Oracle modalities

Andrew W Swan

Carnegie Mellon University

May 23, 2023

What is an oracle?

Definition (Turing '36)

A partial function $\mathbb{N} \rightharpoondown \mathbb{N}$ is *computable* if it can be computed by a Turing machine (a computer program).

Key idea: We can encode computer programs as natural numbers. We write the partial function encoded by e as φ_e.

Theorem (Turing '36)

There is at least one non computable function.

Proof.

$$\kappa(n) := \begin{cases} 1 & \varphi_n(n) \downarrow = 0 \\ 0 & \text{otherwise} \end{cases}$$

□

Definition (Turing '39)

An *oracle Turing machine* is a computer program that can query information from an outside source (an *oracle*). We say a partial function $f : \mathbb{N} \rightharpoonup 2$ is *computable relative to* $\chi : \mathbb{N} \to 2$ if we can compute f using χ as an oracle.
We also say that f is *Turing reducible to* χ and write $f \leq_T \chi$.
Note that this defines a preorder on functions $\mathbb{N} \to 2$. We refer to the poset reflection of this preorder as the *Turing degrees*.

Example

A web browser can send queries (http requests) to a server and receive back information (webpages).

Queries can depend on the result of previous queries. E.g. a webbrowser can request all the images mentioned on a webpage that it just received.

Computability in topos theory

Theorem (Hyland '82)

The Turing degrees embed into the lattice of subtoposes of the effective topos, $\mathcal{E}ff$.

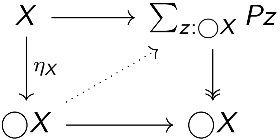
Key ideas:

- Set is a subtopos of $\mathcal{E}ff$. Write ∇ for the corresponding sheafification monad.
- For objects $X, Y \in \mathcal{E}ff$, every morphism $X \to Y$ is computable.
- Applying ∇ "erases computational information:" we can think of maps $\nabla X \to \nabla Y$ as non computable functions $X \to Y$.
- For a given $\chi : \nabla X \to \nabla Y$, we can consider the largest subtopos of $\mathcal{E}ff$ containing χ. The morphisms in the subtopos are *computable relative to* χ.

Modalities

Definition (Rijke, Shulman, Spitters)

A *uniquely eliminating modality* is an operation on types $\bigcirc : \mathcal{U} \to \mathcal{U}$ together with unit $\eta_X : X \to \bigcirc X$ for each $X : \mathcal{U}$ such that the canonical map $\prod_{z:\bigcirc X} \bigcirc(P(z)) \to \prod_{x:X} \bigcirc P(\eta_X(x))$ is an equivalence for $X : \mathcal{U}$ and $P : \bigcirc X \to \mathcal{U}$:



A type X is

- $\bigcirc$-*modal* if $\eta_X : X \to \bigcirc X$ is an equivalence.
- $\bigcirc$-*separated* if for all $x, y : X$, $x = y$ is $\bigcirc$-modal.
- $\bigcirc$-*connected* if $\bigcirc X$ is contractible.

Definition

Let $a : A \vdash B(a)$ be a family of types. A type X is *B-null* if for all $a : A$ the canonical map $X \to X^{B(a)}$ is an equivalence:

$$\begin{array}{ccc} B(a) & \longrightarrow & X \\ \downarrow & \overset{!}{\nearrow} & \\ 1 & & \end{array}$$

Theorem (Rijke, Shulman, Spitters)

There is a modality $\bigcirc_B$, defined as a higher inductive type, such that a type is $\bigcirc_B$-modal precisely if it is B-null.

Cubical Assemblies

Carrying out the construction of cubical sets internally in the category of assemblies, we can get a realizability model of HoTT:

Theorem (Uemura)

The category of cubical assemblies consists of cubical sets constructed internally in the lcc of assemblies. Cubical assemblies form a model of cubical type theory and thereby HoTT.

Theorem (S, Uemura)

Cubical assemblies have a reflective subuniverse that satisfies Church's thesis "all functions are computable."

Cubical Assemblies

Carrying out the construction of cubical sets internally in the category of assemblies, we can get a realizability model of HoTT:

Theorem (Uemura)

The category of cubical assemblies consists of cubical sets constructed internally in the lcc of assemblies. Cubical assemblies form a model of cubical type theory and thereby HoTT.

Theorem (S, Uemura)

Cubical assemblies have a reflective subuniverse that satisfies Church's thesis "all functions are computable."

We can use Church's thesis to give an example of a map $\chi : \mathbb{N} \to \nabla 2$ that does not extend to a map $\mathbb{N} \to 2$: the characteristic function of the halting set.

$\neg\neg$-sheafification

Definition

Write ∇ for $\neg\neg$-*sheafification*, i.e. nullification of all propositions P such that $\neg\neg P$ is true.

In general the existence of ∇ in cubical assemblies is problematic due to size issues. However, for many purposes we can use the 0-truncated version.

Theorem (S)

∇_0, the modality reflecting onto 0-truncated $\neg\neg$-sheaves, exists in cubical assemblies. Moreover, we can describe it explicitly: $\nabla_0 X$ is the discrete, uniform cubical assembly on the connected components of X.

Oracle modalities

Definition

For oracles $\bigcirc$ and $\bigcirc'$ we say $\bigcirc$ is *Turing reducible* to $\bigcirc'$ and write $\bigcirc \leq_T \bigcirc'$ if every $\bigcirc$-connected type is $\bigcirc'$-connected.

Definition

Given $\chi : A \to \nabla B$ the associated *oracle modality*, $\bigcirc_\chi$ is the nullification of $a : A \vdash \chi(a)\downarrow$.

Intuition:

- Any function in cubical sets appears as a map $\chi : A \to \nabla B$ in cubical assemblies.
- $\bigcirc$ is the smallest modality forcing χ to be a total function.

Since $\bigcirc_\chi$ is a special case of nullification, we can view it as a HIT. The point constructors are the same as I/O monads (free monads on polynomial endofunctors), but it also has path constructors:

- For $x : X$ we have $\eta(x) : \bigcirc_\chi X$. "Everything computable without the oracle is still computable with it."
- Given $a : A$ and $f : \chi(a)\downarrow \to \bigcirc_\chi X$, then $\mathrm{sup}(a, f) : \bigcirc_\chi X$. "If we can compute an element of $\bigcirc_\chi X$ by querying the oracle at a, then it is oracle computable."

Since $\bigcirc_\chi$ is a special case of nullification, we can view it as a HIT. The point constructors are the same as I/O monads (free monads on polynomial endofunctors), but it also has path constructors:

- For $x : X$ we have $\eta(x) : \bigcirc_\chi X$. "Everything computable without the oracle is still computable with it."
- Given $a : A$ and $f : \chi(a)\downarrow \to \bigcirc_\chi X$, then $\mathrm{sup}(a, f) : \bigcirc_\chi X$. "If we can compute an element of $\bigcirc_\chi X$ by querying the oracle at a, then it is oracle computable."
- If $z : \chi(a)\downarrow$ then $\mathrm{sup}(a, f) = f(z)$. "Only the final value of the computation matters: computing the same thing with the oracle and without the oracle are propositionally equal."

Since $\bigcirc_\chi$ is a special case of nullification, we can view it as a HIT. The point constructors are the same as I/O monads (free monads on polynomial endofunctors), but it also has path constructors:

- For $x : X$ we have $\eta(x) : \bigcirc_\chi X$. "Everything computable without the oracle is still computable with it."
- Given $a : A$ and $f : \chi(a)\downarrow \to \bigcirc_\chi X$, then $\mathrm{sup}(a, f) : \bigcirc_\chi X$. "If we can compute an element of $\bigcirc_\chi X$ by querying the oracle at a, then it is oracle computable."
- If $z : \chi(a)\downarrow$ then $\mathrm{sup}(a, f) = f(z)$. "Only the final value of the computation matters: computing the same thing with the oracle and without the oracle are propositionally equal."
- The above two constructors also apply to paths. "If we can compute a path using an oracle query to a, then the path belongs to $\bigcirc_\chi$."

Dimension shift

Theorem (Christensen, Opie, Rijke, Scoccola)

For any modality $\bigcirc$ *there is a modality* $\bigcirc^{=}$ *such that a type* X *is* $\bigcirc^{=}$*-modal iff it is* $\bigcirc$*-separated. For nullification of* $a : A \vdash B(a)$ *we can describe it explicitly as nullification of the pointwise suspension* $a : A \vdash \Sigma B(a)$*.*

Dimension shift

Theorem (Christensen, Opie, Rijke, Scoccola)

For any modality $\bigcirc$ there is a modality $\bigcirc^{=}$ such that a type X is $\bigcirc^{=}$-modal iff it is $\bigcirc$-separated. For nullification of $a : A \vdash B(a)$ we can describe it explicitly as nullification of the pointwise suspension $a : A \vdash \Sigma B(a)$.

Intuition: We can use an oracle χ to construct paths in $\bigcirc^{=}_{\chi} X$ but not points. More formally,

Observation

Any $\neg\neg$-separated set is $\bigcirc^{=}_{\chi}$-separated.

Any map $\mathbb{N} \to \bigcirc^{=}_{\chi}\mathbb{N}$ is computable, but e.g. there can be a map $\mathbb{N} \to \bigcirc^{=}_{\chi}\mathbb{S}^1$ equal to `loop` if $\varphi_e e\downarrow$ and otherwise `refl`. In fact,

$$\pi_1(\bigcirc^{=}_{\chi}\mathbb{S}^1) = \bigcirc_{\chi}\mathbb{Z} \neq \bigcirc^{=}_{\chi}\mathbb{Z} = \mathbb{Z}$$

Observation

Suppose that $\bigcirc_\beta \leq_T \bigcirc_\alpha^=$. Then β is computable.

Proof.

The generators of $\bigcirc_\beta$ are -1-truncated. Hence they are $\bigcirc_\alpha^=$-modal. By the assumption $\bigcirc_\beta \leq_T \bigcirc_\alpha^=$ they are also $\bigcirc_\alpha$-connected. Hence they are contractible, which can only happen when β is already computable. □

Q: What happens in higher dimensions, when we can no longer assume the generators are n-truncated?

- The Turing degrees and the homotopy groups of spheres are both well studied objects with rich mathematical structure.
- There are very simple examples of modalities in cubical assemblies that inherit characterisitics from both structures.
- We leave it for future work to find more interesting examples of interaction between computability theory and homotopy theory.

Some related results have been formalised in cubical Agda `https://github.com/awswan/oraclemodality`.

Thanks for your attention!