Differential Logical Relations

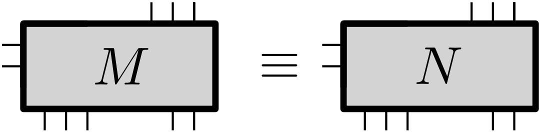
Joint work with Francesco Gavazzo and Akira Yoshimizu

Ugo Dal Lago

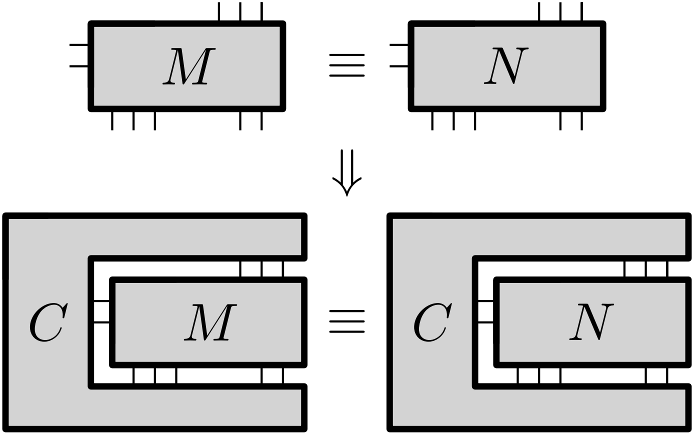
IFIP WG 2.2 Annual Meeting, Vienna, September 23rd 2019

This work has been supported by
the ERC CoG DIAPASoN, GA 818616.

Comparing Interacting Programs



Comparing Interacting Programs



Program Equivalence

- Let $\equiv \subseteq \Lambda \times \Lambda$ be a notion of equivalence. What are the minimal requirements we should put on $\equiv$?

Program Equivalence

- Let $\equiv \subseteq \Lambda \times \Lambda$ be a notion of equivalence. What are the minimal requirements we should put on $\equiv$?
- **Adequacy**: for every M, N,

$$M \equiv N \implies Obs(M) = Obs(N)$$

where $Obs : \Lambda \to \mathbb{X}$ is the *observation function*.

Program Equivalence

- Let $\equiv \subseteq \Lambda \times \Lambda$ be a notion of equivalence. What are the minimal requirements we should put on $\equiv$?
- **Adequacy**: for every M, N,

$$M \equiv N \implies Obs(M) = Obs(N)$$

  where $Obs : \Lambda \to \mathbb{X}$ is the *observation function*.
- **Congruence**: for every M, N, C,

$$M \equiv N \implies C[M] \equiv C[N].$$

Program Equivalence

- Let $\equiv \subseteq \Lambda \times \Lambda$ be a notion of equivalence. What are the minimal requirements we should put on $\equiv$?
- **Adequacy**: for every M, N,

$$M \equiv N \implies Obs(M) = Obs(N)$$

  where $Obs : \Lambda \to \mathbb{X}$ is the *observation function*.
- **Congruence**: for every M, N, C,

$$M \equiv N \implies C[M] \equiv C[N].$$

- Examples, the realm of λ-calculus:
  - The largest adequate congruence, **context equivalence** [Morris1968].
  - **Logical relations** [Plotkin1973].
  - **Applicative** [Abramsky1990] or **open bisimilarity**.

Program Equivalence

- Let $\equiv \subseteq \Lambda \times \Lambda$ be a notion of equivalence. What are the minimal requirements we should put on $\equiv$?
- **Adequacy**: for every M, N,

$$M \equiv N \implies Obs(M) = Obs(N)$$

  where $Obs : \Lambda \to \mathbb{X}$ is the *observation function*.
- **Congruence**: for every M, N, C,

$$M \equiv N \implies C[M] \equiv C[N].$$

- Examples, the realm of λ-calculus:
  - The largest adequate congruence, **context equivalence** [Morris1968].
  - **Logical relations** [Plotkin1973].
  - **Applicative** [Abramsky1990] or **open bisimilarity**.
- What if $\mathbb{X}$ is a metric space?

Program *Distance*

- Let $\delta : \Lambda \times \Lambda \to \mathbb{R}$.

Program *Distance*

- Let $\delta : \Lambda \times \Lambda \to \mathbb{R}$.
- **Adequacy**: for every M, N,

$$\delta(M, N) \geq \delta_{\mathbb{X}}(Obs(M), Obs(N))$$

Program *Distance*

- Let $\delta : \Lambda \times \Lambda \to \mathbb{R}$.
- **Adequacy**: for every M, N,

$$\delta(M, N) \geq \delta_{\mathbb{X}}(Obs(M), Obs(N))$$

- **Non-Expansiveness**: for every M, N, C,

$$\delta(M, N) \geq \delta(C[M], C[N])$$

Program *Distance*

- Let $\delta : \Lambda \times \Lambda \to \mathbb{R}$.
- **Adequacy**: for every M, N,

$$\delta(M, N) \geq \delta_{\mathbb{X}}(Obs(M), Obs(N))$$

- **Non-Expansiveness**: for every M, N, C,

$$\delta(M, N) \geq \delta(C[M], C[N])$$

- In *probabilistic* computation, one is naturally lead to observe a quantitative property, and $\mathbb{X}$ is simply $\mathbb{R}$.

Program *Distance*

- Let $\delta : \Lambda \times \Lambda \to \mathbb{R}$.
- **Adequacy**: for every M, N,

$$\delta(M, N) \geq \delta_{\mathbb{X}}(Obs(M), Obs(N))$$

- **Non-Expansiveness**: for every M, N, C,

$$\delta(M, N) \geq \delta(C[M], C[N])$$

- In *probabilistic* computation, one is naturally lead to observe a quantitative property, and $\mathbb{X}$ is simply $\mathbb{R}$.
- But even when computation is *deterministic*, one could well work with $\mathbb{X} = \mathbb{R}$ when real numbers are part of the underlying language [ReedPierce2010,AGHKC2017].

An Example

1

A Survey Of Techniques for Approximate Computing

Sparsh Mittal, Oak Ridge National Laboratory

Approximate computing trades off computation quality with the effort expended and as rising performance demands confront with plateauing resource budgets, approximate computing has become, not merely attractive, but even imperative. In this paper, we present a survey of techniques for approximate computing (AC). We discuss strategies for finding approximable program portions and monitoring output quality, techniques for using AC in different processing units (e.g., CPU, GPU and FPGA), processor components, memory technologies etc., and programming frameworks for AC. We classify these techniques based on several key characteristics to emphasize their similarities and differences. The aim of this paper is to provide insights to researchers into working of AC techniques and inspire more efforts in this area to make AC the mainstream computing approach in future systems.

Categories and Subject Descriptors: [**General and reference**]: Surveys and overviews; [**Hardware**]: Power and energy; [**Computer systems organization**]: Processors and memory architectures

General Terms: Design, Performance

Additional Key Words and Phrases: Review, classification, approximate computing technique (ACT), approximate storage, quality configurability, CPU, GPU, FPGA, neural networks

ACM Reference Format:
S. Mittal, "A Survey Of Techniques for Approximate Computing", 20xx. *ACM Comput. Surv.* a, b, Article 1 (2015), 34 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

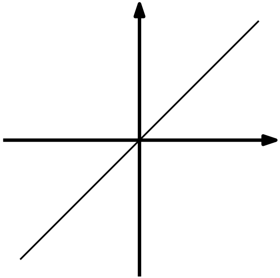
As large-scale applications such as scientific computing, social media and financial analysis gain prominence, the computational and storage demands of modern systems

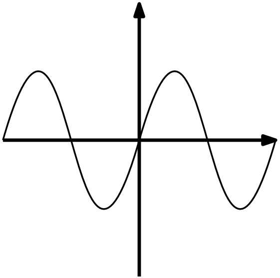
An Example

$$M_{ID} \equiv \lambda x.x \qquad M_{SIN} \equiv \lambda x.\texttt{sin}\ x$$

An Example

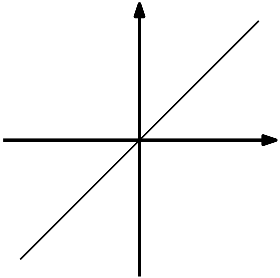
$$M_{ID} \equiv \lambda x.x \qquad\qquad M_{SIN} \equiv \lambda x.\texttt{sin}\ x$$

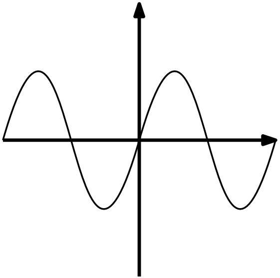




An Example

$$M_{ID} \equiv \lambda x.x \qquad M_{SIN} \equiv \lambda x.\mathtt{sin}\, x$$

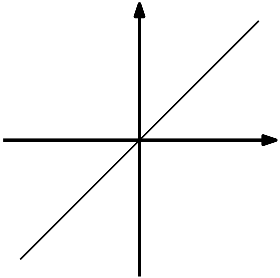


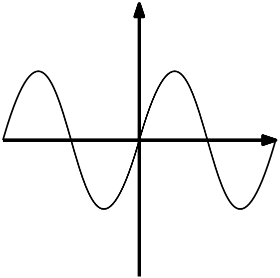


$$\delta(M_{ID}, M_{SIN}) = +\infty$$

An Example

$M_{ID} \equiv \lambda x.x$ $\qquad$ $M_{SIN} \equiv \lambda x.\mathtt{sin}\, x$





$$\delta(M_{ID}, M_{SIN}) = +\infty$$

What if the *environment* feeds the function with values close to 0, only?

An Example

A Semantics for Approximate Program Transformations

Edwin Westbrook and Swarat Chaudhuri
Department of Computer Science, Rice University
Houston, TX 77005
Email: {emw4,swarat}@rice.edu

Abstract

An approximate program transformation *is a transformation that can change the semantics of a program within a specified empirical error bound. Such transformations have wide applications: they can decrease computation time, power consumption, and memory usage, and can, in some cases, allow implementations of incomputable operations. Correctness proofs of approximate program transformations are by definition quantitative. Unfortunately, unlike with standard program transformations, there is as of yet no modular way to prove correctness of an approximate transformation itself. Error bounds must be proved for each transformed program individually, and must be reproved each time a program is modified or a different set of approximations are applied.*

as floating-point numbers, lossy compression, and approximation algorithms for NP-hard problems. Such techniques are often used to trade off accuracy of the result for **reduced resource usage**, for resources such as computation time, power, and memory. In addition, some approximation techniques are also used to ensure **computability**. For example, true representations of real numbers (e.g., [7], [1]), require some operations, such as comparison, to be incomputable; floating-point comparison, in contrast, is efficiently decidable on modern computers.

Recently, there has been a growing interest in *language-based approximations*, where *approximate program transformations* are performed by the programming language environment [21], [12], [19], [18], [4], [3], [16]. Such approaches allow the user to give an *exact program* as a specification, and then apply some

Types

$$\tau, \rho ::= REAL \,\big|\, \tau \to \rho \,\big|\, \tau \times \rho.$$

A Toy Language

Types

$$\tau, \rho ::= REAL \mid \tau \to \rho \mid \tau \times \rho.$$

Typing Rules

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{}{\Gamma \vdash r : REAL} \qquad \frac{f_n \in \mathcal{F}_n}{\Gamma \vdash f_n : REAL^n \to REAL} \qquad \frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \to \rho}$$

$$\frac{\Gamma \vdash M : \tau \to \rho \quad \Gamma \vdash N : \tau}{\Gamma \vdash MN : \rho} \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \rho}{\Gamma \vdash \langle M, N \rangle : \tau \times \rho} \qquad \frac{}{\Gamma \vdash \pi_1 : \tau \times \rho \to \tau} \qquad \frac{}{\Gamma \vdash \pi_2 : \tau \times \rho \to \rho}$$

$$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \texttt{iflz}\ M\ \texttt{else}\ N : REAL \to \tau} \qquad \frac{\Gamma \vdash M : \tau \to \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \texttt{iter}\ M\ \texttt{base}\ N : REAL \to \tau}$$

A Toy Language

Types

$$\tau, \rho ::= REAL \,\big|\, \tau \to \rho \,\big|\, \tau \times \rho.$$

Typing Rules

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{}{\Gamma \vdash r : REAL} \qquad \frac{f_n \in \mathcal{F}_n}{\Gamma \vdash f_n : REAL^n \to REAL} \qquad \frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \to \rho}$$

$$\frac{\Gamma \vdash M : \tau \to \rho \quad \Gamma \vdash N : \tau}{\Gamma \vdash MN : \rho} \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \rho}{\Gamma \vdash \langle M, N \rangle : \tau \times \rho} \qquad \frac{}{\Gamma \vdash \pi_1 : \tau \times \rho \to \tau} \qquad \frac{}{\Gamma \vdash \pi_2 : \tau \times \rho \to \rho}$$

$$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \texttt{iflz}\ M\ \texttt{else}\ N : REAL \to \tau} \qquad \frac{\Gamma \vdash M : \tau \to \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \texttt{iter}\ M\ \texttt{base}\ N : REAL \to \tau}$$

Denotational Semantics

$$[\![REAL]\!] = \mathbb{R}; \qquad [\![\tau \to \rho]\!] = [\![\tau]\!] \to [\![\rho]\!]; \qquad [\![\tau \times \rho]\!] = [\![\tau]\!] \times [\![\rho]\!].$$

Distance Spaces

$(\!|REAL|\!) = \mathbb{R}^{\infty}_{\geq 0}$; $\quad (\!|\tau \to \rho|\!) = [\![\tau]\!] \times (\!|\tau|\!) \to (\!|\rho|\!)$; $\quad (\!|\tau \times \rho|\!) = (\!|\tau|\!) \times (\!|\rho|\!)$

Distance Spaces

$(\!|REAL|\!) = \mathbb{R}^{\infty}_{\geq 0}$; $\quad (\!|\tau \to \rho|\!) = [\![\tau]\!] \times (\!|\tau|\!) \to (\!|\rho|\!)$; $\quad (\!|\tau \times \rho|\!) = (\!|\tau|\!) \times (\!|\rho|\!)$

The distance between two programs of type $\tau \to \rho$ is a function which:

- Given an input in $[\![\tau]\!]$...
- And a distance in $(\!|\tau|\!)$...
- Returns a distance in $(\!|\rho|\!)$.

Distance Spaces

$$(\!|REAL|\!) = \mathbb{R}^{\infty}_{\geq 0}; \qquad (\!|\tau \to \rho|\!) = [\![\tau]\!] \times (\!|\tau|\!) \to (\!|\rho|\!); \qquad (\!|\tau \times \rho|\!) = (\!|\tau|\!) \times (\!|\rho|\!)$$

DLRs as Ternary Relations

$$\begin{aligned}
\delta_{REAL}(M, r, N) &\Leftrightarrow |NF(M) - NF(N)| \leq r; \\
\delta_{\tau \times \rho}(M, (d_1, d_2), N) &\Leftrightarrow \delta_\tau(\pi_1 M, d_1, \pi_1 N) \wedge \delta_\rho(\pi_2 M, d_2, \pi_2 N) \\
\delta_{\tau \to \rho}(M, d, N) &\Leftrightarrow (\forall V \in CV(\tau).\ \forall x \in (\!|\tau|\!).\ \forall W \in CV(\tau). \\
&\qquad \delta_\tau(V, x, W) \Rightarrow \delta_\rho(MV, d([\![V]\!], x), NW) \wedge \delta_\rho(MW, d([\![V]\!], x), NV)).
\end{aligned}$$

Distance Spaces

$$(\!|REAL|\!) = \mathbb{R}^{\infty}_{\geq 0}; \qquad (\!|\tau \to \rho|\!) = [\![\tau]\!] \times (\!|\tau|\!) \to (\!|\rho|\!); \qquad (\!|\tau \times \rho|\!) = (\!|\tau|\!) \times (\!|\rho|\!)$$

DLRs as Ternary Relations

$$\begin{aligned}
\delta_{REAL}(M, r, N) &\Leftrightarrow |NF(M) - NF(N)| \leq r; \\
\delta_{\tau\times\rho}(M, (d_1, d_2), N) &\Leftrightarrow \delta_\tau(\pi_1 M, d_1, \pi_1 N) \wedge \delta_\rho(\pi_2 M, d_2, \pi_2 N) \\
\delta_{\tau\to\rho}(M, d, N) &\Leftrightarrow (\forall V \in CV(\tau).\ \forall x \in (\!|\tau|\!).\ \forall W \in CV(\tau). \\
&\qquad \delta_\tau(V, x, W) \Rightarrow \delta_\rho(MV, d([\![V]\!], x), NW) \wedge \delta_\rho(MW, d([\![V]\!], x), NV)).
\end{aligned}$$

Theorem (Fundamental Lemma, Version I)

For every $\vdash M : \tau$, *there is* $d \in (\!|\tau|\!)$ *such that* $\delta_\tau(M, d, M)$.

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?
  - In fact, this is true only at ground types.
  - *Example*: the distance between M_{ID} and itself is something like $\lambda\!\!\lambda\langle x, \varepsilon\rangle.\varepsilon$.

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?
  - In fact, this is true only at ground types.
  - *Example*: the distance between M_{ID} and itself is something like $\lambda\!\!\lambda\langle x, \varepsilon\rangle.\varepsilon$.
- **Too weak?**
  - The distance d is arbitrary, and can even be infinite.
  - In ordinary logical relations, the FL enables compositional reasoning, when applied to the environment, the context.

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?
  - In fact, this is true only at ground types.
  - *Example*: the distance between M_{ID} and itself is something like $\lambda\!\!\lambda\langle x, \varepsilon\rangle.\varepsilon$.
- **Too weak?**
  - The distance d is arbitrary, and can even be infinite.
  - In ordinary logical relations, the FL enables compositional reasoning, when applied to the environment, the context.
  - *The same here*:

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?
  - In fact, this is true only at ground types.
  - *Example*: the distance between M_{ID} and itself is something like $\lambda\!\!\lambda\langle x, \varepsilon\rangle.\varepsilon$.
- **Too weak?**
  - The distance d is arbitrary, and can even be infinite.
  - In ordinary logical relations, the FL enables compositional reasoning, when applied to the environment, the context.
  - *The same here*:

$$C : \tau \to REAL$$

$$M, N : \tau$$

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?
  - In fact, this is true only at ground types.
  - *Example*: the distance between M_{ID} and itself is something like $\lambda\!\!\lambda\langle x, \varepsilon\rangle.\varepsilon$.
- **Too weak?**
  - The distance d is arbitrary, and can even be infinite.
  - In ordinary logical relations, the FL enables compositional reasoning, when applied to the environment, the context.
  - *The same here*:

$$
\begin{array}{cc}
C : \tau \to REAL & (C, d, C) \in \delta_{\tau \to REAL} \\
M, N : \tau & (M, e, N) \in \delta_{\tau}
\end{array}
$$

On the Fundamental Lemma — So What?

- **Why not null distances?**
  - The distance between a program M *and itself* is null, isn't it?
  - In fact, this is true only at ground types.
  - *Example*: the distance between M_{ID} and itself is something like $\lambda\!\!\lambda\langle x, \varepsilon\rangle.\varepsilon$.
- **Too weak?**
  - The distance d is arbitrary, and can even be infinite.
  - In ordinary logical relations, the FL enables compositional reasoning, when applied to the environment, the context.
  - *The same here*:

$$
\begin{array}{cc}
C : \tau \to REAL & (C, d, C) \in \delta_{\tau \to REAL} \\
M, N : \tau & (M, e, N) \in \delta_{\tau}
\end{array}
$$

$$\Downarrow$$

$$(C[M], d([\![M]\!], e), C[N]) \in \delta_{REAL}$$

Back to the Example

Claim

$$\delta_{REAL \to REAL}(M_{ID}, \lambda\langle x, y\rangle.y + |x - \sin x|, M_{SIN})$$

Back to the Example

Claim

$$\delta_{REAL \to REAL}(M_{ID}, \lambda\!\!\!\lambda \langle x, y\rangle . y + |x - \sin x|, M_{SIN})$$

Proof.

Consider any pairs of real numbers $r, s \in \mathbb{R}$ such that $|r - s| \leq \varepsilon$, where $\varepsilon \in \mathbb{R}^{\infty}_{\geq 0}$. We have that:

$$\begin{aligned} |\sin r - s| &= |\sin r - r + r - s| \leq |\sin r - r| + |r - s| \\ &\leq |\sin r - r| + \varepsilon = f(r, \varepsilon) \\ |\sin s - r| &= |\sin s - \sin r + \sin r - r| \\ &\leq |\sin s - \sin r| + |\sin r - r| \leq |s - r| + |\sin r - r| \\ &\leq \varepsilon + |\sin r - r| = f(r, \varepsilon). \end{aligned}$$

where $f = \lambda\!\!\!\lambda \langle x, y\rangle . y + |x - \sin x|$. □

Back to the Example

- Now, consider, e.g., the context $C = (\lambda x.x(x\theta))[\cdot]$.

Back to the Example

- Now, consider, e.g., the context $C = (\lambda x.x(x\theta))[\cdot]$.
- C can be seen as *a term* having type $\tau = (REALS \to REALS) \to REALS$. A self-distance d for C can thus be defined as an element of

  $$(\!|\tau|\!) = [\![REALS \to REALS]\!] \times (\!|REALS \to REALS|\!) \to [\![REALS]\!].$$

  namely $F = \lambda\!\!\lambda \langle g, h\rangle .h(g(\theta), h(\theta, 0))$.

Back to the Example

- Now, consider, e.g., the context $C = (\lambda x.x(x\theta))[\cdot]$.
- C can be seen as *a term* having type $\tau = (REALS \rightarrow REALS) \rightarrow REALS$. A self-distance d for C can thus be defined as an element of

  $$(\!|\tau|\!) = [\![REALS \rightarrow REALS]\!] \times (\!|REALS \rightarrow REALS|\!) \rightarrow [\![REALS]\!].$$

  namely $F = \lambda\!\!\lambda \langle g, h\rangle.h(g(\theta), h(\theta, 0))$.
- This allows for **compositional reasoning** about program distances: the overall impact of replacing M_{SIN} by M_{ID} can be evaluated by computing $F([\![M_{ID}]\!], f)$ or $F([\![M_{SIN}]\!], f)$.

Back to the Example

- Now, consider, e.g., the context $C = (\lambda x.x(x\theta))[\cdot]$.
- C can be seen as *a term* having type $\tau = (REALS \to REALS) \to REALS$. A self-distance d for C can thus be defined as an element of

  $$(\!|\tau|\!) = [\![REALS \to REALS]\!] \times (\!|REALS \to REALS|\!) \to [\![REALS]\!].$$

  namely $F = \lambda\!\!\lambda\langle g, h\rangle.h(g(\theta), h(\theta, 0))$.
- This allows for **compositional reasoning** about program distances: the overall impact of replacing M_{SIN} by M_{ID} can be evaluated by computing $F([\![M_{ID}]\!], f)$ or $F([\![M_{SIN}]\!], f)$.
- Of course the context C needs to be taken into account, but **once and for all**: the functional F can be built without knowing either M_{SIN} or M_{ID}.

Hereditarily Null Distances

$$(\!|REAL|\!)^0 = \{0\} \qquad (\!|\tau \times \rho|\!)^0 = (\!|\tau|\!)^0 \times (\!|\rho|\!)^0$$

$$(\!|\tau \to \rho|\!)^0 = \{f \mid \forall x \in [\![\tau]\!]. \forall y \in (\!|\tau|\!)^0 . f(x, y) \in (\!|\rho|\!)^0\}$$

Hereditarily Null Distances

$$(\!|REAL|\!)^0 = \{0\} \qquad (\!|\tau \times \rho|\!)^0 = (\!|\tau|\!)^0 \times (\!|\rho|\!)^0$$

$$(\!|\tau \to \rho|\!)^0 = \{f \mid \forall x \in [\![\tau]\!].\forall y \in (\!|\tau|\!)^0.f(x,y) \in (\!|\rho|\!)^0\}$$

This is much larger than $\{\lambda\!\!\lambda\langle x, y\rangle.0\}$

Different Kinds of Distances

Hereditarily Null Distances

$$(\!|REAL|\!)^0 = \{0\} \qquad (\!|\tau \times \rho|\!)^0 = (\!|\tau|\!)^0 \times (\!|\rho|\!)^0$$

$$(\!|\tau \to \rho|\!)^0 = \{f \mid \forall x \in [\![\tau]\!].\forall y \in (\!|\tau|\!)^0.f(x,y) \in (\!|\rho|\!)^0\}$$

Hereditarily Finite Distances

$$(\!|REAL|\!)^{<\infty} = \mathbb{R}_{\geq 0}; \qquad (\!|\tau \times \rho|\!)^{<\infty} = (\!|\tau|\!)^{<\infty} \times (\!|\rho|\!)^{<\infty};$$

$$(\!|\tau \to \rho|\!)^{<\infty} = \{f \in (\!|\tau \to \rho|\!) \mid \forall x \in [\![\tau]\!].\forall t \in (\!|\tau|\!)^{<\infty}.f(x,t) \in (\!|\rho|\!)^{<\infty}\}.$$

Different Kinds of Distances

Hereditarily Null Distances

$$(\!|REAL|\!)^0 = \{0\} \qquad (\!|\tau \times \rho|\!)^0 = (\!|\tau|\!)^0 \times (\!|\rho|\!)^0$$

$$(\!|\tau \to \rho|\!)^0 = \{f \mid \forall x \in [\![\tau]\!].\forall y \in (\!|\tau|\!)^0.f(x,y) \in (\!|\rho|\!)^0\}$$

Hereditarily Finite Distances

$$(\!|REAL|\!)^{<\infty} = \mathbb{R}_{\geq 0}; \qquad (\!|\tau \times \rho|\!)^{<\infty} = (\!|\tau|\!)^{<\infty} \times (\!|\rho|\!)^{<\infty};$$

$$(\!|\tau \to \rho|\!)^{<\infty} = \{f \in (\!|\tau \to \rho|\!) \mid \forall x \in [\![\tau]\!].\forall t \in (\!|\tau|\!)^{<\infty}.f(x,t) \in (\!|\rho|\!)^{<\infty}\}.$$

Lemma

Whenever $\vdash M, N : \tau$, M *is logically related to* N *iff* $\delta_\tau(M, d, N)$ *where* $d \in (\!|\tau|\!)^0$.

Different Kinds of Distances

Hereditarily Null Distances

$$(\!|REAL|\!)^0 = \{0\} \qquad (\!|\tau \times \rho|\!)^0 = (\!|\tau|\!)^0 \times (\!|\rho|\!)^0$$

$$(\!|\tau \to \rho|\!)^0 = \{f \mid \forall x \in [\![\tau]\!].\forall y \in (\!|\tau|\!)^0.f(x,y) \in (\!|\rho|\!)^0\}$$

Hereditarily Finite Distances

$$(\!|REAL|\!)^{<\infty} = \mathbb{R}_{\geq 0}; \qquad (\!|\tau \times \rho|\!)^{<\infty} = (\!|\tau|\!)^{<\infty} \times (\!|\rho|\!)^{<\infty};$$

$$(\!|\tau \to \rho|\!)^{<\infty} = \{f \in (\!|\tau \to \rho|\!) \mid \forall x \in [\![\tau]\!].\forall t \in (\!|\tau|\!)^{<\infty}.f(x,t) \in (\!|\rho|\!)^{<\infty}\}.$$

Lemma

Whenever $\vdash M, N : \tau$, M *is logically related to* N *iff* $\delta_\tau(M, d, N)$ *where* $d \in (\!|\tau|\!)^0$.

Theorem (Fundamental Lemma, Version II)

For every $\vdash M : \tau$, *there is* $d \in (\!|\tau|\!)^{<\infty}$ *such that* $\delta_\tau(M, d, M)$.

Conclusions

- **Other Interesting Results**
  - Differential logical relations are examples of generalized metric domains, which (contrarily to metric spaces) form a *cartesian closed category*.
  - Not only logical relations, but also metric logical relations can be seen as special kinds of distances.

Conclusions

- **Other Interesting Results**
  - Differential logical relations are examples of generalized metric domains, which (contrarily to metric spaces) form a *cartesian closed category*.
  - Not only logical relations, but also metric logical relations

$$\delta(x,0,y) \Rightarrow x = y$$
$$\delta(x,d,y) \Rightarrow \delta(y,d,x)$$
$$\delta(x,d,y) \wedge \delta(y,e,y) \wedge \delta(y,f,z) \Rightarrow \delta(x,d+e+f,z)$$

Conclusions

- **Other Interesting Results**
  - Differential logical relations are examples of generalized metric domains, which (contrarily to metric spaces) form a *cartesian closed category*.
  - Not only logical relations, but also metric logical relations can be seen as special kinds of distances.

- **Further Work**
  - Capture more expressive constructs and calculi (e.g. *recursion* by way of step-indexing).
  - Higher-order continuity?
  - Abstracting functional distances by way of step functions.
  - Make behavioral metrics *context-dependent*.

Thank You!

Questions?