Automata for the commutative closure of regular sets

Verónica Becher Simón Lew Deveali Ignacio Mollo

April 13, 2025

Abstract

Consider A^*, the free monoid generated by the finite alphabet A with the concatenation operation. Two words have the same commutative image when one is a permutation of the symbols of the other. The commutative closure of a set $L \subseteq A^*$ is the set $\mathcal{C}(L) \subseteq A^*$ of words whose commutative image coincides with that of some word in L. We provide an algorithm that, given a regular set L, produces a finite state automaton that accepts the commutative closure $\mathcal{C}(L)$, provided that this closure set is regular. The problem of deciding whether $\mathcal{C}(L)$ is regular was solved by Ginsburg and Spanier in 1966 using the decidability of Presburger sentences, and by Gohon in 1985 via formal power series. The problem of constructing an automaton that accepts $\mathcal{C}(L)$ has already been studied in the literature. We give a simpler algorithm using an algebraic approach.

1 Primary definitions and Statement of the main result

An alphabet is a finite, non-empty set of symbols denoted by A. We call the elements of A letters, and the finite sequences of letters words. We write A^* for the set of all words over the alphabet A. A language of A^* is any set of words written with letters of A.

A finite state automaton $\mathcal{A}$ is specified by a tuple $\langle Q, A, E, I, T\rangle$ where Q is the set of states, A is the alphabet, $E \subseteq Q \times A \times Q$ is the transition set, I is the set of initial states and T is the set of final states. A finite state automaton $\mathcal{A} = \langle Q, A, E, I, T\rangle$ is deterministic if there is exactly one initial state, $|I| = 1$ and for all $p \in Q$ and for all $a \in A$ there exists at most one $q \in Q$ such that (p, a, q) is in E.

As usual, we say that a word in A^* is accepted by $\mathcal{A}$ if it is the label of a computation in $\mathcal{A}$ starting from an initial state and ending in a final state. The language accepted by the finite state automaton $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of all words accepted by $\mathcal{A}$. A regular expression over the alphabet A is a formula obtained inductively from the elements of A and the symbols in $\{\emptyset, \cup, \lambda, \cdot, *, (,)\}$ as follows: $\emptyset$, λ, and any letter in A are regular expressions; If E and F are regular expressions, then $(E \cup F)$, $(E \cdot F)$, and (E^*) are also regular expressions. We use the standard definition for the language denoted by E, see [8]. For ease of reading, we simply write E to refer to the language denoted by E.

A language is regular if there exists a finite state automaton accepting it. Equivalently, if there exists a regular expression that denotes it.

Notation. *We write $|w|_a$ to denote the number of occurrences of the letter a in the word w. For example $|010|_0 = 2$.*

The commutative image of a word is the number of occurrences of each alphabet letter in the word. It is known as the Parikh morphism. Formally, given an alphabet A of k letters, $A = \{a_1, a_2, ..., a_k\}$, the commutative image $\varphi : A^* \to \mathbb{N}^k$ of a word $w \in A^*$ is defined as

$$\varphi(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_k})$$

The commutative image of a language $L \subseteq A^*$ is $\varphi(L) \subseteq \mathbb{N}^k$, $\varphi(L) = \{\varphi(w) : w \in L\}$. The commutative closure of a language is defined as follows.

Definition 1 (Commutative Closure)**.** *Let A be an alphabet of k letters and let φ be the commutative image of the words over A. If $L \subseteq A^*$, the commutative closure of L, $\mathcal{C}(L) \subseteq A^*$,*

$$\mathcal{C}(L) = \{w \in A^* \ : \ \varphi(w) \in \varphi(L)\} = \varphi^{-1}(\varphi(L)).$$

Comment. *The commutative closure of a regular language is not always regular. For example, $\mathcal{C}((ab)^*) = \{w \in A^* \ : \ |w|_a = |w|_b\}$.*

Whether the commutative closure of a language is regular was addressed by Ginsburg and Spanier in 1966 [3] and by Gohon in 1985 [4]. The first proof is indirect and relies on the decidability of Presburger sentences. The second provides a simpler algorithm based on formal power series and a result from Eilenberg and Schützenberger [2]. We describe this algorithm in the next section.

Proposition 1 ([3, 4])**.** *It is decidable to determine whether the commutative closure of a regular language of A^* is regular.*

Once we know that the commutative closure of a regular language in A^* is regular, how can we construct the finite state automaton that recognises it? The following theorem answers this question and is our main result.

Theorem 1. *Given a regular expression of a language L in A^* whose commutative closure $\mathcal{C}(L)$ is regular, our algorithm constructs a finite state deterministic automaton for its commutative closure.*

Example 1 (Regular commutative closure)**.** *The commutative closure of the language denoted by $b(a^2 \cup b^2)^*$ is regular. The language consists of words with an even number of occurrences of the letter a and an odd number of occurrences of the letter b. Our algorithm produces the finite state automaton in Figure 1.*

The problem of constructing a finite state automaton that accepts the commutative closure of a regular language has already been studied by Hoffmann in [5]. In his work, Hoffmann provides, in the specific case of group languages, an asymptotic upper bound for the number of states of the resulting automaton, expressed in terms of the number of states of the original automaton. In Proposition 20, we present an upper bound for the general case of regular languages with a regular commutative closure. Specifically, we estimate the number of states in the constructed finite state automaton based on the length of the rational expression representing the commutative image of the given regular set. As a result, the two bounds—Hoffmann's and the one presented here—are not directly comparable in the cases he addressed.

Besides, in Proposition 21 we give an upper bound on the number of operations required for our construction, starting from the rational expression of the commutative image of the given regular set.

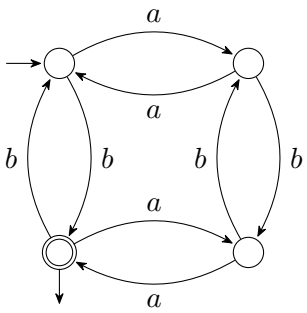


Figure 1: Finite state automaton accepting $\mathcal{C}(b(a^2 \cup b^2)^*)$.

2 Gohon's decision algorithm

Gohon's algorithm [4] decides whether the commutative closure of a regular language of A^* is regular. Instead of starting from the language, it starts directly from the commutative image of the language. That is, this decision algorithm works directly on $\mathbb{N}^k$. When dealing with the commutative closure of sets and when transforming expressions denoting sets, an algebraic approach is both appropriate and useful.

A monoid is a set equipped with an associative binary operation and a neutral element for that operation. The set A^* is the free monoid over a finite alphabet A with the concatenation as the monoid operation and the empty word λ acting as the unit element. The free commutative monoid generated by a set A is the quotient of A^* (the free monoid on A) by the congruence defined by the relations $ab = ba$ for all letters a and b in A. This commutative monoid is denoted by $A^{\oplus}$, where the monoid operation is written as $+$, and the neutral element is written as 0. The monoid $A^{\oplus}$ is isomorphic to $\mathbb{N}^k$, whose elements are the k-tuples of natural numbers $\sigma = (s_1, s_2, \ldots, s_k)$. It is also isomorphic to $a_1^* \times a_2^* \times \cdots \times a_k^*$, where $a_1, \ldots, a_k$ are the letters of A.

Regular expressions are generalised to the setting of $\mathbb{N}^k$ and referred to as rational expressions. These are defined as one would expect.

Definition 2 (Rational expression)**.** *A rational expression over $\mathbb{N}^k$ is a formula obtained inductively from elements of $\mathbb{N}^k$ and the symbols in $\{\emptyset, \cup, +, \oplus, (,)\}$ as follows: $\emptyset$ and every element of $\mathbb{N}^k$ are rational expressions; If E and F are rational expressions, then $(E \cup F)$, $(E + F)$ and $(E^{\oplus})$ are also rational expressions.*

The set of $\mathbb{N}^k$ denoted by a rational expression is defined similarly to the set of A^* denoted by a regular expression.

The set denoted by a rational expression, contained in $\mathbb{N}^k$, is defined in an analogous manner as we defined the set contained in A^* denoted by a regular expression. We simply write E to refer to the set denoted by E and use the standard precedence (first $\oplus$, then $+$ and last $\cup$).

Definition 3 (Rational set)**.** *A set S of $\mathbb{N}^k$ is rational if it is denoted by a rational expression over $\mathbb{N}^k$.*

The star height of a rational expression is defined for any monoid, we give it just for $\mathbb{N}^k$.

Definition 4 (Star height of a rational expression)**.** *The star height of a rational expression E, denoted $h(E)$, is the maximum number of stars nested in the expression E. It is defined inductively for rational expressions of $\mathbb{N}^k$:*

$$\begin{array}{ll} \text{If } E = \emptyset \text{ or } E \text{ denotes an element of } \mathbb{N}^k \text{ then} & h(E) = 0. \\ \text{If } E = F \cup G \text{ or } E = F + G \text{ then} & h(E) = \max(h(F), h(G)). \\ \text{If } E = F^{\oplus} \text{ then} & h(E) = 1 + h(F). \end{array}$$

Example 2. *The star height of $\big((0,1)^{\oplus} + (1,0)\big)^{\oplus}$ is 2.*

Definition 5 (Star height of a rational set)**.** *The star height of a rational set S of $\mathbb{N}^k$, written $h(S)$, is the minimum star height of the rational expressions of $\mathbb{N}^k$ that denote S.*

Since the monoid $\mathbb{N}^k$ is commutative, every rational set can be denoted by an expression of star height at most 1 (see [8, Exer. I.6.5]).

Proposition 2. *The star height of a rational set $S \in \mathbb{N}^k$ is at most 1.*

We therefore define expressions where we restrict the star height.

Definition 6 (Linear expression and semi-linear expression)**.** *An expression E denoting a rational set of $\mathbb{N}^k$ is linear if $E = \gamma + B^{\oplus}$ where $\gamma \in \mathbb{N}^k$ and B is a finite set of $\mathbb{N}^k$; it is semi-linear if it is the finite union of linear expressions.*

Example (Continuation of Example 2)**.** $\big((0,1)^{\oplus} + (1,0)\big)^{\oplus}$ *is equivalent to the rational expression* $(1,0)^{\oplus} \cup \Big((1,0) + \big((0,1) \cup (1,0)\big)^{\oplus}\Big)$ *which is semi-linear and has star height* 1.

Definition 7 (Non-ambiguous rational operations)**.** *In $\mathbb{N}^k$ we define the non-ambiguous rational operations, a specialization of the rational operations for which we use the same symbols:*

$S \cup T$ is non-ambiguous if $S \cap T = \emptyset$;

$S + T$ is non-ambiguous if $\forall s, s' \in S, \forall t, t' \in T$, $(s + t = s' + t')$ implies $(s = s'$ and $t = t')$;

$S^{\oplus} = \bigcup_{n \in \mathbb{N}} \underbrace{S + \cdots + S}_{n \text{ times}}$ is non-ambiguous if each of the sums and unions are non-ambiguous.

By commutativity of the monoid, if we consider $S = \{s_1, s_2, \ldots, s_l\} \subseteq \mathbb{N}^k$ then

$$S^{\oplus} = s_1^{\oplus} + s_2^{\oplus} + \cdots + s_l^{\oplus} = \{n_1 s_1 + n_2 s_2 + \cdots + n_l s_l \ : \ n_i \in \mathbb{N}\}.$$

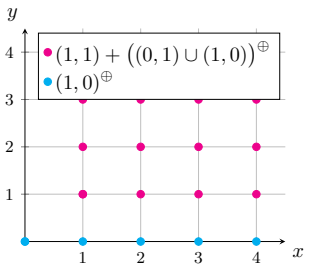
So, $S^{\oplus}$ is non-ambiguous if and only if whenever

$$n_1 s_1 + \cdots + n_l s_l = m_1 s_1 + \cdots + m_l s_l$$

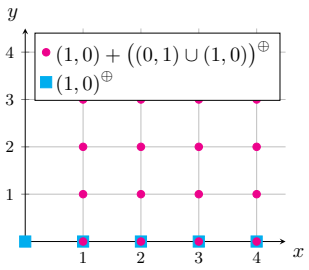
we have $n_i = m_i$ for each i.

Definition 8 (Free basis)**.** *A finite set $B \subseteq \mathbb{N}^k$ is a free basis if $B^{\oplus}$ is a non-ambiguous rational expression.*

Example 3. *$B = \{(1,0), (3,1), (1,1)\}$ is not free because $2(1,0) + (1,1) = (3,1)$. On the other hand $B' = \{(1,0), (1,1)\}$ is a free basis.*



$(1,0)^{\oplus} \cup \Big((1,1) + \big((0,1) \cup (1,0)\big)^{\oplus}\Big)$



$(1,0)^{\oplus} \cup \Big((1,0) + \big((0,1) \cup (1,0)\big)^{\oplus}\Big)$

Figure 2: Semi-simple and semi-linear expressions denoting the same set

Definition 9 (Simple expression and semi-simple expression)**.** *An expression E denoting a set of $\mathbb{N}^k$ is simple if E is linear ($E = \gamma + B^{\oplus}$) and B is a free basis; and it is semi-simple if it is the unambiguous union of simple expressions.*

Example (Continuation of Example 2)**.** *The semi-linear rational expression $(1,0)^{\oplus} \cup (1,0) + ((0,1) \cup (1,0))^{\oplus}$ is not semi-simple. Both $(1,0)^{\oplus}$ and $(1,0) + ((0,1) \cup (1,0))^{\oplus}$ are simple because both bases are free. However, their union is ambiguous, because their intersection equals $(1,0) + (1,0)^{\oplus}$. An equivalent semi-simple expression is*

$$(1,0)^{\oplus} \cup \Big((1,1) + \big((0,1) \cup (1,0)\big)^{\oplus}\Big).$$

These expressions are depicted in Figure 2.

Proposition 3 (Eilenberg and Schützenberger [2, Theorem 4])**.** *Any rational set of $\mathbb{N}^k$ can be denoted by a semi-simple expression.*

Comment. *For the effectiveness of Proposition 3, see Proposition 18 and the results of Chistikov and Hasse in [1].*

Finally, we consider formal series on k commutative variables $x_1, \ldots, x_k$. In particular, we are interested in the characteristic series of sets of $\mathbb{N}^k$.

Definition 10. *For every rational subset S of $\mathbb{N}^k$ we denote by $\underline{S}$ the characteristic series of S, defined as follows: For each element $\sigma \in \mathbb{N}^k$, if $\sigma = (s_1, \ldots, s_k) \in S$ then the coefficient of $x_1^{s_1} \cdots x_k^{s_k}$ in $\underline{S}$ is 1, and it is 0 otherwise.*

Notation. *Given a formal series T over k commutative variables $x_1, \ldots, x_k$ and an element $\sigma = (s_1, \ldots, s_k) \in \mathbb{N}^k$ we write $T[(s_1, \ldots, s_k)]$ to denote the coefficient of $x_1^{s_1} \cdots x_k^{s_k}$ in T.*

Comment. *The characteristic series $\underline{S}$ of a rational subset $S \subseteq \mathbb{N}^k$ is unique and completely characterizes S.*

Proposition 4 ([4, Proposition 3.1])**.** *Let S be a rational set of $\mathbb{N}^k$. Its characteristic series $\underline{S}$ can be computed recursively from any semi-simple expression of S, as follows:*
Let $\gamma = (c_1, c_2, \ldots, c_k)$ an element of $\mathbb{N}^k$, E and F subexpressions, and $B = \{\beta_1, \beta_2, \ldots, \beta_l\}$ a free basis. Then:

$$\begin{aligned}
\underline{\gamma} &= x_1^{c_1} x_2^{c_2} \ldots x_k^{c_k} \\
\underline{E + F} &= \underline{E}\ \underline{F} \\
\underline{E \cup F} &= \underline{E} + \underline{F} \\
\underline{B^{\oplus}} &= \frac{1}{(1 - \underline{\beta_1})(1 - \underline{\beta_2}) \cdots (1 - \underline{\beta_l})}.
\end{aligned}$$

Comment. *Note that in Proposition 4 it is essential to consider a semi-simple expression. For instance, the rule* $\underline{E \cup F} = \underline{E} + \underline{F}$ *would fail to produce coefficients smaller than* 1 *in the formal series for an ambiguous union in the expression.*

Example 4. *Let* U *the set denoted by* $(1,1) + (1,1)^{\oplus}$*, and depicted in Figure 3. Then,* $\underline{U} = \dfrac{xy}{(1-xy)}$.

Proposition 5 ([4, Proposition 3.3])**.** *Let* S *be a rational subset of* $\mathbb{N}^k$*. There exists a unique pair of polynomials* $P(S)$ *and* $Q(S) \in \mathbb{Z}[x_1, x_2, ..., x_k]$ *satisfying the following conditions:*

$\underline{S} = P(S)/Q(S)$.

$P(S)/Q(S)$ *is irreducible.*

$Q(S)[(0, \ldots, 0)] = 1$*; that is, its constant term is* 1.

Furthermore, for every pair of polynomials P *and* $Q \in \mathbb{Z}[x_1, x_2, ..., x_k]$ *such that* $\underline{S} = P/Q$*, there exists a polynomial* $R \in \mathbb{Z}[x_1, x_2, ..., x_k]$ *such that* $P = R.P(S)$ *and* $Q = R.Q(S)$.

Comment. *Proposition 5 gives a canonical representation of the rational subsets of* $\mathbb{N}^k$ *and is fundamental for proving Proposition 6, Gohon's main result in [4]. However, Proposition 5 is false for non-commutative monoids such as* $A^\star$.

As we already mentioned, not every regular language $L \subseteq A^*$ has a regular commutative closure $\mathcal{C}(L)$, as a subset of A^*. However, the commutative image of every regular set of A^* is always a rational set of $\mathbb{N}^k$.

Comment. *The commutative image* $\varphi(L)$ *of any regular language* $L \subseteq A^*$ *is a rational set of* $\mathbb{N}^k$.

On the other hand, the fact that a set S is rational in $\mathbb{N}^k$ says nothing about whether $\varphi^{-1}(S)$ is a regular set of A^* or not. A well studied subclass of the rational sets in finitely generated monoids is the class of the recognizable sets [8]. Our interest in these sets comes from the fact that, for any surjective morphism, the inverse image of a recognizable set of $\mathbb{N}^k$ is a regular set of A^*.

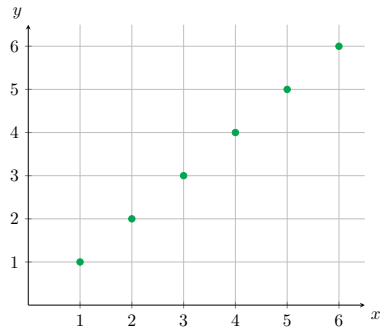
Definition 11 (Recognizable set)**.** *Let* A *be a finite alphabet and let* $\varphi : A^* \to \mathbb{N}^k$ *be the commutative image. A subset* S *of* $\mathbb{N}^k$ *is recognizable if* $\varphi^{-1}(S)$ *is regular.*

Note that if S is recognizable in $\mathbb{N}^k$ then $\varphi(\varphi^{-1}S) = S$ is rational and thus $\mathrm{Rec}(\mathbb{N}^k) \subseteq \mathrm{Rat}(\mathbb{N}^k)$. The following is an equivalent characterization of recognizable sets that is attributed to Mezei.

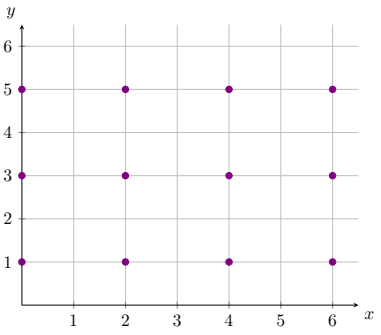
Definition 12 (Mezei [8, Corollary II.2.20])**.** *A set* S *of* $\mathbb{N}^k$ *is recognizable if there exists a family of sets* $\{T_{i,j}\}_{i \in I, 1 \leq j \leq k}$ *with* I *finite and each* $T_{i,j}$ *rational subsets of* $\mathbb{N}$ *such that*

$$S = \bigcup_{i \in I} T_{i,1} \times \cdots \times T_{i,k}.$$

Comment. *The characterization of recognizable sets as in Definition 12 is not unique. For instance, when* S *is* $\mathbb{N}$*, we already find multiple characterizations:* $1^{\oplus}$*;* $1 \cup (1 + 1^{\oplus})$*;* $1 \cup \cdots \cup n \cup \big((n+1) + 1^{\oplus}\big)$*;* $2^{\oplus} \cup (1 + 2^{\oplus})$*;* $\ldots$



$U = (1,1) + (1,1)^{\oplus}$



$V = (0,1) + \big((2,0) \cup (0,2)\big)^{\oplus}$

Figure 3: On the left, an infinite set of $\mathbb{N}^2$ that is not recognizable, because it is impossible to characterize it as a finite union of products of rational sets in $\mathbb{N}$. On the right, an infinite set of $\mathbb{N}^2$ that is recognizable: it can be described as the product between the even and the odd numbers, both rational sets in $\mathbb{N}$.

The next result is the main theorem obtained by Gohon in [4].

Proposition 6 (Gohon [4, Theorem 4.6])**.** *For every rational set S of $\mathbb{N}^k$, we can associate a fraction $P/Q = \underline{S}$, where P and Q are polynomials of k commutative variables $x_1, x_2, ..., x_k$ with coefficients in $\mathbb{Z}$, such that S is recognizable if and only if $Q = 1$, or Q is a product of polynomials of the form $(1 - x_j^{p_j})$.*

Gohon's algorithm starts from a semi-simple expression and obtains its characteristic series. It reduces the polynomials until all factors of more than one variable in the denominator are simplified. This is possible exactly when the set is recognizable.

Example (Continuation of Example 4)**.** $\underline{U} = \dfrac{xy}{(1-xy)}$. *The set U, shown in Figure 3, is not recognizable because $\underline{U}$ is irreducible and its denominator features a factor with more than one variable. In fact, if we consider the alphabet $A = \{a, b\}$ the set U corresponds to $\mathcal{C}((ab)^+) = \{w \in A^+ \ : \ |w|_a = |w|_b\} = \mathcal{C}((ab)^*) - \{\lambda\}$, which is not regular.*

Example 5. $V = (0,1) + \big((2,0) \cup (0,2)\big)^{\oplus}$, $\underline{V} = \dfrac{y}{(1-x^2)(1-y^2)}$. *The set V, shown in Figure 3, is recognizable because $\underline{V}$ has no factors of more than one variable in the denominator. Notice that V is the commutative image of $L = b(a^2 \cup b^2)^*$ in Example 1 and we have already given an automaton that accepts $\mathcal{C}(L)$; thus, V is recognizable.*

3 Resimple expressions

We introduce a new kind of rational expression, that we call resimple expressions, which are relatively simple and denote recognizable sets of $\mathbb{N}^k$. Our goal is to construct these resimple expressions from the characteristic series of recognizable sets.

Notation. *From now on, let $\varphi : A^\star \to \mathbb{N}^k$ denote the commutative image morphism. We write $\mathbf{e}_i$ to denote the element of $\mathbb{N}^k$ whose components are all 0 except for the i-th, which is 1. The set $\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$ is the set of generators of $\mathbb{N}^k$.*

Definition 13 (Primary element)**.** *An element $\sigma \in \mathbb{N}^k$ is primary if $\sigma = n\mathbf{e}_j$ for some $n \in \mathbb{N}_{>0}$, and some j between 1 and k. When we want to make explicit the index j we say that σ is j-primary.*

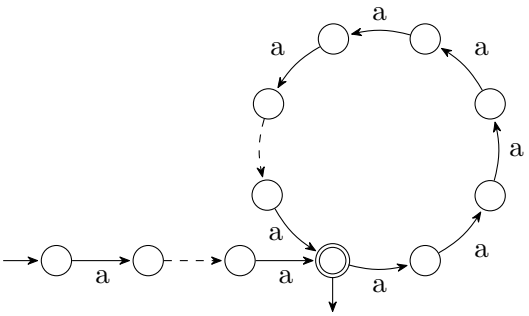


Figure 4: Finite state automaton for $\varphi^{-1}(R)$ when R is an atomic resimple expression of $\mathbb{N}$ ($k = 1$ and $A = \{a\}$).

Definition 14 (Primary basis)**.** *A free basis $B \subseteq \mathbb{N}^k$ is primary if all its elements are primary.*

We use the name atomic resimple for simple expressions that denote recognizable sets.

Definition 15 (Atomic resimple expression)**.** *An expression R of a recognizable set of $\mathbb{N}^k$ is atomic resimple if R is simple of the form $R = \gamma + B^{\oplus}$ where B is a free primary basis.*

Proposition 7. *If $R = (c_1, \ldots, c_k) + B^{\oplus}$ is an atomic resimple expression then it denotes a recognizable set.*

Proof. B is a free primary base, so all its elements are primary. Also, because it is free, there is at most one j-primary element in B for each j. Let us consider S_j a subset of $\mathbb{N}$

$$S_j = \begin{cases} c_j + p_j^{\oplus} & \text{if } p_j \mathbf{e}_j \in B \\ c_j & \text{otherwise.} \end{cases}$$

Then the set denoted by R is exactly $S_1 \times \cdots \times S_k$, a recognizable set. □

An atomic resimple expression is a rational expression of $\mathbb{N}^k$. We just write R for the set denoted by an atomic resimple expression R. Starting from an atomic resimple expression $R = c + p^{\oplus}$, where $c, p \in \mathbb{N}$, we can construct a finite state deterministic automaton $\mathcal{A} = \langle Q, \{a\}, E, I, T\rangle$ that accepts the language $\varphi^{-1}(R) \subseteq A^*$, where φ is the commutative image. The automaton is depicted in Figure 4. We must do the same for any k. To accomplish this we must first return to A^* to introduce an operation between regular languages.

Definition 16 (Shuffle)**.** *The shuffle of two words w and v in A^*, denoted with $w \between v$, is the subset of A^* defined by*

$$w \between v = \{w_1 v_1 \cdots w_n v_n \ : \ w = w_1 \cdots w_n, v = v_1 \cdots v_n, \text{with } w_i, v_j \in A^* \text{ for each } i, j \text{ and } n \in \mathbb{N}\}$$

The shuffle of two words is additively extended to the shuffle of two languages in a natural way. If $L, K \subseteq A^\star$ are two languages, their shuffle is defined as follows:

$$L \between K = \bigcup_{w \in L, v \in K} w \between v.$$

Example 6. $ab \between ba = \{abba, baba, baab, abab\}$.

The shuffle of languages is an associative operation on $\mathcal{P}(A^\star)$. Therefore, if $L_1, \ldots, L_l$ are languages, we denote

$$L_1 \between \cdots \between L_l = \bigcup_{w_i \in L_i} w_1 \between \cdots \between w_l.$$

We introduce the shuffle product automaton.

Definition 17 (Shuffle product automaton)**.** *Let $\mathcal{A}' = \langle Q', A, E', I', T' \rangle$ and $\mathcal{A}'' = \langle Q'', A, E'', I'', T'' \rangle$ be deterministic finite state automata. We define the shuffle product automaton $\mathcal{A} = \mathcal{A}' \between \mathcal{A}''$ as $\mathcal{A} = \langle Q' \times Q'', A, E, I' \times I'', T' \times T'' \rangle$, where*

$$\begin{aligned} E = &\{((p', p''), a, (q', p'')) : p'' \in Q'' \text{ and } ((p', a, q') \in E'\} \cup \\ &\{((p', p''), a, (p', q'')) : p' \in Q' \text{ and } ((p'', a, q'') \in E''\}. \end{aligned}$$

Inductively, given finite state deterministic automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$, we define an automaton $\mathcal{A} = \mathcal{A}_1 \between \cdots \between \mathcal{A}_n$ by repeating this procedure.

It is easy to see that $L(\mathcal{A}' \between \mathcal{A}'') = L(\mathcal{A}') \between L(\mathcal{A}'')$. And, furthermore, that $L(\mathcal{A}_1 \between \cdots \between \mathcal{A}_n) = L(\mathcal{A}_1) \between \cdots \between L(\mathcal{A}_n)$.

Proposition 8. *The shuffle of two regular languages in A^* is a regular language.*

Proposition 9. *Let A' and A'' be disjoint alphabets, and let $A = A' \cup A''$ be the union of both. Suppose we have two finite state deterministic automata $\mathcal{A}'$ and $\mathcal{A}''$ defined on alphabets A' and A'' respectively. If we consider both as automata on A, then the shuffle between them $\mathcal{A}' \between \mathcal{A}''$ is also a deterministic finite state automaton.*

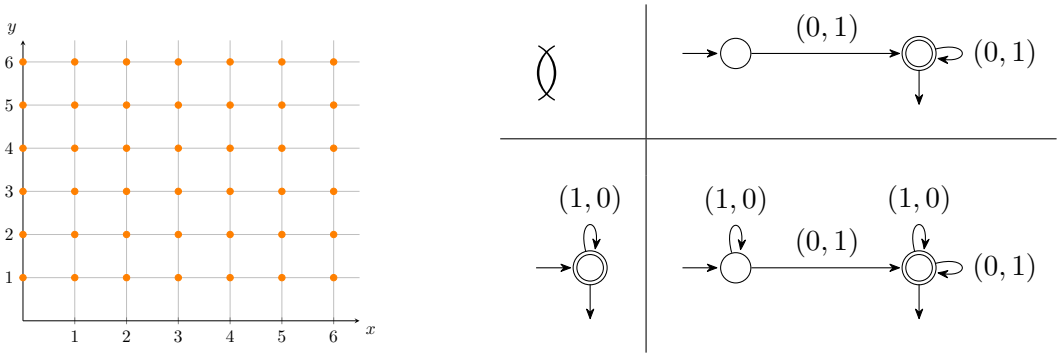
Proof. Assume, forcontradiction, that $\mathcal{A}' \between \mathcal{A}''$ is not deterministic. Since $\mathcal{A}'$ and $\mathcal{A}''$ are deterministic, both have only one initial state, so $\mathcal{A}' \between \mathcal{A}''$ has only one initial state too. Then there must be a state (p', p'') and a letter $a \in A$ such that there's more than one transition labelled a leaving (p', p''). Since the alphabets are disjoint there are two exclusive possibilities: either $a \in A'$ or $a \in A''$. Without loss of generality we assume $a \in A'$. As there are no transitions in E'' labelled with a, both must be in E'. So there is more than one transition in E' with origin p' and label a. This is impossible because $\mathcal{A}'$ is deterministic. □

Notation. *In the sequel we write automaton as an abbreviation of finite state deterministic automaton.*

Now we build the automaton over A^* for any atomic resimple expression, as we did for $k = 1$.

Proposition 10. *Let $R = (c_1, \ldots, c_k) + B^\oplus$ be an atomic resimple expression denoting a set $S \subseteq \mathbb{N}^k$. We can build an automaton $\mathcal{A}$ that accepts the language $\varphi^{-1}(S) \subseteq A^*$, where φ is the commutative image.*

Proof. By Proposition 7 we know that S can be written in the form $S_1 \times \cdots \times S_k$, where each S_j is denoted by a resimple expression $S_j = c_j + p_j^\oplus$. For each coordinate j we can obtain a complete automaton $\mathcal{A}_j$ labelled over the alphabet $\{a_j\}$ that accepts $\varphi^{-1}(S_j)$, see Figure 4. Consider the automaton $\mathcal{A}_j$ as labelled in the larger alphabet $A = \{a_1, \ldots, a_k\}$. Then the shuffle product $\mathcal{A} = \mathcal{A}_1 \between \cdots \between \mathcal{A}_k$ accepts the language $\varphi^{-1}(S)$. Observe that $w \in L(\mathcal{A})$, if and only if, for every coordinate j we have $|w|_{a_j} \in S_j$, which is equivalent to $\varphi(w) \in S$. □



(a) $(0,1) + \big((1,0) \cup (0,1)\big)^{\oplus}$ (b) Construction of the automaton.

Figure 5: Original set and the automata construction of Proposition 10.

Comment. *The automata in the examples accept languages in A^*, where $A = \{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$ is the set of the generators of $\mathbb{N}^k$ considered as an alphabet. For clarity, all examples are for the case $\mathbb{N}^2$. From now on, the automaton for a resimple expression R will denote the automaton that accepts $\varphi^{-1}(R)$.*

Example 7. *The expression $R = (0,1) + \big((1,0) \cup (0,1)\big)^{\oplus}$ is resimple, see Figure 5a. The automaton for this expression is the shuffle product between the automata for $R_x = 1^{\oplus}$ and the automata for $R_y = 1 + 1^{\oplus}$, see Figure 5b.*

It is possible to calculate the exact size of the automaton built in Proposition 10.

Proposition 11. *Let $R = (c_1, \ldots, c_k) + B^{\oplus}$ be an atomic resimple expression. For each coordinate j, let $p_j \in \mathbb{N}$ be*

$$p_j = \begin{cases} n & \text{if } n\mathbf{e}_j \in B \\ 2 & \text{otherwise.} \end{cases}$$

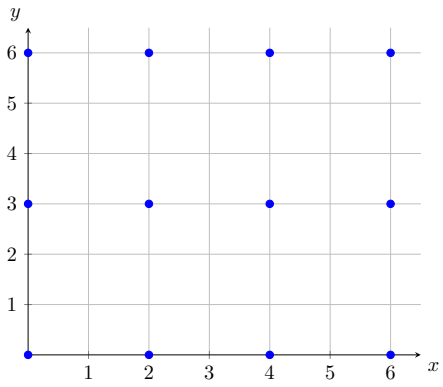
Then, the complete automaton defined in Proposition 10 accepts $\varphi^{-1}(R)$ and it has exactly $\prod_{j=1}^{k} c_j + p_j$ states.

Comment. *If there is no j-primary element in B then the resimple expression for that component, $R_j = \gamma_j$, denotes a finite set. Therefore, the automaton for R_j would have $\gamma_j + 1$ states. However, to make it complete it is necessary to add a sink state, having $\gamma_j + 2$ states in total. This explains $p_j = 2$.*

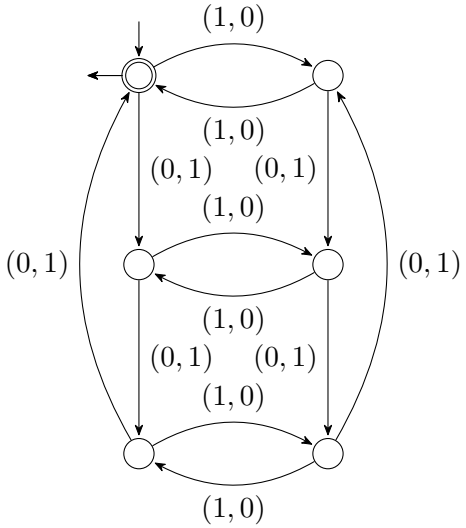
Example 8. *An atomic resimple expression and its automaton appear in Figure 6.*

Yet another instance of an atomic resimple expression can be found in Example 1.

Definition 18 (Resimple expression of a set of $\mathbb{N}^k$)**.** *A resimple expression of a recognizable set of $\mathbb{N}^k$ is a formula that is obtained inductively from atomic resimple expressions and the boolean operations: union, intersection and complement ($\cup, \cap$ and ^{c}): An atomic resimple expression is resimple. If E, F are resimple expressions then $E \cup F$, $E \cap F$ and E^c are resimple expressions.*



(a) $\left((2,0) \cup (0,3)\right)^{\oplus}$



(b) Automaton for $\varphi^{-1}\Big(\left((2,0) \cup (0,3)\right)^{\oplus}\Big)$

Figure 6: Original set and the resulting automaton.

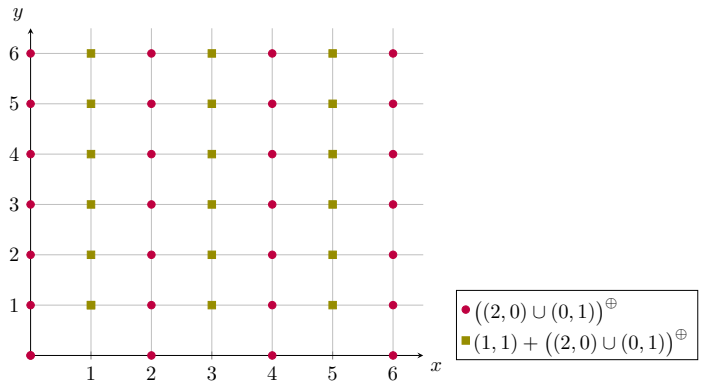


Figure 7: $\Big((0,1) + \left((1,0) \cup (0,1)\right)^{\oplus}\Big) \cup \Big(\left((2,0) \cup (0,3)\right)^{\oplus}\Big)$

Example 9. $(0,1) + ((1,0) \cup (0,1))^{\oplus} \cup ((2,0) \cup (0,3))^{\oplus}$ *is resimple, see Figure 7.*

Proposition 12. *Let S be a set of $\mathbb{N}^k$. If S has a resimple expression that denotes it then S a recognizable set in $\mathbb{N}^k$).*

Proof. The family of recognizable sets of $\mathbb{N}^k$ forms a Boolean algebra (see [8]) and by Proposition 7 we know that atomic resimple expressions denote recognizable sets. □

Definition 19 (Consistent resimple expression)**.** *An expression is resimple consistent if it is resimple and all the bases of its atoms are the same.*

Example (Continuation of Example 9)**.** *The resimple expression* $\Big((0,1) + \left((1,0) \cup (0,1)\right)^{\oplus}\Big) \cup \left((2,0) \cup (0,3)\right)^{\oplus}$*, is not consistent because* $\{(1,0),(0,1)\} \neq \{(2,0),(0,3)\}$*. However,* $\left((2,0) \cup (0,1)\right)^{\oplus} \cup \Big((1,1) + \left((2,0) \cup (0,1)\right)^{\oplus}\Big)$ *is a consistent resimple expression that denotes the same set.*

Theorem 2. *For any resimple expression denoting a recognizable set S in $\mathbb{N}^k$ an automaton can be effectively constructed to recognize the language $\varphi^{-1}(S) \subseteq A^*$, where φ is the commutative image.*

Proof. This is an immediate consequence of the Proposition 10 and the fact that regular languages in A^* form an effective boolean algebra. □

Propositions 13 and 14 below are standard and can be found in any automata theory book (see for instance [8]). We give the proof of Proposition 13 in detail to introduce the intersection automaton.

Proposition 13. *The intersection of two regular languages in A^* is regular.*

Proof. Let $\mathcal{A}' = \langle Q', A, E', I', T' \rangle$ and $\mathcal{A}'' = \langle Q'', A, E'', I'', T'' \rangle$. We define $\mathcal{A} = \mathcal{A}' \cap \mathcal{A}''$ as $\mathcal{A} = \langle Q' \times Q'', A, E, I' \times I'', T' \times T'' \rangle$, with

$$E = \{((p', p''), a, (q', q'')) : (p', a, q') \in E' \text{ and } (p'', a, q'') \in E''\}.$$

It is easy to see that $L(\mathcal{A}) = L(\mathcal{A}') \cap L(\mathcal{A}'')$. □

The automaton $\mathcal{A}' \cap \mathcal{A}''$ that we used in the previous proof is called the *intersection automaton* between the automata $\mathcal{A}'$ and $\mathcal{A}''$ and it is a general construction that allows us to obtain an automaton to recognise the intersection of two languages from two automata that recognise each of those languages.

Proposition 14. *If $\mathcal{A}'$ and $\mathcal{A}''$ are deterministic automata, then the intersection automaton $\mathcal{A} = \mathcal{A}' \cap \mathcal{A}''$ is also deterministic.*

We are interested in seeing that the intersection automaton does not grow significantly.

Proposition 15. *If $\mathcal{A}'$ and $\mathcal{A}''$ are defined from resimple expressions that are consistent with each other, then the number of states in the intersection automaton $\mathcal{A} = \mathcal{A}' \cap \mathcal{A}''$ for one coordinate is at most the maximum between the number of states in $\mathcal{A}'$ and $\mathcal{A}''$ for that coordinate.*

Proof. Since in each coordinate the two automata end up with the same period, there is a mapping between the states of the two automata. In each coordinate, the intersection of the two automata is just the bigger one, see Figure 8, possibly with different final states. □

Comment. *If in some coordinate the automaton were complete then the result of the intersection would be complete in that specific coordinate. To make one coordinate complete it is enough to add a single sink state.*

Example (Continuation of Example 9)**.** *Construction of the automaton for $\big((2,0) \cup (0,1)\big)^{\oplus} \cup (1,1) + \big((2,0) \cup (0,1)\big)^{\oplus}$, see Figure 9.*

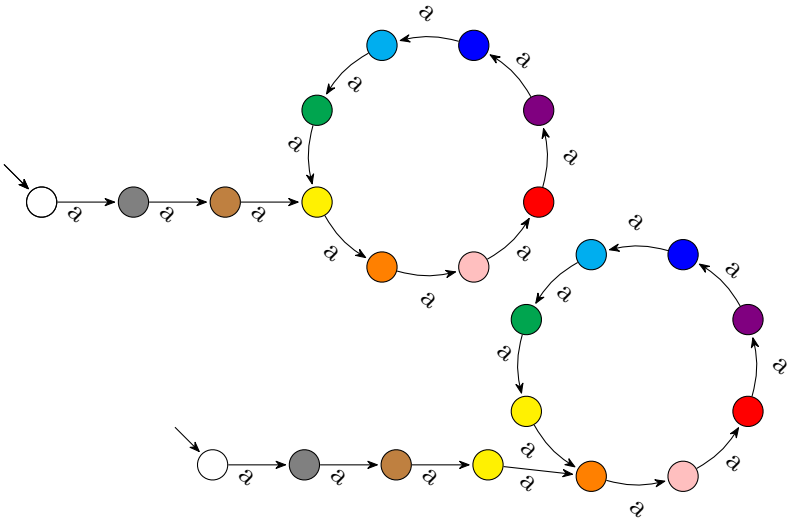


Figure 8: The automata for two resimple consistent expressions with $k = 1$. The intersection automaton is equal to the bigger automaton (right) since all the states of the smaller automaton (left) are paired up with the states of the bigger one (right), color by color.

4 From the polynomials to the resimple expression

We provide an effective method for obtaining a resimple expression that denotes the given recognizable set, starting from the polynomials of its characteristic series.

Recall that Gohon's algorithm converts the semi-simple expression into a characteristic series that expresses as a fraction of polynomials P' and Q'. Then it reduces this fraction by simplifying all the factors of more than one variable in Q', which can be done only if the denoted set is recognizable. The obtained denominator Q is a product of polynomials of the form $(1 - x_j^{p_j})$ with $p_j \in \mathbb{N}_{>0}$, or 1. Also, it can be guaranteed that for every coordinate j of $\{1, \ldots, k\}$ there is at most one factor of the form $(1 - x_j^{p_j})$.

Lemma 16 (Gohon [4, Lemma 4.3])**.** *Let S be a rational set of $\mathbb{N}^k$. There exists a semi-simple expression denoting S and for each coordinate j the following two conditions hold:*

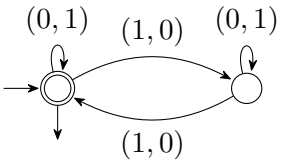
- *There is at most one j-primary element in each basis of the expression.*
- *If two bases in the expression contain a j-primary element, then it is the same in both.*

Furthermore, Gohon's proof gives an effective procedure for computing such a semi-simple expression. In the resulting expression, the non null coordinate of a j-primary element is the least common multiple of all the j-th coordinates of the j-primary elements in the original expression.

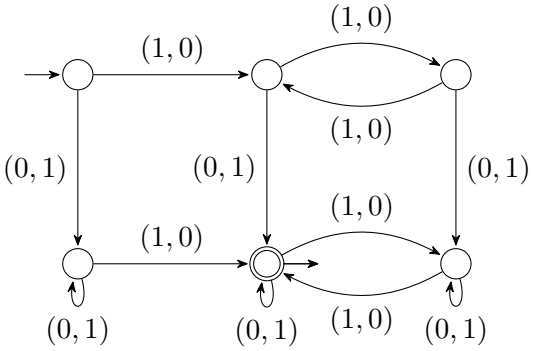
Lemma 17. *Let S a recognizable set of $\mathbb{N}^k$). Then, we can compute from a semi-simple expression of S two polynomials P and $Q \in \mathbb{Z}[x_1, x_2, ..., x_k]$ such that:*

$$\underline{S} = P/Q,$$

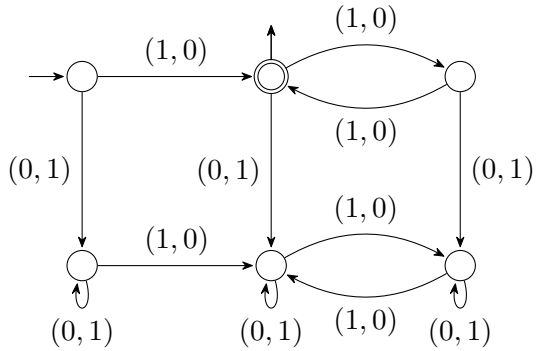
$$Q = 1 \text{ or } \left(\exists J \subseteq \{1, 2, \ldots, k\}, Q = \prod_{j \in J} (1 - x_j^{p_j}), p_j \in \mathbb{N} \setminus \{0\}\right).$$



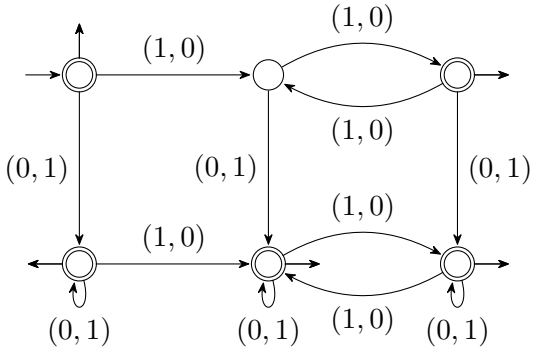
(a) Automaton for $\big((2,0) \cup (0,1)\big)^{\oplus}$.



(b) Automaton for $(1,1) + \big((2,0) \cup (0,1)\big)^{\oplus}$.



(c) Automaton for
$\Big(\big((2,0) \cup (0,1)\big)^{\oplus}\Big)^{c} \cap \Big((1,1) + \big((2,0) \cup (0.1)\big)^{\oplus}\Big)^{c}$



(d) Final automaton, complement of Figure 9c

Figure 9: Construction of the automaton for $\big((2,0) \cup (0,1)\big)^{\oplus} \cup \Big((1,1) + \big((2,0) \cup (0,1)\big)^{\oplus}\Big)$.

Proof. Let E be a semi-simple expression of S. By Lemma 16 we can compute an equivalent, E', such that for every $j \in \{1, 2, \ldots, k\}$ there exists at most one j-primary generator amongst all bases in E'. From E' we compute $\underline{S}$ using Proposition 4. We obtain two polynomials P' and Q' such that $\underline{S} = P'/Q'$ and Q' is a product of polynomials of the form $(1 - x_1^{b_1} \cdots x_k^{b_k})$. For each $j \in \{1, \ldots, k\}$ there exists at most one polynomial of the form $(1 - x_j^{p_j})$ and by Proposition 6 we can deduce that P' is divisible by the polynomials of more than one variable that constitute Q'. After this simplification we obtain P and Q. □

Now we show that for a recognizable set S of $\mathbb{N}^k$ we can give a consistent resimple expression for S

Theorem 3. *Let S be a recognizable set of $\mathbb{N}^k$. Then there exists a consistent resimple expression that denotes S and it can be effectively computed from any semi-simple rational expression that denotes S.*

Proof. From a semi-simple expression for S we obtain two polynomials P and Q as in Lemma 17.

If $Q = 1$ then S is finite and we can calculate a resimple expression that denotes it by directly applying Proposition 4. In this case, the characteristic series for S coincides with the polynomial P and thus can be written as

$$P = \sum_{1 \leq h \leq m} \underline{d_h}.$$

with each $\underline{d_h}$ being a monomial of the form $x_1^{\delta_1} x_2^{\delta_2} \cdots x_k^{\delta_k}$. By applying Proposition 4 inversely, we can transform each $\underline{d_h}$ into a matching expression $d_h \in \mathbb{N}^k$, and we make the union of them. The resimple expression in this case is:

$$\bigcup_{1 \leq h \leq m} d_h.$$

Now consider the case $Q \neq 1$. So by Lemma 17 there exists a set $J \subseteq \{1, 2, \ldots, k\}$ such that $Q = \prod_{j \in J} (1 - x_j^{p_j})$, and without loss of generality, we can assume that $J = \{1, 2, \ldots, k'\}$ with $1 \leq k' \leq k$. In this case P can be written as

$$P = \sum_{1 \leq h \leq m} \mu_h \underline{d_h}$$

where $\mu_h \in \mathbb{Z} - \{0\}$ and $d_h \in \mathbb{N}^k$, for each $h = 1, ..., m$. Note that unlike the finite case there may be negative terms here. For each coordinate $j \in J$, we define the primary element $b_j = p_j \ \mathbf{e}_j$ where p_j coincides with the corresponding exponent in Q. Furthermore, for each $h = 1, ..., m$, , we denote $S_h = d_h + b_1^{\oplus} + b_2^{\oplus} + \cdots + b_{k'}^{\oplus}$. From these definitions we have

$$\underline{S_h} = \underline{d_h}/Q.$$

Then,

$$\underline{S} = \sum_{1 \leq h \leq m} \mu_h \underline{S_h}.$$

Consider the equivalence relation $\sim$ over $\mathbb{N}^k$ defined as

$$s \sim s' \iff \left(\text{ for every } h = 1, .., m, \quad s \in S_h \iff s' \in S_h \right).$$

Note that each S_h is an atomic resimple expression, thus each equivalence class is defined by a resimple expression and, by definition of the series, Definition 10, we also have

$$s \sim s' \iff \left(\text{ for every } h = 1, .., m, \quad \underline{S_h}[s] = \underline{S_h}[s'] \right).$$

Then, $s \sim s'$ exactly when $\underline{S}[s] = \underline{S}[s']$. Notice that the number of equivalence classes is 2^m because for every $h = 1, .., m$, we have $\underline{S_h}[s] \in \{0, 1\}$. To refer to an equivalence class of $\sim$ we write

$$\widetilde{s} = \{s' \in \mathbb{N}^k : s \sim s'\}.$$

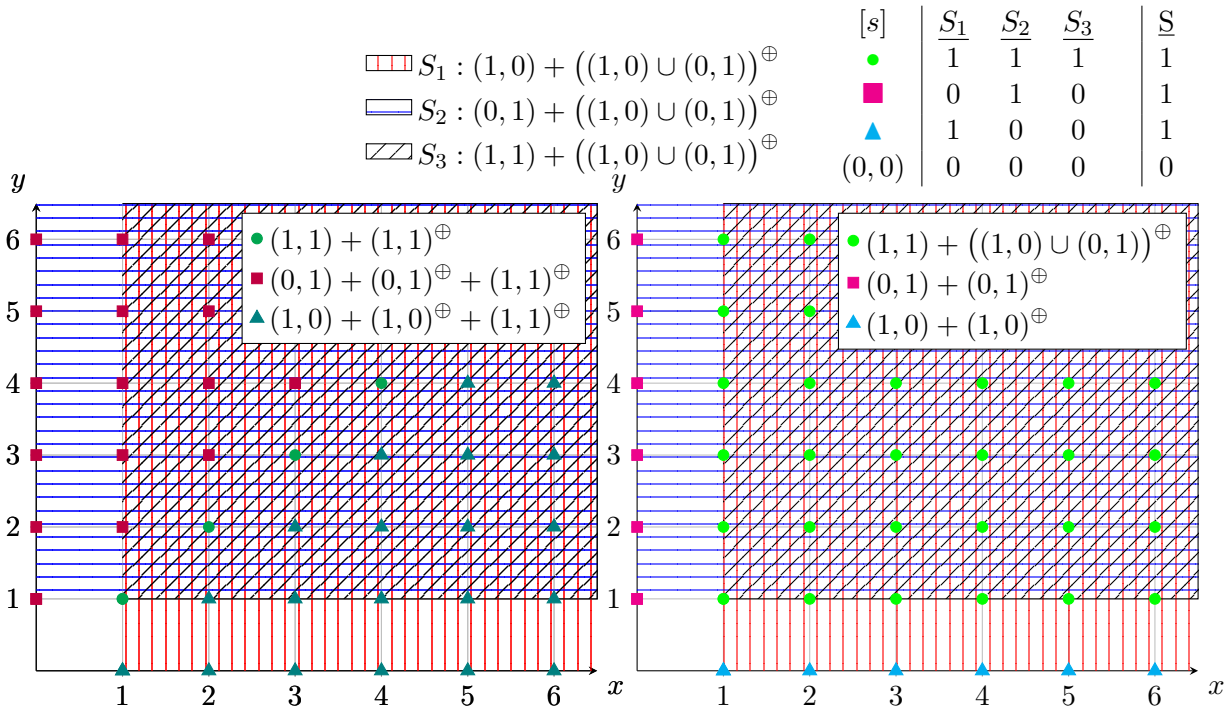
Each equivalence class $\widetilde{s}$ is denoted by the following resimple expression

$$C_{\widetilde{s}} = T_1 \cap T_2 \cap \cdots \cap T_m, \quad \text{where for } h = 1, .., m, \quad T_h = \begin{cases} S_h, & \text{if } \underline{S_h}[s] = 1 \\ (S_h)^c, & \text{if } \underline{S_h}[s] = 0. \end{cases}$$

We are only interested in the equivalence classes $\widetilde{s}$ such that $s \in S$. Hence, we must take the union of the resimple expressions $C_{\widetilde{s}}$ just for those $\widetilde{s}$ such that $\underline{S}[s] = 1$. Thus, the resimple expression for S is

$$\bigcup_{\text{good } \widetilde{s}} C_{\widetilde{s}}$$

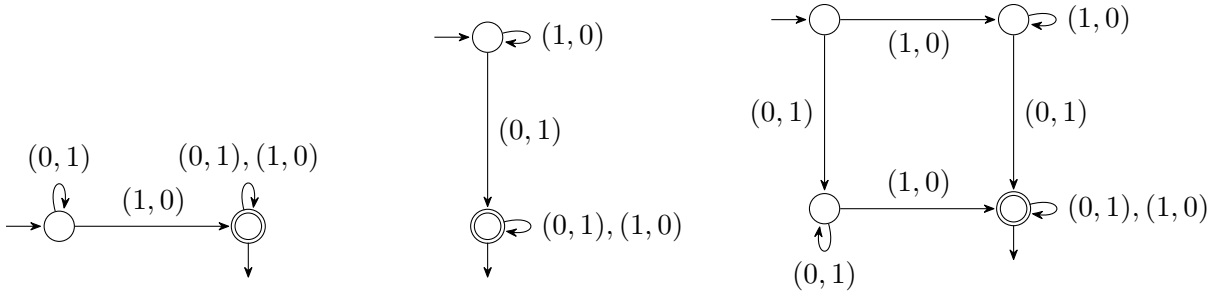
where $\widetilde{s}$ is good if $\underline{S}[s] = 1$; equivalently, $\widetilde{s}$ is good if $\sum_{h=1}^{m} \mu_h \underline{S_h}[s] = 1$. Furthermore, all the atomic resimple expressions S_h have the same basis $B = \{b_1, b_2, \ldots, b_{k'}\}$, so the given resimple expression is consistent. □



(a) Original representation of S.

(b) Alternative representation of S using equivalence classes.

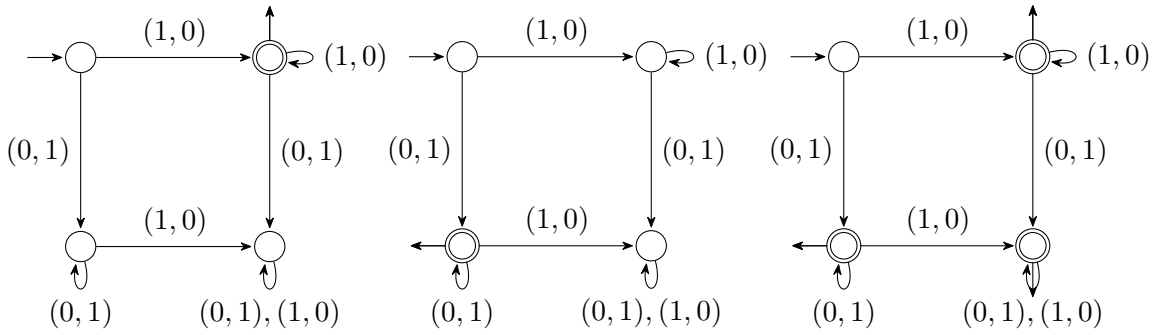
Figure 10: Two representations of $S = (S_1 \cap S_2 \cap S_3) \cup (S_1 \cap {S_2}^c \cap {S_3}^c) \cup ({S_1}^c \cap S_2 \cap {S_3}^c)$.



(a) $\mathcal{A}_1$ for S_1

(b) $\mathcal{A}_2$ for S_2

(c) $\mathcal{A}_3$ for S_3



(d) Automaton for S_d.

(e) Automaton for S_e.

(f) Automaton for S

Figure 11: Construction of the automaton for $S = (S_1 \cap S_2 \cap S_3)^c \cap (S_1 \cap {S_2}^c \cap {S_3}^c)^c \cap ({S_1}^c \cap S_2 \cap {S_3}^c)^c$, where $S_1 = (1,0) + \big((1,0) \cup (0,1)\big)^{\oplus}$, $S_2 = (0,1) + \big((1,0) \cup (0,1)\big)^{\oplus}$, $S_3 = (1,1) + \big((1,0) \cup (0,1)\big)^{\oplus} = S_1 \cap S_2 \cap S_3$, $S_d = S_1 \cap {S_2}^c \cap {S_3}^c$, $S_e = {S_1}^c \cap S_2 \cap {S_3}^c$.

Comment. *The automaton obtained from the proof of Theorem 2 is not always minimal. For example, the one in Figure 11f is not minimal. The minimal automaton is obtained by applying Moore's algorithm and it is shown in Figure 12.*

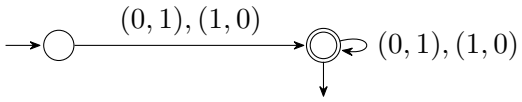


Figure 12: Minimum automaton for S.

Example 10. *Let us consider S the recognizable set depicted in Figure 10a, denoted by the semi-simple expression $S = (1,1)+(1,1)^{\oplus} \cup (1,0)+\big((1,0)\cup(1,1)\big)^{\oplus} \cup (0,1)+\big((0,1)\cup(1,1)\big)^{\oplus}$. Its characteristic series is:*

$$\begin{aligned}
\underline{S} &= \frac{xy}{(1-xy)} + \frac{x}{(1-xy)(1-x)} + \frac{y}{(1-xy)(1-y)} \\
&= \frac{xy(1-x)(1-y) + x(1-y) + y(1-x)}{(1-xy)(1-x)(1-y)} \\
&= \frac{xy - x^2y - xy^2 + x^2y^2 + x - xy + y - xy}{(1-xy)(1-x)(1-y)} \\
&= \frac{(1-xy)(x+y-xy)}{(1-xy)(1-x)(1-y)} \\
&= \frac{x+y-xy}{(1-x)(1-y)}.
\end{aligned}$$

Then, we separate each term of the numerator and we obtain $\underline{S_1}, \underline{S_2}$ and $\underline{S_3}$. We omit the minus for the last term in order to give the resimple expressions, but it is considered afterwards for the equivalence classes.

$$\underline{S_1} = \frac{x}{(1-x)(1-y)} \qquad \underline{S_2} = \frac{y}{(1-x)(1-y)} \qquad \underline{S_3} = \frac{xy}{(1-x)(1-y)}$$

$$S_1 = (1,0) + \big((1,0)\cup(0,1)\big)^{\oplus} \qquad S_2 = (0,1) + \big((1,0)\cup(0,1)\big)^{\oplus} \qquad S_3 = (1,1) + \big((1,0)\cup(0,1)\big)^{\oplus}.$$

Finally, we find the corresponding resimple expression C, illustrated in Figure 10b. This expression comes from considering three equivalence classes: elements in S_1, S_2 and S_3; in S_1 but not in the other two; and in S_3 but not in the other two. The other equivalence classes are not considered because its elements are not in S, or because the class is empty. Our algorithm stops here.

For the sake of this example we check that the characteristic series defined from C and the characteristic series of S coincide.

$$\begin{aligned}
C &= (S_1 \cap S_2 \cap S_3) \cup (S_1 \cap {S_2}^c \cap {S_3}^c) \cup ({S_1}^c \cap S_2 \cap {S_3}^c) \\
&= (S_3) \cup (S_1 \cap {S_2}^c) \cup (S_1^c \cap S_2) \\
&= (1,1) + \big((1,0)\cup(0,1)\big)^{\oplus} \cup (1,0) + (1,0)^{\oplus} \cup (0,1) + (0,1)^{\oplus} \\
\underline{S} &= \frac{xy}{(1-x)(1-y)} + \frac{x}{(1-x)} + \frac{y}{(1-y)} \\
&= \frac{xy}{(1-x)(1-y)} + \frac{x(1-y)}{(1-x)(1-y)} + \frac{y(1-x)}{(1-x)(1-y)} \\
&= \frac{xy + x(1-y) + y(1-x)}{(1-x)(1-y)}.
\end{aligned}$$

After obtaining the resimple expression C we are able to construct the automaton.

5 Algorithm

We prove Theorem 1 by giving the following algorithm. Let A be an alphabet of size k.

Input: E a regular expression over A^*.

Output: If $\mathcal{C}(E)$ is regular then the output is a complete finite state automaton $\mathcal{A}$ over A^* that accepts $\mathcal{C}(E)$, the commutative closure of E. Otherwise, the algorithm stops and warns that $\mathcal{C}(E)$ is not regular.

1. Obtain E' a semi-simple consistent expression over $\mathbb{N}^k$ for $\varphi(E)$ applying Proposition 3 (to make it semi-simple) and Lemma 16 (to make it consistent).
2. Obtain P' and Q' two polynomials in $\mathbb{Z}[x_1, ..., x_k]$ such that $\underline{E'} = \dfrac{P'}{Q'}$, applying Proposition 4 to E'.
3. Obtain P and Q in $\mathbb{Z}[x_1, ..., x_k]$ such that $\dfrac{P}{Q}$ irreducible by simplifing all possible factors of $\dfrac{P'}{Q'}$.
4. If Q has any factor of more than one variable then E it is not recognizable. The algorithm stops and warns that $\mathcal{C}(E)$ is not regular.
5. Obtain C a consistent resimple expression over $\mathbb{N}^k$ from the polynomials P and Q . applying Theorem 3.
6. Obtain the automaton $\mathcal{A}$ applying Theorem 2 to the expression C from the previous step.

6 Complexity

We use the asymptotic big O notation asserting that for functions $f, g : \mathbb{N} \to \mathbb{R}$, $f(n)$ is $\mathcal{O}(g(n))$ if there exists C such that for all sufficiently large n $|g(n)| \leq |Cf(n)|$.

We assume an alphabet of k letters and use the following notation to refer to various size functions. For any set B, the number of elements of B is $|B|$. For any $\sigma = (b_1, \ldots, b_k) \in \mathbb{N}^k$, $||\sigma|| = \max_{1 \leq j \leq k} b_j$; similarly, for any $B \subseteq \mathbb{N}^k$ we denote $||B|| = \max_{\sigma \in B} ||\sigma||$. Finally, for a semi-linear expression $E = \bigcup_{i \in I} \gamma_i + B_i^{\oplus}$ we write $||E|| = \max(\max_{i \in I} ||\gamma_i||, \max_{i \in I} ||B_i||, 2)$.

6.1 State complexity

By the state complexity of a regular language we mean the minimum number of states of a complete automaton that accepts that language [10]. It is a natural measure for operations on regular languages and, in turn, it gives us a lower bound for the temporal and spatial complexity of operations on automata. The study of state complexity dates back at least to [7].

Proposition 18 ([1]). *Every semi-linear set denoted by a semi-linear expression* $E = \bigcup_{i \in I} \gamma_i + B_i^{\oplus}$ *has an equivalent semi-simple expression* $E' = \bigcup_{i \in I'} \gamma_i' + B_i'^{\oplus}$ *where*

$$||\gamma_i'|| \leq ||E||^{|I| \cdot \mathcal{O}(k^6)}, \quad ||B_i'|| \leq ||E||^{|I| \cdot \mathcal{O}(k^4)}, \quad |I'| \leq ||E||^{\mathcal{O}(k^5)}.$$

Although our algorithm starts from a regular expression in A^*, we give the bound on the semi-simple expression of the set of $\mathbb{N}^k$. Let a semi-simple expression be

$$E = \bigcup_{i \in I} \gamma_i + B_i^{\oplus}.$$

Definition 20 (Value p_j). *Given a semi-simple expression* $E = \bigcup_{i \in I} \gamma_i + B_i^{\oplus}$, *for each* $j = 1, \ldots k$, *we define* $p_j = lcm(m_1, \ldots, m_{|I|})$ *where*

$$m_i = \begin{cases} m & \text{if } m\mathbf{e}_j \in B_i \\ 1 & \text{otherwise.} \end{cases}$$

Proposition 19. *Let* $E = \bigcup_{i \in I} \gamma_i + B_i^{\oplus}$. *For each* $j \in \{1, \ldots, k\}$, $p_j = \mathcal{O}\Big(e^{\sqrt{n.log(n)}}\Big)$, *where* n *is the size of the semi-simple expression* E, *that is* $n = |I|\ ||E||$.

Proof. Let $p_j = lcm(m_1, \ldots, m_{|I|})$, where m_i are in Definition 20. So, $m_1 + \ldots + m_{|I|} \leq n$. Let $F(n) = \max\{lcm(o_1, \ldots, o_l) : o_1 + \cdots + o_l = n, l \in \mathbb{N}\}$. Then,

$$p_j = lcm(m_1, \ldots, m_{|I|}) \leq F(n).$$

The problem of finding a good approximation for $F(n)$ is known as Landau's problem. It is well studied and, as shown in [9], $F(n) = \mathcal{O}\Big(e^{\sqrt{n.log(n)}}\Big)$.

□

Proposition 20. *The state complexity of the automaton constructed by our algorithm, taking a semi-simple expression* E, *is at most* $\mathcal{O}\Big(e^{k\sqrt{n.log(n)}}\Big)$ *with* n *the size of* E.

Proof. When introducing resimple expressions and their automata, we observed that the maximum number of states per coordinate does not increase when performing boolean operations between automata derived from consistent resimple expressions (Proposition 15). Therefore, starting from a consistent semi-simple expression $E' = \bigcup_{i \in I'} \gamma_i' + B_i^{\oplus}$ and using Proposition 11, the number of states per coordinate can be bounded by

$$l_j = \max_{i \in I'}\{|\gamma_i'|_j\} + p_j - 1.$$

Then, the state complexity of an automaton derived from a semi-simple consistent expression E' is at most

$$\prod_{j=1}^{k} l_j = \prod_{j=1}^{k} (\max_{i \in I}\{|\gamma_i'|_j\} + p_j - 1) = \mathcal{O}\big((\gamma_{max} + p_{max})^k\big),$$

where $p_{max} = \max_{j \in \{1,\ldots,k\}} p_j = \mathcal{O}\Big(e^{\sqrt{n.log(n)}}\Big)$, $\gamma_{max} = \max_{i \in I} |\gamma_i| = \mathcal{O}(n)$. Clearly $\gamma_{max} \leq ||E'||$, then γ_{max} can be bounded by n.

To transform the semi-simple expression E into a consistent E' we need to change each of the j-primary bases by $p_j\ \mathbf{e}_j$. In the worst case, for each coordinate, we must add to some term c up to $(p_j-1)\mathbf{e}_j$. Then $\max_{i\in I}\{|\gamma'_i|_j\} \leq \max_{i\in I}\{|\gamma_i|_j\} + (p_j-1)$. This does not alter the total state complexity,

$$\prod_{j=1}^{k}(\max_{i\in I}\{|\gamma_i|_j\} + 2(p_j-1)) = \mathcal{O}((\gamma_{max}+p_{max})^k) = \mathcal{O}\left(\left(n + e^{\sqrt{n.log(n)}}\right)^k\right).$$

Thus, the total state complexity is $\mathcal{O}\left(\left(e^{\sqrt{n.log(n)}}\right)^k\right)$. □

Comment. *If we choose to reduce the polynomials as much as possible, we reach the smallest possible p_j, so in general we obtain an automaton as small as possible in each of its coordinates. However, as we have already seen in Example 10, even if the algorithm starts from irreducible P/Q, the resulting automaton is not necessarily the minimum automaton.*

Comment. *In [5] Hoffmann proves for the case of group languages a state complexity of $\mathcal{O}(n^k e^{k\sqrt{n.log(n)}})$, with n being the number of states of the permutation automaton.*

6.2 Time complexity

We call elementary operations any arithmetic operation on natural, rational or real numbers.

Proposition 21. *Our algorithm has a time complexity of $\mathcal{O}(m^2 2^m + m^{7k})$ elementary operations in the worst case, where $m = \mathcal{O}\left(e^{k\sqrt{n.log(n)}}\right)$, n is the size of the semi-simple expression E and k is the size of the alphabet.*

Proof. To transform the semi-simple expression E into a consistent one E' we need to change all the j-primary bases to $p_j\mathbf{e}_j$. For that we need to consider at most $np_1\cdots\ p_k$ simple terms that we use to build the expression E' according to the Lemma 16.

Then E' consists of $t = \mathcal{O}\left(|I|\prod_{j=1}^{k} p_j\right)$ simple terms. Since $|I|$ is a disjoint union and we assume a fixed alphabet, we can limit $\mathcal{O}((\gamma_{max}+p_{max})^k)$, otherwise we would have repeated terms. Also, $\mathcal{O}(\prod_{j=1}^{k} p_j) = \mathcal{O}({p_{max}}^k)$. Then,

$$t = \mathcal{O}\left(((\gamma_{max}+p_{max})p_{max})^k\right) = \mathcal{O}(m^2).$$

When converted to a series, each term of the consistent semi-simple expression generates a fraction. By taking a common denominator, in the worst case, we have to multiply each term by the denominator of the remaining ones. Note that each denominator has at most k factors of the form $(1-\underline{d})$, since the basis from which they come has at most k elements.

If we distribute each of the denominators, the polynomial will have at most 2^k terms. So, when taking a common denominator, we have to multiply each numerator by the $(t-1)$ remaining denominators. Therefore, there remain $2^k(t-1)t$ terms in the numerator. Note that $\mathcal{O}\left(2^k t^2\right) = \mathcal{O}(t^2)$.

Now we need to simplify the denominator factors of more than one variable. We do it by reducing P'/Q'. For this we factor P' and Q' with cost $\mathcal{O}(m^{7k}), see[6]$. This upper bound is because we can bound the degree of each variable of the polynomial by $\mathcal{O}(\gamma_{max} + p_{max})$ and the coefficients, as they are from a characteristic series, are 1 or -1.

In the recognizable case, after factoring and reducing, there cannot be more than m terms. This is because all the resimple expressions have the same infinite part and (by the same argument we used to to bound $|I|$), there are most $(\gamma_{max} + p_{max})^k$ termns. We have at most m automata to intersect. As we already mentioned, the maximum number of states of the intersection automaton is $\mathcal{O}(m)$. Notice that the number of subclasses defined in the construction of the resimple expression is at most $\mathcal{O}(2^m)$. Thus, we have $\mathcal{O}(m2^m)$ boolean operations, each with linear cost in m. Then, the total cost of computing the final automaton from the polynomials is $\mathcal{O}(m^2 2^m)$. We obtain a total worst-case time complexity $\mathcal{O}(m^2 2^m + m^{7k} + m^4)$ which is $\mathcal{O}(m^2 2^m + m^{7k})$. □

7 Acknowledgements

We thank Jacques Sakarovitch for insightful discussions at an early stage of this work. This research was supported by a grant from the University of Buenos Aires.

References

[1] D. Chistikov and C. Haase. The taming of the semi-linear set. In 43rd international colloquium on automata, languages, and programming, ICALP 2016, Rome, Italy, July 12–15, 2016. Proceedings, page 13. Id/No 128.

[2] S. Eilenberg and M.P Schützenberger. Rational sets in commutative monoids. Journal of Algebra, 13(2):173–191, 1969.

[3] S. Ginsburg and E. H Spanier. Bounded regular sets. Proceedings of the American Mathematical Society, 17(5):1043–1049, 1966.

[4] P. Gohon. An algorithm to decide whether a rational subset of $\mathbb{N}^k$ is recognizable. Theoretical Computer Science, 41:51–59, 1985.

[5] S. Hoffmann. State complexity bounds for the commutative closure of group languages. Journal of Automata, Languages and Combinatorics, 28(1-3):27–57, 2023.

[6] A. K. Lenstra. Factoring multivariate integral polynomials. Theoretical Computer Science, 34:207–213, 1984.

[7] A. N. Maslov. Estimates of the number of states of finite automata. Soviet Mathematics, Doklady, 11:1373–1375, 1970.

[8] J. Sakarovitch. Elements of automata theory. Cambridge University Press, Cambridge, 2009. Translated from the 2003 French original by Reuben Thomas.

[9] M. Szalay. On the maximal order in S_n and S_n^*. Acta Arithmetica, 37:321–331, 1980.

[10] S. Yu. Handbook of Formal Languages: Volume 1 Word, Language, Grammar, chapter Regular Languages, pages 41–110. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

Verónica Becher vbecher@dc.uba.ar
Simón Lew Deveali sdeveali@dc.uba.ar
Ignacio Mollo Cunningham imcgham@gmail.com

Departmento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, e Instituto de Ciencias de la Computación(ICC) de la Universidad de Buenos Aires y CONICET. Pabellón 0, Ciudad Universitaria, (1428) Buenos Aires, Argentina