The Complexity of Semiautomatic Structures

Sanjay Jain, Singapore

Bakhadyr Khoussainov, Auckland

Frank Stephan, Singapore

Dan Teng, Singapore

Siyuan Zou, Singapore

Finite Automata

Recognising Multiples of Three

Three states: Remainders $\mathbf{0}$ (initial), $\mathbf{1}$, $\mathbf{2}$.
Update of state on digit: $(\mathbf{s}, \mathbf{d}) \mapsto (\mathbf{s} + \mathbf{d}) \bmod \mathbf{3}$;
for example, state $\mathbf{2}$ and input $\mathbf{8}$ give new state $\mathbf{1}$.
Accept numbers where final state is $\mathbf{0}$.

```
Input:   2 5 6 1 0 2 4 2 0 4 8
State: 0 2 1 1 2 2 1 2 1 1 2 1
Final Decision: Reject
```

Multiples of $\mathbf{p}$

States $\{\mathbf{0}, \mathbf{1}, \ldots, \mathbf{p} - \mathbf{1}\}$; initial state $\mathbf{0}$.
Update: $(\mathbf{s}, \mathbf{d}) \mapsto ((\mathbf{s} \cdot \mathbf{10}) + \mathbf{d}) \bmod \mathbf{p}$.
Accept numbers where final state is $\mathbf{0}$.

Automatic Structures - Example

Operations calculated or verified by finite automata
Automaton reads (from front or from end) inputs and has missing digits be replaced by symbol different from the alphabet. Here decimal adder with three states: n (no carry and correct), c (carry and correct), i (incorrect). Automaton works from the back to the front; start state and accepting state are n; states i and c are rejecting.

```
Correct Addition          Incorrect Addition
 # 2 3 5 8 . 2 2 5         3 3 3 3 . 3 3 #
 # 9 1 1 2 . # # #         # # 2 2 . 2 2 2
 1 1 4 7 0 . 2 2 5         # 1 5 5 . 5 5 2
n c n n c n n n n n       i i n n n n n n n
```

Alignment at the positions of “.”; if no alignment rule is given, alignment at the first member of the string; “#” are placed to fill up free positions after alignment is done.

Automatic Structures - Formal

In an automatic structure,

- the domain is coded as a regular set;
- each relation in the structure is recognised by a finite automaton reading all inputs at same speed;
- each function in the structure is verified by a finite automaton, that is, the automaton recognises the graph consisting of all tuples of valid combinations of inputs and outputs.

Examples: integers with addition and order; rationals with order, minimum and maximum; positive terminating decimal numbers with addition; finite subsets of the natural numbers with union and intersection and set-inclusion.

The inventors: Bernard R. Hodgson (1976, 1983); Bakhadyr Khoussainov and Anil Nerode (1995); Achim Blumensath and Erich Grädel (1999, 2000)

Groups and Order

An ordered group $(\mathbf{G}, +, <)$ satisfies the group axioms, that $<$ is transitive, that for each $\mathbf{x}, \mathbf{y} \in \mathbf{G}$ exactly one of $\mathbf{x} < \mathbf{y}$, $\mathbf{x} = \mathbf{y}$ and $\mathbf{y} < \mathbf{x}$ is true, that for each $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{G}$ the condition $\mathbf{x} < \mathbf{y}$ implies $\mathbf{x} + \mathbf{z} < \mathbf{y} + \mathbf{z}$ and $\mathbf{z} + \mathbf{x} < \mathbf{z} + \mathbf{y}$. A group is left-ordered if $\mathbf{x} < \mathbf{y}$ only implies $\mathbf{z} + \mathbf{x} < \mathbf{z} + \mathbf{y}$ but not the other condition.

Theorem [Jain, Khoussainov, Stephan, Teng and Zou 2014]. Every automatic ordered group is Abelian, even if only the group operation and not the ordering is automatic. However, the Klein bottle group with lexicographic order is a left-ordered automatic group.

Klein bottle group: Two generators $\mathbf{a}, \mathbf{b}$ with $\mathbf{a} \circ \mathbf{b} = \mathbf{b}^{-1} \circ \mathbf{a}$ and $\mathbf{a}^{\mathbf{i}}\mathbf{b}^{\mathbf{j}} < \mathbf{a}^{\mathbf{h}}\mathbf{b}^{\mathbf{k}} \Leftrightarrow \mathbf{i} < \mathbf{h} \vee (\mathbf{i} = \mathbf{h} \wedge \mathbf{j} < \mathbf{k})$.

Two-Dimensional Integer-Groups

Theorem [Jain, Khoussainov, Stephan, Teng and Zou 2014]. The ordered group $(\mathbb{Z} + \sqrt{3} \cdot \mathbb{Z}, +, <)$ is automatic.

Representation. Sequences $a_n \ldots a_1 a_0 . a_{-1} \ldots a_{-m}$ of coefficients in $\{-3, -2, -1, 0, 1, 2, 3\}$ representing $a = \sum_{k=-m,\ldots,n} u^k \cdot a_k$ aligned at the dot where $u = 2 + \sqrt{3}$.

Important Equation is $4u^k = u^{k+1} + u^{k-1}$.

Basic Automatic Algorithm. (Next Slide) Assume that $d_k \in \{-9, \ldots, 9\}$ for all k. This algorithm checks whether $d = \sum_k d_k \cdot u^k$ is negative, zero or positive.

Comparison. To check whether $a < b$, compute digits $d_k = b_k - a_k$ and determine the sign of d.

Addition. To check whether $a + b = c$, compute all digits $d_k = a_k + b_k - c_k$ and determine the sign of d.

Basic Automatic Algorithm.

Input $\mathbf{a_n a_{n-1} \ldots a_2 a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m}}$.
Initialisation $\mathbf{v = 0}$; $\mathbf{w = 0}$; $\mathbf{k = n + 1}$.
While $\mathbf{k > -m}$ and $\mathbf{v, w \in \{-30, -29, \ldots, 29, 30\}}$
Do Begin $\mathbf{k = k - 1}$; $\mathbf{(v, w) = (4v + w, -v + a_k)}$ End;
Represented Value is

$$\mathbf{v \cdot u^{k+1} + w \cdot u^k + \sum_{h<k} a_h \cdot u^h};$$

If $\mathbf{v > 30}$ Then Say "positive"; If $\mathbf{v < -30}$ Then Say "negative"; If $\mathbf{-30 \leq v \leq +30}$ Then Take Sign of $\mathbf{v \cdot u + w}$.

Verification. If $\mathbf{w}$ is out of range then so is $\mathbf{v}$.
If $\mathbf{v}$ is out of range then $\mathbf{v}$ determines the sign.

Algorithm can be carried out by finite automaton as $\mathbf{v, w}$ take only finitely many possible values.

Does Addition Determine Order?

Question [Jain, Khoussainov, Stephan, Teng and Zou 2014]. Is there an automatic copy $(\mathbf{A},+)$ of the integers with addition such that $<$ is not automatic?

Comment. This is equivalent to asking whether there is an automatic copy $(\mathbf{A},+)$ of the integers such that $\{\mathbf{x} \in \mathbf{A} : \mathbf{x} \geq \mathbf{0}\}$ is not regular.

Theorem [Jain, Khoussainov, Stephan, Teng and Zou 2014]. There is an automatic copy of $\{\mathbf{x} \cdot \mathbf{2}^{\mathbf{y}} \cdot \mathbf{3}^{\mathbf{z}} : \mathbf{x},\mathbf{y},\mathbf{z} \in \mathbb{Z}\}$ in which the addition is automatic but not the order.

The reason is that for every integers $\mathbf{a},\mathbf{k}$ there are integers $\mathbf{b},\mathbf{c},\mathbf{d}$ with $\mathbf{a}/\mathbf{6}^{\mathbf{k}} = \mathbf{b}/\mathbf{2}^{\mathbf{k}} + \mathbf{c}/\mathbf{3}^{\mathbf{k}} + \mathbf{d}$ and $\mathbf{0} \leq \mathbf{b} < \mathbf{2}^{\mathbf{k}}$ and $\mathbf{0} \leq \mathbf{c} < \mathbf{3}^{\mathbf{k}}$ where $\mathbf{b}$ is represented in binary and $\mathbf{c}$ is represented in ternary. The addition on numbers represented in that way is automatic but the order not.

Semiautomatic Structures

Automatic structures are quite restrictive and many structures cannot be represented.

Theorem [Tsankov 2011]. The additive group of the rationals is not automatic.

Semiautomatic structures try to represent more structures using automata. Idea: Instead of requiring that a function is an automatic function in all inputs, one requires only that the projected functions obtained by fixing all but one inputs by constants are automatic; similarly for relations including equality.

More formally, a structure like $(\mathbb{Q}, =, <; +)$ is semiautomatic if the sets and relations and functions before the semicolon are automatic and those after the semicolon are only semiautomatic.

Semiautomatic Groups and Rings

Theorem [Tsankov 2011]. A subring $(\mathbf{A}, +, =, <; \cdot)$ of the rationals is semiautomatic iff there is a positive natural number $\mathbf{p}$ such that every element in $\mathbf{A}$ is of the form $\mathbf{x} \cdot \mathbf{p}^{\mathbf{y}}$ for some $\mathbf{x}, \mathbf{y} \in \mathbb{Z}$.

Proposition.
The ordered group $(\mathbb{Q}, =, <; +)$ is semiautomatic.
The groups $(\mathbb{Q}, =; \cdot)$ and $(\mathbb{Z}^\infty, =; +)$ are semiautomatic.

Theorem.
If $\mathbf{a}$ is a fixed square-root of an integer then the field $(\mathbb{Q} + \mathbf{a} \cdot \mathbb{Q}; +, \cdot, =, <)$ is semiautomatic.

Open Question.
Are $(\mathbb{Q}, =, <; +, \cdot)$ and $(\mathbb{Q}, =; +, \cdot)$ semiautomatic?

Word Problem of Groups

Definition

Let a finite set of generators, say $\mathbf{A} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ of a semigroup be given and let it include the inverses (if they exist). Then $\{(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in \mathbf{A}^*$ and $\mathbf{v}, \mathbf{w}$ represent the same semigroup element$\}$ is called the word problem of the semigroup.

Theorem

The word problem of a finitely generated subgroup of a semiautomatic group is polynomial time decidable.

Theorem

There is a semiautomatic monoid where the word problem is undeciable.

Algorithm for Group

Let $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ be the generators. There are automatic functions $\mathbf{f_a}, \mathbf{f_b}, \mathbf{f_c}, \mathbf{f_d}$ mapping representatives $\mathbf{x}$ to representatives of $\mathbf{x} \circ \mathbf{a}$, $\mathbf{x} \circ \mathbf{b}$, $\mathbf{x} \circ \mathbf{c}$, $\mathbf{x} \circ \mathbf{d}$, repsectively. Each function has output at most $\mathbf{k}$ symbols longer than input, for some constant $\mathbf{k}$.

On input $\mathbf{x}, \mathbf{y}$, one checks $\mathbf{x} = \mathbf{y}$ by starting with a representative of the neutral element and then applying the functions for the symbols in $\mathbf{x}$ and then the functions for the inverses of symbols in $\mathbf{y}$, the latter in inverted order.

Then one evaluates the regular language which recognises all representatives of $\mathbf{0}$.

Each of $\mathbf{f_a}, \mathbf{f_b}, \mathbf{f_c}, \mathbf{f_d}$ runs in linear time and the length of the word in the memory increases at most by $\mathbf{k} \cdot |\mathbf{xy}|$, hence the overall time is quadratic. The final test of being the neutral element is linear.

Example for Semigroup

Let $\mathbf{B} \subseteq \{\mathbf{a}\} \cdot \{\mathbf{a}, \mathbf{b}\}^*$ be some set and consider the semigroup of all words $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}^*$ with concatenation. Furthermore, let π exchange $\mathbf{a}, \mathbf{b}$ and leave $\mathbf{c}$ unchanged. New equality $\equiv$: let $\mathbf{v_0 c v_1 c \ldots c v_k} \equiv \mathbf{w_0 c w_1 c \ldots c w_k}$ (where $\mathbf{v_h}, \mathbf{w_h} \in \{\mathbf{a}, \mathbf{b}\}^*$) iff $\mathbf{v_0} = \mathbf{w_0}$ and $\mathbf{v_k} = \mathbf{w_k}$ and $\mathbf{v_h} = \mathbf{w_h} \vee (\mathbf{v_h} = \pi(\mathbf{w_h}) \wedge \mathbf{w_h} \in \mathbf{B}) \vee (\mathbf{v_h} = \pi(\mathbf{w_h}) \wedge \mathbf{v_h} \in \mathbf{B})$ for all other $\mathbf{h}$.

Now for $\mathbf{u} \in \{\mathbf{a}\} \cdot \{\mathbf{a}, \mathbf{b}\}^*$, $\mathbf{u} \in \mathbf{B} \Leftrightarrow \mathbf{cuc} \equiv \mathbf{c}\pi(\mathbf{u})\mathbf{c}$.

Similarly equality $\equiv$ in the semigroup can be mapped back to membership of $\mathbf{B}$ with a polynomial time truth-table reduction.

All representatives of an a semigroup member form a finite set; the semigroup operation with a fixed element can be implemented as concatenation with a fixed word. Thus the monoid is semiautomatic.

Vector Space Functions

Theorem
The structures $(\mathbb{Q}^{\mathbf{n}}, \mathbb{Z}^{\mathbf{n}}; +, \cdot, =, <, \mathbf{F})$ are semiautomatic where $\mathbf{F}$ is the set of all functions from $\mathbb{Q}^{\mathbf{n}}$ to $\mathbb{Q}^{\mathbf{n}}$ computed by programs of finitely many lines with the following instruction set:

- Perform a fixed linear mapping changing some of the coordinates;
- Introduce a new coordinate and assign to it a value which is a fixed linear combination of the other coordinates;
- Jump forward to some fixed line number iff some fixed linear mappings to $\mathbb{Q}$ produces (a) an integer; (b) a positive number; (c) a fixed value; (d) a Boolean combination of (a)–(c); (e) unconditionally;
- Terminate with output of the first $\mathbf{n}$ coordinates.

Vector Space

Representation

Convoluted tuples of binary integers of form $\mathbf{conv(a_1, b_1, \ldots, a_n, b_n, c)}$ with $\mathbf{c > 0}$ and $\mathbf{b_m \in \{0, 1, \ldots, c-1\}}$ for all $\mathbf{m \in \{1, \ldots, n\}}$ representing $\mathbf{(a_1 + b_1/c_1, \ldots, a_n + b_n/c_n)}$.

Operations

The operations might be replaced by more basic ones for an implementation.

A value $\mathbf{a' + b'/c}$ is $\mathbf{0}$ iff $\mathbf{a', b'}$ are $\mathbf{0}$; is an integer iff $\mathbf{b' = 0}$; is non-negative iff $\mathbf{a' \geq 0}$.

Addition of two coordinates: if $\mathbf{b_i + b_j < c}$ then $\mathbf{a' = a_i + a_j}$ and $\mathbf{b' = b_i + b_j}$ else $\mathbf{a' = a_i + a_j + 1}$ and $\mathbf{b' = b_i + b_j - c}$.

Negating $\mathbf{a_i + b_i/c}$: If $\mathbf{b_i = 0}$ then $\mathbf{a' = -a_i}$ and $\mathbf{b' = 0}$ else $\mathbf{a' = 1 - a_i}$ and $\mathbf{b' = c - b_i}$.

Basic Operations – Continued

Multiplication of coordinate $\mathbf{a_i + b_i/c}$ with fixed $\mathbf{p \in \{1, 2, \ldots\}}$: Let $\mathbf{h}$ be minmal number with $\mathbf{p \cdot b_i \geq c \cdot h}$ (only choices $\mathbf{h = 0, 1, \ldots, c-1}$) and $\mathbf{a' = a_i \cdot p + h}$ and $\mathbf{b' = b_i \cdot p - h \cdot c}$.

Multiplication of some coordinates with fixed $\mathbf{1/p}$: Replace $\mathbf{c}$ by $\mathbf{c \cdot p}$, for non-changed coordinates replace $\mathbf{b_i}$ by $\mathbf{b_i \cdot p}$; for changed coordinates replace let $\mathbf{h}$ be remainder of $\mathbf{a_i}$ by $\mathbf{p}$ and replace $\mathbf{a_i}$ by $\mathbf{(a_i - h)/p}$ and $\mathbf{b_i}$ by $\mathbf{b_i + h \cdot a_i}$.

Forming sum of fixed $\mathbf{1/p}$ with $\mathbf{a_i + b_i/c}$: If $\mathbf{b_i \cdot p + c < c \cdot p}$ then $\mathbf{a' = a_i}$ and $\mathbf{b' = b_i \cdot p + c}$ else $\mathbf{a' = a_i + 1}$ and $\mathbf{b' = b_i \cdot p + c - c \cdot p}$; Replace $\mathbf{c}$ by $\mathbf{c \cdot p}$ and $\mathbf{b_j}$ by $\mathbf{b_j \cdot p}$ everywhere.

Forming sum of fixed $\mathbf{p}$ and $\mathbf{a_i + b_i/c}$ by letting $\mathbf{a' = a_i + p}$ and $\mathbf{b' = b_i}$.

Vector Operations and Multiplication

Theorem

For dimension $\mathbf{n} \geq 10$, there is no semiautomatic structure as in the previous theorem with the following mapping added to the list of permitted mappings the multiplication of coordinates.

Together with tests for being an integer, this permits to define a function whose range is a Diophantine set in the first coordinate and a $\mathbf{0}$ in the second. One can concatenate the resulting automatic function with repeatedly subtracting 1 and then doing a test for $\mathbf{0}$ in order to check whether a value $\mathbf{x}$ is in the Diophantine set. This reduces the Matiyasevich result and contradicts the decidability of the just indicated algorithmic approach. So the structure is not semiautomatic.

Vector Operations and Inversion

Theorem

There is no semiautomatic structure as in the previous theorem with the following mapping added to the list of permitted mappings the mapping $\mathbf{x \mapsto 1/x}$.

One can decode chain-fractions like $\mathbf{1/(a_0 + 1/(a_1 + (a_2 + r)))}$ by repeatedly inverting and separting out $\mathbf{a', b'/c}$ in $\mathbf{a' + b'/c}$ giving $\mathbf{a_0, a_1, a_2}$ as input for a Diophantine calculations on integers. Now one can exploit that the square of a positive integer $\mathbf{a \in \{1, 2, \ldots\}}$ is $\mathbf{1/(1/a - 1/(a+1)) - a}$; thus $\mathbf{a \cdot b = ((a+b)^2 - a^2 - b^2)/2}$. This permits to apply Matiyasevich's Technique and to show that the structure cannot be semiautomatic.

Theorem

The structure $(\mathbb{Q}; +, \cdot, /, =, <)$ is semiautomatic.

Summary

This talk gave an overview of the results presented last year at CSR 2014.

For groups and monoids, the complexity of the word problem of finitely generated submonoids was determined; it is in polynomial time for a group and can be arbitrarily complex for monoids.

While the linear operations on n-dimensional vector spaces are compatible with some enhanced operations and still permit to form a semiautomatic structure, a coding of Matiyasevich’s Theorem prevents this from being done when adding to this structure either multiplication of coordinates or forming the multiplicative inverse.