Clone with Differentia over infinite alphabet is automatic

Mohamed Dahmoune

LACL, Paris-Est Créteil (UPEC)

mohamed.dahmoune@u-pec.fr

June 2012

Outline

Outline

Outline

Outline

Outline

1. Introduction
2. Decidability of clone
3. Decidability of differentia
4. Current and futur work

Introduction, ∞

- Set of possible states of a system is finite: its behavior and its specification can be modeled by finite automata
- Specification of infinite behavior: its specification can be modeled by special automata: Büchi, Muller, Rabin
- Domain system is infinite: its verification is in general unreliable
- Requires an extension of finite alphabet automata and regular models:
  - Register and Pebble automata [SF94, KF94, NSV01, Tan10]
  - $\mathfrak{M}$-automata [Bès08]
  - Data automata [BM06, BDM$^+$06]
  - Variable automata [GKS10]

Introduction, ∞, motivations

- Control systems with an infinite data source
- Modeling and|or verification of timed system
- Systems with integer parameters [BHM03]
- Log systems[1] [Via09, BHJS07]
- Semi-structured data and XML documents[2] [CFB$^+$02, BCC$^+$03]

[1] Log systems that store data belonging to an infinite domain.

[2] An XML document can be viewed as a tree whose leaves and branches are usually associated with values belonging to an infinite domain.

Introduction, ∞, motivations

Example (Timed system)

- $s_i \in \Sigma$: finite set of states
- $t_i \in \mathbb{R}^+$ or $\in \mathbb{N}$ discrete or continuous time
- Evolution of the system: word over infinite alphabet $(\Sigma \times \mathbb{R}^+)$ or $(\Sigma \times \mathbb{N})$
- $(s_0, t_0)(s_1, t_1) \dots (s_n, t_n)$

Example (Process management)

- $a_i \in \Sigma$: ensemble fini des actions
- $p_i \in \mathbb{N}$: numro du processus
- Action history: word over infinite alphabet $(\Sigma \times \mathbb{N})$
- $(a_0, p_0)(a_1, p_1) \dots (a_n, p_n)$

Introduction, relational structures, decidability

Definition

A relational structures $\mathfrak{A}$ is a tuple $(A; R_1, R_2, \ldots)$, where A is a non-empty set, called the **domain** of $\mathfrak{A}$, and each $R_i \subseteq A^{r_i}$ is a relation on A with arity r_i.

- $FO(\mathfrak{A})$: all **first-order** statements φ such as $\mathfrak{A} \models \varphi$
- Can we decide whether a statement φ is True or False in $\mathfrak{A}$?

Example

- $FO(\mathbb{N}; =, +)$ is decidable,
- $FO(\mathbb{N}; =, +, \times)$ is not.

Introduction, decidability, classical techniques

- Completeness
- Elimination of quantifiers
- Composition
  - Disjoint union of structures
  - Sum of structures
  - Product of structures
  - Power[3] of a structure
- Interpretation/Reduction
- Automata
  - Automaticity
  - $\mathfrak{M}$-automata [Bès08]

[3] $(\mathbb{N}; \times, =)$ is power of $(\mathbb{N}; +, =)$

Introduction, automaticity

Definition

Let σ be a finite alphabet. A relational structure $\mathfrak{A}$ is **automatic**, if there exists an injective mapping $c : A \longrightarrow \sigma^\star$ such that the images by c of A and all R_i are regular.

- $c(A) = \{c(x) \mid x \in A\}$ is a regular language over $\sigma^\star$.
- Each R_i: $c(R_i) = \{(c(x_1), c(x_2), \ldots, c(x_{r_i})) \mid (x_1, x_2, \ldots, x_{r_i}) \in R_i\}$ is a regular language over $(\sigma^\star)^{r_i}$ too.

Theorem

If a relational structure $\mathfrak{A}$ is automatic then its first-order theory $FO(\mathfrak{A})$ is decidable [BG00].

Introduction, $\mathfrak{M}$-automaticity

In [Bès08] Bès introduced the concept of $\mathfrak{M}$-automata, which is a natural notion of automata for finite words over an infinite alphabet.

Definition

Let Σ denotes an alphabet, finite or not, and let $\mathfrak{M}$ denotes a relational structure with domain Σ. An $\mathfrak{M}$-**automaton** is a finite n-tape synchronous non-deterministic automaton which reads finite words over $\Sigma \cup \{\#\}$ ($\#$ is a padding symbol). Transition rules are triplets of the form (q, φ, q'), where q, q' are states of the automaton, and $\varphi(x_1, \dots, x_n)$ is a first-order formula in the language of $\mathfrak{M}$.

Alexis Bès.

An Application of the Feferman-Vaught Theorem to Automata and Logics for Words over an Infinite Alphabet

Logical Methods in Computer Science, 4(1:8):1–23, 2008.

Introduction, $\mathfrak{M}$-automaticity

Definition

Let $n \geq 1$. A relation $X \subseteq (\Sigma^\star)^n$ is said to be $\mathfrak{M}$-**recognizable** if and only if there exists an $\mathfrak{M}$-automaton $\mathcal{A}$ with n tapes such that $X = L(\mathcal{A})$.

Definition

Let $\mathfrak{M} = (\Sigma; ...)$ and $\mathfrak{N} = (N; R_1, R_2, \dots)$ be two structures. We say that $\mathfrak{N}$ is $\mathfrak{M}$-**automatic** if there exists an injective mapping called coding $c : N \longrightarrow \Sigma^\star$ such that the images by c of N and all R_i are $\mathfrak{M}$-recognizable relations.

Theorem ([Bès08])

Let $\mathfrak{N}$ be a relational $\mathfrak{M}$-automatic structure. If $FO(\mathfrak{M})$ is decidable then $FO(\mathfrak{N})$ is decidable.

Introduction, clone, differentia

- Σ denumerable infinite alphabet
- *finite word* is a finite sequence of symbol (letter) from the alphabet Σ
- Σ^* denotes all finite words over Σ
- Σ^* denotes the integer finite words where $\Sigma = \mathbb{Z}$
- Σ^* denotes the natural finite words where $\Sigma = \mathbb{N}$
- ε denotes the empty word

In the following *word* refers always to *finite word*

Introduction, clone, differentia

Clone

- For a word x of Σ^*, the predicate $clone(x)$ is true, if and only if, x ends with two identical letters

$$clone(x) \Leftrightarrow x = uaa \text{ with } u \in \Sigma^* \text{ and } a \in \Sigma$$

- Shelah [She75] mentions a result of Stupp [Stu75] which was later improved by Muchnik concerning the iteration structure, in which, one define the clone predicate

Differentia

- For a word x of Σ^*, the predicate $diff(x)$ is true, if and only if, all letters of x are different (distinct)

$$diff(x) \Leftrightarrow x = x_1 x_2 \dots x_n \ : \ \forall i \forall j \ : \ i \neq j \Rightarrow x_i \neq x_j$$

Introduction, infinite clone differentia structure, motivation

Nice and interesting properties definable in $(\Sigma^*; \prec, clone, diff, less)$

Example (List of process ID)

$L \in \Sigma^*$ with $L = 3\ 4\ 2\ 1\ 1\ 2\ 0\ 7\ 7$

1. L ends with two different IDs
2. L contains two consecutive identical IDs
3. All IDs in the list L are identical
4. All IDs in the list L are distinct
5. L ends with k identical IDs
6. L contains k consecutive identical IDs
7. L is a decreasing sequence
8. L is an increasing sequence

Introduction, infinite clone differentia structure, motivation

Theorem ([Choffrut et Grigorieff, 2009])

For an infinit alphabet Σ*, the* $\exists\forall\forall$ *theory of the structure* $(\Sigma^*; \prec, \varepsilon, Pred, Eqlast)$ *is undecidable, where:*

- $Pred(a_1 \ldots a_n) = a_1 \ldots a_{n-1}$
- *EqLast denotes the binary relation* $\{(ua, va) \mid u, v \in \Sigma^*, a \in \Sigma\}$*, i.e. the set of pairs of words which end with the same letter*

Christian Choffrut and Serge Grigorieff.
Finite *n*-tape automata over possibly infinite alphabets: Extending a theorem of Eilenberg et al.
Theor. Comput. Sci., 410(1):16–34, 2009.

Outline

Automaticity of $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Definition

$(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$ denotes the infinite clone structure:

- $\Sigma^\star$ denotes the set of integer finite words, $\Sigma = \mathbb{Z}$.
- $x \prec y$ if and only if x is a strict prefix of y.
- $clone(x) \Leftrightarrow x = uaa$ with $u \in \Sigma^\star$ and $a \in \Sigma$.
- $less(x) \Leftrightarrow x = uab$ with $u \in \Sigma^\star$, $a, b \in \Sigma$ and $a < b$.
- $R_{p,q}(x) \Leftrightarrow x = uab$ with $u \in \Sigma^\star$, $a, b \in \Sigma$ and $|b - a| \equiv q[p]$ avec $p, q \in \Sigma$.
- $x \sim y \Leftrightarrow |x| = |y|$.
- $\oplus(x, y, z) \Leftrightarrow \begin{cases} x = x_1 x_2 \ldots x_i \ldots x_n \text{ and} \\ y = y_1 y_2 \ldots y_i \ldots y_n \text{ and} \\ z = (x_1 + y_1)(x_2 + y_2) \ldots (x_i + y_i) \ldots (x_n + y_n) \end{cases}$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{less}, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Example

- $x = [-1, 2, 5]$
- $y = [-1, 2, 5, 2, -3, 4, 4]$
- $x \prec y$, $\mathit{less}(x)$, $\mathit{clone}(y)$ are true.
- $R_{2,1}(x)$ is also true because $|5-2| \equiv 1[2]$.

Example

- $w = [3, 5, 2]$
- $x = [1, 3, 0]$
- $y = [4, 8, 2]$
- $z = [6, 0, 9]$
- $\oplus(w, x, y)$ is true but not $\oplus(x, y, z)$.

Automaticity of $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Theorem

The infinite clone structure $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$ *is* $\mathfrak{Z}$*-automatic, with* $\mathfrak{Z} = (\Sigma; 0, <, +)$.

Proof.

Let $\Sigma = \mathbb{Z}$. To prove that the infinite clone structure is $\mathfrak{Z}$-automatic, we use a coding d, that establish the $\mathfrak{Z}$-automaticity of the structure's domain and the $\mathfrak{Z}$-automaticity of the structure's predicates. This coding returns the differential of each word in $\Sigma^\star$. $\square$

Definition

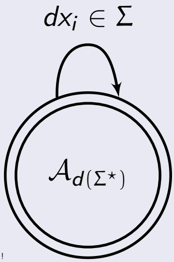
Let $x = x_1x_2 \dots x_n \in \Sigma^\star$ an integer word of length n. We call **differential** of x the integer word denoted by $d(x) = x_1 d_2 d_3 \dots d_n \in \Sigma^\star$ where $d_k = x_k - x_{k-1}$. By convention $d(\varepsilon) = \varepsilon$.

Example

- $x = [-1, 5, 2, 2, -3, 4, 4]$
- $d(x) = [-1, 6, -3, 0, -5, 7, 0]$

Automaticity of $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q \in \Sigma})$

Proof.



Domain's codes $d(\Sigma^\star)$ is $\mathfrak{Z}$-automatic. $\mathcal{A}_{d(\Sigma^\star)}$ is a $\mathfrak{Z}$-automaton that recognizes $d(\Sigma^\star)$ all codes over $\Sigma^\star$.

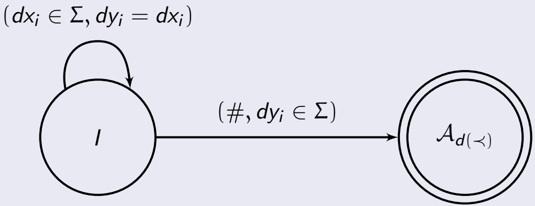
$\square$

Example

- $x = [-1, 5, 2, 2, -3, 4, 4]$
- $d(x) = [-1, 6, -3, 0, -5, 7, 0]$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{less}, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Proof.



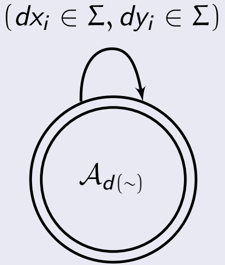
Prefix predicate's codes, $d(\prec)$, is $\mathfrak{Z}$-automatic. $x, y \in \Sigma^\star : x \prec y \Leftrightarrow d(x) \prec d(y)$. The $\mathfrak{Z}$-automaton $\mathcal{A}_{d(\prec)}$ is able to verify whether a word is prefix of another. □

Example

$$x = [1, 5, 7, 3, 4, 6] \qquad d(x) = [1, 4, 2, -4, 1, 2]$$
$$y = [1, 5, 7] \qquad d(y) = [1, 4, 2]$$

Automaticity of $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Proof.



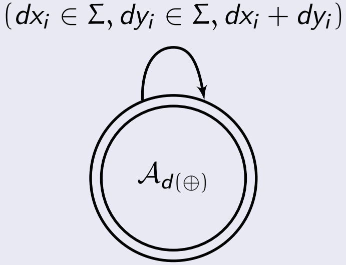
Equal length predicate's codes, $d(\sim)$, is $\mathfrak{Z}$-automatic. It is easy to verify that two words have the same length if and only if the code of the first and the second have the same length. For $x, y \in \Sigma^\star$, we have $x \sim y \Leftrightarrow d(x) \sim d(y)$. The $\mathfrak{Z}$-automaton $\mathcal{A}_{d(\sim)}$ is able to verify whether two words have the same length.

□

Example

- $x = [1, 5, 7]$ $d(x) = [1, 4, 2]$
- $y = [3, 4, 6]$ $d(y) = [3, 1, 2]$

Proof.



Sum letter by letter relation's codes, $d(\oplus)$, is also $\mathfrak{Z}$-automatic. $\mathcal{A}_{d(\oplus)}$ is a $\mathfrak{Z}$-automaton able to do the sum letter by letter. For $x, y, z \in \Sigma$, we have $\oplus(x, y, z) \Leftrightarrow \oplus(d(x), d(y), d(z))$.

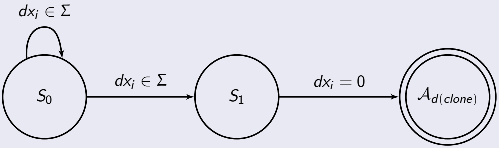
$\square$

Example

- $x = [1, 3, 5]$ $d(x) = [1, 2, 2]$
- $y = [2, 4, 4]$ $d(y) = [2, 2, 0]$
- $z = [3, 7, 9]$ $d(z) = [3, 4, 2]$

Automaticity of $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Proof.

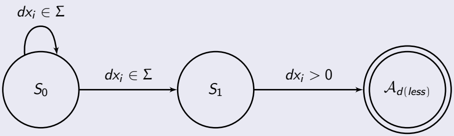


Clone predicate's codes, $d(clone)$, is $\mathfrak{Z}$-automatic. $clone(x) \Leftrightarrow d(x)$ ends with 0. $\mathcal{A}_{d(clone)}$ is a $\mathfrak{Z}$-automaton that checks the clone property. □

Example

- $x = [1, 5, 7, 3, 4, 4]$
- $d(x) = [1, 4, 2, -4, 1, 0]$

Proof.



Less predicate's codes, $d(less)$, is $\mathfrak{Z}$-automatic.

$less(x) \Leftrightarrow d(x)$ ends with $a > 0$.

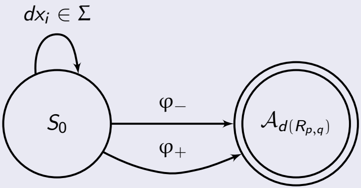
$\mathcal{A}_{d(less)}$ is a $\mathfrak{Z}$-automaton that checks if the code of an integer word ends with a positive integer. □

Example

- $x = [1, 5, 7, 3, 4, 14]$
- $d(x) = [1, 4, 2, -4, 1, 10]$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{less}, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Proof.



$$\varphi_+ : \quad dx_i \geq 0 \wedge \exists N \geq 0 \; \underbrace{N + N + \cdots + N}_{\underbrace{p}_{\mathit{constant}} \; \mathit{times}} + \underbrace{q}_{\mathit{constant}} = dx_i$$

$$\varphi_- : \quad dx_i < 0 \wedge \exists N \geq 0 \; \underbrace{N + N + \cdots + N}_{\underbrace{p}_{\mathit{constant}} \; \mathit{times}} + \underbrace{q}_{\mathit{constant}} + dx_i = 0$$

For $p, q \in \Sigma$, $d(R_{p,q})$, is $\mathfrak{Z}$-automatic.
The following $\mathfrak{Z}$-automaton is able to check the $R_{p,q}$ predicate. $\square$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{less}, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Corollary

The first-order theory of the infinite clone structure $(\Sigma^\star; \prec, \mathit{clone}, \mathit{less}, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$ is decidable.

Proof.

Direct consequence of the $\mathfrak{Z}$-automaticity of a structure is the decidability of its first-order theory. $\square$

Outline

Automaticity of $(\Sigma^\star; \prec, clone, less, \sim, \oplus, (R_{p,q})_{p,q\in\Sigma})$

Definition

$(\Sigma^\star; \prec, clone, diff)$ denotes the infinite clone differentia structure, where:

- $\Sigma^\star$ denotes the set of natural finite words, $\Sigma = \mathbb{N}$.
- $x \prec y$ if and only if x is a strict prefix of y.
- $clone(x)$ is true if and only if x ends with two identical letters: $clone(x) \Leftrightarrow x = uaa$ with $u \in \Sigma^\star$ and $a \in \Sigma$.
- $diff(x)$ is true if and only if all letters of x are different (distinct): $diff(x) \Leftrightarrow x = x_1 x_2 \ldots x_n$ with $x_1, x_2, \ldots, x_n \in \Sigma$ and $\forall i \forall j$ $i \neq j \Rightarrow x_i \neq x_j$.

Automaticity of $(\Sigma^{\star}; \prec, \mathit{clone}, \mathit{diff})$

Example

- $x = [7, 2, 5]$
- $y = [7, 2, 5, 2, 3, 4, 4]$
- $x \prec y$, $\mathit{clone}(y)$ are true.
- $\mathit{diff}(x)$ is true but not $\mathit{diff}(y)$.

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$

Theorem

The infinite clone differentia structure, $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$, is automatic.

Proof.

Let $\Sigma = \mathbb{N}$. To prove that the infinite clone differentia structure is automatic, we define a coding ν which establishes the rationality of the structure's domain, of the prefix, of the *clone* and of the differentia predicate. $\square$

Basic idea...

1. Let $D = \{x \in \Sigma^\star \mid \mathit{diff}(x)\}$
2. Any natural word can be divided into sub words of D.
3. Any word of D can be codded by a natural word and vice versa.

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$

Definition

The function s divides a natural word into a sequence of sub words of D.

$$\begin{array}{rcl} s : \Sigma^\star & \longrightarrow & D^\star \\ x = x_1 x_2 \ldots x_n & \longmapsto & s(x) = u_1 u_2 \ldots u_m \end{array}$$

- $m \leq n$
- a word in D stays the same under the function s, $x \in D \Leftrightarrow s(x) = x$

Example

$$[3,0,6,0,1,2,5,2,4,5,5,5,3,4,8,9] \xrightarrow{s} [3,0,6][0,1,2,5][2,4,5][5][5,3,4,8,9]$$

$$[5,2,6,8,3,6,4,8,5,4,4,4,5,6,8,5] \xrightarrow{s} [5,2,6,8,3][6,4,8,5][4][4][4,5,6,8][5]$$

$$[1,3,5,7,8,0,9,6,4,2] \xrightarrow{s} [1,3,5,7,8,0,9,6,4,2] \text{ one sub word}$$

Definition

For an ordered set $S = \{x_1 < x_2 < \cdots < x_i < \ldots\} \subseteq \Sigma$, we denote by $I_S^{x_i}$ the index of the element x_i in S. We have $I_S^{x_i} = i$.

We denote by δ the bijection which encodes every word in D by a natural word in $(\Sigma - \{0\})^\star$ where:

$$\begin{array}{rcl} \delta : D & \longrightarrow & (\Sigma - \{0\})^\star \\ x = x_1 x_2 \ldots x_i \ldots x_n & \longmapsto & \delta(x) = y_1 y_2 \ldots y_i \ldots y_n \\ y_i & = & I^{x_i}_{\Sigma - \{x_1, x_2, \ldots, x_{i-1}\}} \end{array}$$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$

Example

Let $x = [5, 3, 4, 2, 1]$. Then x is a natural word whose letters are distinct. We can verify that :

$$\begin{aligned}
\delta(x) &= \delta([5,3,4,2,1]) \\
&= [I^5_{\{0,1,2,3,4,5,\dots\}}, I^3_{\{0,1,2,3,4,\dots\}}, I^4_{\{0,1,2,4,\dots\}}, I^2_{\{0,1,2,3,\dots\}}, I^1_{\{0,1,3,\dots\}}] \\
&= [6,4,4,3,2]
\end{aligned}$$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$

Example

Inversely, we can get x from $[6, 4, 4, 3, 2]$:

- the 6^{th} element in $\Sigma = \{0, 1, 2, 3, 4, 5, 6, \ldots\}$ is $5 = x_1$
- the 4^{th} element in $\Sigma - \{x_1\} = \{0, 1, 2, 3, 4, 6, \ldots\}$ is $3 = x_2$
- the 4^{th} element in $\Sigma - \{x_1, x_2\} = \{0, 1, 2, 4, 6, \ldots\}$ is $4 = x_3$
- the 3^{rd} element in $\Sigma - \{x_1, x_2, x_3\} = \{0, 1, 2, 6, \ldots\}$ is $2 = x_4$
- the 2^{nd} element in $\Sigma - \{x_1, x_2, x_3, x_4\} = \{0, 1, 6, \ldots\}$ is $1 = x_5$

So $\delta^{-1}([6, 4, 4, 3, 2]) = [5, 3, 4, 2, 1]$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$

Example

$$\nu : \Sigma^\star \longrightarrow \{a, b, c\}^\star$$

$$\xrightarrow{\Sigma} \quad [5, 2, 6, 8, 3, 6, 4, 8, 5, 4, 4, 4, 5, 6, 8, 5]$$

$$\xrightarrow{s} \quad [5, 2, 6, 8, 3][6, 4, 8, 5][4][4][4, 5, 6, 8][5]$$

$$\xrightarrow{\delta} \quad [6, 3, 5, 6, 3][7, 5, 7, 5][5][5][5, 5, 5, 6][6]$$

$$\xrightarrow{-/+} \quad [+6, +3, -5, +6, +3][+7, -5, +7, +5][-5][-5][+5, -5, +5, +6][+6]$$

$$\xrightarrow{\mathit{separator}} \quad [+6, +3, -5, +6, +3, +0, -5, +7, +5, -0, -0, +0, -5, +5, +6, +0]$$

$$\xrightarrow{\{a,b,c\}} \quad [a^6 ba^3 ba^5 ca^6 ba^3 bba^5 ca^7 ba^5 bccba^5 ca^5 ba^6 bb]$$

Automaticity of $(\Sigma^\star; \prec, \mathit{clone}, \mathit{diff})$

Lemma

Universe's codes $\nu(\Sigma^\star)$ is regular.

Proof.

A word on $\{a, b, c\}$ corresponds to a code via ν:

- a sequence of (positive letters, a negative letter, positive letters, zero)
- a sequence of only positive letters

$$\nu(\Sigma^\star) \quad = \quad (P^\star NP^\star(b+c))^\star P^\star NP^\star bP^\star + P^\star \text{ where } P = a^+ b \ , \ N = a^+ c$$

$\square$

Example

$[5, 2, 6, 8, 3, 6, 4, 8, 5, 4, 4, 4, 5, 6, 8, 5]$

$[+6, +3, -5, +6, +3, +0, -5, +7, +5, -0, -0, +0, -5, +5, +6, +0]$

Lemma

Clone predicate's codes $\nu(\mathit{clone})$ is regular.

Proof.

The code of every non-empty *clone* word in $\Sigma^\star$ is a ternary word that ends with *cb*.

$$\mathit{clone}(x) \quad \Leftrightarrow \quad \nu(x) \in \nu(\Sigma^\star) \cap \{a, b, c\}^\star cb$$

□

Example

$[5, 2, 6, 8, 3, 6, 4, 8, 5, 4, 4]$

$[+6, +3, -5, +6, +3, +0, -5, +7, +5, -0, +0]$

$[a^6 ba^3 ba^5 ca^6 ba^3 bba^5 ca^7 ba^5 bcb]$

Automaticity of $(\Sigma^\star; \prec, clone, diff)$

Lemma

Differentia predicate's codes $\nu(diff)$ is regular.

Proof.

The code of every non-empty *diff* word in $\Sigma^\star$ is a sequence of only positive letters.

$$\nu(diff) = \nu(D) = P^\star = (a^+ b)^\star$$

$\square$

Example

$[5, 2, 6, 8, 3]$

$[+6, +3, +5, +6, +3]$

$[a^6 ba^3 ba^5 ca^6 ba^3 b]$

Automaticity of $(\Sigma^{\star}; \prec, \mathit{clone}, \mathit{diff})$

Corollary

The first-order theory of the infinite clone differentia structure $FO(\mathfrak{D})$ is decidable.

Proof.

Decidability of the first-order theory of the infinite clone differentia structure is a direct consequence of its automaticity. $\square$

Outline

Current and futur work

- Automaticity and decidability of some structures theory over countable infinite alphabet
- Developing most advanced coding that maintain the decidability and the automaticity of structures while adding predicates more complex and expressive, such as the predicate $diff_k$
- Studying the complexity
- Identify appropriate models of automata in order to identify everything that can be defined in such theory

Marco Brambilla, Stefano Ceri, Sara Comai, Piero Fraternali, and Ioana Manolescu.
Specification and design of workflow-driven hypertexts.
J. Web Eng., 1(2):163–182, 2003.

Mikolaj Bojanczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin.
Two-Variable Logic on Data Trees and XML Reasoning.
In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database system*, pages 10–19, Chicago, États-Unis, 2006. ACM Press.

Alexis Bès.
An application of the feferman-vaught theorem to automata and logics for words over an infinite alphabet.
Logical Methods in Computer Science, 4(1:8):1–23, 2008.

Achim Blumensath and Erich Grädel.
Automatic structures.
In *LICS*, pages 51–62, 2000.

Ahmed Bouajjani, Peter Habermehl, Yan Jurski, and Mihaela Sighireanu.
Rewriting systems with data.
In *Proceedings of the 16th international symposium on Fundamentals of Computation Theory*, pages 1–22, Berlin, Heidelberg, 2007. Springer-Verlag.

Ahmed Bouajjani, Peter Habermehl, and Richard Mayr.
Automatic verification of recursive procedures with one integer parameter.
Theor. Comput. Sci., 295:85–106, February 2003.

Mikolaj Bojanczyk and Anca Muscholl.
Two-variable logic on words with data.
In *In LICS'06*, pages 7–16. IEEE, 2006.

Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, and Maristella Matera.
Designing Data-Intensive Web Applications.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

O. Grumberg, O. Kupferman, and S. Sheinvald.

Variable automata over infinite alphabets.
In *LATA10*, volume 6031 of *LNCS*, pages 561–572. Springer, 2010.

Michael Kaminski and Nissim Francez.
Finite-memory automata.
Theor. Comput. Sci., 134(2):329–363, 1994.

Frank Neven, Thomas Schwentick, and Victor Vianu.
Towards regular languages over infinite alphabets.
In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science*, MFCS '01, pages 560–572, London, UK, 2001. Springer-Verlag.

Yael Shemesh and Nissim Francez.
Finite-state unification automata and relational languages.
Inf. Comput., 114(2):192–213, 1994.

Saharon Shelah.
The monadic theory of order.
Annals of Mathematics, 102:379–419, 1975.

J. Stupp.
The lattice model is recursive in the original model.
Manuscript, The Hebrew University, Jerusalem, 1975.

Tony Tan.
On pebble automata for data languages with decidable emptiness problem.
J. Comput. Syst. Sci., 76(8):778–791, 2010.

Victor Vianu.
Automatic verification of database-driven systems: a new frontier.
In *ICDT'09*, pages 1–13, 2009.