# Higher Order Computability

John Longley & Dag Normann

April 4, 2022

## Contents



# 1 Theory of Computability Models

## 1.1 Further Examples of Higher-Order Models

### 1.1.1 Kleene's Second Model

Whereas $K_1$ captures a notion of computability for finitary data such as numbers, our next example embodies a notion of computability in an *infinitary* setting.

When our computation returns a result $F(g) \in \mathbb{N}$, it must do so after only a finite number of 'computation steps'

If $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is computable, the function $F$ can be completely described by recording, for each finite sequence $m_0, \ldots, m_{r-1}$, whether the information '$g(0) = m_0, \ldots, g(r-1) = m-1$' suffices to determine the value of $F(g)$, and if so, what the value is. We write $\langle \cdots \rangle$for standard computable operation

$\mathbb{N}^* \to \mathbb{N}$

$$
\begin{array}{ll}
f(\langle m_0, \dots, m_{r-1} \rangle) = m - 1 & \text{if } F(g) = m \text{ whenever } g(0) = m_0, \dots, g(r-1) = m_{r-1} \\
f(\langle m_0, \dots, m_{r-1} \rangle) = 0 & \text{'}g(0) = m_0, \dots, g(r-1) = m_{r-1}\text{' does not} \\
& \text{suffice to determine } F(g)
\end{array}
$$

We can then compute $F(g)$ from just $f$ and $g$

$$
\begin{aligned}
F(g) = (f \mid g) :=& f(\langle g(0), \dots, g(r-1) \rangle) - 1 \\
& \text{where } r = \mu r. f(\langle g(0), \dots, g(r-1) \rangle) > 0
\end{aligned}
$$

Even if $f \in \mathbb{N}^{\mathbb{N}}$ does not represent some $F$ in this way, we can regard the above formula as defining a *partial* computable function $(f \mid -) : \mathbb{N}^{\mathbb{N}} \rightharpoonup \mathbb{N}$

A small tweak is to obtain an application operation with codomain $\mathbb{N}^{\mathbb{N}}$ rather than $\mathbb{N}$. In effect, the computaion now accepts an additional argument $n \in N$, which can be assumed to be known before any queries to $g$ are made. We may therefore define

$$
\begin{aligned}
(f \odot g)(n) :=& f(\langle n, g(0), \dots, g(r-1) \rangle) - 1 \\
& \text{where } r = \mu r. f(\langle n, g(0), \dots, g(r-1) \rangle) > 0
\end{aligned}
$$

In general, this will define a **partial** function $\mathbb{N} \rightharpoonup \mathbb{N}$. We shall henceforth consider $f \odot g$ to be 'defined' only if the above formula yields a *total* function $\mathbb{N} \to \mathbb{N}$. In this way, we obtain a partial application operation

$$\odot : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \rightharpoonup \mathbb{N}^{\mathbb{N}}$$

The structure $(\mathbb{N}^{\mathbb{N}}, \odot)$ is known to be **Kleene's second model**

We can also restrict $\odot$ to the set of total *computable*' functions $\mathbb{N} \to \mathbb{N}$. Since the natural candidates for $k, s$ in $K_2$ are computable, this gives us both a PCA $K_2^{\text{eff}}$ and a relative PCA $(K_2; K_2^{\text{eff}})$

### 1.1.2 Typed $\lambda$-Calculi

We work in the type world $\mathsf{T} = \mathsf{T}^{\to}(\beta_0, \dots, \beta_{r-1})$ where $\beta_i$ are base types. We assume given some set $C$ of **constant symbols** $c$, each with an assigned type $\tau_c \in \mathsf{T}$; we also suppose we have an unlimited supply of **variable symbols** $x^\sigma$ for each $\sigma$.

$\Xi$ specifies **basic evaluation contexts** $K[-]$

**Example 1.1.** Let $C$ consist of the following constants

$$\begin{aligned} \hat{0} &: \mathtt{N} \\ suc &: \mathtt{N} \to \mathtt{N} \\ rec_\sigma &: \sigma \to (\sigma \to \mathtt{N} \to \sigma) \to \mathtt{N} \to \sigma \end{aligned}$$

Let $\Delta$ consist of the $\delta$-rules

$$\begin{aligned} rec_\sigma x f \hat{0} &\rightsquigarrow x \\ rec_\sigma x f (suc\ n) &\rightsquigarrow f(rec_\sigma x f n) n \end{aligned}$$

and let $\Xi$ consist of the contexts of the forms

$$[-]N, \quad suc[-], \quad rec_\sigma PQ[-]$$

The first means that if $M \rightsquigarrow M'$ then $MN \rightsquigarrow MN'$

This defines Gödel's **System** $T$

**Example 1.2** (Plotkin's PCF)**.** Let $C$ consists of the following constants

$$\begin{aligned} \hat{0} &: \mathtt{N} \quad \forall n \in \mathbb{N} \\ suc, pre &: \mathtt{N} \to \mathtt{N} \\ ifzero &: \mathtt{N}, \mathtt{N}, \mathtt{N} \to \mathtt{N} \\ Y_\sigma &: (\sigma \to \sigma) \to \sigma \quad \forall \sigma \in \mathsf{T} \end{aligned}$$

Let $\Delta$ consist of rules of the forms

$$\begin{aligned} suc\ \hat{n} &\rightsquigarrow \widehat{n+1} \\ pre\ \widehat{n+1} &\rightsquigarrow \hat{n} \\ pre\ \hat{0} &\rightsquigarrow \hat{0} \\ ifzero\ \hat{0} &\rightsquigarrow \lambda xy.x \\ ifzero\ \widehat{n+1} &\rightsquigarrow \lambda xy.y \\ Y_\sigma f &\rightsquigarrow f(Y_\sigma f) \end{aligned}$$

and let $\Xi$ consist of the contexts of the forms

$$[-]N, \quad suc[-], \quad pre[-], \quad ifzero[-]$$

The resulting language is known as Plotkin's PCF (*Programming language for Computable Functions*)

Given $\Delta$ and $\Xi$ as above, let us define a **one-step reduction** relation $\rightsquigarrow$ between closed terms of the same type by means of the following inductive definition

1. $(\lambda x^\sigma.M)N \rightsquigarrow M[x^\sigma \mapsto N]$ ($\beta$**-reduction**)
2. If $(cM_0 \dots M_{r-1} \rightsquigarrow N) \in \Delta$ and $-^*$ denotes some type-respecting substitution of closed terms for the free variables of $cM_0 \dots M_{r-1}$, then $(cM_0^* \dots M_{r-1}^*) \rightsquigarrow N^*$
3. If $M \rightsquigarrow M'$ and $K[-] \in \Xi$, then $K[M] \rightsquigarrow K[M']$

We call the entire system the **typed $\lambda$-calculus** specified by $\vec{\beta}, C, \Delta, \Xi$

Depending on the choice of $\Delta$ and $\Xi$, the relation $\rightsquigarrow$ may or may not be **deterministic** in the sense that $M \rightsquigarrow M' \wedge M \rightsquigarrow M'' \Rightarrow M' = M''$. Note that both System T and PCF have deterministic reduction relations.

Typically we are interested in the possible values of a closed term $M$, that is, those terms $M'$ s.t. $M \rightsquigarrow^* M'$ and $M'$ can be reduced no further. In System T, every closed $M : \mathtt{N}$ reduces to a unique numeral $suc^k\hat{0}$. In PCF, there are terms such as $Y_{\mathtt{N}}(\lambda x^{\mathtt{N}}.x)$ that can never be reduced to a value. Thus System T is in essence a language for **total** functions, whilst PCF is a language for **partial** ones

Let $=_{op}$ be the **operational equivalence** relation on closed terms generated by

$$M \rightsquigarrow M' \Rightarrow M =_{op} M', \quad M =_{op} M' \Rightarrow MN =_{op} M'N \wedge PM =_{op} PM'$$

## 1.2 Simulations Between Computability Models

### 1.2.1 Simulations and Transformations

**Definition 1.1.** Let **C** and **D** be lax computability models over type worlds $T, U$ respectively. A **simulation** $\gamma$ of **C** in **D** (written in $\gamma : \mathbf{C} \longrightarrow\!\!\!\!\triangleright\ \mathbf{D}$) consist of

- a mapping $\sigma \mapsto \gamma\sigma : T \to U$
- for each $\sigma \in T$, a relation $\Vdash^\gamma_\sigma \subseteq \mathbf{D}(\gamma\sigma) \times \mathbf{C}(\sigma)$

satisfying the following

1. For all $a \in \mathbf{C}(\sigma)$ there exists $a' \in \mathbf{D}(\gamma\sigma)$ s.t. $a' \Vdash^\gamma_\sigma a$
2. Every operation $f \in \mathbf{C}[\sigma, \tau]$ is **tracked** by some $f' \in \mathbf{D}[\gamma\sigma, \gamma\tau]$, in the sense that whenever $f(a) \downarrow$ and $a' \Vdash^\gamma_\sigma a$, we have $f'(a) \Vdash^\gamma_\tau f(a)$

For any $\mathbf{C}$ we have the **identity** simulation $\mathrm{id}_{\mathbf{C}} : \mathbf{C} \longrightarrow\!\!\triangleright \mathbf{C}$ given by $\mathrm{id}_{\mathbf{C}}\, \sigma = \sigma$ and $a' \Vdash_{\sigma}^{\mathrm{id}_{\mathbf{C}}} a$ iff $a' = a$

Given simulations $\gamma : \mathbf{C} \longrightarrow\!\!\triangleright \mathbf{D}$ and $\delta : \mathbf{D} \longrightarrow\!\!\triangleright \mathbf{E}$ we have the composite simulation $\delta \circ \gamma : \mathbf{C} \longrightarrow\!\!\triangleright \gamma\, \mathbf{E}$ defined by $(\delta \circ \gamma)\sigma = \delta(\gamma\sigma)$ and $a' \Vdash_{\sigma}^{\delta\circ\gamma} a$ iff there exists $a'' \in \mathbf{D}(\gamma\sigma)$ with $a'' \Vdash_{\sigma}^{\gamma} a$ and $a' \Vdash_{\gamma\sigma}^{\delta} a''$.

**Definition 1.2.** Let $\mathbf{C}$, $\mathbf{D}$ be lax computability models and suppose $\gamma, \delta : \mathbf{C} \longrightarrow\!\!\triangleright \mathbf{D}$ are simulations. We say $\gamma$ is **transformable** to $\delta$, and write $\gamma \preceq \delta$, if for each $\sigma \in |\mathbf{C}|$ there is an operation $t \in \mathbf{D}[\gamma\sigma, \delta\sigma]$ s.t.

$$\forall a \in \mathbf{C}(\sigma), a' \in \mathbf{D}(\gamma\sigma). a' \Vdash_{\sigma}^{\gamma} a \Rightarrow t(a') \Vdash_{\sigma}^{\delta} a$$

We write $\gamma \sim \delta$ if both $\gamma \preceq \delta$ and $\delta \preceq \gamma$

**Definition 1.3.** Models $\mathbf{C}, \mathbf{D}$ are **equivalent** ($\mathbf{C} \simeq \mathbf{D}$) if there exist simulations $\gamma : \mathbf{C} \longrightarrow\!\!\triangleright \mathbf{D}$ and $\delta : \mathbf{D} \longrightarrow\!\!\triangleright \mathbf{C}$ s.t. $\delta \circ \gamma \sim \mathrm{id}_{\mathbf{C}}$ and $\gamma \circ \delta \sim \mathrm{id}_{\mathbf{D}}$

A model is **essentially untyped** if it is equivalent to a model over the singleton type world $\mathsf{O}$

*Exercise* 1.2.1. Show that a model $\mathbf{C}$ is essentially untyped iff it contains a **universal type**: that is, a datatype $U$ s.t. for each $A \in |\mathbf{C}|$ there exists operations $e \in \mathbf{C}[A, U]$, $r \in \mathbf{C}[U, A]$ with $r(e(a)) = a$ for all $a \in A$

*Proof.* $\Leftarrow$: Let $\mathsf{O} = \{U\}$. For each $f \in \mathbf{C}[A, B]$, $\mathbf{D}$ contains $\bar{f} \in \mathbf{D}[U, U]$ s.t. $\bar{f}()$ Let

$$\mathbf{D}[U, U] = \{\bar{f} : e[A] \to e[B] : f \in \mathbf{C}[A, B]\}$$

where $\bar{f}(e(a)) = e(f(a))$

each $A \in |\mathbf{C}|$, let $\gamma(A) = U$ and define $a' \Vdash_{A}^{\gamma} a$ iff $a' = e(a)$ $\square$

**Definition 1.4.** Suppose $\mathbf{C}, \mathbf{D}$ are lax models with weak products and weak terminals $(I, i)$, $(J, j)$ respectively. A simulation $\gamma : \mathbf{C} \longrightarrow\!\!\triangleright \mathbf{D}$ is **cartesian** if

1. for each $\sigma, \tau \in |\mathbf{C}|$ there exists $t \in \mathbf{D}[\gamma\sigma \bowtie \gamma\tau, \gamma(\sigma \bowtie \tau)]$ s.t.

$$\pi_{\gamma\sigma}(d) \Vdash_{\sigma}^{\gamma} a \wedge \pi_{\gamma\tau}(d) \Vdash_{\tau}^{\gamma} b \Rightarrow$$
$$\exists c \in \mathbf{C}(\sigma \bowtie \tau). \pi_{\sigma}(c) = a \wedge \pi_{\tau}(c) = b \wedge t(d) \Vdash_{\sigma\bowtie\tau}^{\gamma} c$$

2. there exists $u \in \mathbf{D}[J, \gamma I]$ s.t. $u(j) \Vdash_{I}^{\gamma} i$

**Definition 1.5.** Let $\mathbf{A}$ and $\mathbf{B}$ be lax relative TPCAs over the type worlds $\mathsf{T}, \mathsf{U}$ respectively. An **applicative simulation** $\gamma : \mathbf{A} \longrightarrow\!\!\triangleright \mathbf{B}$ consists of

- a mapping $\sigma \mapsto \gamma\sigma : \mathsf{T} \to \mathsf{U}$
- for each $\sigma \in \mathsf{T}$, a relation $\Vdash^{\gamma}_{\sigma} \subseteq \mathbf{B}^{\circ}(\gamma\sigma) \times \mathbf{A}^{\circ}(\sigma)$

satisfying the following

1. For all $a \in \mathbf{A}^{\circ}(\sigma)$ there exists $b \in \mathbf{B}^{\circ}(\gamma\sigma)$ with $b \Vdash^{\gamma}_{\sigma} a$
2. For all $a \in \mathbf{A}^{\sharp}(\sigma)$ there exists $b \in \mathbf{B}^{\sharp}(\gamma\sigma)$ with $b \Vdash^{\gamma}_{\sigma} a$
3. 'Application in **A** is effective in **B**': that is, for each $\sigma, \tau \in \mathsf{T}$, there exists some $r \in \mathbf{B}^{\sharp}(\gamma(\sigma \to \tau) \to \gamma\sigma \to \gamma\tau)$, called a **realizer for** $\gamma$ **at** $\sigma$**,**$\tau$, s.t. for all $f \in \mathbf{A}^{\circ}(\sigma \to \tau)$, $f' \in \mathbf{B}^{\circ}(\gamma(\sigma \to \tau))$, $a \in \mathbf{A}^{\circ}(\sigma)$ and $a' \in \mathbf{B}^{\circ}(\gamma\sigma)$ we have

$$f' \Vdash_{\sigma \to \tau} f \wedge a' \Vdash_{\sigma} a \wedge f \cdot a \downarrow \Rightarrow r \cdot f' \cdot a' \Vdash_{\tau} f \cdot a$$

**Theorem 1.6.** *Suppose* **C** *and* **D** *are (lax) weakly cartesian closed models, and suppose* **A** *and* **B** *are the corresponding (lax) relative TPCAs with pairing via the correspondence of Theorem* **??**. *Then cartesian simulations* $\mathbf{C} \longrightarrow\!\!\!\!\triangleright\ \mathbf{D}$ *correspond precisely to applicative simulations* $\mathbf{A} \longrightarrow\!\!\!\!\triangleright\ \mathbf{B}$

*Proof.* Suppose first that $\gamma : \mathbf{C} \longrightarrow\!\!\!\!\triangleright\ \mathbf{D}$ is a cartesian simulation

1. Definition
2. Suppose $a \in \mathbf{A}^{\sharp}(\sigma)$ where $\mathbf{A}^{\circ}(\sigma) = A$. Then we may find $g \in \mathbf{C}[I, A]$ with $g(i) = a$, where $(I, i)$ is a weak terminal in **C**. Take $g' \in \mathbf{D}[\gamma I, \gamma A]$ tracking $g$, and compose it with $u \in \mathbf{D}[J, \gamma I]$, we obtain $g'' \in \mathbf{D}[J, \gamma A]$. Then $g''(j) \in \mathbf{B}^{\sharp}(\gamma\sigma)$, and it is easy to see that $g''(j) \Vdash^{\gamma}_{\sigma} a$
3. Let $\sigma, \tau$ be any types; then by the definition of weakly cartesian closedness, we have $app_{\sigma\tau} \in \mathbf{C}[(\sigma \to \tau) \times \sigma, \tau]$ tracked by some $app'_{\sigma\tau} \in \mathbf{D}[\gamma((\sigma \to \tau) \times \sigma), \gamma\tau]$. By definition of cartesian simulation, we have $t \in \mathbf{D}[\gamma(\sigma \to \tau) \times \gamma\sigma, \gamma((\sigma \to \tau) \times \sigma)]$, we have an operation $\mathbf{D}[\gamma(\sigma \to \tau) \times \gamma\sigma, \gamma\tau]$, and hence an operation $\mathbf{D}[\gamma(\sigma \to \tau), \gamma\sigma \to \gamma\tau]$, and then an operation $\mathbf{D}[J, \gamma(\sigma \to \tau) \to \gamma\sigma \to \gamma\tau]$, and hence realizer $r \in \mathbf{B}^{\sharp}(\gamma(\sigma \to \tau) \to \gamma\sigma \to \gamma\tau)$ with the required properties: for all $f \in \mathbf{A}^{\circ}(\sigma \to \tau)$, $f' \in \mathbf{B}^{\circ}(\gamma(\sigma \to \tau))$, $a \in \mathbf{A}^{\circ}(\sigma)$, $a' \in \mathbf{B}^{\circ}(\gamma\sigma)$, and $f' \Vdash_{\sigma \to \tau} f$, $a' \Vdash_{\sigma} a$, $f \cdot a \downarrow$, we have $t(f', a') \Vdash_{(\sigma \to \tau) \times \sigma} (f, a)$. Then $app'_{\sigma\tau}(t(f', a')) \Vdash^{\gamma}_{\tau} app_{\sigma\tau}(f, a)$

Conversely, suppose $\gamma : \mathbf{A} \longrightarrow\!\triangleright\, \mathbf{B}$ is an applicative simulation. To see that $\gamma$ is a simulation $\mathbf{C} \longrightarrow\!\triangleright\, \mathbf{D}$, it suffices to show that every operation in $\mathbf{C}$ is tracked by one in $\mathbf{D}$. But given $f \in \mathbf{C}[\sigma, \tau]$, we may find a corresponding element $a \in \mathbf{A}^\sharp(\sigma \to \tau)$, whence some $a' \in \mathbf{B}^\sharp(\gamma(\sigma \to \tau))$ with $a' \Vdash^\gamma_{\sigma\to\tau} a$; by using a realizer $r \in \mathbf{B}^\sharp$ for $\gamma$ at $\sigma$,$\tau$, we have an element $a'' \in \mathbf{B}^\sharp(\gamma\sigma \to \gamma\tau)$ and so a corresponding operation $f' \in \mathbf{D}[\gamma\sigma, \gamma\tau]$.

It remains to show that $\gamma$ is cartesian. For any types $\sigma$,$\tau$, we have by assumption an element $pair_{\sigma\tau} \in \mathbf{A}^\sharp(\sigma \to \tau \to \sigma \times \tau)$, yielding some $p \in \mathbf{C}[\sigma, \tau \to \sigma \times \tau]$. Since $\gamma$ is a simulation, this is tracked by some $p' \in \mathbf{D}[\gamma\sigma, \gamma(\tau \to \sigma \times \tau)]$. From the weak product structure in $\mathbf{D}$ we may thence obtain an operation

$$p'' \in \mathbf{D}[\gamma\sigma \times \gamma\tau, \gamma(\tau \to \sigma \times \tau) \times \gamma\tau]$$

and together with a realizer for $\gamma$ at $\tau$ and $\sigma \times \tau$, this yields an operation $t \in \mathbf{D}[\gamma\sigma \times \gamma\tau, \gamma(\sigma \times \tau)]$ with the required properties.

$i \in \mathbf{A}^\sharp(I)$, hence there is $b \in \mathbf{B}^\sharp(\gamma I)$ with $b \Vdash^\gamma_I i$. But $b = u(j)$ for some $u \in \mathbf{D}[J, \gamma I]$ $\square$

The notion of a transformation between simulations carries across immediately to the relative TPCA setting: an applicative simulation $\gamma : \mathbf{A} \longrightarrow\!\triangleright\, \mathbf{B}$ is transformable to $\delta$ if for each type $\sigma$ there exists $t \in \mathbf{B}^\sharp(\gamma\sigma \to \delta\sigma)$ s.t. $a' \Vdash^\gamma_\sigma a$ implies $t \cdot a' \Vdash^\delta_\sigma a$

**Definition 1.7.** Suppose $\mathbf{A}$ and $\mathbf{B}$ are (lax) relative TPCAs over $\mathsf{T}$ and $\mathsf{U}$ respectively, and suppose $\gamma : \mathbf{A} \longrightarrow\!\triangleright\, \mathbf{B}$ is an applicative simulation

1. $\gamma$ is **discrete** if $b \Vdash^\gamma a$ and $b \Vdash^\gamma a'$ imply $a = a'$

2. $\gamma$ is **single-valued** if for all $a \in \mathbf{A}$ there is exactly one $b \in \mathbf{B}$ with $b \Vdash^\gamma a$. $\gamma$ is **projective** if $\gamma \sim \gamma'$ for some single-valued $\gamma'$

3. If $\mathbf{A}$ and $\mathbf{B}$ have booleans $tt_{\mathbf{A}}$, $ff_{\mathbf{A}}$ and $tt_{\mathbf{B}}$, $ff_{\mathbf{B}}$ respectively, $\gamma$ **respects booleans** if there exists $q \in \mathbf{B}^\sharp$ s.t.

$$b \Vdash^\gamma tt_{\mathbf{A}} \Rightarrow q \cdot b = tt_{\mathbf{B}}, \quad b \Vdash^\gamma ff_{\mathbf{A}} \Rightarrow q \cdot b = ff_{\mathbf{B}}$$

4. If $\mathbf{A}$ and $\mathbf{B}$ have numerals $\hat{n}$ and $\tilde{n}$ respectively, $\gamma$ **respects numerals** if there exists $q \in \mathbf{B}^\sharp$ s.t. for all $n \in \mathbb{N}$

$$b \Vdash^\gamma \hat{n} \Rightarrow q \cdot b = \tilde{n}$$

5. If $\mathsf{T} = \mathsf{U}$, we say $\gamma$ is **type-respecting** if $\gamma$ is the identity on types, and moreover:

   - $\Vdash^{\gamma}_{\sigma} = \mathrm{id}_{\mathsf{T}[\sigma]}$ whenever $\mathsf{T}$ fixes the interpretation of $\sigma$
   - Application is itself a realizer for $\gamma$ at each $\sigma, \tau$: that is,

$$f' \Vdash^{\gamma}_{\sigma\to\tau} f \wedge a' \Vdash^{\gamma}_{\sigma} a \wedge f \cdot a \downarrow \Rightarrow f' \cdot a' \Vdash^{\gamma}_{\tau} f \cdot a$$

   - If $\mathsf{T}$ has product structure, then **A**, **B** have pairing and

$$a' \Vdash^{\gamma}_{\sigma} a \wedge b' \Vdash^{\gamma}_{\tau} b \Rightarrow pair \cdot a' \cdot b' \Vdash^{\gamma}_{\sigma\times\tau} pair \cdot a \cdot b$$
$$d' \Vdash^{\gamma}_{\sigma\times\tau} d \Rightarrow fst \cdot d' \Vdash^{\gamma}_{\sigma} fst \cdot d \wedge snd \cdot d' \Vdash^{\gamma}_{\sigma} snd \cdot d$$

*Remark.* Show that if $\gamma$ respects numerals then $\gamma$ respects booleans

### 1.2.2 Simulations and Assemblies

A simulation $\gamma : \mathbf{C} \longrightarrow\!\!\triangleright\ \mathbf{D}$ naturally induces a functor $\gamma_* : \mathcal{A}\mathrm{sm}(\mathbf{C}) \to \mathcal{A}\mathrm{sm}(\mathbf{D})$ , capturing the evident idea that any datatypes implementable within **C** must also be implementable in **D**

- On objects $X$, define $\gamma_*(X)$ by $|\gamma_*(X)| = |X|$, $\rho_{\gamma_*(X)} = \gamma\rho_X$ and $b \Vdash_{\gamma_*(X)} x$ iff $\exists a \in \mathbf{C}(\rho_X)$, $a \Vdash_X x \wedge b \Vdash^{\gamma}_{\rho_X} a$
- On morphisms $f : X \to Y$, define $\gamma_*(f) = f$. Note that if $r \in \mathbf{C}[\rho_X, \rho_Y]$ tracks $f$ as a morphism in $\mathcal{A}\mathrm{sm}(\mathbf{C})$, and $r' \in \mathbf{D}[\gamma\rho_X, \gamma\rho_Y]$ tracks $r$ w.r.t. $\gamma$, then $r'$ tracks $f$ as a morphism in $\mathcal{A}\mathrm{sm}(\mathbf{C})$

Moreover, a transformation $\xi : \gamma \to \delta$ yields a natural transformation $\xi_* : \gamma_* \to \delta_*$: just take $\xi_{*X} = \mathrm{id}_{|X|} : \gamma_*(X) \to \delta_*\,(X)$, and note that if $t$ witnesses $\gamma \preceq \delta$ at $X$, then $t$ tracks $\xi_{*\,X}$.

**Definition 1.8.** 1. $\gamma$ respects numerals iff $\gamma_*$ preserves the natural number object

## 1.3 Examples of Simulations and Transformations

**Example 1.3.** Suppose **C** is any (lax) computability model with weak products, and consider the following variation on the 'product completion' construction described in the proof of Theorem **??**. Let $\mathbf{C}^{\times}$ be the computability model whose datatypes are sets $A_0 \times \cdots \times A_{m-1}$ where $A_i \in |\mathbf{C}|$, and whose operations $f \in \mathbf{C}^{\times}[A_0 \times \cdots \times A_{m-1}, B_0 \times \cdots \times B_{n-1}]$ are those partial functions represented by some operation in $\mathbf{C}[A_0 \bowtie \cdots \bowtie A_{m-1}, B_0 \bowtie \cdots \bowtie B_{n-1}]$.

Clearly the inclusion $\mathbf{C} \hookrightarrow \mathbf{C}^\times$ and $\mathbf{C}^\times \to \mathbf{C}$ sending $A_0 \times \ldots A_{m-1}$ to $A_0 \bowtie \cdots \bowtie A_{m-1}$ are simulations. Moreover, they constitute an equivalence $\mathbf{C} \simeq \mathbf{C}^\times$. This shows that every strict (lax) computability model with weak products is equivalent to one with standard products

**Proposition 1.9.** *For any partial computable $f : \mathbb{N}^r \rightharpoonup \mathbb{N}$ there is an applicative expression $e_f : \mathtt{N}^{(r)} \to \mathtt{N}$ (involving constants 0, suc, $rec_{\mathtt{N}}$, min) s.t. in any model $\boldsymbol{A}$ with numerals and minimization we have $[\![e_f]\!]_v \in \boldsymbol{A}^\sharp$ (with the obvious valuation $v$) and*

$$\forall n_0, \ldots, n_{r-1}, m. f(n_0, \ldots, n_{r-1}) = m \Rightarrow [\![e_f]\!]_v \cdot \hat{n}_0 \cdot \cdots \cdot \hat{n}_{r-1} = \hat{m}$$

**Example 1.4.** Let $\mathbf{A}$ be any untyped (lax relative) PCA, or more generally any model with numerals $\bar{0}, \bar{1}, \ldots$ and minimization. We may then define a single-valued applicative simulation $\kappa : K_1 \longrightarrow\!\!\!\!\triangleright\, \mathbf{A}$ by taking $a \Vdash^\kappa n$ iff $a = \hat{n}$. Condition 3 is satisfied because the application operation $\cdot : \mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$ of $K_1$ is representable by an element of $\mathbf{A}^\sharp$

Model with single datatype $\mathbb{N}$ and whose operations $\mathbb{N} \rightharpoonup \mathbb{N}$ are precisely the Turing-computable partial functions. $K_1$ is weakly cartesian closed.

For any partial computable $f : \mathbb{N}^r \rightharpoonup \mathbb{N}$ there is an applicative expression $e_f : \mathtt{N}^{(r)} \to \mathtt{N}$ (involving constants 0, suc, $rec_{\mathtt{N}}$, min) s.t. in any model $\mathbf{A}$ with numerals and minimization we have $[\![e_f]\!]_v \in \mathbf{A}^\sharp$ (with the obvious valuation $v$) and

$$\forall n_0, \ldots, n_{r-1}, m. f(n_0, \ldots, n_{r-1}) = m \Rightarrow [\![e_f]\!]_v \cdot \hat{n}_0 \cdot \cdots \cdot \hat{n}_{r-1} = \hat{m}$$

In particular models, many choices of numerals may be available. For instance, if $\mathbf{A} = \mathbf{A}^\sharp = \Lambda / \sim$, then besides the *Curry numerals*, we also have the **Church numerals** $\tilde{n} = \lambda f.\lambda x.f^n x$

**Example 1.5.** The model $\Lambda^0 / = \beta$ consisting of closed $\lambda$-terms modulo $\beta$-equivalence. Let $\lceil - \rceil$ be any effective **Gödel numbering** of $\lambda$-terms as natural numbers, and define $\gamma : \Lambda^0 / =_\beta \longrightarrow\!\!\!\!\triangleright\, K_1$ by

$$m \Vdash^\gamma [M] \quad \text{iff} \quad m = \lceil M' \rceil \text{ for some } M' =_\beta M$$

Justaposition of $\lambda$-terms is tracked by a computable function $(\lceil M \rceil, \lceil N \rceil) \mapsto \lceil MN \rceil$ at the level of Gödel numbers. Thus $\gamma$ is applicative

We have transformations $\mathrm{id}_{K_1} \preceq \gamma \circ \kappa$ and $\gamma \circ \kappa \preceq \mathrm{id}_{K_1}$; these correspond to the observation that the 'encoding' and 'decoding' mappings $n \mapsto \lceil \hat{n} \rceil$ and $\lceil \hat{n} \rceil \mapsto n$ are computable. Also there is a term $P \in \Lambda^0$ s.t. $P(\widehat{\lceil M \rceil}) =_\beta M$ for any $M \in \Lambda^0$ by Kleene's enumeration theorem, therefore $\kappa \circ \gamma \preceq \mathrm{id}_{\Lambda^0 / =_\beta}$

However we don't have $\mathrm{id}_{\Lambda^0/=_\beta} \preceq \kappa \circ \gamma$

It can be easily shown that $K_1$ is not equivalent to $\Lambda^0/=_\beta$. Let us say a relative PCA $\mathbf{A}$ has **decidable equality** if there is an element $q \in \mathbf{A}^\sharp$ s.t.

$$q \cdot x \cdot y = \begin{cases} \mathit{tt} & x = y \\ \mathit{ff} & \text{otherwise} \end{cases}$$

Clearly $K_1$ has decidable equality, and it is easy t see that if $\mathbf{A} \simeq \mathbf{B}$ then $\mathbf{A}$ has decidable equality iff $\mathbf{B}$ does. However, if a total relative PCA $\mathbf{A}$ were to contain such an element, we could define $v = Y(q\ \mathit{ff})$ so that $v = q\ \mathit{ff}\ v$ which would yield a contradiction

Here $Y\ f = f(Y\ f)$

**Example 1.6.** We may translate System T terms $M$ to PCF terms $M^\theta$ simply by replacing all constants $rec_\sigma$ by suitable implementations of these recursors in PCF

In any untyped model, let $Z$ be a guarded recursion operator, define

$$R = \lambda rxfm.\mathit{if}(iszero\ m)(kx)(\lambda y.f(pre\ m))(rxf(pre\ m)\hat{0})$$

and take $rec = \lambda xfm.(ZR)xfmi$.

Such a translation induces a type-respecting applicative simulation $\theta$ : $\mathrm{T}^0 / =_{op} \longrightarrow\!\!\triangleright\ \mathrm{PCF}^0 / =_{op}$. This simulation is single-valued as $M =_{op} M' \Rightarrow M^\theta =_{op} M'^\theta$. It is easy to see that $\theta$ respects numerals

**Example 1.7.** We may also translate PCF into the untyped $\lambda$-calculus. One such translation $\phi$ may be defined on constants as follows. We write $\langle M, N\rangle$ for $pair\ M\ N$ and define $\hat{0} = \langle \lambda xy.x, \lambda xy.x$ and $\widehat{n+1} = \langle \lambda xy.y, \hat{n}\rangle$

$$\begin{aligned} \hat{n}^\phi &= \hat{n} \\ suc^\phi &= \lambda z.\langle \lambda xy.y, z\rangle \\ pre^\phi &= \lambda z.(fst\ z)\hat{0}(snd\ z) \\ ifzero^\phi &= fst \\ Y^\phi_\sigma &= (\lambda xy.y(xxy))(\lambda xy.y(xxy)) \end{aligned}$$

This induces an applicative simulation $\phi : \mathrm{PCF}^0 / =_{op} \longrightarrow\!\!\triangleright\ \Lambda^0/ =_\beta$ respecting numerals. Moreover if $M =_{op} M'$ then $M^\phi =_\beta M'^\phi$

**Example 1.8** (Interpretations of System $T$)**.** Let $\mathbf{A}$ be any total relative TPCA with numerals $\hat{0}, \hat{1}, ...$ of type `N`, and associated operations $suc$, $rec_\sigma$. To any closed term $M : \sigma$ in Gödel's System $T$, we may associate an element $[\![M]\!] \in \mathbf{A}^\sharp(\sigma)$ as follows: replace each occurrence of $\lambda$ by $\lambda^*$ to obtain a meta-expression $M^*$, then expand $M^*$ to an applicative expression $M^\dagger$ and evaluate it in **A**, interpreting the constants $\hat{0}$, $suc$, $rec_\sigma$ in the obvious way

Clearly $[\![MN]\!] = [\![M]\!] \cdot [\![N]\!]$ and if $M \rightsquigarrow M'$ then $[\![M]\!] = [\![M']\!]$. We therefore obtain a type-respecting simulation $[\![-]\!] : T^0/=_{op} \longrightarrow\!\!\!\triangleright\ \mathbf{A}$

Now suppose $\gamma : \mathbf{A} \longrightarrow\!\!\!\triangleright\ \mathbf{B}$ is a type- and numeral-respecting applicative simulation between total models with numerals. By the above, we have applicative simulations $[\![-]\!]_{\mathbf{A}} : \mathrm{T}^0 / =_{op} \longrightarrow\!\!\!\triangleright\ \mathbf{A}$ and $[\![-]\!]_{\mathbf{B}} : \mathrm{T}^0 / =_{op} \longrightarrow\!\!\!\triangleright\ \mathbf{B}$, but it nee not in general be the case that $\gamma \circ [\![-]\!]_{\mathbf{A}} \sim [\![-]\!]_{\mathbf{B}}$. This shows that a model **B** may in general admit several inequivalent applicative simulations of $\mathrm{T}^0 / =_{op}$

**Example 1.9** (Interpretations of PCF)**.** Let **A** be any strict relative TPCA with numerals and general recursion. Since **A** contains elements playing playing the role of the PCF constants $\hat{n}$, $suc$, $pre$, $ifzero$ and $Y_\sigma$, we may translate closed PCF terms $M : \sigma$ into expressions $\tilde{M} : \sigma$ over $A$ by simply replacing $\lambda$ with $\lambda^*$ and expanding. Note that $M =_{op} M'$ implies $[\![\tilde{M}]\!] \simeq [\![\tilde{M'}]\!]$ in **A**. This does not itself give us an applicative simulation $\mathrm{PCF}^0 / =_{op} \longrightarrow\!\!\!\triangleright$ **A** since $[\![\tilde{M}]\!]$ may sometimes be undefined. However, we may obtain such a simulation $\theta_{\mathbf{A}}$ via a suspension trick: let $\theta_{\mathbf{A}}\sigma = \mathbb{N} \rightarrow \sigma$ for each $\sigma$, and take $a \Vdash_\sigma^{\theta_{\mathbf{A}}} [M]$ iff $a \cdot \hat{0} \simeq [\![\tilde{M}]\!]$. (For example, we have $[\![\lambda^* u.\tilde{M}]\!] \Vdash [M]$ for any $M$) $\theta_{\mathbf{A}}$ is an applicative morphism, being realized at each $\sigma, \tau$ by $\lambda^* fxu.(fu)(xu)$

### 1.3.1 Effective and Continuous Models

It is natural to want to identify a class of computability models that are genuinely 'effective', in the sense that their data can be represented in some reasonable finitary way and their operations are ultimately implementable on a Turing machine.

**Definition 1.10.** An **effective (relative) TPCA** is a (relative) TPCA **A** with numerals equipped with a numeral-respecting applicative simulation $\gamma$ : $\mathbf{A} \longrightarrow\!\!\!\triangleright\ K_1$

Syntactic models in general, such as $\Lambda^0/\sim$ is effective TPCAs: The simulation $\gamma$ is given by a Gödel numbering of syntactic terms.

We may define $\gamma : K_2^{\mathrm{eff}} \longrightarrow\!\!\!\triangleright\ K_1$ by setting $m \Vdash^\gamma f$ iff $\forall n.m \cdot n = f(n)$

**Example 1.10.** The following shows what can happen if the numeral-respecting condition in definition is omitted. Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be some fixed noncomputable function. The definition of $K_1$ may be relativized to $f$ as follows: let $T'_0, T'_1, \ldots$ be a sensibly chosen enumeration of all Turing machines equipped with an *oracle* for $f$, so that a computation may ask for the value of some $f(m)$ as a single step. Now take $e \cdot^f n$ to be the result of applying $T'_e$ to the input $n$, and let $K_1^f = (\mathbb{N}, \cdot^f)$. The proof that $K_1$ is a PCA readily carries over to $K_1^f$

There is an applicative simulation $\gamma : K_1^f \longrightarrow\!\!\!\!\triangleright\, K_1$: we may take $m \Vdash^\gamma n$ iff $m \cdot^f 0 = n$

**Definition 1.11.** A **continuous (relative) TPCA** is a (relative) TPCA **A** with numerals equipped with a numeral-respecting applicative simulation $\gamma : \mathbf{A} \longrightarrow\!\!\!\!\triangleright\, K_2$

**Definition 1.12.** A continuous TPCA $(\mathbf{A}, \gamma)$ is **full continuous** if the following hold:

1. For every $f : \mathbb{N} \to \mathbb{N}$ there is some $a \in \mathbf{A}(\mathtt{N} \to \mathtt{N})$ that represents $f$
2. Moreover, a realizer for some such $a$ may be computed from $f$ within $K_2$ - that is, there exists $h \in K_2$ s.t. for all $f \in \mathbb{N}^{\mathbb{N}}$ we have