On the Use of Weak Automata for Deciding Linear Arithmetic with Integer and Real Variables*

Bernard Boigelot, Sébastien Jodogne, and Pierre Wolper

Université de Liège,
Institut Montefiore, B28,
4000 Liège, Belgium
{boigelot,jodogne,pw}@montefiore.ulg.ac.be,
http://www.montefiore.ulg.ac.be/~{boigelot,jodogne,pw}/

Abstract. This paper considers finite-automata based algorithms for handling linear arithmetic with both real and integer variables. Previous work has shown that this theory can be dealt with by using finite automata on infinite words, but this involves some difficult and delicate to implement algorithms. The contribution of this paper is to show, using topological arguments, that only a restricted class of automata on infinite words are necessary for handling real and integer linear arithmetic. This allows the use of substantially simpler algorithms and opens the path to the implementation of a usable system for handling this combined theory.

1 Introduction

Among the techniques used to develop algorithms for deciding or checking logical formulas, finite automata have played an important role in a variety of cases. Classical examples are the use of infinite-word finite automata by Büchi [Büc62] for obtaining decision procedures for the first and second-order monadic theories of one successor as well as the use of tree automata by Rabin [Rab69] for deciding the second-order monadic theory of n successors. More recent examples are the use of automata for obtaining decision and model-checking procedures for temporal and modal logics [VW86a,VW86b,VW94,KVW00]. In this last setting, automata-based procedures have the advantage of moving the combinatorial aspects of the procedures to the context of automata, which are simple graph-like structures well adapted to algorithmic development. This separation of concerns between the logical and the algorithmic has been quite fruitful for instance in the implementation of model checkers for linear-time temporal logic [CVWY90,Hol97].

* This work was partially funded by a grant of the "Communauté française de Belgique - Direction de la recherche scientifique - Actions de recherche concertées" and by the European Commission (FET project ADVANCE, contract No IST-1999-29082).

As already noticed by Büchi [Büc60,Büc62], automata-based approaches are not limited to sequential and modal logics, but can also be used for Presburger arithmetic. To achieve this, one adopts the usual encoding of integers in a base $r \geq 2$, thus representing an integer as a word over the alphabet $\{0, \ldots, r-1\}$. By extension, n-component integer vectors are represented by words over the alphabet $\{0, \ldots, r-1\}^n$ and a finite automaton operating over this alphabet represents a set of integer vectors. Given that addition and order are easily represented by finite automata and that these automata are closed under Boolean operations as well as projection, one easily obtains a decision procedure for Presburger arithmetic. This idea was first explored at the theoretical level, yielding for instance the very nice result that base-independent finite-automaton representable sets are exactly the Presburger sets [Cob69,Sem77,BHMV94]. Later, it has been proposed as a practical means of deciding and manipulating Presburger formulas [BC96,Boi98,SKR98,WB00]. The intuition behind this applied use of automata for Presburger arithmetic is that finite automata play with respect to Presburger arithmetic a role similar to the one of Binary Decision Diagrams (BDDs) with respect to Boolean logic. These ideas have been implemented in the LASH tool [LASH], which has been used successfully in the context of verifying systems with unbounded integer variables.

It almost immediately comes to mind that if a finite word over the alphabet $\{0, \ldots, r-1\}$ can represent an integer, an infinite word over the same alphabet extended with a fractional part separator (the usual dot) can represent a real number. Finite automata on infinite words can thus represent sets of real vectors, and serve as a means of obtaining a decision procedure for real additive arithmetic. Furthermore, since numbers with empty fractional parts can easily be recognized by automata, the same technique can be used to obtain a decision procedure for a theory combining the integers and the reals. This is not presently handled by any tool, but can be of practical use, for instance in the verification of timed systems using integer variables [BBR97]. However, turning this into an effective implemented system is not as easy as it might first seem. Indeed, projecting and complementing finite automata on infinite words is significantly more difficult than for automata on finite words. Projection yields nondeterministic automata and complementing or determinizing infinite-word automata is a notoriously difficult problem. A number of algorithms have been proposed for this [Büc62,SVW87,Saf88,KV97], but even though their theoretical complexity remains simply exponential as in the finite-word case, it moves up from $2^{O(n)}$ to $2^{O(n \log n)}$ and none of the proposed algorithms are as easy to implement and fine-tune as the simple Rabin-Scott subset construction used in the finite-word case.

However, it is intuitively surprising that handling reals is so much more difficult than handling integers, especially in light of the fact that the usual polyhedra-based approach to handling arithmetic is both of lower complexity and easier to implement for the reals than for the integers [FR79]. One would expect that handling reals with automata should be no more difficult than handling integers[1]. The conclusion that comes out of these observations is that infinite-word automata constructed from linear arithmetic formulas must have a special structure that makes them easier to manipulate than general automata on infinite words. That this special structure exists and that it can exploited to obtain simpler algorithms is precisely the subject of this paper.

As a starting point, let us look at the topological characterization of the sets definable by linear arithmetic formulas. Let us first consider a formula involving solely real variables. If the formula is quantifier free, it is a Boolean combination of linear constraints and thus defines a set which is a finite Boolean combination of open and closed sets. Now, since real linear arithmetic admits quantifier elimination, the same property also holds for quantified formulas. Then, looking at classes of automata on infinite words, one notices that the most restricted one that can accept Boolean combinations of open and closed sets is the class of deterministic weak automata [SW74,Sta83]. These accept all ω-regular sets in the Borel class $F_\sigma \cap G_\delta$ and hence also finite Boolean combinations of open and closed sets. So, with some care about moving from the topology on vectors to the topology on their encoding as words, one can conclude that the sets representable by arithmetic formulas involving only real variables can always be accepted by deterministic weak automata on infinite words. If integers are also involved in the formula, there is no established quantifier elimination result for the combined theory and one cannot readily conclude the same. A first result in this paper closes this loophole. It establishes that sets definable by quantified linear arithmetic formulas involving both real and integer variables are within $F_\sigma \cap G_\delta$ and thus are representable by deterministic weak automata. Rather than using a quantifier elimination type argument to establish this, our proof relies on separating the integer and fractional parts of variables and on topological properties of $F_\sigma \cap G_\delta$.

The problematic part of the operations on automata needed to decide a first-order theory is the sequence of projections and complementations needed to eliminate a string of quantifiers alternating between existential and universal ones. The second result of this paper shows that for sets defined in linear arithmetic this can be done with constructions that are simple adaptations of the ones used for automata on finite words. Indeed, deterministic weak automata can be viewed as either Büchi or co-Büchi automata. The interesting fact is that co-Büchi automata can be determinized by the "breakpoint" construction [MH84,KV97], which basically amounts to a product of subset constructions. Thus, one has a simple construction to project and determinize a weak automaton, yielding a deterministic co-Büchi automaton, which is easily complemented into a deterministic Büchi automaton. In the general case, another round of projection will lead to a nondeterministic Büchi automaton, for which a general determinization procedure has to be used. However, we have the result that for automata obtained from linear arithmetic formulas, the represented sets stay

[1] Note that one cannot expect reals to be easier to handle with automata than integers since, by nature, this representation includes explicit information about the existence of integer values satisfying the represented formula.

within those accepted by deterministic weak automata. We prove that this implies that the automata obtained after determinization will always be weak.

Note that this cannot be directly concluded from the fact that the represented sets stay within those representable by deterministic weak automata. Indeed, even though the represented sets can be accepted by deterministic weak automata, the automata that are obtained by the determinization procedure might not have this form. Fortunately, we can prove that this is impossible. For this, we go back to the link between automata and the topology of the sets of infinite words they accept. The argument is that ω-regular sets in $F_\sigma \cap G_\delta$ have a topological property that forces the automata accepting them to be inherently weak, i.e. not to have strongly connected components containing both accepting and non accepting cycles.

As a consequence of our results, we obtain a decision procedure for the theory combining integer and real linear arithmetic that is suitable for implementation. The fact that this theory is decidable was known [BBR97], but the results of this paper move us much closer to an implemented tool that can handle it effectively.

2 Automata-Theoretic and Topological Background

In this section we recall some automata-theoretic and topological concepts that are used in the paper.

2.1 Automata on Infinite Words

An infinite word (or ω-word) w over an alphabet Σ is a mapping $w : \mathbb{N} \to \Sigma$ from the natural numbers to Σ. A Büchi automaton on infinite words is a five-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of states;
- Σ is the input alphabet;
- δ is the transition function and is of the form $\delta : Q \times \Sigma \to 2^Q$ if the automaton is nondeterministic and of the form $\delta : Q \times \Sigma \to Q$ if the automaton is deterministic;
- q_0 is the initial state;
- F is a set of accepting states.

A run π of a Büchi automaton $A = (Q, \Sigma, \delta, q_0, F)$ on an ω-word w is a mapping $\pi : \mathbb{N} \to Q$ that satisfies the following conditions:

- $\pi(0) = q_0$, i.e. the run starts in the initial state;
- For all $i \geq 0$, $\pi(i+1) \in \delta(\pi(i), w(i))$ (nondeterministic automata) or $\pi(i+1) = \delta(\pi(i), w(i))$ (deterministic automata), i.e. the run respects the transition function.

Let $\mathit{inf}(\pi)$ be the set of states that occur infinitely often in a run π. A run π is said to be accepting if $\mathit{inf}(\pi) \cap F \neq \emptyset$. An ω-word w is accepted by a Büchi automaton if that automaton has some accepting run on w. The language $L_\omega(A)$ of infinite words defined by a Büchi automaton A is the set of ω-words it accepts.

A co-Büchi automaton is defined exactly as a Büchi automaton except that its accepting runs are those for which $\mathit{inf}(\pi) \cap F = \emptyset$.

We will also use the notion of *weak* automata [MSS86]. For a Büchi automaton $A = (Q, \Sigma, \delta, q_0, F)$ to be *weak*, there has to be a partition of its state set Q into disjoint subsets $Q_1, \ldots, Q_m$ such that

- for each of the Q_i either $Q_i \subseteq F$ or $Q_i \cap F = \emptyset$, and
- there is a partial order $\leq$ on the sets $Q_1, \ldots, Q_m$ such that for every $q \in Q_i$ and $q' \in Q_j$ for which, for some $a \in \Sigma$, $q' \in \delta(q, a)$ ($q' = \delta(q, a)$ in the deterministic case), $Q_j \leq Q_i$.

For more details, a survey of automata on infinite words can be found in [Tho90].

2.2 Topology

Given a set S, a distance $d(x, y)$ defined on this set induces a topology on subsets of S. A neighborhood $N_\varepsilon(x)$ of a point $x \in S$ is the set $N_\varepsilon(x) = \{y \mid d(x, y) < \varepsilon\}$. A set $C \subseteq S$ is said to be open if for all $x \in C$, there exists $\varepsilon > 0$ such that the neighborhood $N_\varepsilon(x)$ is contained in C. A closed set is a set whose complement with respect to S is open. We will be referring to the first few levels of the Borel hierarchy which are shown in Figure 1. The notations used are the following:

- F are the closed sets,
- G are the open sets,
- F_σ is the class of countable unions of closed sets,
- G_δ is the class of countable intersections of open sets,
- $F_{\sigma\delta}$ is the class of countable intersections of F_σ sets,
- $G_{\delta\sigma}$ is the class of countable unions of G_δ sets,
- $\mathcal{B}(X)$ represents the finite Boolean combinations of sets in X.

An arrow between classes indicates proper inclusion.

3 Topological Characterization of Arithmetic Sets

We consider the theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$, where $+$ represents the predicate $x + y = z$. Since any linear equality or order constraint can be encoded into this theory, we refer to it as additive or linear arithmetic over the reals and integers. It is the extension of Presburger arithmetic that includes both real and integer variables. In this section, we prove that the sets representable in this theory belong to the topological class $F_\sigma \cap G_\delta$ defined relatively to the Euclidean distance between vectors. This result is formalized by the following theorem.

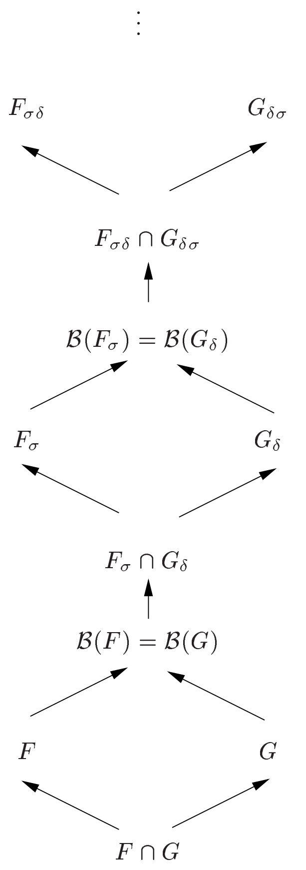

Fig. 1. The first few levels of the Borel hierarchy.

Theorem 1. *Let $S \subseteq \mathbb{R}^n$, with $n > 0$, be a set defined in the theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$. This set belongs to the topological class $F_\sigma \cap G_\delta$ induced by the distance*

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2} .$$

Proof. Since $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ is closed under negation, it is actually sufficient to show that each formula of this theory defines a set that belongs to F_σ, i.e., a set that can be expressed as a countable union of closed sets.

Let φ be a formula of $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$. To simplify our argument, we will assume that all free variables of φ are reals. This can be done without loss of generality since quantified variables can range over both $\mathbb{R}$ and $\mathbb{Z}$. We introduce $u < v$ as a shorthand for $u \leq v \wedge \neg(u = v)$.

The first step of our proof consists of modifying φ in the following way. We replace each variable x that appears in φ by two variables x_I and x_F representing respectively the integer and the fractional part of x. Formally, this operation replaces each occurrence in φ of a free variable x by the sum $x_I + x_F$ while adding to φ the constraints $0 \leq x_F$ and $x_F < 1$, and transforms the quantified variables of φ according to the following rules:

$$\begin{aligned}
(\exists x \in \mathbb{R})\phi &\longrightarrow (\exists x_I \in \mathbb{Z})(\exists x_F \in \mathbb{R})(0 \leq x_F \wedge x_F < 1 \wedge \phi[x/x_I + x_F]) \\
(\forall x \in \mathbb{R})\phi &\longrightarrow (\forall x_I \in \mathbb{Z})(\forall x_F \in \mathbb{R})(x_F < 0 \vee 1 \leq x_F \vee \phi[x/x_I + x_F]) \\
(Qx \in \mathbb{Z})\phi &\longrightarrow (Qx_I \in \mathbb{Z})\phi[x/x_I],
\end{aligned}$$

where $Q \in \{\exists, \forall\}$, ϕ is a subformula, and $\phi[x/y]$ denotes the result of replacing by y each occurrence of x in ϕ. The transformation has no influence on the set represented by φ, except that the integer and fractional parts of each value are now represented by two distinct variables.

Now, the atomic formulas of φ are of the form $p = q + r$, $p = q$ or $p \leq q$, where p, q and r are either integer variables, sums of an integer and of a fractional variable, or integer constants. The second step consists of expanding these atomic formulas so as to send into distinct atoms the occurrences of the integer and of the fractional variables. This is easily done with the help of simple arithmetic rules, for the truth value of the atomic formulas that involve both types of variables has only to be preserved for values of the fractional variables that belong to the interval $[0, 1[$. The less trivial expansion rules[2] are given below:

$$\begin{aligned}
(x_I + x_F) = (y_I + y_F) &\longrightarrow x_I = y_I \wedge x_F = y_F \\
(x_I + x_F) \leq (y_I + y_F) &\longrightarrow x_I < y_I \vee (x_I = y_I \wedge x_F \leq y_F) \\
(x_I + x_F) = (y_I + y_F) + (z_I + z_F) &\longrightarrow (x_I = y_I + z_I \wedge x_F = y_F + z_F) \\
&\quad\ \vee (x_I = y_I + z_I + 1 \wedge x_F = y_F + z_F - 1) \\
(x_I + x_F) = (y_I + y_F) + z_I &\longrightarrow x_I = y_I + z_I \wedge x_F = y_F \\
x_I = (y_I + y_F) + (z_I + z_F) &\longrightarrow (x_I = y_I + z_I \wedge y_F + z_F = 0) \\
&\quad\ \vee (x_I = y_I + z_I + 1 \wedge y_F + z_F = 1)
\end{aligned}$$

After the transformation, each atomic formula of φ is either a formula ϕ_I involving only integer variables or a formula ϕ_F over fractional variables. We now distribute existential (resp. universal) quantifiers over disjunctions (resp. conjunctions), after rewriting their argument into disjunctive (resp. conjunctive) normal form, and then apply the simplification rules

$$\begin{aligned}
(Qx_I \in \mathbb{Z})(\phi_I \,\alpha\, \phi_F) &\longrightarrow (Qx_I \in \mathbb{Z})(\phi_I)\ \alpha\ \phi_F \\
(Qx_F \in \mathbb{R})(\phi_I \,\alpha\, \phi_F) &\longrightarrow \phi_I\ \alpha\ (Qx_F \in \mathbb{R})(\phi_F),
\end{aligned}$$

where $Q \in \{\exists, \forall\}$ and $\alpha \in \{\vee, \wedge\}$.

Repeating this operation, we eventually get a formula φ that takes the form of a finite Boolean combination $\mathcal{B}(\phi_I^{(1)}, \phi_I^{(2)}, \ldots, \phi_I^{(m)}, \phi_F^{(1)}, \phi_F^{(2)}, \ldots, \phi_F^{(m')})$ of subformulas $\phi_I^{(i)}$ and $\phi_F^{(i)}$ that involve respectively only integer and fractional variables.

[2] In these rules, the expression $p = q + r + s$ is introduced as a shorthand for $(\exists u)(u = q + r \wedge p = u + s)$, where the quantifier is defined over the appropriate domain.

Let $x_I^{(1)}, x_I^{(2)}, \ldots, x_I^{(k)}$ be the free integer variables of φ. For each assignment of values to these variables, the subformulas $\phi_I^{(i)}$ are each identically true or false, hence we have

$$\varphi \equiv \bigvee_{(a_1,\ldots,a_k)\in\mathbb{Z}^k} \Big((x_I^{(1)}, \ldots, x_I^{(k)}) = (a_1, \ldots, a_k) \wedge \mathcal{B}_{(a_1,\ldots,a_k)}(\phi_F^{(1)}, \ldots, \phi_F^{(m')}) \Big) .$$

Each subformula $\phi_F^{(i)}$ belongs to the theory $\langle \mathbb{R}, +, \leq, 1 \rangle$, which admits the elimination of quantifiers [FR79]. The sets of reals vectors satisfying these formulas are thus finite Boolean combinations of linear constraints with open or closed boundaries. It follows that, for each $(a_1, \ldots, a_k) \in \mathbb{Z}^k$, the set described by $\mathcal{B}_{(a_1,\ldots,a_k)}(\phi_F^{(1)}, \ldots, \phi_F^{(m')})$ is a finite Boolean combination of open and closed sets and, since any open set is a countable union of closed sets, is within F_σ. Therefore, the set described by φ is a countable union of F_σ sets and is also within F_σ.

4 Representing Sets of Integers and Reals with Finite Automata

In this section, we recall the finite-state representation of sets of real vectors as introduced in [BBR97].

In order to make a finite automaton recognize numbers, one needs to establish a mapping between these and words. Our encoding scheme corresponds to the usual notation for reals and relies on an arbitrary integer base $r > 1$. We encode a number x in base r, most significant digit first, by words of the form $w_I \star w_F$, where w_I encodes the integer part x_I of x as a finite word over $\{0, \ldots, r-1\}$, the special symbol "$\star$" is a separator, and w_F encodes the fractional part x_F of x as an infinite word over $\{0, \ldots, r-1\}$. Negative numbers are represented by their r's complement. The length p of $|w_I|$, which we refer to as the *integer-part length* of w, is not fixed but must be large enough for $-r^{p-1} \leq x_I < r^{p-1}$ to hold.

According to this scheme, each number has an infinite number of encodings, since their integer-part length can be increased unboundedly. In addition, the rational numbers whose denominator has only prime factors that are also factors of r have two distinct encodings with the same integer-part length. For example, in base 10, the number $11/2$ has the encodings $005 \star 5(0)^\omega$ and $005 \star 4(9)^\omega$, "ω" denoting infinite repetition.

To encode a vector of real numbers, we represent each of its components by words of identical integer-part length. This length can be chosen arbitrarily, provided that it is sufficient for encoding the vector component with the highest magnitude. An encoding of a vector $\boldsymbol{x} \in \mathbb{R}^n$ can indifferently be viewed either as a n-tuple of words of identical integer-part length over the alphabet $\{0, \ldots, r-1, \star\}$, or as a single word w over the alphabet $\{0, \ldots, r-1\}^n \cup \{\star\}$.

Since a real vector has an infinite number of possible encodings, we have to choose which of these the automata will recognize. A natural choice is to accept all encodings. This leads to the following definition.

Definition 1. *Let $n > 0$ and $r > 1$ be integers. A* Real Vector Automaton (RVA) *A in base r for vectors in $\mathbb{R}^n$ is a Büchi automaton over the alphabet $\{0, \ldots, r-1\}^n \cup \{\star\}$, such that*

- *Every word accepted by A is an encoding in base r of a vector in $\mathbb{R}^n$, and*
- *For every vector $\boldsymbol{x} \in \mathbb{R}^n$, A accepts either all the encodings of $\boldsymbol{x}$ in base r, or none of them.*

An RVA is said to *represent* the set of vectors encoded by the words that belong to its accepted language. Efficient algorithms have been developed for constructing RVA representing the sets of solutions of systems of linear equations and inequations [BRW98]. Since it is immediate to constrain a number to be an integer with an RVA and since, using existing algorithms for infinite-word automata, one can apply Boolean operations as well as projection to RVA, it follows that one can construct an RVA for any formula of the arithmetic theory we are considering.

5 Weak Automata and their Properties

If one examines the constructions given in [BRW98] to build RVA for linear equations and inequations, one notices that they have the property that all states within the same strongly connected component are either accepting or nonaccepting. This implies that these automata are *weak* in the sense of [MSS86] (see Section 2).

Weak automata have a number of interesting properties. A first one is that they can be represented both as Büchi and co-Büchi. Indeed, a weak Büchi automaton $A = (Q, \Sigma, \delta, q_0, F)$ is equivalent to the co-Büchi automaton $A = (Q, \Sigma, \delta, q_0, Q \setminus F)$, since a run eventually remains within a single component Q_i in which all states have the same status with respect to being accepting. A consequence of this is that weak automata can be determinized by the fairly simple "breakpoint" construction [MH84,KV97] that can be used for co-Büchi automata. This construction is the following.

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic co-Büchi automaton. The deterministic co-Büchi automaton $A' = (Q', \Sigma, \delta', q'_0, F')$ defined as follows accepts the same ω-language.

- $Q' = 2^Q \times 2^Q$, the states of A' are pairs of sets of states of A.
- $q'_0 = (\{q_0\}, \emptyset)$.
- For $(S, R) \in Q'$ and $a \in \Sigma$, the transition function is defined by
  - if $R = \emptyset$, then $\delta((S, R), a) = (T, T \setminus F)$ where $T = \{q \mid \exists p \in S \text{ and } q \in \delta(p, a)\}$, T is obtained from S as in the classical subset construction, and the second component of the pair of sets of states is obtained from T by eliminating states in F;
  - if $R \neq \emptyset$, then $\delta((S, R), a) = (T, U \setminus F)$ where $T = \{q \mid \exists p \in S \text{ and } q \in \delta(p, a)\}$, and $U = \{q \mid \exists p \in R \text{ and } q \in \delta(p, a)\}$, the subset construction set is now applied to both S and R and states in F are removed from U.

– $F' = 2^Q \times \{\emptyset\}$.

When the automaton A' is in a state (S, R), R represents the states of A that can be reached by a run that has not gone through a state in F since that last "breakpoint", i.e. state of the form $(S, \emptyset)$. So, for a given word, A has a run that does not go infinitely often through a state in F iff A' has a run that does not go infinitely often through a state in F'. Notice that the difficulty that exists for determinizing Büchi automata, which is to make sure that the *same* run repeatedly reaches an accepting state disappears since, for co-Büchi automata, we are just looking for a run that eventually avoids accepting states.

It is interesting to notice that the construction implies that all reachable states (S, R) of A' satisfy $R \subseteq S$. The breakpoint construction can thus be implemented as a subset construction in which the states in R are simply tagged. One can thus expect it to behave in practice very similarly to the traditional subset construction for finite-word automata.

Another property of weak automata that will be of particular interest to us is the topological characterization of the sets of words that they can accept. Consider the topology on the set of ω-words induced by the distance

$$d(w, w') = \begin{cases} \frac{1}{|common(w,w')|+1} & \text{if } w \neq w' \\ 0 & \text{if } w = w', \end{cases}$$

where $|common(w, w')|$ denotes the length of the longest common prefix of w and w'. In this topology, weak deterministic automata accept exactly the ω-regular languages that are in $F_\sigma \cap G_\delta$. This follows from the results on the Staiger-Wagner class of automata [SW74,Sta83], which coincides with the class of deterministic weak automata, as can be inferred from [SW74] and is shown explicitly in [MS97]. Given the result proved in Section 3, it is tempting to conclude that the encodings of sets definable in the theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ can always be accepted by weak deterministic automata. This conclusion is correct, but requires shifting the result from the topology on numbers to the topology on words, which we will do in the next section. In the meantime, we need one more result in order to be able to benefit algorithmically from the fact that we are dealing with $F_\sigma \cap G_\delta$ sets, i.e. that any deterministic automaton accepting a $F_\sigma \cap G_\delta$ set is essentially a weak automaton.

Consider the following definition.

Definition 2. *A Büchi automaton is* inherently weak *if none of the reachable strongly connected components of its transition graph contains both accepting (including at least one accepting state) and non accepting (not including any accepting state) cycles.*

Clearly, if an automaton is inherently weak, it can directly be transformed into a weak automaton. The partition of the state set is its partition into strongly connected components and all the states of a component are made accepting or not, depending on whether the cycles in that component are accepting or not.

We will now prove the following.

Theorem 2. *Any deterministic Büchi automaton that accepts a language in $F_\sigma \cap G_\delta$ is inherently weak.*

To prove this, we use the fact that the language accepted by an automaton that is not inherently weak must have the following *dense oscillating sequence* property.

Definition 3. *A language $L \subseteq \Sigma^\omega$ has the* dense oscillating sequence *property if, $w_1, w_2, w_3, \ldots$ being words and $\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots$ being distances, one has that $\exists w_1 \forall \varepsilon_1 \exists w_2 \forall \varepsilon_2 \ldots$ such that $d(w_i, w_{i+1}) \leq \varepsilon_i$ for all $i \geq 1$, $w_i \in L$ for all odd i, and $w_i \notin L$ for all even i.*

The fact that the language accepted by an automaton that is not inherently weak has the dense oscillating sequence property is an immediate consequence of the fact that such an automaton has a reachable strongly connected component containing both accepting and non accepting cycles. Given this, it is sufficient to prove the following lemma in order to establish Theorem 2.

Lemma 1. *An ω-regular language that has the dense oscillating sequence property cannot be accepted by a weak deterministic automaton and hence is not in $F_\sigma \cap G_\delta$.*

Proof. We proceed by contradiction. Assume that a language L having the dense oscillating sequence property is accepted by a weak deterministic automaton A. Consider the first word w_1 in a dense oscillating sequence for L. This word eventually reaches an accepting component Q_{i_1} of the partition of the state set of A and will stay within this component. Since ε_1 can be chosen freely, it can be taken small enough for the run of A on w_2 to also reach the component Q_{i_1} before it starts to differ from w_1. Since w_2 is not in L, the run of A on w_2 has to eventually leave the component Q_{i_1} and will eventually reach and stay within a non accepting component $Q_{i_2} < Q_{i_1}$. Repeating a similar argument, one can conclude that the run of A on w_3 eventually reaches and stays within an accepting component $Q_{i_3} < Q_{i_2}$. Carrying on with this line of reasoning, one concludes that the state set of A must contain an infinite decreasing sequence of distinct components, which is impossible given that it is finite.

6 Deciding Linear Arithmetic with Real and Integer Variables

We first show that the result of Section 3 also applies to the sets of words encoding sets defined in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$. In order to do so, we need to establish that the topological class $F_\sigma \cap G_\delta$ defined over sets of reals is mapped to its ω-word counterpart by the encoding relation described in Section 4.

Theorem 3. *Let $n > 0$ and $r > 1$ be integers, and let $L(S) \subseteq (\{0, \ldots, r-1\}^n \cup \{\star\})^\omega$ be the set of all the encodings in base r of the vectors belonging to the set $S \subseteq \mathbb{R}^n$. If the set S belongs to $F_\sigma \cap G_\delta$ (with respect to Euclidean distance), then the language $L(S)$ belongs to $F_\sigma \cap G_\delta$ (with respect to ω-word distance).*

Proof. Not all words over the alphabet $\{0,\ldots,r-1\}^n \cup \{\star\}$ encode a real vector. Let V be the set of all the valid encodings of vectors in base r. Its complement $\overline{V}$ can be partitioned into a set $\overline{V}_0$ containing only words in which the separator "$\star$" does not appear, and a set $\overline{V}_+$ containing words in which "$\star$" occurs at least once.

The set $\overline{V}_0 \cup V$ is closed. Indeed, each element of its complement is a word that does not encode validly a vector and that contains at least one separator. Such a word admits a neighborhood entirely composed of words satisfying the same property, which entails that the complement of $\overline{V}_0 \cup V$ is open. In the same way, one obtains that the set $\overline{V}_+ \cup V$ is open.

Let now consider an open set $S \subseteq \mathbb{R}^n$. The language $L' = L(S) \cup \overline{V}_+$ is open. Indeed, each word $w \in L(S)$ has a neighborhood entirely composed of words in $L(S)$ (formed by the encodings of vectors that belong to a neighborhood of the vector encoded by w), and of words that do not encode vectors but contain at least one separator. Moreover, each word $w \in \overline{V}_+$ admits a neighborhood fully composed of words in $\overline{V}_+$. Since $L(S) = L' \cap (\overline{V}_0 \cup V)$, we have that $L(S)$ is the intersection of an open and of a closed set.

The same line of reasoning can be followed with a closed set $S \subseteq \mathbb{R}^n$. The language $L'' = L(S) \cup \overline{V}_0$ is easily shown to be closed, which, since $L(S) = L'' \cap (\overline{V}_+ \cup V)$, implies that $L(S)$ is the intersection of a closed and of an open set.

We are now ready to address the case of a set $S \subseteq \mathbb{R}^n$ that belongs to $F_\sigma \cap G_\delta$. Since S is in F_σ, it can be expressed as a countable union of closed sets S_1, S_2, ... The languages $L(S_1)$, $L(S_2)$, ... are Boolean combinations of open and of closed sets, and thus belong to the topological class F_σ. Therefore, $L(S) = L(S_1) \cup L(S_2) \cup \cdots$ is a countable union of sets in F_σ, and thus belongs itself to F_σ. Now, since S is in G_δ, it can also be expressed as a countable intersection of open sets S'_1, S'_2, ... The languages $L(S'_1)$, $L(S'_2)$, ... belong to the topological class G_δ. Hence, $L(S) = L(S'_1) \cap L(S'_2) \cap \cdots$ is a countable intersection of sets in G_δ, and thus belongs itself to G_δ. This concludes our proof of the theorem.

Knowing that the encodings of sets definable in the theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ are in $F_\sigma \cap G_\delta$, we use the results of Section 5 to conclude the following.

Theorem 4. *Every deterministic RVA representing a set definable in* $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ *is inherently weak.*

This property has the important consequence that the construction and the manipulation of RVA obtained from arithmetic formulas can be performed effectively by algorithms operating on weak automata. Precisely, to obtain an RVA for an arithmetic formula one can proceed as follows.

For equations and inequations, one uses the constructions given in [BRW98] to build weak RVA. Computing the intersection, union, and Cartesian product of sets represented by RVA simply reduces to performing similar operations with the languages accepted by the underlying automata, which can be done by simple product constructions. These operations preserve the weak nature of the automata. To complement a weak RVA, one determinizes it using the breakpoint construction, which is guaranteed to yield an inherently weak automaton (Theorem 4) that is easily converted to a weak one. This deterministic weak RVA is then complemented by inverting the accepting or non-accepting status of each of its components, and then removing from its accepted language the words that do not encode validly a vector (which is done by means of an intersection operation).

Applying an existential quantifier to a weak RVA is first done by removing from each transition label the symbol corresponding to the vector component that is projected out. This produces a non-deterministic weak automaton that may only accept some encodings of each vector in the quantified set, but generally not all of them. The second step thus consists of modifying the automaton so as to make it accept every encoding of each vector that it recognizes. Since different encodings of a same vector differ only in the number of times that their leading symbol is repeated, this operation can be carried out by the same procedure as the one used with finite-word number automata [Boi98]. This operation does not affect the weak nature of the automaton, which can then be determinized by the breakpoint construction, which has to produce an inherently weak RVA easily converted to a weak automaton.

Thus, in order to decide whether a formula of $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ is satisfiable, one simply builds an RVA representing its set of solutions, and then check whether this automaton accepts a nonempty language. This also makes it possible to check the inclusion or the equivalence of sets represented by RVA. The main result of this paper is that, at every point, the constructed automaton remains weak and thus only the simple breakpoint construction is needed as a determinization procedure.

7 Conclusions

A probably unusual aspect of this paper is that it does not introduce new algorithms, but rather shows that existing algorithms can be used in a situation where *a priori* they could not be expected to operate correctly. To put it in other words, the contribution is not the algorithm but the proof of its correctness.

The critical reader might be wondering if all this is really necessary. After all, algorithms for complementing Büchi automata exist, either through determinization [Saf88] or directly [Büc62,SVW87,Kla91,KV97] and the more recent of these are even fairly simple and potentially implementable. There are no perfectly objective grounds on which to evaluate "simplicity" and "ease of implementation", but it is not difficult to convince oneself that the breakpoint construction for determinizing weak automata is simpler than anything proposed for determinizing or complementing Büchi automata. Indeed, it is but one step of the probably simplest complementation procedure proposed so far, that of [KV97]. Furthermore, there is a complexity improvement from $2^{O(n \log n)}$ to $2^{O(n)}$; experience with the subset construction as used for instance in the LASH tool [LASH] indicates that the breakpoint construction is likely to operate very well in practice; and being able to work with deterministic automata allows minimization [Löd01], which leads to a normal form.

An implementation and some experiments would of course substantiate the claims to simplicity and ease of implementation. It is planned in the context of the LASH tool and will be made available [LASH]. However, this paper is not about an implementation, but about the fact that, with the help of what might appear to be pure theory, one can obtain very interesting conclusions about algorithms for handling the theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$.

References

[BBR97] B. Boigelot, L. Bronne, and S. Rassart. An improved reachability analysis method for strongly linear hybrid systems. In *Proc. 9th Int. Conf.on Computer Aided Verification*, volume 1254 of *Lecture Notes in Computer Science*, pages 167–178, Haifa, June 1997. Springer-Verlag.

[BC96] A. Boudet and H. Comon. Diophantine equations, Presburger arithmetic and finite automata. In *Proceedings of CAAP'96*, number 1059 in Lecture Notes in Computer Science, pages 30–43. Springer-Verlag, 1996.

[BHMV94] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bulletin of the Belgian Mathematical Society*, 1(2):191–238, March 1994.

[Boi98] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces.* PhD thesis, Université de Liège, 1998.

[BRW98] Bernard Boigelot, Stéphane Rassart, and Pierre Wolper. On the expressiveness of real and integer arithmetic automata. In *Proc. 25th Colloq. on Automata, Programming, and Languages (ICALP)*, volume 1443 of *Lecture Notes in Computer Science*, pages 152–163. Springer-Verlag, July 1998.

[Büc60] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift Math. Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[Büc62] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. Internat. Congr. Logic, Method and Philos. Sci. 1960*, pages 1–12, Stanford, 1962. Stanford University Press.

[Cob69] A. Cobham. On the base-dependence of sets of numbers recognizable by finite automata. *Mathematical Systems Theory*, 3:186–192, 1969.

[CVWY90] Constantin Courcoubetis, Moshe Y. Vardi, Pierre Wolper, and Mihalis Yannakakis. Memory efficient algorithms for the verification of temporal properties. In *Proc. 2nd Workshop on Computer Aided Verification*, volume 531 of *Lecture Notes in Computer Science*, pages 233–242, Rutgers, June 1990. Springer-Verlag.

[FR79] J. Ferrante and C. W. Rackoff. *The Computational Complexity of Logical Theories*, volume 718 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin-Heidelberg-New York, 1979.

[Hol97] Gerard J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, May 1997. Special Issue: Formal Methods in Software Practice.

[Kla91] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, San Juan, October 1991.

[KV97] O. Kupferman and M. Vardi. Weak alternating automata are not that weak. In *Proc. 5th Israeli Symposium on Theory of Computing and Systems*, pages 147–158. IEEE Computer Society Press, 1997.

[KVW00] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, March 2000.

[LASH] The Liège Automata-based Symbolic Handler (LASH). Available at `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`.

[Löd01] C. Löding. Efficient minimization of deterministic weak ω-automata, 2001. Submitted for publication.

[MH84] S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.

[MS97] O. Maler and L. Staiger. On syntactic congruences for ω-languages. *Theoretical Computer Science*, 183(1):93–112, 1997.

[MSS86] D.E. Muller, A. Saoudi, and P.E. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity. In *Proc. 13th Int. Colloquium on Automata, Languages and Programming*. Springer-Verlag, 1986.

[Rab69] M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Transaction of the AMS*, 141:1–35, 1969.

[Saf88] S. Safra. On the complexity of omega-automata. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, White Plains, October 1988.

[Sem77] A. L. Semenov. Presburgerness of predicates regular in two number systems. *Siberian Mathematical Journal*, 18:289–299, 1977.

[SKR98] T. R. Shiple, J. H. Kukula, and R. K. Ranjan. A comparison of Presburger engines for EFSM reachability. In *Proceedings of the 10th Intl. Conf. on Computer-Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*, pages 280–292, Vancouver, June/July 1998. Springer-Verlag.

[Sta83] L. Staiger. Finite-state ω-languages. *Journal of Computer and System Sciences*, 27(3):434–448, 1983.

[SVW87] A. Prasad Sistla, Moshe Y. Vardi, and Pierre Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.

[SW74] L. Staiger and K. Wagner. Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen. *Elektron. Informationsverarbeitung und Kybernetik EIK*, 10:379–392, 1974.

[Tho90] Wolfgang Thomas. Automata on infinite objects. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science – Volume B: Formal Models and Semantics*, chapter 4, pages 133–191. Elsevier, Amsterdam, 1990.

[VW86a] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the First Symposium on Logic in Computer Science*, pages 322–331, Cambridge, June 1986.

[VW86b] Moshe Y. Vardi and Pierre Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Science*, 32(2):183–221, April 1986.

[VW94] Moshe Y. Vardi and Pierre Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, November 1994.

[WB00] Pierre Wolper and Bernard Boigelot. On the construction of automata from linear arithmetic constraints. In *Proc. TACAS'2000*, volume 1785 of *Lecture Notes in Computer Science*, pages 1–19, Berlin, March 2000. Springer-Verlag.