

CP3208 Progress Report

Qi Ji

April 2019

1 Project description

For this UROP project, I intend to investigate automatic and semiautomatic structures.

2 Progress

2.1 Initial direction – recursion theory

At the start of the project, I initially wanted to focus on recursion theory, possibly gaining some insights in learning theory from the perspective of recursion theory.

Along the way, over discussions with my supervisor, I became acquainted with the recently-invented notion of semiautomatic structures [JKSTZ17]. Having found this topic more interesting, I began investigating automatic and semiautomatic structures.

2.2 Regularity and finite automata

We skip the discussion of the characterisations of regular sets.

Relations over a base set $A \subset \Sigma^*$ are usually encoded as subsets of A^n , where n is the a -rity of our relation. Since there are several inputs, we use the standard method of encoding several strings synchronously. Given $a = a_0a_1a_2 \dots a_n$ and $b = b_0b_1 \dots b_m$ in A , the convolution $\text{conv}(a, b)$ is defined as $c_0c_1 \dots c_{\max(m, n)}$ where

- $c_k = \begin{pmatrix} a_k \\ b_k \end{pmatrix}$ if $k \leq m, k \leq n$,
- $c_k = \begin{pmatrix} a_k \\ \# \end{pmatrix}$ if $m < k \leq n$, and
- $c_k = \begin{pmatrix} \# \\ b_k \end{pmatrix}$ if $n < k \leq m$.

where $\#$ is a fixed padding character outside Σ . This process naturally identifies the set $A \times A$ with the set of all convolutions $\{\text{conv}(x, y) : x, y \in A\}$. This process is generalisable to higher arities. Since the alphabet is still finite, notions of regularity are preserved and we call a n -ary relation $R \subseteq A^n$ over A automatic if its representation is regular. Similarly a function is automatic if its graph is regular.

2.3 Group-like structures

Group-like algebraic structures are characterised by having a binary operation.

Definition 1. Let G be a set and let $\circ : G \times G \rightarrow G$.

The structure (G, \circ) is called a **semigroup** iff \circ is associative, that is for all $x, y, z, \in G$, $x \circ (y \circ z) = (x \circ y) \circ z$.

The structure (G, \circ, e) is called a **monoid** iff (G, \circ) is a semigroup and for all $x \in G$, $x \circ e = e \circ x = x$.

The structure (G, \circ, e) is called a **group** iff (G, \circ, e) is a monoid and for all $x \in G$, there exists $y \in G$ satisfying $x \circ y = e$.

Example 2. The structure $(\{1, 2, \dots\}, +)$ is a semigroup. Adding 0 to it gives us the monoid of natural numbers $(\mathbb{N}, +, 0)$. Considering its closure gives us the additive group of integers $(\mathbb{Z}, +, 0)$.

Definition 3. A semigroup (G, \circ) is called **finitely-generated** iff there exists finite subset $F \subseteq G$ where for each $x \in G$, there exists $y_1, y_2, \dots, y_n \in F$ satisfying $x = y_1 \circ y_2 \circ \dots \circ y_n$. The set F is set of generators for G . In the case where (G, \circ, e) is actually a group, for convenience we require that F is closed under inversion.

Remark. Definition 3 generalises to monoids and groups in the obvious way. Also since every element is expressible as a product of elements from F , we sometimes think of the finitely generated semigroup as words over F , and we can identify $y \in G$ with words in F^* . Also note that the structures in Example 2 are finitely-generated, with the generator 1.

Groups have been studied by mathematicians for centuries in their own right, but only recently have the connection between group theory and automata theory been made. [Hod76; Hod83; KN94; ECHLPT92]

2.4 Automatic Groups and Monoids

The low complexity of automata motivates the search for automatic presentations of various algebraic structures. For example, the word problem for groups in general is well-known to be undecidable. However, most reasonable formalisations of automatic groups will have their word problem solvable with a quadratic time complexity. [ECHLPT92; KKM11]

We first introduce the framework proposed by Hodgson [Hod76; Hod83] and Khoussainov and Nerode [KN94].

Definition 4. A semigroup (G, \circ) is called **fully automatic** iff

- G is regular over Σ^* with Σ being a finite alphabet,
- $\circ : G \times G \rightarrow G$ is an automatic function.

Fully automatic monoids and groups are defined analogously.

Remark. In the original literature the authors referred to their definition as just “automatic”. We follow the convention in [Ste18] and use the term “fully automatic” in the sense that the *full* semigroup operation is automatic. At the same time, we also disambiguate it from the following more popular definition, attributed to Epstein, Cannon, Holt, Levy, Paterson and Thurston. [ECHLPT92]

Definition 5. Let (G, \circ) be a semigroup generated by a finite subset $F \subseteq G$. The semigroup (G, \circ) is **automatic** iff

- G is a regular subset of F^* ,
- each $x \in G$ has exactly a unique representative in F^* , and
- for each $y \in G$, the multiplication map $(\circ y) : G \rightarrow G$ defined as $x \mapsto x \circ y$ is automatic.

The notion of having exactly 1 representative is equivalent to allowing multiple representatives but with equality being automatic. To see why, consider how we can unambiguously choose the lexicographically smallest element as the unique representative.

Example 6. Consider the semigroup of (Δ^*, \circ) where $2 \leq |\Delta| < \infty$. We can verify how this semigroup is automatic as the for each $y \in \Delta^*$ the concatenation map $x \mapsto xy$ is automatic. This group itself is not fully automatic, as in order to recognise the set $\{conv(x, y, z) : xy = z\}$ with a synchronous finite automata, we have to store y as we are reading in x and comparing it with z . The problem is that length of y is unbounded, so the graph of \circ cannot be recognised with finitely many states. In fact this group has no fully automatic representation.

This example illustrates how being fully automatic, as in Definition 4 is strictly stronger than being automatic, as in Definition 5.

Kharlampovich, Khousainov and Miasnikov were the first to formally consider the related concept of a Cayley automatic group [KKM11]. It is sometimes also called graph automatic because it considers the Cayley graph of a group as the automatic structure.

Definition 7. A finitely generated group G generated by F is **Cayley automatic** iff the following conditions hold for some finite alphabet Σ ,

- representatives of G form a regular subset H of Σ^* ,
- each $x \in G$ has a unique representative in H ,
- for each $y \in F$, the right multiplication by y map is an automatic map $H \rightarrow H$.

Remark. We note that Definition 5 is a special case of this, with the extra condition that natural representatives are chosen, that is $\Sigma = F$ the generating set. As a result Definition 7 defines a strictly bigger class of automatic groups.

Example 8. The Heisenberg group $\mathcal{H}_3(\mathbb{Z})$ defined as

$$\left\{ \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix} : a, b, c \in \mathbb{Z} \right\}$$

is Cayley automatic (6.6 in [KKM11]), but not automatic (8.1.1 in [ECHLPT92]).

Definition 9. G is **Cayley biautomatic** if the third condition in Definition 7 also holds for left multiplication.

It is interesting to note that there exists a Cayley automatic group that is not Cayley biautomatic. [MS12]

2.5 Ordinals

Definition 10 (Ordinals). A set α is an ordinal iff every $\beta \in \alpha$ is a subset of α and (α, \in) is a well-order.

Ordinals can be thought of as equivalence classes of well-ordered sets in set theory. They naturally describe how many times a process is iterated, possibly transfinitely many times. The class of all ordinals are well-ordered by \in , so if α, β are ordinals, $\alpha < \beta$ denotes $\alpha \in \beta$.

Description 11 (Ordinal arithmetic). The sum of two ordinals $\alpha + \beta$ expresses the order type of α placed before β , which is defined as the ordinal order-isomorphic to the ordered set $(\{0\} \times \alpha \cup \{1\} \times \beta, <)$ where $(0, \gamma) < (0, \gamma')$ iff $\gamma < \gamma'$ in α , $(1, \delta) < (1, \delta')$ iff $\delta < \delta'$ in β , and $(0, \xi) < (1, \xi')$ for all $\xi \in \alpha, \xi' \in \beta$.

The product of two ordinals $\alpha \cdot \beta$ can be thought of as β copies of α , which is defined as the ordinal order-isomorphic to the set $\beta \times \alpha$ equipped with dictionary ordering.

We note that the initial segment containing the first ω many ordinals is exactly the natural numbers, which add and multiply just like natural numbers.

2.6 General Automatic Structures

For general structures we consider the more general framework of Hodgson [Hod76; Hod83] and Khoussainov and Nerode [KN94].

Definition 12. A structure $(A, R_1, R_2, \dots, R_k, f_1, f_2, \dots, f_h)$ is automatic iff

- the set A is regular in Σ^* where Σ is some finite alphabet,
- all the relations R_1, R_2, \dots, R_k are automatic, and
- all the functions f_1, f_2, \dots, f_h are automatic.

We do not distinguish between structures that are automatic and structures that are merely isomorphic to an automatic structure.

This useful fact demonstrates the closure properties of automatic structures.

Theorem 13. *Any functions and relations that are definable using a formula of first-order in terms of automatic functions and relations is again automatic. Furthermore there is an effective procedure to construct the resultant automata from that used in the parameters to define the function or relation.*

Example 14. A fully automatic group (G, \circ) satisfying Definition 4 is an automatic structure.

Example 15. The structure $(\omega^3, +, <)$ is automatic. Any ordinal below ω^3 is of the form $\omega^2 \cdot c_2 + \omega \cdot c_1 + c_0$ with $c_0, c_1, c_2 \in \mathbb{N}$. We can express any ordinal $\alpha = \omega^2 \cdot a_2 + \omega \cdot a_1 + a_0 < \omega^3$ as $\text{conv}(a_0, a_1, a_2)$. Consider $\alpha, \beta \in \omega^3$, where $\alpha = \omega^2 \cdot a_2 + \omega \cdot a_1 + a_0$ and $\beta = \omega^2 \cdot b_2 + \omega \cdot b_1 + b_0$. The sum $\alpha + \beta$ can be given by

$$\alpha + \beta = \begin{cases} \omega^2 \cdot (a_2 + b_2) + \omega \cdot b_1 + b_0 & \text{if } b_2 > 0 \\ \omega^2 \cdot a_2 + \omega \cdot (a_1 + b_1) + b_0 & \text{if } b_2 = 0, b_1 > 0 \\ \omega^2 \cdot a_2 + \omega \cdot a_1 + (a_0 + b_0) & \text{if } b_2 = 0, b_1 = 0 \end{cases}$$

because for any $n, m \in \mathbb{N}$ with $n > m$, we have $\omega^m + \omega^n = \omega^n$. This means we can define addition on our representatives using the expression given above as

$$\text{conv}(a_0, a_1, a_2) + \text{conv}(b_0, b_1, b_2) = \begin{cases} \text{conv}(b_0, b_1, a_2 + b_2) & \text{if } b_2 > 0 \\ \text{conv}(b_0, a_1 + b_1, a_2) & \text{if } b_2 = 0, b_1 > 0 \\ \text{conv}(a_0 + b_0, a_1, a_2) & \text{if } b_2 = 0, b_1 = 0 \end{cases}$$

and since there exists an automatic representation of $(\mathbb{N}, +, <)$, the addition function is in fact automatic, as it only performs checks that are automatic in our presentation of \mathbb{N} . Because comparison of ordinals below ω^3 is first-order definable in terms of addition, $<$ is an automatic relation.

We prove half of Delhommé's characterisation of automatic ordinals. [Del04]

Theorem 16 (Delhommé). *Let α be an ordinal, $(\alpha, +, <)$ is automatic iff $\alpha < \omega^\omega$. Here the domain of $+$ is the set of all pairs (β, γ) with $\beta + \gamma < \alpha$.*

Proof. We only prove the easier half, that if $\alpha < \omega^\omega$, then $(\alpha, +, <)$ is automatic. First observe that Example 15 generalises to all $n \in \mathbb{N}$. Now any $\alpha < \omega^\omega$ is bounded above by ω^n for some natural number n , so $(\alpha, +, <)$ embeds in $(\omega^n, +, <)$ in a natural way. \square

2.7 Semiautomatic Structures

Seeking more general ways to utilise finite automata for representing non-automatic structures, Jain, Khousainov, Stephan, Teng and Zou proposed semiautomatic structures as a generalisation of Definition 12 [JKSTZ17].

Definition 17. Let $f : R^n \rightarrow R$ be a function. f is **semiautomatic** iff fixing $n - 1$ inputs, the resultant $R \rightarrow R$ function is automatic. The definition of a semiautomatic relation is analogous since we can view it as a $\{0, 1\}$ -valued function.

Definition 18. A structure $(A, f_1, f_2, \dots, f_k; g_1, g_2, g_h)$ is **semiautomatic** iff

- the set A is regular in Σ^* where Σ is some finite alphabet,
- all the functions f_1, f_2, \dots, f_k are automatic, and
- all the functions g_1, g_2, \dots, g_h are semiautomatic

where without loss of generality we treat relations as functions.

2.8 Semiautomatic Groups

Jain, Khoussainov and Stephan used semiautomaticity to describe group-like structures [JKS18]. They cited a related result by Tsankov [Tsa09].

Theorem 19. *The structure $(\mathbb{Q}, +, =)$ has no automatic presentation.*

Theorem 20 (21 in [JKSTZ17]). *The ordered group $(\mathbb{Q}, <, =; +)$ of rationals is semiautomatic.*

We note that \mathbb{Q} is not finitely generated, which leads on to this still open question.

Question 21. *Are all finitely generated semiautomatic groups Cayley automatic?*

2.9 Semiautomatic Rings and Fields

Theorem 22. *The ordered rings*

- $(\mathbb{Z}[\phi], +, <, =; \cdot)$ where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio, and
- $(\mathbb{Z}(\sqrt{n}), \mathbb{Z}, +, <, =; \cdot)$ for every natural number n

admit semiautomatic presentations.

The following result was only sketched in [JKSTZ17]. In fact the authors sketched a more general construction.

Theorem 23. *The polynomial ring $(\mathbb{N}[x]; +, \cdot, =)$ is semiautomatic.*

Proof. The construction is similar to that of Theorem 37 in [JKSTZ17]. We let $A \subseteq \Sigma^*$ be an semiautomatic representation of $(\mathbb{N}, +, <; \cdot)$ and let \oplus, \otimes, \bullet be symbols outside Σ .

We represent elements of $\mathbb{N}[x]$ as strings in $B = \{\epsilon\} \cup ((A \cdot \{\bullet\})^* \cdot A)$, where we encode nonzero polynomials like $c_n x^n + \dots + c_1 x + c_0$ as $c_0 \bullet c_1 \bullet \dots \bullet c_n$, where $c_n \neq 0$.

For each string $w \in C = \{\epsilon\} \cup ((B \cdot \{\oplus, \otimes\})^* \cdot B)$ we assign a value $val(w)$ in B as follows

- $val(\epsilon)$ is 0 which is represented by ϵ ;
- For $w \in B$, $val(w)$ is the base element with trailing zeroes stripped;
- $val(w \oplus \epsilon) = val(w)$;
- $val(w \otimes \epsilon)$ is 0 which is represented by ϵ ;
- $val(w \oplus v) = val(w) + val(v)$ when $v \in B$;
- $val(w \otimes v) = val(w) \cdot val(v)$ when $v \in B$ and v does not represent 0.

Note that val is not automatic, however its graph is a directed union of automatic functions. For each natural number n , $w \in C$, define $val_n(w) = val(w)$ if w represents a polynomial of degree n or less and all coefficients are below n . In the other case $val_n(w) = @$ if the degree of w is above n or some coefficient greater or equal to n is encountered. Notice that val_n is automatic as whenever its value goes to $@$, it remains at $@$ until a multiplication with zero occurs. In any other case the automaton only has to handle finitely many possible values. The finiteness property in here still holds because arithmetic in \mathbb{N} has neither additive inverses nor zero divisors.

For $v = \sum_{i \leq \deg(v)} c_i x^i$ we define $d(v) = \max(\deg(v), c_i)$. The rest of the argument is similar to that of Theorem 37 in [JKSTZ17]. We can check if $w \in C$ is equal to a fixed value v by comparing $val_{d(w)}(w)$ with v . We can also do addition with a fixed v as follows. If $val_{d(v)}(w) \neq @$ then we represent $v + w$ using the result, else we use $v \oplus w$. If $v \neq \epsilon$ the product $v \cdot w$ are represented by ϵ , otherwise we represent the product with $v \otimes w$. \square

2.10 Semiautomatic Ordinals

We begin by pointing out that $(\mathbb{N}, \cdot, =)$ has no automatic representation, which means that for any infinite ordinal α , there also cannot be any automatic representations of $(\alpha, \cdot, =)$.

I initially tried to directly apply the technique of Theorem 23 on $(\omega^\omega; +, \cdot, =)$, but the finiteness property in that proof was too strong, since for ordinals large coefficients could disappear, for instance $n + \omega = \omega$ and $\omega \cdot n \cdot \omega = \omega \cdot \omega$ for arbitrarily large $n \in \mathbb{N}$. After having some difficulty extending the result of Theorem 23, I tried another approach, incorporating semiautomaticity to Theorem 16.

Lemma 24. *Fix $n \in \mathbb{N}$, consider the automatic representation of $(\omega^n, +, <)$ similar to that in Example 15, also fix $d < \omega$, the following functions are automatic:*

- $\beta \mapsto \beta \cdot d$ the right-multiply by d map;
- the map that sends β to $\beta \cdot \omega$ if $\beta \cdot \omega < \omega^n$, $@$ otherwise.

Proof. Since d is fixed, right-multiplying by d can easily be accomplished by repeated addition, since addition in this representation is automatic by Theorem 16.

To multiply by ω on the right, for simplicity we demonstrate with the case $n = 4$. We denote the map needed as f , for any $\beta \in \omega^n$ represented as convolution of its coefficients $\text{conv}(a_0, a_1, a_2, a_3)$, we obtain using rules of ordinal multiplication

$$f(\beta) = \begin{cases} (\omega^3 \cdot a_3 + \omega^2 \cdot a_2 + \omega \cdot a_1 + a_0) \cdot \omega & \text{if } \beta \cdot \omega < \omega^4 \\ @ & \text{otherwise} \end{cases}$$

$$= \begin{cases} @ & \text{if } a_3 > 0 \\ \omega^3 & \text{if } a_3 = 0, a_2 > 0 \\ \omega^2 & \text{if } a_3 = 0, a_2 = 0, a_1 > 0 \\ \omega & \text{if } a_3 = 0, a_2 = 0, a_1 = 0, a_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

The @ case is only for error handling due to our assumptions on the domain of multiplication. So we can define a notion of right-multiplying by ω on our representatives as

$$f(\text{conv}(a_0, a_1, a_2, a_3)) = \begin{cases} @ & \text{if } a_3 > 0 \\ \text{conv}(0, 0, 0, 1) & \text{if } a_3 = 0, a_2 > 0 \\ \text{conv}(0, 0, 1, 0) & \text{if } a_3 = 0, a_2 = 0, a_1 > 0 \\ \text{conv}(0, 1, 0, 0) & \text{if } a_3 = 0, a_2 = 0, a_1 = 0, a_0 > 0 \\ \text{conv}(0, 0, 0, 0) & \text{otherwise} \end{cases}$$

we can easily see that only automatic checks on the input are needed and therefore f is automatic. To complete the proof we just have to observe that this construction works for any $n \in \mathbb{N}$. \square

Lemma 25. *Using the same automatic representation of $(\omega^n, +, <)$, for any $\beta \in \omega^n$ the map that sends γ to $\beta \cdot \gamma$ defined for all γ satisfying $\beta \cdot \gamma < \omega^n$ is automatic.*

Proof. First express the ordinals in normal form

$$\begin{aligned} \beta &= \omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \dots + \omega \cdot b_1 + b_0 \\ \gamma &= \omega^l \cdot c_l + \omega^{l-1} \cdot c_{l-1} + \dots + \omega \cdot c_1 + c_0 \end{aligned}$$

where $b_k, c_l > 0$. We then make the following general observation that

$$\begin{aligned} \beta \cdot \gamma &= \beta \cdot \omega^l \cdot c_l + \beta \cdot \omega^{l-1} \cdot c_{l-1} + \dots + \beta \cdot \omega \cdot c_1 + \beta \cdot c_0 \\ &= (\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \dots + \omega \cdot b_1 + b_0) \cdot \omega^l \cdot c_l \\ &\quad + (\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \dots + \omega \cdot b_1 + b_0) \cdot \omega^{l-1} \cdot c_{l-1} \\ &\quad + \dots \\ &\quad + (\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \dots + \omega \cdot b_1 + b_0) \cdot \omega \cdot c_1 \\ &\quad + (\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \dots + \omega \cdot b_1 + b_0) \cdot c_0 \\ &= \omega^{k+l} \cdot c_l + \omega^{k+l-1} \cdot c_{l-1} + \dots + \omega^{k+1} \cdot c_1 \\ &\quad + (\omega^k \cdot (b_k \cdot c_0) + \omega^{k-1} \cdot b_{k-1} + \dots + \omega^{b_1} + b_0) \cdot 1_{c_0 \neq 0} \end{aligned}$$

where $1_{c_0 \neq 0}$ is 1 if $c_0 \neq 0$ and 0 otherwise.

If our assumptions on the domain of γ holds, then $k + l < n$. We have

$$\begin{aligned} & \beta \cdot \text{conv}(c_0, c_1, \dots, c_l, 0, \dots, 0) \\ &= \text{conv}(\overbrace{0, \dots, 0}^{k+1 \text{ many}}, c_1, c_2, \dots, c_l, 0, \dots, 0) \\ &+ \begin{cases} 0 & \text{if } c_0 = 0 \\ \text{conv}(b_0, b_1, \dots, b_k - 1, b_k \cdot c_0, 0, \dots, 0) & \text{otherwise} \end{cases} \end{aligned}$$

We see that this function is automatic as β is already fixed so each b_i constant, therefore computing $b_k \cdot c_0$ given c_0 is automatic, in addition the checks are also automatic and the final addition is also automatic due to Theorem 16. \square

Theorem 26. *For any $\alpha < \omega^\omega$, the structure $(\alpha, +, <, =; \cdot)$ is semiautomatic. Similar to Theorem 16 we only consider the domain of $+$ to be the set of all pairs (β, γ) with $\beta + \gamma < \alpha$ and the domain of \cdot to be the set of all pairs (β, γ) with $\beta \cdot \gamma < \alpha$.*

Proof. Similarly to proof of Theorem 16 without loss of generality we consider $\alpha = \omega^n$ for some natural number n . We also use the same representation. Fix $\gamma < \omega^n$ and we want to show that for all β such that $\beta \cdot \gamma < \omega^n$, the map $\beta \mapsto \beta \cdot \gamma$ is automatic. We can express γ in normal form as

$$\gamma = \omega^n \cdot c_n + \dots + \omega \cdot c_1 + c_0,$$

then applying left-distributivity we have

$$\begin{aligned} \beta \cdot \gamma &= \beta \cdot (\omega^n \cdot c_n + \dots + \omega \cdot c_1 + c_0) \\ &= \beta \cdot \omega^n \cdot c_n + \dots + \beta \cdot \omega \cdot c_1 + \beta \cdot c_0. \end{aligned}$$

This expresses right-multiplication by γ as a composition of functions that are proven to be automatic in Lemma 24 and Theorem 16, which is still automatic. Additionally by our assumptions on the domain the @ case will not happen. This completes the proof for right-multiplication. The proof for left-multiplication was completed in Lemma 25. \square

3 Future Directions

Even at ω^ω , we can still try to find additional interesting presentations of the multiplication structure of ordinals. If successful, we can move on to larger ordinals such as ϵ_0 .

Another topic to investigate would be to consider a natural generalisation of Theorem 22

Question 27. *Let n be a natural number. Consider the ordered ring $(\mathbb{Z}(\sqrt[n]{n}), +, \cdot, <)$. What can be made semiautomatic? What can be made automatic?*

If successful, we can explore generalisations to possibly all m -th roots.

References

- [Del04] Christian Delhommé. ‘Automaticité des ordinaux et des graphes homogènes’. In: *Comptes rendus - Mathématique* 339.1 (2004), pp. 5–10.
- [EHLPT92] David B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. Levy, M. S. Paterson and W. P. Thurston. *Word Processing in Groups*. Natick, MA, USA: A. K. Peters, Ltd., 1992.
- [Hod76] Bernard Ralph Hodgson. ‘Théories décidables par automate fini’. PhD thesis. Département de mathématiques et de statistique, Université de Montréal, 1976.
- [Hod83] Bernard Ralph Hodgson. ‘Décidabilité par automate fini’. In: *Annales des sciences mathématiques du Québec* 7.1 (1983), pp. 39–57.
- [JKS18] Sanjay Jain, Bakhadyr Khoussainov and Frank Stephan. ‘Finitely generated semiautomatic groups’. In: *Computability* 7 (2–3 2018), pp. 273–287.
- [JKSTZ17] Sanjay Jain, Bakhadyr Khoussainov, Frank Stephan, Dan Teng and Siyuan Zou. ‘Semiautomatic structures’. In: *Theory of Computing Systems* 61.4 (2017), pp. 1254–1287.
- [KKM11] Olga Kharlampovich, Bakhadyr Khoussainov and Alexei Miasnikov. ‘From automatic structures to automatic groups’. In: *Groups, Geometry, and Dynamics* 8 (1 July 2011).
- [KN94] Bakhadyr Khoussainov and Anil Nerode. ‘Automatic Presentations of Structures’. In: *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC ’94, Indianapolis, Indiana, USA, 13-16 October 1994*. Ed. by Daniel Leivant. Vol. 960. Lecture Notes in Computer Science. Springer, 1994, pp. 367–392.
- [MŠ12] Alexei Miasnikov and Zoran Šunić. ‘Cayley Graph Automatic Groups Are Not Necessarily Cayley Graph Biautomatic’. In: *Language and Automata Theory and Applications*. Ed. by Adrian-Horia Dediu and Carlos Martín-Vide. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 401–407.
- [Ste18] Frank Stephan. ‘Methods and theory of automata and languages’. In: *Lecture Notes, School of Computing, National University of Singapore* (2018).
- [Tsa09] Todor Tsankov. ‘The additive group of the rationals does not have an automatic presentation’. In: *The Journal of Symbolic Logic* 76 (May 2009).