Effectively Computable Ordinal Functions

Peter Koepke, University of Bonn

EMU 2009, CUNY, New York, August 18, 2009

Contents

PRIMITIVE RECURSIVE SET (ORDINAL) FUNCTIONS (R. JENSEN and C. Karp, R. Gandy)

A function $F: V \longrightarrow V$ (F: Ord $\longrightarrow$ Ord) is a *primitive recursive set* (*ordinal*) *function* iff it is generated by the following scheme

- $P_{n,i}(\vec{x}) = x_i$, $1 \leqslant n \in \omega$, $\vec{x} = (x_1, \ldots, x_n)$, $1 \leqslant i \leqslant n$
- $F(x) = 0$
- $F(x, y) = x \cup \{y\}$ ($F(\alpha) = \alpha \cup \{\alpha\} = \alpha + 1$)
- $C(x, y, u, v) = x$ if $u \in v$, $= y$ otherwise

PRIMITIVE RECURSIVE SET (ORDINAL) FUNCTIONS

- $F(\vec{x}, \vec{y}) = G(\vec{x}, H(\vec{x}), \vec{y})$
- $F(\vec{x}, \vec{y}) = G(H(\vec{x}), \vec{y})$
- Recursion:

$$F(z, \vec{x}) = G(\bigcup \{F(u, \vec{x}) | u \in z\}, z, \vec{x})$$

Set recursion

$$F(z,\vec{x})=G(\bigcup\{F(u,\vec{x})|u\in z\},z,\vec{x})$$

allows course-of-value recursion:

$$\begin{aligned}F^*\restriction \mathrm{TC}(\{z\}) &= \bigcup\{F^*\restriction \mathrm{TC}(\{u\})|u\in z\}\cup\\ &\cup\{(z,G^*(\bigcup\{F^*\restriction \mathrm{TC}(\{u\})|u\in z\}))\}\end{aligned}$$

Ordinal recursion

$$\begin{aligned} F(\alpha, \vec{x}) &= G(\bigcup \{F(\beta, \vec{x}) | \beta \in \alpha\}, \alpha, \vec{x}) \\ &= G(\lim_{\beta < \alpha} F(\beta, \vec{x}), \alpha, \vec{x}) \end{aligned}$$

appears weaker: how can courses-of-values be coded into single ordinals?

R. Jensen and M. Schröder:

Theorem. Let $F\colon \mathrm{Ord} \to \mathrm{Ord}$. Then F is primitive ordinal recursive iff F is primitive set recursive.

The **Proof** uses the constructible hierarchy.

The constructible hierarchy (Gödel)

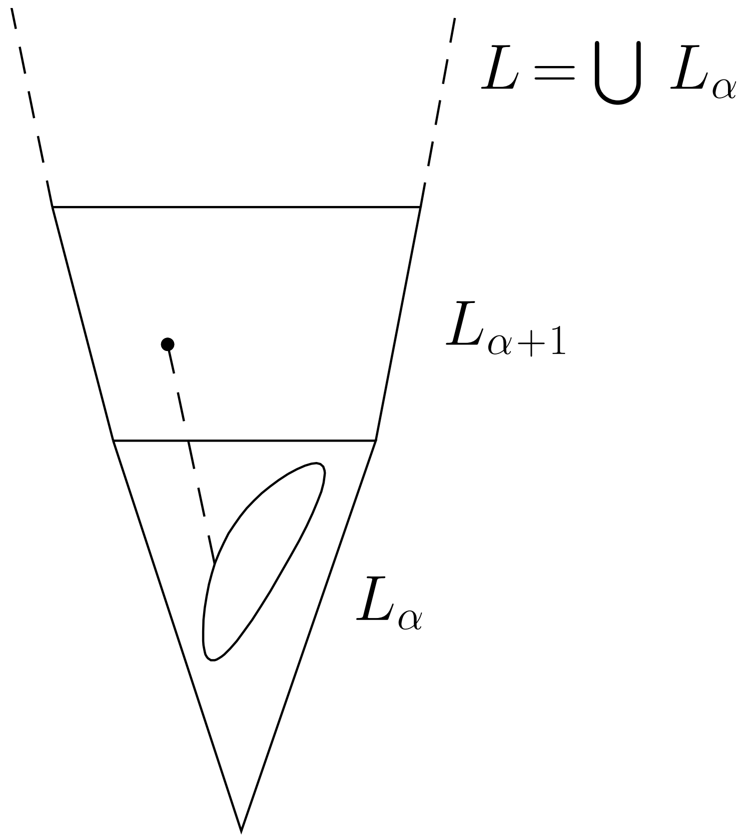
- $L_0 = \emptyset$

- $L_{\alpha+1} = \mathrm{Def}(L_\alpha) =$ the set of all subsets of L_α which are first-order definable in the structure $(L_\alpha, \in)$ from parameters

- $L_\lambda = \bigcup_{\alpha<\lambda} L_\alpha$, if λ is a limit ordinal

- $L = \bigcup_{\alpha\in\mathrm{Ord}} L_\alpha$ is the *constructible universe*

The constructible hierarchy



The constructible hierarchy

Every element of L is of the form

$$\begin{aligned} x_0 &= \{u_0 \in L_{\alpha_0} | L_{\alpha_0} \models \varphi_0(u_0, x_1, \ldots)\} \\ &= \{u_0 \in L_{\alpha_0} | L_{\alpha_0} \models \varphi_0(u_0, \{u_1 \in L_{\alpha_1} | L_{\alpha_1} \models \varphi_1(u_1, x_2, \ldots)\}, \ldots)\} \\ &= \{u_0 \in L_{\alpha_0} | L_{\alpha_0} \models \varphi_0(u_0, \{u_1 \in L_{\alpha_1} | L_{\alpha_1} \models \varphi_1(u_1, \{u_2 \in \\ &\quad L_{\alpha_2} | L_{\alpha_2} \models \varphi_2(u_2, x_3, \ldots)\}, \ldots)\}, \ldots)\} \\ &= \ldots \end{aligned}$$

and can be "named" by a <u>finite</u> sequence of ordinals like

$$\alpha_0, \varphi_0, \alpha_1, \varphi_1, \alpha_2, \varphi_2, \ldots$$

The constructible hierarchy

Finite sequences of ordinals can be coded by single ordinals due to Gödel pairing functions: there are primitive recursive ordinal functions G, G_1, G_2 such that

- $G\colon \mathrm{Ord} \times \mathrm{Ord} \leftrightarrow \mathrm{Ord}$
- $\forall\alpha\, G(G_1(\alpha), G_2(\alpha)) = \alpha$

The basic operations for the (coded) constructible universe are primitive recursive ordinal functions (Takeuti; Jensen, Schröder)

Recursive ordinal functions (Jensen, Karp)

A function $F\colon V \to V$ ($F\colon \mathrm{Ord} \to \mathrm{Ord}$) is a *set* (*ordinal*) *recursive function* iff it is generated by the above scheme together with the minimisation rule

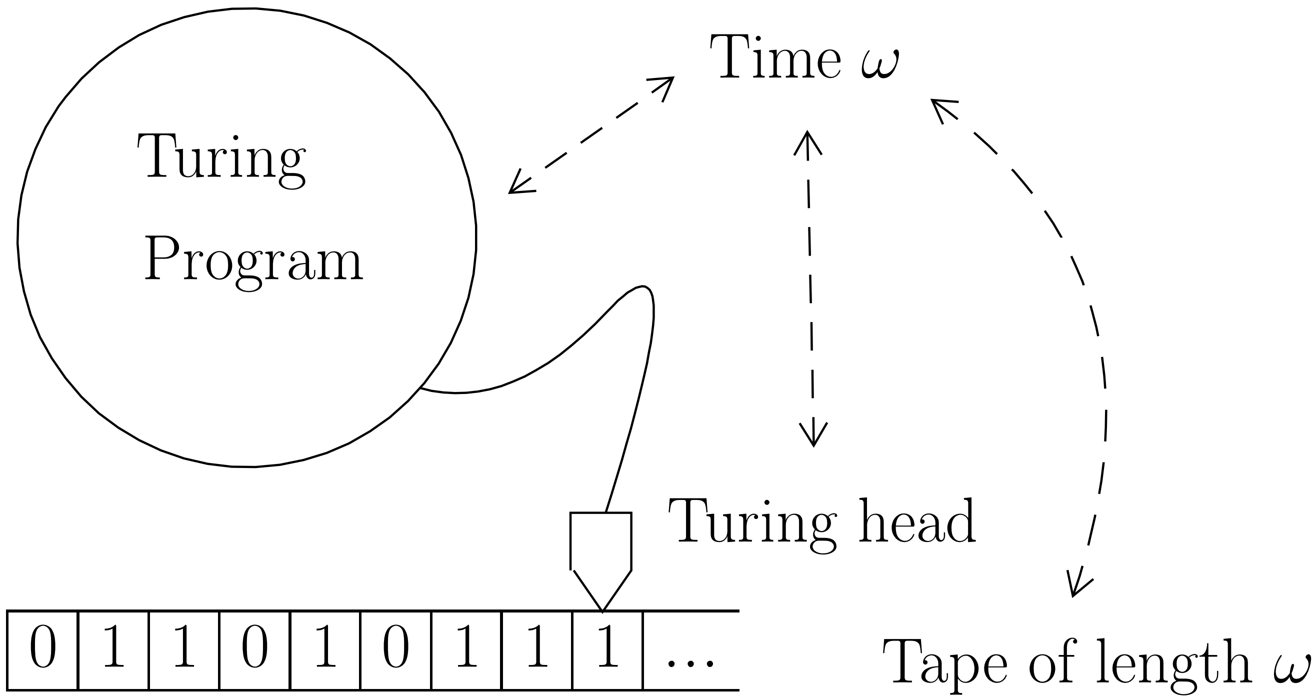
– $F(\vec{x}) = \min\{\xi \mid G(\xi, \vec{x}) = 0\}$, provided that $\forall \vec{x}\, \exists \xi\, G(\xi, \vec{x}) = 0$

Recursive ordinal functions

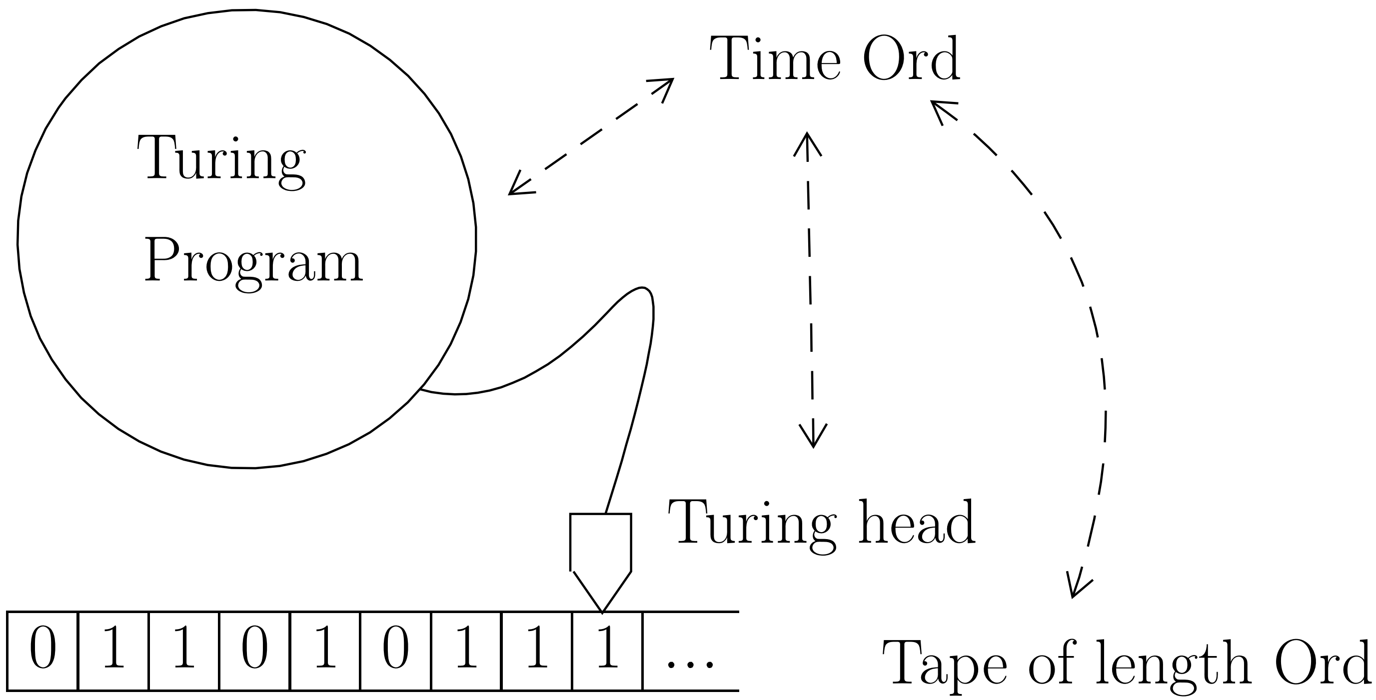
Theorem. For $f\colon \mathrm{Ord} \to \mathrm{Ord}$ the following are equivalent:

- f is ordinal recursive
- f is set recursive
- f is $\Delta_1(L)$

TURING machines



Ordinal TURING machines (OTMs)



Ordinal TURING machines (OTMs)

- successor steps of computations are determined by standard commands:

  ```
  m: if read=0 (or 1) then write 0 (or 1), go
  right (or left), and jump to instruction n
  ```

- limit steps λ are determined by liminf's:

  - $\text{command}(\lambda) = \liminf_{\alpha<\lambda} \text{command}(\alpha)$
  - $\text{head}(\lambda) = \liminf_{\alpha<\lambda} \text{head}(\alpha)$
  - $\text{cell}_\gamma(\lambda) = \liminf_{\alpha<\lambda} \text{cell}_\gamma(\alpha)$

OTM Computability $\leftrightarrow$ constructibility

Theorem (K) A set X of ordinals is OTM computable iff $X \in L$, i.e. if X is *constructible*.

Proof. $(\rightarrow)$ Any OTM computation can be carried out inside the model L, hence $X \in L$.

$(\leftarrow)$ The following OTM algorithm computes all constructible sets: assume that a structure (X, R) is written on the tape which is (pre-)isomorphic to $(L_\alpha, \in)$. Extend (X, R) to a structure (X', R') (pre-)isomorphic to $(L_{\alpha+1}, \in)$: for each $\in$-formula $\varphi(v_0, v_1, \ldots, v_m)$ and $x_1, \ldots, x_n \in X$ pick a new point $z \in X' \setminus X$ and for $x_0 \in X$ let

$$x_0 R' z \text{ iff } (X, R) \vDash \varphi[x_0, x_1, \ldots, x_m]$$

Every constructible set of ordinals occurs in the construction and is hence OTM computable.

Total functions $\mathrm{Ord} \rightarrow \mathrm{Ord}$

Theorem (K, B. Seyfferth) $f\colon \mathrm{Ord} \rightarrow \mathrm{Ord}$ is OTM computable iff f is $\Delta_1(L)$.

Proof. ($\rightarrow$) Let f be computable by the program P.

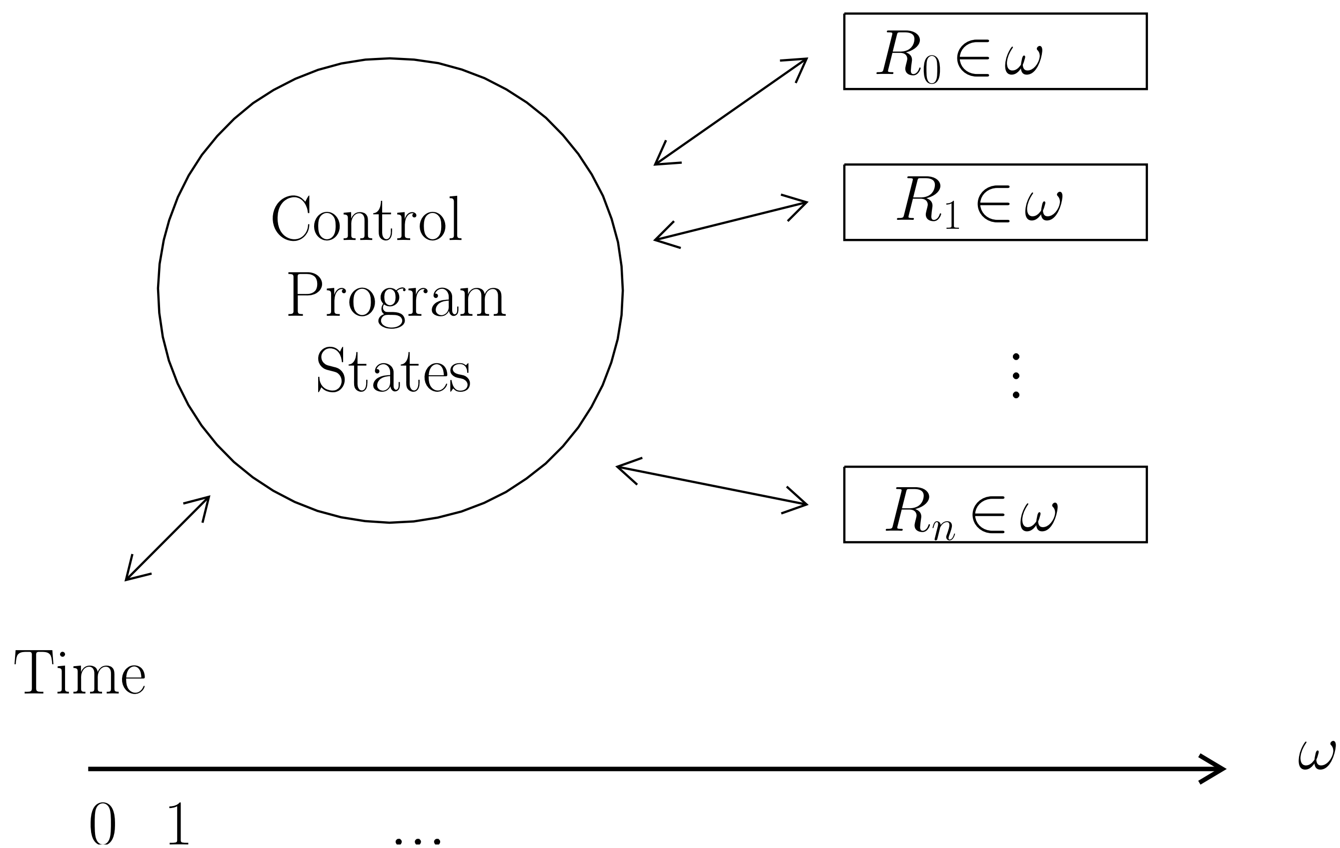
$f(\alpha) = \beta$ iff $\exists$computation C according to P with input α and output β

iff $\exists$computation $C \in L$ according to P with input α and output β

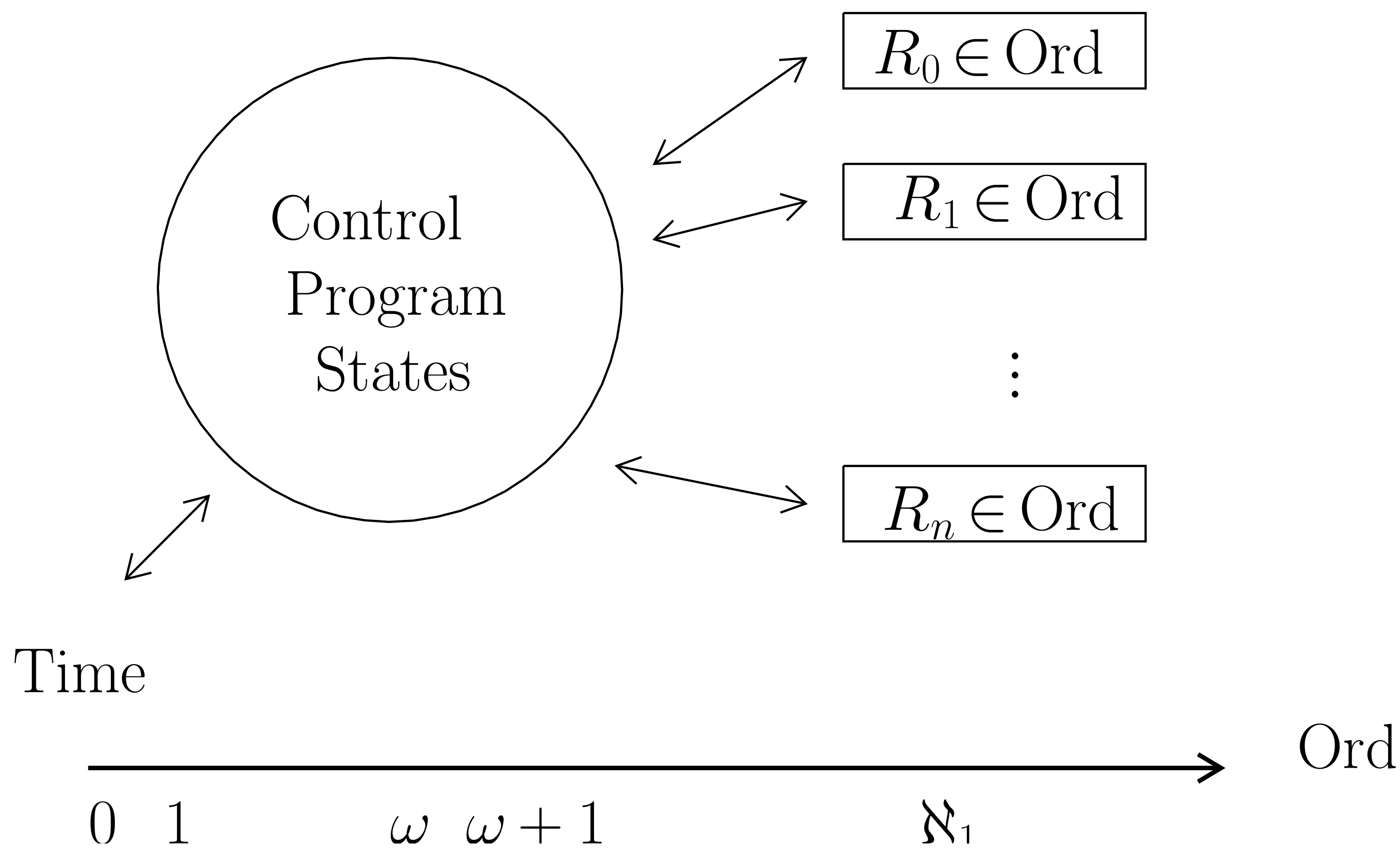
iff $\forall$computation $C \in L$(if C is according to P with input α then C outputs β

($\leftarrow$) Let f: Ord $\rightarrow$ Ord be defined in $(L, \in)$ by the Σ_1-formula $\varphi(x, y)$. Then compute $f(\alpha)$ as follows: enumerate L as described above. In the enumeration search for some structure (X, R) and $x, y \in X$ such that $(X, R) \vDash \varphi(x, y)$ and $\mathrm{otp}_R(x) = \alpha$, $\mathrm{otp}_R(y) = \beta$.

Register machines



Ordinal register machines, (ORMs)



ORM Computability $\leftrightarrow$ constructibility

Theorem (K, R. Siders) A set X of ordinals is ORM computable iff $X \in L$, i.e. if X is *constructible*.

Proof. ($\rightarrow$) Any ORM computation can be carried out inside the model L, hence $X \in L$.

ORM Computability $\leftrightarrow$ constructibility

($\leftarrow$) Since L has a canonical wellordering every point $x \in L$ can be "named" by a single ordinal α ; x is the "interpretation" $I(\alpha)$ of the name α . To compute Σ_0-properties of $I(\alpha)$ one suffices to compute Σ_0-properties of sets $I(\alpha')$ with $\alpha' < \alpha$. This amounts to a recursion which can be organised by a *stack*. Such stacks can be emulated by ORMs.

A recursion theorem

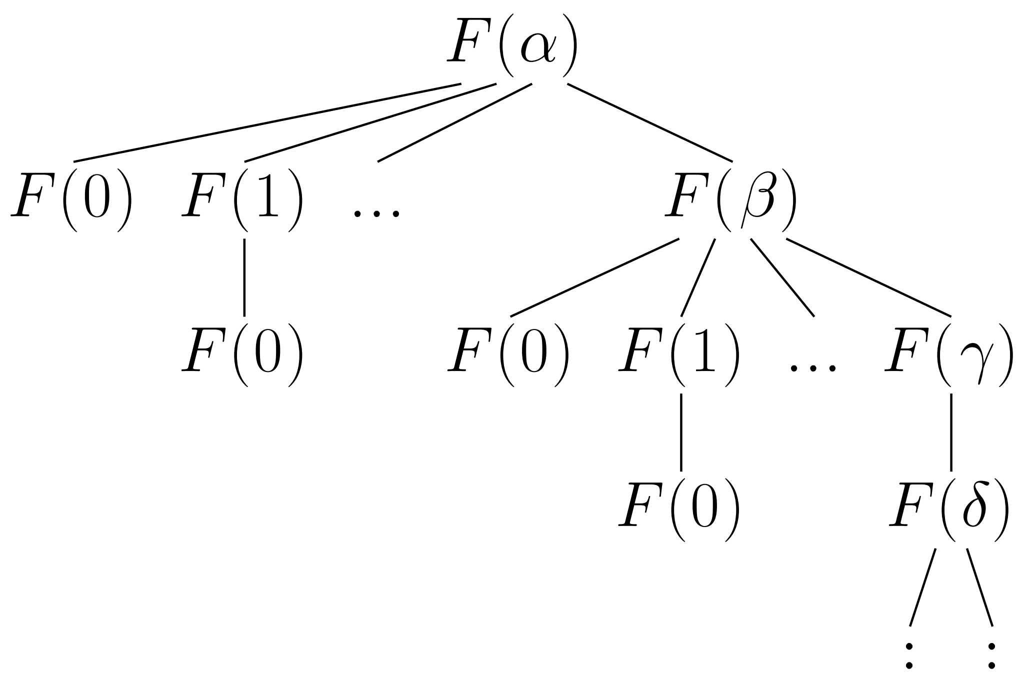
Let $H\colon \mathrm{Ord}^3 \to \mathrm{Ord}$ be ORM computable. Define

$$F(\alpha) = \begin{cases} 1 \text{ iff } \exists \nu < \alpha \; H(\alpha, \nu, F(\nu)) = 1 \\ 0 \text{ else} \end{cases}$$

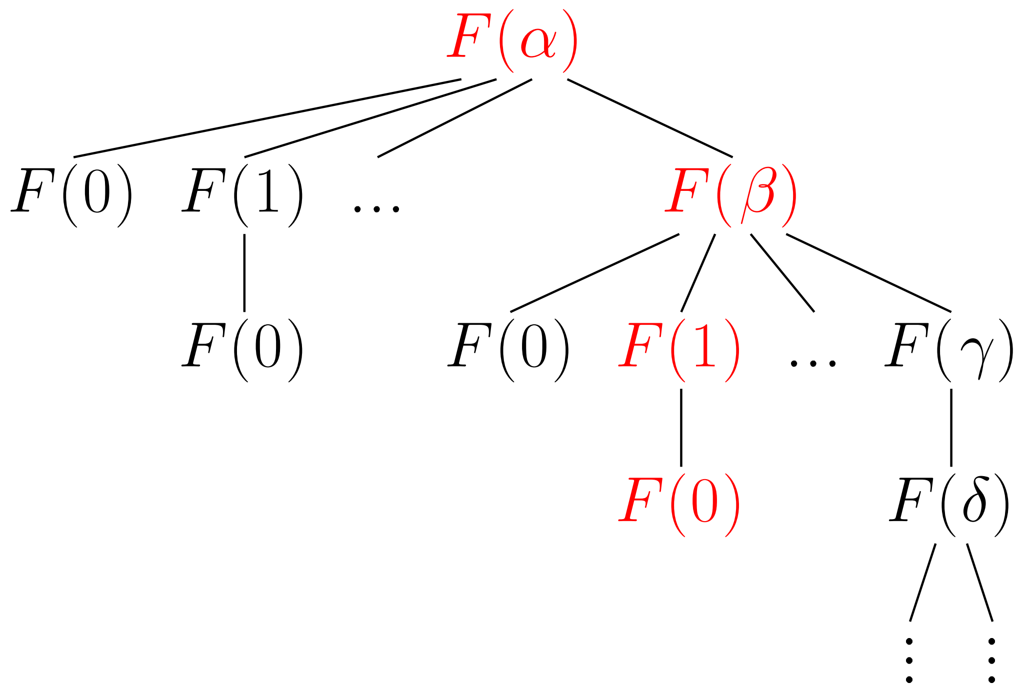
Then $F\colon \mathrm{Ord} \to \mathrm{Ord}$ is ORM computable.

A recursion theorem

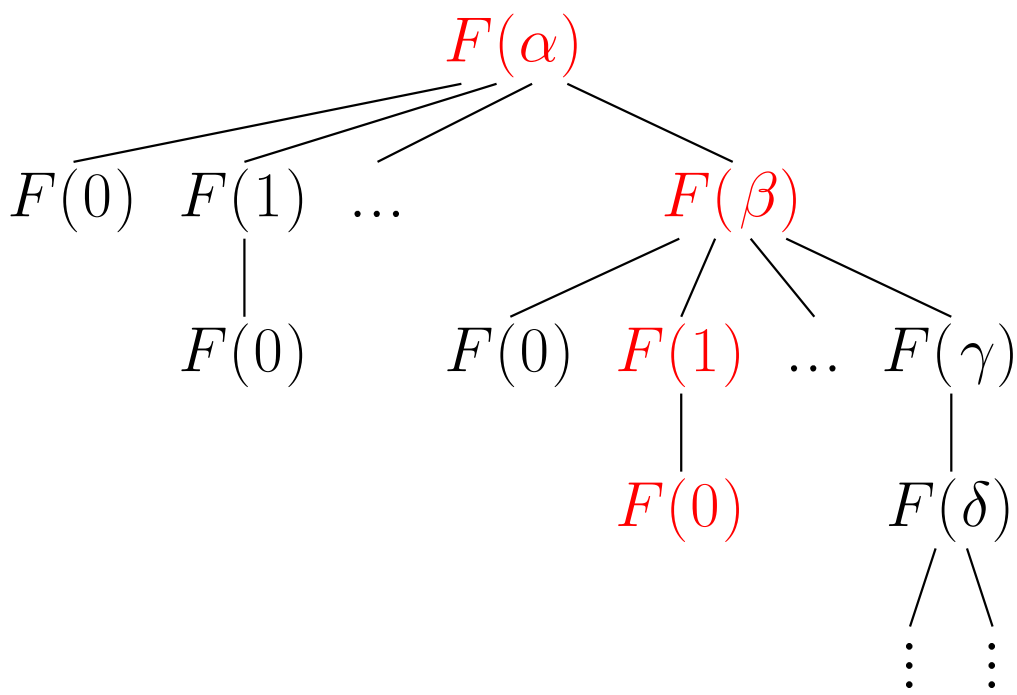
$$F(\alpha) = 1 \text{ iff } \exists \beta < \alpha \; H(\alpha, \beta, F(\beta)) = 1$$



A recursion theorem



A recursion theorem



Search for a good path using a stack $F(\alpha)?, F(\beta)?, F(\gamma)?, \ldots$

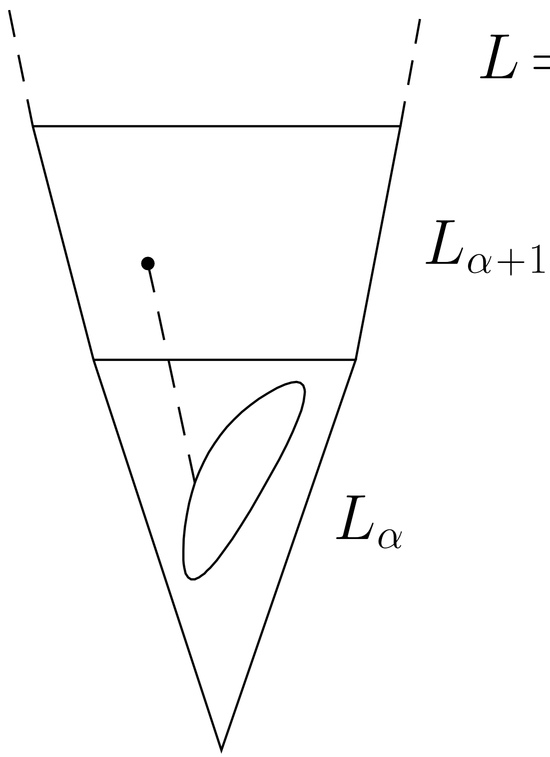
A recursion theorem

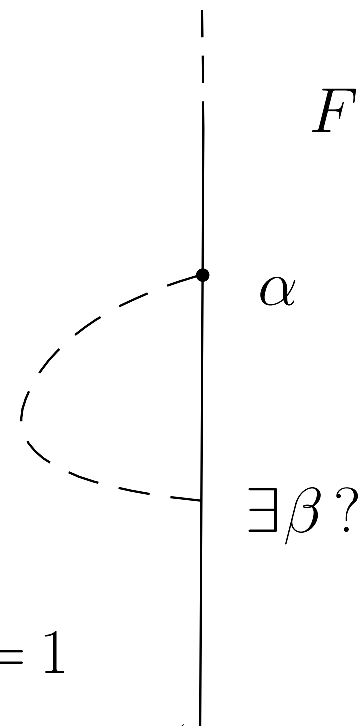
Code the stack $\alpha_0 > \alpha_1 > \ldots > \alpha_{n-1}$ into one register

$$R_m = 3^{\alpha_0} + 3^{\alpha_1} + \ldots + 3^{\alpha_{n-2}} + 3^{\alpha_{n-1}}.$$

The constructible model L

$$L = \bigcup L_\alpha$$





$F(\alpha) = 1$

iff $\exists\beta < \alpha \; H(\alpha, \beta, F(\beta)) = 1$

Total functions $\mathrm{Ord} \rightarrow \mathrm{Ord}$

Theorem (K) $f\colon \mathrm{Ord} \rightarrow \mathrm{Ord}$ is ORM computable iff f is $\Delta_1(L)$.

Proof. ($\rightarrow$) Let f be computable by the program P.

$f(\alpha) = \beta$ iff $\exists$computation C according to P with input α and output β

iff $\exists$computation $C \in L$ according to P with input α and output β

iff $\forall$computation $C \in L$(if C is according to P with input α then C outputs β

($\leftarrow$) Let f: Ord $\rightarrow$ Ord be defined in $(L, \in)$ by the formula $\exists z\psi(x, y, z)$ where ψ is Σ_0. Then compute $f(\alpha)$ as follows: compute a "name" $\dot{\alpha}$ for α ; search for ordinals $\dot{\beta}$ and $\dot{\gamma}$ such that $\psi(\alpha, I(\dot{\beta}), I(\dot{\gamma}))$; if such $\dot{\beta}$, $\dot{\gamma}$ are found, compute and output $\beta = I(\dot{\beta})$.

Theorem. For $f\colon \mathrm{Ord} \rightarrow \mathrm{Ord}$ the following are equivalent:

- f is recursive à la Jensen and Karp
- f is $\Delta_1(L)$
- f is OTM computable
- f is ORM computable

The Church-Turing thesis according to Odifreddi

For $f:\omega\rightarrow\omega$ the following are equivalent:

- f is recursive
- f is finitely definable
- f is Herbrand-Gödel computable
- f is representable in a consistent formal system extending $\mathcal{R}$

The Church-Turing thesis according to Odifreddi

For $f:\omega\to\omega$ the following are equivalent:

- f is recursive
- f is flowchart (or "while") computable
- f is λ-computable

Theorem. For $f\colon \mathrm{Ord} \to \mathrm{Ord}$ the following are equivalent:

- f is recursive à la Jensen and Karp
- f is $\Delta_1(L)$
- f is OTM computable
- f is ORM computable
- f is "while" computable on the ordinals
- f is computable by the methods of Kripke, Platek, Machover, Takeuti

Conclusion

There is a stable and well-characterised notion of effectively computable ordinal function.