Ultimate periodicity of b-recognisable sets : a quasilinear procedure

Victor Marsault⋆ and Jacques Sakarovitch

Telecom-ParisTech and CNRS, 46 rue Barrault 75013 Paris, France

Abstract It is decidable if a set of numbers, whose representation in a base b is a regular language, is ultimately periodic. This was established by Honkala in 1986.
We give here a structural description of minimal automata that accept an ultimately periodic set of numbers. We then show that it can be verified in linear time if a given minimal automaton meets this description.
This yields a $O(n \log(n))$ procedure for deciding whether a general deterministic automaton accepts an ultimately periodic set of numbers.

1 Introduction

Given a fixed positive integer b, called the *base*, every positive integer n is represented (in base b) by a *word* over the digit alphabet $A_b = \{0, 1, \ldots, b-1\}$ which does not start with a 0. Hence, *sets* of numbers are represented by *languages* of ${A_b}^*$. Depending on the base, a given set of integers may be represented by a simple or complex language: the set of powers of 2 is represented by the rational language 10^* in base 2; whereas in base 3, it can only be represented by a context-sensitive language, much harder to describe.

A set of numbers is said to be *b-recognisable*if it is represented by a recognisable, or rational, or regular, language over ${A_b}^*$. On the other hand, a set of numbers is *recognisable* if it is, via the identification of $\mathbb{N}$ with a^* ($n \leftrightarrow a^n$), a recognisable, or rational, or regular, language of the free monoid a^*. A set of numbers is recognisable if, and only if it is *ultimately periodic* (UP) and we use the latter terminology in the sequel as it is both meaningful and more distinguishable from b-recognisable. It is common knowledge that every UP-set of numbers is b-recognisable for every b, and the above example shows that a b-recognisable set for some b is not necessarily UP, nor c-recognisable for all c. It is an exercice to show that if b and c are *multiplicatively dependent* integers (that is, there exist integers k and l such that $b^k = c^l$), then every b-recognisable set is a c-recognisable set as well (*cf.* [9] for instance). A converse of these two properties is the theorem of Cobham: *a set of numbers which is both b- and c-recognisable, for multiplicatively independent b and c, is UP*, established in 1969 [5], a strong and deep result whose proof is difficult (*cf.* [4]).

After Cobham's theorem, the next natural (and last) question left open on b-recognisable sets of numbers was the decidability of ultimate periodicity. It was positively solved in 1986:

⋆ Corresponding author, victor.marsault@telecom-paristech.fr

Theorem 1 (Honkala [11]). *It is decidable whether an automaton over* ${A_b}^*$ *accepts a UP-set of numbers.*

The complexity of the decision procedure was not an issue in the original work. Neither were the properties or the structure of automata accepting UP-set of numbers. Given an automaton $\mathcal{A}$ over ${A_b}^*$, bounds are computed on the parameters of a potential UP-set of numbers accepted by $\mathcal{A}$. The property is then decidable as it is possible to enumerate all automata that accept sets with smaller parameters and check whether any of them is equivalent to $\mathcal{A}$.

As explained below, subsequent works on automata and number representations brought some answers regarding the complexity of the decision procedure, explicitly or implicitly. The present paper addresses specifically this problem and yields the following statement.

Theorem 2. *It is decidable* in linear time *whether a minimal DFA* $\mathcal{A}$ *over* ${A_b}^*$ *accepts a UP-set of numbers.*

As it is often the case, this complexity result is obtained as the consequence of a structural characterisation. Indeed, we describe here a set of structural properties for an automaton: the shape of its strongly connected components (SCC's) and that of its graph of SCC's, that we gather under the name of UP-criterion. Theorem 2 then splits into two results:

Theorem 3. *A minimal DFA* $\mathcal{A}$ *over* ${A_b}^*$ *accepts a UP-set of numbers if, and only if, it satisfies the UP-criterion.*

Theorem 4. *It is decidable in linear time whether a minimal DFA* $\mathcal{A}$ *over* ${A_b}^*$ *satisfies the UP-criterion.*

As for Cobham's theorem (*cf.* [4,7]), new insights on the problem tackled here are obtained when stating it in a higher dimensional space. Let $\mathbb{N}^d$ be the additive monoid of d-tuples of integers. Every d-tuple of integers may be represented in base b by a d-tuple of words of ${A_b}^*$ of *the same length*, as shorter words can be padded by 0's without changing the corresponding value. Such d-tuples can be read by (finite) automata over $({A_b}^d)^*$ — automata reading on d synchronised tapes — and a subset of $\mathbb{N}^d$ is b-recognisable if the set of the b-representations of its elements is accepted by such an automaton.

On the other hand, recognisable and rational sets of $\mathbb{N}^d$ are defined in the classical way but they do not coincide as $\mathbb{N}^d$ *is not a free monoid*. A subset of $\mathbb{N}^d$ is *recognisable* if is saturated by a congruence of finite index, and the family of recognisable sets is denoted by $\operatorname{Rec}\mathbb{N}^d$. A subset of $\mathbb{N}^d$ is *rational* if is denoted by a rational expression, and the family of rational sets is denoted by $\operatorname{Rat}\mathbb{N}^d$. Rational sets of $\mathbb{N}^d$ have been characterised by Ginsburg and Spanier as sets definable in the *Presburger arithmetic* $\langle\, \mathbb{N}, + \,\rangle$ ([10]), hence the name *Presburger definable* that is most often used in the literature.

It is also common knowledge that every rational set of $\mathbb{N}^d$ is b-recognisable for every b, and the example in dimension 1 is enough to show that a b-recognisable set is not necessarily rational. The generalisation of Cobham's theorem: *a subset* *of* $\mathbb{N}^d$ *which is both b- and c-recognisable, for multiplicatively independent b and c, is rational*, is due to Semenov (*cf.* [4,7]). The generalisation of Honkala's theorem went as smoothly.

Theorem 5 (Muchnik [15]). *It is decidable whether a b-recognisable subset of* $\mathbb{N}^d$ *is rational.*

Theorem 6 (Leroux [13]). *It is decidable* in polynomial time *whether a minimal DFA* $\mathcal{A}$ *over* $({A_b}^d)^*$ *accepts a rational subset of* $\mathbb{N}^d$.

The algorithm underlying Theorem 5 is triply exponential whereas the one described in [13], based on sophisticated geometric constructions, is quadratic — an impressive improvement — but not easy to explain.

There exists another way to devise a proof for Honkala's theorem which yields another extension. In [10], Ginsburg and Spanier also proved that there exists a formula in Presburger arithmetic deciding whether a given subset of $\mathbb{N}^d$ is recognisable. In dimension 1, it means that being a UP-set of numbers is expressible in Presburger arithmetic. In [2], it was then noted that since addition in base p is realised by a finite automaton, every Presburger formula is realised by a finite automaton as well. Hence a decision procedure that establishes Theorem 1.

Generalisation of base p by non-standard numeration systems then gives an extension of Theorem 1, best expressed in terms of abstract numeration systems. Given a totally ordered alphabet A, any *rational* language L of A^* defines an *abstract numeration system* (ANS) $\mathcal{S}_L$ in which the integer n is represented by the $n{+}1$-th word of L in the radix ordering of A^* (*cf.* [12]). A set of integers whose representations in the ANS $\mathcal{S}_L$ form a rational language is called $\mathcal{S}_L$-recognisable and it is known that every UP-set of numbers is $\mathcal{S}_L$-recognisable for every ANS $\mathcal{S}_L$ ([12]). The next statement then follows.

Theorem 7. *If* $\mathcal{S}_L$ *is an abstract numeration system in which addition is realised by a finite automaton, then it is decidable whether a* $\mathcal{S}_L$*-recognisable set of numbers is UP.*

For instance, Theorem 7 implies that ultimate periodicity is decidable for sets of numbers represented by rational sets in a Pisot base system [8]. The algorithm underlying Theorem 7 is exponential (if the set of numbers is given by a DFA) and thus (much) less efficient than Leroux's constructions for integer base systems. On the other hand, it applies to a much larger family of numeration systems. All this was mentioned for the sake of completeness, and the present paper does not follow this pattern.

Theorem 6, restricted to dimension 1, readily yields a quadratic procedure for Honkala's theorem. The improvement from quadratic to quasilinear complexity achieved in this article is not a natural simplification of Leroux's construction for the case of dimension 1. Although the UP-criterion bears similarities with some features of Leroux's construction, it is not derived from [13], nor is the proof of quasilinear complexity.

The paper is organised as follows. In Section 2, we treat the special case of determining whether a given minimal group automaton accepts an ultimately periodic set of numbers. We describe canonical automata, which we call Pascal automata, that accept such sets. We then show how to decide in linear time whether a given minimal group automaton is the quotient of some Pascal automaton.

Section 3 introduces the UP-criterion and sketches both its completeness and correctness. An automaton satisfying the UP-criterion is a directed acyclic graph (DAG) 'ending' with at most two layers of non-trivial strongly connected components (SCC's). If the root is seen at the top, the upper (non-trivial) SCC's are circuits of 0's and the lower ones are quotients of Pascal automata. It is easy, and of linear complexity to verify that an automaton has this overall structure.

Proofs have been consistently removed in order to comply with space constraints. A full version of this work is available on arXiv [14].

2 The Pascal automaton

2.1 Preliminaries

On automata We consider only finite deterministic finite automata, denoted by $\mathcal{A} = \langle Q, A, \delta, i, T \rangle$, where Q is the set of *states*, i the *initial state* and T the set of *final states*; A is the *alphabet*, A^* is the *free monoid* generated by A and the *empty word* is denoted by ε; $\delta : Q \times A \to Q$ is the *transition function*.

As usual, δ is extended to a function $Q \times A^* \to Q$ by $\delta(q, \varepsilon) = q$ and $\delta(q, ua) = \delta(\delta(q, u), a)$; and $\delta(q, u)$ will also be denoted by $q \cdot u$. When δ is a total function, $\mathcal{A}$ is said to be *complete*. In the sequel, we only consider automata that are *accessible*, that is, in which every state is reachable from i.

A word u of A^* is *accepted* by $\mathcal{A}$ if $i \cdot u$ is in T. The set of words accepted by $\mathcal{A}$ is called *the* language of $\mathcal{A}$, and is denoted by $L(\mathcal{A})$.

Let $\mathcal{A} = \langle Q, A, \delta, i, T \rangle$ and $\mathcal{B} = \langle R, A, \eta, j, S \rangle$ be two deterministic automata. A map $\varphi : Q \to R$ is an *automaton morphism*, written $\varphi : \mathcal{A} \to \mathcal{B}$ if $\varphi(i) = j$, $\varphi(T) \subseteq S$, and for all q in Q and a in A, such that $\delta(q, a)$ is defined, then $\eta(\varphi(q), a)$ is defined, and $\varphi(\delta(q, a)) = \eta(\varphi(q), a)$. We call φ a *covering* if the following two conditions hold: i) $\varphi(T) = S$ and ii) for all q in Q and a in A, if $\eta(\varphi(q), a)$ is defined, then so is $\delta(q, a)$. In this case, $L(\mathcal{A}) = L(\mathcal{B})$, and $\mathcal{B}$ is called *a quotient* of $\mathcal{A}$. Note that if $\mathcal{A}$ is complete, every morphism satisfies (ii).

Every complete deterministic automaton $\mathcal{A}$ has a *minimal quotient* which is *the minimal automaton* accepting $L(\mathcal{A})$. This automaton is unique up to isomorphism and can be computed from $\mathcal{A}$ in $O(n\,log(n))$ time, where n is the number of states of $\mathcal{A}$ (*cf.* [1]).

Given a deterministic automaton $\mathcal{A}$, every word u induces an application $(q \mapsto q \cdot u)$ over the state set. These applications form a finite monoid, called the *transition monoid* of $\mathcal{A}$. When this monoid happens to be a group (meaning that the action of every letter is a permutation over the states), $\mathcal{A}$ is called a *group automaton*.

On numbers The base b is fixed throughout the paper (it will be a *parameter* of the algorithms, not an input) and so is the digit alphabet A_b. As a consequence, the number of transitions of any deterministic automaton over $A_b{}^*$ is linear in its number of states. Verifying that an automaton is deterministic (resp. a group automaton) can then be done in linear time.

For our purpose, it is far more convenient to write the integers *least significant digits first* (LSDF), and to keep the automata reading *from left to right* (as in Leroux's work [13]). The *value* of a word $u = a_0 a_1 \cdots a_n$ of $A_b{}^*$, denoted by $\overline{u}$, is then $\overline{u} = \sum_{i=0}^{n}(a_i b^i)$ and may be obtained by the recursive formula:

$$\overline{u\,a} = \overline{u} + a\,b^{|u|} \qquad (1)$$

Conversely, every integer n has a unique canonical representation in base b that does not *end* with 0, and is denoted by $\langle n \rangle$. A word of $A_b{}^*$ has value n if, and only if, it is of the form $\langle n \rangle\, 0^k$.

By abuse of language, we may talk about the *set of numbers* accepted by an automaton. An integer n is then accepted if there exists a word of value n accepted by the automaton.

A set $E \subseteq \mathbb{N}$ is *periodic*, of *period* q, if there exists $S \subseteq \{0, 1, \ldots, q-1\}$ such that $E = \{n \in \mathbb{N} \,|\, \exists r \in S \quad n \equiv r\,[q]\}$. Any periodic set E has a *smallest* period p and a corresponding set of *residues* R: the set E is then denoted by E_p^R. The set of numbers in E_p^R and larger than an integer m is denoted by $E_{p,m}^R$.

2.2 Definition of a Pascal automaton

We begin with the construction of an automaton $\mathcal{P}_p^R$ that accepts the set E_p^R, in the case where

$$p \textit{ is coprime with } b.$$

We call any such automaton a *Pascal automaton.*[1] If p is coprime with b, there exists a (smallest positive) integer ψ such that:

$$b^\psi \equiv 1\,[p] \qquad \text{and thus} \qquad \forall x \in \mathbb{N} \qquad b^x \equiv b^{x \bmod \psi}\,[p] \enspace .$$

Therefore, from Equation (1), knowing $\overline{u} \bmod p$ and $|u| \bmod \psi$ is enough to compute $\overline{u\,a} \bmod p$.

Hence the definition of $\mathcal{P}_p^R = \langle\, \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}, A_b, \eta, (0,0), R \times \mathbb{Z}/\psi\mathbb{Z} \,\rangle$, where

$$\forall (s,t) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}\,,\ \forall a \in A_b \qquad \eta((s,t),a) = (s,t)\cdot a = (s + a\,b^t, t+1) \quad (2)$$

By induction on $|u|$, it follows that $(0,0)\cdot u = (\overline{u} \bmod p, |u| \bmod \psi)$ for every u in $A_b{}^*$ and consequently that E_p^R is the set of number accepted by $\mathcal{P}_p^R$.

Example 1. Fig. 1 shows $\mathcal{P}_3^2$, the Pascal automaton accepting integers written in binary and congruent to 2 modulo 3. For clarity, the labels are omitted; transitions labelled by 1 are drawn with thick lines and those labelled by 0 with thin lines.

---

[1] As early as 1654, Pascal describes a computing process that generalises the casting out nines and that determines if an integer n, written *in any base* b, is divisible by an integer p (see [16, Prologue]).

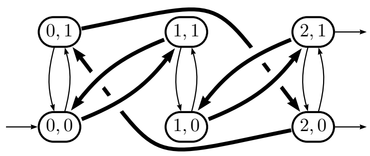

Figure 1: The Pascal automaton $\mathcal{P}_3^2$

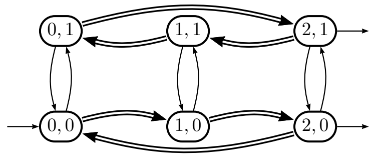

Figure 2: The modified Pascal automaton ${\mathcal{P}'}_3^2$

2.3 Recognition of quotients of Pascal automata

The tricky part of achieving a linear complexity for Theorem 4 is contained in the following statement:

Theorem 8. *It is decidable in linear time whether a minimal DFA $\mathcal{A}$ over A_b is the quotient of a Pascal automaton.*

Simplifications Since $\mathcal{P}_p^R$ is a group automaton, all its quotients are group automata.

The permutation on $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$ realised by $0^{(\psi-1)}$ is the inverse of the one realised by 0 and we call it the action of the "digit" 0^{-1}. Let g be a new letter whose action on $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$ is the one of 10^{-1}. It follows from (2) that for every a in A_b — where a is understood both as a *digit* and as a *number* — the action of a on $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$ (in $\mathcal{P}_p^R$) is equal to the one of g^a0. The same relation holds in any group automaton $\mathcal{A}$ over $A_b{}^*$ that is a quotient of a Pascal automaton, and this condition is tested in linear time.

Let $B = \{0, g\}$ be a new alphabet. Any group automaton $\mathcal{A} = \langle\, Q, A_b, \delta, i, T \,\rangle$ may be transformed into an automaton $\mathcal{A}' = \langle\, Q, B, \delta', i, T \,\rangle$ where, for every q in Q, $\delta'(q, 0) = \delta(q, 0)$ and $\delta'(q, g) = \delta(q, 10^{-1})$. Fig. 2 shows ${\mathcal{P}'}_3^2$ where transitions labelled by 0 are drawn with thin lines and those labelled by g with double lines.[2]

Analysis: computation of the parameters From now on, and for the rest of the section, $\mathcal{A} = \langle\, Q, A_b, \delta, i, T \,\rangle$ is a group automaton which has been consistently transformed into an automaton $\mathcal{A}' = \langle\, Q, B, \delta', i, T \,\rangle$. If $\mathcal{A}'$ is a quotient of a Pascal automaton ${\mathcal{P}'}_p^R$, then the parameters p and R may be computed (or 'read') in $\mathcal{A}'$; this is the consequence of the following statement.

Proposition 1. *Let $\varphi\colon {\mathcal{P}'}_p^R \to \mathcal{A}'$ be a covering. Then, for every (x, y) and (x', y') in $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$, if $x \neq x'$ and $\varphi(x, y) = \varphi(x', y')$, then $y \neq y'$.*

Corollary 1. *If $\mathcal{A}' = \langle\, Q, B, \delta', i, T \,\rangle$ is a quotient of a modified Pascal automaton ${\mathcal{P}'}_p^R$, then p is the length of the g-circuit in $\mathcal{A}'$ which contains i and $R = \{r \mid i \cdot g^r \in T\}$.*

[2] The transformation highlights that the transition monoid of $\mathcal{P}_p^R$ (and thus of ${\mathcal{P}'}_p^R$) is the *semi-direct product* $\mathbb{Z}/p\mathbb{Z} \rtimes \mathbb{Z}/\psi\mathbb{Z}$.

Next, if $\mathcal{A}'$ is a quotient of a (modified) Pascal automaton ${\mathcal{P}'}_p^R$, the equivalence class of the initial state of ${\mathcal{P}'}_p^R$ may be 'read' as well in $\mathcal{A}'$ as the intersection of the 0-circuit and the g-circuit around the initial state of $\mathcal{A}'$. More precisely, and since $(0,0) \xrightarrow[{\mathcal{P}'}_p^R]{g^s} (s,0) \xrightarrow[{\mathcal{P}'}_p^R]{0^t} (s,t)$, the following holds.

Proposition 2. *Let* $\varphi\colon {\mathcal{P}'}_p^R \to \mathcal{A}'$ *be a covering. For all* s *in* $\mathbb{Z}/p\mathbb{Z}$ *and* t *in* $\mathbb{Z}/\psi\mathbb{Z}$*,* $\varphi(s,t) = \varphi(0,0)$ *if, and only if,* $i \cdot g^s = i \cdot 0^{-t}$ *.*

From this proposition follows that, given $\mathcal{A}'$, it is easy to compute the class of $(0,0)$ modulo φ if $\mathcal{A}'$ is indeed a quotient of a (modified) Pascal automaton by φ. Starting from i, one first marks the states on the g-circuit C. Then, starting from i again, one follows the 0^{-1}-transitions: the *first time* C is crossed yields t. This parameter is *characteristic* of φ, as explained now.

Let (s,t) be an element of the semidirect product $G_p = \mathbb{Z}/p\mathbb{Z} \rtimes \mathbb{Z}/\psi\mathbb{Z}$ and $\tau_{(s,t)}$ the permutation on G_p induced by the multiplication *on the left* by (s,t):

$$\tau_{(s,t)}((x,y)) = (s,t)\,(x,y) = (x\,b^t + s, y+t) \quad . \tag{3}$$

The same element (s,t) defines a permutation $\sigma_{(s,t)}$ on $\mathbb{Z}/p\mathbb{Z}$ as well:

$$\forall x \in \mathbb{Z}/p\mathbb{Z} \qquad \sigma_{(s,t)}(x) = x\,b^t + s \quad . \tag{4}$$

Given a permutation σ over a set S, *the orbit* of an element s of S under σ is the set $\{\sigma^i(s) \mid i \in \mathbb{N}\}$. *An orbit* of σ is one of these sets.

Proposition 3. *Let* $\varphi\colon {\mathcal{P}'}_p^R \to \mathcal{A}'$ *be a covering and let* (s,t) *be the state* φ*-equivalent to* $(0,0)$ *with the smallest second component. Then, every* φ*-class is an orbit of* $\tau_{(s,t)}$ *(in* G_p*) and* R *is an union of orbits of* $\sigma_{(s,t)}$ *(in* $\mathbb{Z}/p\mathbb{Z}$*).*

Synthesis: verification that a given automaton is a quotient of a Pascal automaton Given $\mathcal{A}' = \langle Q, B, \delta', i, T \rangle$, let p, R and (s,t) computed as explained above. It is easily checked that R is an union of orbits of $\sigma_{(s,t)}$ and that $\|Q\| = pt$. The last step is the verification that $\mathcal{A}'$ is indeed (isomorphic to) the quotient of ${\mathcal{P}'}_p^R$ by the morphism φ defined by (s,t).

A corollary of Proposition 3 (and of the multiplication law in G_p) is that every class modulo φ contains one, and exactly one, element whose second component is smaller than t. From this observation follows that the multiplication by the generators $0 = (0,1)$ and $g = (1,0)$ in the quotient of ${\mathcal{P}'}_p^R$ by φ may be described on the set of representatives $Q_\varphi = \{(x,z) \mid x \in \mathbb{Z}/p\mathbb{Z}, z \in \mathbb{Z}/t\mathbb{Z}\}$ (beware that z is in $\mathbb{Z}/t\mathbb{Z}$ and not in $\mathbb{Z}/\psi\mathbb{Z}$) by the following formulas:

$$\forall (x,z) \in Q_\varphi$$

$$(x,z)\cdot 0 = (x,z)\,(0,1) = \begin{cases} (x, z+1) & \text{if} \quad z < t-1 \\ \tau^{-1}_{(s,t)}(x,z+1) = (\frac{x-s}{b^t}, 0) & \text{if} \quad z = t-1 \end{cases}$$

$$(x,z)\cdot g = (x,z)\,(1,0) = (x + b^z, z) \quad .$$

Hence $\mathcal{A}'$ is the quotient of ${\mathcal{P}'}_p^R$ by φ if one can mark Q according to these rules, starting from i with the mark $(0,0)$, without conflicts and in such a way that two distinct states have distincts marks. Such a marking is realised by a simple traversal of $\mathcal{A}'$, thus in linear time, and this concludes the proof of Theorem 8.

Remark 1. Theorem 8 states that one can decide in linear time whether a *given* automaton $\mathcal{A}$ is a quotient of a Pascal automaton, and in particular $\mathcal{A}$ has a fixed initial state that plays a crucial role in the verification process.

The following proposition shows that the property (being a quotient of a Pascal automaton) is actually independent of the state chosen to be initial. If it holds for $\mathcal{A}$, it also holds for any automaton derived from $\mathcal{A}$ by changing the initial state. This is a general property that will be used in the general verification process described in the next section.

Proposition 4. *If an automaton* $\mathcal{A} = \langle\, Q, A_b, \delta, i, T\,\rangle$ *is the quotient of* $\mathcal{P}_p^R$*, then for every state* q *in* Q*,* $\mathcal{A}_q = \langle\, Q, A_b, \delta, q, T\,\rangle$ *is the quotient of* $\mathcal{P}_p^S$ *for some set* S*.*

3 The UP-criterion

Let $\mathcal{A} = \langle\, Q, A, E, I, T\,\rangle$ be an automaton, σ the strong connectivity equivalence relation on Q, and γ the surjective map from Q onto Q/σ. The *condensation* $\mathcal{C}_\mathcal{A}$ of $\mathcal{A}$ is the directed acyclic graph with loops (V, E) such that V is the image of Q by γ; and the edge (x, y) is in E if there exists a transition $q \xrightarrow{a} s$ in $\mathcal{A}$, for some q in $\gamma^{-1}(x)$, s in $\gamma^{-1}(y)$ and a in A. The condensation of $\mathcal{A}$ can be computed in linear time by Tarjan's algorithm (*cf.* [6]).

We say that an SCC C of an automaton $\mathcal{A}$ is *embeddable* in another SCC D of $\mathcal{A}$ if there exists an injective function $f : C \to D$ such that, for all q in C and a in A: if $q \cdot a$ is in C then $f(q \cdot a) = (f(q) \cdot a)$, and if $q \cdot a$ is not in C, then $f(q) \cdot a = q \cdot a$.

Definition 1 (The UP-criterion). *Let* $\mathcal{A}$ *be a complete deterministic automaton and* $\mathcal{C}_\mathcal{A}$ *its condensation. We say that* $\mathcal{A}$ *satisfies the UP-criterion (or equivalently that* $\mathcal{A}$ *is a UP-automaton) if the following five conditions hold.*

UP-0 *The successor by* 0 *of a final (resp. non-final) state of* $\mathcal{A}$ *is final (resp. non-final).*

UP-1 *Every non-trivial SCC of* $\mathcal{A}$ *that contains an internal transition labelled by a digit different from 0 is mapped by* γ *to a leaf of* $\mathcal{C}_\mathcal{A}$*.*
Such and SCC is called a Type 1 SCC.

UP-2 *Every non-trivial SCC of* $\mathcal{A}$ *which is not of Type 1:*
i) is a simple circuit labelled by 0 (or 0-circuit);
ii) is mapped by γ *to a vertex of* $\mathcal{C}_\mathcal{A}$ *which has a unique successor, and this successor is a leaf.*
Such an SCC is called a Type 2 SCC.

UP-3 *Every Type 1 SCC is the quotient of a Pascal automaton $\mathcal{P}_p^R$, for some R and p.*

UP-4 *Every Type 2 SCC is embeddable in the unique Type 1 SCC associated with it by (UP-2).*

It should be noted that (UP-0) is not a specific condition, it is more of a precondition (hence its numbering 0) to ensure that either all representations of an integer are accepted, or none them are. Moreover, (UP-1) and (UP-2) (together with the completeness of $\mathcal{A}$) imply the converse of (UP-1), namely that every SCC mapped by γ to a leaf of $\mathcal{C}_\mathcal{A}$ is a Type 1 SCC.

Example 2. Fig. 3 shows a simple but complete example of a UP-automaton. The framed subautomata are the minimisation of Pascal automata $\mathcal{P}_3^{\{1,2\}}$ on the top and $\mathcal{P}_5^{\{1,2,3,4\}}$ on the bottom. The two others non-trivial SCC's, $\{B_2, C_2\}$ and $\{D_2\}$, are reduced to 0-circuits. Each of them has successors in only one Pascal automaton.

The dotted lines highlight (UP-4). The circuit $(B_2,\ C_2)$ is embeddable in the Pascal automaton $\{A,\ B,\ C\}$ with the map $B_2 \mapsto B$ and $C_2 \mapsto C$. A similar observation can be made for the circuit (D_2).

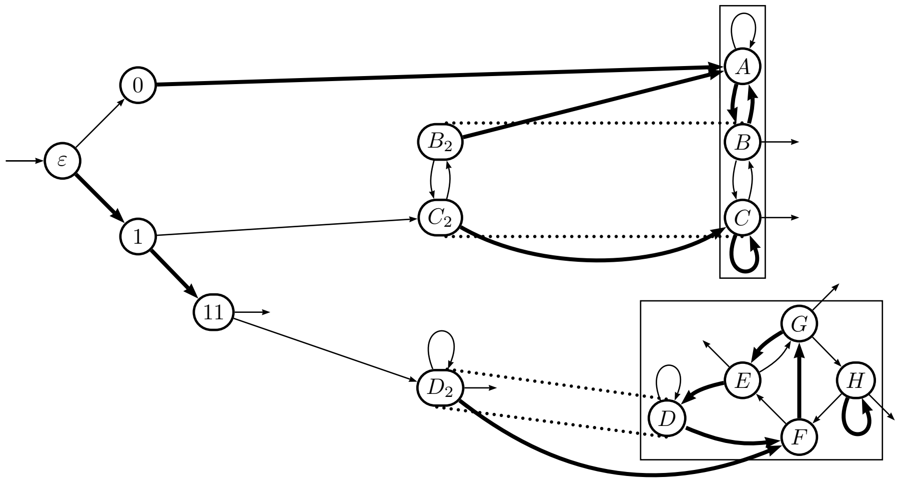

Figure 3: A complete example of the UP-criterion

Completeness and correctness of the UP-criterion are established as follows.

1. Every UP-set of numbers is accepted by a UP-automaton;
2. The UP-criterion is stable by quotient;
3. Every UP-automaton accepts a UP-set of numbers.

The first two steps ensure completeness for minimal automata (as every b-recognisable set of numbers is accepted by a *unique minimal automaton)*, the third one plays for correctness.

3.1 Every UP-set of numbers is accepted by a UP-automaton

Proposition 5. *For every integers m and p and for every set R of residues there exists a UP-automaton accepting $E^R_{p,m}$.*

When the period divides a power of the base Let E^R_p be a periodic set of numbers such that $p\,|\,b^j$ for some j. An automaton accepting E^R_p is obtained by a generalisation of the method for recognising if an integer written in base 10 is a multiple of 5, namely checking if its unit digit is 0 or 5: from (1) follows:

Lemma 1. *Let d be an integer such that $d\,|\,b^j$ (and $d \not| \, b^{j-1}$) and u in $A_b{}^*$ of length j. Then, w in $A_b{}^*$ is such that $\overline{w} \equiv \overline{u}\,[d]$ if, and only if, $w = u\,v$ for a certain v.*

The case of periodic sets of numbers Let E^R_p be a periodic set of numbers. In contrast with Sect. 2.2, p and b are not supposed to be coprime anymore. Given a integer p, there exist k and d such that $p = k\,d$, k and b are coprime, and $d\,|\,b^j$ for a certain j. The Chinese remainder theorem, a simplified version of which is given below, allows to break the condition: 'being congruent to r modulo p' into two simpler conditions.

Theorem 9 (Chinese remainder theorem). *Let k and d be two coprime integers. Let r_k, r_d be two integers. There exists a unique integer $r < k\,d$ such that $r \equiv r_k\,[k]$ and $r \equiv r_d\,[d]$.*

Moreover, for every n such that $n \equiv r_k\,[k]$ and $n \equiv r_d\,[d]$, we have $n \equiv r\,[kd]$.

Let us assume for now that R is a singleton $\{r\}$, with r in $\{0, 1, \ldots, p-1\}$ and define $r_d = (r \bmod d)$ and $r_k = (r \bmod k)$. Theorem 9 implies:

$$\forall n \in \mathbb{N} \qquad n \equiv r\,[p] \qquad \Longleftrightarrow \qquad n \equiv r_k\,[k] \quad \text{and} \quad n \equiv r_d\,[d] \ . \qquad (5)$$

The Pascal automaton $\mathcal{P}^{r_k}_k$ accepts the integers satisfying $n \equiv r_k\,[k]$ and an automaton accepting the integers satisfying $n \equiv r_d\,[d]$ can be defined from Lemma 1. The *product* of the two automata accepts the integers satisfying both equations of the right-hand side of (5) and this is a UP-automaton. Figure 4 shows the automaton accepting integers congruent to 5 modulo 12 in base 2.

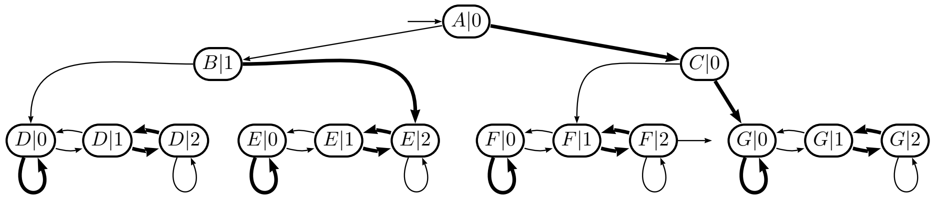

Figure 4: Automaton accepting integers congruent to 5 modulo 12 in base 2

The case where R is not a singleton is laboured but essentially the same. We denote by $\mathcal{B}^R_p$ the automaton accepting E^R_p.

The case of arbitrary UP-sets of numbers Let us denote by $\mathcal{D}_m$ the automaton accepting words whose value is greater than m. It consists in a complete b-tree T_m of depth $\lceil log_b(m) \rceil$ plus a final sink state. Every state may be labelled by the value of the word reaching it and it is final if its label is greater than m. Additionally, every leaf of T_m loops onto itself by reading a 0 and reaches the sink state by reading any other digit. Every $\mathcal{D}_m$ is obviously a UP-automaton.

An arbitrary UP-set of numbers $E^R_{p,m}$ is accepted by the product $\mathcal{B}^R_p \times \mathcal{D}_m$, denoted by $\mathcal{B}^R_{p,m}$. The very special form of $\mathcal{D}_m$ makes it immediate that this product is a UP-automaton, and this complete the proof of Proposition 5.

3.2 The UP-criterion is stable by quotient

Proposition 6. *If $\mathcal{A}$ is a UP-automaton, then every quotient of $\mathcal{A}$ is also a UP-automaton.*

The UP-criterion relies on properties of SCC's that are stable by quotient. The proof of Proposition 6 then consists essentially of proving that SCC's are mapped into SSC's by the quotient.

3.3 Every UP-automaton accepts a UP-set of numbers

Let $\mathcal{A}$ be a UP-automaton and $\mathcal{C}_\mathcal{A}$ its condensation. We call *branch* of $\mathcal{C}_\mathcal{A}$ any path going from the root to a leaf using no loops. There is finitely many of them. The inverse image by γ of a branch of $\mathcal{C}_\mathcal{A}$ define a subautomaton of $\mathcal{A}$. Since a finite union of UP-sets of numbers is still UP, it is sufficient to prove the following statement.

Proposition 7. *Let $\mathcal{A}$ be a UP-automaton and $\mathcal{C}_\mathcal{A}$ its condensation. The inverse image by γ of a branch of $\mathcal{C}_\mathcal{A}$ accepts a UP-set of numbers.*

4 Conclusion and future work

This work almost closes the complexity question raised by the Honkala's original paper [11]. The simplicity of the arguments in the proof should not hide that the difficulty was to make the proofs simple. Two questions remain: getting rid, in Theorem 2 of the minimality condition; or of the condition of determinism.

We are rather optimistic for a positive answer to the first one. Since the minimisation of a DFA whose SCC's are simple cycles can be done in linear time (*cf.* [3]), it should be possible to verify in linear time that the higher part of the UP-criterion (DAG and Type 2 SCC's) is satisfied by the minimised of a given automaton without performing the whole minimisation. It remains to find an algorithm deciding in linear time whether a given DFA has the same behaviour as a Pascal automaton. This is the subject of still ongoing work of the authors.

On the other hand, defining a similar UP-criterion for nondeterministic automata seems to be much more difficult. The criterion relies on the form and relations between SCC's, and the determinisation process is prone to destroy them.

References

1. Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.
2. Jean-Paul Allouche, Narad Rampersad, and Jeffrey Shallit. Periodicity, repetitions, and orbits of an automatic sequence. *Theoret. Comput. Sci.*, 410:2795–2803, 2009.
3. Jorge Almeida and Marc Zeitoun. Description and analysis of a bottom-up DFA minimization algorithm. *Inf. Process. Lett.*, 107(2):52–59, 2008.
4. Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Soc. Math.*, 1:191–238, 1994. Corrigendum, *Bull. Belg. Soc. Math.* 1:577 (1994).
5. Alan Cobham. On the base-dependance of the sets of numbers recognizable by finite automata. *Math. Systems Theory*, 3:186–192, 1969.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3rd ed.).* MIT Press, 2009.
7. Fabien Durand and Michel Rigo. On Cobham's theorem, 2011. HAL-00605375, to appear in *AutoMathA Handbook*, (J-E. Pin, Ed.), E.M.S..
8. Christiane Frougny. Representation of numbers and finite automata. *Math. Systems Theory*, 25:37–60, 1992.
9. Christiane Frougny and Jacques Sakarovitch. Number representation and finite automata. in *Combinatorics, Automata and Number Theory*, V. Berthé, M. Rigo (Eds), Encyclopedia of Mathematics and its Applications 135, Cambridge Univ. Press (2010) 34–107.
10. Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas and languages. *Pacif. J. Math.*, 16:285–296, 1966.
11. Juha Honkala. A decision method for the recognizability of sets defined by number systems. *RAIRO Theor. Informatics and Appl.*, 20:395–403, 1986.
12. Pierre Lecomte and Michel Rigo. Abstract numeration systems. in *Combinatorics, Automata and Number Theory*, V. Berthé, M. Rigo (Eds), Encyclopedia of Mathematics and its Applications 135, Cambridge Univ. Press (2010) 108–162.
13. Jérôme Leroux. A polynomial time Presburger criterion and synthesis for number decision diagrams. In *Logic in Computer Science 2005 (LICS'2005)*, pages 147–156. IEEE Comp. Soc. Press, 2005. New version at arXiv:cs/0612037v1.
14. Victor Marsault and Jacques Sakarovitch. Ultimate periodicity of b-recognisable sets : a quasilinear procedure. http://arxiv.org/abs/1301.2691.
15. A. Muchnik. The definable criterion for definability in Presburger arithmetic and its applications. *Theoret. Computer Sci.*, 290:1433–1444, 2003. Late publication in a journal of a preprint (in russian) issued in 1991.
16. Jacques Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009. Corrected English translation of *Éléments de théorie des automates*, Vuibert, 2003.