INFORMATIQUE THÉORIQUE ET APPLICATIONS

A. BERTONI
P. MASSAZZA

On the inclusion problem for finitely ambiguous rational trace languages

Informatique théorique et applications, tome 32, nº 1-3 (1998), p. 79-98
<http://www.numdam.org/item?id=ITA_1998__32_1-3_79_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 32, n° 1-2-3, 1998, pp. 79-98)

ON THE INCLUSION PROBLEM FOR FINITELY AMBIGUOUS RATIONAL TRACE LANGUAGES

by A. BERTONI [1] and P. MASSAZZA [1]

Communicated by C. CHOFFRUT

Abstract. – *In this work we prove that the Inclusion Problem is decidable for a particular class of trace languages that is the class* $R_{\mathrm{FIN}}(\Sigma, C)$ *of finitely ambiguous rational trace languages over an alphabet* $\Sigma = A \uplus B$ *with the commutation relation* $C = (A \times \Sigma \cup \Sigma \times A)\backslash I$. © Elsevier, Paris

Résumé. – Dans cet article nous prouvons que le problème d'inclusion est décidable pour une classe particulière de langages de traces, à savoir la classe $R_{\mathrm{FIN}}(\Sigma, C)$ des langages de traces rationnels finiment ambigus sur l'alphabet $\Sigma = A \uplus B$ avec la relation de commutation $C = (A \times \Sigma \cup \Sigma \times A)\backslash I$. © Elsevier, Paris

1. INTRODUCTION

Trace languages have been introduced by Mazurkiewicz [18] and have been widely studied in the context of the behaviour of concurrent processes. Trace languages are subsets of free partially commutative monoids for which interesting decision problems have been analyzed. In particular, given a class $\mathcal{C}$ of trace languages the Inclusion (Equivalence) Problem for $\mathcal{C}$ consists of deciding whether for L_1, $L_2 \in \mathcal{C}$ it holds $L_1 \subseteq L_2 (L_1 = L_2)$.

It is immediate to see that the Equivalence Problem is reducible to the Inclusion Problem for every class $\mathcal{C}$ since $L_1 = L_2$ iff $L_1 \subseteq L_2$ and $L_2 \subseteq L_1$. So, if Inclusion is decidable then Equivalence is decidable too. Moreover, if a class $\mathcal{C}$ is closed under union then both the problems are either decidable or undecidable since $L_1 \subseteq L_2$ iff $L_1 \cup L_2 = L_2$.

In this paper we deal with particular subclasses of the class $\mathrm{Rat}(\Sigma, C)$ of rational trace languages, a class that has been widely studied and for

(*) Received February 1997, revised October 1997, accepted March 1998.

([1]) Università degli Studi di Milano, Dipartimento di Scienze dell'Informazione, Via Comelico 39, 20135 Milano, Italia.

which many results are known. In particular, using a technique due to Ibarra ([15]), in [1], [13] it is shown that the Equivalence Problem is undecidable for Rat(Σ, C) with the commutation relation ${}_a\overset{b}{\wedge}{}_c$. On the other side, when C is transitive the Inclusion Problem turns out to be decidable ([4]). We also recall that the Equivalence Problem is decidable for the subclass of unambiguous rational trace languages with arbitrary commutation relation ([21]), and that for the same class the Inclusion Problem is undecidable for the commutation relation ${}^a_d\Box^b_c$ ([7]).

The main result we present here is the decidability of the Inclusion Problem for the class $R_{\mathrm{FIN}}(\Sigma, C)$ of finitely ambiguous rational trace languages over an alphabet $\Sigma = A \uplus B$ with the commutation relation $C = (A \times \Sigma \cup \Sigma \times A) \backslash I$. As a consequence, we have that the Equivalence Problem for $R_{\mathrm{FIN}}(\Sigma, C)$ is decidable since this class is closed under union.

We follow a technique used in [6], where the Equivalence Problem is shown to be decidable for a class $\mathcal{C}$ of recursive languages that is c-holonomic and c-closed under intersection (i.e. the elements of $\mathcal{C}$ admit holonomic generating functions, finite computable specifications, and their intersection is in $\mathcal{C}$ with a specification computed in a finite time). Hence, given two trace languages T_1, $T_2 \in R_{\mathrm{FIN}}(\Sigma, C)$ we reduce the problem of deciding whether $T_1 \subseteq T_2$ to the problem of verifying the following relation between generating functions:

$$\phi_{T_2}(x) = \phi_{T_1 \cup T_2}(x).$$

By showing that these generating functions are holonomic it turns out that the Inclusion Problem for $R_{\mathrm{FIN}}(\Sigma, C)$ is reduced to the Equivalence Problem for holonomic functions that is well known to be decidable (see for instance [22]).

The paper is organized as follows. Sections 2, 3 provide us with basic definitions about trace languages, generating functions and formal series. In section 4 we present examples of rational trace languages with generating functions that are not holonomic. Section 5 proves that for an alphabet $\Sigma = A \uplus B$ and a commutation relation $C = (A \times \Sigma \cup \Sigma \times A) \backslash I$ languages in $R_{\mathrm{FIN}}(\Sigma, C)$ admit holonomic generating functions. Section 6 states the decidability of the Inclusion Problem for $R_{\mathrm{FIN}}(\Sigma, C)$ under the previous assumptions on Σ and C. Section 7 gives us important results about the parallel complexity of a set of problems regarding the class $R_{\mathrm{FIN}}(\Sigma, C)$: the Inclusion Problem (in the case of fixed degree of ambiguity), the Generating Function Problem (i.e. the problem of finding a differential equation for the generating function) and the Initial Conditions Problem (i.e. the problem of finding a suitable set of initial conditions for the differential equation satisfied by the generating function of a language). It is shown that all of these problems can be efficiently solved in parallel with boolean circuits of depth $O(\log^2 n)$. We also prove that, unless $\mathbf{P} = \mathbf{NP}$, there exists no sequential algorithm that solves the Inclusion Problem for $R_{\mathrm{FIN}}(\Sigma, C)$ in polynomial time w.r.t. the ambiguity degree.

2. TRACE LANGUAGES AND GENERATING FUNCTIONS

Let $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$, be a finite alphabet and Σ^* be the free monoid generated by Σ. A communication relation on Σ is an irreflexive and simmetric relation $C \subseteq \Sigma \times \Sigma$. We denote by $F(\Sigma, C)$ the free partially commutative monoid Σ^*/ρ_C where $\rho_C \subseteq \Sigma^* \times \Sigma^*$ is the congruence generated by C. The following examples define two f.p.c. monoids that are of particular interest for the results we present later.

EXAMPLE 1: *Let be* $\Sigma_4 = \{\sigma_1, \ldots, \sigma_4\}$ *and* $C_4 = {}^{\sigma_1}_{\sigma_4}\square^{\sigma_2}_{\sigma_3}$. *Then* $F(\Sigma_4, C_4) = \{\sigma_1, \sigma_3\}^* \times \{\sigma_2, \sigma_4\}^*$.

EXAMPLE 2: *Let be* $\Sigma_3 = \{\sigma_1, \sigma_2, \sigma_3\}$ *and* $P_3 = {}_{\sigma_1}\overset{\sigma_2}{\wedge}{}_{\sigma_3}$. *Then* $F(\Sigma_3, P_3) = \{\sigma_1, \sigma_3\}^* \times \{\sigma_2\}^*$.

We call *trace* an element of a free partially commutative monoid $F(\Sigma, C)$. A trace t can be interpreted as an equivalence set of words: given a string $w \in \Sigma^*$ we denote by $[w]_{\rho_C}$ the equivalence class of w (i.e. the trace generated by w) and by $|w|$ its length. Given a symbol σ, we indicate by $|w|_\sigma$ the number of occurrences of σ in w.

A *trace language* is a subset of $F(\Sigma, C)$; given a language $L \subseteq \Sigma^*$ and a commutation relation C the trace language generated by L is

$$[L]_{\rho_C} = \{[w]_{\rho_C} | w \in L\}.$$

A trace language $T \subseteq F(\Sigma, C)$ is *rational* iff it is generated by a regular language $L \subseteq \Sigma^*$. We denote by $\mathrm{Rat}(\Sigma, C)$ the set of rational trace languages on $F(\Sigma, C)$.

DEFINITION 1: *A trace language* $T \in Rat(\Sigma, C)$ *is of ambiguity degree* h *iff it is generated by a regular language* $L \subseteq \Sigma^*$ *such that*

$$\forall w \in L \sharp([w]_{\rho_C} \cap L) \leq h.$$

We denote by $R_k(\Sigma, C)$ the class of trace languages in $\mathrm{Rat}(\Sigma, C)$ with ambiguity degree k.

DEFINITION 2: *The class* $R_{\mathrm{FIN}}(\Sigma, C)$ *of finitely ambiguous rational trace languages on* $F(\Sigma, C)$ *is*

$$R_{\mathrm{FIN}}(\Sigma, C) = \bigcup_k R_k(\Sigma, C).$$

A language T in $R_1(\Sigma, C)$ is said *unambiguous*. A language $T \in R_k(\Sigma, C)$ that does not belong to $R_{k-1}(\Sigma, C)$ is said *inherently ambiguous* of degree k, while a language $T \in \mathrm{Rat}(\Sigma, C)$ that is not in $R_{\mathrm{FIN}}(\Sigma, C)$ is said *inherently infinitely ambiguous.*

In [3] it has been shown that if the relation C is not transitive the following inclusions are proper:

$$R_1(\Sigma, C) \subseteq R_2(\Sigma, C) \subseteq \cdots R_{\mathrm{FIN}}(\Sigma, C) \subseteq \mathrm{Rat}(\Sigma, C).$$

An important tool in the analysis of properties related to ambiguity is the concept of generating function. Given a language $L \subseteq \Sigma^*$, the (ordinary) generating function of L is the power series $\phi_L(x) \in \mathbb{N}[[x]]$, $\phi_L(x) = \Sigma_{n \geq 0} f_L(n) x^n$, where $f_L : \mathbb{N} \mapsto \mathbb{N}$ is the counting function of L defined as $f_L(n) = \sharp\{w \in L |\, |w| = n\}$; $\phi_L(x)$ can be interpreted as a function that is analytic in a neighbourhood of the origin (see for instance [14] chapter 2). This notion is immediately extended to trace languages.

DEFINITION 3: *The generating function of a trace language* $T \subseteq F(\Sigma, C)$ *is the function*

$$\phi_T(x) = \sum_{n \geq 0} \sharp\{t \in T |\, |t| = n\} x^n.$$

Generating functions can be used in our context for two kinds of problems:

- To decide whether a given language belongs to a given class
- To state inclusion between two languages in a given class

The most famous result in the first direction is the old theorem of Schützenberger ([8]) that states that unambiguous context free languages admit algebraic generating functions. A direct application of this result let us to prove that a context free language is inherently ambiguous by showing that its generating function is not algebraic (see for instance [10]). Analogously, we can prove that a rational trace language is ambiguous by showing that its generating function in not rational. In fact, languages in $R_1(\Sigma, C)$ admit rational generating functions since it is known that if $L \subseteq \Sigma^*$ is a regular language the generating function $\phi_L(x)$ is a rational function.

EXAMPLE 3: *The language* $T = [\{\sigma_1\sigma_2, \sigma_3\}^* \cup \{\sigma_3\sigma_2, \sigma_1\}^*]_{\rho_{P_3}}$ *is obviously in* $R_2(\Sigma_3, P_3)$*; since*

$$\phi_T(x) = \frac{2}{1 - x - x^2} - \frac{1}{\sqrt{1 - 4x^3}}$$

is algebraic but not rational, T *is not unambiguous.*

In section 5 we prove that for certain commutation relations C trace languages in $R_{\mathrm{FIN}}(\Sigma, C)$ admit holonomic generating functions. Thus, the previous approach can be used for proving that certain rational trace languages are inherently infinitely ambiguous; it also let us to design a decision algorithm for the Inclusion Problem for certain classes of trace languages (see section 6).

3. FORMAL SERIES

In this section, for the sake of completeness, we recall some basic notions on formal series in noncommutative and commutative variables.

Given a commutative ring $\mathbb{K}$, a *formal series* ψ in noncommutative variables Σ is a function $\psi : \Sigma^* \mapsto \mathbb{K}$; the *support* of ψ is the language $\{w \in \Sigma^* | \psi(w) \neq 0\}$. A series ψ is called *proper* if $\psi(\varepsilon) = 0$, where ε is the empty word. We denote by $\mathbb{K}\langle\langle\Sigma\rangle\rangle$ the ring of formal series with the following operations:

- Sum: $(\phi + \psi)(w) = \phi(w) + \psi(w)$
- Product: $(\phi \cdot \psi)(w) = \Sigma_{uv=w}\phi(u)\psi(v)$

Besides the previous operations we also consider the Hadamard product and the star operation:

- Hadamard product: $(\phi \odot \psi)(w) = \phi(w)\psi(w)$
- Star (defined for proper series): $\phi^* = \sum_{n\geq 0} \phi^n$

Important subsets of $\mathbb{K}\langle\langle\Sigma\rangle\rangle$ are the ring $\mathbb{K}\langle\Sigma\rangle$ of polynomials (i.e. formal series with finite support) and the ring of rational series.

DEFINITION 4: *The ring of rational formal series is the smallest subring of* $\mathbb{K}\langle\langle\Sigma\rangle\rangle$ *containing* $\mathbb{K}\langle\Sigma\rangle$ *and closed under* $\star$.

Let $\mathbb{K}^{n\times n}$ be the monoid of $n \times n$ matrices on $\mathbb{K}$ with the usual product. A formal series $\phi \in \mathbb{K}\langle\langle\Sigma\rangle\rangle$ is called *recognizable* if there exist an integer $n \geq 1$, a morphism of monoids $\mu : \Sigma^* \mapsto \mathbb{K}^{n\times n}$ and two matrices $\eta \in \mathbb{K}^{1\times n}$, $\pi \in \mathbb{K}^{n\times 1}$, such that for all $w \in \Sigma^*$ it holds $\phi(w) = \eta\mu(w)\pi$. A result by Schützenberger [20] is the following:

PROPOSITION 1: *A series is recognizable iff it is rational.*

We recall that the class of rational formal series is closed under Hadamard product. In fact, denoting by $\otimes$ the usual Kronecker product between matrices, it holds:

PROPOSITION 2: *Let ϕ_1, ϕ_2 be two recognizable formal series represented by η_1, μ_1, π_1 and η_2, μ_2, π_2 respectively; then $\phi_1 \odot \phi_2$ is represented by $\eta_1 \otimes \eta_2$, $\mu_1 \otimes \mu_2$, $\pi_1 \otimes \pi_2$.*

We also consider formal series in commutative variables on the field $\mathbb{Q}$ of rational numbers. Let Σ^c be the free commutative monoid generated by Σ; the elements of Σ^c are monomials $\underline{\sigma}^{\underline{a}} = \sigma_1^{a_1} \cdots \sigma_n^{a_n}$ where $\underline{\sigma}^{\underline{a}} \cdot \underline{\sigma}^{\underline{b}} = \sigma_1^{a_1+b_1} \cdots \sigma_n^{a_n+b_n}$.

A formal series in commutative variables is a function $\psi : \Sigma^c \mapsto \mathbb{Q}$. The set of formal series in commutative variables Σ with coefficients in $\mathbb{Q}$ is denoted by $\mathbb{Q}[[\Sigma]]$. On $\mathbb{Q}[[\Sigma]]$ we consider the following operations:

- Sum: $(\phi + \psi)(\underline{\sigma}^{\underline{a}}) = \phi(\underline{\sigma}^{\underline{a}}) + \psi(\underline{\sigma}^{\underline{a}})$
- Cauchy product: $(\phi \cdot \psi)(\underline{\sigma}^{\underline{a}}) = \sum_{\underline{\sigma}^{\underline{b}}\underline{\sigma}^{\underline{c}}=\underline{\sigma}^{\underline{a}}} \phi(\underline{\sigma}^{\underline{b}})\psi(\underline{\sigma}^{\underline{c}})$
- Partial derivative:
$$(\partial_i\phi)(\sigma_1^{a_1} \cdots \sigma_i^{a_i} \cdots \sigma_n^{a_n}) = (a_i + 1)\phi(\sigma_1^{a_1} \cdots \sigma_i^{a_i+1} \cdots \sigma_n^{a_n})$$
- Primitive diagonal: if $1 \leq p \leq q \leq n$ then
$$(I_{pq}(\phi))(\sigma_1^{i_1} \cdots \sigma_{q-1}^{i_{q-1}}\sigma_{q+1}^{i_{q+1}} \cdots \sigma_n^{i_n}) = \phi(\sigma_1^{i_1} \cdots \sigma_{q-1}^{i_{q-1}}\sigma_q^{i_p}\sigma_{q+1}^{i_{q+1}} \cdots \sigma_n^{i_n})$$
- Substitution: $\phi(\psi_1(\tau_1, \ldots, \tau_m), \ldots, \psi_n(\tau_1, \ldots, \tau_m))$

Observe that the primitive diagonal I_{pq} maps formal series in n variables onto formal series in $n - 1$ variables; this operation is closely related to the Hadamard product $(\phi \odot \psi)(\underline{\sigma}^{\underline{a}}) = \phi(\underline{\sigma}^{\underline{a}})\psi(\underline{\sigma}^{\underline{a}})$ since

$$\phi \odot \psi = I_{1n+1} \cdots I_{n2n}\phi(\sigma_1, \ldots, \sigma_n)\psi(\sigma_{n+1}, \ldots, \sigma_{2n}),$$
$$I_{12}\phi = \left\{\phi \odot \left(\frac{1}{1 - \sigma_1\sigma_2}\frac{1}{1 - \sigma_3} \cdots \frac{1}{1 - \sigma_n}\right)\right\}(\sigma_1, 1, \sigma_3, \ldots, \sigma_n).$$

We denote by $\mathbb{Q}[\Sigma]$ the set of commutative polynomials. An interesting subclass of $\mathbb{Q}[[\Sigma]]$ is the class of rational formal series. A series ϕ is called rational if it is the power series expansion of a function P/Q, P, $Q \in \mathbb{Q}[\Sigma]$, $Q(0) = 1$; we denote by $\mathbb{Q}[[\Sigma]]_r$ the class of rational formal series.

It is known that $\mathbb{Q}[[\Sigma]]_r$ is not closed w.r.t. the Hadamard product and primitive diagonal [11]: we are interested in extensions of $\mathbb{Q}[[\Sigma]]_r$ that are closed w.r.t. these operations. An answer is given by the class of holonomic series $\mathbb{Q}[[\Sigma]]_h$.

DEFINITION 5: *A formal series $\phi \in \mathbb{Q}[[\Sigma]]$ is said to be holonomic iff there exist some polynomials*

$$p_{ij} \in \mathbb{Q}[\Sigma], \quad 1 \leq i \leq n, \quad 0 \leq j \leq d_i, \quad p_{id_i} \neq 0$$

such that

$$\sum_{j=0}^{d_i} p_{ij} \partial_i^j \phi = 0, \quad 1 \leq i \leq n.$$

We extensively use the closure properties of the class $\mathbb{Q}[[\Sigma]]_h$ that are summarized in the following proposition.

PROPOSITION 3: *The class $\mathbb{Q}[[\Sigma]]_h$ is closed under the operations of sum, Cauchy product, Hadamard product, primitive diagonal, substitution with algebraic series.*

Proof: See for instance [17].

For the sake of completeness, we recall that the class $\mathbb{Q}[[\Sigma]]_h$ properly contains the class $\mathbb{Q}[[\Sigma]]_a$ of algebraic formal series (see [19] for a definition of algebraic series). Thus, we have the following inclusions

$$\mathbb{Q}[[\Sigma]]_r \subset \mathbb{Q}[[\Sigma]]_a \subset \mathbb{Q}[[\Sigma]]_h.$$

We also make use of the following simple result about holonomic series:

THEOREM 1: *Let $\phi(x) = \sum_{n \geq 0} a_n x^n$ be a holonomic series that is not a polynomial. Then there are two integers d, $\widehat{n}$ s.t. for each integer $n \geq \widehat{n}$ there exists an integer $j < d$ s.t. $a_{n+j} \neq 0$.*

Proof: By definition, $\phi(x)$ satisfies a differential equation

$$\sum_{\substack{0 \leq i \leq a \\ 0 \leq j \leq b}} c_{ij} x^i \partial^j \phi = 0.$$

Then, for $n \geq a$ the following difference equation holds

$$\sum_{\substack{0 \leq i \leq a \\ 0 \leq j \leq b}} c_{ij}(n-i+j)\cdots(n-i+1)a_{n-i+j} = 0.$$

This implies that there exist $d \leq (a+b)$ and polynomials $q_0(n), \ldots, q_d(n)$ s.t. $q_d \neq 0$ and

$$q_d(n)a_n = \sum_{1 \leq i \leq d} q_{d-i}(n)a_{n-i}.$$

Thus, for a suitable large $\widehat{n}$, we have that $q_d(n) \neq 0$ for every $n \geq \widehat{n}$. Now, if $a_j = a_{j+1} = \cdots = a_{j+d-1} = 0$ for some $j \geq \widehat{n}$ then $a_n = 0$ for every $n \geq j$ and the series ϕ turns out to be a polynomial. ■

4. LANGUAGES WITH NONHOLONOMIC GENERATING FUNCTION

In this section we exhibit rational trace languages in $R_2(\Sigma_4, C_4)$ and $\mathrm{Rat}(\Sigma_3, P_3)$ with generating functions that are not holonomic. First of all we recall the following problem:

Problem RPC (Reduced Post Correspondence Problem)

Instance: a couple $\langle(\overline{x}, \overline{y}), S\rangle$ where $\overline{x}$, $\overline{y} \in \Sigma^*$ and $S \subset \Sigma^* \times \Sigma^*$ is a finite set, $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$.

Question: Does a sequence $i_1, \ldots, i_n$, $1 \leq i_j \leq m$, exist s.t.

$$\overline{x}x_{i_1}\cdots x_{i_n} = \overline{y}y_{i_1}\cdots y_{i_n}?$$

The first result we give is the following:

THEOREM 2: *There exist languages in $R_2(\Sigma_4, C_4)$ with nonholonomic generating function.*

Proof: Let $\Sigma_1 = \{\sharp, a, b, s\}$ be an alphabet. We consider the RPC Problem for the instance

$$\langle(\sharp ab, \varepsilon), \{(\sharp ab, \sharp), (\sharp s, \sharp a), (ab, ab), (ss, ba), (s, b\sharp), (\varepsilon, s)\}\rangle.$$

It is easy to show that there is an infinite set of solutions corresponding to the words β_k where $\beta_1 = \sharp ab\sharp ss$, $\beta_j = \sharp ab\cdots\sharp(ab)^j\sharp s^{2j}$, $j > 1$.

Now, given a new alphabet $\Sigma_2 = \{x_1, \ldots, x_7\}$ we consider two unambiguous trace languages T_1, $T_2 \in R_1(\Sigma, C)$ (with $\Sigma = \Sigma_1 \cup \Sigma_2$ and $C = \Sigma_1 \times \Sigma_2 \cup \Sigma_2 \times \Sigma_1$) defined as

$$T_1 = [x_1 \sharp ab\{x_2 \sharp ab + x_3 \sharp s + x_4 ab + x_5 ss + x_6 s + x_7\}^*]_{\rho_C}$$
$$T_2 = [x_1\{x_2 \sharp + x_3 \sharp a + x_4 ab + x_5 ba + x_6 b \sharp + x_7 s\}^*]_{\rho_C}.$$

The intersection $T_1 \cap T_2$ is the language $\{[\tau_1]_{\rho_C}, \ldots, [\tau_n]_{\rho_C}, \ldots\}$ with $\tau_n = \alpha_n \beta_n$, for a suitable $\alpha_n \in \Sigma_2^*$ with $|\alpha_n| \leq |\alpha_{n+1}|$. It is easy to observe that $|\tau_{n+1}| - |\tau_n| = \Omega(n)$ and this implies that the generating function of $T_1 \cap T_2$, $\phi_{T_1 \cap T_2}(x) = \sum_{n \geq 1} x^{|\tau_n|}$, is not holonomic (see th. 1).

At last, we consider the language $T_1 \cup T_2$ that is of ambiguity degree 2. Its generating function is

$$\phi_{T_1 \cup T_2}(x) = \phi_{T_1}(x) + \phi_{T_2}(x) - \phi_{T_1 \cap T_2}(x).$$

Since T_1, T_2 are unambiguous their generating functions $\phi_{T_1}(x)$, $\phi_{T_2}(x)$ are rational; hence $\phi_{T_1 \cup T_2}(x)$ is not holonomic.

From the language $T_1 \cup T_2$, by coding symbols of Σ_1 with a prefix code over $\{\sigma_1, \sigma_3\}$ and symbols of Σ_2 with a prefix code over $\{\sigma_2, \sigma_4\}$, we obtain a trace language in $R_2(\Sigma_4, C_4)$ having nonholonomic generating function. ■

The technique used in the proof of the above theorem is also at the basis of the following result.

THEOREM 3: *There exist languages in $Rat(\Sigma_3, P_3)$ with nonholonomic generating function.*

Proof: Let us formulate the instance of the RPC Problem defined in the previous theorem by means of two homomorphisms

$$f : \{1, \ldots, 6\}^* \mapsto \{\sharp, a, b, s\}^*,$$
$$g : \{1, \ldots, 6\}^* \mapsto \{\sharp, a, b, s\}^*,$$

defined as

$$\begin{array}{llllll} f(1) = \sharp ab & f(2) = \sharp s & f(3) = ab & f(4) = ss & f(5) = s & f(6) = \varepsilon \\ g(1) = \sharp & g(2) = \sharp a & g(3) = ab & g(4) = ba & g(5) = b\sharp & g(6) = s \end{array}$$

Then, we consider two languages $W(f)$, $W(g) \subseteq \{1, \ldots, 6, \sharp, a, b, s\}^* \times \{c\}^*$:

$$W(f) = \{(u\sharp abf(u), c^n) | u \in \{1, \ldots, 6\}^+, \quad n = |f(u)| + 3\},$$
$$W(g) = \{(ug(u), c^n) | u \in \{1, \ldots, 6\}^+, \quad n = |g(u)|\}.$$

It is immediate to observe that $A_1 = W(f) \cap W(g)$ is isomorphic to the set of solutions of the RPC Problem: by reasoning as in the proof of th. 2 we can show that the generating function $\phi_{A_1}(x)$ of A_1 is not holonomic.

Observe now that the complements of $W(f)$ and $W(g)$ are rational (see for instance [9], p. 495), hence $A_2 = \overline{W}(f) \cup \overline{W}(g)$ is rational. Since $A_1 \cup A_2 = \{1, \ldots, 6, \sharp, a, b, s\}^* \times \{c\}^*$ and $A_1 \cap A_2 = \emptyset$ we have

$$\phi_{A_1}(x) + \phi_{A_2}(x) = \frac{1}{1 - 10x} \frac{1}{1 - x}.$$

We know that $\phi_{A_1}(x)$ is not holonomic, so we conclude that $\phi_{A_2}(x)$ is not holonomic.

Once again, we can use a prefix code over $\{\sigma_1, \sigma_3\}$ to encode the symbols $1, \ldots, 6, \sharp, a, b, s$. Thus we obtain a rational language $\widetilde{A}_2 \in \mathrm{Rat}(\Sigma_3, P_3)$ whose generating function is not holonomic. ∎

5. HOLONOMIC FUNCTIONS AND $R_{\mathrm{FIN}}(\Sigma, C)$

In this section we prove that languages in $R_{\mathrm{FIN}}(\Sigma, C)$ admit holonomic generating functions if $\Sigma = A \uplus B$ and $C = (A \times \Sigma \cup \Sigma \times A) \backslash I$.

THEOREM 4: *Let us consider an alphabet $\Sigma = A \uplus B$, a commutation relation $C = (A \times \Sigma \cup \Sigma \times A) \backslash I$ and a trace language T in $R_{\mathrm{FIN}}(\Sigma, C)$. Then the generating function $\phi_T(x)$ is holonomic.*

Proof: W.l.o.g. we consider $A = \{a, b\}$ and $B = \{c, d\}$. Let T be a trace language generated by a regular language $L \subseteq \{a, b, c, d\}^*$ with ambiguity degree h and let $\langle \Sigma = \{a, b, c, d\}, Q = \{q_1, \ldots, q_n\}, \delta : Q \times \Sigma \mapsto Q, q_1, F \subseteq Q \rangle$ be a finite state automaton accepting L. The linear representation of the automaton is given by $\langle \eta, M_a, M_b, M_c, M_d, \pi \rangle$ where

$$\eta_i \begin{cases} 1 & i = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \pi_i = \begin{cases} 1 & q_i \in F \\ 0 & \text{otherwise} \end{cases}$$
$$\forall \sigma \in \Sigma \quad (M_\sigma)_{ij} = \begin{cases} 1 & \delta(q_i, \sigma) = q_j \\ 0 & \text{otherwise} \end{cases}$$

We consider the series $\xi_T \in \mathbb{Q}[[A]]_r\langle\langle B\rangle\rangle$, defined as

$$\xi_T(w) = \sum_{l,k} \lambda_{l,k,w} a^l b^k$$

where $\lambda_{l,k,w}$ is the number of words of L in the trace $[wa^l b^k]_{\rho_C}$ of T. Such series is rational and its linear representation is given by $\widehat{\eta} \in \mathbb{Q}[[A]]_r^{1\times n}$, $\widehat{\mu} : B^* \mapsto \mathbb{Q}[[A]]_r^{n\times n}$ and $\widehat{\pi} \in \mathbb{Q}[[A]]_r^{n\times 1}$ defined as

$$\widehat{\eta}(a,b) = \eta(I - aM_a - bM_b)^{-1}$$
$$\widehat{\mu}_\sigma(a,b) = M_\sigma(I - aM_a - bM_b)^{-1} \quad \sigma \in B$$
$$\widehat{\pi}(a,b) = \pi$$

Let us consider now the alphabet $A_h = \{a_1, \ldots, a_h, b_1, \ldots, b_h\}$ consisting of h isomorphic copies of the variables a, b. Then, for every integer i, $1 \le i \le h$, the series $\xi_i = \sum_{w\in B^*} w \sum_{l,k} \lambda_{l,k,w} a_i^l b_i^k$ is a rational series in $\mathbb{Q}[[A_h]]_r\langle\langle B\rangle\rangle$ represented by $\widehat{\eta}(a_i,b_i), \widehat{\mu}_\sigma(a_i,b_i), \widehat{\pi}(a_i,b_i)$.

By proposition 2, for every integer m, $1 \le m \le h$, the Hadamard product

$$\bigodot_{1\le i\le m} \xi_i = \sum_{w\in B^*} w \sum_{l_1,k_1,\ldots,l_m,k_m} \lambda_{l_1,k_1,w} \cdots \lambda_{l_m,k_m,w} a_1^{l_1} b_1^{k_1} \cdots a_m^{l_m} b_m^{k_m}$$

is a rational series in $\mathbb{Q}[[A_h]]_r\langle\langle B\rangle\rangle$ represented by

$$\widetilde{\eta}_m = \widehat{\eta}(a_1,b_1) \otimes \cdots \otimes \widehat{\eta}(a_m,b_m),$$
$$\widetilde{\mu}_{\sigma m} = \widehat{\mu}_\sigma(a_1,b_1) \otimes \cdots \otimes \widehat{\mu}_\sigma(a_m,b_m),$$
$$\widetilde{\pi}_m = \widehat{\pi}(a_1,b_1) \otimes \cdots \otimes \widehat{\pi}(a_m,b_m).$$

By mapping c, d into a new variable t, the series $\bigodot_{1\le i\le m} \xi_i$ turns out to be a series in commutative variables, $\psi_m \in \mathbb{Q}[[a_1,\ldots,a_m,b_1,\ldots,b_m,t]]_r$,

$$\psi_m = \sum_{w\in B^*} t^{|w|} \sum_{l_1,k_1,\ldots,l_m,k_m} \lambda_{l_1,k_1,w} \lambda_{l_m,k_m,w} a_1^{l_1} b_1^{k_1} \cdots a_m^{l_m} b_m^{k_m}.$$

It is easily shown that ψ_m is the rational series $\psi_m = \widetilde{\eta}_m(I - t\widetilde{\mu}_{cm} - t\widetilde{\mu}_{dm})^{-1}\widetilde{\pi}_m$.

By applying to ψ_m the operators $I_a = I_{a_1a_2} \cdots I_{a_{m-1}a_m}$, $I_b = I_{b_1b_2} \cdots I_{b_{m-1}b_m}$ and renaming a_1, b_1 with a, b we obtain the series

$$\widehat{\psi}_m = I_a I_b \psi_m = \sum_{w\in B^*} t^{|w|} \sum_{l,k} (\lambda_{l,k,w})^m a^l b^k.$$

Since ψ_m is rational, by proposition 3 $\widehat{\psi}_m$ is holonomic. Now, we construct the polynomial

$$\Lambda(x) = \sum_{1\le m\le h} \frac{\prod_{\substack{j\neq m\\ 0\le j\le h}} (x-j)}{\prod_{\substack{j\neq m\\ 0\le j\le h}} (m-j)}$$

This is a polynomial $\Lambda(x) = \sum_{1\le m\le h} c_m x^m$ of degree h; moreover, it holds that $\Lambda(0) = 0$ and $\Lambda(m) = 1$ for all m, $1 \le m \le h$.

By considering the linear combination $\sum_{1\le m\le h} c_m\widehat{\psi}_m$ we observe that

$$\begin{aligned}
\sum_{1\le m\le h} c_m\widehat{\psi}_m &= \sum_{1\le m\le h}\sum_{w\in B^*} t^{|w|}\sum_{l,k} c_m(\lambda_{l,k,w})^m a^l b^k\\
&= \sum_{w\in B^*} t^{|w|}\sum_{l,k} a^l b^k \sum_{1\le m\le h} c_m(\lambda_{l,k,w})^m\\
&= \sum_{w\in B^*} t^{|w|}\sum_{l,k} a^l b^k \Lambda(\lambda_{lkw})\\
&= \sum_{wa^lb^k\in T} t^{|w|} a^l b^k.
\end{aligned}$$

It is easily shown that the generating function of T can be obtained by applying the substitution $a, b \mapsto t$ to the holonomic function $\sum_{1\le m\le h} c_m\widehat{\psi}_m$. By proposition 3 we conclude that the generating function of T is holonomic. ■

An application of the previous theorem leads us to the following:

COROLLARY 1: $R_{\text{FIN}}(\Sigma_3, P_3) \subset Rat(\Sigma_3, P_3)$.

Proof: The rational trace language $\widetilde{A}_2$ exhibited in th. 3 has a generating function that is not holonomic, hence it is inherently infinitely ambiguous. ■

6. THE INCLUSION PROBLEM FOR $R_{\text{FIN}}(\Sigma, C)$

In this section we study the Inclusion Problem for languages in $R_{\text{FIN}}(\Sigma, C)$, formally described as follows:

Problem In(FIN,Σ, C) (Inclusion for languages in $R_{\text{FIN}}(\Sigma, C)$)

Instance: an integer k and two deterministic finite state automata M_1, M_2 accepting languages $L_1, L_2 \subseteq \Sigma^*$ s.t.

$$\forall w \in \Sigma^* \quad \sharp([w]_{\rho_C} \cap L_1) \le k, \sharp([w]_{\rho_C} \cap L_2) \le k$$

Question: Is $[L_1]_{\rho_C} \subseteq [L_2]_{\rho_C}$?

We know that for unambiguous languages in $R_1(\Sigma_4, C_4)$, the Inclusion Problem is undecidable [7]; moreover, the same negative result holds for languages in $Rat(\Sigma_3, P_3)$ [1], [13]. Here we prove that for languages in $R_{\mathrm{FIN}}(\Sigma_3, P_3)$ the Inclusion Problem turns out to be decidable.

THEOREM 5: *Let $\Sigma = A \cup B$ be an alphabet and $C = (A \times \Sigma \cup \Sigma \times A)\backslash I$ a commutation relation. Then $In(FIN, \Sigma, C)$ is decidable.*

Proof: Let (k, M_1, M_2) be an instance of $\mathrm{In}(\mathrm{FIN}, \Sigma, C)$ and let T_1, T_2 be the trace languages generated by the languages accepted by M_1, M_2 respectively. The language $T_1 \cup T_2$ has ambiguity degree at most $2k$; therefore, by th. 4, the generating functions $\phi_{T_2}(x), \phi_{T_1 \cup T_2}(x)$ are holonomic. Since $T_1 \subseteq T_2$ iff

$$\phi_{T_2}(x) = \phi_{T_1 \cup T_2}(x)$$

the problem is reduced to verify whether the holonomic function $A(x) = \phi_{T_2}(x) - \phi_{T_1 \cup T_2}(x) = \sum_{n \geq 0} a_n x^n$ is identically null.

Since the succession $\{a_n\}$ associated to $A(x)$ satisfies a linear difference equation with polynomial coefficients $\sum_{k=0}^{g} q_k(n) a_{n-k} = 0$ (see th. 1), in order to show that $A(x) = 0$ it is sufficient to show that for any j, $0 \leq j \leq g$, it holds $a_j = 0$, that is $\sharp\{t \in T_2 | \, |t| = j\} = \sharp\{t \in T_1 \cup T_2 | \, |t| = j\}$. ∎

7. COMPLEXITY REMARKS

A fundamental step in the solution of the Inclusion Problem (see th. 5) is that of finding a differential equation satisfied by the generating function of a language in $R_k(\Sigma, C)$, that is solving

Problem GFun(k,Σ, C) (generating function for languages in $R_k(\Sigma, C)$)

Instance: a deterministic finite state automaton M accepting a language $L \subseteq \Sigma^*$ s.t.

$$\forall w \in \Sigma^* \quad \sharp([w]_{\rho_C} \cap L) \leq k,$$

Answer: A linear differential equation with polynomial coefficients satisfied by the generating function $\phi_L(x)$.

We note that the solution of the previous problem gives us a partial description of the generating function $\phi_L(x)$ since a function satisfying a differential equation is univocally determined only if a suitable set of initial conditions is known. Hence, it is useful to consider the following

Problem CoeffGFun(k,Σ, C) (initial coefficients of generating functions for languages in $R_k(\Sigma, C)$)

Instance: an integer r (in unary notation) and a deterministic finite state automaton M accepting a language $L \subseteq \Sigma^*$ s.t.

$$\forall w \in \Sigma^* \quad \sharp([w]_{\rho_C} \cap L) \leq k,$$

Answer: A vector $[A_1, \ldots, A_r]$ s.t. $A_k = \sharp\{[w]_{\rho_C} | w \in L \wedge |w| = k\}$.

Here we study the complexity of GFun(k,Σ, C) and CoeffGFun(k,Σ, C) for fixed k and we show that they can be efficiently solved in parallel. In order to analyze the parallel complexity we refer to the boolean circuit model. Basic definitions associated with this model can be found for instance in [16]; here we only recall that NC^k is the class of problems solvable by log-space uniform families of boolean circuits with depth $O(\log^k n)$ and with a polynomial number of gates. With respect to this model, we have the following

THEOREM 6: *Let $\Sigma = A \uplus B$ be an alphabet and $C = (A \times \Sigma \cup \Sigma \times A) \backslash I$ a commutation relation. Then, for fixed k, GFun(k,Π, C) is in NC^2.*

Proof: W.l.o.g. we consider $A = \{a, b\}$ and $B = \{c, d\}$. Let T be the trace language generated by a regular language $L \subseteq \{a, b, c, d\}^*$ with ambiguity degree k and let $\langle \eta, M_a, M_b, M_c, M_d, \pi \rangle$, be the linear representation of the automaton accepting L.

The proof of th. 4 implicitely defines an algorithm to compute a differential equation satisfied by the generating function of T; it can be summarized as follows.

Input: the linear representation $\langle \eta, M_a, M_b, M_c, M_d, \pi \rangle$

Step 1 Compute

$$\widehat{\eta}(a, b) = \eta(I - aM_a - bM_b)^{-1}$$
$$\widehat{\mu}_\sigma(a, b) = M_\sigma(I - aM_a - bM_b)^{-1} \quad \sigma \in \{c, d\}$$
$$\widehat{\pi}(a, b) = \pi$$

Step 2 for $1 \leq m \leq k$ do

Step 2.1 Compute

$$\begin{aligned}\widetilde{\eta}_m &= \widehat{\eta}(a_1, b_1) \otimes \cdots \otimes \widehat{\eta}(a_m, b_m) \\ \widetilde{\mu}_{\sigma m} &= \widehat{\mu}_\sigma(a_1, b_1) \otimes \cdots \otimes \widehat{\mu}_\sigma(a_m, b_m) \quad \sigma \in \{c, d\} \\ \widetilde{\pi}_m &= \widehat{\pi}(a_1, b_1) \otimes \cdots \otimes \widehat{\pi}(a_m, b_m)\end{aligned}$$

Step 2.2 Compute

$$\psi_m(a_1, \ldots, a_m, b_1, \ldots, b_m, t) = \widetilde{\eta}_m(I - t\widetilde{\mu}_{cm} - t\widetilde{\mu}_{dm})^{-1}\widetilde{\pi}_m$$

Step 2.3 Construct the system of linear differential equations for the rational (hence holonomic) function ψ_m

Step 2.4 Construct the system of linear differential equations for the holonomic function $\widehat{\psi}_m = I_a I_b \psi_m$

Step 3 Compute the coefficients $c_m (1 \leq m \leq k)$ of the polynomial $\Lambda(x)$ and construct the system of linear differential equations for the holonomic function $\sum_{1 \leq m \leq k} c_m \widehat{\psi}_m$

Step 4 Compute and output the linear differential equation with polynomial coefficients for the generating function of T obtained by applying the substitution $a, b \mapsto t$ to the holonomic function $\sum_{1 \leq m \leq k} c_m \widehat{\psi}_m$

We take as size of the input the number of states n of the automaton accepting L. For each step of the algorithm we consider the most expensive operation.

In Step 1 we invert a matrix of order n with polynomial coefficients in two variables of degree 1. This is in NC^2 ([16]). In Step 2.1 we compute $O(1)$ Kronecker products of matrices with coefficients that are rational functions defined by polynomials in $O(1)$ variables and degree $O(n)$. This is in NC^1. In Step 2.2 we invert a matrix of order $n^{O(1)}$ with polynomial coefficients in $O(1)$ variables of degree $n^{O(1)}$. This is in NC^2 too. Step 2.3 requires to compute $O(1)$ differential equations associated with the rational function ψ_m (one equation for each variable). Since $\psi_m = P/Q$ for suitable polynomials P, Q, for each variable σ the associated differential equation turns out to be $\partial_\sigma \psi_m - \psi_m(\frac{\partial_\sigma P}{P} - \frac{\partial_\sigma Q}{Q}) = 0$. This is in NC^1.

Step 2.4 is reduced to a constant number of Hadamard products of holonomic functions by rational functions described by polynomials in $O(1)$ variables and degree $O(1)$ (see formula at the bottom of page 6). This is in NC^2 ([5]). Step 3 consists of a linear combination of a constant number of holonomic functions. In [22] it is shown that the sum of two holonomic functions described by systems of equations of order $\underline{\alpha}, \underline{\beta}$ is described by a system of equations of order $\underline{\alpha} + \underline{\beta}$. Such a system is obtained by an elimination technique reducible to the computation of a determinant, problem that is known to be in NC^2. An analogous elimination technique can be used for showing that the substitution in step 4 is in NC^2.

Since we have $O(1)$ steps we conclude that Gfun(k,Σ, C) is in NC^2. ■

An important result we need is stated in the following

THEOREM 7: *Let* $\Sigma = A \uplus B$ *be an alphabet and* $C = (A \times \Sigma \cup \Sigma \times A)\backslash I$ *a commutation relation. Then, for fixed k, CoeffGFun(k,ΣC) is in* NC^2.

Proof: W.l.o.g. we consider $A = \{a, b\}$ and $B = \{c, d\}$. Given a formal series ϕ, let $\{\phi\}_r$ denote the polynomial obtained by truncating ϕ at degree r, defined as

$$\{\phi\}_r(\underline{x}) = \begin{cases} 0 & \text{degree}(\underline{x}) > r \\ \phi(\underline{x}) & \text{otherwise} \end{cases}$$

We first show that it is easy to compute $\{\frac{P}{1-Q}\}_r$ for polynomials P, Q with $Q(\underline{0}) = 0$. Let N be the maximum degree of P, Q since $\frac{1}{1-Q} = \prod_{k \geq 0}(1 + Q^{2^k})$ it follows that

$$\left\{\frac{P}{1-Q}\right\}_r = \left\{P \prod_{k=1}^{\lceil \log r \rceil} (1 + \{Q^{2^k}\}_r)\right\}_r$$

Since the product of two polynomials is in NC^1, $\{\frac{P}{1-Q}\}_r$ is computed by a circuit of depth $O(\log \log r \cdot \log Nr)$ and size $Nr^{O(1)}$ (this result can be improved but for our aims it is sufficient).

Now, let us consider an instance of CoeffGFun(k,Σ, C), that is an integer r and the linear representation $\langle \eta, M_a, M_b, M_c, M_d, \pi \rangle$ of an automaton accepting a regular language L that generates a trace language with ambiguity degree k. We take as size of the input the number $n+r$, where n is the number of states of the automaton. The answer, that is the vector $[A_1, \ldots, A_r]$ where $A_i = \sharp\{[w]_{\rho_C} | w \in L \wedge |w| = i\}$ can be computed as follows.

First of all, we compute the rational series $\psi_m(a_1, \ldots, a_m, b_1, \ldots, b_m, t)$ $(1 \leq m \leq k)$ by following step 1, step 2.1 and step 2.2 of the algorithm described in the previous proof. Then we compute the polynomials $\{\psi_m\}_r$ $(1 \leq m \leq k)$ and we take the coefficients Y_{ljsm} of $t^l a_1^j \cdots a_m^j b_1^s \cdots b_m^s$ in $\{\psi_m\}_r$.

Now, we proceed by computing the coefficients c_m $(1 \leq m \leq k)$ of

$$\Lambda(x) = \sum_{1 \leq m \leq k} \frac{\prod_{\substack{j \neq m \\ 0 \leq j \leq k}} (x - j)}{\prod_{\substack{j \neq m \\ 0 \leq j \leq k}} (m - j)}$$

and we obtain the answer by observing that

$$A_i = \sum_{\substack{l+j+s=i \\ 1 \leq m \leq k}} c_m Y_{ljsm} \quad 0 \leq i \leq r.$$

The analysis performed in the previous proof and the result about the computation of the truncation of a rational series allow us to state that the computation of $\{\psi_m\}_r$ is in NC^2. Since the iterated sum of integers is in NC^1 we conclude that CoeffGFun(k, Σ, C) is in NC^2. ■

We turn now to the Inclusion problem for languages in $R_k(\Sigma, C)$ for arbitrary (but fixed) k. This problem can be formally described as follows:

Problem In(k, Σ, C) (Inclusion for languages in $R_k(\Sigma, C)$)

Instance: two deterministic finite state automata M_1, M_2 accepting languages $L_1, L_2 \subseteq \Sigma^*$ s.t.

$$\forall w \in \Sigma^* \quad \sharp([w]_{\rho_C} \cap L_1) \leq k, \ \sharp([w]_{\rho_C} \cap L_2) \leq k$$

Question: Is $[L_1]_{\rho_C} \subseteq [L_2]_{\rho_C}$?

The parallel complexity of In(k, Σ, C) is stated in the following

THEOREM 8: *Let $\Sigma = A \cup B$ be an alphabet and $C = (A \times \Sigma \cup \Sigma \times A)\backslash I$ a commutation relation. Then, for fixed k, In(k, Σ, C) is in NC^2.*

Proof: W.l.o.g. we consider $A = \{a, b\}$ and $B = \{c, d\}$. Let T_1, T_2 be the trace languages generated by two regular languages $L_1, L_2 \subseteq \{a, b, c, d\}^*$ with ambiguity degree k and let $\langle \eta_1, M_{1a}, M_{1b}, M_{1c}, M_{1d}, \pi_1 \rangle$, $\langle \eta_2, M_{2a}, M_{2b}, M_{2c}, M_{2d}, \pi_2 \rangle$ be the linear representations of the automata accepting L_1, L_2. Moreover, let $T_3 = T_1 \cup T_2$ be the trace language generated by $L_3 = L_1 \cup L_2$ with ambiguity degree $2k$ and let $\langle \eta_3, M_{3a}, M_{3b}, M_{3c}, M_{3d}, \pi_3 \rangle$ be the linear representation of the automaton accepting L_3. We take as size of the input the number $n = \max(n_1, n_2)$ where n_i $(1 \leq i \leq 2)$ is the number of states of the automaton accepting L_i. Then, the order of the matrices $M_{3\sigma}$ is $O(n^2)$.

Following the proof of th. 5, we solve problem GFun($2k, \Sigma, C$) for languages L_2 and L_3, obtaining the differential equations (of order

$n^{O(1)}$) for $\phi_{T_2}(x)$ and $\phi_{T_3}(x)$. Then, we construct the differential equation for $\phi_{T_2}(x) - \phi_{T_3}(x)$ (of order $N = n^{O(1)}$); at last, we solve CoeffGFun$(2k, \Sigma, C)$ for the instances $\langle N, \langle \eta_2, M_{2a}, M_{2b}, M_{2c}, M_{2d}, \pi_2\rangle\rangle$, $\langle N, \langle \eta_3, M_{3a}, M_{3b}, M_{3c}, M_{3d}, \pi_3\rangle\rangle$ respectively. By the results presented earlier all of these steps are in NC^2. ■

The previous result states that In(k, Σ, C) is efficiently solved in parallel for fixed k. We turn now to the analysis of In(FIN, $\Sigma.C$) and we show that the existence of a sequential algorithm that works in polynomial time w.r.t. the ambiguity degree implies the equality **P** = **NP**. To this aim we need to recall the following

Problem BPC (Bounded Post Correspondence)

Instance: a finite set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i, y_i \in \Sigma^*$, $(1 \le i \le n)$.

Question: Does a sequence $i_1, \ldots, i_p$ $(1 \le i_j \le n, p \le n)$ exist s.t.

$$x_{i_1} \cdots x_{i_p} = y_{i_1} \cdots y_{i_p}?$$

It is well known that BPC is **NP**-complete [12]. In the following theorem we show a polynomial reduction from BPC to In(FIN, Σ_3, C_3).

THEOREM 9: *If In(FIN, Σ_3, C_3) is in* **P** *then* **P** = **NP**.

Proof: Let us consider an instance of BPC specified by means of two homomorphism

$$f : \{1, \ldots, n\}^* \mapsto \{x_1, \ldots, x_n\}^*,$$
$$g : \{1, \ldots, n\}^* \mapsto \{y_1, \ldots, y_n\}^*,$$

defined as $f(i) = x_i$, $g(i) = y_i$, with $x_i, y_i \in \Sigma^*$. Let $\mathcal{N}_n$ denote the alphabet $\mathcal{N}_n = \{1, \ldots, n\}$ and let be $\widehat{\Sigma} = \Sigma \cup \mathcal{N}_n$; given a symbol $c \notin \widehat{\Sigma}$ we indicate by A_1 the product of monoids $A_1 = \widehat{\Sigma}^* \times \{c\}^*$. We consider two (finite) languages $W(f), W(g) \subseteq A_1$:

$$W(f) = \{(uf(u), c^i) | u \in \mathcal{N}_n^j,\ j \le n, i = |f(u)|\},$$
$$W(g) = \{(ug(u), c^i) | u \in \mathcal{N}_n^j,\ j \le n, i = |g(u)|\}.$$

It is immediate to observe that $W(f) \cap W(g)$ is isomorphic to the set of solutions for the BPC instance; hence there exists a solution iff $A_1 \nsubseteq \overline{W}(f) \cup \overline{W}(g)$. Let $\widetilde{\Sigma}$ be the alphabet $\widetilde{\Sigma} = \widehat{\Sigma} \cup \{c\}$ and $\widetilde{C}$ be the commutation relation $\widetilde{C} = \{c\} \times \widehat{\Sigma} \cup \widehat{\Sigma} \times \{c\}$, then A_1 is trivially in $R_1(\widetilde{\Sigma}, \widetilde{C})$. Now, we have to show that there exists a deterministic finite state automaton with $O(n^d)$ states accepting a language $\widehat{L}$ that generates $\overline{W}(f) \cup \overline{W}(g)$ with ambiguity degree $O(n)$.

We start by considering the language $\overline{W}(f)$ and we show that it is in $R_n(\widetilde{\Sigma}, \widetilde{C})$. First of all we observe that there exists a deterministic f.s. automaton with $O(1)$ states that accepts a language L_1 that generates the language

$$\{(w, c^i) | w \in \widehat{\Sigma}^*, w \notin \mathcal{N}_n^* \Sigma^*, i \geq 0\}$$

with ambiguity degree 1. Then, the language

$$\{(\alpha\beta, c^i) | \alpha \in \mathcal{N}_n^*, \beta \in \Sigma^*, (|\alpha| \leq n \wedge i \neq |f(\alpha)|) \vee |\alpha| > n\}$$

is generated with ambiguity degree 1 by a language L_2 accepted by a deterministic f.s. automaton with $O(n)$ states. At last, the language

$$\{(\alpha\beta, c^i) | \alpha \in \mathcal{N}_n^l, \beta \in \Sigma^*, l \leq n, i = |f(\alpha)| = |\beta|, \beta \neq f(\alpha)\}$$

is generated with ambiguity degree n by a language $L_3 \subset B_1^* \mathcal{N}_n B_1^* B_2^* B_3 B_2^*$, where $B_1 = \{1c^{|f(1)|}, \ldots, nc^{|f(n)|}\}$, $B_2 = \{f(1), \ldots, f(n)\}$, $B_3 = \{f(1)c^{|f(1)|}, \ldots, f(n)c^{|f(n)|}\}$. L_3 is the language accepted by a deterministic f.s. automaton with $O(n)$ states that accepts a word w iff $w = w_1 i w_2 v_1 \alpha v_2$ with $w_1 \in B_1^r$, $w_2 \in B_1^s$, $(r + s < n)$, $|v_1| = |w_1|_c$, $f(i) \neq \alpha$, $|f(i)| = |\alpha|, |v_2| = |w_2|_c$. So there exists a deterministic f.s. automaton with $O(n^2)$ states that accepts the language $L_1 \cup L_2 \cup L_3$ that generates $\overline{W}(f)$ with ambiguity degree n. Similarly, we obtain a deterministic f.s. automaton with $O(n^2)$ states that accepts a language H that generates $\overline{W}(g)$ with ambiguity degree n. Hence the language $\overline{W}(f) \cup \overline{W}(g)$ is generated with ambiguity degree at most $2n$ by a language accepted by a deterministic f.s. automaton with $O(n^4)$ states.

By coding symbols in $\widetilde{\Sigma}$ with a prefix code on $\{\sigma_1, \sigma_3\}$, if there exists an algorithm that solves $\mathrm{In}(\mathrm{FIN}, \Sigma_3, C_3)$ in polynomial time w.r.t. the ambiguity degree, we also have a polynomial time algorithm for BPC. ■

REFERENCES

1. I. J. J. Aalbersberg, H. J. Hoogeboom, *Characterizations of the decidability of some problems for regular trace languages*, Mathematical Systems Theory, 22 (1989), p. 1-19.
2. J. Berstel, C. Reutenauer, *Rational series and their languages*, Springer-Verlag, Berlin Heidelberg, 1988.
3. A. Bertoni, G. Mauri, N. Sabadini, *Equivalence and membership problems for regular trace languages*, Lectures Notes in Computer Sciences, 140 (1982), p. 61-72.

4. A. Bertoni, G. Mauri, N. Sabadini, *Unambiguous regular trace languages*, Proc. of the Coll. on Algebra, Combinatorics and Logic in Computer Science, Colloquia Mathematica Soc. J. Bolyay, North Holland, vol. 42 (1985), p. 113-123.
5. A. Bertoni, P. Massazza, *A Parallel Algorithm for the Hadamard Product of Holonomic Formal Series*, International Journal of Algebra and Computation, vol. 4, n° 4 (1994), p. 561-573.
6. A. Bertoni, P. Massazza, N. Sabadini, *Holonomic generating functions and context free languages*, International Journal of Foundations of Computer Science, vol. 3, n° 2 (1992), p. 181-191.
7. A. Bertoni, M. Goldwurm, G. Mauri, N. Sabadini, *Counting Techniques for Inclusion, Equivalence and Membership Problems*, in The Book of Traces (V. Diekert, G. Rozemberg eds.), World Scientific (1995), p. 131-163.
8. N. Chomsky, M. P. Schuetzenberger, *The algebraic theory of context-free languages*, Computer Programming and Formal Systems, North Holland, 1963, p. 118-161.
9. V. Diekert, Y. Metivier, *Partial Commutation and Traces*, in Handbook of Formal Languages (G. Rozemberg, A. Salomaa eds.), vol. 3, Beyond Words, Springer (1997), p. 457-533.
10. P. Flajolet *Analytic Models and Ambiguity of Context-Free Languages*, Theoretical Computer Science, 49 (1987), p. 283-309.
11. H. Furstenberg, *Algebraic functions over finite fields*, Journal of Algebra, 7 (1967), p. 271-277.
12. M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.
13. A. Gibbons, W. Rytter, *On the decidability of some problems about rational subsets of free partially commutative monoids*, Theoretical Computer Science, 48 (1986), p. 329-337.
14. P. Henrici, *Applied and computational complex analysis*, vol. 1, John Wiley Interscience, 1974.
15. O. H. Ibarra, *Reversal-bounded multicounter machines and their decision problems*, Journal of the Association of Computing Machinery, vol. 25 (1978), p. 116-133.
16. R. M. Karp, V. Ramachandran, *Parallel Algorithms for Shared-Memory Machines*, in Handbook of theoretical computer science, J. van Leeuwen ed., Elsevier (1990), p. 871-941.
17. L. Lipshitz, *D-finite Power Series*, Journal of Algebra, 122 (1989), p. 353-373.
18. A. Mazurkiewicz, *Concurrent program schemes and their interpretations*, DAIMI Rep. PB-78, Aarhus Univ., 1977.
19. A. Salomaa, M. Soittola, *Automata-Theoretic Aspects of formal power series*, Springer-Verlag, New York, 1978.
20. M. P. Schuetzenberger, *On the definition of a family of automata*, Information and Control, 4 (1961), p. 245-270.
21. S. Varricchio, *On the decidability of the equivalence problem for partially commutative rational power series*, LITP, Univ. Paris VII, Internal report 90.97, 1990.
22. D. Zeilberger, *A holonomic systems approach to special functions identities*, J. Comput. Appl. Math. 32 (1990), p. 321-368.