A Generalised Dynamical System, Infinite Time Register Machines and Π^1_1-Comprehension *P. Koepke & P.D.Welch; Cie2011*

A Generalised Dynamical System

Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. We are going to consider transfinite iterations of such f as a *generalised dynamical system*. We may (and shall) shall think of f acting on the points of an n-dimensional lattice torus where we identify ∞ with 0. We set this up as follows. Given a point $r = (r_1, \ldots, r_n) \in \mathbb{N}^n$ we define iterates:

$$\begin{aligned} r^0 &= (r_1^0, \ldots, r_n^0) = (r_1, \ldots, r_n); \\ r^{\alpha+1} &= (r_1^{\alpha+1}, \ldots, r_n^{\alpha+1}) = f((r_1^\alpha, \ldots, r_n^\alpha)); \\ r^\lambda &= (r_1^\lambda, \ldots, r_n^\lambda) = (\mathrm{Liminf}^*_{\alpha\to\lambda}\, r_1^\alpha, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_2^\alpha, \ldots, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_n^\alpha) \end{aligned}$$

where we define:

$$\begin{aligned} r_i^\lambda = \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_i^\alpha &=_{df} \mathrm{Liminf}_{\alpha\to\lambda}\, r_i^\alpha \quad \text{if the latter is } < \omega \\ &=_{df} 0 \quad \text{otherwise.} \end{aligned}$$

A Generalised Dynamical System

Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. We are going to consider transfinite iterations of such f as a *generalised dynamical system*. We may (and shall) shall think of f acting on the points of an n-dimensional lattice torus where we identify ∞ with 0. We set this up as follows. Given a point $r = (r_1, \ldots, r_n) \in \mathbb{N}^n$ we define iterates:

$$\begin{aligned} r^0 &= (r_1^0, \ldots, r_n^0) = (r_1, \ldots, r_n); \\ r^{\alpha+1} &= (r_1^{\alpha+1}, \ldots, r_n^{\alpha+1}) = f((r_1^\alpha, \ldots, r_n^\alpha)); \\ r^\lambda &= (r_1^\lambda, \ldots, r_n^\lambda) = (\mathrm{Liminf}^*_{\alpha\to\lambda}\, r_1^\alpha, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_2^\alpha, \ldots, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_n^\alpha) \end{aligned}$$

where we define:

$$\begin{aligned} r_i^\lambda = \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_i^\alpha &=_{df} \mathrm{Liminf}_{\alpha\to\lambda}\, r_i^\alpha \quad \text{if the latter is } < \omega \\ &=_{df} 0 \quad \text{otherwise.} \end{aligned}$$

A Generalised Dynamical System

Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. We are going to consider transfinite iterations of such f as a *generalised dynamical system*. We may (and shall) shall think of f acting on the points of an n-dimensional lattice torus where we identify ∞ with 0. We set this up as follows. Given a point $r = (r_1, \ldots, r_n) \in \mathbb{N}^n$ we define iterates:

$$\begin{aligned} r^0 &= (r_1^0, \ldots, r_n^0) = (r_1, \ldots, r_n); \\ r^{\alpha+1} &= (r_1^{\alpha+1}, \ldots, r_n^{\alpha+1}) = f((r_1^\alpha, \ldots, r_n^\alpha)); \\ r^\lambda &= (r_1^\lambda, \ldots, r_n^\lambda) = (\mathrm{Liminf}^*_{\alpha\to\lambda}\, r_1^\alpha, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_2^\alpha, \ldots, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_n^\alpha) \end{aligned}$$

where we define:

$$\begin{aligned} r_i^\lambda = \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_i^\alpha &=_{df} \mathrm{Liminf}_{\alpha\to\lambda}\, r_i^\alpha \quad && \text{if the latter is } < \omega \\ &=_{df} 0 && \text{otherwise.} \end{aligned}$$

A Generalised Dynamical System

Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. We are going to consider transfinite iterations of such f as a *generalised dynamical system*. We may (and shall) shall think of f acting on the points of an n-dimensional lattice torus where we identify ∞ with 0. We set this up as follows. Given a point $r = (r_1, \ldots, r_n) \in \mathbb{N}^n$ we define iterates:

$$
\begin{aligned}
r^0 &= (r_1^0, \ldots, r_n^0) = (r_1, \ldots, r_n); \\
r^{\alpha+1} &= (r_1^{\alpha+1}, \ldots, r_n^{\alpha+1}) = f((r_1^\alpha, \ldots, r_n^\alpha)); \\
r^\lambda &= (r_1^\lambda, \ldots, r_n^\lambda) = (\mathrm{Liminf}^*_{\alpha\to\lambda}\, r_1^\alpha, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_2^\alpha, \ldots, \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_n^\alpha)
\end{aligned}
$$

where we define:

$$
\begin{aligned}
r_i^\lambda = \mathrm{Liminf}^*_{\alpha\to\lambda}\, r_i^\alpha &=_{df} \mathrm{Liminf}_{\alpha\to\lambda}\, r_i^\alpha \quad && \text{if the latter is } < \omega \\
&=_{df} 0 && \text{otherwise.}
\end{aligned}
$$

A Generalised Dynamical System

Suppose $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$. We are going to consider transfinite iterations of such f as a *generalised dynamical system*. We may (and shall) shall think of f acting on the points of an n-dimensional lattice torus where we identify ∞ with 0. We set this up as follows. Given a point $r = (r_1, \ldots, r_n) \in \mathbb{N}^n$ we define iterates:

$$\begin{aligned} r^0 &= (r_1^0, \ldots, r_n^0) = (r_1, \ldots, r_n); \\ r^{\alpha+1} &= (r_1^{\alpha+1}, \ldots, r_n^{\alpha+1}) = f((r_1^\alpha, \ldots, r_n^\alpha)); \\ r^\lambda &= (r_1^\lambda, \ldots, r_n^\lambda) = (\text{Liminf}^*_{\alpha\to\lambda}\, r_1^\alpha, \text{Liminf}^*_{\alpha\to\lambda}\, r_2^\alpha, \ldots, \text{Liminf}^*_{\alpha\to\lambda}\, r_n^\alpha) \end{aligned}$$

where we define:

$$\begin{aligned} r_i^\lambda = \text{Liminf}^*_{\alpha\to\lambda}\, r_i^\alpha &=_{df} \text{Liminf}_{\alpha\to\lambda}\, r_i^\alpha \quad \text{if the latter is} < \omega \\ &=_{df} 0 \quad \text{otherwise.} \end{aligned}$$

We may define a *zero set*:

$$Z_f =_{\mathrm{df}} \{p \in \mathbb{N}^n \mid \exists \alpha \quad p^\alpha = O\}.$$

Q: Given an arbitrary f, how hard is it to determine if $p \in Z_f$? Does the degree of difficulty in general depend on the dimension n?

As a more amusing example let $p = (p_0, p_1, p_2) \in (\mathbb{N}^n)^3$ be a triple of points on the n-dimensional lattice. In general they thus form a proper triangle. Then define the *triangle set*:

$$T_f = \{(p_0, p_1, p_2) \in (\mathbb{N}^n)^3 \mid \exists \alpha \;\; p_0^\alpha = p_1^\alpha = p_2^\alpha\}.$$

T_f is thus the set of triangles that eventually collapse to a point.

We may define a *zero set*:

$$Z_f =_{\text{df}} \{p \in \mathbb{N}^n \mid \exists\alpha \quad p^\alpha = O\}.$$

Q: Given an arbitrary f, how hard is it to determine if $p \in Z_f$? Does the degree of difficulty in general depend on the dimension n?

As a more amusing example let $p = (p_0, p_1, p_2) \in (\mathbb{N}^n)^3$ be a triple of points on the n-dimensional lattice. In general they thus form a proper triangle. Then define the *triangle set*:

$$T_f = \{(p_0, p_1, p_2) \in (\mathbb{N}^n)^3 \mid \exists\alpha \;\; p_0^\alpha = p_1^\alpha = p_2^\alpha\}.$$

T_f is thus the set of triangles that eventually collapse to a point.

We may define a *zero set*:

$$Z_f =_{\mathrm{df}} \{p \in \mathbb{N}^n \mid \exists \alpha \quad p^\alpha = O\}.$$

Q: Given an arbitrary f, how hard is it to determine if $p \in Z_f$? Does the degree of difficulty in general depend on the dimension n?

As a more amusing example let $p = (p_0, p_1, p_2) \in (\mathbb{N}^n)^3$ be a triple of points on the n-dimensional lattice. In general they thus form a proper triangle. Then define the *triangle set*:

$$T_f = \{(p_0, p_1, p_2) \in (\mathbb{N}^n)^3 \mid \exists \alpha \;\; p_0^\alpha = p_1^\alpha = p_2^\alpha\}.$$

T_f is thus the set of triangles that eventually collapse to a point.

We may define a *zero set*:

$$Z_f =_{\mathrm{df}} \{p \in \mathbb{N}^n \mid \exists\alpha \quad p^\alpha = O\}.$$

Q: Given an arbitrary f, how hard is it to determine if $p \in Z_f$? Does the degree of difficulty in general depend on the dimension n?

As a more amusing example let $p = (p_0, p_1, p_2) \in (\mathbb{N}^n)^3$ be a triple of points on the n-dimensional lattice. In general they thus form a proper triangle. Then define the *triangle set*:

$$T_f = \{(p_0, p_1, p_2) \in (\mathbb{N}^n)^3 \mid \exists\alpha \;\; p_0^\alpha = p_1^\alpha = p_2^\alpha\}.$$

T_f is thus the set of triangles that eventually collapse to a point.

The *existence* of such sets as Z_f, T_f needs a certain amount of analysis to establish.

Definition

(i) Let GDS^n_Z be the statement: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (Z_f \text{ exists})$."
(ii) Let GDS^n_T be: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (T_f \text{ exists})$."
(iii) $\mathrm{GDS}_T \Leftrightarrow \forall n\, \mathrm{GDS}^n_T$; similarly GDS_Z etc.

Question: Is it harder to establish GDS_T *than* GDS_Z *say?*

We have:

Theorem

$$\mathrm{ATR}_0 \vdash \mathrm{GDS}_Z \longleftrightarrow \mathrm{GDS}_T \longleftrightarrow \Pi^1_1\text{-}\mathrm{CA}_0.$$

(Here ATR_0 is the subsystem of second order number theory, *arithmetical transfinite recursion* and Π^1_1-CA_0 is the subsystem of Π^1_1 *comprehension*.)

The *existence* of such sets as Z_f, T_f needs a certain amount of analysis to establish.

Definition

(i) Let GDS^n_Z be the statement: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (Z_f$ exists)."
(ii) Let GDS^n_T be: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (T_f$ exists)."
(iii) $\mathrm{GDS}_T \Leftrightarrow \forall n\, \mathrm{GDS}^n_T$; similarly GDS_Z etc.

Question: Is it harder to establish GDS_T *than* GDS_Z *say?*

We have:

Theorem

$$\mathrm{ATR}_0 \vdash \mathrm{GDS}_Z \longleftrightarrow \mathrm{GDS}_T \longleftrightarrow \Pi^1_1\text{-}\mathrm{CA}_0.$$

(Here ATR_0 is the subsystem of second order number theory, *arithmetical transfinite recursion* and Π^1_1-CA_0 is the subsystem of Π^1_1 *comprehension*.)

The *existence* of such sets as Z_f, T_f needs a certain amount of analysis to establish.

Definition

(i) Let GDS^n_Z be the statement: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (Z_f$ exists)."
(ii) Let GDS^n_T be: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (T_f$ exists)."
(iii) $\mathrm{GDS}_T \Leftrightarrow \forall n\, \mathrm{GDS}^n_T$; similarly GDS_Z etc.

Question: Is it harder to establish GDS_T *than* GDS_Z *say?*

We have:

Theorem

$$\mathrm{ATR}_0 \vdash \mathrm{GDS}_Z \longleftrightarrow \mathrm{GDS}_T \longleftrightarrow \Pi^1_1\text{-}\mathrm{CA}_0.$$

(Here ATR_0 is the subsystem of second order number theory, *arithmetical transfinite recursion* and Π^1_1-CA_0 is the subsystem of Π^1_1 *comprehension*.)

The *existence* of such sets as Z_f, T_f needs a certain amount of analysis to establish.

Definition

(i) Let GDS^n_Z be the statement: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (Z_f$ exists)."
(ii) Let GDS^n_T be: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (T_f$ exists)."
(iii) $\mathrm{GDS}_T \Leftrightarrow \forall n\, \mathrm{GDS}^n_T$; similarly GDS_Z etc.

Question: Is it harder to establish GDS_T *than* GDS_Z *say?*

We have:

Theorem

$$\mathrm{ATR}_0 \vdash \mathrm{GDS}_Z \longleftrightarrow \mathrm{GDS}_T \longleftrightarrow \Pi^1_1\text{-}\mathrm{CA}_0.$$

(Here ATR_0 is the subsystem of second order number theory, *arithmetical transfinite recursion* and Π^1_1-CA_0 is the subsystem of Π^1_1 *comprehension*.)

The *existence* of such sets as Z_f, T_f needs a certain amount of analysis to establish.

Definition

(i) Let GDS^n_Z be the statement: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (Z_f$ exists)."
(ii) Let GDS^n_T be: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (T_f$ exists)."
(iii) $\mathrm{GDS}_T \Leftrightarrow \forall n\, \mathrm{GDS}^n_T$; similarly GDS_Z etc.

Question: Is it harder to establish GDS_T *than* GDS_Z *say?*

We have:

Theorem

$$\mathrm{ATR}_0 \vdash \mathrm{GDS}_Z \longleftrightarrow \mathrm{GDS}_T \longleftrightarrow \Pi^1_1\text{-}\mathrm{CA}_0.$$

(Here ATR_0 is the subsystem of second order number theory, *arithmetical transfinite recursion* and Π^1_1-CA_0 is the subsystem of Π^1_1 *comprehension*.)

The *existence* of such sets as Z_f, T_f needs a certain amount of analysis to establish.

Definition

(i) Let GDS^n_Z be the statement: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (Z_f$ exists)."
(ii) Let GDS^n_T be: "$\forall f : \mathbb{N}^n \longrightarrow \mathbb{N}^n (T_f$ exists)."
(iii) $\mathrm{GDS}_T \Leftrightarrow \forall n\, \mathrm{GDS}^n_T$; similarly GDS_Z etc.

Question: Is it harder to establish GDS_T *than* GDS_Z *say?*

We have:

Theorem

$$\mathrm{ATR}_0 \vdash \mathrm{GDS}_Z \longleftrightarrow \mathrm{GDS}_T \longleftrightarrow \Pi^1_1\text{-CA}_0.$$

(Here ATR_0 is the subsystem of second order number theory, *arithmetical transfinite recursion* and Π^1_1-CA_0 is the subsystem of Π^1_1 *comprehension*.)

Infinite Time Register Machines

Definition

Let N be a natural number. An *N-register machine* has:
(i) *registers* $R_0, R_1, \ldots, R_{N-1}$ which can hold natural numbers.
(ii) *register program*: a finite list $P = I_0, I_1, \ldots, I_{s-1}$ of *instructions* with *indices* $< s$, of the form $Z(n), T(m,n), S(n), J(m,n,p)$ and an *oracle query instruction* $O(n)$ where $m, n < N$;
(iii) At ordinal time τ the machine will be in a *configuration* consisting of a program state $I(\tau) < s$ of an instruction about to be performed and the register contents $\vec{R}(\tau) = \langle R_0(\tau), \ldots R_{N-1}(\tau)\rangle$.

At limit times λ we by fiat set:
(i) $I(\lambda)$ to be that I_k where $k = \liminf_{\sigma\to\lambda} I(\sigma)$;
(ii) $R_k(\lambda) = \liminf^*_{\sigma\to\lambda} R_k(\sigma)$.

Infinite Time Register Machines

Definition

Let N be a natural number. An *N-register machine* has:
(i) *registers* $R_0, R_1, \ldots, R_{N-1}$ which can hold natural numbers.
(ii) *register program*: a finite list $P = I_0, I_1, \ldots, I_{s-1}$ of *instructions* with *indices* $< s$, of the form $Z(n), T(m,n), S(n), J(m,n,p)$ and an *oracle query instruction* $O(n)$ where $m, n < N$;
(iii) At ordinal time τ the machine will be in a *configuration* consisting of a program state $I(\tau) < s$ of an instruction about to be performed and the register contents $\vec{R}(\tau) = \langle R_0(\tau), \ldots R_{N-1}(\tau)\rangle$.

At limit times λ we by fiat set:
(i) $I(\lambda)$ to be that I_k where $k = \text{liminf}_{\sigma\to\lambda} I(\sigma)$;
(ii) $R_k(\lambda) = \text{liminf}^*_{\sigma\to\lambda} R_k(\sigma)$.

ITRM Computable Functions

Definition

A partial function $F : \mathbb{N}^N \rightharpoonup \mathbb{N}$ is *computable (with oracle Z)* if there is some M-register program P such that for every N-tuple $(a_0, \ldots, a_{N-1}) \in \mathrm{dom}(F)$,

$$P^Z : (a_0, \ldots, a_{N-1}, 0, 0, \ldots, 0) \downarrow F(a_0, \ldots, a_{N-1}).$$

Definition

(i) (*n-register halting set*)

$$H_n =_{\mathrm{df}} \{\langle e, r_0, \ldots, r_{n-1}\rangle \mid P_e(r_0, \ldots, r_{n-1}) \downarrow\}$$

(ii) ITRM_n is the assertion:"The *n-register halting set* H_n exists."

(iii) **ITRM** is the similar relativized statement that

"For any $Z \subseteq \omega$, for any $n < \omega$ the *n-register halting set* H_n^Z exists."

ITRM Computable Functions

Definition

A partial function $F : \mathbb{N}^N \rightharpoonup \mathbb{N}$ is *computable (with oracle Z)* if there is some M-register program P such that for every N-tuple $(a_0, \ldots, a_{N-1}) \in \mathrm{dom}(F)$,

$$P^Z : (a_0, \ldots, a_{N-1}, 0, 0, \ldots, 0) \downarrow F(a_0, \ldots, a_{N-1}).$$

Definition

(i) (*n-register halting set*)

$$H_n =_{\mathrm{df}} \{\langle e, r_0, \ldots, r_{n-1}\rangle \mid P_e(r_0, \ldots, r_{n-1}) \downarrow\}$$

(ii) ITRM_n is the assertion:"The *n-register halting set* H_n exists."

(iii) **ITRM** is the similar relativized statement that

"For any $Z \subseteq \omega$, for any $n < \omega$ the *n-register halting set* H_n^Z exists."

The strengths of these statements

Theorem

$\mathrm{ATR}_0 \vdash \mathbf{ITRM} \longleftrightarrow \Pi^1_1\text{-CA}_0$.

Theorem

(i) ITRM_n *can be proven in* $\mathrm{KP} +$ *"there exist* $n+1$ *admissible ordinals."*

(ii) There is a fixed $k < \omega$ *so that for any* $n < \omega$

$$\mathrm{ATR}_0 + \mathrm{ITRM}_{n\cdot k} \vdash \text{"}\mathrm{HJ}(n, \varnothing) \textit{ exists.}\text{"}$$

The strengths of these statements

Theorem

$\mathrm{ATR}_0 \vdash \mathbf{ITRM} \longleftrightarrow \Pi^1_1\text{-}\mathrm{CA}_0$.

Theorem

(i) ITRM_n *can be proven in* $\mathrm{KP} +$ *"there exist* $n+1$ *admissible ordinals."*

(ii) There is a fixed $k < \omega$ *so that for any* $n < \omega$

$$\mathrm{ATR}_0 + \mathrm{ITRM}_{n \cdot k} \vdash \text{“}\mathrm{HJ}(n, \varnothing) \textit{ exists.”}$$

Proof of $GDS_T \leftrightarrow \Pi^1_1$-$CA_0$

Assume the last theorems.
$(\rightarrow)$: for any Z, we shall define a function $f : \mathbb{N}^m \longrightarrow \mathbb{N}^m$ so that $H^Z_n \leq_T T_f$ for some $m > n$. We use:

Fact H^Z_n is decidable by some n_0-register machine with program $P^Z_{e(n)}$: $e(n) \in \mathbb{N}$ may be effectively chosen so that

$$\langle e, r_0, \ldots, r_{n-1} \rangle \in H^Z_n \Leftrightarrow P^Z_{e(n)}(e, r_0, \ldots, r_{n-1}, 0, \ldots, 0) \downarrow 1.$$

We'll make some further harmless modifications to $P^Z_{e(n)}$; this will describe a a program $P^Z_{e(n)'}$: firstly, $P^Z_{e(n)'}$ uses a final extra register R_{n_0} which however is only addressed before the halting procedure in the following fashion: if $P^Z_{e(n)}(e, r_0, \ldots, r_{n-1}, 0, \ldots, i)$ is about to halt with a 1 in R_0, then the value of R_{n_0} is set to 0; otherwise it is untouched.

Define three starting points $p_i = p_i^0 \in \mathbb{N}^{n_0+1} (i < 3)$ with coordinates $(e, r_0, \ldots, r_{n-1}, 0, \ldots, i)$.
The machine $P^Z_{e(n)'}$ defines a function $f : \mathbb{N}^{n_0+1} \to \mathbb{N}^{n_0+1}$.
Check The p_i define a triangle which eventually collapses under f iff $(e, r_0, \ldots, r_{n-1}) \in H^Z_n$. Since GDS_T is a set, by ACA_0, so is H^Z_n.

$(\leftarrow)$: Assume Π^1_1-CA_0 and so **ITRM**. Take any $f : \mathbb{N}^n \to \mathbb{N}^n$; code this by a subset $Z \subseteq \mathbb{N}$. Write an algorithm $P^{Z_f}_e$ so that for any such Z_f it determines whether any input of a triangle's coordinates collapses. If this program uses n_0 registers then the set $H^{Z_f}_{n_0}$ guarantees for us that T_f exists.

Define three starting points $p_i = p_i^0 \in \mathbb{N}^{n_0+1} (i < 3)$ with coordinates $(e, r_0, \ldots, r_{n-1}, 0, \ldots, i)$.
The machine $P^Z_{e(n)'}$ defines a function $f : \mathbb{N}^{n_0+1} \to \mathbb{N}^{n_0+1}$.
Check The p_i define a triangle which eventually collapses under f iff $(e, r_0, \ldots, r_{n-1}) \in H^Z_n$. Since GDS_T is a set, by ACA_0, so is H^Z_n.

$(\leftarrow)$: Assume Π^1_1-CA_0 and so **ITRM**. Take any $f : \mathbb{N}^n \to \mathbb{N}^n$; code this by a subset $Z \subseteq \mathbb{N}$. Write an algorithm $P^{Z_f}_e$ so that for any such Z_f it determines whether any input of a triangle's coordinates collapses. If this program uses n_0 registers then the set $H^{Z_f}_{n_0}$ guarantees for us that T_f exists.

Why admissibles? A halting criterion

Lemma

Let $\langle I(\alpha), \vec{R}(\alpha) \rangle \mid \alpha < \theta)$ *be a computation of the n register machine with program P and with oracle Z for order type* θ *many stages. If there is some configuration* $(I', \vec{R'})$ *so that*

$$\mathrm{otp}(\{\beta < \theta \mid (I(\beta) = I' \wedge \vec{R}(\beta) = (I', \vec{R'})\}) \geq \omega^{\omega}$$

then the computation will never halt.

With two admissible ordinals

We sketch the argument that if a 1-register ITRM (!) with oracle $Z = \varnothing$, halts, then it does so before the second admissible ordinal.

• Recall $\alpha > \omega$ is *admissible* if $L_\alpha \models KP$.

• An admissible ordinal is Π_2-*reflecting*: that is if $\varphi(\vec{y})$ is any Π_2 formula in $\mathcal{L}_{\dot{\in}}$ with $\vec{y} \in L_\alpha$, then if $L_\alpha \models \varphi(\vec{y})$ then there is $\beta < \alpha$ with $L_\beta \models \varphi(\vec{y})$.

Suppose that the program does not halt on its starting input by stage $\eta =_{\rm df} \omega_1^{\rm CK}$. Let $\delta_0 < \eta$ be any ordinal. Then by fiat:

$$R_0(\eta) = \text{Liminf}^*_{\delta_0 < \beta \to \eta} R_0(\beta) = k \leq \omega$$

and

$$I(\eta) = \text{Liminf}_{\delta_0 < \beta \to \eta} I(\beta) = I_l \text{ where } l < m.$$

With two admissible ordinals

We sketch the argument that if a 1-register ITRM (!) with oracle $Z = \varnothing$, halts, then it does so before the second admissible ordinal.
• Recall $\alpha > \omega$ is *admissible* if $L_\alpha \models KP$.
• An admissible ordinal is Π_2-*reflecting*: that is if $\varphi(\vec{y})$ is any Π_2 formula in $\mathcal{L}_{\dot{\in}}$ with $\vec{y} \in L_\alpha$, then if $L_\alpha \models \varphi(\vec{y})$ then there is $\beta < \alpha$ with $L_\beta \models \varphi(\vec{y})$.

Suppose that the program does not halt on its starting input by stage $\eta =_{\mathrm{df}} \omega_1^{\mathrm{CK}}$. Let $\delta_0 < \eta$ be any ordinal. Then by fiat:

$$R_0(\eta) = \mathrm{Liminf}^*_{\delta_0 < \beta \to \eta} R_0(\beta) = k \leq \omega$$

and

$$I(\eta) = \mathrm{Liminf}_{\delta_0 < \beta \to \eta} I(\beta) = I_l \text{ where } l < m.$$

Case 1 : $\text{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = k < \omega \wedge \text{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
This is a $\Sigma_2 \wedge \Pi_2$ statement about δ_0. Hence by Π_2 reflection:

$$\exists\delta_1 \in (\delta_0,\eta) : \quad R_0(\delta_1) = R_0(\eta) = \text{Liminf}_{\delta_0<\beta\to\delta_1} R_0(\beta) = k \wedge I(\delta_1) = I_l.$$

Define by a Σ_1 recursion a sequence of ordinals δ_ι for $0 < \iota < \eta$ with $R_0(\delta_\iota) = R' \wedge I(\delta_\iota) = I'$, with the constellation $(I', R') = (I_l, k)$ ready to be used in Criterion. Hence the computation never halts.

Case 2: $\text{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = \omega \wedge \text{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
Now consider the computation between $[\eta, \eta+\eta]$: either it halts before the latter ordinal, or as in a Case 1, $\text{Liminf}_{\delta_0<\beta\to\eta+\eta} R_0(\beta) = \omega$, so we reset R_0 to zero. But now the $I_{\eta+\eta}$ might be different! But there are only finitely many of these, so by the second admissible ω_2^{ck} we'll be done.

For n-register machines this follows by an induction using the above idea.

Case 1 : $\text{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = k < \omega \wedge \text{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
This is a $\Sigma_2 \wedge \Pi_2$ statement about δ_0. Hence by Π_2 reflection:

$$\exists\delta_1 \in (\delta_0, \eta) : \quad R_0(\delta_1) = R_0(\eta) = \text{Liminf}_{\delta_0<\beta\to\delta_1} R_0(\beta) = k \wedge I(\delta_1) = I_l.$$

Define by a Σ_1 recursion a sequence of ordinals δ_ι for $0 < \iota < \eta$ with $R_0(\delta_\iota) = R' \wedge I(\delta_\iota) = I'$, with the constellation $(I', R') = (I_l, k)$ ready to be used in Criterion. Hence the computation never halts.

Case 2: $\text{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = \omega \wedge \text{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
Now consider the computation between $[\eta, \eta+\eta]$: either it halts before the latter ordinal, or as in a Case 1, $\text{Liminf}_{\delta_0<\beta\to\eta+\eta} R_0(\beta) = \omega$, so we reset R_0 to zero. But now the $I_{\eta+\eta}$ might be different! But there are only finitely many of these, so by the second admissible ω_2^{ck} we'll be done.

For n-register machines this follows by an induction using the above idea.

Case 1 : $\text{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = k < \omega \wedge \text{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
This is a $\Sigma_2 \wedge \Pi_2$ statement about δ_0. Hence by Π_2 reflection:

$$\exists\delta_1 \in (\delta_0, \eta) : \quad R_0(\delta_1) = R_0(\eta) = \text{Liminf}_{\delta_0<\beta\to\delta_1} R_0(\beta) = k \wedge I(\delta_1) = I_l.$$

Define by a Σ_1 recursion a sequence of ordinals δ_ι for $0 < \iota < \eta$ with $R_0(\delta_\iota) = R' \wedge I(\delta_\iota) = I'$, with the constellation $(I', R') = (I_l, k)$ ready to be used in Criterion. Hence the computation never halts.

Case 2: $\text{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = \omega \wedge \text{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
Now consider the computation between $[\eta, \eta+\eta]$: either it halts before the latter ordinal, or as in a Case 1, $\text{Liminf}_{\delta_0<\beta\to\eta+\eta} R_0(\beta) = \omega$, so we reset R_0 to zero. But now the $I_{\eta+\eta}$ might be different! But there are only finitely many of these, so by the second admissible ω_2^{ck} we'll be done.

For n-register machines this follows by an induction using the above idea.

Case 1 : $\mathrm{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = k < \omega \wedge \mathrm{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
This is a $\Sigma_2 \wedge \Pi_2$ statement about δ_0. Hence by Π_2 reflection:

$$\exists\delta_1 \in (\delta_0, \eta) : \quad R_0(\delta_1) = R_0(\eta) = \mathrm{Liminf}_{\delta_0<\beta\to\delta_1} R_0(\beta) = k \wedge I(\delta_1) = I_l.$$

Define by a Σ_1 recursion a sequence of ordinals δ_ι for $0 < \iota < \eta$ with $R_0(\delta_\iota) = R' \wedge I(\delta_\iota) = I'$, with the constellation $(I', R') = (I_l, k)$ ready to be used in Criterion. Hence the computation never halts.

Case 2: $\mathrm{Liminf}_{\delta_0<\beta\to\eta} R_0(\beta) = \omega \wedge \mathrm{Liminf}_{\delta_0<\beta\to\eta} I(\beta) = I_l$.
Now consider the computation between $[\eta, \eta+\eta]$: either it halts before the latter ordinal, or as in a Case 1, $\mathrm{Liminf}_{\delta_0<\beta\to\eta+\eta} R_0(\beta) = \omega$, so we reset R_0 to zero. But now the $I_{\eta+\eta}$ might be different! But there are only finitely many of these, so by the second admissible ω_2^{ck} we'll be done.

For n-register machines this follows by an induction using the above idea.

Generalization

• Consider iterations of *continuous* functions $f : [0, 2\pi)^n \to [0, 2\pi)$. With some starting point $p^0 = p$, such may contain many convergent subsequences, but consider the *liminf* of these as the point p^ω; continue.

• Q1. How long (in terms of p, f) does it take for the process to cycle?
• Q2. What is the collection S_f of "cycling" or "stable" points?

(These seem to go beyond Π^1_1-CA_0 and need Π_3-reflection rather than the Π_2-reflection of admissibility. For Q1 the answer is "$\leq$ the first $\Pi_3(p, f)$ reflecting ordinal.")

• Apply the above to your favourite machine model, or other process: e.g. Blum-Schub-Smale.

Generalization

• Consider iterations of *continuous* functions $f : [0, 2\pi)^n \to [0, 2\pi)$. With some starting point $p^0 = p$, such may contain many convergent subsequences, but consider the *liminf* of these as the point p^ω; continue.

• Q1. How long (in terms of p, f) does it take for the process to cycle?
• Q2. What is the collection S_f of "cycling" or "stable" points?

(These seem to go beyond Π^1_1-CA_0 and need Π_3-reflection rather than the Π_2-reflection of admissibility. For Q1 the answer is "$\leq$ the first $\Pi_3(p, f)$ reflecting ordinal.")

• Apply the above to your favourite machine model, or other process: e.g. Blum-Schub-Smale.