A Timeline of Lattice Embeddings into the Turing Degrees

José Jeremías Valenzuela Morales

The George Washington University

josejeremias.valenzuelamorales@gwu.edu

Online Logic Seminar
April 16, 2026

Motivation

To quote M. Lerman [L98]:
”The attempt to determine the largest natural decidable fragment of the elementary theory of a given degree structure [...] produces a separation of the uniform fragment of the elementary theory from the pathological fragment”.

Motivation

To quote M. Lerman [L98]:
"The attempt to determine the largest natural decidable fragment of the elementary theory of a given degree structure [...] produces a separation of the uniform fragment of the elementary theory from the pathological fragment".

Theorem (Lac68)

The elementary theory of $(\mathcal{D}, \leq_T)$ *is undecidable.*

Motivation

To quote M. Lerman [L98]:
"The attempt to determine the largest natural decidable fragment of the elementary theory of a given degree structure [...] produces a separation of the uniform fragment of the elementary theory from the pathological fragment".

Theorem (Lac68)

The elementary theory of $(\mathcal{D}, \leq_T)$ *is undecidable.*

In this talk, we are interested in exploring decidability results about the $\exists$-theory of $(\mathcal{D}, \leq_T)$, $(\mathcal{D}, \leq_T, \vee, ')$, and $(\mathcal{D}, \leq_T, \vee, 0, ')$.

A classical example

Theorem (KP54)

Given any finite collection of noncomputable c.e. sets $\langle A_i \rangle_m$, for every $n \geq 1$ a collection of noncomputable c.e. sets $\langle B_i \rangle_n$ can be found such that:

1. $B_k \leq_T (\oplus_{i=1}^m A_i)'$ *for every* $1 \leq k \leq n$.
2. $B_k \not\leq_T A_i$ *for every* $1 \leq i \leq m$
3. $B_k \not\leq_T B_j$ *for every* $1 \leq j \leq n$ *different than* k.
4. *For each* k, $A_k \not\leq_T B_i$ *for every* $1 \leq i \leq n$.

A classical example

Theorem (KP54)

Given any finite collection of noncomputable c.e. sets $\langle A_i \rangle_m$, for every $n \geq 1$ a collection of noncomputable c.e. sets $\langle B_i \rangle_n$ can be found such that:

1. $B_k \leq_T (\oplus_{i=1}^m A_i)'$ *for every* $1 \leq k \leq n$.
2. $B_k \not\leq_T A_i$ *for every* $1 \leq i \leq m$
3. $B_k \not\leq_T B_j$ *for every* $1 \leq j \leq n$ *different than* k.
4. *For each* k, $A_k \not\leq_T B_i$ *for every* $1 \leq i \leq n$.

In the proof, the first condition is satisfied vacuously by using $(\oplus_{i=1}^n A_i)'$ as an oracle.

Definition (Requirements)

With index $i, j, k \leq s$ for every stage s, our requirements are as follows:

- R_e : For every $1 \leq i \leq m$, $\exists x[\phi_e^{A_i}(x)\downarrow \implies \phi_e^{A_i}(x) \neq \chi_{B_k}(x)]$.
- S_e : For every $1 \leq j \leq n$ different than k, $\exists x[\phi_e^{B_j}(x)\downarrow \implies \phi_e^{B_j}(x) \neq \chi_{B_k}(x)]$.
- T_e : For every $1 \leq i \leq n$, $\exists x[\phi_e^{B_i}(x)\downarrow \implies \phi_e^{B_i}(x) \neq \chi_{A_k}(x)]$.

Definition (Requirements)

With index $i, j, k \leq s$ for every stage s, our requirements are as follows:

- R_e : For every $1 \leq i \leq m$, $\exists x[\phi_e^{A_i}(x) \downarrow \implies \phi_e^{A_i}(x) \neq \chi_{B_k}(x)]$.
- S_e : For every $1 \leq j \leq n$ different than k, $\exists x[\phi_e^{B_j}(x) \downarrow \implies \phi_e^{B_j}(x) \neq \chi_{B_k}(x)]$.
- T_e : For every $1 \leq i \leq n$, $\exists x[\phi_e^{B_i}(x) \downarrow \implies \phi_e^{B_i}(x) \neq \chi_{A_k}(x)]$.

Prior to the initialization of any strategy, B_k is defined as the empty set for every $1 \leq k \leq n$. It can also be assumed that witnesses come from an increasing monotonic sequence of natural numbers.

Strategy for R_e: For every $1 \leq i \leq m$ and $1 \leq k \leq n$.

1. Choose a new unique witness $x \in \mathbb{N}$ and use our oracle to check if $\phi_e^{A_i}(x)$ halts.
2. If $\phi_e^{A_i}(x) \downarrow = 0$, enumerate x into B_k.

Strategy for R_e: For every $1 \leq i \leq m$ and $1 \leq k \leq n$.

1. Choose a new unique witness $x \in \mathbb{N}$ and use our oracle to check if $\phi_e^{A_i}(x)$ halts.
2. If $\phi_e^{A_i}(x) \downarrow= 0$, enumerate x into B_k.

Strategy for S_k: For every $1 \leq j \leq n$ different than k and $1 \leq k \leq n$.

1. Choose a new unique witness $x \in \mathbb{N}$ and wait for $\phi_e^{B_j}(x)$ to halt.
2. If $\phi_e^{B_j}(x) \downarrow= 0$ with use u, enumerate x into B_k and put a restraint on $B_j \restriction_u$.

Strategy for T_k: For every $1 \leq i \leq m$ and $1 \leq k \leq n$.

1. Choose a new unique witness $x \in A_i$ and wait for $\phi_e^{B_k}(x)$ to halt.
2. If $\phi_e^{B_k}(x) \downarrow = 0$ with use u, put a restraint on $B_k \restriction_u$.

Strategy for T_k: For every $1 \leq i \leq m$ and $1 \leq k \leq n$.

1. Choose a new unique witness $x \in A_i$ and wait for $\phi_e^{B_k}(x)$ to halt.
2. If $\phi_e^{B_k}(x) \downarrow = 0$ with use u, put a restraint on $B_k \upharpoonright_u$.

Injuries: Both R_e and S_k strategies might cause injuries to lower priority strategies when enumerating elements into B_k. When this happens, the injured strategies chooses a new witness and restart.

Theorem

Any finite upper semilattice L can be embedded into $(\mathcal{D}, \leq_T)$.

Theorem

Any finite upper semilattice L can be embedded into $(\mathcal{D}, \leq_T)$.

We may define a *rank* for each element of our finite upper semilattice. Each $x \in L$ with no $y \in L$ such that $y \leq_L x$ will have rank 0 and so on. As the lattice is finite, then there will only be finitely many ranks and finitely many elements of any given rank.

Theorem

Any finite upper semilattice L can be embedded into $(\mathcal{D}, \leq_T)$.

We may define a *rank* for each element of our finite upper semilattice. Each $x \in L$ with no $y \in L$ such that $y \leq_L x$ will have rank 0 and so on. As the lattice is finite, then there will only be finitely many ranks and finitely many elements of any given rank.

To construct our embedding, we can use the main theorem of [KP54] to build the degrees to which L will embed into rank by rank inside $(\mathcal{D}, \leq_T)$.

Theorem

Any finite upper semilattice L can be embedded into $(\mathcal{D}, \leq_T)$.

We may define a *rank* for each element of our finite upper semilattice. Each $x \in L$ with no $y \in L$ such that $y \leq_L x$ will have rank 0 and so on. As the lattice is finite, then there will only be finitely many ranks and finitely many elements of any given rank.

To construct our embedding, we can use the main theorem of [*KP*54] to build the degrees to which L will embed into rank by rank inside $(\mathcal{D}, \leq_T)$.

As a direct result, we can obtain that the $\exists$-theory of $(\mathcal{D}, \leq_T)$ is decidable.

$(\mathcal{D}, \leq_T, \vee, ')$

Definition

A jump upper semilattice is a structure $\mathcal{J} = \langle J, \leq, \vee, j \rangle$ such that:

- $\langle J, \leq \rangle$ is a partial ordering;
- For all $x, y \in J$, $x \vee y$ is the least upper bound of x and y;
- $j(\cdot)$ is a unary jump operator i.e. for all $x, y \in J$, $x < j(x)$; and if $x \leq y$, then $j(x) \leq j(y)$.

Definition

A jump upper semilattice is a structure $\mathcal{J} = \langle J, \leq, \vee, j \rangle$ such that:

- $\langle J, \leq \rangle$ is a partial ordering;
- For all $x, y \in J$, $x \vee y$ is the least upper bound of x and y;
- $j(\cdot)$ is a unary jump operator i.e. for all $x, y \in J$, $x < j(x)$; and if $x \leq y$, then $j(x) \leq j(y)$.

Theorem (M03)

Every countable JUSL can be embedded into the Turing degrees.

$(\mathcal{D}, \leq_T, \vee, ')$

Definition

A jump upper semilattice is a structure $\mathcal{J} = \langle J, \leq, \vee, j \rangle$ such that:

- $\langle J, \leq \rangle$ is a partial ordering;
- For all $x, y \in J$, $x \vee y$ is the least upper bound of x and y;
- $j(\cdot)$ is a unary jump operator i.e. for all $x, y \in J$, $x < j(x)$; and if $x \leq y$, then $j(x) \leq j(y)$.

Theorem (M03)

Every countable JUSL can be embedded into the Turing degrees.

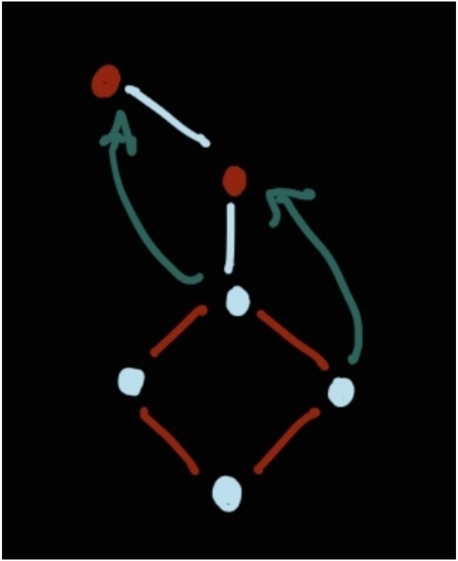
As a direct result, we have that the $\exists$-theory of $(\mathcal{D}, \leq_T, \vee, ')$ is decidable.

$(\mathcal{D}, \leq_T, \vee, 0, ')$

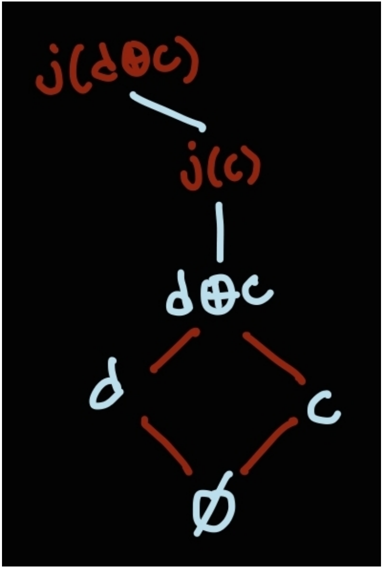
Theorem (L06)

The $\exists$-theories of $(\mathcal{D}, \leq_T, \vee, 0, ')$ and $(\mathcal{D}_{REA}, \leq_{\mathcal{T}}, \vee, \prime, ')$ are decidable.

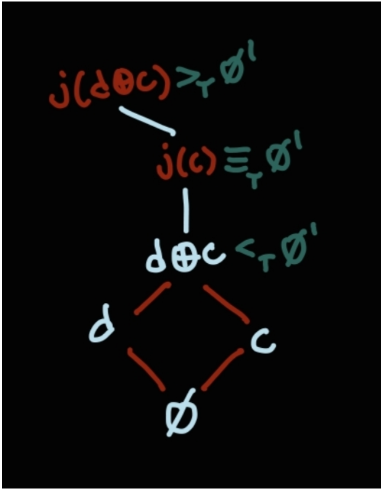
Science fiction: Low Embeddings



Science fiction: Low Embeddings



Science finction: Low Embeddings



True Stages

Definition (Global 1-true stages)

A global 1-true stage ordering is a computable partial ordering $\preccurlyeq$ on $\mathbb{N}^{<\mathbb{N}}$ that satisfies the following properties:

1. $\langle\rangle \preccurlyeq \tau$ for all τ;
2. If $\sigma \preccurlyeq \tau$, then $\sigma \subseteq \tau$;
3. For each $X \in \mathbb{N}^{\mathbb{N}}$, there is an infinite sequence of initial segments of X such that $\tau_0 \preccurlyeq \tau_1 \preccurlyeq \ldots \subset X$; and
4. For every $\tau \subset \sigma \subset \rho$, if $\tau \preccurlyeq \rho$, then $\tau \preccurlyeq \sigma$.

True Stages

Definition (Global 1-true stages)

A global 1-true stage ordering is a computable partial ordering $\preccurlyeq$ on $\mathbb{N}^{<\mathbb{N}}$ that satisfies the following properties:

1. $\langle\rangle \preccurlyeq \tau$ for all τ;
2. If $\sigma \preccurlyeq \tau$, then $\sigma \subseteq \tau$;
3. For each $X \in \mathbb{N}^{\mathbb{N}}$, there is an infinite sequence of initial segments of X such that $\tau_0 \preccurlyeq \tau_1 \preccurlyeq \ldots \subset X$; and
4. For every $\tau \subset \sigma \subset \rho$, if $\tau \preccurlyeq \rho$, then $\tau \preccurlyeq \sigma$.

When $\sigma \preccurlyeq \tau$, we say that τ believes σ. For substrings defined as in property (3), we say that τ is X-true and write $\tau \preccurlyeq X$.

Definition

We say that $\preccurlyeq$ is complete if, for every $X \in \mathbb{N}^{\mathbb{N}}$ we have that $\mathcal{T}_X := \langle \tau \in \mathbb{N}^{<\mathbb{N}} : \tau \preccurlyeq X \rangle \equiv_T X'$ uniformly in X.

Definition

We say that $\preccurlyeq$ is complete if, for every $X \in \mathbb{N}^{\mathbb{N}}$ we have that $\mathcal{T}_X := \langle \tau \in \mathbb{N}^{<\mathbb{N}} : \tau \preccurlyeq X \rangle \equiv_T X'$ uniformly in X.

Theorem

There is a complete, global, 1-true stage system.

Definition

We say that $\preccurlyeq$ is complete if, for every $X \in \mathbb{N}^{\mathbb{N}}$ we have that $\mathcal{T}_X := \langle \tau \in \mathbb{N}^{<\mathbb{N}} : \tau \preccurlyeq X \rangle \equiv_T X'$ uniformly in X.

Theorem

There is a complete, global, 1-true stage system.

Following Lachlan's notion of *true stages*, our constructions will use $\mathcal{T}_\emptyset$ as their $\emptyset'$-oracle. As we cannot directly access $\mathcal{T}_\emptyset$, we will define a series of stages $\mathcal{T}_\sigma$ that approximate it in a computable way.

σ Stages

Definition (σ Stages)

Consider an approximation of $\emptyset'$ via finite binary sequences $\sigma_s = \langle \sigma_{i \leq s, s} \rangle \in \mathbb{N}^{<\mathbb{N}}$ defined as follows:

1. $\sigma_{0,0} = 0$, and
2. $\sigma_{i,s} = \begin{cases} 1 & \phi_i(i) \downarrow \text{ in } s \text{ steps} \\ 0 & \text{Otherwise} \end{cases}$.

σ Stages

Definition (σ Stages)

Consider an approximation of $\emptyset'$ via finite binary sequences $\sigma_s = \langle \sigma_{i \leq s,s} \rangle \in \mathbb{N}^{<\mathbb{N}}$ defined as follows:

1. $\sigma_{0,0} = 0$, and
2. $\sigma_{i,s} = \begin{cases} 1 & \phi_i(i)\downarrow \text{ in } s \text{ steps} \\ 0 & \text{Otherwise} \end{cases}.$

Then, at stage s we call $\mathcal{T}_\sigma = \langle \tau \in \mathbb{N}^{<\mathbb{N}} : \tau \preccurlyeq \sigma_s \rangle$ the *current* approximation of $\emptyset'$.

σ Stages

Definition (σ Stages)

Consider an approximation of $\emptyset'$ via finite binary sequences $\sigma_s = \langle \sigma_{i \leq s, s} \rangle \in \mathbb{N}^{<\mathbb{N}}$ defined as follows:

1. $\sigma_{0,0} = 0$, and
2. $\sigma_{i,s} = \begin{cases} 1 & \phi_i(i) \downarrow \text{ in } s \text{ steps} \\ 0 & \text{Otherwise} \end{cases}$.

Then, at stage s we call $\mathcal{T}_\sigma = \langle \tau \in \mathbb{N}^{<\mathbb{N}} : \tau \preccurlyeq \sigma_s \rangle$ the *current* approximation of $\emptyset'$.

Definition

We say that our construction $\mathcal{T}_{\sigma_j}$-believes $\mathcal{T}_{\sigma_k}$ when $\mathcal{T}_{\sigma_k} \subseteq \mathcal{T}_{\sigma_j}$ and $\tau \preccurlyeq \sigma_j$ for every $\tau \preccurlyeq \sigma_k$.

Injuries

As only $\emptyset$-true stages will be correct, we must consider injuries arising from the use of incomplete oracles in our construction.

Injuries

As only $\emptyset$-true stages will be correct, we must consider injuries arising from the use of incomplete oracles in our construction.

Definition (σ-injuries)

Let $k < s$ and $|\mathcal{T}_{\sigma_k}| = l$. At stage s, if it is found that $\mathcal{T}_{\sigma_k} \neq \mathcal{T}_{\sigma_s} \upharpoonright_l$ we say that every strategy that has used oracle $\mathcal{T}_{\sigma_k}$ *in order to carry out an action* has been σ-injured and require attention.

Some considerations

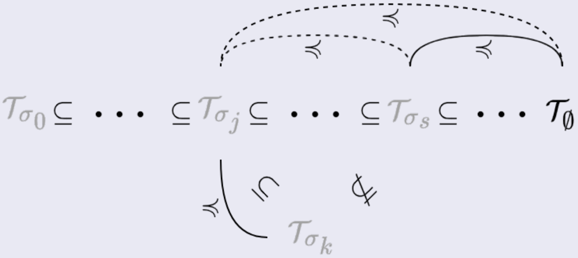
As long as our constructions are able to successfully recover from σ-injuries, they will be able to properly carry out their work over the infinite path of $\emptyset$-true stages. In order to achieve this, they will follow these rules:

1. They will always act with the current $\mathcal{T}_\sigma$ as an oracle;
2. At the beginning of every stage, the current $\mathcal{T}_\sigma$ will be used to revisit, in order of priority, all previously initiated strategies in order to identify possible σ-injuries. In other words, they will update their set of $\mathcal{T}_\sigma$-beliefs at the beginning of every stage; and
3. σ-injuries will take priority over regular injuries that arise from the interaction between strategies.

Lemma

Let the relations between $\mathcal{T}_{\sigma_0}$, $\mathcal{T}_{\sigma_j}$, $\mathcal{T}_{\sigma_k}$, $\mathcal{T}_{\sigma_s}$, and $\mathcal{T}_{\emptyset}$ hold as expressed in figure 1. Then, the following two statements hold:

- *If our construction $\mathcal{T}_{\emptyset}$-believes $\mathcal{T}_{\sigma_s}$ and $\mathcal{T}_{\sigma_s}$-believes $\mathcal{T}_{\sigma_j}$, then it $\mathcal{T}_{\emptyset}$-believes $\mathcal{T}_{\sigma_j}$; and*
- *If our construction $\mathcal{T}_{\emptyset}$-believes $\mathcal{T}_{\sigma_s}$ but does not $\mathcal{T}_{\sigma_s}$-believe $\mathcal{T}_{\sigma_k}$, then it does not $\mathcal{T}_{\emptyset}$-believe $\mathcal{T}_{\sigma_k}$ either.*



A toy example

With index i in $\mathbb{N}$, these are our requirements:

- L_i: $\mathbf{c}$ is low;
- N_i: $\mathbf{d}$ is not low; and
- $\dots$.

A toy example

With index i in $\mathbb{N}$, these are our requirements:

- L_i: **c** is low;
- N_i: **d** is not low; and
- $\ldots$.

What will we do?

Use $\mathcal{T}_\sigma$ to ensure L_i and N_i always hold.

L_i strategies

At first, **c** is set to be the empty set.

1. Upon initialization at stage $s > 1$, $\mathcal{T}_\sigma$ is used to check if $\Phi_s^{\mathbf{c}[s]}(s)$ halts. If so, a restrain is put on **c** with the use of that computation, u_s.
2. If revisited at stage k, $\mathcal{T}_\sigma$ is used to check if $\Phi_s^{\mathbf{c}[k]}(s)$ halts. If so, a restrain is put on **c** with the use of that computation, u_k.

L_i strategies

At first, **c** is set to be the empty set.

1. Upon initialization at stage $s > 1$, $\mathcal{T}_\sigma$ is used to check if $\Phi_s^{\mathbf{c}[s]}(s)$ halts. If so, a restrain is put on **c** with the use of that computation, u_s.
2. If revisited at stage k, $\mathcal{T}_\sigma$ is used to check if $\Phi_s^{\mathbf{c}[k]}(s)$ halts. If so, a restrain is put on **c** with the use of that computation, u_k.

Injuries

1. If there is a low enumeration, say between u_m and u_n, all L_i strategies with $i > m$ are restarted in order of priority.
2. If it is discovered that a restrain was placed based on incorrect information from a past $\mathcal{T}_\sigma$, all L_i strategies with $i \geq k$ are restarted in order of priority.

Some Questions.

1. Is this weaker than Lerman's framework?
2. Can you "bruteforce" the emeddings of finite JUSL with finite support?
3. Are there any nontrivial examples where this "bruteforcing" makes sense?

References

[Lac68] A.H. Lachlan, Distributive initial segments of the degrees of unsolvability, Z. Math. Logik Grundlag Math., vol. 14 (1968), pp. 457–472.

[L98] M. Lerman, A necessary and suficient condition for embedding ranked finite partial lattices into the computably enumerable degrees, Ann. Pure Appl. Logic 94 (1998) 143–180.

[KP54] S.C. Kleene and E.L. Post, The upper semi-lattice of the degrees of recursive unsolvability, Annals of Mathematics, vol. 59 (1954), pp. 379–407.

[M03] Antonio Montalbán. (2003). Embedding Jump Upper Semilattices into the Turing Degrees. The Journal of Symbolic Logic, 68(3), 989–1014.

Thank you!