The Universal Turing machine and relative computability

Julia F. Knight

January 5, 2012

The Universal Turing machine

Before Turing, there were single-purpose computers, designed to compute a particular function. One of Turing's great contributions was the idea of a *universal* machine, which could simulate any single-purpose machine, by taking as part of its input a set of instructions, or *program*. The computers on our desks implement this idea.

Relative computability

Turing also defined the notion of *relative* computability. He imagined a machine that would compute a function f, given answers to questions about membership in a set X. Turing imagined the answers coming from an "oracle".

The computers on our desks also implement this idea. We imagine an interactive program, with the role of the oracle played by the user. Or, the role of the oracle may be played by a *CD*-rom. We do not need a different computer for each set X.

List of machines, or programs

- We have an effective list of the programs. We write φ_e for the partial function computed by program number e on this list.
- We also have an effective list of the possibly interactive programs. We write φ_e^X for the partial function computed using program number e with oracle X.

Notation: We write $\varphi_e(n)\downarrow$ if program e eventually halts, given input n. Similarly, we write $\varphi_e^X(n)\downarrow$ if interactive program e eventually halts, given oracle X and input n.

The halting set and jumps

The effective lists immediately suggest a way to build more complicated sets.

- The *halting set* K is $\{e : \varphi_e(e)\downarrow\}$. This set is computably enumerable but not computable; i.e., we can effectively list the elements, but we cannot effectively list the complement, so we cannot compute the characteristic function.
- For an arbitrary set X, the *jump* is $X' = \{e : \varphi_e^X(e)\downarrow\}$. This set is computably enumerable but not computable relative to X.

Iterating the jump

We can iterate the jump to get $X^{(n)}$, for $n \in \omega$.

- $X^{(0)} = X$
- $X^{(n+1)} = (X^{(n)})'$

We can continue the iteration process through the computable ordinals.

- Let $X^{(\omega)} = \{< n, x >: x \in X^{(n)}\}$.
- $X^{(\alpha+1)} = (X^{(\alpha)})'$
- for limit α, $X^{(\alpha)}$ represents $\{< \beta, x >: x \in X^{(\beta)}\}$—to make this precise, we code the ordinal β by a natural number, using Kleene's system of ordinal notation.

The Kleene-Mostowski hierarchy and the Davis-Mostowski hierarchy

Stephen Kleene and Andrzej Mostowski independently defined what is now called the *arithmetical hierarchy*. Martin Davis and Andrzei Mostowski independently defined what is now called the *hyperarithmetical hierarchy*, extending the arithmetical hierarchy through the "computable" ordinals. I will give the definition in a uniform way.

- **Arithmetical Hierarchy**. For $1 \leq n < \omega$, a set is Σ^0_n if it is computably enumerable relative to $\emptyset^{(n-1)}$. A set is Π^0_n if the complement is Σ^0_n, and it is Δ^0_n if it is both Σ^0_n and Π^0_n.
- **Hyperarithmetical hierarchy**. For computable $\alpha \geq \omega$, a set is Σ^0_α if it is c.e. relative to $\emptyset^{(\alpha)}$. A set is Π^0_α if the complement is Σ^0_α, and it is Δ^0_α if it is both Σ^0_α and Π^0_α.

Barwise and the unity of computability, set theory, and model theory

There was a period around 1970 when the model theory of infinitary logic, a version of set theory, and "higher" recursion theory were united. Jon Barwise was a great spokesperson for these developments, but there were many contributors, including Georg Kreisel, Saul Kripke, Richard Platek, Carole Karp, Ronald Jensen, John Schlipf, Jean-Pierre Ressayre, and Gerald Sacks and several of his students.

I will state a well-known result of Barwise, and then a less-well-known result of John Gregory.

Barwise Compactness

Barwise Compactness. Let A be a countable "admissible"[1] set, and let L_A be the set of $L_{\omega_1\omega}$ formulas in A. Suppose T is a set of L_A sentences that is "Σ_1 on A"[2], s.t. every "A-finite"[3] subset of T has a model. Then T has a model.

[1] An *admissible set* is a transitive set that satisfies the axioms of Kripke-Platek set theory—a weak version of set theory in which we drop the power set axiom, and restrict the separation and collection axioms to formulas with just bounded quantifiers.

[2] A set is Σ_1 *on A* if it is defined by a finitary formula with only existential and bounded quantifiers.

[3] A set is *A-finite* if it is an element of A.

Gregory's Theorem

Theorem. Let A be a countable admissible set. Suppose T is a set of L_A sentences that is Σ_1 on A. If T has a pair of countable models M and N s.t. N is a proper L_A-elementary extension of M, then T has an uncountable model.

Gregory said that there were known examples (presumably due to other people) showing that the condition T is Σ_1 on A cannot be dropped. He did not give an example.

Last summer, John Baldwin asked for such an example, in connection with work with Martin Körwein, Typani Hyttinen, and Sy Friedman. I will describe an example constructed this fall with three students: Jesse Johnson, Victor Ocasio, and Steven VanDenDriessche.

The least admissible set

Let $A = L_{\omega_1^{CK}}$—this is the least admissible set containing ω.

Facts

1. The A-finite sets are just the hyperarithmetical sets.
2. The L_A-formulas are essentially the "computable" infinitary formulas—the infinite disjunctions and conjunctions are over c.e. sets. We write $\mathcal{M} \prec_\infty \mathcal{N}$ if $\mathcal{M}$ is an L_A-elementary substructure of $\mathcal{N}$.
3. The sets that are Σ_1 on A are the Π^1_1 sets—with a definition of the form $(\forall f \in \omega^\omega)\,(\exists s)\,R(f|s, s, n)$, where R is computable.

Example

Johnson-K-Ocasio-VanDenDriessche. There is a set T of computable infinitary sentences, in a computable language L, s.t. T has exactly two models up to isomorphism, $\mathcal{M}$ and $\mathcal{N}$, where $\mathcal{M} \prec_\infty \mathcal{N}$. Moreover, for each computable ordinal α, the set of computable Σ_α sentences in T is hyperarithmetical.

Language of T and universes of $\mathcal{M}$ and $\mathcal{N}$

The language of T consists of unary predicates U_n for $n \in \omega$. Each L-structure represents a family of sets. The set represented by an element x is the set of n s.t. $U_n x$ holds.

The universe of $\mathcal{M}$ is an infinite computable set of constants C, which is partitioned effectively into infinitely many infinite sets C_n. The extra element of $\mathcal{N}$ is a further constant a. We identify the constants with the sets they represent, once we have determined these sets.

Sets represented in $\mathcal{M}$ and $\mathcal{N}$

The set a will be Cohen generic, or at least "hyperarithmetically" generic.

We choose an increasing sequence of computable ordinals $(\alpha_n)_{n\in\omega}$ with limit ω_1^{CK}.

For all $c \in C_n$ and all $k < n$, $U_k c$ iff $U_k a$. Apart from this, the elements of C_n will be mutually α_n-generic, and uniformly $\Delta^0_{\alpha_n+1}$.

The theory T and its models

The computable infinitary theory of $\mathcal{M}$ and $\mathcal{N}$ includes the following.

1. sentences saying that all elements that are not $\Delta^0_{\alpha_n+1}$ satisfy the same predicates U_k for $k \leq n$,
2. sentences saying exactly which $\Delta^0_{\alpha_n+1}$ sets are represented,
3. a sentence saying that distinct elements differ on some U_k.

These sentences are enough to guarantee the following.

Proposition. $\mathcal{M}$ and $\mathcal{N}$ are the only models of T, up to isomorphism.

Controlling truth in $\mathcal{M}$ and $\mathcal{N}$

We choose a in advance. We use iterated forcing to choose the families of sets C_n, first C_0, then C_1, etc. This means that we do not decide truth in $\mathcal{M}$ or $\mathcal{N}$ directly.

Let $\mathcal{M}_n$ be the structure with universe $\cup_{k\leq n} C_k$, and let $\mathcal{N}_n$ be the structure with universe $\cup_{k\leq n} C_k \cup \{a\}$.

Lemma 1. $\mathcal{M}_n \prec_{\alpha_n} \mathcal{M}_{n+1}$

Lemma 2. $\mathcal{M}_n \prec_{\alpha_n} \mathcal{N}_n$

Lemma 3. $\mathcal{N}_n \prec_{\alpha_n} \mathcal{N}_{n+1}$

Showing that $\mathcal{M} \prec_\infty \mathcal{N}$

From the three lemmas, it is easy to finish.

Proposition. $\mathcal{M} \prec_\infty \mathcal{N}$

Proof. Suppose $\mathcal{N} \models \varphi(\overline{c})$, where $\varphi(\overline{c})$ is computable Σ_{α_n} and $\overline{c}$ is in $\mathcal{M}_n$. Since $\mathcal{M}_n \prec_{\alpha_n} \mathcal{N}_n \prec_{\alpha_n} \mathcal{N}$, $\varphi(\overline{c})$ holds in $\mathcal{N}_n$ and in $\mathcal{M}_n$. Since $\mathcal{M}_n \prec_{\alpha_n} \mathcal{M}$, it also holds in $\mathcal{M}$.

Hints on the proof of Lemma 1

- Suppose $\mathcal{M}_{n+1} \models \varphi(\overline{c}, b)$, where $\varphi(\overline{u}, x)$ is a computable infinitary Σ_{α_n} formula, $\overline{c}$ is in in $\mathcal{M}_n$ and $b \in \mathcal{M}_{n+1}$. Let φ be the natural computable infinitary propositional formula saying that $\mathcal{M}_{n+1} \models \varphi(\overline{c}, b)$. Since φ is true, it is forced by some p saying finitely much about finitely many elements of C_{n+1}.
- There is a computable Σ_{α_n} formula $\psi_{\varphi,p}$ characterizing the possible structures $\mathcal{M}_n$ s.t. when we choose C_{n+1}, p will force φ.
- Since $\psi_{\varphi,p}$ is true in $\mathcal{M}_n$, it is forced by some q saying finitely much about finitely many elements of C_n.
- Choose a computable permutation f of $\mathcal{M}_{n+1}$ fixing the elements of $\cup_{k<n} C_k$, $\overline{c}$, and elements mentioned in q, and taking b to some $b' \in C_n$ s.t. $f(p)$ is true. Then $f(\varphi)$ holds in $\mathcal{M}_{n+1}$ and it says that $\mathcal{M}_{n+1} \models \varphi(\overline{c}, b')$.

Conclusion

Does anyone know one of the 40-year old examples that Gregory referred to? Does it resemble ours?

I would like to advertise John Baldwin's talk in the ASL part of this meeting—fresh evidence of connections among different branches of logic.