Computable groups and their orderings

Jennifer Chubb
home.gwu.edu/~jchubb

George Washington University
Washington, DC

Joint Math Meetings
AMS Special Session on Orderings in Logic and Topology
January 8, 2009

Basic notions

A group is computable if its universe is algorithmically identifiable with the natural numbers, and the group operation is computable.

How hard is it to order the elements of such a group so that the ordering is respected by the group operation?

$$x < y \implies gx < gy$$

We will identify orderings with their positive cones (the set of elements $\geq e$), and assess the algorithmic difficulty using the notion of *relative computablility*.

Basic notions

- $A \leq_T B$ if there is an algorithm using B as an oracle that will compute the characteristic function of A.
- The *Turing degree of the set A* is the collection of all sets $\equiv_T$ to A.

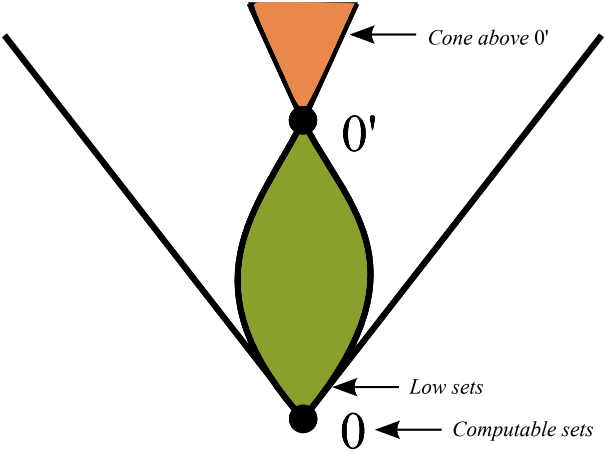
0 is the Turing degree of the computable sets, and $\mathbf{0}'$ is the Turing degree of the *jump* of the empty set (i.e. the *halting problem*).

The *jump of the set A* is the collection of all indices e of programs using oracle A that halt when their index is given as input.

$$A' = \{e \mid P_e^A(e) \downarrow\}$$

A is called *low* if its jump is as low as it can be... the same as $\emptyset'$.

The Turing degrees

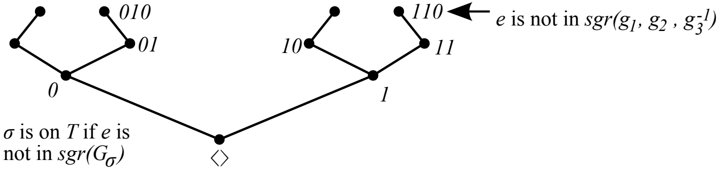


Computable trees of orderings

Reed Solomon showed that the collection of orders of an orderable computable group are in Turing-degree preserving bijective correspondence with the paths on a computable tree.

Let $G - \{e\} = \{g_1, g_2, \ldots\}$, and $sgr(G_\sigma)$ be the semigroup generated by $\{g_i | \sigma(i) = 1\} \cup \{g_i^{-1} | \sigma(i) = 0\}$.

Non-algorithmically, the picture might look like this:



Computable trees of orderings

The paths of this tree are the total left-orderings of G (use normal subsemigroups if you want bi-orderings).

Making the construction of the tree into a computable process requires some guessing, and the result is that the tree has lots of leaves, but the paths are the same.

The set of paths of a computable tree form an *effectively closed set* in Cantor space, and such a class ALWAYS has a low element. (By the Jockusch–Soare Low Basis Theorem.)

So, if it is possible to order a computable group at all, then its not *too* hard.

Computable orders?

A computable group is not always computably orderable (Downey & Kurtz, 1986). They constructed a computable copy of $\bigoplus_\omega \mathbb{Z}$ having no computable ordering of its elements.

This gives some information about the topological space of orderings of groups (as defined by Sikora in 2004).

Corollary (Dabkowska)

If $G \cong \bigoplus_\omega \mathbb{Z}$, the space of orders is homeomorphic to the Cantor space.

Corollary

If $G \cong \bigoplus_\omega \mathbb{Q}$, the space of orders is homeomorphic to the Cantor space.

Corollary

If G is torsion-free abelian of infinite rank, the space of orders is homeomorphic to the Cantor space.

Spectra of orderings

What is the collection of Turing degrees of the orderings of group G?

1. For computable, torsion-free abelian groups of finite rank ≥ 2, it is all Turing degrees.
2. For computable, torsion-free abelian groups of infinite rank, it *includes* all Turing degrees above $\mathbf{0}'$. (And by the Low Basis Theorem, always a low one as well.)
3. A computable, torsion-free abelian group of infinite rank will have an ordering in every Turing degree above the degree of the *dependence algorithm* in its computable divisible closure.

We might ask if 3 is where the low ordering comes from in the Downey–Kurtz example.

Their construction can be modified so that the corresponding dependence algorithm has degree $\mathbf{0}'$, so, no.

Groups with orderings in all *tt*-degrees

A general, sufficient condition.

Theorem

Let G be a group, and $\mathcal{P}$ a computably enumerable family of finite subsets of $G - \{e\}$ satisfying the following conditions for every $p \in \mathcal{P}$.

1. $e \notin sgr(p)$,
2. $(\exists r_0, r_1 \in \mathcal{P})(\exists g \in G)[r_0, r_1 \supset p \wedge g \in r_1 \wedge g^{-1} \in r_0]$, *and*
3. $(\forall g \in G, g \neq e)(\exists r \in \mathcal{P})[r \supseteq p \wedge (g \in r \vee g^{-1} \in r)]$.

Then there is an ordering of G in every truth table- (tt-) degree.

(A related theorem is proved by Dabkowska, Dabkowski, Harizanov, and Togha.)

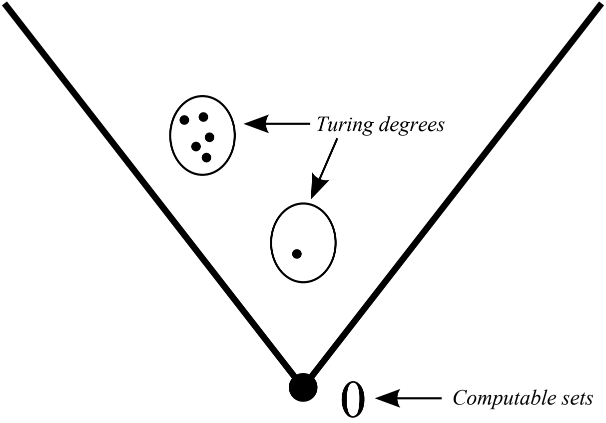
What are *tt*-degrees?

Set A is *tt*-reducible to set B (written $A \leq_{tt} B$) if $A \leq_T B$ and in addition, we have the following:

1. Predictability: There are an algorithm and computable function $h(x)$ so that $h(x)$ gives a bound on the amount of information the algorithm needs from B to determine if $x \in A$.
2. Robustness: If the algorithm gets bad information from the oracle (perhaps another set is used instead of B), it will *still halt*, though possibly it will give the wrong answer about A.

The *tt*-degrees

Each Turing degree shatters into countably many *tt*-degrees (either one or infinitely many).



Sketch of proof.

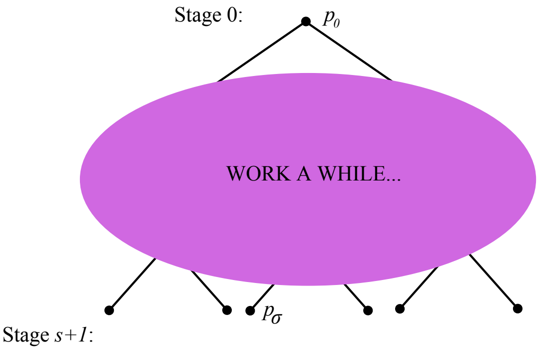
The idea is to build a computable binary tree $\mathcal{T}$ with elements of $\mathcal{P}$ attached to each node.

The key idea is this: If p is attached to σ on $\mathcal{T}$, then the elements of p attached to $\sigma^\frown 0$ and $\sigma^\frown 1$ witness that the branching condition 2 holds of p.

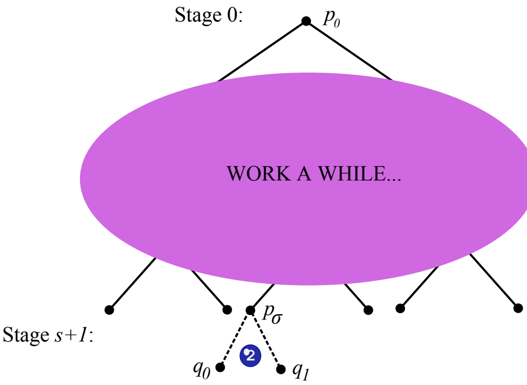
Let $\mathcal{P} = \{p_0, p_1, p_2, \ldots\}$, and $G - \{e\} = \{g_0, g_1, g_2, \ldots\}$ be computable enumerations of these sets.

Our tree $\mathcal{T}$ will be a total computable map from $2^{<\omega}$ into $\mathcal{P}$.

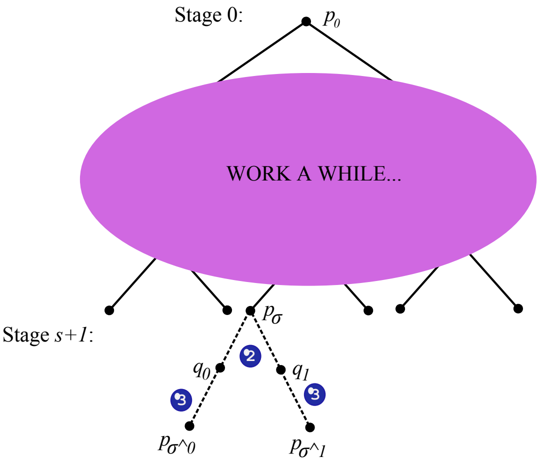
Sketch of proof.



Sketch of proof.



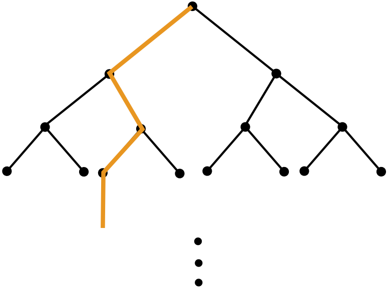
Sketch of proof.



Sketch of proof.

Let A be arbitrary, and define P_A to be $\bigcup_{s\in\omega} \mathcal{T}(A \upharpoonright s)$.

The path (i.e., the set A) is *tt*-equivalent to the ordering P_A.



Sketch of proof.

1. $P_A \leq_{tt} A$.
   - $x \in P_A$ if and only if $x \in \bigcup_{i \in \omega}^{h(x)+1} \mathcal{T}(A \upharpoonright i)$.
   - The function $h(x) = \min_s (x = g_s)$ is a computable bound on resources, and all possible oracles result in a halting computation.
2. $A \leq_{tt} P_A$.
   - To decide if $x \in A$, construct the tree to level x, $\mathcal{T}_x$. Let $h(x) = \max\left(\{S(\sigma) \mid \sigma \in \mathtt{dom}(\mathcal{T}_x)\} \cup \{\mathcal{T}_x(\sigma) \mid |\sigma| = x\}\right)$.
   - If $\mathcal{T}_x(\sigma_A) \subset P_A \upharpoonright h(x)$ for some σ_A of length x, then

$$x \in A \iff S(\sigma_A) \in P_A.$$

   Otherwise, halt and output 0.

Thank you!

References

- Dabkowska, M., *Turing Degree Spectra of Groups and Their Spaces of Orders*, Ph.D. dissertation, George Washington University, 2006.
- Dabkowska, Dabkowski, Harizanov, and Togha, Spaces of orderings and their Turing degree spectra (submitted to *Annals of Pure and Applied Logic*).
- Downey, R., and Kurtz, S., Recursion theory and ordered groups, *Annals of Pure and Applied Logic* (1986).
- Sikora, A., Topology on the spaces of orderings of groups, *Bulletin of the London Mathematics Society* (2004).
- Solomon, R., *Reverse Mathematics and Ordered Groups, PhD dissertation*, Cornell Univeristy, 1998.