Turing Categories and Computability[1]

Robin Cockett

Department of Computer Science
University of Calgary
Alberta, Canada

robin@cpsc.ucalgary.ca

Estonia, March 2010

[1] Joint work with Pieter Hofstra

TURING CATEGORIES

Turing categories

Reducibility

Partial combinatory algebras

Recursion categories

TURING CATEGORIES

$\mathbb{T}$ is a **Turing category** if

- It is a cartesian restriction category
- It has a **Turing object**, T:

$$\begin{tikzcd} T \times A \arrow[r, "\tau_{A,B}"] & B \\ X \times A \arrow[u, dotted, "k \times 1"] \arrow[ur, "f"'] & \end{tikzcd}$$

this an object T with for each A and B a **Turing morphism**, $\tau_{A,B}$, such that for each f there is a total k, called a **index** for f, making the diagram above commute.
In the special case when X is the terminal object $h : 1 \longrightarrow C$ is a total element and we say it is a **code** for f.

Note: none of this structure is canonical!
If the index is uniquely determined then we shall say the Turing category is **extensional** ... (very unusual!)

TURING CATEGORIES

Theorem

In a Turing category, with a Turing object T, every object A is a retract of T.

PROOF: Consider

$$\begin{tikzcd} T \times 1 \arrow[r, "\tau_{1,A}"] & A \\ A \times 1 \arrow[u, "m_A \times 1"] \arrow[ur, "\pi_1"'] & \end{tikzcd}$$

Then we have $A \lhd_{m_A}^{r_A} T$ where $r_A = \langle 1, ! \rangle \tau_{1,A}$. $\square$

In particular $1 \lhd T$ and $T \times T \lhd T$.

TURING STRUCTURE

Theorem

A cartesian restriction category is a Turing category if and only if there is an object T, of which every object is a retract, which has a Turing morphism $T \times T \xrightarrow{\tau_{T,T}} T$.

PROOF: The difficulty is to prove that if every object is a retract of T then having a Turing morphism $\bullet = \bullet^1 = \tau_{T,T}$ suffices. arbitrary objects A and B by assumption we have $A \triangleleft_{r_A}^{m_A} T$ and $B \triangleleft_{r_B}^{m_B} T$ so we may define:

$$
\begin{aligned}
& T \times A \xrightarrow{\tau_{A,B}} B \\
= \quad & T \times A \xrightarrow{1 \times m_A} T \times T \xrightarrow{\bullet} T \xrightarrow{r_b} B
\end{aligned}
$$

Clearly this is a Turing morphism. $\square$

EXAMPLES OF TURING CATEGORIES I

Comp($\mathbb{N}$) the classical category of partial recursive functions:

Objects: 0,1,2, ... the natural numbers.

Maps: $f : n \longrightarrow m$ a partial recursive maps $f : \mathbb{N}^n \longrightarrow N^m$.

Turing object: $1 (= \mathbb{N})$ with Turing map "Kleene application" $\bullet : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}; (n, m) \mapsto \phi_n(m)$: the n^{th} Turing machine run on input m. Note $\mathbb{N} \equiv \mathbb{N} \times \mathbb{N}$ and $1 = \mathbb{N}^0 \lhd \mathbb{N}$.

Note this category is definitely partial (for example it has zero maps).

TURING STRUCTURE

Once one has fixed a Turing map the Turing structure is not unique.

Here is an alternate way to get a Turing structure:
Define $\bullet^{n+1}$ for $n > 1$ by setting $\bullet = \bullet^1$ and defining it inductively using $\bullet^n$ as $(\bullet \times 1)\bullet^n$:

$$\begin{tikzcd}
T \times T \times T^n \arrow[r, "\bullet\times 1"] & T \times T^n \arrow[r, "\bullet^n"] & T \\
X \times T \times T^n \arrow[u, dotted, "(f^{\bullet^n})^{\bullet}\times 1"] \arrow[ur, dotted, "f^{\bullet^n}\times 1"] \arrow[urr, "f"'] & &
\end{tikzcd}$$

This provides $f^{\bullet^{n+1}} = (f^{\bullet^n})^{\bullet}$ showing $\bullet^n$ can be used as $\tau_{T^n,T}$.

TURING STRUCTURE

But what about $\circ = \bullet^0$?

$$\begin{tikzcd} T \arrow[r, "\circ"] & T \\ X \arrow[u, dotted, "f^\circ"] \arrow[ur, "f"'] & \end{tikzcd}$$

Set this to $\circ = T \xrightarrow[\Delta]{} T \times T \xrightarrow[\bullet]{} T$. Now we have

$$\begin{tikzcd} T \arrow[r, "\Delta"] & T \times T \arrow[r, "\bullet"] & T \\ & X \times T \arrow[u, dotted, "(\pi_0 f)^\bullet \times 1"] \arrow[ur, "\pi_0 f"'] & \\ X \arrow[uu, dotted, "(\pi_0 f)^\bullet"] \arrow[ur, "\langle 1, (\pi_0 f)^\bullet \rangle"] \arrow[uurr, bend right, "f"'] & & \end{tikzcd}$$

TURING STRUCTURE

Finally for an arbitrary product of objects $A_1 \times ... \times A_n$ by assumption we have $A_i \lhd_{r_{A_i}}^{m_{A_i}} T$ so we may define:

$$\begin{aligned} T \times A_1 \times ... \times A_n &\xrightarrow{\tau_{A_1 \times . \times A_n, B}} B \\ = \quad T \times A_1 \times ... \times A_n &\xrightarrow{1 \times m_{A_1} \times ... \times m_{A_n}} T \times T^n \xrightarrow{\bullet^n} T \xrightarrow{r_b} B \end{aligned}$$

Clearly this is a Turing morphism.

Thus even given a Turing map there are lots of ways to obtain a Turing structure! This way, however, suggests the following example ...

EXAMPLES OF TURING CATEGORIES II

$\lambda - $ comp the category generated by the λ-calculus (with β-equality):

Objects: $0, 1, 2, \ldots$ the natural numbers

Maps: $f : n \longrightarrow m$ is a tuple of m maps from $f_i : n \longrightarrow 1$ where such a map is a λ-calculus term in variables $x_1, \ldots, x_n$ (with equality given by β-reduction).

Composition: Substitution.

Turing object: 1 with Turing map $\bullet : 2 \longrightarrow 1; (x_1, x_2) \mapsto x_1 x_2$.

This is a total category ... but who said Turing categories could not be total!!

EXAMPLES OF TURING CATEGORIES II (cont.)

$\lambda -$ comp: the Turing structure ...

$$\begin{tikzcd}
1 \times n (= n+1) \arrow[r, "\bullet^{(n)}"] & 1 \\
p \times n (= p+n) \arrow[u, dotted, "h \times 1"] \arrow[ur, "f"'] &
\end{tikzcd}$$

$$\begin{array}{rcl}
\bullet^{(n)}(x_0, x_1, ..., x_n) & = & x_0 x_1 ... x_n \\
h(y_1, ..y_p, x_0, .., x_n) & = & \lambda x_0 x_1 ... x_n . f(y_1, ..y_p, x_0, .., x_n)
\end{array}$$

β-reduction ensures the diagram commutes.

EXAMPLES OF TURING CATEGORIES III

$p\lambda-$comp the category of β-normal λ-terms. We let Λ be the set of closed λ-terms in β-normal form:

Objects: $0, 1, 2, \ldots$ the natural numbers: n is the set Λ^n

Maps: $f : n \longrightarrow m$ is a tuple of m maps from $f_i : n \longrightarrow 1$ where such a map is determined by a λ-calculus term N in variables $x_1, \ldots, x_n$ which is in β-normal form:

$$\Lambda^n \longrightarrow \Lambda : (M_1, \ldots, M_n) \mapsto \begin{cases} N[M_i/x_i] \downarrow_\beta \\ \uparrow \end{cases}$$

where $N[M_i/x_i] \downarrow_\beta$ is the strong normal form of the substituted term – which may not always exist.

Composition: As for partial maps.

Turing object: 1 with Turing maps $\bullet : 2 \longrightarrow 1; (x_1, x_2) \mapsto x_1 x_2$.

This is a partial Turing category!

REDUCIBILITY

In any restriction category say that a restriction idempotent $e' : X \longrightarrow X$ (many-one) **reduces** to $e : Y \longrightarrow Y$, write $e' \leq_m e$, if there is a total map $f : X \longrightarrow Y$ so that $\overline{fe} = e'$.

Say that e' 1-reduces to e, $e' \leq_1 e$ if there is a *monic* f with $\overline{fe} = e'$.

Say that $e : X$ is **m-complete** in case every e' m-reduces to e, that is $e' \leq_m e$. Similarly e is **1-complete** is every e' 1-reduces to e.

NOTE: this is the standard definition: think of $e = \overline{e} : Y \longrightarrow Y$ as a predicate.

REDUCIBILITY (cont.)

Recall $K = \overline{\circ} = \overline{\Delta \bullet}$ – intuitively those computations which terminate on their own codes, we always have:

Theorem

In any Turing category $K = \overline{\circ}$ is m-complete.

PROOF: Suppose $e : X$ then

$$\begin{tikzcd} T \arrow[r, "\circ"] & T \\ X \arrow[u, dotted, "(em_X)^\circ"] \arrow[r, "e"'] & X \arrow[u, tail, "m_X"'] \end{tikzcd}$$

and

$$\begin{aligned} \overline{(em_X)^\circ K} &= \overline{(em_X)^\circ \overline{\circ}} \\ &= \overline{(em_X)^\circ \circ} \\ &= \overline{em_X} = \overline{e} = e \end{aligned}$$

$\square$

REDUCIBILITY (cont.)

What does this mean for total Turing categories?

Can you prove that in a Turing category is total if and only if all predicates are m-complete?

1-REDUCIBILITY

There is no guarantee that f° is monic but if it was K would be 1-complete.

We will use “padding” to obtain an alternative Turing morphism which has this property. Modify the Turing morphism

$$\begin{tikzcd}
T \times T \arrow[r, "r_{T\times T}"'] \arrow[rrr, bend left, "\bullet'"] & (T \times T) \times T \arrow[r, "\pi_1 \times 1"] & T \times T \arrow[r, "\bullet"] & T \\
& X \times T \arrow[ul, dotted, "f^{\bullet'} \times 1"] \arrow[u, dotted, "\langle f^\bullet, m_X \rangle m_{T\times T} \times 1"] \arrow[ur, dotted, "f^\bullet \times 1"] \arrow[urr, "f"'] & &
\end{tikzcd}$$

define $f^{\bullet'} = \langle f^\bullet, m_X \rangle m_{T\times T}$ note that

$$f^{\bullet'} r_{T\times T} \pi_1 = m_X$$

so, in fact, this is a section so certainly $f^{\circ'}$ is monic!

1-REDUCIBILITY

Theorem

In any Turing category $K' = \overline{\circ'}$, as defined above, is 1-complete.

Note: this is *stronger* than 1-complete as the morphism along which the reduction is being obtained is a section.

This also illustrates the non-canonical nature of the Turing morphisms (doing this again gives an infinite family of Turing morphisms).

Note: special properties of $\bullet$ may not be preserved by moving to $\bullet'$. For example if $(T, \bullet)$ is extensional $(T, \bullet')$ will be extensional only when the Turing category is trivial!

PARTIAL COMBINATORY ALGEBRAS

In a cartesian restriction category a **partial combinatory algebra** is:

- an object A
- a partial map $\bullet : A \times A \longrightarrow A$
- two (total) points $1 \xrightarrow{\mathsf{k}} A$ and $1 \xrightarrow{\mathsf{s}} A$.

satisfying:

$$\begin{tikzcd}
A \times A \times A \arrow[r, "(\bullet \times 1)\bullet"] & A \\
A \times A \arrow[u, "\mathsf{k} \times 1 \times 1"] \arrow[ur, "\pi_0"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \times A \times A \times A \arrow[rr, "\bullet^3"] & & A \\
& & A \times A \arrow[u, "\bullet"'] \\
A \times A \times A \arrow[uu, "\mathsf{s} \times 1 \times 1 \times 1"] \arrow[rr, "\theta'_A"'] & & (A \times A) \times (A \times A) \arrow[u, "\bullet \times \bullet"']
\end{tikzcd}$$

and $A \times A \xrightarrow{\mathsf{s} \times 1 \times 1} A \times A \times A \xrightarrow{\bullet^2} A$ is total.

PARTIAL COMBINATORY ALGEBRAS

Equationally we have:

$$(\mathsf{k} \bullet x) \bullet y = x \qquad ((\mathsf{s} \bullet x) \bullet y) \bullet z = (x \bullet z) \bullet (y \bullet z) \qquad x|_{(\mathsf{s} \bullet v) \bullet w} = x$$

These are the usual equations from (total) combinatory algebra with the added requirement (expressed in the last equations) that sxy is total.

Theorem

If $(T, \bullet)$ is a Turing object in a cartesian restriction category then it is a partial combinatory algebra.

PROOF: Use the above commuting requirements to *define* k and s! $\square$

This begs the question: what is the connection between PCAs and Turing categories?

PARTIAL COMBINATORY ALGEBRAS

Given any cartesian restriction category there is a cartesian restriction functor

$$\Gamma : \mathbb{X} \longrightarrow \mathrm{Par} : A \mapsto \mathrm{points}(A) = \mathrm{Total}(\mathbb{X})(1, A)$$

Note: this carries a PCA in $\mathbb{X}$ to an "ordinary" PCA in Par, sets and partial maps.

Let $\mathbb{X}$ be any cartesian restriction category and suppose $\mathbb{A} = (A, \bullet)$ is an applicative system (i.e. $\bullet : A \times A \longrightarrow A$ is a partial operation) then $\Gamma(\mathbb{A})$ is an applicative system in Set. An **applicative set of codes** for $\mathbb{A}$ is a $\mathcal{V} \subseteq \Gamma(\mathbb{A}) = \mathrm{Total}(\mathbb{X})(1, A)$ which is a sub-applicative system (i.e closed to the application).

A map $A \times ... \times A \xrightarrow{h} A$ in $\mathbb{X}$ is $(\mathbb{A}, \mathcal{V})$-**computable** if there is an index $v \in \mathcal{V}$ with $(v \times 1 \times ... \times 1)\bullet^n = h$. Similarly, the maps $A^n \longrightarrow A^m$ $(m > 0)$ is computable in case each projection $A^n \longrightarrow A$ is computable. $h : A^n \longrightarrow 1$ is computable provided $\overline{h} : A \longrightarrow A$ is computable.

COMBINATORY COMPLETENESS

We shall say that an applicative system is **combinatory complete** relative to a set of indices $\mathcal{V}$ in case the $(\mathbb{A}, \mathcal{V})$-computable maps form a cartesian restriction subcategory.

Theorem

An applicative system $\mathbb{A}$, with respect to a set of indices $\mathcal{V}$, is combinatory complete if and only if $\mathcal{V}$ contains indices s *and* k *making $\mathbb{A}$ a partial combinatory algebra.*

FROM PCAs TO TURING CATEGORIES

This gives an very important method of generating Turing categories:

Theorem ($(\mathbb{A}, \mathcal{V})$-computability)

The $(\mathbb{A}, \mathcal{V})$-computable maps of any combinatory complete applicative system over any cartesian restriction category form a Turing category $\mathcal{C}(\mathbb{A}, \mathcal{V})$ with

$$C_A : \mathcal{C}(\mathbb{A}, \mathcal{V}) \longrightarrow \mathbb{X}$$

a faithful cartesian restriction functor.

Given a combinatory algebra in any cartesian restriction category an obvious set of indices to choose is the set of *all* points of the PCA. Conversely one can choose the smallest set generated by a choice of s and k ...

TURING SUBCATEGORIES

Given any cartesian restriction functor from a Turing category $F : \mathbb{T} \longrightarrow \mathbb{X}$ we may factorize it as

$$\mathbb{T} \xrightarrow{E(F)} \mathbb{T}/\cong \xrightarrow{M(F)} \mathbb{X}$$

where $E(F)$ forms the quotient of the category by $f \cong g \Leftrightarrow F(f) = F(g)$ and $M(F)$ is the residual faithful embedding.

$\mathbb{T}/\cong$ is a Turing category, thus, $M(F)$ is a faithful embedding of a Turing category into $\mathbb{X}$:

$$\mathbb{T}/\cong \xrightarrow{M(F)} \mathbb{X}$$

TURING SUBCATEGORIES

Any Turing object $T \in \mathbb{T}$ determines a PCA in $\mathbb{X}$ and a set of indices $\mathcal{V}_F = \{F(p) | p \in \text{points}(T)\}$.
Thus, F induces a faithful functor:

$$C_{F(T)} : \mathcal{C}(F(T), \mathcal{V}_F) \longrightarrow \mathbb{X}$$

Theorem

There is a factorization of any F with domain a Turing category as

$$\mathbb{T} \xrightarrow{F'} \mathrm{Split}(\mathcal{C}(F(T), \mathcal{V}_F)) \longrightarrow \mathrm{Split}(\mathbb{X})$$

Thus, up to splitting, faithful Turing subcategories of $\mathbb{X}$ are determined by combinatory complete applicative systems in $\mathbb{X}$ relative to a set of indices.

FOREVER UNDECIDED

We shall now examine undecidability results in Turing categories.
To get off the ground one needs a good notion of complement.
Joins provide this ...

THEREFORE we shall work now in Turing categories with (finite) joins.

UNDECIDABILITY

PROBLEM: there is a join Turing category in which everything is decidable!

It is the **trivial** join Turing category which has exactly one map between any two objects.

Undecidability proofs work by showing that if such and such is decidable then the Turing category must be trivial.

Lemma

A cartesian join restriction category is trivial in case any of the following are true:

- *The terminal object is a zero object;*
- *The identity map of the terminal object is the zero map;*
- *A total element has its restriction the zero map.*

PROOF: The three conditions are clearly equivalent. If the final object is a zero then $A \cong A \times 1 \cong A \times 0 \cong 0$! $\square$

UNDECIDABILITY

Let $\mathbb{T}$ be a Turing category with joins.

A restriction idempotent e is **complemented** (or **recursive**) in case there is a restriction e' with $ee' = 0$ and $e \vee e' = 1$.

Recall that in a join restriction category if $e : A$ has a complement $e' : A$ then A is the coproduct of the splittings of e and e'.

UNDECIDABILITY OF K

Theorem

In a join Turing category, $\mathbb{T}$, K has a complement if and only if $\mathbb{T}$ is trivial (i.e. exactly one map between each pair of objects).

Proof: Let K' be an idempotent with $K'K = 0$. Set $v = K'^{\bullet}$ be a code of K' (i.e. $(v \times 1)\bullet = K'$ and $\overline{v} = 1$) so that
$vK = v\overline{\Delta\bullet} = \overline{\langle v, v\rangle\bullet}v = \overline{vK'}v = vK'$ but then
$vK = vKK = vK'K = 0 = vKK' = vK'K' = vK'$ so that if
$K \vee K' = 1$ then $0 = \overline{0} = \overline{(vK) \vee (vK')} = \overline{v(K \vee K')} = \overline{v} = 1$ But
this collapses the final object and make the whole category trivial.
$\square$

Note that we have shown that K is “creative” (i.e. given $e = \overline{e}$ with $Ke = 0$ there is a point v with $vK = 0 = pe$). Clearly a creative idempotent in join cartesian restriction category has a complement only when the category is trivial.

RECURSION CATEGORIES

A **recursion category** is a discrete Turing category with joins. Explicitly this means it is a cartesian restriction category, which possesses a Turing structure, which also has joins and meets.

The classical category of computable functions is an example of a recursion category ...

First remarkable fact:

Split recursion categories always have coproducts!

RECURSION CATEGORIES

Theorem

Split recursion categories have coproducts and therefore are distributive restriction categories.

PROOF: The idea of the proof is as follows: if we had a boolean object so that $1 \xrightarrow{\text{true}} \mathsf{Bool} \xleftarrow{\text{false}} 1$ is a coproduct then by taking the product with the Turing object we would get a coproduct

$$1 \times A \xrightarrow{\text{true} \times 1_A} \mathsf{Bool} \times A \xleftarrow{\text{false} \times 1_A} 1 \times A$$

and so be able to take coproducts of the Turing object. However, as every object occurs as a retract of the Turing object it follows that there are coproducts for all objects. $\square$

RECURSION CATEGORIES

We *still* need to show that we have a Boolean object in a recursion category.

First note every total element $a : 1 \longrightarrow A$ is also a restriction monic as clearly $!a$ is an idempotent and so a splits the idempotent $!a \cap 1_A$. We need two elements which intersect at zero.
Consider the elements i and z:

$$
\begin{tikzcd}
1 \times A \arrow[r, "\pi_1"] \arrow[d, "\mathrm{i} \times 1_A"'] & A \\
A \times A \arrow[ur, "\bullet"']
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \times A \arrow[r, "0"] \arrow[d, "\mathrm{z} \times 1_A"'] & A \\
A \times A \arrow[ur, "\bullet"']
\end{tikzcd}
$$

RECURSION CATEGORIES

Call the intersection of these subobjects exists P:

$$
\begin{tikzcd}
P \arrow[r, "p"] \arrow[d, "q"'] & 1 \arrow[d, "\mathrm{i}"] \\
1 \arrow[r, "\mathrm{z}"'] & A
\end{tikzcd}
$$

we wish to show that $P = 0$. Consider

$$
\begin{tikzcd}
P \times 1 \arrow[r, "p\times 1"] \arrow[d, "q\times 1"'] & 1 \times 1 \arrow[d, "\mathrm{i}\times\mathrm{i}"] \arrow[ddr, bend left, "\pi_1\mathrm{i}"] & \\
1 \times 1 \arrow[r, "\mathrm{z}\times\mathrm{i}"'] \arrow[drr, bend right, "0"'] & A \times A \arrow[dr, "\bullet"] & \\
 & & A
\end{tikzcd}
$$

then $(p \times 1)\pi_1\mathrm{i} = 0$ as $\pi_1\mathrm{i}$ is monic $p \times 1 = 0$ but p is a restriction monic which forces $P = 0$.

INSEPARABILITY

A pair of restriction idempotents $e_0, e_1 : X$ are **recursively inseparable** in X if they are disjoint and there is no complemented idempotent e such that $e_0 \leq e$ and $e_1 \leq e'$.

Theorem (F. Lengyel)

Every non-trivial recursion category has inseparable restriction idempotents.

PROOF: The above assures us that we may find two total points $p_0, p_1 : 1 \longrightarrow T$ with $p_0 \cap 1, p_1 \cap 1 : T \longrightarrow T$ disjoint.
Any pair of such points will do.
Set $k_i = \overline{\Delta \bullet (p_i \cap 1)} : T \longrightarrow T$ this predicate is those codes which when applied to themselves evaluate to p_i.
Note k_0 and k_1 are disjoint as

$$k_0 k_1 = \overline{\overline{\Delta \bullet (p_0 \cap 1)}\Delta \bullet (p_1 \cap 1)} = \overline{\Delta \bullet (p_0 \cap 1)(p_1 \cap 1)} = 0.$$

Suppose that $k_i \leq u_i$ and $u_0 u_1 = 0$. We now show that assuming that $u_0 \vee u_1 = 1_T$ implies category is trivial.

INSEPARABILITY

Consider the map $q = u_0p_1 \vee u_1p_0$, note that it is total (as $\overline{q} = \overline{u_0p_1} \vee \overline{u_1p_0} = u_0 \vee u_1 = 1$) and it is, given our assumption, a decider for u_0. Define q' to be a code for q, so that $(q' \times 1)\bullet = q$. Observe that

$$
\begin{array}{rcl}
q'k_0 & = & q'\overline{\Delta \bullet (p_0 \cap 1)} = \overline{q'\Delta \bullet (p_0 \cap 1)}q' = \overline{q'(q' \times 1) \bullet (p_0 \cap 1)}q' \\
& = & \overline{q'(u_0p_1 \vee u_1p_0)(p_0 \cap 1)}q' = q'\overline{u_0p_1(p_0 \cap 1) \vee u_1p_0(p_0 \cap 1)} \\
& = & \overline{0 \vee u_1p_0}q' = \overline{q'u_1p_0}q' = q'u_1
\end{array}
$$

and similarly $q'k_1 = q'u_0$. This shows $q'u_1 = q'u_1u_1 = q'k_0u_1 = 0$ and similarly $q'u_0 = 0$ This is obviously bad and gives the following calculation to clinch it:

$$1_1 = \overline{q'} = \overline{q'(u_0 \vee u_1)} = \overline{q'u_0 \vee q'u_1)} = 0.$$

This suffices to show the category is trivial! $\square$

RECURSION THEOREM

The recursion theorems hold in any Turing category:

Theorem

In any Turing category, for any $f : T \times T \longrightarrow T$ there is a total point $e : 1 \longrightarrow T$ such that $(e \times 1)\bullet = (e \times 1)f$.

PROOF: Set $h = (\Delta \times 1)(\bullet \times 1)f$ then there is a code, $h^\bullet$ with $(h^\bullet \times 1)\bullet$ total and setting $e = (h^\bullet \times h^\bullet)\bullet$ makes

$$
\begin{tikzcd}
A \arrow[rrrr, bend left, "e\times 1"] \arrow[dr, "h^\bullet\times 1"] \arrow[ddrr, bend right, "e\times 1"'] & & & A\times A\times A \arrow[r, "\bullet\times 1"] & A\times A \arrow[dr, "f"] & \\
& A\times A \arrow[urr, "\Delta\times 1"] \arrow[rrrr, "h"] \arrow[d, dotted, "h^\bullet\times 1\times 1"] & & & & A \\
& A\times A\times A \arrow[r, "\bullet\times 1"'] & A\times A \arrow[urrr, "\bullet"'] & & &
\end{tikzcd}
$$

commutative. Note that e is total as $(h^\bullet \times 1)\bullet$ is total. $\square$

EXTENSIONAL PREDICATES

- We say that a restriction idempotent e on a Turing object is **extensional** (with respect to a given choice of Turing structure) in case the following implication holds for every f and g (using the term logic):

$$(e(f(x)) \bullet y = g(x) \bullet y \Rightarrow g(x)_{|e(f(x))} = e(gx)_{|e(f(x))}).$$

- Say that a restriction idempotent e on a Turing object is **non-trivial** in case there are two points, p_0 and p_1 with $p_0 e = p_0$ and $p_1 e = 0$.

Think of f and g as an indexes whose behaviors are the same then the extensionality of e requires that g lies in e *in so far* as f lies in e and is defined.

EXTENSIONAL PREDICATES

An example of an extensional predicate is $\overline{(1 \times p_1) \bullet (1 \cap p_2)}$: here we are testing whether a code on input p_1 will output p_2.

Also there is the following important fact:

Lemma

If e is extensional and has a complement e' then e' is extensional.

Theorem (Rice's theorem)

In a non-trivial recursion category no non-trivial extensional idempotent is complemented.

PROOF: (sketch) Suppose e with complement e' is extensional (so both are) and non-trivial (so both are). Thus, there are points p_0 and p_1 with $p_0 e = p_0$ and $p_1 e' = p_1$. Using the second recursion theorem define a point h by (using the term logic):

$$h \bullet x = p_1 \bullet x_{|e(h)} \vee p_0 \bullet x_{|e'(h)}$$

then

$$\begin{aligned} e(h) \bullet x &= h \bullet x_{|e(h)} = (p_1 \bullet x_{|e(h)} \vee p_0 \bullet x_{|e'(h)})_{|e(h)} \\ &= p_1 \bullet x_{|e(h)} = (p_1)_{|e(h)} \bullet x \end{aligned}$$

so using extensionality we have:

$$(p_1)_{|e(h)} = e((p_1)_{|e(h)}) = e(p_1)_{|e(h)} = 0$$

which implies $e(h) = 0$ but by symmetry $e'(h) = 0$ giving $h = 0$ showing the category must collapse. $\square$

Conclusion ...

The basic ideas of computability can be expressed quite smoothly in Turing Categories but ...

The BIG Question:

Can Turing categories bring new insights to computability theory?