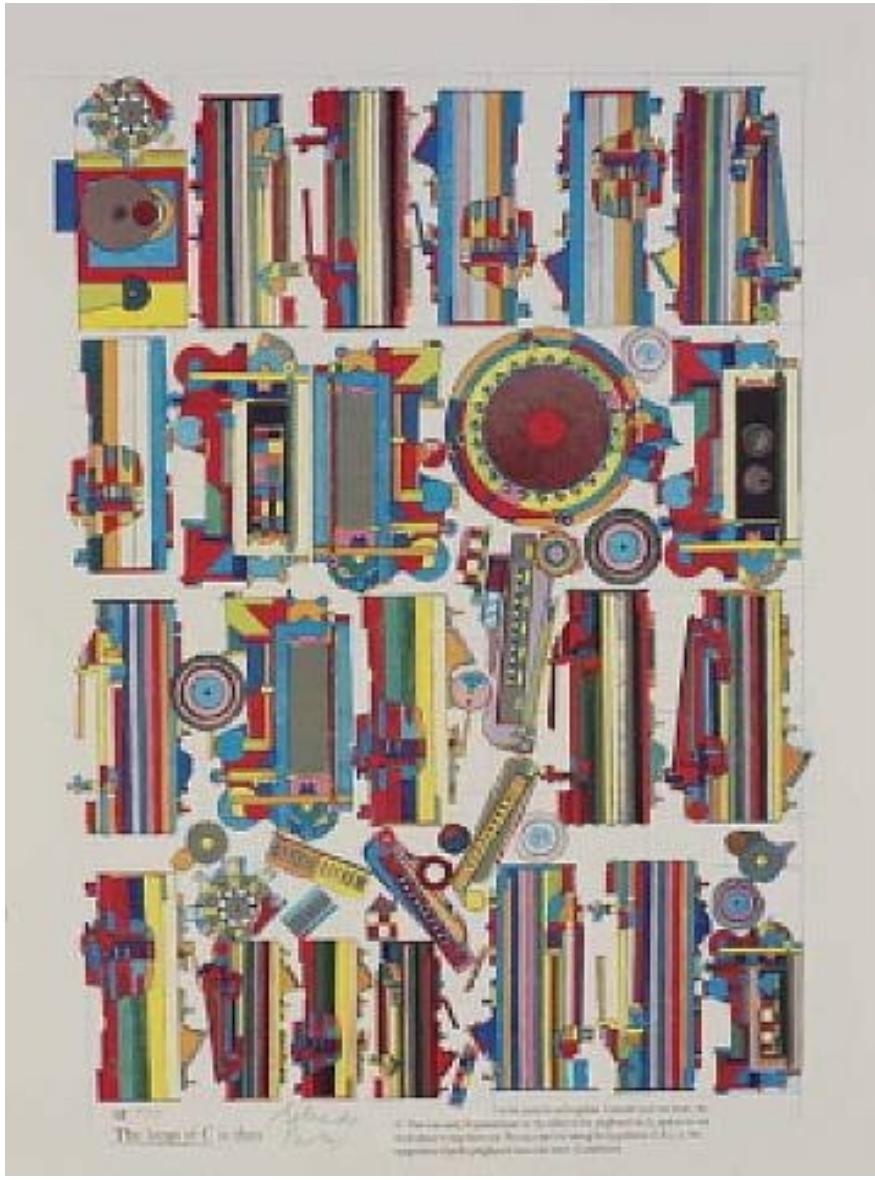
Transfinite dynamical models

P.D.Welch, University of Bristol,

CIE 2012, Cambridge



Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*
  - α-ITTM Dawson

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*
  - α-ITTM Dawson
- ITRM (Infinite Time *R*egister Machine) Koepke-Miller *et al.*

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*
  - α-ITTM Dawson
- ITRM (Infinite Time *R*egister Machine) Koepke-Miller *et al.*
- IBSM (Infinite Time *B*lum-*S*hub-Smale Machines) Various

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*
  - α-ITTM Dawson
- ITRM (Infinite Time *R*egister Machine) Koepke-Miller *et al.*
- IBSM (Infinite Time *B*lum-*S*hub-Smale Machines) Various

Above here: all some form of Liminf component to the architecture

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*
  - α-ITTM Dawson
- ITRM (Infinite Time *R*egister Machine) Koepke-Miller *et al.*
- IBSM (Infinite Time *B*lum-*S*hub-Smale Machines) Various

Above here: all some form of Liminf component to the architecture

- CIBSM (*C*ontinuous IBSM's) Koepke-Seyfferth

Introduction: Part I Various models

- ITTM (*I*nfinite *T*ime *T*uring *M*achines) - Hamkins-Kidder
  - FITTM (*F*eedback-ITTM's) Lubarsky
  - OTTM (*O*rdinal *T*ape *T*uring *M*achines) Koepke *et al.*
  - α-ITTM Dawson
- ITRM (Infinite Time *R*egister Machine) Koepke-Miller *et al.*
- IBSM (Infinite Time *B*lum-*S*hub-Smale Machines) Various

Above here: all some form of Liminf component to the architecture

- CIBSM (*C*ontinuous IBSM's) Koepke-Seyfferth
- HM (*H*ypermachines, the class of all Σ_n-HM's) Friedman-Welch

Part II Various Recursional Theories

Part II Various Recursional Theories

- Kleene Recursion
- Higher Type Recursion Theory Kleene, Sacks, Gandy . . .

Part II Various Recursional Theories

- Kleene Recursion
- Higher Type Recursion Theory — Kleene, Sacks, Gandy . . .
- Inductive Definitions — Aczel, Richter, Cenzer, Moschovakis . . .
- Induction on Abstract Structures — Moschovakis
  leading to Spector Class Theory
- Quasi-Inductive Definitions — Belnap-Gupta, Burgess

Part II Various Recursional Theories

- Kleene Recursion
- Higher Type Recursion Theory — Kleene, Sacks, Gandy . . .
- Inductive Definitions — Aczel, Richter, Cenzer, Moschovakis . . .
- Induction on Abstract Structures — Moschovakis
  leading to Spector Class Theory
- Quasi-Inductive Definitions — Belnap-Gupta, Burgess
- α-Recursion — Kreisel, Sacks, . . .

Part II Various Recursional Theories

- Kleene Recursion
- Higher Type Recursion Theory — Kleene, Sacks, Gandy . . .
- Inductive Definitions — Aczel, Richter, Cenzer, Moschovakis . . .
- Induction on Abstract Structures — Moschovakis
  leading to Spector Class Theory
- Quasi-Inductive Definitions — Belnap-Gupta, Burgess
- α-Recursion — Kreisel, Sacks, . . .
- Rudimentary and Primitive Recursion on sets — Gandy, Jensen, Karp
- Set Recursion — Normann, Sacks
- Safe Set Recursion — Beckmann, Buss, Friedman

Desiderata

Desiderata for a good transfinite version of a machine:

Desiderata

Desiderata for a good transfinite version of a machine:

Theory: a good theory which might include items such as versions of Recursion Theorem, S_n^m-Theorem, Universality, or *Weak Universality*: codes of courses-of-computations within the machine-class must be produceable within that machine-class . . ..

Desiderata

Desiderata for a good transfinite version of a machine:

Theory: a good theory which might include items such as versions of Recursion Theorem, S_n^m-Theorem, Universality, or *Weak Universality*: codes of courses-of-computations within the machine-class must be produceable within that machine-class . . ..

Robustness: links to other forms of inductive definition or a recursional theory (of the types just listed).

Desiderata

Desiderata for a good transfinite version of a machine:

Theory: a good theory which might include items such as versions of Recursion Theorem, S_n^m-Theorem, Universality, or *Weak Universality*: codes of courses-of-computations within the machine-class must be produceable within that machine-class . . ..

Robustness: links to other forms of inductive definition or a recursional theory (of the types just listed).
• Ideally the 'semi-decidable' sets of integers (and of reals) should form a Spector class.

Desiderata

Desiderata for a good transfinite version of a machine:

Theory: a good theory which might include items such as versions of Recursion Theorem, S_n^m-Theorem, Universality, or *Weak Universality*: codes of courses-of-computations within the machine-class must be produceable within that machine-class . . ..

Robustness: links to other forms of inductive definition or a recursional theory (of the types just listed).

- Ideally the 'semi-decidable' sets of integers (and of reals) should form a Spector class.
- When not, then some other proof theoretic feature, or independent representation may be at hand.

Kleene or "Δ^1_1"-recursion

- Rogers called this the "$\aleph_0$-mind": one has machine with a countable tape and work area, and ability to inspect, and manipulate the tape doing countably many operations in a finite time; the oracle version has the ability to query of a set of reals A whether a real that has been computed so far is, or is not, in A.

Kleene or "Δ^1_1"-recursion

• Rogers called this the "$\aleph_0$-mind": one has machine with a countable tape and work area, and ability to inspect, and manipulate the tape doing countably many operations in a finite time; the oracle version has the ability to query of a set of reals A whether a real that has been computed so far is, or is not, in A.

• Equivalent version with a Turing machine being fed an input x step-wise and making a tree T of subcomputation calls to other TM's . . .
$T \in WF$ iff the overall computation converges.

Kleene or "Δ^1_1"-recursion

• Rogers called this the "$\aleph_0$-mind": one has machine with a countable tape and work area, and ability to inspect, and manipulate the tape doing countably many operations in a finite time; the oracle version has the ability to query of a set of reals A whether a real that has been computed so far is, or is not, in A.

• Equivalent version with a Turing machine being fed an input x step-wise and making a tree T of subcomputation calls to other TM's . . .
$T \in WF$ iff the overall computation converges.

• The semi-decidable sets of integers or reals are the $\mathbf{\Pi^1_1}$ sets, and the decidable the Borel (or hyperarithmetic).

Kleene or "Δ^1_1"-recursion

• Rogers called this the "$\aleph_0$-mind": one has machine with a countable tape and work area, and ability to inspect, and manipulate the tape doing countably many operations in a finite time; the oracle version has the ability to query of a set of reals A whether a real that has been computed so far is, or is not, in A.

• Equivalent version with a Turing machine being fed an input x step-wise and making a tree T of subcomputation calls to other TM's . . .
$T \in WF$ iff the overall computation converges.

• The semi-decidable sets of integers or reals are the $\mathbf{\Pi^1_1}$ sets, and the decidable the Borel (or hyperarithmetic).

• Thus, for computations on integers, the natural "companion structure" here is $L_{\omega_1^{ck}}$ whose reals are precisely the Kleene-decidable sets of integers, the *hyperarithmetic* sets.

• Π^1_1 is the original, first Spector class.

ITTM's[1]

• Here the *Liminf* process at limit stages enables complex sets to be built up. One gets the complete Spector class theory: the semi-decidable sets of integers are precisely those $\Sigma_1(L_\zeta)$ where ζ is least such that

$$\exists\Sigma(\zeta < \Sigma \wedge L_\zeta \prec_{\Sigma_2} L_\Sigma) \quad - \quad (\zeta \text{ is "}\Sigma_2\text{-extendible"}).$$

[1] Hamkins & Lewis *Infinite Time Turing Machines*, JSL, 2000.

ITTM's[1]

• Here the *Liminf* process at limit stages enables complex sets to be built up. One gets the complete Spector class theory: the semi-decidable sets of integers are precisely those $\Sigma_1(L_\zeta)$ where ζ is least such that

$$\exists\Sigma(\zeta < \Sigma \wedge L_\zeta \prec_{\Sigma_2} L_\Sigma) \quad - \quad (\zeta \text{ is "}\Sigma_2\text{-extendible"}).$$

• Indeed this pointclass, Γ_2, is normed and enjoys Uniformisation and Scale properties.

[1] Hamkins & Lewis *Infinite Time Turing Machines*, JSL, 2000.

Σ_n-HM's

The Σ_n-*hypermachines*[2] successfully lift all the above from Σ_2 to Σ_n:

[2] Friedman-Welch: *Hypermachines*, JSL June 2011

Σ_n-HM's

The Σ_n-*hypermachines*[2] successfully lift all the above from Σ_2 to Σ_n:

• The semi-decidable sets of integers are precisely those $\Sigma_1(L_\zeta)$ where $\zeta(n)$ is least such that

$$\exists\Sigma(n)\ (\zeta(n) < \Sigma(n) \wedge L_{\zeta(n)} \prec_{\Sigma_n} L_{\Sigma(n)}) \quad - \quad (\zeta(n) \text{ is "}\Sigma_n\text{-extendible"}).$$

• Again this pointclass, Γ_n, is normed and again has the Uniformisation and Scale properties.

[2] Friedman-Welch: *Hypermachines*, JSL June 2011

Σ_n-HM's

The Σ_n-*hypermachines*[2] successfully lift all the above from Σ_2 to Σ_n:

• The semi-decidable sets of integers are precisely those $\Sigma_1(L_\zeta)$ where $\zeta(n)$ is least such that

$$\exists\Sigma(n)\ (\zeta(n) < \Sigma(n) \wedge L_{\zeta(n)} \prec_{\Sigma_n} L_{\Sigma(n)}) \quad - \quad (\zeta(n) \text{ is "}\Sigma_n\text{-extendible"}).$$

• Again this pointclass, Γ_n, is normed and again has the Uniformisation and Scale properties.

• Thus every real in the least model of second order number theory is HM-computable.

---

[2] Friedman-Welch: *Hypermachines*, JSL June 2011

FITTM's: ITTM's with *Feedback*

• Lubarsky's idea: expand the ITTM to allow for 'oracle calls' to ask whether the current worktape w when input to $\widetilde{P}_e$ results in a looping program with constant output; of course $\widetilde{P}_e$ will also be an FITTM program that may elicit further such oracle calls . . ..

FITTM's: ITTM's with *Feedback*

- Lubarsky's idea: expand the ITTM to allow for 'oracle calls' to ask whether the current worktape w when input to $\widetilde{P}_e$ results in a looping program with constant output; of course $\widetilde{P}_e$ will also be an FITTM program that may elicit further such oracle calls . . ..
- Again for a given computation, there's an underlying tree that may be ill-founded - in which case he says the computation *freezes* - so no overall output.

FITTM's: ITTM's with *Feedback*

• Lubarsky's idea: expand the ITTM to allow for 'oracle calls' to ask whether the current worktape w when input to $\widetilde{P}_e$ results in a looping program with constant output; of course $\widetilde{P}_e$ will also be an FITTM program that may elicit further such oracle calls . . ..

• Again for a given computation, there's an underlying tree that may be ill-founded - in which case he says the computation *freezes* - so no overall output.

Definition

Let $A^{\blacktriangledown} =_{df} \{(e,x) \mid \widetilde{P}_e^A(x)$ *enters a loop but with fixed output*$\}$

FITTM's: ITTM's with *Feedback*

• Lubarsky's idea: expand the ITTM to allow for 'oracle calls' to ask whether the current worktape w when input to $\widetilde{P}_e$ results in a looping program with constant output; of course $\widetilde{P}_e$ will also be an FITTM program that may elicit further such oracle calls . . ..
• Again for a given computation, there's an underlying tree that may be ill-founded - in which case he says the computation *freezes* - so no overall output.

Definition

Let $A^{\blacktriangledown} =_{df} \{(e,x) \mid \widetilde{P}_e^A(x)$ *enters a loop but with fixed output*$\}$

• *I.e.* we have an oracle that answers as to whether $?(e,w) \in 0^{\blacktriangledown}?$.

Question What kind of sets/ordinals are decidable by an FITTM?

FITTM's: ITTM's with *Feedback*

• Lubarsky's idea: expand the ITTM to allow for 'oracle calls' to ask whether the current worktape w when input to $\widetilde{P}_e$ results in a looping program with constant output; of course $\widetilde{P}_e$ will also be an FITTM program that may elicit further such oracle calls . . ..
• Again for a given computation, there's an underlying tree that may be ill-founded - in which case he says the computation *freezes* - so no overall output.

Definition

Let $A^{\blacktriangledown} =_{df} \{(e,x) \mid \widetilde{P}_e^A(x)$ *enters a loop but with fixed output*$\}$

• *I.e.* we have an oracle that answers as to whether $?(e,w) \in 0^{\blacktriangledown}?$.
Question What kind of sets/ordinals are decidable by an FITTM?
Conjecture $\supset \Sigma^0_3$ sets of integers are FITTM-computable.

ITRM's

• (Koepke-Miller) We take a standard N-register machine with entries $m \in \mathbb{N}$ and define a transfinite action on it - again a *Liminf* construction, for limit times λ we set

$$R_i(\lambda) = Liminf^*_{\alpha \to \lambda} R_i(\alpha)$$

(where the * means it resets to zero if it ever becomes ω).

ITRM's

• (Koepke-Miller) We take a standard N-register machine with entries $m \in \mathbb{N}$ and define a transfinite action on it - again a *Liminf* construction, for limit times λ we set

$$R_i(\lambda) = Liminf^*_{\alpha \to \lambda} R_i(\alpha)$$

(where the * means it resets to zero if it ever becomes ω).

• (Koepke-Miller) There is no *universal* ITRM. The strength of the machines increase with N. So there can be no corresponding Spector class for this notion.

ITRM's

• (Koepke-Miller) We take a standard N-register machine with entries $m \in \mathbb{N}$ and define a transfinite action on it - again a *Liminf* construction, for limit times λ we set

$$R_i(\lambda) = Liminf^*_{\alpha \to \lambda} R_i(\alpha)$$

(where the * means it resets to zero if it ever becomes ω).

• (Koepke-Miller) There is no *universal* ITRM. The strength of the machines increase with N. So there can be no corresponding Spector class for this notion.

• Let ITRM be the statement that every ITRM either loops or halts; then:

ITRM's

• (Koepke-Miller) We take a standard N-register machine with entries $m \in \mathbb{N}$ and define a transfinite action on it - again a *Liminf* construction, for limit times λ we set

$$R_i(\lambda) = Liminf^*_{\alpha \to \lambda} R_i(\alpha)$$

(where the * means it resets to zero if it ever becomes ω).

• (Koepke-Miller) There is no *universal* ITRM. The strength of the machines increase with N. So there can be no corresponding Spector class for this notion.

• Let ITRM be the statement that every ITRM either loops or halts; then:

$$\mathsf{ATR}_0 \vdash \mathsf{ITRM} \leftrightarrow \Pi^1_1\text{-}\mathsf{CA}_0.$$

ITRM's

• (Koepke-Miller) We take a standard N-register machine with entries $m \in \mathbb{N}$ and define a transfinite action on it - again a *Liminf* construction, for limit times λ we set

$$R_i(\lambda) = Liminf^*_{\alpha \to \lambda} R_i(\alpha)$$

(where the * means it resets to zero if it ever becomes ω).

• (Koepke-Miller) There is no *universal* ITRM. The strength of the machines increase with N. So there can be no corresponding Spector class for this notion.

• Let ITRM be the statement that every ITRM either loops or halts; then:

$$\mathsf{ATR}_0 \vdash \mathsf{ITRM} \leftrightarrow \Pi^1_1\text{-}\mathsf{CA}_0.$$

However we have weak universality: the companion structure is $L_{\omega^{ck}_{\omega}}$ - the first limit of admissibles, and all reals here are ITRM-computable.

Infinite Time Blum-Shub-Smale Machines: IBSM

• A standard BSS machine consists of
(i) a finite number N of registers which hold standard reals $r \in \mathbb{R}$;
(ii) a program of a finite flow diagram with algebraic operations or conditional jumps to be performed at nodes.

A typical instruction is of the form
$(i, \varphi, j, k, l) \in \omega \times \{f \mid f : \mathbb{R}^N \to \mathbb{R}^N, f \text{ rational } \} \times 2 \times \omega \times \omega$

Infinite Time Blum-Shub-Smale Machines: IBSM

• A standard BSS machine consists of
(i) a finite number N of registers which hold standard reals $r \in \mathbb{R}$;
(ii) a program of a finite flow diagram with algebraic operations or conditional jumps to be performed at nodes.

A typical instruction is of the form
$(i, \varphi, j, k, l) \in \omega \times \{f \mid f : \mathbb{R}^N \to \mathbb{R}^N, f \text{ rational } \} \times 2 \times \omega \times \omega$

We allow the machine to run transfinitely by requiring at limit stages λ:
(a) That the next instruction number i is the *liminf* of those as $\alpha \to \lambda$;
(b) requiring that register $R_i(\lambda) = \mathit{Liminf}^*_{\alpha \to \lambda} R_i(\alpha)$.

Infinite Time Blum-Shub-Smale Machines: IBSM

• A standard BSS machine consists of
(i) a finite number N of registers which hold standard reals $r \in \mathbb{R}$;
(ii) a program of a finite flow diagram with algebraic operations or conditional jumps to be performed at nodes.

A typical instruction is of the form
$(i, \varphi, j, k, l) \in \omega \times \{f \mid f : \mathbb{R}^N \to \mathbb{R}^N, f \text{ rational } \} \times 2 \times \omega \times \omega$

We allow the machine to run transfinitely by requiring at limit stages λ:
(a) That the next instruction number i is the *liminf* of those as $\alpha \to \lambda$;
(b) requiring that register $R_i(\lambda) = \mathit{Liminf}^*_{\alpha \to \lambda} R_i(\alpha)$.

Question: What is the strength of such a machine?

Continuous IBSM machines

Koepke-Seyfferth[3] require additionally *continuity* in (b) (o.w. the computation crashes).

[3] *Towards a theory of infinite time Blum-Shub-Smale machines"*, this conference volume

Continuous IBSM machines

Koepke-Seyfferth[3] require additionally *continuity* in (b) (o.w. the computation crashes).
They show:

Theorem (Koepke-Seyfferth)

Any CIBSM computation either crashes or halts in $< \omega^\omega$ steps, or is for ever looping.

Fact (K,S)

(i) Standard TM's can be simulated by IBSM's.
(ii) The standard halting problem is computable by a CIBSM: there is a CIBSM program P which halts with a real whose binary expansion is the characteristic of $0'$.

[3] *Towards a theory of infinite time Blum-Shub-Smale machines"*, this conference volume

Continuous IBSM machines

Koepke-Seyfferth[3] require additionally *continuity* in (b) (o.w. the computation crashes).
They show:

Theorem (Koepke-Seyfferth)

Any CIBSM computation either crashes or halts in $< \omega^\omega$ steps, or is for ever looping.

Fact (K,S)

(i) Standard TM's can be simulated by IBSM's.
(ii) The standard halting problem is computable by a CIBSM: there is a CIBSM program P which halts with a real whose binary expansion is the characteristic of $0'$.

Q.1 Does this model correspond to a recursional theory as listed earlier?
Q.2 It is natural to conjecture that any set of integers in L_{ω^ω} can be computed, and that CIBSM's are weakly universal.

---

[3] *Towards a theory of infinite time Blum-Shub-Smale machines"*, this conference volume

Indeed, CIBSM's are weakly universal

- We wish to see that a code for every course of an CIBSM-computation is also CIBSM-computable.

- Note that we can consider any CIBSM operating with integer, or recursive input, with recursive parameters, as being run inside the model L_{ω^k} for some sufficiently large $k < \omega$, as everything is very absolute.

Indeed, CIBSM's are weakly universal

• We wish to see that a code for every course of an CIBSM-computation is also CIBSM-computable.

• Note that we can consider any CIBSM operating with integer, or recursive input, with recursive parameters, as being run inside the model L_{ω^k} for some sufficiently large $k < \omega$, as everything is very absolute.

• Hence it suffices to show that every real $r \in L_{\omega^\omega}$ is CIBSM-writable, that is there is a CIBSM-program with input 0 which will output r.

Indeed, CIBSM's are weakly universal

• We wish to see that a code for every course of an CIBSM-computation is also CIBSM-computable.

• Note that we can consider any CIBSM operating with integer, or recursive input, with recursive parameters, as being run inside the model L_{ω^k} for some sufficiently large $k < \omega$, as everything is very absolute.

• Hence it suffices to show that every real $r \in L_{\omega^\omega}$ is CIBSM-writable, that is there is a CIBSM-program with input 0 which will output r.

Fact: Let $r \in \mathcal{P}(\mathbb{N}) \cap L_{\omega^\omega}$. Then $\exists \alpha < \omega^\omega (r \leq_1 h_\alpha)$ where we define:

$$h_0 = \mathbb{N}; \quad h_{\beta+1} = h_\beta \cap (h_\beta)'; \quad Lim(\lambda) \to h_\lambda = \bigcap_{\alpha<\lambda} h_\alpha.$$

Hence for any $\alpha < \omega^k$ there is $l < \omega$ so that h_α is computed by stage ω^{k+l}, then r can be computed from that.

Indeed, CIBSM's are weakly universal

• We wish to see that a code for every course of an CIBSM-computation is also CIBSM-computable.

• Note that we can consider any CIBSM operating with integer, or recursive input, with recursive parameters, as being run inside the model L_{ω^k} for some sufficiently large $k < \omega$, as everything is very absolute.

• Hence it suffices to show that every real $r \in L_{\omega^\omega}$ is CIBSM-writable, that is there is a CIBSM-program with input 0 which will output r.

Fact: Let $r \in \mathcal{P}(\mathbb{N}) \cap L_{\omega^\omega}$. Then $\exists \alpha < \omega^\omega (r \leq_1 h_\alpha)$ where we define:

$$h_0 = \mathbb{N}; \quad h_{\beta+1} = h_\beta \cap (h_\beta)'; \quad Lim(\lambda) \to h_\lambda = \bigcap_{\alpha<\lambda} h_\alpha.$$

Hence for any $\alpha < \omega^k$ there is $l < \omega$ so that h_α is computed by stage ω^{k+l}, then r can be computed from that.

• This shows that the 'companion-structure' to CIBSM computing is L_{ω^ω}.

Is there a recursion theory for CIBSM?

- The answer here is yes: *Safe Recursive Set Functions*[4]

Theorem

The least safe recursively closed set M_0 with $\omega \in M_0$ is L_{ω^ω}.

Conclusion: Since codes for sets in M_0 are computable by *CIBSM*'s, we can view this machine model as a computational vehicle for this theory of (set)-recursion.

[4] of *Beckmann, Buss, Friedman* to appear

What else can these machines do? What is L_{ω^ω} good for?

• Note: This is the first level beyond ω of the L and J hierarchies of constructible sets, where $J_\alpha = L_\alpha$.

[5] *This CIE2012 conference volume*

What else can these machines do? What is L_{ω^ω} good for?

• Note: This is the first level beyond ω of the L and J hierarchies of constructible sets, where $J_\alpha = L_\alpha$.

• Example: *The isomorphism problem for tree-automatic wellfounded trees (on $2^{<\omega}$) is $\Delta^0_{\omega^\omega}$ complete* (by Kartzow, Liu, Lohrey) [5]. As a corollary of weak universality, for any two such trees which have rank $< \omega^k$ say) there is a CIBSM-machine checking for an isomorphism.

[5] *This CIE2012 conference volume*

Broadening our viewpoint

- We can view the machines as being embodiments of more abstract *transfinite dynamical systems*.

Broadening our viewpoint

• We can view the machines as being embodiments of more abstract *transfinite dynamical systems*.

An Example
Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. *One may think of* f *acting on the points of an n-dimensional lattice torus where we identify '*∞*' (here* ω*) with 0. To let this act transfinitely we just require that at limit times* λ *the coordinates be the Liminf** *of their previous values.*

Broadening our viewpoint

• We can view the machines as being embodiments of more abstract *transfinite dynamical systems*.

An Example
Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. *One may think of* f *acting on the points of an n-dimensional lattice torus where we identify '*∞*' (here* ω*) with 0. To let this act transfinitely we just require that at limit times* λ *the coordinates be the Liminf** *of their previous values.*

• The existence of a Zero set (those points that eventually sink to the origin) for all such f's is a statement equivalent to Π^1_1-CA_0.

A rational variant

- Suppose now $f : (\mathbb{Q} \cap [0,1])^n \longrightarrow (\mathbb{Q} \cap [0,1])^n$ and consider iterates of f on this *rational torus*.

A rational variant

• Suppose now $f : (\mathbb{Q} \cap [0,1])^n \longrightarrow (\mathbb{Q} \cap [0,1])^n$ and consider iterates of f on this *rational torus*.

• This again can be coded by an oracle $Z \subseteq \mathbb{N}$ but now taking a coordinate-wise $Liminf^*$ operation can be problematic:

(i) The $Liminf$ values may not be rational;

(ii) $Liminf$, even when rational, is a Π_3 calculation.

A rational variant

• Suppose now $f : (\mathbb{Q} \cap [0,1])^n \longrightarrow (\mathbb{Q} \cap [0,1])^n$ and consider iterates of f on this *rational torus*.

• This again can be coded by an oracle $Z \subseteq \mathbb{N}$ but now taking a coordinate-wise $Liminf^*$ operation can be problematic:

(i) The *Liminf* values may not be rational;

(ii) *Liminf*, even when rational, is a Π_3 calculation.

Lemma

Iterates of f, if always defined until γ, where γ is the least $\Pi_3(f)$-reflecting admissible ordinal will be defined, (and be looping) for ever.

A rational variant

• Suppose now $f : (\mathbb{Q} \cap [0,1])^n \longrightarrow (\mathbb{Q} \cap [0,1])^n$ and consider iterates of f on this *rational torus*.

• This again can be coded by an oracle $Z \subseteq \mathbb{N}$ but now taking a coordinate-wise $Liminf^*$ operation can be problematic:

(i) The *Liminf* values may not be rational;

(ii) *Liminf*, even when rational, is a Π_3 calculation.

Lemma

Iterates of f, if always defined until γ, where γ is the least $\Pi_3(f)$-reflecting admissible ordinal will be defined, (and be looping) for ever.

• Q. Is this best possible?

The real thing

• Suppose now $f : (\mathbb{R} \cap [0,1])^n \longrightarrow (\mathbb{R} \cap [0,1])^n$ is continuous and consider iterates of f on this the *real torus*.

Now this kind of iteration requires reals to be carried along, *i.e.* sets of rationals; again the *Liminf** operation is a $\Pi_3(f)$ operation.

Definition

$\zeta \in On$ is f*-extendible* if $L_\zeta[f] \prec_{\Sigma_2} L_\Sigma[f]$ for some $\Sigma > \zeta$.
Let $\zeta(f)$ be the least such.

The real thing

• Suppose now $f : (\mathbb{R} \cap [0,1])^n \longrightarrow (\mathbb{R} \cap [0,1])^n$ is continuous and consider iterates of f on this the *real torus*.

Now this kind of iteration requires reals to be carried along, *i.e.* sets of rationals; again the *Liminf** operation is a $\Pi_3(f)$ operation.

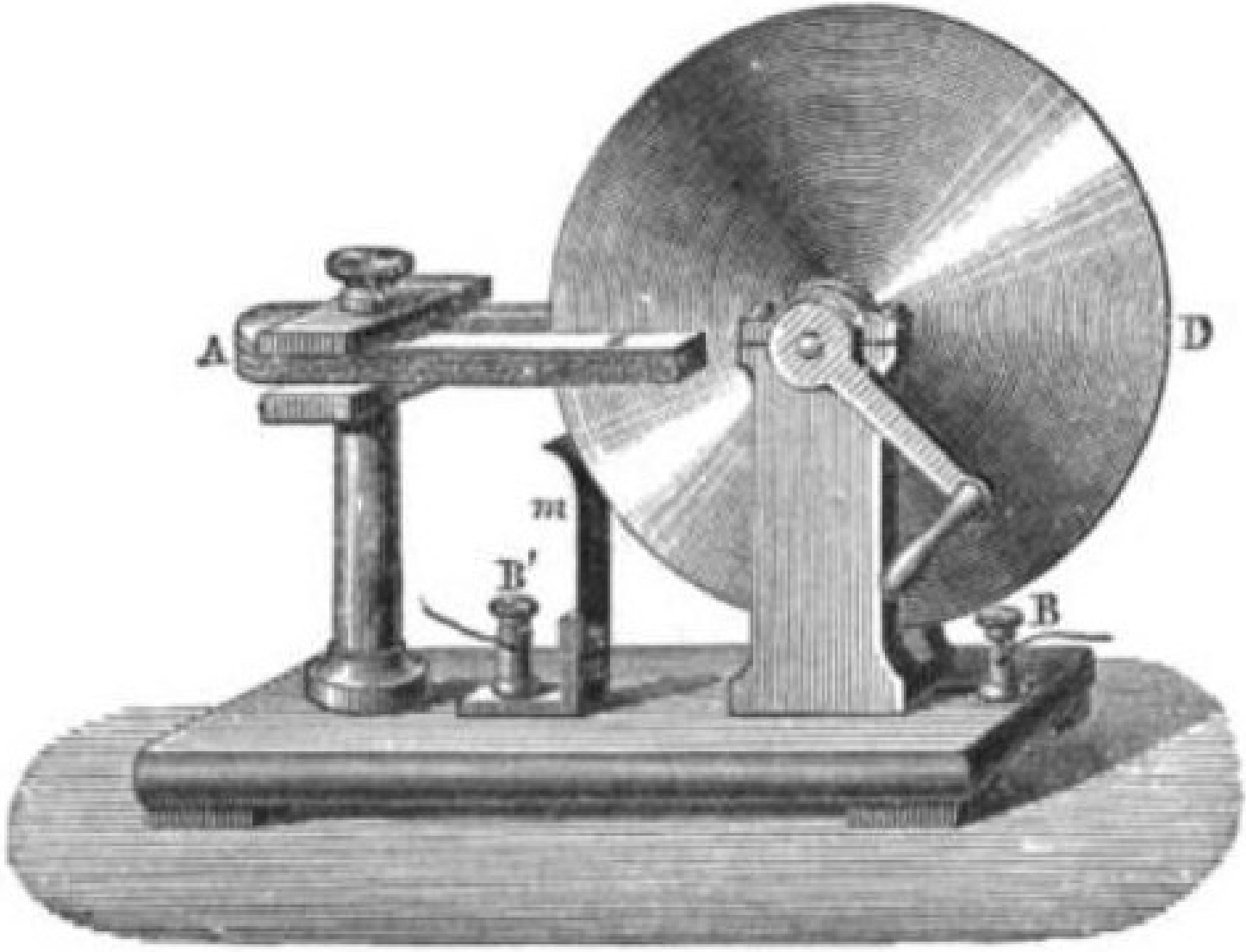
Definition

$\zeta \in On$ is *f-extendible* if $L_\zeta[f] \prec_{\Sigma_2} L_\Sigma[f]$ for some $\Sigma > \zeta$.
Let $\zeta(f)$ be the least such.

Then to determine whether, say iterations of f on starting real x go through $(0,0,\ldots,0)$ can be determined by an iteration of length $\zeta(f \oplus x)$.
But:

Lemma

$$\Pi^1_3\text{- CA}_0 \vdash \text{“ For every } f \textit{ an } f\textit{-extendible exists”}.$$


A
D
B'
B