Sigma^*_i \Sigma^*_j$). 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: $(\Sigma_1, \Sigma_2, \Sigma_4, \Sigma_6, \Sigma_7)$, $(\Sigma_1, \Sigma_3, \Sigma_5, \Sigma_7)$ and the regular expression representing the set $[A^* B^*]$ is:

$$[A^* B^*] = (a + b + c + d) (a + c + d + f) (c + d + f + g) (d + f + g + h) (e + f + g + h) + (a + b + c + d) (a + c + d + f) (a + e + f) (e + f + g + h) + (a + b + c + d) (a + b + e) (a + e + f) (e + f + g + h).$$

**Remark 2:** The lattice $S(\Delta)$ provides also a (nondeterministic) finite automaton (with $\varepsilon$-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 $\Sigma_i$ of $S(\Delta)$, for any $a \in \Sigma_i$, there is a transition from $q$ back to $q$ itself labelled by $a$.
- If $\Sigma_i \subseteq \Sigma_j$, there is an $\varepsilon$-move from $\Sigma_i$ to $\Sigma_j$.

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 $\Sigma^*$ closed with...
respect to the congruence induced by \( \Delta \). A trace language is \textit{recognizable} (resp. \textit{context-free}) if it is recognizable (resp. context-free) as a subset of \( \Sigma^* \).

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 \( X \shuffle Y \) of the languages \( X \) and \( Y \) is defined as follows:

\[
X \shuffle Y = \{ u_1 v_1 u_2 v_2 \ldots u_n v_n \mid n > 0, u_1 \ldots u_n \in X, v_1 \ldots v_n \in Y \}.
\]

It is well known (\textit{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 \(\Delta\).

\textbf{Theorem 3}: \( [XY] = [A^* B^*] \cap (X \shuffle Y) \).

\textbf{Proof}: The inclusion \([XY] \subseteq [A^* B^*] \cap (X \shuffle Y)\) is trivially verified. In order to prove the converse inclusion, let us consider an arbitrary element \( u \in [A^* B^*] \cap (X \shuffle Y) \). Let \( u \in [A^* B^*] \) with \( A \cap B = \emptyset \), implies \( u \sim \pi_A(u) \pi_B(u) \). \( u \in X \shuffle Y \) 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 \( \Delta \). Let \( B \) be another alphabet (disjoint from \( A \)) such that \( \text{Card}(A) = \text{Card}(B) \) and let \( \theta : A \to B \) be a one-to-one mapping between \( A \) and \( B \). \( \theta \) 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:

\[
(a, b) \in \Delta_0 \iff (a, \theta^{-1}(b)) \in \Delta.
\]

Let \( \Sigma = A \cup B \) and let \( \alpha : \Sigma^* \to A^* \) 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 \( \Delta \) can be expressed by this formula:

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

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 $\Delta$ 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


