Dynamic Games and Strategies

Norihiro Yamada
and
Samson Abramsky

Norihiro.Yamada@cs.ox.ac.uk
Samson.Abramsky@cs.ox.ac.uk
Department of Computer Science
University of Oxford

August 28, 2015

Table of Contents

Brief Summary of the Talk

Brief Summary of the Talk

Background and motivation:

- The existing game semantics *a priori* "hides" all the "intermediate moves" when strategies are composed.
- Thus, it is "static" and "not fully intentional".

Brief Summary of the Talk

Background and motivation:

- The existing game semantics *a priori* "hides" all the "intermediate moves" when strategies are composed.
- Thus, it is "static" and "not fully intentional".

Research Programme and Obtained results:

- Developed a game semantics (a CCB), in which the intermediate moves are *not a priori hidden*.
- Formulated an operation which *a posteriori* hides all the intermediate moves step-by-step.

Brief Summary of the Talk

Background and motivation:

- The existing game semantics *a priori* "hides" all the "intermediate moves" when strategies are composed.
- Thus, it is "static" and "not fully intentional".

Research Programme and Obtained results:

- Developed a game semantics (a CCB), in which the intermediate moves are *not a priori hidden*.
- Formulated an operation which *a posteriori* hides all the intermediate moves step-by-step.

Ongoing works:

- Try to establish the *dynamic correspondence property*.
- Study a connection with *propositional equality* in Martin-Löf type theory.

What Is Game Semantics?

What Is Game Semantics?

- *Game semantics* (for programming languages (p.l.)) refers to a particular kind of (denotational) semantics of p.l., in which types T and terms $t : T$ are interpreted as "games" $[\![T]\!]$ and "strategies" $[\![t]\!]$ on the game $[\![T]\!]$.

What Is Game Semantics?

- *Game semantics* (for programming languages (p.l.)) refers to a particular kind of (denotational) semantics of p.l., in which types T and terms $t : T$ are interpreted as "games" $[\![T]\!]$ and "strategies" $[\![t]\!]$ on the game $[\![T]\!]$.
- In the literature, it gave the first *syntax-independent* characterization of the p.l. PCF; since then, a variety of games and strategies have been proposed to characterize various programming features.

What Is Game Semantics?

- *Game semantics* (for programming languages (p.l.)) refers to a particular kind of (denotational) semantics of p.l., in which types T and terms $t : T$ are interpreted as "games" $[\![T]\!]$ and "strategies" $[\![t]\!]$ on the game $[\![T]\!]$.
- In the literature, it gave the first *syntax-independent* characterization of the p.l. PCF; since then, a variety of games and strategies have been proposed to characterize various programming features.
- Below, we informally explain the general idea of games and strategies.

Games

In game semantics, a *game* is a certain forest, in which each branch corresponds to a play (development) of the game.

Games

In game semantics, a *game* is a certain forest, in which each branch corresponds to a play (development) of the game.

- The forest represents all the possible developments of the game.

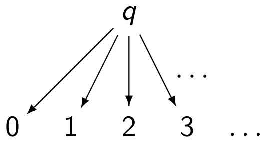
Games

In game semantics, a *game* is a certain forest, in which each branch corresponds to a play (development) of the game.

- The forest represents all the possible developments of the game.
- We usually consider a game played by two players, Player (P) and Opponent (O), in which Opponent always starts a play, and they alternatively make an allowed *move* at each turn.

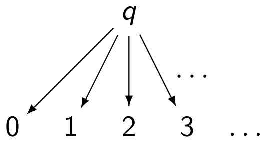
Example: Natural Number Game N

For example, the game N of natural numbers is the following tree:



Example: Natural Number Game N

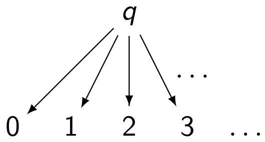
For example, the game N of natural numbers is the following tree:



Thus, a typical play of the game N consists of:

Example: Natural Number Game N

For example, the game N of natural numbers is the following tree:

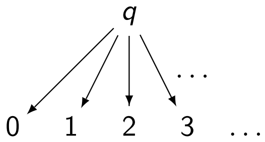


Thus, a typical play of the game N consists of:

1. O's question q ("What's your number?")

Example: Natural Number Game N

For example, the game N of natural numbers is the following tree:



Thus, a typical play of the game N consists of:

1. O's question q ("What's your number?")
2. P's answer n ("My number is n.")

Example: Function Game $N \rightarrow N$

As another example, a typical play of the game $N \rightarrow N$ of the function type is

$$q_o . q_i . n . m$$

Example: Function Game $N \to N$

As another example, a typical play of the game $N \to N$ of the function type is

$$q_o.q_i.n.m$$

This play consists of:

Example: Function Game $N \rightarrow N$

As another example, a typical play of the game $N \rightarrow N$ of the function type is

$$q_o.q_i.n.m$$

This play consists of:

1. O's question q_o ("What's your output?")

Example: Function Game $N \rightarrow N$

As another example, a typical play of the game $N \rightarrow N$ of the function type is

$$q_o.q_i.n.m$$

This play consists of:

1. O's question q_o ("What's your output?")
2. P's question q_i ("Wait, what's your input?")

Example: Function Game $N \to N$

As another example, a typical play of the game $N \to N$ of the function type is

$$q_o.q_i.n.m$$

This play consists of:

1. O's question q_o ("What's your output?")
2. P's question q_i ("Wait, what's your input?")
3. O's answer n to q_i ("OK, here is the input n for you.")

Example: Function Game $N \to N$

As another example, a typical play of the game $N \to N$ of the function type is

$$q_o.q_i.n.m$$

This play consists of:

1. O's question q_o ("What's your output?")
2. P's question q_i ("Wait, what's your input?")
3. O's answer n to q_i ("OK, here is the input n for you.")
4. P's answer m to q_o ("Alright, the output is m.")

Strategies

A *strategy* on a game G is a partial function

$$\sigma : \text{odd-length play } s \text{ of } G \mapsto \text{P's move } m$$

such that the concatenation $s.m$ is again a play of G.

Strategies

A *strategy* on a game G is a partial function

$$\sigma : \text{odd-length play } s \text{ of } G \mapsto \text{P's move } m$$

such that the concatenation $s.m$ is again a play of G.

- That is, a strategy is what tells Player which move she should make next in a deterministic way.

Strategies

A *strategy* on a game G is a partial function

$$\sigma : \text{odd-length play } s \text{ of } G \mapsto \text{P's move } m$$

such that the concatenation $s.m$ is again a play of G.

- That is, a strategy is what tells Player which move she should make next in a deterministic way.
- E.g., a strategy of a natural number $n \in \mathbb{N}$ is defined by $q \mapsto n$; and a strategy of the successor function is defined by $q_o \mapsto q_i$ and $q_o.q_i.n \mapsto n+1$.

Composition in Existing Game Semantics

In the existing game semantics, a composition $\sigma ; \tau : A \to C$ of strategies $\sigma : A \to B$ and $\tau : B \to C$ is defined as

Composition in Existing Game Semantics

In the existing game semantics, a composition $\sigma ; \tau : A \to C$ of strategies $\sigma : A \to B$ and $\tau : B \to C$ is defined as

"intermediate communication plus hiding".

Composition in Existing Game Semantics

In the existing game semantics, a composition $\sigma;\tau : A \to C$ of strategies $\sigma : A \to B$ and $\tau : B \to C$ is defined as

"intermediate communication plus hiding".

As an example, let us see how the "doubling strategy" d and the "successor" strategy s is composed:

N	$\stackrel{s}{\to}$	N	N	$\stackrel{d}{\to}$	N
		q_o			q_o
q_i			q_i		
m			n		
		$m+1$			$2n$

Composition in Existing Game Semantics

$$N \xrightarrow{s} N \xrightarrow{d} N$$

		q
	$\boxed{q}$	
q		
n		
	$\boxed{n+1}$	
		$2(n+1)$

Composition in Existing Game Semantics

$$N \xrightarrow{s} N \xrightarrow{d} N$$

N	$\xrightarrow{s}$	N	$\xrightarrow{d}$	N
				q
q				
n				
				$2(n+1)$

Composition in Existing Game Semantics

$$I \xrightarrow{5} N \xrightarrow{s} N \xrightarrow{d} N$$

			q
q			
n			
			$2(n+1)$

Composition in Existing Game Semantics

$$I \xrightarrow{5} N \xrightarrow{s} N \xrightarrow{d} N$$

			q
$\boxed{q}$			
$\boxed{5}$			
			12

Composition in Existing Game Semantics

$$I \quad \stackrel{5}{\rightarrow} \quad N \quad \stackrel{s}{\rightarrow} \quad N \quad \stackrel{d}{\rightarrow} \quad N$$

q

12

Composition in Existing Game Semantics

$$
\begin{array}{ccccccc}
I & \stackrel{5}{\rightarrow} & N & \stackrel{s}{\rightarrow} & N & \stackrel{d}{\rightarrow} & N \\
 & & & & & & q \\
 & & & & & & \\
 & & & & & & 12
\end{array}
$$

In syntax, this corresponds to:

Composition in Existing Game Semantics

$$I \xrightarrow{5} N \xrightarrow{s} N \xrightarrow{d} N$$

$$q$$

$$12$$

In syntax, this corresponds to:

$$[\lambda x. (\lambda y. \underline{2} \cdot y)\{(\lambda z. z + \underline{1})\, x\}]\, \underline{5}$$

Composition in Existing Game Semantics

$$I \xrightarrow{5} N \xrightarrow{s} N \xrightarrow{d} N$$

q

12

In syntax, this corresponds to:

$$[\lambda x.\,(\lambda y.\,\underline{2}\cdot y)\{(\lambda z.\,z+\underline{1})\,x\}]\,\underline{5} \rightarrow^{*} [\lambda x.\,\underline{2}\cdot(x+\underline{1})]\,\underline{5}$$

Composition in Existing Game Semantics

$$I \quad \stackrel{5}{\rightarrow} \quad N \quad \stackrel{s}{\rightarrow} \quad N \quad \stackrel{d}{\rightarrow} \quad N$$

q

12

In syntax, this corresponds to:

$$[\lambda x. (\lambda y. \underline{2} \cdot y)\{(\lambda z. z + \underline{1})\, x\}]\, \underline{5} \rightarrow^{*} [\lambda x. \underline{2} \cdot (x + \underline{1})]\, \underline{5}$$
$$\rightarrow \underline{2} \cdot (\underline{5} + \underline{1})$$

Composition in Existing Game Semantics

$$I \xrightarrow{5} N \xrightarrow{s} N \xrightarrow{d} N$$

$$q$$

$$12$$

In syntax, this corresponds to:

$$\begin{aligned}[\lambda x.\,(\lambda y.\,\underline{2}\cdot y)\{(\lambda z.\,z+\underline{1})\,x\}]\,\underline{5} &\rightarrow^* [\lambda x.\,\underline{2}\cdot(x+\underline{1})]\,\underline{5} \\ &\rightarrow \underline{2}\cdot(\underline{5}+\underline{1}) \\ &\rightarrow^* \underline{12}\end{aligned}$$

Existing Game Semantics Is Static

From the example, we make the following observations:

Existing Game Semantics Is Static

From the example, we make the following observations:

Observation 1

A reduction in syntax corresponds to *"hiding intermediate moves"* in game semantics.

Existing Game Semantics Is Static

From the example, we make the following observations:

Observation 1

A reduction in syntax corresponds to *"hiding intermediate moves"* in game semantics.

Observation 2

However, this process of hiding is *a priori* executed in the existing game semantics; thus, strategies are always "in normal form".

Existing Game Semantics Is Static

From the example, we make the following observations:

Observation 1

A reduction in syntax corresponds to *“hiding intermediate moves”* in game semantics.

Observation 2

However, this process of hiding is *a priori* executed in the existing game semantics; thus, strategies are always “in normal form”.

Therefore, the existing game semantics is *“static”* in the following sense: If we have a sequence $t_1 \rightarrow^* t_2$ of reductions in syntax, then we just have the equality $[\![t_1]\!] = [\![t_2]\!]$ between strategies.

Existing Game Semantics Is Not Fully Intensional

Observation 3

The "hidden moves" constitute an important part of *intensionality* (or algorithms) in computation. However, they are not explicitly captured in the existing game semantics.

Existing Game Semantics Is Not Fully Intensional

Observation 3

The "hidden moves" constitute an important part of *intensionality* (or algorithms) in computation. However, they are not explicitly captured in the existing game semantics.

To make this point clearer, consider the following two programs:

$$f \stackrel{\mathrm{df.}}{=} \lambda x. \, [(\lambda y. \, \underline{2} \cdot y)\{(\lambda z. \, z + \underline{1}) \, x\}]$$
$$g \stackrel{\mathrm{df.}}{=} \lambda x. \, [(\lambda y. \, y + \underline{1})[(\lambda z. \, z + \underline{1})\{(\lambda w. \, \underline{2} \cdot w) \, x\}]]$$

Existing Game Semantics Is Not Fully Intensional

Observation 3

The "hidden moves" constitute an important part of *intensionality* (or algorithms) in computation. However, they are not explicitly captured in the existing game semantics.

To make this point clearer, consider the following two programs:

$$f \stackrel{\mathrm{df.}}{=} \lambda x. [(\lambda y. \underline{2} \cdot y)\{(\lambda z. z + \underline{1})\, x\}]$$
$$g \stackrel{\mathrm{df.}}{=} \lambda x. [(\lambda y. y + \underline{1})[(\lambda z. z + \underline{1})\{(\lambda w. \underline{2} \cdot w)\, x\}]]$$

Clearly, they denote extensionally equal but intensionally different computations. In the existing game semantics, the intensionality is not fully captured and the two functions are interpreted as equal:

N	$\stackrel{s}{\rightarrow}$	N	$\stackrel{d}{\rightarrow}$	N
				q
		$\boxed{q}$		
q				
n				
		$\boxed{n+1}$		
				$2(n+1)$

N	$\stackrel{d}{\rightarrow}$	N	$\stackrel{s}{\rightarrow}$	N	$\stackrel{s}{\rightarrow}$	N
						q
				$\boxed{q}$		
		$\boxed{q}$				
q						
n						
		$\boxed{2n}$				
				$\boxed{2n+1}$		
						$2n+2$

Existing Game Semantics Is "Static" and "Not Fully Intensional"

$N \xrightarrow{s} N \xrightarrow{d} N$

		q
q		
n		

$2(n+1)$

$N \xrightarrow{d} N \xrightarrow{s} N \xrightarrow{s} N$

			q
q			
n			

$2n+2$

Programme of Research

Motivated by the observations, we propose the following programme of research:

Programme of Research

Motivated by the observations, we propose the following programme of research:

1. In the existing (HO-) games and strategies, we introduce a distinction between *"external"* and *"internal"* moves and some axioms; internal moves are to be *a posteriori* hidden.

Programme of Research

Motivated by the observations, we propose the following programme of research:

1. In the existing (HO-) games and strategies, we introduce a distinction between *"external"* and *"internal"* moves and some axioms; internal moves are to be *a posteriori* hidden.
2. We then formulate the *hiding operation*, which "hides" all the internal moves step-by-step, transforming strategies into a "normal form". The resulting structure is called *"dynamic" games and strategies*.

Programme of Research

Motivated by the observations, we propose the following programme of research:

1. In the existing (HO-) games and strategies, we introduce a distinction between *"external"* and *"internal"* moves and some axioms; internal moves are to be *a posteriori* hidden.
2. We then formulate the *hiding operation*, which "hides" all the internal moves step-by-step, transforming strategies into a "normal form". The resulting structure is called *"dynamic" games and strategies*.
3. Next, we study the categorical structure thereof, so that we may interpret p.l. in it (*"dynamic" game semantics*).

Programme of Research

Motivated by the observations, we propose the following programme of research:

1. In the existing (HO-) games and strategies, we introduce a distinction between *"external"* and *"internal"* moves and some axioms; internal moves are to be *a posteriori* hidden.
2. We then formulate the *hiding operation*, which "hides" all the internal moves step-by-step, transforming strategies into a "normal form". The resulting structure is called *"dynamic" games and strategies*.
3. Next, we study the categorical structure thereof, so that we may interpret p.l. in it (*"dynamic" game semantics*).
4. We then characterize the syntactic notion of reduction in dynamic game semantics, by establishing:

Dynamic Correspondence Property

Let $\mathcal{L}$ be an appropriate p.l. with a small-step operational semantics $\rightarrow$, and $[\![\_]\!]$ the interpretation of $\mathcal{L}$ in dynamic game semantics with the hiding operation $\blacktriangleright$.

Dynamic Correspondence Property

Let $\mathcal{L}$ be an appropriate p.l. with a small-step operational semantics $\rightarrow$, and $[\![\_]\!]$ the interpretation of $\mathcal{L}$ in dynamic game semantics with the hiding operation $\blacktriangleright$.

Conjecture (Dynamic correspondence property)

For any terms t_1, t_2 in the language $\mathcal{L}$, if we have a single-step reduction $t_1 \rightarrow t_2$, then the following diagram commutes:

$$\begin{array}{ccc} t_1 & \rightarrow & t_2 \\ {\scriptstyle [\![\_]\!]} \downarrow & & \downarrow {\scriptstyle [\![\_]\!]} \\ [\![t_1]\!] & \blacktriangleright & [\![t_2]\!] \end{array}$$

where the dotted arrows represent the interpretation $[\![\_]\!]$.

Results Obtained So Far

Results Obtained So Far

So far, we have:

- Established a cartesian closed bicategory $\mathcal{CCD}$ of dynamic games and strategies, which is a generalization of the CCC of HO-games and strategies.
- Formulated two algorithms to execute the hiding operation.

Results Obtained So Far

So far, we have:

- Established a cartesian closed bicategory $\mathcal{CCD}$ of dynamic games and strategies, which is a generalization of the CCC of HO-games and strategies.
- Formulated two algorithms to execute the hiding operation.

Thus, it remains to:

- Give an interpretation of an appropriate p.l. in $\mathcal{CCD}$.
- Establish the dynamic correspondence property.

Results Obtained So Far

So far, we have:

- Established a cartesian closed bicategory $\mathcal{CCD}$ of dynamic games and strategies, which is a generalization of the CCC of HO-games and strategies.
- Formulated two algorithms to execute the hiding operation.

Thus, it remains to:

- Give an interpretation of an appropriate p.l. in $\mathcal{CCD}$.
- Establish the dynamic correspondence property.

In the rest of this talk, we sketch the bicategory $\mathcal{D}$ of dynamic games and strategies, which is similar to but simpler than $\mathcal{CCD}$.

Why Bicategory?

But wait, why is the structure a bicategory (not a category)?

Why Bicategory?

But wait, why is the structure a bicategory (not a category)?

- It is mainly because we cannot have strict identities as long as we do not a priori hide internal moves.

Why Bicategory?

But wait, why is the structure a bicategory (not a category)?

- It is mainly because we cannot have strict identities as long as we do not a priori hide internal moves.
- More concretely: In game semantics, identities are usually the so-called *copy-cat strategies* cp. But then...

Why Bicategory?

But wait, why is the structure a bicategory (not a category)?

- It is mainly because we cannot have strict identities as long as we do not a priori hide internal moves.
- More concretely: In game semantics, identities are usually the so-called *copy-cat strategies* cp. But then...

N	$\stackrel{\sigma}{\rightarrow}$	N	$\stackrel{\text{cp}}{\rightarrow}$	N
				q
		$\boxed{q}$		
q				
n				
		$\boxed{m}$		
				m

Why Bicategory?

But wait, why is the structure a bicategory (not a category)?

- It is mainly because we cannot have strict identities as long as we do not a priori hide internal moves.
- More concretely: In game semantics, identities are usually the so-called *copy-cat strategies* cp. But then...

$$N \xrightarrow{\sigma} N \xrightarrow{\mathrm{cp}} N$$

		q
q		
n		
		m

Why Bicategory?

- We may take the *"external equality"* $\approx$ as the equality in the category; but then it would be essentially the existing category of (HO-) games and strategies.

Why Bicategory?

- We may take the *"external equality"* $\approx$ as the equality in the category; but then it would be essentially the existing category of (HO-) games and strategies.

Solution

We form a bicategory, in which the existence of a 2-cell $\alpha : \sigma \Rightarrow \tau$ is equivalent to the external equality $\sigma \approx \tau$.

Why Bicategory?

- We may take the *"external equality"* $\approx$ as the equality in the category; but then it would be essentially the existing category of (HO-) games and strategies.

Solution

We form a bicategory, in which the existence of a 2-cell $\alpha : \sigma \Rightarrow \tau$ is equivalent to the external equality $\sigma \approx \tau$.

Note: Such 2-cells "witness" the external equality; thus we may interpret *propositional equality* in Martin-Löf type theory by the 2-cells (future work).

The Bicategory $\mathcal{D}$

Definition (The bicategory $\mathcal{D}$ of dynamic games and strategies)

The bicategory $\mathcal{D}$ of dynamic games and strategies is given by:

- **0-cells.** 0-cells are *well-opened* and *explicit*, dynamic games.
- **1-cells.** A 1-cell $\sigma : A \to B$ is a dynamic strategy σ such that $\mathcal{H}(\sigma) : A \multimap B$, where $\mathcal{H}(\_) = (\_)^h = \blacktriangleright$ is the hiding operation.
- **2-cells.** A 2-cell $\alpha : \sigma \Rightarrow \tau$ is the copy-cat strategy $\mathrm{cp}_{\overline{\sigma^h}} = \mathrm{cp}_{\overline{\tau^h}} : \overline{\sigma^h} \multimap \overline{\tau^h}$ if $\sigma \approx \tau$ (otherwise, there is no 2-cell between σ and τ), where $\sigma \approx \tau \overset{\mathrm{df.}}{\Leftrightarrow} \mathcal{H}(\sigma) = \mathcal{H}(\tau)$.

The Bicategory $\mathcal{D}$

Definition (The Bicategory $\mathcal{D}$ (Continued))

- **Vertical composition.** The "vertical" composition on 2-cells is the *standard (with hiding) composition* ";" of strategies.
- **Vertical identity.** The identity id_σ on each 1-cell σ w.r.t. the vertical composition is the copy-cat strategy $\mathrm{cp}_{\overline{\sigma^h}} : \overline{\sigma^h} \multimap \overline{\sigma^h}$.
- **Horizontal compositions.** The "horizontal" composition on 1-cells is the *"non-hiding" composition* "$\ddagger$", and the one on 2-cells is the *parallel product* "$\downdownarrows$".
- **Horizontal pre-identity.** The "pre"-identity id_A on each 0-cell A w.r.t. the horizontal composition on 1-cells is the copy-cat strategy $\mathrm{cp}_A : A \multimap A$.
- **Natural iso.** The components are the copy-cat strategies.

The Bicategory $\mathcal{D}$

Notes:

The Bicategory $\mathcal{D}$

Notes:

- A 1-cell $\sigma : A \to B$ can be seen as an "algorithm" whose "external behavior" $\mathcal{H}(\sigma)$ is a function $A \multimap B$.

The Bicategory $\mathcal{D}$

Notes:

- A 1-cell $\sigma : A \to B$ can be seen as an "algorithm" whose "external behavior" $\mathcal{H}(\sigma)$ is a function $A \multimap B$.
- Clearly, $\mathcal{D}(A, B)(\sigma, \tau) \neq \emptyset \Leftrightarrow \sigma \approx \tau$.

The Bicategory $\mathcal{D}$

Notes:

- A 1-cell $\sigma : A \to B$ can be seen as an "algorithm" whose "external behavior" $\mathcal{H}(\sigma)$ is a function $A \multimap B$.
- Clearly, $\mathcal{D}(A, B)(\sigma, \tau) \neq \emptyset \Leftrightarrow \sigma \approx \tau$.
- We may have *nontrivial 2-cells* that are externally equal to the copy-cat strategies (future work), which corresponds to *nontrivial paths* in HoTT.

The Bicategory $\mathcal{D}$

Notes:

- A 1-cell $\sigma : A \to B$ can be seen as an "algorithm" whose "external behavior" $\mathcal{H}(\sigma)$ is a function $A \multimap B$.
- Clearly, $\mathcal{D}(A, B)(\sigma, \tau) \neq \emptyset \Leftrightarrow \sigma \approx \tau$.
- We may have *nontrivial 2-cells* that are externally equal to the copy-cat strategies (future work), which corresponds to *nontrivial paths* in HoTT.
- Of course, we may generalize the bicategory $\mathcal{D}$ to a tricategory, etc., but the bicategory appears an appropriate structure.

Ongoing Works

Finally, we summarize some of the ongoing works:

Ongoing Works

Finally, we summarize some of the ongoing works:

- Find an appropriate p.l., give an interpretation for it in $\mathcal{CCD}$, and establish the *dynamic correspondence property*, in which compare the two algorithms for the hiding operation (different reduction strategies?).

Ongoing Works

Finally, we summarize some of the ongoing works:

- Find an appropriate p.l., give an interpretation for it in $\mathcal{CCD}$, and establish the *dynamic correspondence property*, in which compare the two algorithms for the hiding operation (different reduction strategies?).
- By axiomatizing elementary steps, develop $\mathcal{CCD}$ to a mathematical model of computation (like TMs), in which *intensionality* in computation is formulated. E.g., it may be useful as a computational complexity measure.

Ongoing Works

Finally, we summarize some of the ongoing works:

- Find an appropriate p.l., give an interpretation for it in $\mathcal{CCD}$, and establish the *dynamic correspondence property*, in which compare the two algorithms for the hiding operation (different reduction strategies?).
- By axiomatizing elementary steps, develop $\mathcal{CCD}$ to a mathematical model of computation (like TMs), in which *intensionality* in computation is formulated. E.g., it may be useful as a computational complexity measure.
- Investigate the connection between the external equality and the propositional equality in HoTT.

Thank you!