Oracle modalities for higher dimensional types

Titouan Leclercq Andrew Swan

ENS de Lyon (student) University of Ljubljana

HoTT / UF 2025

The setting

We're doing synthetic computability in HoTT, so we have:

- $\prod$ and $\sum$ types
- higher inductive types
- universes
- univalence (not needed)

We also assume Markov's Principle in some proofs.

The setting

We're doing synthetic computability in HoTT, so we have:

- $\prod$ and $\sum$ types
- higher inductive types
- universes
- univalence (not needed)

We also assume Markov's Principle in some proofs.
Most of the work is formalized in Cubical Agda (soon to appear on `https://github.com/awswan/higher-computability`).

Modalities in HoTT

Modalities are rich, but we only focus on a few things:

- a modality is a function $\bigcirc : \mathcal{U} \to \mathcal{U}$ with a unit $\eta : (A : \mathcal{U}) \to A \to \bigcirc A$ and fancy stuff;

Modalities in HoTT

Modalities are rich, but we only focus on a few things:

- a modality is a function $\bigcirc : \mathcal{U} \to \mathcal{U}$ with a unit $\eta : (A : \mathcal{U}) \to A \to \bigcirc A$ and fancy stuff;
- a type A is $\bigcirc$-modal when η_A is an equivalence;
- $\mathcal{U}^{\bigcirc}$ is the type of all $\bigcirc$-modal types.

Modalities in HoTT

Modalities are rich, but we only focus on a few things:

- a modality is a function $\bigcirc : \mathcal{U} \to \mathcal{U}$ with a unit $\eta : (A : \mathcal{U}) \to A \to \bigcirc A$ and fancy stuff;
- a type A is $\bigcirc$-modal when η_A is an equivalence;
- $\mathcal{U}^{\bigcirc}$ is the type of all $\bigcirc$-modal types.

$\mathcal{U}^{\bigcirc}$ will be seen as a subuniverse of $\mathcal{U}$. When $\bigcirc$ is well behaved (and it will be), $\mathcal{U}^{\bigcirc}$ is also a model of HoTT / CTT.

Modalities in HoTT

Modalities are rich, but we only focus on a few things:

- a modality is a function $\bigcirc : \mathcal{U} \to \mathcal{U}$ with a unit $\eta : (A : \mathcal{U}) \to A \to \bigcirc A$ and fancy stuff;
- a type A is $\bigcirc$-modal when η_A is an equivalence;
- $\mathcal{U}^{\bigcirc}$ is the type of all $\bigcirc$-modal types.

$\mathcal{U}^{\bigcirc}$ will be seen as a subuniverse of $\mathcal{U}$. When $\bigcirc$ is well behaved (and it will be), $\mathcal{U}^{\bigcirc}$ is also a model of HoTT / CTT.
We need two main modalities:

- ∇ is such that $\mathcal{U}^{\nabla}$ is the classical world, where LEM is true
- given a family $B : A \to \mathrm{Prop}$, the nullification $\bigcirc_B$ is the smallest modality such that in $\mathcal{U}^{\bigcirc_B}$, every $B(a)$ is contractible.

Computability

There is a subset $\mathrm{Comp} \subseteq (\mathbb{N} \rightharpoonup \mathbb{N})$ of computable (partial) functions enumerated by a function $\varphi : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{N}$.
Given $f : A \rightharpoonup B$, $f(a)\uparrow$ when $f(a)$ is undefined, $f(a)\downarrow$ when $f(a)$ is defined.

Computability

There is a subset $\mathrm{Comp} \subseteq (\mathbb{N} \rightharpoonup \mathbb{N})$ of computable (partial) functions enumerated by a function $\varphi : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{N}$.
Given $f : A \rightharpoonup B$, $f(a)\uparrow$ when $f(a)$ is undefined, $f(a)\downarrow$ when $f(a)$ is defined.
Given a function $f : \mathbb{N} \to \mathbb{N}$, φ_e^f is the e-th computable function with oracle for f: it has access to f as a new primitive term.
Turing reduction

$$f \leq_T g \triangleq \exists e \in \mathbb{N}, f = \varphi_e^g$$

induces the Turing equivalence relation $\equiv_T$. Turing degrees are equivalent classes for $\equiv_T$.

Computability

There is a subset $\mathrm{Comp} \subseteq (\mathbb{N} \rightharpoonup \mathbb{N})$ of computable (partial) functions enumerated by a function $\varphi : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{N}$.
Given $f : A \rightharpoonup B$, $f(a) \uparrow$ when $f(a)$ is undefined, $f(a) \downarrow$ when $f(a)$ is defined.
Given a function $f : \mathbb{N} \to \mathbb{N}$, φ_e^f is the e-th computable function with oracle for f: it has access to f as a new primitive term.
Turing reduction

$$f \leq_T g \triangleq \exists e \in \mathbb{N}, f = \varphi_e^g$$

induces the Turing equivalence relation $\equiv_T$. Turing degrees are equivalent classes for $\equiv_T$.
A subset $X \subseteq \mathbb{N}$ is computable if $\chi_X = \varphi_e$ for some $e \in \mathbb{N}$. It is computably enumerable (c.e.) if it is the domain of a partial computable function (φ_e for some $e \in \mathbb{N}$). Example: $K \triangleq \{e \in \mathbb{N} \mid \varphi_e(e) \downarrow\}$.

Synthetic computability

Comp is useless: every function is computable. We need the (internal) Church Thesis instead:

$$\forall f : \mathbb{N} \rightharpoonup \mathbb{N}, \exists e : \mathbb{N}, f = \varphi_e$$

For a noncomputable function "$f : A \to B$", we consider $f : \nabla A \to \nabla B$, equivalently $f : A \to \nabla B$.

$$f(a) \downarrow \triangleq \sum_{b:B} f\ a = \eta_B\ b$$

Synthetic computability

Comp is useless: every function is computable. We need the (internal) Church Thesis instead:

$$\forall f : \mathbb{N} \rightharpoonup \mathbb{N}, \exists e : \mathbb{N}, f = \varphi_e$$

For a noncomputable function "$f : A \to B$", we consider $f : \nabla A \to \nabla B$, equivalently $f : A \to \nabla B$.

$$f(a) \downarrow \triangleq \sum_{b:B} f\ a = \eta_B\ b$$

A function φ_e^f is a function in "the world where f is total".
Consider the family $\lambda(a : A).(f\ a) \downarrow : A \to \mathrm{Type}$ and its nullification is the world of f-computable functions.
Works well with sets, but what about higher inductive types?

Seeing the oracle as counting

For sets, the most important case is a function $f : \mathbb{N} \to \mathbb{N}$.
f can be seen as counting the set $\mathrm{Fin}\ (f\ n)$: the family
$\lambda(n : \mathbb{N}).\mathrm{isFinSet}\ (g\ n)$ works, with $g\ n \triangleq \mathrm{Fin}\ (f\ n)$.
For higher dimension, we have a general notion: finite CW-complex.

CW-complex

A small modification over the usual definition of CW-complex to be closer to enumeration.
A CW-skeleton is a sequence of:

- spaces X_n, (option) integer k_n and maps $\alpha_n : \mathbb{S}^{k_n} \to X_n$ such that
- X_0 is empty
- for every n, we have a (option) pushout square

$$\begin{array}{ccc} \mathbb{S}^{k_n} & \xrightarrow{\alpha_n} & X_n \\ \downarrow & & \downarrow \\ 1 & \longrightarrow & X_{n+1} \end{array}$$

CW-complex

A small modification over the usual definition of CW-complex to be closer to enumeration.
A CW-skeleton is a sequence of:

- spaces X_n, (option) integer k_n and maps $\alpha_n : \mathbb{S}^{k_n} \to X_n$ such that
- X_0 is empty
- for every n, we have a (option) pushout square

$$
\begin{array}{ccc}
\mathbb{S}^{k_n} & \xrightarrow{\alpha_n} & X_n \\
\downarrow & & \downarrow \\
1 & \longrightarrow & X_{n+1}
\end{array}
$$

A CW-complex is (merely) the colimit of a CW-skeleton, a finite CW-complex is the same for a finite CW-skeleton.

First definition

We have the following situation:

$$\begin{array}{ccc} \mathrm{CWFin} & \rightarrowtail & \mathcal{U} \\ & & \uparrow \\ \mathrm{CWFin}^{\nabla} & \rightarrowtail & \mathcal{U}^{\nabla} \end{array}$$

A generalized oracle will be a family $B : A \to \mathrm{CWFin}^{\nabla}$, the associated modality is the nullification of

$$\lambda(a : A).\mathrm{isCWFin}\ (B\ a)$$

First definition

We have the following situation:

$$\begin{array}{ccc} \mathrm{CWFin} & \rightarrowtail & \mathcal{U} \\ & & \uparrow \\ \mathrm{CWFin}^{\nabla} & \rightarrowtail & \mathcal{U}^{\nabla} \end{array}$$

A generalized oracle will be a family $B : A \to \mathrm{CWFin}^{\nabla}$, the associated modality is the nullification of

$$\lambda(a : A).\mathrm{isCWFin}\ (B\ a)$$

Every function $f : A \to \nabla B$ with B a set can be seen as a generalized oracle, with the family $\lambda(a : A).(f\ a)\downarrow$. Knowing that $(f\ a)\downarrow$ is a finite CW means that it is decidable, hence inhabited. So, we indeed have a generalization of oracle modalities.

Motivation

$f : \mathbb{N} \to 2$ is c.e. $\iff$ there exists $g : \mathbb{N} \times \mathbb{N} \to 2$ non-decreasing on the second coordinate such that $f(n) = \max_m g(n, m)$. It can be seen as a grid we fill with elements through time:

$$\begin{array}{ccc}
g(0,0) & g(1,0) & \cdots \\
 & & \\
g(0,1) & g(1,1) & \cdots \\
 & & \\
\vdots & \vdots & \ddots \\
\downarrow & \downarrow & \\
f(0) & f(1) & \cdots
\end{array}$$

We do the same with pushouts for the generalization.

Definition

A family $B : A \to \mathrm{Type}$ is generalized c.e. (g.c.e.) when there is a grid $X_{a,m} : A \times \mathbb{N} \to \mathrm{Type}$ such that:

- $X_{a,0}$ is empty
- for each n and $a : A$, we have (option) $k : \mathbb{N}$, $\alpha : \mathbb{S}^k \to X_{a,n}$ and a pushout square

$$\begin{array}{ccc} \mathbb{S}^k & \xrightarrow{\alpha} & X_{a,n} \\ \downarrow & & \downarrow \\ 1 & \longrightarrow & X_{a,n+1} \end{array}$$

- for each $a : A$, $B(a) \simeq \operatorname*{colim}_{m \to \infty} X_{a,m}$

We call such a grid a c.e. grid.

Definition

A family $B : A \to \mathrm{Type}$ is generalized c.e. (g.c.e.) when there is a grid $X_{a,m} : A \times \mathbb{N} \to \mathrm{Type}$ such that:

- $X_{a,0}$ is empty
- for each n and $a : A$, we have (option) $k : \mathbb{N}$, $\alpha : \mathbb{S}^k \to X_{a,n}$ and a pushout square

$$\begin{array}{ccc} \mathbb{S}^k & \xrightarrow{\alpha} & X_{a,n} \\ \downarrow & & \downarrow \\ 1 & \longrightarrow & X_{a,n+1} \end{array}$$

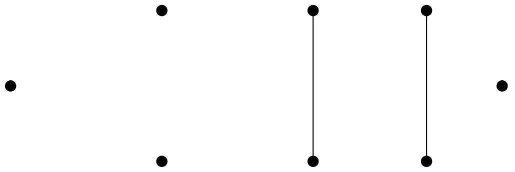
- for each $a : A$, $B(a) \simeq \operatorname*{colim}_{m \to \infty} X_{a,m}$

We call such a grid a c.e. grid.

If $B : A \to \mathrm{Prop}$, it is c.e. iff it is g.c.e.

The case of sets

The candidate for sets which are g.c.e: the K-c.e. sets.
X is K-c.e. $\iff$ there exists $g : \mathbb{N} \times \mathbb{N} \to 2$ such that
$f(n) = \liminf_m g(n, m)$
We can simulate it with unary and binary sets:



1 2 1 2 $\cdots$

The colimit in this case is the lim inf of the sequence of cardinals.

The case of groupoids

If the type is a groupoid, it can be seen as a group:

- ⋄ adding a path gives a constructor in the free group
- ⋄ adding a homotopy between path gives a new relation

Then the oracle of a group gives a finite presentation of it. As finitely presented group can be non computable, it seems hard to know what happens for the computational strength.

Conclusion

We have a generalization of oracle modalities for types of higher dimension.

Conclusion

We have a generalization of oracle modalities for types of higher dimension.
We gave a generalization of being computably enumerable (the usual c.e. propositions are still c.e.).

Conclusion

We have a generalization of oracle modalities for types of higher dimension.
We gave a generalization of being computably enumerable (the usual c.e. propositions are still c.e.).
It seems that going from propositions to sets gives a computational boost similar to having an oracle for K: do this happen for every dimension?
For groupoid, do we have the K'-c.e. sets? $(K' = \{e \in \mathbb{N} \mid \varphi_e^K(e)\downarrow\})$

Tank you!