Towards Embedding Theorem: $Aut(\mathcal{D}_{\alpha e})$ embeds into $Aut(\mathcal{D}_{\alpha})$

Dávid Tóth

University of Leeds, UK
JAISTT, Japan

21st September 2016

Joint work with Mariya Soskova

Basic concepts in α-Computability Theory

Definition (admissible ordinal)

α *admissible* iff $\forall f \in \Sigma_1(L_\alpha) \forall K \in L_\alpha . f[K] \in L_\alpha$.

Definition

$K \subseteq \alpha$ is α-*finite* iff $K \in L_\alpha$.

Definition

$f : \alpha \to \alpha$ is α-*computable* iff $f \in \Sigma_1(L_\alpha)$.

Definition (α-enumeration and α-reducibility and degrees)

$A \leq_{\alpha e} B :\iff$
$\exists W \in \Sigma_1(L_\alpha) \forall \gamma < \alpha [K_\gamma \subseteq A \iff \exists \delta < \alpha(\langle \gamma, \delta \rangle \in W \wedge K_\delta \subseteq B)]$.
$A \leq_\alpha B :\iff A \oplus \overline{A} \leq_{\alpha e} B \oplus \overline{B}$.
$\mathcal{D}_{\alpha e} := 2^\alpha / \equiv_{\alpha e}$ are α-enumeration degrees,
$\mathcal{D}_\alpha := 2^\alpha / \equiv_\alpha$ are α-degrees (*total* α-enumeration degrees).

Embedding Conjecture

Embedding Conjecture

$\exists \eta : \mathrm{Aut}(\mathcal{D}_{\alpha e}) \hookrightarrow \mathrm{Aut}(\mathcal{D}_\alpha)$.

Proof

Follows from the 3 statements:

1. $\mathcal{D}_\alpha$ degrees are embeddable in $\mathcal{D}_{\alpha e}$, i.e. $\exists \iota : \mathcal{D}_\alpha \hookrightarrow \mathcal{D}_{\alpha e}, A \mapsto A \oplus \overline{A}$
2. $\mathcal{D}_\alpha$ are an automorphism base for $\mathcal{D}_{\alpha e}$,
i.e. $\forall f \in \mathrm{Aut}(\mathcal{D}_{\alpha e})\ [f_{|\iota(\mathcal{D}_\alpha)} = 1_{\iota(\mathcal{D}_\alpha)} \implies f = 1_{\mathcal{D}_{\alpha e}}]$
Note (2) is implied by the conjecture generalizing a theorem of Selman:
$\forall X \subseteq \alpha [B \in \Sigma_1(L_\alpha[X]) \implies A \in \Sigma_1(L_\alpha[X])] \iff A \leq_{\alpha e} B$.
3. $\mathcal{D}_\alpha$ are definable in $\mathcal{D}_{\alpha e}$.

Then $\eta(f) := \iota^{-1} \circ f \circ \iota$ is the required embedding.

Definability of $\mathcal{D}_\alpha$ in $\mathcal{D}_{\alpha e}$

Definition (α-semicomputable set)

$A \subseteq \alpha$ is *α-semicomputable* iff
$\exists s_A : \alpha \times \alpha \to \alpha \forall x, y \in \alpha[x \in A \vee y \in A \implies s_A(x, y) \in A \cap \{x, y\}]$.

Definition (Kalimullin pair)

Kalimullin pair: $\mathcal{K}(A, B) :\iff \exists W \in \Sigma_1(L_\alpha)[A \times B \subseteq W \wedge \overline{A} \times \overline{B} \subseteq \overline{W}]$,
Maximal: $\mathcal{K}_{max}(A, B) :\iff \mathcal{K}(A, B) \wedge$
$\forall C, D[A \leq_{\alpha e} C \wedge B \leq_{\alpha e} D \wedge \mathcal{K}(C, D) \implies A \equiv_{\alpha e} C \wedge B \equiv_{\alpha e} D]$,
Nontrivial: $K_{nt}(A, B) :\iff A \notin \Sigma_1(L_\alpha) \wedge B \notin \Sigma_1(L_\alpha)$.

Conjecture

$\iota(\mathcal{D}_\alpha)$ are definable in $\mathcal{D}_{\alpha e}$:
$\forall c \in \mathcal{D}_{\alpha e}[c \text{ total iff } c = 0 \vee \exists a, b \in \mathcal{D}_{\alpha e}(c = a \oplus b \wedge \mathcal{K}_{\max}(a, b))]$

Conjecture

$\iota(\mathcal{D}_\alpha)$ are definable in $\mathcal{D}_{\alpha e}$:
$\forall c \in \mathcal{D}_{\alpha e}[c \text{ total iff } c = 0 \vee \exists a, b \in \mathcal{D}_{\alpha e}(c = a \oplus b \wedge \mathcal{K}_{\max}(a, b))]$

Proof.

The conjecture follows from the 3 statements:
1. A Kalimullin pair is definable in $\mathcal{D}_{\alpha e}$,
Conjecture: $\mathcal{K}(a, b) :\iff \forall x \in \mathcal{D}_{\alpha e}.(a \oplus x) \wedge (b \oplus x) = x$.
2. $\forall c \in \iota(\mathcal{D}_\alpha) - \{0\} \exists a, b \in \mathcal{D}_{\alpha e}[c = a \oplus b \wedge \mathcal{K}_{max}(a, b)]$,
2.1. $\forall c \in \iota(\mathcal{D}_\alpha) - \{0\} \exists C$ α-semicomputable st
$C \oplus \overline{C} \in c \wedge C \notin \Sigma_1(L_\alpha) \wedge C \notin \Pi_1(L_\alpha)]$ (Uses Shore's Splitting Theorem),
2.2. C is α-semicomputable $\wedge C \notin \Sigma(L_\alpha) \wedge C \notin \Pi_1(L_\alpha) \implies \mathcal{K}_{max}(C, \overline{C})$,
3. $\forall a, b \in \mathcal{D}_{\alpha e}[\mathcal{K}_{max}(a, b) \implies \exists$ α-semicomputable C st $C \in a \wedge \overline{C} \in b]$,
thus $a \oplus b$ contains $C \oplus \overline{C}$ and hence is total.
The statement (3) implied by the theorem on the next slide. □

Labelling algorithm

Theorem

$\mathcal{K}_{nt}(A, B) \implies \exists C$ α-semicomputable st $A \leq_{\alpha e} C$ and $B \leq_{\alpha e} \overline{C}$.

Given a set $C \subseteq Q_\alpha = \{\sigma \in 2^{<\alpha} :\}$ and a labelling function
$q_s : \alpha_A \coprod \alpha_B \rightarrowtail Q_\alpha$, define
$A_C := \{a \in \alpha_A : \exists s < \alpha . q_s(a) \in C\}$
$B_C := \{b \in \alpha_B : \exists s < \alpha . q_s(b) \in \overline{C}\}$.
The aim of the algorithm is to label α-rational numbers by the ordinals from α_A and α_B st there exists a cut C on the α-rational line (thus C is α-semicomputable) st $A_C = A$ and $B_C = B$. This implies the theorem.
Hence if $(a, b) \in A \times B$, then try to satisfy $q_s(a) < q_s(b)$,
if $(a, b) \in \overline{A} \times \overline{B}$, then try to satisfy $q_s(b) < q_s(a)$.
As $\mathcal{K}_{nt}(A, B)$, so $\exists W \in \Sigma_1(L_\alpha)[A \times B \subseteq W \wedge \overline{A} \times \overline{B} \subseteq W]$. Thus use W to try to satisfy these conditions.

Labelling algorithm - outline

1. Initially label the α-rationals on the left by the labels from α_B, label the α-rationals on the right by the labels from α_A.
2. Enumerate pairs from W,
3. Run a strategy (a, b) for $(a, b) \in \alpha_A \times \alpha_B$ according to its priority: gradually move a label $a \in \alpha_A$ towards left and the label from $b \in \alpha_B$ towards right trying to satisfy the conditions
   $(a, b) \in A \times B \implies q_s(a) < q_s(b)$ and
   $(a, b) \in \overline{A} \times \overline{B} \implies q_s(b) < q_s(a)$.
   Always require $q_s(a) < q_s(b) \implies (a, b) \in W$,
4. Every strategy will stop acting in less than α-steps, thus every pair of labels will eventually have a fixed position on the α-rational line.

Labelling algorithm

Note in some cases the labels cannot be ordered, e.g. when initially $q_s(b_2) < q_s(b_1) < q_s(a_1) < q_s(a_2)$, but next $(a_1, b_1) \searrow W_{s_1}$ and $(a_2, b_2) \searrow W_{s_2}$.
If $q_s(b) < q_s(a)$, but $(a, b) \in W$, then declare the interval $[q_s(a), q_s(b)]$ a dead zone (DZ) and prevent the pairs (a_i, b_i) with the strategies (a_i, b_i) of the lower priority than the strategy (a, b) moving the labels inside the dead zone.
Define the cut C and its complement D by
$C := \{\rho \in Q_\alpha : \exists b \notin B \exists s < \alpha . \rho \leq q_s(b) \downarrow \text{ or } \{\rho, q(b)\} \subseteq \text{ a PDZ }\}$.
$D := \{\rho \in Q_\alpha : \exists a \notin A \exists s < \alpha . q_s(a) \downarrow \leq \rho \text{ or } \{\rho, q(a)\} \subseteq \text{ a PDZ }\}$.
Then $C \cap D = \emptyset$, $A_C = A$ and $B_C = B$ as required.

Labelling algorithm in α-Computability Theory

- using Q_α instead of $\mathbb{Q}$,
- at a limit stage the labelling function is the union of the previous ones, i.e. for $q_\delta = \bigcup_{\gamma<\delta} q_\gamma$ for δ limit,
- two priority orderings for strategies: major by the order of the enumeration of the pair and the minor fixed (hence no clearing of the labels necessary),
- at stage $s < \alpha$ the strategy of the priority p will be allowed to label only the rationals with the binary representation of whose every nonzero substring (i.e. string containing at least one other symbol than 0) is of order type in $[(\Sigma_{\beta<s}\beta) + p, (\Sigma_{\beta<s+1}\beta) + p + \omega)]$. This is to guarantee that at every stage there would be enough space for new adjacent labels.
- termination of the strategy (a, b) is guaranted by the admissibility of α and the algorithm does not work for nonadmissible ordinals,
- break of the symmetry in proofs, e.g. in general it is not true that $\neg K \subseteq A \iff K \subseteq \overline{A}$ for $K \in L_\alpha$.

Thank you.