Left-orders of low computational complexity

INSTITUTO DE CIENCIAS MATEMÁTICAS

Co-funded by the Horizon 2020 programme of the European Union

"la Caixa" Foundation

Hang Lu Su

ICMAT-UAM

Outline

1. Left-orderable groups.
2. Formal languages.
3. Define computing left-orders as an algorithmic problem.
4. Overview of results. State the main intuitive ideas.

Related papers:

- *Formal language convexity in left-orderable groups.* IJAC, arXiv:1905.13001 (Su, 2020)
- *Regular left-orders.*, arXiv:2104.04475 (Antolín, Rivas, and Su, 2021)

I will focus more on the overall picture of our research and try to motivate it to the best of abilities.

Left-orderable groups

Def: a group G is *left-orderable* if there exists a strict total order $\prec$ on the elements of G which is invariant under left-multiplication:

$$g \prec h \iff fg \prec fh, \qquad \forall g, h, f \in G.$$

Ex: $\mathbb{Z}, \mathbb{Z}^2$: $x < y \iff z + x < z + y$.

Non-ex: any group with an element of finite order. If $g \in G$ such that $g^n = 1$, and wlog $1 \prec g$, then $1 \prec g \prec g^2 \prec \cdots \prec g^n = 1$ is a contradiction.

Ex: Free abelian groups, free groups, virtually poly-$\mathbb{Z}$, torsion-free one-relator groups, Thompson's groups, braid groups, RAAGs, all surface groups except the one for the projective plane are left-orderable.

Ex: For a countable group G, G is left-orderable iff it is isomorphic to a subgroup of the set of orientation-preserving homeomorphisms on $\mathbb{R}$.

Applications of left-orderable groups

Hanna-Neumann conjecture (proven): Let H and K be non-trivial free groups. Then $\mathrm{rk}(H \cap K) - 1 \leq (\mathrm{rk}(H) - 1)(\mathrm{rk}(K) - 1)$. Proof using left-orderable groups by Mineyev.

Zero divisor conjecture (not proven): Let R be a ring without zero divisors and G be torsion free. The conjecture says that the group ring RG has no zero divisors.

What is known: if G is left-orderable, then RG has no zero divisors.

Recent development: Giles Gardam disproved the unit conjecture, which implies the zero divisor conjecture.

Left-orderability as an algorithmic problem

Recall: The Word Problem. Given finitely generated $G = \langle X \mid R \rangle$, we write elements of G as words over X with an implicit evaluation map $\pi : X^* \to G$. We want to algorithmically determine whether $\pi(u) = \pi(v)$ for $u, v \in X^*$, or equivalently, whether $\pi(u)^{-1}\pi(v) = 1$. Therefore, we want to algorithmically determine if a word represents the identity.

Our problem: Given a left-orderable group, algorithmically determine whether $\pi(u) \prec \pi(v)$, and by left-invariance, this is equivalent to determining whether $1 \prec \pi(u)^{-1}\pi(v)$. Therefore, we want to algorithmically determine if a word represents an element greater than the identity.

Fact: Given a left-order $\prec$, the set $P = \{g \in G \mid g \succ 1\}$ uniquely defines $\prec$ and vice-versa. **Looking at the set of all positive elements is the same as looking at the left-order.**

We want to look at the words which map to P.

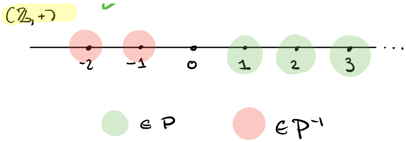
Positive cones I

Let's put some emphasis on this set of elements which are $\succ 1_G$. It is called a *positive cone* and can be defined independently of a left-order.

Roughly speaking, a positive cone is trying to capture the notion of additive positivity in your group.
$P \subset G$ is a *positive cone* for G if

- $PP \subseteq P$,
- $G = P \sqcup P^{-1} \sqcup \{1_G\}$.

Ex: $(\mathbb{Z}, +)$ admits *two* positive cones, P and P^{-1}.



Positive cones II

Ex: (continued) $(\mathbb{Z}, +)$ admits this natural left-order

$$\cdots < -1 < 0 < 1 < 2 < \ldots$$

The left order is equivalent to defining positive elements

$$P = \{z > 0 \mid z \in \mathbb{Z}\}$$

in the sense that

$$x < y \iff -x + y > 0.$$

Point: in general, a left-order $\prec$ defines a positive cone $P_{\prec} = \{g \in G \mid g \succ 1\}$ and P defines a left-order $\prec_P$ such that $g \prec h \iff g^{-1}h \in P$. This equivalence is straightforward to prove (but we won't do it).

End Part 1

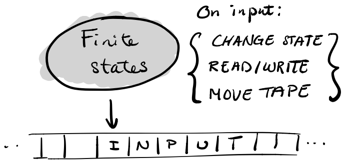
Recap: We want to study the positive cone algorithmically in terms of words over the generators evaluating to the positive cone. How will we do this?

We choose to do it in terms of formal languages.

Questions?

Formal languages and decidability

Formal languages capture degrees of decidability of a set of inputs.



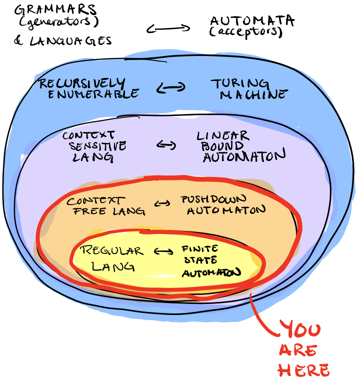
It is well-known that there exists a TM (with prescribed states and transitions) which simulates a computer, and that any algorithm on a computer can be represented by a TM.

A TM *accepts* a word ($\sim$ input) if it reaches a an accept state, and *halts* if there is no more transition after it finishes processing the word.

A TM is an "algorithm" which may not necessarily halt if the word is not accepted. The class of languages accepted by a TM is called *recursively enumerable*, and are *semi-decidable*. A TM which always halts is called an *algorithm*. The class of languages accepted by an algorithm are *decidable*.

Formal languages and Chomsky hierarchy

There's a hierarchy of automata (abstract machines) which process words (inputs).



If words accepted by algorithms are a starting point for decidability, then to be of lower complexity means to be accepted by a less sophisticated algorithm.

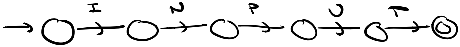
Regular languages

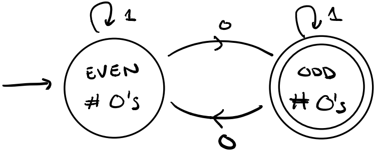
Regular languages are languages accepted by finite state automata (the finite state part of the TM).

Finite state automata (FSA) are directed graphs with

- edge-labels
- vertices ($\sim$ memory), some of which are special and labeled start states or and accept state.

Accepted words are words which form a path from the state start to the end state, ex 1: "INPUT"; ex 2: all binary strings with an odd number of 0's.

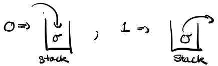




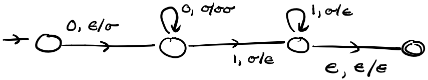
Context-free languages

Context-free languages are accepted by pushdown automata.
A pushdown automaton is a FSA with a stack, which allows you to have infinite memory but access it in a restricted way.

Ex: To accept the language $\{0^n 1^n \mid n \geq 1\}$, we need to record the number of 0's onto the stack and match it to the number of 1's. Stack symbol used is σ.



Formally, the stack is encoded in the edge labels. Vertices record which number we are looking at.



A one-counter automaton is a pushdown automaton with only one non-trivial stack symbol. The example we gave was one-counter.

End of Part 2

Summary: we defined the concept of decidability as the set of words accepted by algorithms (Turing machines that always halt). Formal languages are a collection of words with an attached degree of decidability; to be of low complexity means to be highly decidable because you are accepted by an automaton which is less sophisticated than a Turing machine which always halts.

For our research, we consider low complexity to be languages accepted by finite state automata or one-counter automata.

Questions?

Part 3: Defining computing left-orders as an algorithmic problem

Inspiration I: Word Problem and Formal Languages

The Word Problem has been studied in terms of decidability by taking $\pi^{-1}(1_G)$, all the words which represent the identity, as a formal language. The theory has been quite successful in the sense that there is a partial classification of groups with respect the complexity of $\pi^{-1}(1_G)$. If G if fg, the Word Problem is

- regular $\iff$ G finite, (Anisimov, 1971)
- one-counter $\iff$ G is virtually cyclic, (Herbst, 1991)
- context-free $\iff$ G virtually free, (Muller and Schupp, 1983)
- context-sensitive $\Longleftarrow$ G automatic, (Shapiro, 1994)
- decidable $\iff$ there exists fg simple group H and fp group K such that $G \leq H \leq K$. (Thompson, 1980)

Perhaps we can arrive at such a classification for left-orders?

Adapting the problem from WP to LO

Problem: Given a left-orderable group, classify as a formal language the set of words w such that $\pi(w) \in P$.

Obstruction: $\pi^{-1}(1_G)$ regular $\iff$ G finite implies that $\pi^{-1}(P)$ is never regular except for the trivial group (Antolín, Rivas, and Su, 2021). This means that we cannot have automata of lowest complexity which recognizes all positive words.

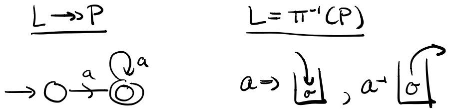
Choice: We choose to study languages L which surject to P, but such that L is not necessarily equal to $\pi^{-1}(P)$. L is regarded as a "normal form" but it does not necessarily biject with P.

Adapting the problem from WP to LO

Choice: We choose to study languages L which surject to P, but such that L is not necessarily equal to $\pi^{-1}(P)$. L is regarded as a "normal form" but it does not necessarily biject with P.

In some sense, this choice is quite natural and gets rid of the lower bound on complexity.

Ex: $\mathbb{Z} = \langle a \rangle$ has a set of normal forms $\{a^n \mid n \in \mathbb{Z}\}$.



A regular language for a positive cone is $\{a^n \mid n > 0\}$. $\pi^{-1}(P)$ is the set of words over $\{a, a^{-1}\}$ such that the exponent sum is positive, and this language is one-counter.

Inspiration II

The "normal form" idea is the same as with automatic groups.
Automatic groups are groups with

- a regular "normal form" (i.e. there exist L regular surjecting to G)
- an FSA which detects pairs of words differing by ≤ 1 generator $\implies$ fellow-travel property.

Automatic groups have been generalized by taking the normal forms to be some language of a different complexity, and maintaining a generalized form of the fellow-travel property.

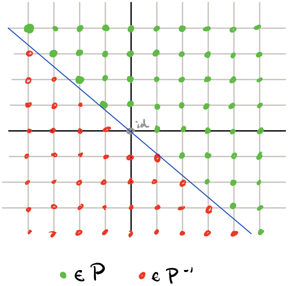
We study normal forms of positive cone elements of all complexity. Maybe one way of thinking about our research is that we are discovering left-orderable groups with an attached computational structure to compute words representing positive elements.

Adapting the problem from WP to LO II

So far, we have talked about computing positive elements of a left-order. But which left-order? Left-orders are highly non-unique. For each positive cone P, P^{-1} is another positive cone.

Ex: $\mathbb{Z}$, $P =$ positive integers, $P^{-1} =$ negative integers.

Ex: $\mathbb{Z}^2$ has **uncountably** many positive cones defined by half-spaces. Since every slope defines a different positive cone for $\mathbb{Z}^2$.



Adapting the problem from WP to LO II

Problem: Given a left-orderable group, classify as a formal language the set of words w such that $\pi(w) \succ 1$ or equivalently $\pi(w) \in P$.

Obstruction: while equality is unique in a group, a left-order is far from unique.

Full statement of problem: Given a finitely generated left-orderable group G, find a positive cone P such that there exists a formal language L surjecting to P, hopefully of minimal complexity across all possible positive cones.

We want to find the positive cone easiest to compute, then give you a way of computing it.

Of course in the absence of an optimal positive cone language, any will do...

End of Part 3

Summary: We looked at some results of the Word Problem in terms of formal languages, and also at automatic groups and discussed how the theory of those is similar to the theory we are trying to create for left-orderable groups.

Given a left-orderable group, we want to find a left-order that's easiest to compute, then give you a way of computing it.

Questions?

Overview of research results

With the time remaining, I will mostly give pictures and intuitive ideas without proofs.

Shorthand: *positive cone language* is a language which surjects to a positive cone.

Outline:

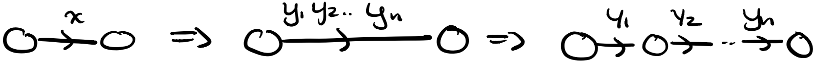
1. Closure properties of positive cone languages.
2. Geometric property of regular positive cones and its implications.
3. Combining groups together and their effect on positive cone complexity.

Closure properties of positive cone languages I

Main idea: positive cone languages have nice closure properties.

Thm: Having a $\mathcal{C}$ positive cone language ($\mathcal{C}$ = complexity in the Chomsky hierarchy) is independent of finite generating sets.

Intuition: If X and Y are two generating sets, replace X=arrows by Y-words in the finite state part of the automaton of complexity $\mathcal{C}$



Suppose that X and Y are generating sets and $x = y_1 \dots y_n$.

Takeaway: automata are essentially the same under different alphabets.

Closure properties of positive cone languages II

Main idea: positive cone languages have nice closure properties.

Thm: Positive cones language complexity is closed under extensions and wreath products (Antolín, Rivas, and Su, 2021). i.e. if Q and N have positive cone languages of complexity $\mathcal{C}$ then G has a positive cone language of complexity $\mathcal{C}$.

Intuition: Write words in normal form and use closure properties of formal languages.

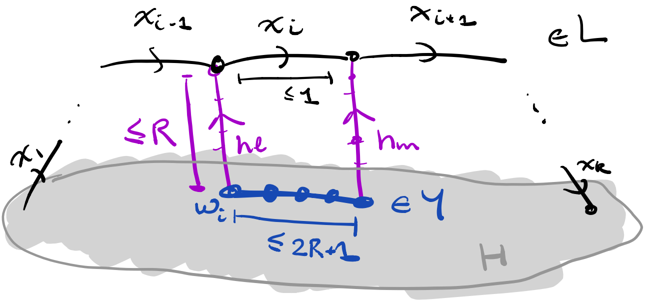
Extension: $G = QN$, $w_g = w_q w_n$. $P = P_Q N \cup P_N$
$\implies L = L_Q Y^* \cup L_N$ where Y is the generating set of N.

Wreath product: $G = N \wr Q$, $g = (q_1 n_1 q_1^{-1})(q_2 n_2 q_2^{-1}) \dots (q_n n_m q_m^{-1})p$, where $q_1 \succ_Q q_2 \succ_Q \cdots \succ_Q q_n$, $\implies$ some complicated positive cone language for G which is of complexity $\mathcal{C}$.

Closure properties of positive cone languages III

Main idea: positive cone languages have nice closure properties.

Thm: Regular positive cones are closed under finite index subgroups. (Su, 2020)



Intuition: Use the regular language for the positive cone of the overgroup to "fish" the regular positive cone language for the subgroup.

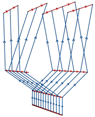
Closure properties of positive cone languages IV

Positive cone language complexity is positive cone dependent.

Thm: Baumslag-Solitar groups $B(1,q) = \langle a, b \mid aba^{-1} = b^q \rangle$ where $q > 1$ has both regular and non-regular positive cones (Antolín, Rivas, and Su, 2021).

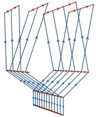
Regular: Every element can be written in normal form $g = a^n(a^{-m}b^k a^m)$. A positive cone is defined by elements where $k > 0$.

One-counter but not regular: There is a well-known isomorphism $B(1,q) \cong \mathbb{Z}[1/q] \rtimes \mathbb{Z} \implies$ we have a lexicographic order given by $P_{\mathbb{Z}}\mathbb{Z}[1/q] \cup P_{\mathbb{Z}[1/q]}$.



Geometric property of regular positive cones I

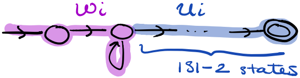
Why can't this set be represented by any regular language?



Def: A set is $P \subseteq G$ is *coarsely connected* if any path from P to P stays within an R-neighbourhood of P.

Thm: Regular positive cone $P \implies$ the set P is coarsely connected (Alonso, Antolin, and Rivas, 2020).

Intuition: Let $w \in L$, $w = x_1 \ldots x_n$, and $w_i = x_1 \ldots x_i$. For every prefix w_i, there exists u_i such that $w_i u_i \in L$ and $|u_i| \leq |S|$, the number of states of $L \implies$ every prefix can be connected to P with distance $\leq |S|$.



Trees $\sim$ not coarsely connected.

Geometric property of regular positive cones II

Just like for the Word Problem and for the theory of automatic group, positive cone formal language complexity has geometrical implications for left-orderable groups.

Tree-like positive cone $\sim$ not coarsely connected $\implies$ not regular.

Thm: Non-abelian free groups have no coarsely connected positive cones and hyperbolic groups with coarsely connected positive cones have to be very distorted in the sense of not being connected by quasi-geodesics (Alonso, Antolin, and Rivas, 2020).

Thm: More generally, free products have no regular positive cones (Hermiller and Sunic, 2017) and acylindrically hyperbolic groups have no regular quasi-geodesic positive cones (Su, 2020).

Combining groups together I

Combining groups give rise to new left-orders which are of lower complexity by using the new group structure. In other words, we can get non-trivial new orders out of combining groups.

This idea is pretty new ($\sim$ 2018), and therefore we have only tried taking the Cartesian product with $\mathbb{Z}$ and looking at its effects on the positive cone language complexity.

Recall trees $\sim$ not coarsely connected and that free products has no regular positive cones (Hermiller and Sunic, 2017)

Thm: Free products of groups with regular positive cones have one-counter-orders (Dicks and Sunic, 2020).

Thm: $(A * B) \times \mathbb{Z}$ has regular left-orders (Antolín, Rivas, and Su, 2021).

Intuition: Use $\mathbb{Z}$ as a stack.

Combining groups together II

Combining groups give rise to new left-orders which are of lower complexity by using the new group structure.

You can also observe this change at the topological level. You can take the space of left-orders as a topological space.

Thm: Let A, B be left-orderable. $A * B$ has no isolated orders (Deroin, Navas, and Rivas, 2014). In particular F_2 has no isolated orders.

Thm: $F_2 \times \mathbb{Z}$ has both isolated and non-isolated orders (Mann and Rivas, 2018).

Thm: Moreover, $F_2 \times \mathbb{Z}$ has a finitely generated positive cone (fg $\implies$ isolated). Examples of groups with fg positive cones are rare and this example is part of a new infinite family of groups with k-finitely generated positive cones for any $k \geq 3$ (Su, 2020). Whether such a family existed was a question left open by (Navas, 2011).

The end!

Bibliography I

Alonso, J., Y. Antolin, and C. Rivas (2020). “On the geometry of positive cones in finitely generated groups”. In: *ArXiv e-prints*. arXiv: 2001.10286.

Anisimov, A. V. (1971). “The group languages. (Russian. English summary)”. In: *Kibernetika (Kiev)* 4, pp. 18–24.

Antolín, Y., C. Rivas, and H.L. Su (2021). “Regular left-orders on groups”. In: arXiv: 2104.04475 [math.GR].

Deroin, B., A. Navas, and C. Rivas (2014). *Groups, Orders, and Dynamics*. arXiv: 1408.5805.

Dicks, W. and Z. Sunic (2020). “Orders on trees and free products of left-ordered groups”. In: *Canadian Mathematical Bulletin* 63 (2), pp. 335–347.

Herbst, T. (1991). “On a subclass of context-free groups”. In: *Theor. Informatics and Applications* 25, pp. 255–272.

Bibliography II

Hermiller, S. M. and Z. Sunic (2017). "No positive cone in a free product is regular". In: *IJAC* 27, pp. 1113–1120.

Mann, K. and C. Rivas (2018). "Group orderings, dynamics and rigidity". In: *Ann. Inst. Fourier* 68 (4), pp. 1399–1445.

Muller, D.E. and P.E. Schupp (1983). "Groups, the theory of ends and context-free languages". In: *J. Comput. System Sci.* 26, pp. 296–310.

Navas, A. (2011). "A remarkable family of left-ordered groups: Central extensions of Hecke groups". In: *Journal of Algebra* 328 (1), pp. 31–42.

Shapiro, M. (1994). "a note on context-sensitive languages and word problems". In: *Int. J. Alg. Computat.* 4, pp. 493–497.

Su, H.L. (2020). "Formal language convexity in left-orderable groups". In: *Internat. J. Algebra Comput* 30 (07), pp. 1437–1456.

Thompson, R.J. (1980). "Embedding into finitely generated groups which preserve the word problem". In: