International Journal of Foundations of Computer Science
© World Scientific Publishing Company

UNAMBIGUOUS CONSTRAINED AUTOMATA

MICHAËL CADILHAC
Laboratoire d'informatique théorique et quantique
Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128 succursale Centre-ville, Montréal (Québec) H3C 3J7 Canada
cadilhac@iro.umontreal.ca

ALAIN FINKEL
LSV, ENS Cachan, CNRS
61 av. du Président Wilson, F-94230 Cachan, France
finkel@lsv.ens-cachan.fr

Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche
portant la référence "REACHARD-ANR-11-BS02-001"

PIERRE MCKENZIE
Université de Montréal and Chaire Digiteo ENS Cachan-École Polytechnique
Same address as first author
mckenzie@iro.umontreal.ca

Supported by Digiteo and the Natural Sciences and Engineering Research Council of Canada

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

The class of languages captured by Constrained Automata (CA) that are unambiguous is shown to possess more closure properties than the provably weaker class captured by deterministic CA. Problems decidable for deterministic CA are nonetheless shown to remain decidable for unambiguous CA, and testing for *regularity* is added to this set of decidable problems. Unambiguous CA are then shown incomparable with deterministic reversal-bounded machines in terms of expressivity, and a *deterministic* model equivalent to unambiguous CA is identified.

Keywords: Constrained automata; Parikh automata; unambiguity; regularity test.

1. Introduction

A recent trend in automata theory is to study flavors of nondeterminism, which are introduced to provide a scale of expressiveness in different models (see [5] for a survey). The usual goal is to strike a balance between the expressiveness of nondeterministic models and the undecidability properties that often come with nondeterminism. A natural restriction to nondeterminism is *unambiguity*, i.e., the property that despite the underlying nondeterminism, there be at most one way to accept an input word. Within the context of finite automata, unambiguity and nondeterminism are equally expressive, but many open problems concerning the state complexity of unambiguity remain. Within more general contexts, the first question is often whether unambiguity offers more expressiveness than determinism; if so, then the examination of the closure and decidability properties of the new class often reveals that it inherits good properties. Another line of attack is to find a deterministic model equivalent to an unambiguous model, so as to understand how unambiguity affects a given model.

In [10], Klaedtke and Rueß studied Constrained Automata (CA),[a] a model whose expressive power lies between regular languages and context-sensitive languages [10, 4]. Klaedtke and Rueß successfully used the CA in the model-checking of hardware circuits, suggesting that CA is a model of interest for real-life applications. Bouajjani and Habermehl [2] also used a variant of CA for the model-checking of FIFO-channel systems. The deterministic variant (DetCA) of the CA enjoys more closure properties (e.g., complement) and decidability properties (e.g., universality) than the CA, but is unable to express languages as simple as $\{a, b\}^* \cdot \{a^n b^n \mid n \geq 1\}$ [4]. Buoyed by Colcombet's recent systematic examination of unambiguity [5], here we initiate the study of unambiguous CA (UnCA).

We show that UnCA enjoy more closure properties than DetCA, while being more expressive. The class of languages UnCA defines is indeed closed under Boolean operations, inverse morphisms, commutative closure, reversal, and right and left quotient. We show that the problems known to be decidable for DetCA (emptiness, universality, finiteness, inclusion) remain decidable for UnCA. As the main technical result of this paper, we show that regularity is decidable for UnCA; by contrast, regularity is known to be undecidable for CA [4], while its status was unknown for DetCA. Finally, although DetCA are less powerful than UnCA, we present a natural *deterministic* model equivalent to UnCA; as a result of independent interest, we show that the nondeterministic variant of this model has the same expressive power as CA.

Section 2 in this paper contains preliminaries. Section 3 investigates the closure and expressiveness properties of UnCA. Section 4 compares UnCA and deterministic reversal-bounded counter machines. Section 5 proceeds with the decidability properties of UnCA, showing, as our main result, that regularity is decidable. Section 6 shows that there is a natural equivalent deterministic model to UnCA. Section 7 concludes with a brief discussion.

2. Preliminaries

Integers, Vectors, Monoids. We write $\mathbb{N}$ for the nonnegative integers. Let $d \geq 1$. Vectors in $\mathbb{N}^d$ are noted in bold, e.g., $\mathbf{v}$ whose elements are $v_1, v_2, \ldots, v_d$. We write

---

[a] In [10], the model under study is called *Parikh automata*. CA are but an effectively equivalent model with an arguably simpler definition.

$\mathbf{e}_i \in \{0,1\}^d$ for the vector having a 1 only in position i and $\mathbf{0}$ for the all-zero vector. We view $\mathbb{N}^d$ as the additive monoid $(\mathbb{N}^d, +)$, with $+$ the component-wise addition and $\mathbf{0}$ the identity element. Given an order on some set $\Sigma = \{\ell_1, \ell_2, \dots, \ell_n\}$ we often refer to the components of a vector $\mathbf{v} \in \mathbb{N}^{|\Sigma|}$ by x_{ℓ_i} instead of x_i. In particular, for $\ell \in \Sigma$, x_ℓ refers to the i-th component of $\mathbf{x}$ where i is such that $\ell_i = \ell$.

Let $s \geq 0$ and $p \geq 1$, we define the congruence $\equiv_{s,p}$, by $x \equiv_{s,p} y$ iff $(x = y < s) \vee (x, y \geq s \wedge x = y \pmod{p})$, for $x, y \in \mathbb{N}$; we write $[x]_{s,p}$ for the equivalence class of x under $\equiv_{s,p}$. We extend $\equiv_{s,p}$ component-wise to vectors $\mathbf{x}, \mathbf{y} \in \mathbb{N}^d$ by letting $\mathbf{x} \equiv_{s,p} \mathbf{y}$ iff $x_i \equiv_{s,p} y_i$ for all $1 \leq i \leq d$; similarly, $[\mathbf{x}]_{s,p}$ is the equivalence class of $\mathbf{x}$ under this relation.

For a monoid $(M, \cdot)$ and $S \subseteq M$, we write S^* for the monoid generated by S, i.e., the smallest submonoid of $(M, \cdot)$ containing S. A (monoid) *morphism* from $(M, \cdot)$ to $(N, \circ)$ is a function $h\colon M \to N$ such that $h(m_1 \cdot m_2) = h(m_1) \circ h(m_2)$, and, with e_M (resp. e_N) the identity element of M (resp. N), $h(e_M) = e_N$. Moreover, if $M = S^*$ for some finite set S (and this will always be the case), then h need only be defined on the elements of S. In this case, h is said to be *erasing* if there is an $s \in S$ such that $h(s) = e_N$. If in addition $N = T^*$ for some finite set T, h is said to be *length-preserving* if for all $s \in S$, $h(s) \in T$.

Semilinear Sets, Parikh Image. A subset C of $\mathbb{N}^d$ is *linear* if there exist $\mathbf{c} \in \mathbb{N}^d$ and a finite set $P \subseteq \mathbb{N}^d$ of *periods* such that $C = \mathbf{c} + P^*$. The subset C is said to be *semilinear* if it is equal to a finite union of linear sets: $\{4n+56 \mid n > 0\}$ is semilinear while $\{2^n \mid n > 0\}$ is not. We will often use the fact that the semilinear sets are those sets of natural numbers definable in first-order logic with addition [6]. A semilinear set is said to be *effectively semilinear* if its description as a set of $\mathbf{c}$'s and P's, or equivalently as a formula, can be computed from the input to the problem. Let $\Sigma = \{\ell_1, \ell_2, \dots, \ell_n\}$ be an alphabet[b] and write ε for the empty word. The *Parikh image* [6] is the morphism $\mathsf{Pkh}\colon \Sigma^* \to \mathbb{N}^n$ defined by $\mathsf{Pkh}(\ell_i) = \mathbf{e}_i$, for $1 \leq i \leq n$ — in particular, we have that $\mathsf{Pkh}(\varepsilon) = \mathbf{0}$. The Parikh image of a language L is defined as $\mathsf{Pkh}(L) = \{\mathsf{Pkh}(w) \mid w \in L\}$. The name of this morphism stems from Parikh's theorem [12], which states that the Parikh image of any context-free language is semilinear. For $L \subseteq \Sigma^*$ and $C \subseteq \mathbb{N}^n$, define $L{\restriction_C}$ (read "L constrained by C") as $\{w \in L \mid \mathsf{Pkh}(w) \in C\}$.

Languages, Operations. Let $u = a_1 a_2 \cdots a_n \in \Sigma^*$, $a_i \in \Sigma$. We write $|u|_\ell$, for $\ell \in \Sigma$, for the number $|\{i \mid a_i = \ell\}|$. We define $u^{\mathrm{R}} = a_n \cdots a_2 a_1$ as the *reversal* of u. For $L_1, L_2 \subseteq \Sigma^*$, define L_1^{R} as the set of the reversals of each word in L_1; $(L_1)^{-1}L_2 = \{v \mid (\exists u \in L_1)[u \cdot v \in L_2]\}$ as the *left quotient* of L_2 by L_1; and $L_1(L_2)^{-1} = \{u \mid (\exists v \in L_2)[u \cdot v \in L_1]\}$ as the *right quotient* of L_1 by L_2. A language $L \subseteq \Sigma^*$ is *bounded* [7] if there exist $n > 0$ and a sequence of words $w_1, w_2, \dots, w_n \in \Sigma^+$, which we call *a socle of* L, such that $L \subseteq w_1^* w_2^* \cdots w_n^*$. The

---

[b] We will always assume some implicit ordering on the alphabets.

iteration set of L w.r.t. this socle is (uniquely) defined as $\mathsf{Iter}_{(w_1,w_2,\ldots,w_n)}(L) = \{(i_1, i_2, \ldots, i_n) \in \mathbb{N}^n \mid w_1^{i_1} w_2^{i_2} \cdots w_n^{i_n} \in L\}$; an iteration set contains *all* possible ways to iterate the words in the socle to obtain a word in L.

Automata. An automaton is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is the finite set of states, Σ is an alphabet, $\delta \subseteq Q \times \Sigma \times Q$ is the set of transitions, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. For a transition $t = (q, a, q') \in \delta$, we write $t = q \bullet\!\!-a\!\!\to q'$ and define $\mathsf{From}(t) = q$ and $\mathsf{To}(t) = q'$. Moreover, we define $\mu_A \colon \delta^* \to \Sigma^*$ to be the length-preserving morphism given by $\mu_A(t) = a$, with, in particular, $\mu_A(\varepsilon) = \varepsilon$, and we write μ when A is clear from the context. A *path* π on A is a word $\pi = t_1 t_2 \cdots t_n \in \delta^*$ such that $\mathsf{To}(t_i) = \mathsf{From}(t_{i+1})$ for $1 \leq i < n$; we extend From and To to paths, letting $\mathsf{From}(\pi) = \mathsf{From}(t_1)$ and $\mathsf{To}(\pi) = \mathsf{To}(t_n)$. We say that $\mu(\pi)$ is the *label* of π. A path π is said to be *initial* if $\mathsf{From}(\pi) = q_0$, *final* if $\mathsf{To}(\pi) \in F$, and *accepting* if it is both initial and final; we write $\mathsf{Run}(A)$ for the language over δ of accepting paths (or *runs*) on A. We write $L(A)$ for the language of A, i.e., the labels of the accepting paths. The automaton A is said to be *deterministic* if $(p \bullet\!\!-a\!\!\to q \in \delta \land p \bullet\!\!-a\!\!\to q' \in \delta)$ implies $q = q'$. An ε-automaton is an automaton $A = (Q, \Sigma, \delta, q_0, F)$ as above, except with $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ so that in particular μ_A becomes an erasing morphism.

Affine Functions. We consider the vectors in $\mathbb{N}^d$ to be *column* vectors. A function $f \colon \mathbb{N}^d \to \mathbb{N}^d$ is a (total and positive) *affine function* of dimension d if there exist a matrix $M \in \mathbb{N}^{d \times d}$ and $\mathbf{v} \in \mathbb{N}^d$ such that for any $\mathbf{x} \in \mathbb{N}^d$, $f(\mathbf{x}) = M.\mathbf{x} + \mathbf{v}$. We abusively write $f = (M, \mathbf{v})$. We let $\mathcal{F}_d$ be the set of such functions; we view $\mathcal{F}_d$ as the monoid $(\mathcal{F}_d, \diamond)$ with $(f \diamond g)(\mathbf{x}) = g(f(\mathbf{x}))$, where the identity element of the monoid is the identity function, i.e., $(Id, \mathbf{0})$ where Id is the identity matrix of dimension d. Let U be a monoid morphism from Σ^* to $\mathcal{F}_d$. For $w \in \Sigma^*$, we write U_w for $U(w)$, so that the application of $U(w)$ to a vector $\mathbf{v}$ is written $U_w(\mathbf{v})$, and U_ε is the identity function. We define $\mathcal{M}(U)$ as the multiplicative matrix monoid generated by the matrices used to define U, i.e., $\mathcal{M}(U) = \{M \mid (\exists a \in \Sigma)(\exists \mathbf{v})[U_a = (M, \mathbf{v})]\}^*$.

Definition 1 (Constrained automaton [4]) *A constrained automaton (CA) is a pair (A, C) where A is an ε-automaton with d transitions and $C \subseteq \mathbb{N}^d$ is semilinear. Its language is $L(A, C) = \mu(\mathsf{Run}(A)\restriction_C)$. The CA is said to be:*

- Deterministic (DetCA) *if A is a deterministic automaton;*
- Unambiguous (UnCA) *if A is an unambiguous ε-automaton.*

We write $\mathcal{L}_{\mathrm{CA}}$, $\mathcal{L}_{\mathrm{DetCA}}$,[c] and $\mathcal{L}_{\mathrm{UnCA}}$ for the classes of languages recognized by CA, DetCA, and UnCA, respectively.

---

[c] In [4], $\mathcal{L}_{\mathrm{CA}}$ and $\mathcal{L}_{\mathrm{DetCA}}$ are written $\mathcal{L}_{\mathrm{PA}}$ and $\mathcal{L}_{\mathrm{DetPA}}$, in reference to Parikh automata [10], which are equivalent to CA.

3. Closure properties and expressiveness of UnCA

In this section, we show closure and nonclosure properties, and we give languages witnessing the strict inclusion chain $\mathcal{L}_{\text{DetCA}} \subsetneq \mathcal{L}_{\text{UnCA}} \subsetneq \mathcal{L}_{\text{CA}}$. We start with a tool that will prove useful when combining UnCA:

Lemma 2. *For any UnCA (A, C), there is an UnCA (A', C') where A' has no ε-transition, $L(A) = L(A')$, and $L(A, C) = L(A', C')$.*

Proof. Let (A, C) be an UnCA with $A = (Q, \Sigma, \delta, q_0, F)$. We first note that for $p, q \in Q$, and $\ell \in \Sigma \cup \{\varepsilon\}$, we may suppose there is at most one way to reach q from p reading ℓ. Indeed, suppose there are two paths π_1, π_2 from p to q labeled ℓ, and suppose there is a path ρ_1 from q_0 to p (otherwise, we may remove p) and a path ρ_2 from q to a final state (otherwise, we may remove q). Then $\rho_1\pi_1\rho_2$ and $\rho_1\pi_2\rho_2$ are two accepting paths with the same label, contradicting the unambiguity of A. In particular, this implies that there is no cycle of ε-transitions, since if π is such a cycle, one may go from and to $\mathsf{From}(\pi)$ reading ε using *two* paths: π and the empty path. In the same vein, we note that we may suppose that for a state q, if there is a path of ε-transitions from q to a final state q', then it is unique and there is no such path between q and a different final state.

Now we "backward-close" the ε-automaton A. For $p, q \in Q$ and $\ell \in \Sigma$, define $P(p, \ell, q)$ to be the only path from p to q labeled ℓ which ends in a transition labeled ℓ; if none exists, set $P(p, \ell, q)$ to $\perp$. Likewise, define $E(p)$ to be the unique path labeled ε from p to a final state of A, if it exists, and $\perp$ otherwise. Note that $E(p) = \varepsilon$ if $p \in F$. Define $A' = (Q, \Sigma, \delta', q_0, F')$ where:

$$\begin{aligned} \delta' &= \{p \bullet\!\!-\ell\!\!\rightarrow q \mid \ell \in \Sigma \wedge P(p, \ell, q) \neq \perp\} \ , \\ F' &= \{p \mid E(p) \neq \perp\} \ . \end{aligned}$$

Clearly, this automaton has the same language as A. Further, we argue that it is unambiguous. Let $h \colon \delta'^* \to \delta^*$ be the morphism defined by $h(p \bullet\!\!-\ell\!\!\rightarrow q) = P(p, \ell, q)$. It is not hard to see that h is a bijection from $\mathsf{Run}(A')$ to:

$$\mathsf{Run}(A)^{-\varepsilon} = \mathsf{Run}(A)(\{t \in \delta \mid \mu(t) = \varepsilon\}^*)^{-1} \ ,$$

that is, the initial paths in A ending in a state from which we can reach a final state by following ε-transitions.

Now note that there is a one-to-one correspondence that preserves labels between $\mathsf{Run}(A)$ and $\mathsf{Run}(A)^{-\varepsilon}$ that consists in removing the trailing ε-transitions. As h also preserves labels, this implies that A' is unambiguous and $L(A') = L(A)$. Moreover, with $\pi' \in \mathsf{Run}(A')$, the only path $\pi \in \mathsf{Run}(A)$ with the same label is given by $\pi = h(\pi')E(\mathsf{To}(\pi'))$. We thus define the constraint set C' so that $\mathsf{Pkh}(\pi') \in C'$ iff $\mathsf{Pkh}(\pi) \in C$. For this, we need to know, given the Parikh image of a run in A', in which state q the run ends, so that we can add $\mathsf{Pkh}(E(q))$ to retrieve the Parikh image of the similar path in A:

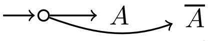
Fact 3. *Let A be an automaton. For each final state q of A, the set of Parikh images of paths in $\mathsf{Run}(A)$ ending in q is effectively semilinear. Moreover, those sets are disjoint.*

Now, order $\delta' = \{t'_1, t'_2, \dots, t'_k\}$. Next, for $q \in F'$, let R_q be the semilinear set of Parikh images of initial paths in A' ending in q. Note that Fact 3 implies that for $\pi \in \mathsf{Run}(A')$, $\mathsf{Pkh}(\pi) \in R_q$ iff $\mathsf{To}(\pi) = q$. Then define $C' \subseteq \mathbb{N}^k$ by letting $\mathbf{x} = (x_1, x_2, \dots, x_k) \in C'$ iff:

$$\bigwedge_{q \in F'} \left[(\mathbf{x} \in R_q) \to \left(\sum_{i=1}^{k} x_i \times \mathsf{Pkh}(h(t'_i)) + \mathsf{Pkh}(E(q)) \right) \in C \right] .$$

Concluding the proof of Lemma 2, a word $w \in L(A', C')$ iff $w \in L(A')$ and the Parikh image of the only path labeled w in $\mathsf{Run}(A')$ is in C', that is iff $w \in L(A)$ and the Parikh image of the only path labeled w in $\mathsf{Run}(A)$ is in C, that is iff $w \in L(A, C)$. □

Proposition 4. *$\mathcal{L}_{\mathrm{UnCA}}$ is closed under union.*

Proof (sketch). First, we note that for an UnCA (A, C) over the alphabet Σ, there is an UnCA (A', C') with $L(A') = \Sigma^*$ and $L(A', C') = L(A, C)$. The ε-automaton A' is defined as  where $\overline{A}$ is a deterministic automaton for $\overline{L(A)}$ and the two new transitions are labeled by ε. Then C' is defined to reject if the transition to $\overline{A}$ is taken, and to accept if the run is in A and its Parikh image is in C. Clearly, A' is unambiguous.

Now let (A, C) and (B, D) be two UnCA over the same alphabet Σ (w.l.o.g.), and with $L(A) = L(B) = \Sigma^*$, as per the previous discussion. We design an automaton that runs A and B in parallel. We rely on Lemma 2 to synchronize the two automata. For any word w, there will be exactly one way to read w over A and B to reach acceptance, thus only one way to read w over both at the same time. Finally, we constrain this automaton by extracting the paths in A and B and checking that at least one of them is in its respective constraint set. □

Proposition 5. *$\mathcal{L}_{\mathrm{UnCA}}$ is closed under complement, intersection, inverse morphisms and commutative closure.*

Proof. (Complement and intersection) Let (A, C) be an UnCA. A word w is *not* in $L(A, C)$ iff either $w \notin L(A)$ or $w \in L(A)$ but the Parikh image of the only path for w in A is rejected by C. Thus $\overline{L(A, C)} = \overline{L(A)} \cup L(A, \overline{C})$. Now $\overline{L(A)}$ is regular, thus $\overline{L(A)} \in \mathcal{L}_{\mathrm{UnCA}}$. Moreover, $(A, \overline{C})$ is an UnCA. Thus $\overline{L(A, C)}$ is the union of the languages of two UnCA, and by Proposition 4, it is in $\mathcal{L}_{\mathrm{UnCA}}$. Closure under intersection follows from the closure under union and complement.

(Inverse morphisms) Let (A, C) be an UnCA over Σ and $h \colon \mathrm{T}^* \to \Sigma^*$ be a language morphism. Write $A = (Q, \Sigma, \delta, q_0, F)$ and $P(q, u, q')$ the only path in A

from q to q' labeled u if it exists, and $\perp$ otherwise. We set $\mathsf{Path}(q, \varepsilon, q) = \varepsilon$ for any q. Define $B = (Q, \mathrm{T}, \delta', q_0, F)$ by: $\delta' = \{q \bullet\!\!-\ell\!\!\to q' \in Q \times \mathrm{T} \times Q \mid P(q, h(\ell), q') \neq \perp\}$. Now B is unambiguous as A is. Define $C' \subseteq \mathbb{N}^{|\delta'|}$ by letting $\mathbf{x} = (x_t)_{t \in \delta'} \in C'$ iff $\sum_{t = q \bullet\!-\ell\!\to q' \in \delta'} x_t \times \mathsf{Pkh}(P(q, h(\ell), q')) \in C$. It is clear that $L(B, C') = h^{-1}(L(A, C))$, concluding the proof.

(Commutative closure) It is shown in [4] that the commutative closure of a language in $\mathcal{L}_{\mathrm{CA}}$ is in $\mathcal{L}_{\mathrm{DetCA}}$, and thus in $\mathcal{L}_{\mathrm{UnCA}}$. □

Note that $\mathcal{L}_{\mathrm{DetCA}}$ is not closed under reversal, as $\{a, b\}^* \cdot \{a^n b^n \mid n \geq 1\}$ is not in $\mathcal{L}_{\mathrm{DetCA}}$ while its reversal is [4]. Thus it is a curiosity, especially for a class described by a deterministic model (see the forthcoming Theorem 32), that we have:

Proposition 6. *$\mathcal{L}_{\mathrm{UnCA}}$ is closed under reversal.*

Proof. Let (A, C) be an UnCA. Let B be the ε-automaton A in which a new state q_{f} is set to be the only final state, and with a transition from each former final state to q_{f} labeled ε. Clearly, B is unambiguous. Adjust C into C' so that the added transitions in B do not affect the acceptance of a word, i.e., $L(B, C') = L(A, C)$. Then define D as the ε-automaton B in which every transition is reversed, i.e., $q \bullet\!\!-\ell\!\!\to q'$ is a transition of B iff $q' \bullet\!\!-\ell\!\!\to q$ is a transition of D; the order on the transition set of D is the same as that of B. Additionally, set q_{f} as the initial state and the former initial state of B as the only final state. Then D is unambiguous: clearly, $\mathsf{Run}(B)$ is the set of paths which are the reversal of paths in $\mathsf{Run}(D)$ and where each transition is reversed, thus the accepting paths in D labeled w are the reversal of the accepting paths in B labeled w^{R}. As B is unambiguous, only one such path may exist, thus D is unambiguous. Hence $L(D, C') = (L(B, C'))^{\mathrm{R}} = (L(A, C))^{\mathrm{R}}$. □

Proposition 7. *Let $L_1 \in \mathcal{L}_{\mathrm{CA}}$ and $L_2 \in \mathcal{L}_{\mathrm{UnCA}}$. Then $L_1^{-1} L_2 \in \mathcal{L}_{\mathrm{UnCA}}$.*

Proof. Let (A, C) be a CA, (B, D) an UnCA, with $A = (Q_A, \Sigma, \delta_A, q_{0,A}, F_A)$ and $B = (Q_B, \Sigma, \delta_B, q_{0,B}, F_B)$. We suppose, thanks to Lemma 2, that no transition of B is labeled by ε, and that each state of B is reachable from $q_{0,B}$ and can reach a final state. For $q \in Q_B$, define $B^{\to q}$ (resp. $B^{q \to}$) to be the ε-automaton B where the initial state (resp. the only final state) is q, and note that $B^{\to q}$ is unambiguous, as any path from q to a final state can be prefixed with a path from $q_{0,B}$ to q to make an accepting path in B. First, we note that a consequence of Parikh's theorem [12] is that:

Fact 8. *For any $q_B \in Q_B$, the set $E^{q_B} = \{(\mathsf{Pkh}(\pi), \mathsf{Pkh}(\rho)) \mid \pi \in \mathsf{Run}(A) \wedge \rho \in \mathsf{Run}(B^{q_B \to}) \wedge \mu_A(\pi) = \mu_B(\rho)\}$ is effectively semilinear.*

A word w is in $(L(A, C))^{-1} L(B, D)$ iff there is a state $q_B \in Q_B$ and a word $u \in L(A, C)$ such that $u \in L(B^{q_B \to})$, $w \in L(B^{\to q_B})$, and the Parikh image of one (in fact, the only) path for u in $B^{q_B \to}$ concatenated with the path for w in $B^{\to q_B}$ is in D. This is the case iff there is a state $q_B \in Q_B$ and a pair $(\mathbf{x}, \mathbf{y}) \in E^{q_B}$ such that

$\mathbf{x} \in C$ and the Parikh image $\mathbf{z}$ of the only path in $B^{\to q_B}$ labeled w plus $\mathbf{y}$ is in D. In symbols, a word w is in $(L(A,C))^{-1}L(B,D)$ iff it is in:

$$\bigcup_{q_B \in Q_B} L(B^{\to q_B},\ \{\mathbf{z} \mid (\exists(\mathbf{x},\mathbf{y}) \in E^{q_B})[\mathbf{x} \in C \land \mathbf{y} + \mathbf{z} \in D]\})\ .$$

As $\mathcal{L}_{\text{UnCA}}$ is closed under union (Proposition 4), this implies the result. ☐

Remark 9. *In the previous proof, if (B,D) is a DetCA, then we obtain at the end a set of DetCA, the union of the languages of which is $(L(A,C))^{-1}L(B,C)$. As $\mathcal{L}_{\text{DetCA}}$ is closed under union, this shows that $\mathcal{L}_{\text{DetCA}}$ is also closed under left quotient. Moreover, if (B,D) is an UnCA, then $B^{q\to}$ is unambiguous: if two accepting paths therein have the same label, then there are two ways to get from $q_{0,B}$ to q reading the same word, and since a final state can be reached from q, the unambiguity of B implies that they are the same paths. Likewise, if (B,D) is a DetCA, then $B^{q\to}$ is deterministic. Thus a similar proof as the above shows that both $\mathcal{L}_{\text{UnCA}}$ and $\mathcal{L}_{\text{DetCA}}$ are closed under* right *quotient — in the case of DetCA, this settles those questions left open in* [10]. *An alternative proof of the closure under right quotient of $\mathcal{L}_{\text{UnCA}}$ is to note that $L_1(L_2)^{-1} = ((L_2^{\text{R}})^{-1}L_1^{\text{R}})^{\text{R}}$. Thus the closure of $\mathcal{L}_{\text{UnCA}}$ under reversal (Proposition 6) and under left quotient (Proposition 7) imply that $\mathcal{L}_{\text{UnCA}}$ is indeed closed under right quotient.*

We introduce an expressiveness lemma inspired by those in [4], and which is shown in a similar way. It is based on the idea that given a path $\pi = \pi_1\rho\pi_2\rho\pi_3$ where ρ is a cycle, grouping the cycles together (i.e., considering $\pi_1\rho^2\pi_2\pi_3$ and $\pi_1\pi_2\rho^2\pi_3$) affects neither its being an accepting path, nor its Parikh image.

Lemma 10. *Let $L \subseteq \Sigma^*$ be in $\mathcal{L}_{\text{CA}}$. There exist $p, \ell \geq 1$ such that for any $v_0, v_1, \ldots, v_\ell \in \Sigma^*$ and $u_1, u_2, \ldots, u_\ell \in \Sigma^{\geq p}$ such that $v_0u_1v_1 \cdots u_\ell v_\ell \in L$, there exist $1 \leq i < j \leq \ell$ and a nonempty $w \in \Sigma^*$ with $|w| \leq p$ such that:*
(1). $u_i = u_{i,1} \cdot w \cdot u_{i,2}$ and $u_j = u_{j,1} \cdot w \cdot u_{j,2}$,
(2). $v_0u_1v_1 \cdots (u_{i,1} \quad \cdot \quad u_{i,2})v_i \cdots (u_{j,1} \cdot w^2 \cdot u_{j,2})v_j \cdots u_\ell v_\ell \in L$,
(3). $v_0u_1v_1 \cdots (u_{i,1} \cdot w^2 \cdot u_{i,2})v_i \cdots (u_{j,1} \quad \cdot \quad u_{j,2})v_j \cdots u_\ell v_\ell \in L$.

Now let P_1 be the prefixes of the semi-Dyck language of two parentheses:

$$P_1 = \{a_1a_2 \cdots a_k \in \{\sqsubset, \sqsupset\}^* \mid (\forall i)[a_i \in \Sigma \land |a_1a_2 \cdots a_i|_\sqsubset \geq |a_1a_2 \cdots a_i|_\sqsupset]\}\ .$$

Proposition 11. *$P_1 \notin \mathcal{L}_{\text{CA}}$ and $\overline{P_1} \in \mathcal{L}_{\text{CA}} \setminus \mathcal{L}_{\text{UnCA}}$.*

Proof. ($P_1 \notin \mathcal{L}_{\text{CA}}$.) We use Lemma 10. Suppose $P_1 \in \mathcal{L}_{\text{CA}}$, and let p, ℓ be as in Lemma 10. Define $v_0 = \varepsilon$ and for all $1 \leq i \leq \ell$, $u_i = \sqsubset^p$, $v_i = \sqsupset^p$. Lemma 10 then asserts that there is $1 \leq k \leq p$ such that $u_1v_1 \cdots \sqsubset^{p-k}\sqsupset^p \cdots \sqsubset^{p+k}\sqsupset^p \cdots u_\ell v_\ell \in P_1$, a contradiction.

($\overline{P_1} \in \mathcal{L}_{\text{CA}}$.) We only note that we can design a CA for $\overline{P_1}$ which guesses a position in the input word at which the number of $\sqsubset$'s read so far is less than the number of $\sqsupset$'s.

($\overline{P_1} \notin \mathcal{L}_{\mathrm{UnCA}}$.) If $\overline{P_1} \in \mathcal{L}_{\mathrm{UnCA}}$, then $P_1 \in \mathcal{L}_{\mathrm{UnCA}}$ (Proposition 5), but as $\mathcal{L}_{\mathrm{UnCA}} \subseteq \mathcal{L}_{\mathrm{CA}}$, this contradicts $P_1 \notin \mathcal{L}_{\mathrm{CA}}$. ☐

Theorem 12. $\mathcal{L}_{\mathrm{DetCA}} \subsetneq \mathcal{L}_{\mathrm{UnCA}} \subsetneq \mathcal{L}_{\mathrm{CA}}$.

Proof. ($\mathcal{L}_{\mathrm{DetCA}} \subsetneq \mathcal{L}_{\mathrm{UnCA}}$.) The inclusion follows from the fact that a deterministic automaton is unambiguous, thus a DetCA is an UnCA. The strictness of the inclusion is shown in [3]: the language $\{a, b\}^* \cdot \{a^n b^n \mid n \geq 1\}$ is in $\mathcal{L}_{\mathrm{UnCA}} \setminus \mathcal{L}_{\mathrm{DetCA}}$. Additionally, we already hinted that $\mathcal{L}_{\mathrm{DetCA}}$ is not closed under reversal while $\mathcal{L}_{\mathrm{UnCA}}$ is (Proposition 6), implying that the two classes differ.

($\mathcal{L}_{\mathrm{UnCA}} \subsetneq \mathcal{L}_{\mathrm{CA}}$.) We already noted that the inclusion is immediate, as an UnCA is a CA. Its strictness comes from Proposition 11, or alternatively, from the fact that $\mathcal{L}_{\mathrm{CA}}$ is not closed under complement while $\mathcal{L}_{\mathrm{UnCA}}$ is. ☐

Proposition 13. $\mathcal{L}_{\mathrm{UnCA}}$ *is neither closed under concatenation with a regular language, nor under length-preserving morphisms, nor under starring.*

Proof. (Concatenation.) Let $\Sigma = \{\sqsubset, \sqsupset\}$. The language $L_< = \{w \in \Sigma^* \mid |w|_\sqsubset < |w|_\sqsupset\}$ is in $\mathcal{L}_{\mathrm{DetCA}}$ and such that $\overline{P_1} = L_< \cdot \Sigma^* \notin \mathcal{L}_{\mathrm{UnCA}}$. Thus if $\mathcal{L}_{\mathrm{UnCA}}$ were closed under concatenation, then $\overline{P_1}$ would be in $\mathcal{L}_{\mathrm{UnCA}}$, contradicting Proposition 11.

(Length-preserving morphisms and starring.) Let $\mathrm{T} = \{|\!\!\sqsubset, \sqsupset\!\!|\}$, then $L_< \cdot \mathrm{T}^* \in \mathcal{L}_{\mathrm{UnCA}}$. The length-preserving morphism $h \colon (\Sigma \cup \mathrm{T})^* \to \Sigma^*$ defined by $h(|\!\!\sqsubset) = h(\sqsubset) = \sqsubset$, $h(\sqsupset\!\!|) = h(\sqsupset) = \sqsupset$ is such that $h(L_< \cdot \mathrm{T}^*) = L_< \cdot \Sigma^* \notin \mathcal{L}_{\mathrm{UnCA}}$. For starring, it is shown in [4] that with $L = \{a^n b^n \mid n \in \mathbb{N}\} \in \mathcal{L}_{\mathrm{DetCA}}$, $L^* \notin \mathcal{L}_{\mathrm{CA}} \supsetneq \mathcal{L}_{\mathrm{UnCA}}$. ☐

4. UnCA and RBCM

It is known that *one-way reversal-bounded counter machines* (RBCM) [9] are as powerful as CA [10], while deterministic such machines (DetRBCM) are more powerful than DetCA [4]. In this section, we carry this study further by showing that the expressive power of DetRBCM is incomparable with that of UnCA.

Proposition 14. $\mathcal{L}_{\mathrm{DetRBCM}}$ *and* $\mathcal{L}_{\mathrm{UnCA}}$ *are incomparable.*

Proof. ($\mathcal{L}_{\mathrm{DetRBCM}} \nsubseteq \mathcal{L}_{\mathrm{UnCA}}$.) A DetRBCM can deterministically use extra information provided in the input word to check for a certain property later in the input; this is illustrated by the fact that the following language is in $\mathcal{L}_{\mathrm{DetRBCM}}$:

$$L = \{\ell^n w \mid w = a_1 a_2 \cdots a_n \cdots \in \{\sqsubset, \sqsupset\}^* \wedge |a_1 a_2 \cdots a_n|_\sqsubset < |a_1 a_2 \cdots a_n|_\sqsupset\} \ .$$

Indeed, the DetRBCM starts by counting the number of ℓ's, then decrements this counter while reading w and counting the number of $\sqsubset$'s and $\sqsupset$'s. When the number of ℓ's reaches zero, the machine checks whether the number of $\sqsubset$'s read so far is strictly less than the number of $\sqsupset$'s read and accepts iff it is the case. This is done by decrementing the counters simultaneously until one of them reaches 0, while keeping the input head in place; this is deterministic w.r.t. the values of the counters.

Suppose $L \in \mathcal{L}_{\mathrm{UnCA}}$. Proposition 7 then asserts that $(\{\ell\}^*)^{-1}L \cap \{\sqsubset, \sqsupset\}^*$ is in $\mathcal{L}_{\mathrm{UnCA}}$. But this latter language is $\overline{P_1} \notin \mathcal{L}_{\mathrm{UnCA}}$ (Proposition 11), a contradiction.

($\mathcal{L}_{\mathrm{UnCA}} \not\subseteq \mathcal{L}_{\mathrm{DetRBCM}}$.) The language $\{a,b\}^* \cdot \{a^n b^n \mid n \geq 1\}$ is in $\mathcal{L}_{\mathrm{UnCA}}$ but not in $\mathcal{L}_{\mathrm{DetRBCM}}$ [4]. □

5. Decision problems for UnCA

We recall the following decidability results, which hold equally well for UnCA:

Proposition 15 ([10, 4]) *Given a CA, it is decidable whether its language is empty, and whether its language is finite.*

With the closure properties of $\mathcal{L}_{\mathrm{UnCA}}$ of Proposition 5, this implies:

Proposition 16. *Given an UnCA, it is decidable whether its language is Σ^*. Given two UnCA, it is decidable whether the language of the first is included in the language of the second.*

The rest of this section is devoted to the main technical result of our paper, namely that it is decidable whether the language of an UnCA is regular. Our technique is in two steps: we first show that it is decidable whether a *bounded* CA language (given additionally a socle of the language) is regular (Lemma 20) then reduce the decision in the general case to the decision with bounded CA languages.

Definition 17. *A set C is* unary *if it is equal to a finite union of linear sets, each period of each linear set having at most one nonzero coordinate.*

Lemma 18 ([8]) *Let $L \subseteq w_1^* w_2^* \cdots w_n^*$. The language L is regular iff $\mathsf{Iter}_{(w_1,w_2,\ldots,w_n)}(L)$ is unary.*

Lemma 19 ([8]) *Given a semilinear set C, it is decidable whether C is unary.*

Lemma 20. *Given a CA (A, C) and words $w_1, w_2, \ldots, w_n$ such that $L(A, C)$ is a bounded language and $(w_1, w_2, \ldots, w_n)$ is one of its socles, it is decidable whether $L(A, C)$ is regular.*

Proof. Let (A, C) be a CA with $L(A, C) \subseteq w_1^* w_2^* \cdots w_n^*$. Let T be the set of new symbols $\{\ell_1, \ell_2, \ldots, \ell_n\}$ and define the morphism $h \colon \mathrm{T}^* \to \Sigma^*$ by $h(\ell_i) = w_i$ for all i. Now let (A', C') be the CA with language $h^{-1}(L(A, C)) \cap \ell_1^* \ell_2^* \cdots \ell_n^*$ obtained by the (effective) closures of CA. Then for $\mathbf{i} \in \mathbb{N}^n$, $\ell_1^{i_1} \ell_2^{i_2} \cdots \ell_n^{i_n} \in L(A', C')$ iff $w_1^{i_1} w_2^{i_2} \cdots w_n^{i_n} \in L(A, C)$. Hence:

$$\mathsf{Pkh}(L(A', C')) = \mathsf{Iter}_{(w_1,w_2,\ldots,w_n)}(L(A, C)) \ .$$

Now $\mathsf{Pkh}(L(A', C'))$ is a semilinear set that can be (effectively) obtained [10], and we may thus check whether it is unary using Lemma 19. This amounts to deciding, by Lemma 18, whether $L(A, C)$ is regular. □

Lemma 21. *Let* $U \subseteq \mathbb{N}^d$ *be a unary set. There exist* $s \geq 0$, $p \geq 1$ *and* $F \subseteq [0..s+p]^d$ *such that* $U = \bigcup_{\mathbf{f}\in F}[\mathbf{f}]_{s,p}$.

Proof. Set s to the maximum basis entry, over the bases of the linear sets L_j forming U, and p to the product of the nonzero entries occurring in the L_j periods. An F can be defined from the basis-and-periods representation of the L_js. □

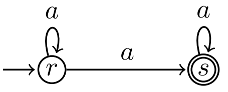
We continue with a lemma that allows us to focus on languages of *paths*:

Lemma 22. *The language of an UnCA* (A, C) *is regular iff* $\mathsf{Run}(A)\restriction_C$ *is regular.*

Proof. First, suppose $\mathsf{Run}(A)\restriction_C$ is regular, for a CA (A, C). As by definition $L(A, C) = \mu(\mathsf{Run}(A)\restriction_C)$ and regular languages are closed under morphisms, we have that $L(A, C)$ is regular. This part does not rely on unambiguity.

Second, consider an UnCA (A, C). We remark that if an accepting path of A is labeled by a word in $L(A, C)$, then it is in $\mathsf{Run}(A)\restriction_C$ (the converse is true of any CA). Indeed, since a path labeled by a word w in $L(A, C)$ is, by unambiguity, the only path labeled w in $\mathsf{Run}(A)$, it has its Parikh image in C. In other words, $\mathsf{Run}(A)\restriction_C = \mu^{-1}(L(A, C)) \cap \mathsf{Run}(A)$. Now, as the class of regular languages is closed under inverse morphisms and intersection, if $L(A, C)$ is regular then $\mathsf{Run}(A)\restriction_C$ is regular. □

Remark 23. *The inclusion* $\mathsf{Run}(A)\restriction_C \supseteq \mu^{-1}(L(A, C)) \cap \mathsf{Run}(A)$ *is crucial to the proof of Lemma 22 and to the decidability of regularity for UnCA. Indeed, both this inclusion and Lemma 22 fail for CA — in fact, regularity is undecidable for CA* [4]. *For example, let* A *be the automaton:*



Define C *to constrain the two loops on* r *and* s *to occur the same number of times. Then* $L(A, C) = \{a^{2n+1} \mid n \in \mathbb{N}\}$, *a regular language. But with* t_1, t_2, t_3 *the three transitions of* A, *from left to right,* $\mathsf{Run}(A)\restriction_C = \{t_1^n t_2 t_3^n \mid n \in \mathbb{N}\}$, *a nonregular language.*

An *elementary cycle* of A is a nonempty path starting and ending in the same state and with no other state appearing twice. Let $b_1, b_2, \ldots, b_\ell \in \delta^*$ be the elementary cycles of A. Recall that $\mathsf{Run}(A)$ is the set of initial runs of A that are accepting. Let $\mathsf{Init}(A)$ be the set of initial runs of A. Define the finite set

$$V = \{v \in \mathsf{Init}(A) : \forall i \forall j [(v = u b_i u' b_j u'') \Rightarrow (i \neq j)]\}$$

of runs with no two explicit occurrences of the same b_i. For $v = t_1 t_2 \cdots t_{|v|} \in V$ traversing the states $s_0, s_1, s_2, \ldots, s_{|v|}$ in A, let $B_v \subseteq \delta^*$ be the bounded language

$$b^*_{(s_0,1)} b^*_{(s_0,2)} \cdots b^*_{(s_0,\cdot)}\ t_1\ b^*_{(s_1,1)} \cdots b^*_{(s_1,\cdot)}\ t_2\ \cdots t_{|v|}\ b^*_{(s_{|v|},1)} \cdots b^*_{(s_{|v|},\cdot)} \tag{1}$$

where, for $0 \leq j \leq |v|$, $b_{(s_j,1)}, \ldots, b_{(s_j,\cdot)}$ lists the b_is rooted at s_j in any order. Define

$$\begin{aligned}\mathsf{socle}_v &= (b_{(s_0,1)}, \ldots, b_{(s_0,\cdot)}, t_1, b_{(s_1,1)}, \ldots, t_2, \ldots, t_{|v|}, b_{(s_{|v|},1)}, \ldots, b_{(s_{|v|},\cdot)}) \\ &= (w_{v,1}, w_{v,2}, \ldots, w_{v,d_v})\end{aligned}$$

for the appropriate $d_v \in \mathbb{N}$ and $w_{v,1}, w_{v,2}, \ldots, w_{v,d_v} \in \delta^*$. Fix $\mathbf{z}_v \in (\mathbb{N}^{|\delta|})^{d_v}$ as

$$\mathbf{z}_v = (\mathsf{Pkh}(w_{v,1}), \mathsf{Pkh}(w_{v,2}), \ldots, \mathsf{Pkh}(w_{v,d_v}))$$

and fix $\chi_v \in \{0,1\}^{d_v}$ as the d_v-tuple having 0 everywhere except at the $|v|$ positions in which the t_j forming v, $1 \leq j \leq |v|$, now occur in socle_v. Hence, slightly abusing the scalar product, $\mathbf{z}_v \cdot \chi_v = \mathsf{Pkh}(v) \in \mathbb{N}^{|\delta|}$. When $\mathbf{x} \in \mathbb{N}^{d_v}$, we let $\mathsf{socle}_v^{\mathbf{x}}$ stand for $w_{v,1}^{x_1} w_{v,2}^{x_2} \cdots, w_{v,d_v}^{x_{d_v}}$.

Let $w \in \mathsf{Init}(A)$. We write

$$w \Rightarrow w'$$

to mean that w' results from deleting the first explicit elementary cycle encountered for the second time as w is scanned from left to right, i.e., $w \Rightarrow w'$ if

$$\begin{aligned} w &= \underbrace{u\ \overbrace{t_1 t_2 \cdots t_{k-1} t_k}^{b_i}\ u'\ t_1 t_2 \cdots t_{k-1}}_{\in V}\ t_k u'' \\ w' &= u\ t_1 t_2 \cdots t_{k-1} t_k\ u' u'' \end{aligned}$$

for some unique $u, u', u'' \in \delta^*$ and $1 \leq i \leq \ell$. When $v \in V$, we write

$$w \overset{*}{\Rightarrow} v$$

to mean that zero or more applications of "$\Rightarrow$" lead from w to v; note then that every word w' in the derivation from w to v belongs to $\mathsf{Init}(A)$ and satisfies $\mathsf{States}(w) = \mathsf{States}(w') = \mathsf{States}(v)$.

Lemma 24. $\mathsf{Run}(A)\lceil_C$ *is regular iff* $(\forall v \in V)$ $[B_v\lceil_C \cap \mathsf{Run}(A)$ *is regular*$]$.

Proof. If $\mathsf{Run}(A)\lceil_C$ is regular, then $\forall v \in V$, $B_v \cap (\mathsf{Run}(A)\lceil_C) = B_v\lceil_C \cap \mathsf{Run}(A)$ is regular. Conversely, suppose that $\forall v \in V$, $B_v\lceil_C \cap \mathsf{Run}(A)$ is regular. For each $v \in V$, appealing to Lemma 18 and Lemma 21, fix $s_v \geq 0$, $p_v \geq 1$ and $F_v \subseteq [0..s_v + p_v]^{d_v}$ such that

$$\mathsf{Iter}_{\mathsf{socle}_v}(B_v\lceil_C \cap \mathsf{Run}(A)) = \bigcup_{\mathbf{f} \in F_v} [\mathbf{f}]_{s_v,p_v} \subseteq \mathbb{N}^{d_v}. \tag{2}$$

We will construct an NFA N fulfilling

$$(\forall w \in \mathsf{Run}(A))[\mathsf{Pkh}(w) \in C \text{ iff } N \text{ accepts } w], \tag{3}$$

which shows that $\mathsf{Run}(A)\lceil_C = \mathsf{Run}(A) \cap L(N)$ is regular.

The states of the NFA N are the initial state q_0 together with the states

$$(v, u, \mathbf{x}_v)$$

where $v, u \in V$ and $\mathbf{x}_v \in [0..s_v + p_v]^{d_v}$. The final states of N are the states

$$(v, v, \mathbf{x})$$

such that $\mathbf{x} \in [\mathbf{f}]_{s_v,p_v}$ for some $\mathbf{f} \in F_v$.

The transition function Δ of N is completely specified by

$$\Delta(q_0, \varepsilon) = \{(v, \varepsilon, \chi_v) \ : \ v \in V\}$$

$$\Delta((v, u, \mathbf{x}), t) = \begin{cases} \{(v, ut, \mathbf{x})\} & \text{if } ut \in V \\ \{(v, u', \mathbf{x} + \mathbf{e}_j^{d_v})\} & \text{else if } ut = u'b_i \text{ for any } j \text{ such that} \\ & w_{v,j} = b_i \text{ occurs as some } b_{(\cdot,\cdot)} \text{ in (1) for } v, \end{cases}$$

where $+$ sums in $\mathbb{N}^{d_v}/\equiv_{s_v,p_v}$ and $\mathbf{e}_j^{d_v} \in \mathbb{N}^{d_v}$ is the unit vector with jth entry 1.

The NFA on input $w \in \delta^*$ thus first chooses $v \in V$. This v is fixed until the end of the computation and the NFA is trying to ascertain that $w \overset{*}{\Rightarrow} v$. The absence of a j such that $w_{v,j} = b_i = b_{(\cdot,\cdot)}$ in the definition of Δ signals a bad choice for v and blocks the computation. The following invariant is easy to see: $\forall w \in \delta^*$ and states $(v, u, \mathbf{y})$ and $(v, u', \mathbf{x})$ with $\mathsf{socle}_v^{\mathbf{y}} \in B_v$,

$$\text{if } (v, u, \mathbf{y}) \xrightarrow{w} (v, u', \mathbf{x}) \text{ then } \mathsf{socle}_v^{\mathbf{x}} \in B_v \text{ and } (\exists \mathbf{a} \in [\mathbf{x}]_{s_v,p_v}) \\ [\mathbf{z}_v \cdot \mathbf{a} = \mathbf{z}_v \cdot \mathbf{y} + \mathsf{Pkh}(uw) - \mathsf{Pkh}(u')].$$

Claim: With "$\cdot$" meaning "don't care", the following holds $\forall w \in \delta^*$ and $\forall v, u \in V$ such that $uw \in \mathsf{Run}(A)$: if $uw \overset{*}{\Rightarrow} v$ then $(v, u, \cdot) \xrightarrow{w} (v, v, \cdot)$.
Proof that the claim implies (3): Let $w \in \mathsf{Run}(A)$. Suppose that $\mathsf{Pkh}(w) \in C$. Because "$\Rightarrow$" is always applicable to a word in $\mathsf{Init}(A) \setminus V$, there exists $v \in V$ such that $w \overset{*}{\Rightarrow} v$. Applying the claim with $u = \varepsilon$ and $\mathbf{y} = \chi_v$,

$$q_0 \xrightarrow{\varepsilon} (v, \varepsilon, \chi_v) \xrightarrow{w} (v, v, \mathbf{x})$$

for some $\mathbf{x} \in [0..s_v + p_v]^{d_v}$. By the invariant, some $\mathbf{a} \in [\mathbf{x}]_{s_v,p_v}$ satisfies $\mathbf{z}_v \cdot \mathbf{a} = \mathsf{Pkh}(w)$. Since $\mathsf{Pkh}(w) \in C$, $\mathbf{a} \in \mathsf{Iter}_{\mathsf{socle}_v}(B_v{\restriction}_C \cap \mathsf{Run}(A))$. By (2), $\mathbf{a} \in [\mathbf{f}]_{s_v,p_v}$ for $\mathbf{f} \in F_v$. But then $\mathbf{x} \equiv_{s_v,p_v} \mathbf{a} \equiv_{s_v,p_v} \mathbf{f}$, so $(v, v, \mathbf{x})$ is accepting and N accepts w.

Conversely, suppose that N accepts w. Then for some $v \in V$ and $\mathbf{f} \in F_v$,

$$q_0 \xrightarrow{\varepsilon} (v, \varepsilon, \chi_v) \xrightarrow{w} (v, v, \mathbf{x})$$

where $\mathbf{x} \in [\mathbf{f}]_{s_v,p_v}$. By the invariant, some $\mathbf{a} \in [\mathbf{x}]_{s_v,p_v}$ satisfies $\mathbf{z}_v \cdot \mathbf{a} = \mathsf{Pkh}(w)$. Since $\mathbf{a} \equiv_{s_v,p_v} \mathbf{x} \equiv_{s_v,p_v} \mathbf{f}$, $\mathbf{a} \in \mathsf{Iter}_{\mathsf{socle}_v}(B_v{\restriction}_C \cap \mathsf{Run}(A))$ by (2) and $\mathbf{z}_v \cdot \mathbf{a} \in C$. So $\mathsf{Pkh}(w) = \mathbf{z}_v \cdot \mathbf{a} \in C$. This completes the proof that the claim implies (3).
Proof of the claim: It remains to prove the claim, by induction on $|w|$. Let $|w| = 0$. For all $v, u \in V$, $uw = u \overset{*}{\Rightarrow} v$ implies $u = v$ since "$\Rightarrow$" is not applicable to $u \in V$. Hence $(v, u, \cdot) \xrightarrow{\varepsilon} (v, v, \cdot)$. For the induction step, let $w = tw' \in \delta^*$ for some $t \in \delta$. Let $v, u \in V$. Suppose that $utw' \in \mathsf{Run}(A)$. Let $utw' \overset{*}{\Rightarrow} v$. If $ut \in V$, then

$$(v, u, \cdot) \xrightarrow{t} (v, ut, \cdot) \xrightarrow{w'} (v, v, \cdot)$$

where $\xrightarrow{w'}$ uses induction. If $ut \notin V$, then $ut = u'b_i$ for some i. Since $\mathsf{States}(utw') = \mathsf{States}(v)$, the initial state of b_i is traversed by v. Hence b_i occurs as some $b_{(\cdot,\cdot)}$ listed in (1) for v, so j exists such that $w_{v,j} = b_i = b_{(\cdot,\cdot)}$. Then

$$(v, u, \cdot) \xrightarrow{t} (v, u', \cdot) \xrightarrow{w'} (v, v, \cdot)$$

where $\xrightarrow{w'}$ again follows by induction since $u' \preceq u \in V$, so $u' \in V$, but now also subtly appealing to the unique decomposition of $utw' \overset{*}{\Rightarrow} v$ as $utw' \Rightarrow u'w' \overset{*}{\Rightarrow} v$. In both cases, $(v, u, \cdot) \xrightarrow{w} (v, v, \cdot)$ as required, completing the inductive step, proving the claim and proving the lemma. □

Theorem 25. *Regularity for UnCA is decidable.*

Proof. Let an UnCA (A, C) with transition set δ be given. An upper bound r on the length of words in δ^* having no repeated elementary cycle is easily computed. Then the finite set V can be computed, by examining every word in δ^* of length at most r. It remains to check the condition from Lemma 24 that for each $v \in V$, the language $B_v\!\restriction_C \cap \mathsf{Run}(A)$ is regular, as follows. Let D be a DFA for B_v. The language $B_v\!\restriction_C$ is that of the CA (D, C') where C' checks that the accepted word has a Parikh image in C. Since $\mathsf{Run}(A)$ is regular, another CA has the language $B_v\!\restriction_C \cap \mathsf{Run}(A)$. The latter is a bounded CA language for which we have a socle socle_v. By Lemma 20, checking the regularity of this language is decidable. □

A DetCA is an UnCA; moreover, DetCA are effectively equivalent [10] to deterministic extended automata over $(\mathbb{Z}^k, +, \mathbf{0})$ (defined in [11]). Thus:

Corollary 26. *Given a DetCA or an extended automaton over* $(\mathbb{Z}^k, +, \mathbf{0})$*, it is decidable whether its language is regular.*

6. A deterministic form of UnCA

We present a *deterministic* model equivalent to UnCA. This model is a restriction of the affine Parikh automaton [4] and can be seen as a simple register automaton. As a result of independent interest, we show that CA are equivalent to the nondeterministic variant of this model, and that a seemingly more powerful model (so-called *finite-monoid affine Parikh automata* [3]) is in fact equivalent to CA (resp. UnCA) in its nondeterministic (resp. deterministic) form.

Definition 27 (Affine Parikh automaton [4]) *An* affine Parikh automaton (APA) *of dimension d is a triple (A, U, C) where A is an automaton with transition set δ, $U\colon \delta^* \to \mathcal{F}_d$ is a morphism, and $C \subseteq \mathbb{N}^d$ is semilinear. Its language is $L(A, U, C) = \mu_A(\{\pi \in \mathsf{Run}(A) \mid U_\pi(\mathbf{0}) \in C\})$. The APA is said to be:*

- Deterministic (DetAPA) *if A is deterministic;*
- Finite-monoid (FM-APA, FM-DetAPA) [3] *if $\mathcal{M}(U)$ is finite;*

- Moving (M-APA, M-DetAPA) *if for all* $t \in \delta$, $U_t = (M, \mathbf{v})$ *is such that* M *is a 0-1-matrix with exactly one* 1 *per row.*

We consider only FM- and M-(Det)APA in the present work. We write $\mathcal{L}_{\text{FM-APA}}$*,* $\mathcal{L}_{\text{FM-DetAPA}}$*,* $\mathcal{L}_{\text{M-APA}}$*, and* $\mathcal{L}_{\text{M-DetAPA}}$ *for the classes of languages recognized by FM-APA, FM-DetAPA, M-APA, and M-DetAPA respectively.*

Remark 28. *An M-(Det)APA of dimension d can be seen as a finite-state* (deterministic) register automaton *with d registers* $r_1, r_2, \ldots, r_d$*: each transition performs actions of the type* $r_i \leftarrow r_{j_i} + k_i$*, with* $k_i \in \mathbb{N}$*,* $1 \leq j_i \leq d$*, for* $1 \leq i \leq d$*, and the device accepts iff the underlying automaton accepts and the values of the registers at the end of the computation belong to a prescribed semilinear set.*

Theorem 29. $\mathcal{L}_{\text{CA}} = \mathcal{L}_{\text{M-APA}} = \mathcal{L}_{\text{FM-APA}}$.

Proof. ($\mathcal{L}_{\text{CA}} \subseteq \mathcal{L}_{\text{M-APA}}$.) Given a CA (A, C) where $A = (Q, \Sigma, \delta, q_0, F)$ and $\delta = \{t_1, t_2, \ldots, t_n\}$, we define an M-APA (A, U, C) by setting, for all $t_i \in \delta$, $U_{t_i}(\mathbf{x}) = \mathbf{x} + \mathsf{Pkh}(t_i)$. For a path $\pi \in \delta^*$, we have that $U_\pi(\mathbf{0}) = \mathsf{Pkh}(\pi)$. This implies that $L(A, U, C) = \mu(\{\pi \in \mathsf{Run}(A) \mid \mathsf{Pkh}(\pi) \in C\}) = L(A, C)$, and moreover, that $U_t = (M, \mathbf{v})$ is such that M is the identity matrix, thus (A, U, C) is an M-APA.

($\mathcal{L}_{\text{M-APA}} \subseteq \mathcal{L}_{\text{FM-APA}}$.) Composing 0-1-matrices with exactly one 1 per row results in the same type of matrices. Thus the multiplicative $\mathcal{M}(U)$ of an M-APA (A, U, C) is finite, i.e., (A, U, C) is an FM-APA.

($\mathcal{L}_{\text{FM-APA}} \subseteq \mathcal{L}_{\text{CA}}$.) Let (A, U, C) be an FM-APA, where $A = (Q, \Sigma, \delta, q_0, F)$. For $t \in \delta$, we write $U_t = (M_t, \mathbf{v}_t)$, and for $t_1 t_2 \cdots t_n \in \delta^+$, we let $M_{t_1 t_2 \cdots t_n} = M_{t_n} \cdots M_{t_2} \cdot M_{t_1}$. As it is consistent to do, we set $M_\varepsilon = Id$, the identity matrix. We show that $L(A, U, C)$ can be expressed as the union of the languages of a finite number of CA, and that those CA are unambiguous if A is deterministic. We work in 3 steps. (1.) We devise a finite set of automata and show that they recognize the runs π on A while "knowing" M_π (Fact 30). (2.) We show that this extra knowledge allows for the extraction of $U_\pi(\mathbf{0})$ when π is read (Fact 31). We design a semilinear set to constrain this extracted value by C. (3.) We conclude that replacing the labels t of those CA by $\mu_A(t)$ gives a finite set of CA recognizing $L(A, U, C)$.

Step 1: Automata for the Paths of A. The simplest way to construct an automaton for $\mathsf{Run}(A)$ is by replacing the label of each transition t of A by t itself, i.e., we obtain the automaton $(Q, \delta, \Delta, q_0, F)$ where $t = q \bullet\!\!-\ell\!\!\rightarrow q' \in \delta \Leftrightarrow q \bullet\!\!-t\!\!\rightarrow q' \in \Delta$. This is the first idea of the present construction. The second idea is that we want, when in a state q, all the possible M_π's for π accepted from q to be the same. Write $\mathcal{M} = \mathcal{M}(U)$. We define, for $q \in Q$ and $M \in \mathcal{M}$, $B^{\rightarrow(q,M)} = (Q \times \mathcal{M},\ \delta,\ \Delta,\ (q, M),\ F \times \{M_\varepsilon\})$, where $\Delta = \{(q, M) \bullet\!\!-t\!\!\rightarrow (q', M') \mid t = q \bullet\!\!-\mu(t)\!\!\rightarrow q' \in \delta \wedge M'.M_t = M\}$.

It is important to note that even if A is deterministic, $B^{\rightarrow(q,M)}$ may not be deterministic. Indeed, let Z be the all-zero matrix, and suppose that, for some $t \in \delta$, $M_t = Z$. Then *any* matrix M' verifies $M'.M_t = Z$, thus from the state $(\mathsf{From}(t), Z)$ there is a transition labeled t to *any* state $(\mathsf{To}(t), M')$ for $M' \in \mathcal{M}$. We now show that these automata indeed recognize the paths π in A, while "knowing" M_π. Write $A^{\rightarrow q}$ for A where the initial state is set to q, then:

Fact 30. *For any* $q \in Q$ *and* $M \in \mathcal{M}$, $L(B^{\rightarrow(q,M)}) = \{\pi \in \mathsf{Run}(A^{\rightarrow q}) \mid M_\pi = M\}$. *In particular,* $\mathsf{Run}(A) = \bigcup_{M \in \mathcal{M}} L(B^{\rightarrow(q_0,M)})$.

Step 2: Retrieving $U_\pi(\mathbf{0})$. In this step, we argue that our previous construction helps in retrieving the value of $U_\pi(\mathbf{0})$ when π is read over some $B^{\rightarrow(q,M)}$. The main ingredient is the following simple property: for $t \in \delta$ and $\rho \in \delta^*$, $U_{t\rho}(\mathbf{0}) = M_\rho.\mathbf{v}_t + U_\rho(\mathbf{0})$. We now show a property on paths *over* $B^{\rightarrow(q,M)}$. First, identify Δ with $\{T_1, T_2, \ldots, T_n\}$, and each T_i with $(q_i, M_i)\bullet\!\!-t_i\!\!\rightarrow(q'_i, M'_i)$; next, write μ_B for the μ function of one of the $B^{\rightarrow(q,M)}$'s — this morphism is independent of (q, M). Then:

Fact 31. *For any* $q \in Q$, $M \in \mathcal{M}$, *and* $\Pi \in \mathsf{Run}(B^{\rightarrow(q,M)})$, *we have* $U_{\mu_B(\Pi)}(\mathbf{0}) = \sum_{i=1}^{n} |\Pi|_{T_i} \times (M'_i.\mathbf{v}_{t_i})$.

Now define $C' \subseteq \mathbb{N}^n$ by $(x_1, x_2, \ldots, x_n) \in C' \Leftrightarrow (\sum_{i=1}^{n} x_i \times (M'_i.\mathbf{v}_{t_i})) \in C$. Fact 30 and Fact 31 imply that, for $q \in Q$ and $M \in \mathcal{M}$, $L(B^{\rightarrow(q,M)}, C') = \{\pi \in \mathsf{Run}(A^{\rightarrow q}) \mid M_\pi = M \wedge U_\pi(\mathbf{0}) \in C\}$.

Step 3: from Paths to their Labels. For $q \in Q$ and $M \in \mathcal{M}$, define $D^{\rightarrow(q,M)}$ to be the automaton $B^{\rightarrow(q,M)}$ where a transition labeled t in $B^{\rightarrow(q,M)}$ is relabeled $\mu_A(t)$ in $D^{\rightarrow(q,M)}$. Then $L(D^{\rightarrow(q,M)}, C') = \mu_A(L(B^{\rightarrow(q,M)}, C'))$. Since $\mathsf{Run}(A) = \bigcup_{M \in \mathcal{M}} B^{\rightarrow(q_0,M)}$, this implies that $L(A, U, C) = \bigcup_{M \in \mathcal{M}} L(D^{\rightarrow(q_0,M)}, C')$. As $\mathcal{M}$ is finite by hypothesis, $L(A, U, C)$ is the finite union of CA languages. The closure of $\mathcal{L}_{\text{CA}}$ under union [10] implies that $L(A, U, C) \in \mathcal{L}_{\text{CA}}$. ☐

Theorem 32. $\mathcal{L}_{\text{UnCA}} = \mathcal{L}_{\text{M-DetAPA}} = \mathcal{L}_{\text{FM-DetAPA}}$.

Proof (sketch). $\mathcal{L}_{\text{UnCA}} \subseteq \mathcal{L}_{\text{M-DetAPA}}$ is shown in [3]; $\mathcal{L}_{\text{M-DetAPA}} \subseteq \mathcal{L}_{\text{FM-DetAPA}}$ is immediate. For $\mathcal{L}_{\text{FM-DetAPA}} \subseteq \mathcal{L}_{\text{UnCA}}$, we add a step to the proof of the inclusion $\mathcal{L}_{\text{FM-APA}} \subseteq \mathcal{L}_{\text{CA}}$ of Theorem 29. We note, using the same notations, that if A is deterministic, then for any $q \in Q$ and $M \in \mathcal{M}$, $D^{\rightarrow(q,M)}$ is unambiguous. $\mathcal{L}_{\text{UnCA}}$ being closed under union (Proposition 4) this proves the inclusion. ☐

Remark 33. *Theorems 29 and 32 are* effective, *in the sense that one can go from one model to another following an algorithm. This implies in particular, together with Theorem 25, that regularity is decidable for FM-DetAPA; we note that it is not decidable for DetAPA* [3], *which describes a class of languages strictly larger than that of UnCA though expected to be incomparable with that of CA.*

7. Conclusion

We showed that $\mathcal{L}_{\text{UnCA}}$ is a class of languages that is closed under the Boolean operations, inverse morphisms, commutative closure, reversal, and right and left quotient, and that provably fails to be closed under concatenation with a regular language, length-preserving morphisms, and starring. Further, the following problems are decidable for $\mathcal{L}_{\mathrm{UnCA}}$: emptiness, universality, finiteness, inclusion, and regularity. Deciding regularity for UnCA and DetCA is our main result.

We propose three future research avenues. First, the properties of UnCA indicate its suitability for model-checking, and we could envisage real-world applications of verification using UnCA. Second, we translated unambiguous CA to a natural model of *deterministic* register automata; the close inspection of this translation can lead to further advances in our understanding of unambiguity, in particular in the open problems dealing with unambiguous finite automata [5]. Third, we note that the closure properties of $\mathcal{L}_{\mathrm{UnCA}}$ imply that this class can be described by a natural algebraic object (see [1]). This will certainly help in linking UnCA to a first-order logic framework, and thus to Boolean circuit complexity. Hence we hope that UnCA can play a role in the study of complexity classes such as NC^1.

Acknowledgement. We thank Andreas Krebs for stimulating discussions and comments concerning this work and the anonymous referees for their careful reading. The first author thanks Benno Salwey and Dave Touchette for comments on early versions of this paper.

References

[1] C. Behle, A. Krebs and S. Reifferscheid, Typed monoids: an Eilenberg-like theorem for non regular languages, *CAI*, (Springer-Verlag, 2011), pp. 97–114.

[2] A. Bouajjani and P. Habermehl, Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations, *Theoretical Computer Science* **221**(1–2) (1999) 211–250.

[3] M. Cadilhac, A. Finkel and P. McKenzie, Bounded Parikh automata, *WORDS*, (2011), pp. 93–102.

[4] M. Cadilhac, A. Finkel and P. McKenzie, Affine Parikh automata, *RAIRO - Theoretical Informatics and Applications* **46**(04) (2012) 511–545.

[5] T. Colcombet, Forms of determinism for automata, *STACS*, (2012), pp. 1–23.

[6] S. Ginsburg and E. Spanier, Semigroups, Presburger formulas and languages, *Pacific Journal of Mathematics* **16**(2) (1966) 285–296.

[7] S. Ginsburg and E. H. Spanier, Bounded ALGOL-like languages (1964).

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

[9] O. H. Ibarra, Reversal-bounded multicounter machines and their decision problems, *J. ACM* **25**(1) (1978) 116–133.

[10] F. Klaedtke and H. Rueß, Monadic second-order logics with cardinalities, *ICALP*, *LNCS* **2719**, (Springer-Verlag, 2003), pp. 681–696.

[11] V. Mitrana and R. Stiebe, Extended finite automata over groups, *Discrete Appl. Math.* **108**(3) (2001) 287–300.

[12] R. J. Parikh, On context-free languages, *Journal of the ACM* **13**(4) (1966) 570–581.