Computability for totally disconnected locally compact groups

André Nies

Joint work with Alexander Melnikov

APM, Russia, November, 2021

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

What is

a totally disconnected locally compact group?

T.d.l.c. groups

- All topological groups will be separable.
- Note that each open subgroup U of a topological group G is also closed. If G is compact then U has finite index in G.
- A topological space is called totally disconnected if the clopen subsets form a basis.
- Van Dantzig's 1936 theorem says that if a totally disconnected group G is locally compact, then the identity of G has a neighbourhood basis consisting of compact open subgroups.
- One can use this to show that each t.d.l.c. group is topologically isomorphic to a closed subgroup of the symmetric group $\mathrm{Sym}(\mathbb{N})$: let G act by left translation on the compact open cosets.

T.d.l.c. groups in recent times

- Some research is analogous to the study of Lie groups, but in the disconnected setting.
- E.g. work on p-adic Lie groups: group theoretic characterization by Lazard (1965).
- T.d.l.c. groups G are frequently studied since the 1990s, e.g. by George Willis and his co-workers.
- There are important native notions, such as the scale function $s_G(x) = \{\min |V : x^{-1}Vx \cap V| : V \text{ is compact open subgroup}\}$.

Examples of t.d.l.c. groups

$(\mathbb{Q}_p, +)$ for a prime p

- This is the additive group of the p-adic numbers.
- The proper open subgroups are compact, and are all of the form $p^r\mathbb{Z}_p$ for some $r \in \mathbb{Z}$.

$\mathrm{Aut}(L_d)$ for $d \geq 3$.

- An undirected tree is a connected graph with no cycles.
- $G = \mathrm{Aut}(L_d)$ is the group of automorphism of the undirected tree L_d where each vertex has degree d (Tits, 1970).
- Each proper open subgroup of G is compact.
- Each compact subgroup of G is contained in the stabilizer of a vertex, or in the stabilizer of an edge.

Generalizing the computability notion for discrete, and for profinite groups

- The class of t.d.l.c groups contains both the profinite, and the countable discrete groups.
- For each of those subclasses, a generally accepted notion of computability exists.
- Can we generalize both of them simultaneously to the t.d.l.c. groups?

How to define computability for uncountable structures?

Broadly speaking, one uses a computable, countable "structure" of approximations to elements. Need to say what it means to approximate an element.

- $(\mathbb{R}, +, \times)$ is computable as the completion of $(\mathbb{Q}, +, \times)$ w.r.t. the Euclidean metric.
- $(\mathbb{Q}_p, +, \times)$ is computable, as the completion of $(\mathbb{Q}, +, \times)$ w.r.t. the p-adic metric.

Computability for profinite groups

For profinite groups, a definition of computability not directly involving a metric has been studied since it was introduced by La Roche 1981, Smith 1981.

A profinite group G is computable if $G = \varprojlim(A_i, \psi_i)$ for a computable sequence $(A_i, \psi_i)_{i\in\mathbb{N}}$ of finite groups and morphisms $\psi_i \colon A_{i+1} \to A_i$ which are onto.

- Concretely, this inverse limit is the closed subgroup of $\prod_i A_{i\in\mathbb{N}}$ consisting of the functions f such that $\psi_i(f(i+1)) = f(i)$ for each $i \in \mathbb{N}$.
- These functions f form the paths on a computable tree where at level i there are $|\ker\psi_i|$ many branchings.

Two definitions of computability for t.d.l.c. groups G that are equivalent.

Recall that in the uncountable setting, to define computability one needs a computable countable approximation structure.

- The first definition is by viewing G as a closed subgroup of $\mathrm{Sym}(\mathbb{N})$ and use a tree of finite injections as the approximation structure. This is natural if G is given as a group of automorphisms, e.g. $\mathrm{Aut}(L_d)$.
- The second definition is via the ordered groupoid of compact open cosets of G as an approximation structure. This is natural e.g. for $(\mathbb{Q}_p, +)$.

Definition 1 of computable t.d.l.c. groups:

via closed subgroups of $\mathrm{Sym}(\mathbb{N})$

Some tree terminology

$\mathbb{N}^*$ denotes the tree of strings with natural number entries.

Let $T \subseteq \mathbb{N}^*$ be a computable subtree without dead ends.

Let $[T]$ be the set of paths. Note that

$$[T] \text{ is compact} \iff \text{each level of } T \text{ is finite.}$$

For $\sigma \in T$ let $[\sigma]_T = \{X \in [T] \colon \sigma \prec X\}$, the paths extending σ.

- Consider strings $\sigma_i \in \mathbb{N}^*$, $i = 0, 1$ of the same length $N \le \infty$.
- By $\sigma_0 \oplus \sigma_1$ we denote the string of length $2N$ which alternates between σ_0 and σ_1.
- E.g. $\sigma_0 = (1, 3), \sigma_1 = (4, 0)$, yields $(1, 4, 3, 0)$.

A computable presentation of $\mathrm{Sym}(\mathbb{N})$

$\mathrm{Sym}(\mathbb{N})$ is the group of permutations of $\mathbb{N}$.

The approximation structure for $\mathrm{Sym}(\mathbb{N})$ is the computable tree $\mathit{Tree}(\mathrm{Sym}(\mathbb{N})) = \{\sigma \oplus \tau \colon$

σ, τ are 1-1 $\wedge\ \sigma(\tau(k)) = k\ \wedge\ \tau(\sigma(i)) = i$ whenever defined$\}$.

- The paths of $\mathit{Tree}(\mathrm{Sym}(\mathbb{N}))$ can be viewed as the permutations of $\mathbb{N}$, paired with their inverses:

$$[\mathit{Tree}(\mathrm{Sym}(\mathbb{N}))] = \{f \oplus f^{-1} \colon \ f \in \mathrm{Sym}(\mathbb{N})\}.$$

- The operations of $\mathrm{Sym}(\mathbb{N})$ are given by computable functions on $\mathit{Tree}(\mathrm{Sym}(\mathbb{N}))$, such as $\sigma \oplus \tau \mapsto \tau \oplus \sigma$ for the inverse.

Computable closed subgroups of $\mathrm{Sym}(\mathbb{N})$

Recall that $\mathrm{Sym}(\mathbb{N})$ is viewed as the set of paths of
$T = \mathit{Tree}(\mathrm{Sym}(\mathbb{N})) = \{\sigma \oplus \tau :$

$$\sigma, \tau \text{ are 1-1} \wedge \sigma(\tau(k)) = k \wedge \tau(\sigma(i)) = i \text{ whenever defined}\}.$$

Definition (Essentially Greenberg, Melnikov, N., Turetsky 2019)

A closed subgroup G of $\mathrm{Sym}(\mathbb{N})$ is computable if its corresponding tree, namely $\mathit{Tree}(G) = \{\eta \in \mathit{Tree}(\mathrm{Sym}(\mathbb{N})) \colon [\eta] \cap G \neq \emptyset\}$ is computable.

- The closed subgroups of $\mathrm{Sym}(\mathbb{N})$ are just the automorphism groups G of structures M with domain $\mathbb{N}$.
- If one can decide whether a finite injection on M can be extended to an automorphism, such a group G is computable.
- E.g., $G = \mathrm{Aut}(\mathbb{Q}, <)$ is computable because one can decide

Definition (Computably locally compact trees)

We say that a computable tree T without leaves is computably locally compact if

- $\{\sigma \in T \colon [\sigma]_T \text{ is compact}\}$ is decidable, and
- there is a computable function $h(\sigma, i)$ such that:
  if $[\sigma]_T$ is compact and $\rho \in T$ extends σ, then

  $$\rho(i) \leq h(\sigma, i) \text{ for each } i < |\rho|.$$

  That is, $[\sigma]_T$ is compact in an effective way.

Definition (First definition of computably t.d.l.c. groups)

Let G be a computable closed subgroup of $\mathrm{Sym}(\mathbb{N})$ that is t.d.l.c. We say that G computably t.d.l.c. if $\mathit{Tree}(G)$ is computably locally compact.

Some t.d.l.c. groups that are computable according to this definition

Let $d \geq 3$. The t.d.l.c. group $G = \mathrm{Aut}(L_d)$ is computably t.d.l.c..

- One can decide whether a finite injection α on L_d can be extended to an automorphism: check whether it preserves distances. Each $\eta \in \mathit{Tree}(\mathrm{Sym}(\mathbb{N}))$ corresponds to such an injection. So one can decide whether $\eta \in \mathit{Tree}(G)$.
- $[\eta]_{\mathit{Tree}(G)}$ is compact for every nonempty such string η:
  If η maps x to y, then it maps a ball $B_n(x)$ in L_d to $B_n(y)$.

  This yields a computable bound $h(\eta, i)$ as required.

Thus, $\mathit{Tree}(G)$ is computably locally compact.

Computable $\mathbb{N}$-valued functions on computably t.d.l.c. groups

Definition

A function $s : G \to \mathbb{N}$ is computable if there is an oracle Turing machine that using a path $X \in [\mathit{Tree}(G)]$ on its oracle tape (and the empty string as input) halts with output $s(X)$.

Intuitively, the machine can use as much of its oracle X as it likes, but it has to eventually come up with a value $s(X)$, using only an initial segment of X.

Scale function

The scale function on G (Willis, 1994) is defined by

$$s_G(x) = \{\min |V : x^{-1}Vx \cap V| : V \text{ is compact open subgroup}\}.$$

This tells us how much at least conjugation by x "moves" a compact open subgroup. Willis (1994) has shown that the scale function is continuous. This is a necessary condition for being computable.

- Scale function is trivial if there is a compact open normal subgroup, e.g. for discrete, and profinite groups.
- Scale function is computable for $\mathrm{Aut}(L_d)$:
  If x fixes a vertex, or inverts an edge , the scale is 1.
  If x translates along a geodesic then $s(g) = (d-1)^\ell$ where ℓ is the translation distance.
- s_G is computably approximable from above, by going through the possible V and reducing the approximation when necessary.

Can the scale function be noncomputable?

Question (Open at this stage)
Is there a computably t.d.l.c. group G such that the scale function s_G is noncomputable?

- An example would likely consist of a computably t.d.l.c. group G and a uniformly computable sequence $(h_i)_{i \in \mathbb{N}}$ of elements of G such that the function $i \to s_G(h_i)$ is noncomputable.
- G can't be discrete, or profinite.

Definition of computable t.d.l.c. groups 2:

via approximation groupoids

Approximation groupoids

Groupoid: small category where each morphism has an inverse. Equivalently, usual group axioms but the group operation is partial.

Let G be t.d.l.c.. Define a groupoid $\mathcal{W}(G)$ that is also a lower semilattice. Domain: the open compact cosets of G, and $\emptyset$.

A, B, C denote such cosets. U, V, W denote compact open subgroups.
$A: U \to V$ means that A is a right coset of U and a left coset of V.

If $A: U \to V$ and $B: V \to W$ then $A \cdot B: U \to W$.

- $(\mathcal{W}(G), \cdot)$ is a groupoid. E.g., if $A: U \to V$ then $A^{-1}: V \to U$.
- $(\mathcal{W}(G), \subseteq, \emptyset)$ is a lower semilattice with least element $\emptyset$, since the intersection of two cosets is empty, or again a coset.

$\mathcal{W}(G)$ satisfies the axioms of "inductive groupoids";
see e.g. Lawson's 1998 book Inverse semigroups.

Provenance of approximation groupoids

- The notion of approximation groupoids goes back to an idea of Tent, that appeared in a paper with Kechris and Tent in J. Symb. Logic 2018.
- The idea was further elaborated in a paper with Tent and Schlicht on the complexity of the isomorphism problem for oligomorphic groups that has just appeared in J. Math. Logic 2021. There we introduced the term "coarse group".
- Coarse groups are given by a ternary relation on open cosets, saying that $AB \subseteq C$, where A, B, C are cosets.
- We will see that for the applications to computable structure theory we have in mind, it is necessary to have the groupoid and lower semilattice structure explicitly.

Computable t.d.l.c. groups (Def. 2)

Recall that the approximation groupoid of a t.d.l.c. group G is the countable structure $(\mathcal{W}(G), \cdot, \subseteq, \cap, \emptyset)$ on the compact open cosets together with $\emptyset$.
An l.s.l.-groupoid is called computable if

- the relation $\{\langle x, y\rangle \colon\ x \cdot y \text{ is defined}\}$ is computable, and
- the operations $\cdot, \cap$ are computable.

Definition (Definition 2 of computably t.d.l.c. groups)

Let G be a t.d.l.c. group. We say that G is computably t.d.l.c. if

- its approximation groupoid $\mathcal{W}(G)$ has a computable copy s. t.
- the function mapping a pair of subgroups $U, V \in \mathcal{W}(G)$ to $|U : U \cap V|$ is computable.

$\mathbb{Q}_p$ is computable in this sense

Show that $G = (\mathbb{Q}_p, +)$ is computable t.d.l.c. in the sense of Definition 2.

- The open proper subgroups are of the form $U_r = p^r\mathbb{Z}_p$ for some $r \in \mathbb{Z}$. Note $\mathbb{Q}_p/U_r \cong C_{p^\infty}$ (Prüfer group).
- So each compact open coset has the form $C_{r,a} = U_r + \overline{a}$, where $r \in \mathbb{Z}, a \in C_{p^\infty}$, and $\overline{a}$ is the canonical coset representative of a.
- $C_{r,a} \cdot C_{s,b} = C_{r,a+b}$ if $r = s$, and undefined else.
- $C_{r,a} \subseteq C_{s,b}$ iff $r \geq s$ and $p^{r-s}b = a$.
- $C_{r,a} \cap C_{s,b} = \emptyset$ unless one is contained in the other.
- For $r \leq s$, we have $|U_r : U_s| = p^{s-r}$.

Theorem (Uniform equivalence of the two definitions)

G is computably t.d.l.c. in the sense of closed subgroups of $\mathrm{Sym}(\mathbb{N})$

$$\iff$$

G is computably t.d.l.c. in the sense of approximation groupoids.

Proof, "$\Rightarrow$": Suppose $G \leq_c \mathrm{Sym}(\mathbb{N})$. $\mathit{Tree}(G)$ is the tree for G.

- Each compact open subset of G is of the form $\mathcal{K}_D = \bigcup_{\eta \in D} [\eta]_{\mathit{Tree}(G)}$ for finite D. We can decide whether $\mathcal{K}_D$ is compact.
- Encode such finite sets D by natural numbers; use as inputs/outputs.
- We can decide whether $\mathcal{K}_D \subseteq \mathcal{K}_E$.
  We can compute $\mathcal{K}_D \cdot \mathcal{K}_E$ and $(\mathcal{K}_D)^{-1}$.
- Hence we can decide whether $\mathcal{K}_D$ is a subgroup, and whether it is a coset.

Theorem (recall)

G computably t.d.l.c. in sense of closed subgroups of $\mathrm{Sym}(\mathbb{N}) \iff$
G computably t.d.l.c. in sense of approximation groupoids.

Proof, "$\Leftarrow$":

- Let $\mathcal{W}$ be a computable copy of the approximation groupoid of G. Let $\widetilde{G}$ denote the group of permutations p that preserve the $\subseteq$-relation on $\mathcal{W}$, and $p(A) \cdot B = p(A \cdot B)$ whenever $A \cdot B$ is defined.
- First we verify that $\Phi\colon G \cong \widetilde{G}$ where $\Phi(g)$ is the left translation action of g, i.e. $A \mapsto gA$ for $A \in \mathcal{W}$.
- Then we show that $\widetilde{G}$ is computably t.d.l.c. as a closed subgroup of $\mathrm{Sym}(\mathbb{N})$. E.g., we have to decide whether the finite injection α given by $\sigma \oplus \tau \in \mathit{Tree}(\mathrm{Sym}(\mathbb{N}))$ can be extended to some $p \in \widetilde{G}$.
- Suppose σ sends A_i to B_i and τ sends A_i to C_i, where $i < |\sigma|$ and the A_i, B_i, C_i are cosets. Then α can be extended iff $\bigcap_{i<|\sigma|} B_i A_i^{-1} \cap A_i C_i^{-1} \neq \emptyset$ in $\mathcal{W}$, which is decidable.

The countable case

Corollary

A countable discrete group G has a computable t.d.l.c. copy $\iff$
G has a computable copy H in the usual sense.

"$\Rightarrow$".

- The approximation groupoid $\mathcal{W}(G)$ has a computable copy.
- Since G is discrete, $U = \{1\}$ is an element of $\mathcal{W}(G)$.
- Consider $H =$ the left cosets of U. Since U is normal, this is a group under the restriction of the groupoid operation to H.
- Clearly it is computable, and isomorphic to G.

"$\Leftarrow$". $\mathcal{W}(G)$ consists of finite cosets. So its operations and the index function are computable.

The most intuitive definition of computably presented. t.d.l.c. group is perhaps: the domain is an effectively locally compact, totally disconnected space on which the group operations are computable.

Definition (Generalizes Def. 1 of computability for t.d.l.c. groups)

G is computably t.d.l.c. if $G \cong ([T], \mathrm{Op}, \mathrm{Inv})$ (called a computable Baire presentation) where T is a computably locally compact tree, and $\mathrm{Inv}\colon [T] \to [T]$ and $\mathrm{Op}\colon [T] \times [T] \to [T]$ are given by Turing functionals.

Somewhat surprisingly, this is not more general than the definition based on closed subgroups of $\mathrm{Sym}(\mathbb{N})$:

Proposition

If G is computably t.d.l.c. in this sense, then $\mathcal{W}(G)$ has a computable copy. Moreover, there is a bicomputable $\Phi\colon G \cong \widetilde{G}$ where $\widetilde{G}$ is the group of "left automorphisms" of $\mathcal{W}(G)$.

Some algebraic groups over $\mathbb{Q}_p$

Using this kind of presentations, we have shown

Example

$\mathrm{SL}_2(\mathbb{Q}_p)$ is computably t.d.l.c.

Autostability

We say that computably t.d.l.c. G is autostable if for any computable Baire presentations $([T], \mathrm{Op}, \mathrm{Inv})$ and $([T'], \mathrm{Op}', \mathrm{Inv}')$ of G, there is a bicomputable group homeomorphism $[T] \to [T']$.

Proposition

Let G be computably t.d.l.c. Then G is autostable iff $\mathcal{W}(G)$ is.

Using this one can show that $(\mathbb{Q}_p, +)$ is autostable.

Computable t.d.l.c. abelian groups

Conditions for computability in the abelian case

By van Dantzig, if G is abelian t.d.l.c., there is a compact open $K \leq G$ and $L = G/K$ is discrete. So the sequence $0 \to K \to G \to L \to 0$ is exact; G is a topological extension of L by K.

Theorem (Lupini, Melnikov and N.)

Let G be abelian t.d.l.c. group. The following are equivalent.

(1) G is computable t.d.l.c.

(2) $G \cong \varprojlim (A_i, \phi_i)$ for a computable diagram (A_i, ϕ_i) where

- the A_i are abelian discrete groups,
- $\phi_i \colon A_i \to A_{i-1}$ $(i > 0)$ are epimorphisms with finite kernels
- from i can compute a bound for the elements in the kernel.

(3) There are a computable profinite group K and a computable discrete group L such that G is an extension of L by K via a computable co-cycle $c \colon L \times L \to K$.

Pontryagin-van Kampen duality

Let G be an abelian locally compact group. The dual $\widehat{G}$ is the group of characters of G, with the compact open topology.

Pontryagin-van Kampen theorem: the natural embedding $G \to \widehat{\widehat{G}}$ mapping $g \in G$ to $\lambda\phi.\phi(g) \in \widehat{\widehat{G}}$ is a topological isomorphism. E.g the dual of $\mathbb{Z}$ is the unit circle, whose dual is $\mathbb{Z}$ again.

G	$\widehat{G}$
compact	discrete
compact connected	discrete torsion free
compact totally disconnected	discrete torsion
$0 \to K \to G \to L \to 0$	$0 \leftarrow \widehat{K} \leftarrow \widehat{G} \leftarrow \widehat{L} \leftarrow 0$

The duals of t.d.l.c. groups are thus the extensions of torsion discrete groups by profinite groups.

In what sense does duality hold computably?

Theorem (Lupini, Melnikov and N., 2021)

Suppose G is an abelian t.d.l.c. group.

If G is computable then $\widehat{G}$ is computably metrized Polish.
(That is, $\widehat{G}$ is the completion of a dense computable subgroup D with a computable metric. E.g. $G = \mathbb{Z}$, $\widehat{G}$ the unit circle, D the rational unit circle.)

If G is an extension of a torsion discrete group by a profinite group (i.e., $\widehat{G}$ is t.d.l.c. as well), then

$$G \text{ is computable} \iff \widehat{G} \text{ is computable.}$$

References:

Kechris, N. and Tent, The complexity of topological group isomorphism, The Journal of Symbolic Logic, 83(3), 1190-1203. Arxiv: 1705.08081

N., Schlicht and Tent, Coarse groups, and the isomorphism problem for oligomorphic groups. J. Math Logic, in press, Arxiv: 1903.08436.

Lupini, Melnikov and N. Computable topological abelian groups. Submitted. Arxiv: 2105.12897

Melnikov and N. Computable totally disconnected locally compact groups. In preparation.