An Overview on Recent Results of Semiautomatic Groups and Semigroups

Frank Stephan, Singapore

Joint work with

Sanjay Jain, Singapore

Bakhadyr Khoussainov, Auckland

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: the automaton recognises the graph consisting of all valid (input,output)-tuples.

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).

Characterising automatic functions

Theorem [Case, Jain, Seah and Stephan 2013].
A function $f : \Sigma^* \to \Sigma^*$ is automatic iff there is a Turing machine with exactly one tape which computes f in linear time and which lets its output start at the same position where originally the input started.

Turing machine can use tape alphabet Γ much larger than Σ; time-bound linear in input-length.

Finite Automaton	Turing Machine
Goes in one direction	Goes forward and backward
Reads symbols	Reads and writes symbols
Finitely many states	Finitely many states; however, utilises tape as additional memory

Groups

A group $(\mathbf{G}, +)$ satisfies the following axioms:

- (Associativity) $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{G}\,[(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})]$;
- (Neutral element) $\mathbf{0} \in \mathbf{G} \wedge \forall \mathbf{x} \in \mathbf{G}\,[\mathbf{x} + \mathbf{0} = \mathbf{x} \wedge \mathbf{0} + \mathbf{x} = \mathbf{x}]$;
- (Inverse element) $\forall \mathbf{x} \in \mathbf{G}\,\exists \mathbf{y} \in \mathbf{G}\,[\mathbf{x} + \mathbf{y} = \mathbf{0}]$.

Abelian groups are commutative: $\forall \mathbf{x}, \mathbf{y} \in \mathbf{G}\,[\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}]$.

Examples are integers, rationals and reals with addition as well as finite groups (remainder groups):

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\cdot$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Abelian-by-Finite Groups

Definition. A group $(\mathbf{G}, +)$ is Abelian-by-Finite iff it has a Abelian subgroup $(\mathbf{A}, +)$ and a finite subset $\mathbf{F} \subseteq \mathbf{G}$ such that $\mathbf{G} = \{\mathbf{x} + \mathbf{y} : \mathbf{x} \in \mathbf{A} \wedge \mathbf{y} \in \mathbf{F}\}$.

Definition. A group $(\mathbf{G}, +)$ is finitely generated iff there is a finite set $\mathbf{F}$ such that $\mathbf{G}$ equals to the set of finite sums over the members of $\mathbf{F}$.

Theorem [Oliver and Thomas 2005]. A finitely generated group is automatic iff it is Abelian-by-finite.

Theorem [Nies and Thomas 2008]. Every finitely generated subgroup of an automatic group is Abelian-by-finite.

Question [Nies 2007]. Is every torsion-free automatic group Abelian-by-finite?

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

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

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

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

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

Basic Automatic Algorithm.

Input $\mathbf{a_n a_{n-1} \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}}$.
Initialisation $\mathbf{v = 0}$; $\mathbf{w = 0}$; $\mathbf{k = n + 1}$.
While $\mathbf{k > -m}$ and $\mathbf{v, w \in \{-30, -29, \dots, 29, 30\}}$
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 $\{\mathbf{x} \cdot \mathbf{2^y} \cdot \mathbf{3^z} : \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{Z}\}$ 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 $\mathbf{a}/\mathbf{6^k} = \mathbf{b}/\mathbf{2^k} + \mathbf{c}/\mathbf{3^k} + \mathbf{d}$ and $\mathbf{0} \leq \mathbf{b} < \mathbf{2^k}$ and $\mathbf{0} \leq \mathbf{c} < \mathbf{3^k}$ 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?

Automatic Group Operation

Theorem [Jain, Khoussainov, Stephan, Teng and Zou 2014]. If $(\mathbf{A}, +; =)$ is a semiautomatic group with automatic group operation then $(\mathbf{A}, +, =)$ is fully automatic. If $(\mathbf{A}, +; <, =)$ is an ordered semiautomatic group with automatic group operation then $(\mathbf{A}, +, <, =)$ is fully automatic ordered group.

Note that $\mathbf{x = y \Leftrightarrow \exists z\,[x + z = 0 \text{ and } y + z = 0]}$ and therefore comparing with the fixed element $\mathbf{0}$ is sufficient for testing equality. Similarly, in an ordered group, $\mathbf{x < y \Leftrightarrow \exists z\,[x + z = 0 \text{ and } 0 < y + z]}$.

Remark. This is not true for semigroups. For example, if $\mathbf{r > 0}$ is a nonrecursive real then $(\mathbb{N} + \mathbf{r} \cdot \mathbb{N}, +, =; <)$ is semiautomatic but the structure has no automatic copy. There is a semigroup $(\mathbf{A}, +; =)$ which is not automatic.

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 [Based on Known Methods].
The word problem of a finitely generated subgroup of a semiautomatic group is polynomial time decidable.

Theorem [Jain, Khoussainov, Stephan, Teng, Zou 2015].
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 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.

Cayley Automatic Groups

Definition [Kharlampovich, Khoussainov and Miasnikov 2011]. A group $(\mathbf{A}, =; \{\mathbf{x} \mapsto \mathbf{x} \circ \mathbf{a} : \mathbf{a} \in \mathbf{A}\})$ is Cayley automatic iff it is finitely generated, the domain is regular, the equality is automatic and for every $\mathbf{a} \in \mathbf{A}$, the mapping $\mathbf{x} \mapsto \mathbf{x} \circ \mathbf{a}$ is automatic. If a finitely generated group satisfies that $(\mathbf{A}, =; \circ)$ is semiautomatic then it is called Cayley biautomatic.

Theorem [Miasnikov and Šunić 2012].
There are Cayley automatic groups which are not Cayley biautomatic.
The conjugacy problem and the first-order theory of some Cayley automatic groups are undecidable.

Theorem [Jain, Khoussainov and Stephan 2016].
If $(\mathbf{A}, \circ)$ is a Cayley automatic group then $(\mathbf{A}; \circ, =)$ is semiautomatic.

Implication

Let a Cayley automatic representation $(\mathbf{B}, =; \{\mathbf{x} \mapsto \mathbf{x} \circ \mathbf{a} : \mathbf{a} \in \mathbf{A}\})$ be given.

Now $\mathbf{A} = \{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbf{B}\}$ with $(\mathbf{x}, \mathbf{y})$ representing $\mathbf{x}^{-1} \circ \mathbf{y}$.

Inversion: $(\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{y}, \mathbf{x})$.
Group operation with constants:
$(\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{x}, \mathbf{y} \circ \mathbf{a})$ represents $(\mathbf{x}^{-1} \circ \mathbf{y}) \circ \mathbf{a}$;
$(\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{x} \circ \mathbf{a}^{-1}, \mathbf{y})$ represents $\mathbf{a} \circ (\mathbf{x}^{-1} \circ \mathbf{y})$.
$(\mathbf{x}, \mathbf{y})$ equals $\mathbf{a}$ iff $\mathbf{x} \circ \mathbf{a}^{-1} = \mathbf{y}$ what can be checked for every fixed $\mathbf{a} \in \mathbf{B}$.

In summary: $(\mathbf{A}, \mathbf{x} \mapsto \mathbf{x}^{-1}; \circ, =)$ is semiautomatic and equals the given Cayley automatic group.

Separation: Open Problem for Finitely Generated Groups.
There are semiautomatic groups which are not finitely generated and thus not Cayley automatic.

Inversion

Proposition.
If $(\mathbf{A}; \circ, =)$ is semiautomatic, so is $(\mathbf{B}, \mathbf{x} \mapsto \mathbf{x}^{-1}; \circ, =)$ for a suitably coded copy $\mathbf{B}$ of $\mathbf{A}$.

Here $\mathbf{B} = \{\mathbf{x}, \mathbf{x}' : \mathbf{x} \in \mathbf{A}\}$ consist of two regular copies of $\mathbf{A}$ where for each $\mathbf{x} \in \mathbf{A}$, $\mathbf{x}'$ denotes the complement of $\mathbf{x}$.
The mappings $\mathbf{x} \mapsto \mathbf{x} \circ \mathbf{a}$ and $\mathbf{x} \mapsto \mathbf{a} \circ \mathbf{x}$ are extended from domain $\mathbf{A}$ to domain $\mathbf{B}$ by defining $\mathbf{x}' \circ \mathbf{a} = (\mathbf{a}^{-1} \circ \mathbf{x})'$ and $\mathbf{a} \circ \mathbf{x}' = (\mathbf{x} \circ \mathbf{a}^{-1})'$.
Furthermore, one tests whether $\mathbf{x}' = \mathbf{a}$ by testing whether $\mathbf{x} = \mathbf{a}^{-1}$, so the representatives of $\mathbf{a}$ form the regular set $\{\mathbf{x} : \mathbf{x} \in \mathbf{A}$ and $\mathbf{x} = \mathbf{a}\} \cup \{\mathbf{x}' : \mathbf{x} \in \mathbf{A}$ and $\mathbf{x} = \mathbf{a}^{-1}\}$.
The inversion maps $\mathbf{x} \in \mathbf{A}$ to $\mathbf{x}'$ and $\mathbf{x}'$ with $\mathbf{x} \in \mathbf{A}$ to $\mathbf{x}$. So $'$ is appended if it is not there and deleted if it is at the end of $\mathbf{x}$. The special symbol $'$ is at the end of $\mathbf{x}$ or absent.

Nilpotent Groups

A finitely generated group has nilpotency class $\mathbf{k}$ iff for all elements $\mathbf{a_0}, \mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_k}$ the sequence $\mathbf{b_0} = \mathbf{a_0}$ and $\mathbf{b_{h+1}} = \mathbf{b_h^{-1}} \circ \mathbf{a_h^{-1}} \circ \mathbf{b_h} \circ \mathbf{a_h}$ ends in a $\mathbf{b_k}$ such that $\mathbf{b_k}$ vanishes. Note that $\mathbf{b_h} \circ \mathbf{a_h} = \mathbf{a_h} \circ \mathbf{b_h} \circ \mathbf{b_{h+1}}$ and therefore one calls $\mathbf{b_{h+1}}$ also the communitator of $\mathbf{a_h}, \mathbf{b_h}$; groups of nilpotency class $\mathbf{1}$ are Abelian.

Theorem [Kharlampovich, Khoussainov and Miasnikov 2011]. Finitely generated groups of nilpotency class $\mathbf{2}$ are Cayley automatic.

Theorem [Jain, Khoussainov and Stephan 2016].
If $(\mathbf{A}, \circ)$ is finitely generated, has nilpotency class $\mathbf{3}$ and $\mathbf{B}$ is its commutator subgroup and $\bullet$ the restriction of $\circ$ to one operator being from $\mathbf{B}$ then $(\mathbf{A}, \mathbf{B}, \mathbf{x} \mapsto \mathbf{x^{-1}}, \bullet; \circ, =)$ is semiautomatic. For some choices of $\mathbf{A}$, the structure $(\mathbf{A}, \mathbf{B}, =, \bullet; \circ)$ is not semiautomatic.

Summary

This talk gave an overview of the results from papers at CSR 2014, CCR 2015 and a submitted paper 2016.

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.

For finitely generated groups, one has the implications

automatic $\Rightarrow$ Cayley biautomatic $\Rightarrow$ Cayley automatic $\Rightarrow$ semiautomatic

and for all groups one has the implications

Cailey biautomatic $\Rightarrow$ Cayley automatic $\Rightarrow$ semiautomatic $\Leftarrow$ automatic

where no further arrow holds. It is open whether every finitely generated semiautomatic group is Cayley automatic.