Representing Periodic Temporal Information with Automata

Bernard Boigelot
Sébastien Jodogne
Pierre Wolper

Université de Liège
Dense or Discrete Time?

• Arguments for both, but discrete variables are essential for representing *periodicity*.

• $\exists x \in \mathbb{N} \ y = 24x$ means that $y$ has *period* 24.

• $\exists x \in \mathbb{R} \ y = 24x$ is trivially true.

• Nevertheless using variables over a dense domain also has its advantages. It would be interesting to have dense time and periodicity.

• The corresponding theoretical framework is the combined theory of the reals and integers, the subject of this talk.
Dealing with integers and reals

- We focus on the additive theory of the reals and integers (no multiplication of variable).

- It is well known that the first-order additive theory of the reals is decidable, and that the same is true of the additive theory of the integers (Presburger Arithmetic).

- For the combined theory, decidability is known but little has been done about developing usable decision procedures. Note that, as for the integers, quantifier elimination would require extending the theory.

- This talk is about using finite automata for deciding the combined theory of the reals and integers.
Handling the Integers with Automata

• Encode integers as binary strings.

• $n$-component vectors are encoded as strings over the alphabet $2^n$.

• A subset of $\mathbb{Z}^n$ is thus a language over the alphabet $2^n$.

• Finite automata are used to represent such languages.
Encoding Integers by Strings

**Principles:**

- Binary representation,
- Unbounded numbers,
- Most significant bit first.
- 2’s complement for negative numbers (at least $p$ bits for a number $x$ such that $-2^{p-1} \leq x < 2^{p-1}$).

**Examples:**

- $4 : 0100, 00100, 000100, \ldots$
- $-4 : 100, 1100, 11100, \ldots$
Encoding Vectors by Strings

Vectors are represented by using same length encodings of the components and reading them bit by bit.

An $n$-component vector is thus encoded by a word over $2^n$.

Examples:
(4, 3) is encoded by (100, 011), i.e. the word (1, 0)(0, 1)(0, 1) over $2^2$.

(−2, 12) is encoded by (11110, 01100), i.e. the word (1, 0)(1, 1)(1, 1)(1, 0)(0, 0).
Building Automata for Presburger Formulas

- One can fairly easily construct automata for equations and inequations.

- For general formulas, the automata are obtained by applying Boolean operations and projection to the automata obtained for basic formulas.

- To simplify operations, we use automata that accept all valid encodings of a given subset of $\mathbb{Z}^n$.

- This has been implemented (the LASH tool) and works well in practice.
Handling Reals with Automata

• Extending this representation to reals can be done quite naturally and yields a tool for handling the combined theory of integers and reals.

• Handling the reals is done by moving to automata on infinite words, which from a practical algorithmic point of view is quite problematic.

• This is surprising since the additive theory of the reals is easier to handle than the corresponding theory over the integers.

• Can this be explained? Yes! A very special type of infinite word automata are sufficient for handling the additive theory of the reals and integers.
Representing sets of Real Vectors by Automata: The Real Vector Automata (RVA)

- Reals are encoded in binary by infinite words built on the alphabet \{0, 1, \star\}. Negative numbers are encoded using 2’s complement.

**Examples:**

- \(L_2(3.5) = 0^+11 \star 1(0)^\omega \cup 0^+11 \star 0(1)^\omega\)
- \(L_2(-4) = 1^+00 \star (0)^\omega \cup 1^+011 \star (1)^\omega;\)

- Vectors with \(n\) real components are encoded by infinite words over the alphabet \{0, 1\}^n \cup \{\star\}.

- An RVA representing a set \(S \subseteq \mathbb{R}^n\) is a Büchi automaton accepting all the encodings of the vectors in \(S\).
Properties of RVA

- RVAs representing sets of the form \( \{ \vec{x} \in \mathbb{R}^n \mid \vec{a} \cdot \vec{x} \leq b \} \), with \( \vec{a} \in \mathbb{Z}^n, b \in \mathbb{Z} \), can easily be constructed;

- The set \( \mathbb{Z} \) is representable by an RVA (hence integerhood can be represented);

- Given RVAs representing sets \( S_1, S_2 \subseteq \mathbb{R}^n \), it is possible to algorithmically construct RVAs representing the sets
  
  \begin{itemize}
  \item \( S_1 \cup S_2, S_1 \cap S_2, S_1 \times S_2 \),
  \item \( \overline{S_1} = \mathbb{R}^n \setminus S_1 \),
  \item \( S_1_{\neq i} = \{ (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \mid (\exists x_i \in \mathbb{R})((x_1, \ldots, x_n) \in S_1) \} \);
  \end{itemize}

- It is decidable whether the set represented by an RVA is empty or not.
RVAs and arithmetic

It follows from the properties above that, for every subset of $\mathbb{R}^n$ definable in the first-order theory of $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$, one can algorithmically construct an RVA that represents it.

RVAs can thus be used as a tool to decide this theory.

**Problem:** Some of the algorithms for manipulating RVAs (in particular the complementation procedure) are not usable in practice.

**Solution:** We will show that

- The sets definable in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ satisfy some topological properties;
- automata representing such sets have a special structure; and this structure makes the use of much simpler algorithms possible.
Properties of Arithmetic Sets

• On the reals, Boolean combinations of linear (in)equalities define Boolean combinations of open and closed sets.

• The first-order theory of the reals admits quantifier elimination.

• Thus, only Boolean combinations of open and closed sets can be defined in the first-order theory of the reals.

• This should translate to properties of the automata accepting the encodings of these sets.

• However, we are looking at the first-order theory of the reals and integers for which no quantifier elimination result is known. Can we say something of the topology of the sets defined in this theory?
A Little Topological Background

Let $S$ be a set and $d(x, y)$ a distance defined on the elements of $S$.

- A neighborhood of a point $x \in S$ is a set $N_\varepsilon(x) = \{y \in S \mid d(x, y) < \varepsilon\}$, with $\varepsilon > 0$;

- A set $U \subseteq S$ is open if for every $x \in U$, there exists $\varepsilon > 0$ such that $N_\varepsilon(x) \subseteq U$;

- A set $U \subseteq S$ is closed if the set $S \setminus U$ is open;
• The Borel hierarchy defines a collection of classes of sets, that starts with the following.

  – The closed sets: $F$;

  – The open sets: $G$;

  – The countable unions of closed sets: $F_\sigma$;

  – The countable intersections of open sets: $G_\delta$;

  – The countable intersections of sets in $F_\sigma$: $F_{\sigma\delta}$;

  – ...
The Borel Hierarchy: A Graphical Representation

- $X \rightarrow Y : X \subset Y$;
- $\mathcal{B}(X)$: Boolean combinations of sets in $X$. 
Topological Properties of Arithmetical Sets

We consider the topology induced by the Euclidean distance

\[ d(\vec{x}, \vec{y}) = \left( \sum_{i=1}^{n} |x_i - y_i|^2 \right)^{1/2} \]

on the vectors of \( \mathbb{R}^n \).

**Theorem:** The sets definable in the first-order theory \( \langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle \) are in the topological class \( F_\sigma \cap G_\delta \).

**Proof:** If \( \varphi \) is a formula of \( \langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle \) then so is \( \neg \varphi \). It is thus sufficient to prove that every definable set is in \( F_\sigma \).

Let \( \varphi \) be a formula of \( \langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle \).
1. Let us replace each variable $x$ appearing in $\varphi$ by $x_I + x_F$, with

- $x_I$ the integer part of $x$;
- $x_F$ the fractional part of $x$.

Example:

$$(\exists x \in \mathbb{R})\varphi \quad \rightarrow \quad (\exists x_I \in \mathbb{Z})(\exists x_F \in \mathbb{R})$$

$$\quad \quad (0 \leq x_F < 1 \land \varphi[x/x_I + x_F])$$
2. Integer and fractional variables are then separated in the atomic formulas.

Example:

\[(x_I + x_F) = (y_I + y_F) + (z_I + z_F) \quad \rightarrow \quad (x_I = y_I + z_I \land x_F = y_F + z_F) \]
\[\lor (x_I = y_I + z_I + 1 \land x_F = y_F + z_F - 1)\]

3. The quantifiers are then distributed over the Boolean operators and unnecessary ones are eliminated.

Example:

\[(Qx_I \in \mathbb{Z})(\phi_I \alpha \phi_F) \quad \rightarrow \quad (Qx_I \in \mathbb{Z})(\phi_I) \alpha \phi_F,\]

where

- \(Q \in \{\exists, \forall\}, \alpha \in \{\land, \lor\},\)
- \(\phi_I\) only contains integer variables,
- \(\phi_F\) only contains fractional variables.
4. One then obtains a formula $\varphi$ of the form

$$B(\phi_I^{(1)}, \phi_I^{(2)}, \ldots, \phi_I^{(m)}, \phi_F^{(1)}, \phi_F^{(2)}, \ldots, \phi_F^{(m')}).$$

For each value $(a_1, a_2, \ldots, a_k) \in \mathbb{Z}^k$ of the free integer variable of this formula, each subformula $\phi_I^{(i)}$ is identically true or false. One thus has

$$\varphi \equiv \bigvee_{\bar{a} \in \mathbb{Z}^k} \left( (x_I^{(1)}, \ldots, x_I^{(k)}) = (a_1, \ldots, a_k) \right)$$

$$\wedge B(a_1, \ldots, a_k)(\phi_F^{(1)}, \ldots, \phi_F^{(m')}).$$

The formula $\varphi$ hence defines a countable union of Boolean combinations of open and closed sets, thus a set in $F_\sigma$. 
Automata and the Topology on Words

Consider the topology on infinite words induced by the distance

\[ d(w, w') = \frac{1}{|\text{commonprefix}(w, w')| + 1}. \]

Theorem [SW74, MS97] : The \( \omega \)-regular languages in the class \( F_\sigma \cap G_\delta \) are exactly those accepted by weak deterministic automata.

A weak automaton is a Büchi automaton whose set of states can be partitioned into sets \( Q_1, Q_2, \ldots, Q_m \) such that

- There exists a partial order \( \leq \) among these sets with the property that
  \[(\forall q \in Q_i, q' \in Q_j)(q \rightarrow^* q' \Rightarrow Q_j \leq Q_i);\]

- Each \( Q_i \) contains only accepting or nonaccepting states.
The previous result does not guarantee that any automaton built for a set in $F_\sigma \cap G_\delta$ is weak, but we have the following.

**Definition:** An automaton is *inherently weak* is none of its strongly connected components contains both accepting and nonaccepting cycles.

**Theorem:** Any deterministic Büchi automaton accepting an language in $F_\sigma \cap G_\delta$ is inherently weak.

**Proof:**
- For any language $L$ accepted by a deterministic automaton that is not inherently weak, $(\exists w_1)(\forall \varepsilon_1 > 0)(\exists w_2)(\forall \varepsilon_2 > 0)(\exists w_3)\ldots$
  - $d(w_i, w_{i+1}) < \varepsilon_1$ for $i = 1, 2, 3, \ldots$,
  - $w_1, w_3, w_5, \ldots \in L$, and $w_2, w_4, w_6, \ldots \notin L$.
- No language with this property can be accepted by a weak automaton.
The topologies on vectors and words are different. To use the fact that we are dealing with sets in $F_\sigma \cap G_\delta$ in the automaton context, we need the following.

**Theorem:** If $S \subseteq \mathbb{R}^n$ is a set in $F_\sigma \cap G_\delta$ (wrt Euclidean distance), then $L_r(S)$ is a set in $F_\sigma \cap G_\delta$ (wrt distance on words).

- The proof has to take into account the fact that every word is not necessarily an encoding of a vector.

- Dual encodings also prevent a direct mapping between the topologies.

- Nevertheless, the proof goes through for the class $F_\sigma \cap G_\delta$. 
Computing with RVAs

From the results we have just seen, it follows that:

**Theorem:** Any deterministic RVA representing a set defined by a formula of the theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ is inherently weak.

This property allows us to work with RVAs that are weak automata and makes possible to use algorithms that are specific to this class of automata.

- **Linear equations and inequations**: The algorithms proposed in [BRW98] produce weak automata.

- **Intersection, union, Cartesian product, projection**: One uses the corresponding operations on languages. The weak nature of the automata is preserved.
• **Complementation** :

1. The weak RVA is viewed as a *co-Büchi* automaton (a word is accepted if there is an execution of the automaton on that word that does not go infinitely often through accepting states).
2. For co-Büchi automata, there is a simple determinization procedure (see next slide).
3. The resulting deterministic automaton is complemented into a Büchi automaton.
4. The resulting automaton must be inherently weak and hence can easily be transformed into a weak automaton.

• **Satisfiability** : One checks whether the RVA has a reachable accepting strongly connected component.
Determinizing co-Büchi automata

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 $\omega$-language.

- $Q' = 2^Q \times 2^Q$.
- $q'_0 = (\{q_0\}, \emptyset)$.
- For $(S, R) \in Q'$ and $a \in \Sigma$, $\delta'$ is defined by
  - if $R = \emptyset$, then $\delta'((S, R), a) = (T, T \setminus F)$ where $T = \{q \mid \exists p \in S \text{ and } q \in \delta(p, a)\}$;
  - 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 $U = \{q \mid \exists p \in R \text{ and } q \in \delta(p, a)\}$.
- $F' = 2^Q \times \emptyset$. 
Conclusions

- These results do not introduce new algorithms, but show that known algorithms can be used in situations where this was a priori impossible.

- Weak deterministic automata have a canonical minimized form [Löding01]. There is thus a canonical form for RVAs.

- From a practical point of view, RVAs seem just as usable as automata representations of sets of integers.

- An implementation exists and confirms this.