Querying the Universe: The Dodd-Jensen Core Model and Ordinal Turing Machines

Nathanael Leedom Ackerman
University of Pennsylvania

Effective Mathematics of the Uncountable
CUNY

Aug. 20, 2009

Outline

(1) Review Of Ordinal Turing Machines

(2) Oracles And Absoluteness

(3) Query Machines

(4) Writing The Core Model

(5) Query Machines And Oracles (if time allows)

Ordinal Turing Machines

An *Ordinal Turing Machine* T consists of

- Four ordinal length, 0 or 1 valued, tapes: $\mathit{Tape}(\mathit{Input})$, $\mathit{Tape}(\mathit{Output})$, $\mathit{Tape}(\mathit{Parameter})$ and $\mathit{Tape}(\mathit{Work})$.
- *State*: A finite set of states including two distinguished states, *Start* and *Halt*.
- *Program*: A function from $\mathit{State} \times \{0,1\}^4 \to \mathit{State} \times \{0,1\}^4 \times \{\mathit{left}, \mathit{right}, \mathit{neither}\}^4$.

Notice that an ordinal Turing machine is completely determined by $\langle \mathit{State}, \mathit{Program} \rangle$ (which we call its *code*). We denote by OTM the collection of all codes of Ordinal Turing Machines.

Run of an Ordinal Turing Machines

Given a code for an ordinal Turing machine $T = \langle State, Program \rangle$, a *run* of T consists of

- For each tape $Tape(i)$ a function $Value(i) : Ord \to 2^{Ord}$
- A function $Position : Ord \to Ord^4$ such that $Position(0) = \langle 0, 0, 0, 0 \rangle$
- A function $CurState : Ord \to State$ such that $CurState(0) = Start$

such that they are consistent with *Program* (in the standard way).

We want to think of these functions as taking the a time and returning the condition of the machine at that time.

Run of an Ordinal Turing Machines on Input

A run of T on input $I \in 2^{Ord}$ with parameters $P \in Ord^{<\omega}$ is a run such that

- $Value(Input)(0) = I$
- $Value(Parameters)(0) =$ Characteristic Function of P
- $Value(Output)(0) = Value(Work)(0) = \emptyset$

Notice that a run is completely determined by its starting values and the code for the ordinal Turing machine.

Writing Sets

Definition

A run *writes a class* $S \subseteq Ord$ if it enters a halt state at a time t and $Value(Output)(t) = S$.

We say an ordinal Turing machine T writes a class $S \subseteq Ord$ if there is a finite set of ordinals P such that the run of T with input $\emptyset$ with parameters P writes S.

Writing Sets

Definition

A run *writes a class* $S \subseteq Ord$ if it enters a halt state at a time t and $Value(Output)(t) = S$.

We say an ordinal Turing machine T writes a class $S \subseteq Ord$ if there is a finite set of ordinals P such that the run of T with input $\emptyset$ with parameters P writes S.

Our first observation is

Theorem

If $T \in OTM$ writes $S \subseteq Ord$ then S is a set.

Proof.

If a run halts at time t then the output tape is a subset of t. □

Writing L

A natural question is

Question: What sets can be written?

Writing L

A natural question is

Question: What sets can be written?

Theorem

Every set that can be written by a $T \in OTM$ is in L.

Proof.

Every code for an ordinal Turing machine is in L, every finite subset of the ordinals is in L, and runs of an OTM on fixed input are absolute between models of set theory. □

Writing L

A natural question is

Question: What sets can be written?

Theorem

Every set that can be written by a $T \in OTM$ is in L.

Proof.

Every code for an ordinal Turing machine is in L, every finite subset of the ordinals is in L, and runs of an OTM on fixed input are absolute between models of set theory. □

Theorem (Koepke)

For every set $S \in Powerset(Ord) \cap L$ there is a $T \in OTM$ such that T writes S

Writing Inner Models

This suggests the following definition:

Definition

Suppose M is an class inner model of set theory. Suppose $GTuring$ is a collection of codes for "generalized ordinal Turing machines" (with ordinal tapes and time). We say that $GTuring$ *writes* M if the collection of sets which are written by a $T \in GTuring$ is $Powerset(Ord) \cap M$.

Writing Inner Models

This suggests the following definition:

Definition

Suppose M is an class inner model of set theory. Suppose *GTuring* is a collection of codes for "generalized ordinal Turing machines" (with ordinal tapes and time). We say that *GTuring writes M* if the collection of sets which are written by a $T \in GTuring$ is $Powerset(Ord) \cap M$.

We now have the question:

Question: What class inner models can be written by more general ordinal Turing machines?

Oracles

Definition

Suppose $A \subseteq Ord$. A *Ordinal Turing Machine with Oracle A* is an ordinal Turing machine with (another) distinguished tape, $Tape(Oracle)$.

A run of such a machine is defined exactly as a run of an ordinary ordinal Turing machine except that $Value(Oracle)$ is the constant function A. (i.e. the oracle tape always has A written on it).

A code for such a machine is a pair $\langle T, A \rangle$ where T is a code for an ordinal Turing machine. We let $OTM(A)$ be the collection of codes of ordinal Turing machines with oracle A

Notice that a run of an element of $OTM(A)$ is completely determined by the input, the parameters, and the code.

Writing Sets From Oracles

Definition

We say that a set S is *written* by $OTM(A)$ if there is an element $\langle T, A\rangle \in OTM(A)$, a finite set of ordinals P, and a run of $\langle T, A\rangle$ with input $\emptyset$ and parameters P such that the run halts with S on the output tape.

Writing Sets From Oracles

Definition

We say that a set S is *written* by $OTM(A)$ if there is an element $\langle T, A\rangle \in OTM(A)$, a finite set of ordinals P, and a run of $\langle T, A\rangle$ with input $\emptyset$ and parameters P such that the run halts with S on the output tape.

Theorem

All sets which can be written by $OTM(A)$ are in $\bigcup_{\alpha \in Ord} L[A \cap \alpha]$ $(= L[OTM(A)] = L[A])$

Proof.

Suppose S is written by a run of $\langle T, A\rangle \in OTM(A)$ with parameters P which halts at time t. Then S is also written by a run $\langle T, A \cap t\rangle \in OTM(A)$ with parameters P. □

Writing Inner Models

Theorem (Koepke)

If A is a set then $OTM(A)$ writes $L[A]$.

Proof.

A slight modification of Koepke's proof. □

Corollary

For all class $A \subseteq Ord$, $OTM(A)$ writes exactly those sets in $\bigcup_{\alpha \in Ord} L[A \cap \alpha]$

Corollary

For every class inner model M, there is a class $A_M \subseteq Ord$ such that $OTM(A_M)$ writes M

Absoluteness and Generalized Ordinal Turing Machines

While every model of set theory can be written by an ordinal Turing machine with some oracle, in general, the oracle needs to be a class which equi-constructible with the entire universe.

Absoluteness and Generalized Ordinal Turing Machines

While every model of set theory can be written by an ordinal Turing machine with some oracle, in general, the oracle needs to be a class which equi-constructible with the entire universe.

This suggests the question

Question: Is there a more general notion of an ordinal Turing machine (whose collection of "codes" forms a set C) but which writes a class inner model larger than $L[C]$ (when one exists)?

Answer: No

Answer: No...If we assume that runs of the machines are absolute between class inner models.

Proof: Every finite set of ordinals is in $L[C]$ and so every set which can be written in V by our general notion of Turing machines can be written in $L[C]$ with the same run (as the runs are absolute).

Absoluteness and Generalized Ordinal Turing Machines

Answer: No...If we assume that runs of the machines are absolute between class inner models.

Proof: Every finite set of ordinals is in $L[C]$ and so every set which can be written in V by our general notion of Turing machines can be written in $L[C]$ with the same run (as the runs are absolute).

In particular if we want to write a class inner model which is larger than L with a generalized ordinal Turing which is contained in L, then our notion of computation must not be absolute.

Or put another way, our generalized notion of a Turing machine must be able to "ask questions" about the ambient set theoretic universe.

Set Theory With A Choice Function

In order to make precise the notion of a machine that can ask questions of the ambient universe we first must define which set theory we are using.

Definition

Let $L_{ST} = \{\in, V, F\}$ and let $GBCF$ be the theory in L_{ST} containing

- The Axioms of Gödel-Berney's Set Theory.
- $\{(x, y) : F(x) = y\}$ is a class.
- $F : V \to Ordinals$ is a bijection.

In particular our set theory satisfies the Axiom of Global Choice with a preferred choice function.

Definition of a Query

Definition

We call a question that our generalized ordinal Turing machine can ask of the universe a *query*.

Specifically a *query* is a formula (possibly with parameters) which is the graph of a function from $Powerset(Ord)$ to $Powerset(Ord)$ in any class inner model of $GBCF$ (containing the parameters).

Definition of a Query

Definition

We call a question that our generalized ordinal Turing machine can ask of the universe a *query*.

Specifically a *query* is a formula (possibly with parameters) which is the graph of a function from $Powerset(Ord)$ to $Powerset(Ord)$ in any class inner model of $GBCF$ (containing the parameters).

Suppose $\varphi(x, y)$ is a formula such that every class inner model satisfies $(\forall x)(\exists y)\varphi(x, y)$. Then there is a query associated to $\varphi(x, y)$ given by:

$$\varphi^F(x, y) \Leftrightarrow \varphi(x, y) \wedge (\forall z)\varphi(x, z) \rightarrow F(z) \leq F(y)$$

Pure/Non-Pure Queries

Definition

We say a query is *pure*, if the formula describing it is in the language $L_\in = \{\in\}$ (possibly with parameters)

Definition

We say a query is $\Delta_n^F(A)$ if it is equivalent to a formula $\varphi^F(x, y)$ where $\varphi(x, y)$ is a Δ_n formula in $\{\in\}$ with parameter A.

Query Machines

Definition

A *Query Machine with query* $Q(x) = y$ is an ordinal Turing Machine with two extra distinguished tapes: $Tape(QueryIn)$ and $Tape(QueryOut)$.

A run of such a machine is defined as a run of an ordinary ordinal Turing machine except that

$$(\forall t \in Ord)Q(Value(QueryIn)(t)) = Value(QueryOut)(t)$$

A code for such a machine is a pair $\langle T, Q, A\rangle$ where T is a code for an ordinal Turing machine, Q is the query and A is the parameter used in Q. We let $OTM(Q)$ be the collection of codes of query machines with query Q

A Query Machine with Query Q is such that $Tape(QueryOut)$ always has the result of applying Q to $Tape(QueryIn)$.

Runs of Query Machines

Definition

We say that a set S is written by $OTM(Q)$ if there is an element $T \in OTM(Q)$, a finite set of ordinals P, and a run of T with input $\emptyset$ and parameters P such that the run halts with S on the output tape.

Note a run is determined (in a fixed class inner model M) by its starting conditions as well as the code for the query machine.

Runs of Query Machines

Definition

We say that a set S is written by $OTM(Q)$ if there is an element $T \in OTM(Q)$, a finite set of ordinals P, and a run of T with input $\emptyset$ and parameters P such that the run halts with S on the output tape.

Note a run is determined (in a fixed class inner model M) by its starting conditions as well as the code for the query machine.

Theorem

Suppose $\langle Q_i : i \in \beta \rangle$ is a set size collection of queries. Then we can create a single query, Q^*, from which they all can be recovered.

Proof.

We can encode the sequence $\langle Q_i(x) : i \in \beta \rangle$ as a single subset of ordinals in a uniform way. I.e. $Q^*(x)(\beta \cdot \alpha + i) = Q_i(x)(\alpha)$. □

Absolute Queries

Theorem

If Q is a $\Delta_1(A)$ query then the collection of sets written by $OTM(Q)$ in V is the same as the collection of sets written by $OTM(Q)$ in $L[A]$.

Proof.

If $x \in L[A]$ then $Q^{L[A]}(x) \in L[A]$. But because Q is a $\Delta_1(A)$ function it is absolute. Hence $Q^V(x) = Q^{L[A]}(x)$. So any run of an element of $OTM(Q)$ in V is the same as that in $L[A]$. □

Hence if we only use $\Delta_1(A)$ queries we can never written a model larger than the one containing the collection of codes for the query machines.

Writing the Universe

If we allow Δ_1^F queries instead of just Δ_1 queries, then we loose absoluteness.

Theorem

There is a Δ_0^F query Q_V such that $OTM(Q_V)$ writes V.

Proof.

Let $\varphi(x, y)$ be the statement "If x is treated as a subset of $Powerset(Ord)$ then $y \notin x$" and let $Q_V(x) = y \Leftrightarrow \varphi^F(x, y)$

So $Q_V(x) = y$ if y is the F minimal element of the universe not in x. With Q_V we can recover $F^{-1}(Ord) \cap Powerset(Ord)$ by repeated queries. Hence as $F^{-1}(Ord) = V$ every set of ordinals in V can be written by $OTM(Q_V)$. □

Notice in particular that $OTM(Q_V)$ is in L even though $OTM(Q_V)$ writes V.

Writing Intermediate Models

We have seen that query machines can write L as well as write the entire universe. This suggests the question:

Question: Is there a query Q (with $OTM(Q) \in L$) which can write a class inner model strictly between L and V (under some conditions)?

Writing Intermediate Models

We have seen that query machines can write L as well as write the entire universe. This suggests the question:

Question: Is there a query Q (with $OTM(Q) \in L$) which can write a class inner model strictly between L and V (under some conditions)?

Answer: Yes. Further, we can find a pure query Q^{DJ} such that the $OTM(Q^{DJ})$ always writes the Dodd-Jensen core model.

But before we do this lets give a brief review of the core model.

Outline of the Core Model Below a Measurable

Definition

A *Dodd-Jensen Mouse* is a transitive model $M = J_\alpha^U$ such that

(i) U is a normal κ-complete iterable M-ultrafilter on some $\kappa < \alpha$

(ii) All iterated ultrapowers of M by U are well-founded

(iii) $M = H_1^M(\gamma \cup p)$ (the Σ_1 Skolem hull) for some $\gamma < \kappa$ and some finite $p \subseteq \alpha$

Definition

The *Dodd-Jensen Core Model* is $K^{DJ} = L[\{M : M \text{ is a mouse}\}]$

Important Property of Mice

For our purposes the most important properties of mice is

Theorem

For every mouse M there is a minimal pair (β_M, λ_M) with β_M an ordinal and λ_M a cardinal such that

$$L[M] = L[J_{\beta_M}^{C_{\lambda_M}}]$$

(where C_{λ_M} is the closed unbounded filter on λ_M)

Corollary

$K^{DJ} = L[\{J_{\beta_M}^{C_{\lambda_M}} : M \text{ is a mouse}\}]$

Writing the Core Model Below a Measurable

Theorem

There is a pure query Q_K such that $OTM(Q_K)$ writes K^{DJ} in any model of set theory.

Proof.

Let $Q_K(x) = y$ be the formula which says

Case 1: If $x = \langle \alpha, \beta, \lambda \rangle$ then

$y = L_\alpha[\{J_{\beta_M}^{C_{\lambda_M}} : \beta_M < \beta, \lambda_M < \lambda \text{ and } M \text{ is a mouse}\}] \cap$ *Powerset*(*Ord*) ordered by the canonical ordering of L

Case 2: Otherwise

$$y = \bigcup x$$

Writing the Core Model Below a Measurable

Proof.

It is then clear that in any class inner model $Q_K(x) = y$ is the graph of a function from sets of ordinals to sets of ordinals. Hence Q_K is a pure query.

It is also clear that that every set in $Powerset(Ord) \cap K^{DJ}$ can be written by a query machine in $OTM(Q_K)$.

However, if S can be written by a machine $OTM(Q_K)$ which halts at time t then S must be in $L_{|t|^{++}}[\{J_{\beta_M}^{C_{\lambda_M}} : |\beta_M|, |\lambda_M| < |t|^+$ and M is a mouse$\}]$. Hence S must be in K^{DJ} and $OTM(Q_K)$ writes K^{DJ}. □

Query Machines and Oracles

Theorem

For every $A \subseteq Ord$ there is a $\Delta_0(A)$ query machine Q_A such that $OTM(Q_A)$ writes the same sets as $OTM(A)$

Proof.

Let $Q(x) = A \cap \bigcup x$ □

Query Machines and Oracles

Theorem

For every $A \subseteq Ord$ there is a $\Delta_0(A)$ query machine Q_A such that $OTM(Q_A)$ writes the same sets as $OTM(A)$

Proof.

Let $Q(x) = A \cap \bigcup x$ □

Theorem

For every $\Sigma_n(A)$ query Q there is a $\Sigma_n(A)^V$ oracle O_Q such that $OTM(Q)$ writes the same sets as $OTM(O_Q)$

Proof.

Let O^Q encode all output of all the query machines run with Q □

Query Machines and Oracles

Theorem

There is a Δ_0^F query Q such that for every model $M \models GBC$ and every $A \in M$ there is an expansion of M to a model $M_A \models GBCF$ where $OTM(Q)$ (in M_A) writes the same sets as $OTM(A)$.

Proof.

Let $\varphi(x, y) = (\bigcup x) \cdot 3 \leq y \leq (\bigcup x) \cdot 3 + 1$ and let $Q(x) = y \Leftrightarrow \varphi^F(x, y)$.

Next let be any bijection such that if $x \in Ord$

- $F(x) = \gamma \cdot 3$ when $x \in A$
- $F(x) = \gamma \cdot 3 + 1$ when $x \notin A$.

Given a machine in $OTM(A)$ we can find a machine in $OTM(Q)$ which mimics it (when run in M_F). Likewise, given a machine in $OTM(Q)$ (run in M_F) we can find a machine in $OTM(A)$ which mimics it.