Hypercoherences as games for space-efficient iterations?

Nguyễn Lê Thành Dũng (a.k.a. Tito) — nltd@nguyentito.eu
Laboratoire d'informatique de Paris Nord, Villetaneuse, France
partially inspired by joint work with Pierre Pradic (University of Oxford)
Concurrent Games Café, January 29th, 2021

Some motivations from implicit complexity (1)

At the beginning of this enterprise I wanted to prove “concrete” statements like this:

Typical theorem in implicit computational complexity

A function can be computed by some program of type T in a language P
if and only if it belongs to the complexity class $\mathcal{C}$.

There is (was?) a whole industry doing this with $P =$ something involving *linear* types

Some motivations from implicit complexity (1)

At the beginning of this enterprise I wanted to prove "concrete" statements like this:

Typical theorem in implicit computational complexity

A function can be computed by some program of type T in a language P
if and only if it belongs to the complexity class $\mathcal{C}$.

There is (was?) a whole industry doing this with $P =$ something involving *linear* types

An example dear to my heart: $P =$ simply typed λ-calculus, $\mathcal{C} =$ regular languages

Theorem (Hillebrand & Kanellakis 1996)

For any type A and any simply typed *λ-term $t : \mathtt{Str}_\Sigma[A] \to \mathtt{Bool}$ (using* Church encodings*), the language $\{w \in \Sigma^* \mid t\,\overline{w} =_\beta \mathtt{true}\}$ is* regular. *Conversely, every regular language can be defined this way.*

(see also my *Implicit automata in typed λ-calculi* paper series with Pradic)

Some motivations from implicit complexity (1)

At the beginning of this enterprise I wanted to prove "concrete" statements like this:

Typical theorem in implicit computational complexity

A function can be computed by some program of type T in a language P
if and only if it belongs to the complexity class $\mathcal{C}$.

There is (was?) a whole industry doing this with $P =$ something involving *linear* types

An example dear to my heart: $P =$ simply typed λ-calculus, $\mathcal{C} =$ regular languages

Theorem (Hillebrand & Kanellakis 1996)

For any type A and any simply typed *λ-term $t : \mathtt{Str}_\Sigma[A] \to \mathtt{Bool}$ (using* Church encodings*), the language $\{w \in \Sigma^* \mid t\,\overline{w} =_\beta \mathtt{true}\}$ is* regular. *Conversely, every regular language can be defined this way.*

(see also my *Implicit automata in typed λ-calculi* paper series with Pradic)

Proof idea: compute $[\![t\,\overline{w}]\!]$ in the cartesian closed category of finite sets
$\longrightarrow$ *semantic evaluation* technique, makes denotational models relevant!

Some motivations from implicit complexity (2)

Typical theorem in implicit computational complexity

A function can be computed by some program of type T in a language P
if and only if it belongs to the complexity class $\mathcal{C}$.

N. & Pradic, *From normal functors to logarithmic space queries* (sorry for the clickbait), 2019:

- $P =$ something involving linear types (more or less Elementary Linear Logic)
- $T =$ somewhat less conventional choice (doesn't matter here)
- partial results: $\mathsf{L} \subseteq \mathcal{C} \subseteq \mathsf{NL}$, upper bound obtained using *coherence spaces*
  conjecture: $\mathcal{C} = \mathsf{L}$

Some motivations from implicit complexity (2)

Typical theorem in implicit computational complexity

A function can be computed by some program of type T in a language P
if and only if it belongs to the complexity class $\mathcal{C}$.

N. & Pradic, *From normal functors to logarithmic space queries* (sorry for the clickbait), 2019:

- $P =$ something involving linear types (more or less Elementary Linear Logic)
- $T =$ somewhat less conventional choice (doesn't matter here)
- partial results: $\mathsf{L} \subseteq \mathcal{C} \subseteq \mathsf{NL}$, upper bound obtained using *coherence spaces*
  conjecture: $\mathcal{C} = \mathsf{L}$

This talk: sketch of a few ideas to make a tiny bit of progress on this conjecture,
involving *hypercoherences*, with some intuitions from game semantics

First I have to recall hypercoherences + their connection with games from:
Ehrhard, *Parallel and serial hypercoherences*, 2000

Hypercoherences in a nutshell

A hypercoherence X := a set $|X|$ + choice of *coherent* subsets $\Gamma(X) \subset \mathcal{P}_{\mathrm{fin}}(|X|) \setminus \{\varnothing\}$,
containing all singletons ($\mathcal{P}_{\mathrm{fin}}(S)$ = *finite* subsets of S)
strictly coherent := coherent & non-singleton, *strictly incoherent* := $\mathcal{P}_{\mathrm{fin}}(|X|) \setminus (\Gamma(X) \cup \{\varnothing\})$

Hypercoherences in a nutshell

A hypercoherence X := a set $|X|$ + choice of *coherent* subsets $\Gamma(X) \subset \mathcal{P}_{\text{fin}}(|X|) \setminus \{\varnothing\}$,
containing all singletons ($\mathcal{P}_{\text{fin}}(S)$ = *finite* subsets of S)
strictly coherent := coherent & non-singleton, *strictly incoherent* := $\mathcal{P}_{\text{fin}}(|X|) \setminus (\Gamma(X) \cup \{\varnothing\})$

- linear negation: $|X^{\perp}| = |X|$, exchange coherence and incoherence
- $|X \multimap Y| = |X| \times |Y|$ and $\Gamma(X \multimap Y)$ to be defined later

Hypercoherences in a nutshell

A hypercoherence X := a set $|X|$ + choice of *coherent* subsets $\Gamma(X) \subset \mathcal{P}_{\text{fin}}(|X|) \setminus \{\varnothing\}$,
containing all singletons ($\mathcal{P}_{\text{fin}}(S)$ = *finite* subsets of S)
strictly coherent := coherent & non-singleton, *strictly incoherent* := $\mathcal{P}_{\text{fin}}(|X|) \setminus (\Gamma(X) \cup \{\varnothing\})$

- linear negation: $|X^{\perp}| = |X|$, exchange coherence and incoherence
- $|X \multimap Y| = |X| \times |Y|$ and $\Gamma(X \multimap Y)$ to be defined later

Cliques $c \sqsubset X$ (semantic inhabitants): $c \subseteq |X|$ and $\mathcal{P}_{\text{fin}}(c) \setminus \{\varnothing\} \subseteq \Gamma(X)$
Morphism $X \to Y$:= clique of $X \multimap Y$, composed by relational composition (thm: it works)

Hypercoherences in a nutshell

A hypercoherence X := a set $|X|$ + choice of *coherent* subsets $\Gamma(X) \subset \mathcal{P}_{\text{fin}}(|X|) \setminus \{\varnothing\}$,
containing all singletons ($\mathcal{P}_{\text{fin}}(S)$ = *finite* subsets of S)
strictly coherent := coherent & non-singleton, *strictly incoherent* := $\mathcal{P}_{\text{fin}}(|X|) \setminus (\Gamma(X) \cup \{\varnothing\})$

- linear negation: $|X^\perp| = |X|$, exchange coherence and incoherence
- $|X \multimap Y| = |X| \times |Y|$ and $\Gamma(X \multimap Y)$ to be defined later

Cliques $c \sqsubset X$ (semantic inhabitants): $c \subseteq |X|$ and $\mathcal{P}_{\text{fin}}(c) \setminus \{\varnothing\} \subseteq \Gamma(X)$
Morphism $X \to Y$:= clique of $X \multimap Y$, composed by relational composition (thm: it works)

For simplicity we will consider mostly *finite* hypercoherences ($\mathrm{Card}(|X|) < \infty$)

The iteration problem

In our implicit complexity stuff, the bottleneck for the semantic evaluation argument is:

Decision problem

Inputs: a finite hypercoherence X, 2 points $x, y \in |X|$, a list of endormophisms $c_1, \ldots, c_n \sqsubset X \multimap X$.
Output: are x and y related by $c_n \circ \cdots \circ c_1$? (yes/no)

So what's the complexity of this?

The iteration problem

In our implicit complexity stuff, the bottleneck for the semantic evaluation argument is:

Decision problem

Inputs: a finite hypercoherence X, 2 points $x, y \in |X|$, a list of endormophisms $c_1, \ldots, c_n \sqsubset X \multimap X$.
Output: are x and y related by $c_n \circ \cdots \circ c_1$? (yes/no)

So what's the complexity of this?

- NL: guess $x = z_0, z_1, \ldots, z_n = y$ such that $(z_{i-1}, z_i) \in c_i$

The iteration problem

In our implicit complexity stuff, the bottleneck for the semantic evaluation argument is:

Decision problem

Inputs: a finite hypercoherence X, 2 points $x, y \in |X|$, a list of endormophisms $c_1, \ldots, c_n \sqsubset X \multimap X$.
Output: are x and y related by $c_n \circ \cdots \circ c_1$? (yes/no)

So what's the complexity of this?

- NL: guess $x = z_0, z_1, \ldots, z_n = y$ such that $(z_{i-1}, z_i) \in c_i$
- UL: the non-determinism is *unambiguous* since this sequence is *unique* if it exists, thanks to:

Elementary property (related to Berry's stability)

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^\perp$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Proof. $\mathcal{P}_{\mathrm{fin}}(c \cap d) \setminus \{\varnothing\} \subseteq \Gamma(X) \cap \Gamma(X^\perp) = \{\text{singletons of } |X|\}$. $\square$

In our implicit complexity stuff, the bottleneck for the semantic evaluation argument is:

Decision problem

Inputs: a finite hypercoherence X, 2 points $x, y \in |X|$, a list of endormophisms $c_1, \ldots, c_n \sqsubset X \multimap X$.
Output: are x and y related by $c_n \circ \cdots \circ c_1$? (yes/no)

So what's the complexity of this?

- NL: guess $x = z_0, z_1, \ldots, z_n = y$ such that $(z_{i-1}, z_i) \in c_i$
- UL: the non-determinism is *unambiguous* since this sequence is *unique* if it exists, thanks to:

Elementary property (related to Berry's stability)

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^\perp$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Proof. $\mathcal{P}_{\mathrm{fin}}(c \cap d) \setminus \{\varnothing\} \subseteq \Gamma(X) \cap \Gamma(X^\perp) = \{\text{singletons of } |X|\}$. $\square$

We want to use games to do better (L) for restricted versions of the problem.

Hypercoherences as games (1)

Elementary property

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^\perp$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Morally the final position resulting from an interaction: strategy c vs counter-strategy d.

Hypercoherences as games (1)

Elementary property

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^\perp$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Morally the final position resulting from an interaction: strategy c vs counter-strategy d.

Intuition: coherent = $\ominus$, incoherent = $\oplus$. Assume $|X| \notin \Gamma(X)$ i.e. c plays first.

- Take $S_1 \in \Gamma(X)$ such that $c \subseteq S_1$. Let $d_1 = d \cap S_1$.

Hypercoherences as games (1)

Elementary property

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^{\perp}$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Morally the final position resulting from an interaction: strategy c vs counter-strategy d.

Intuition: coherent = $\ominus$, incoherent = $\oplus$. Assume $|X| \notin \Gamma(X)$ i.e. c plays first.

- Take $S_1 \in \Gamma(X)$ such that $c \subseteq S_1$. Let $d_1 = d \cap S_1$.
- Take $S_2 \in \Gamma(X^{\perp})$ such that $d_1 \subseteq S_2 \subseteq S_1$. Let $c_2 = c \cap S_2$.

Hypercoherences as games (1)

Elementary property

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^\perp$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Morally the final position resulting from an interaction: strategy c vs counter-strategy d.

Intuition: coherent = $\ominus$, incoherent = $\oplus$. Assume $|X| \notin \Gamma(X)$ i.e. c plays first.

- Take $S_1 \in \Gamma(X)$ such that $c \subseteq S_1$. Let $d_1 = d \cap S_1$.
- Take $S_2 \in \Gamma(X^\perp)$ such that $d_1 \subseteq S_2 \subseteq S_1$. Let $c_2 = c \cap S_2$.
- Take $S_3 \in \Gamma(X)$ such that $c_2 \subseteq S_3 \subseteq S_2$. Let $d_3 = d_1 \cap S_3$.

Elementary property

Let X be a hypercoherence. For $c \sqsubset X$ and $d \sqsubset X^\perp$, we have $\mathrm{Card}(c \cap d) \leq 1$.

Morally the final position resulting from an interaction: strategy c vs counter-strategy d.

Intuition: coherent = $\ominus$, incoherent = $\oplus$. Assume $|X| \notin \Gamma(X)$ i.e. c plays first.

- Take $S_1 \in \Gamma(X)$ such that $c \subseteq S_1$. Let $d_1 = d \cap S_1$.
- Take $S_2 \in \Gamma(X^\perp)$ such that $d_1 \subseteq S_2 \subseteq S_1$. Let $c_2 = c \cap S_2$.
- Take $S_3 \in \Gamma(X)$ such that $c_2 \subseteq S_3 \subseteq S_2$. Let $d_3 = d_1 \cap S_3$.
- etc.

For n large enough, this fails ($c \cap d = \varnothing$) or is equal to $c \cap d$.

Hypercoherences as games (2)

Ehrhard calls a *tower* of X any sequence of alternating polarities (coh/incoh)

$$|X| = S_0 \supsetneq S_1 \supsetneq \cdots \supsetneq S_n$$

where each S_i is a *maximal* subset of the right polarity of S_{i-1}.
Towers are *plays*, and their elements are *positions*. **Depth** of X := maximum possible n.

Hypercoherences as games (2)

Ehrhard calls a *tower* of X any sequence of alternating polarities (coh/incoh)

$$|X| = S_0 \supsetneq S_1 \supsetneq \cdots \supsetneq S_n$$

where each S_i is a *maximal* subset of the right polarity of S_{i-1}.
Towers are *plays*, and their elements are *positions*. **Depth** of $X :=$ maximum possible n.

Conjecture (which would suffice for our implicit complexity purposes)

The iteration problem becomes easier for finite hypercoherences *of bounded depth*.

Hypercoherences as games (2)

Ehrhard calls a *tower* of X any sequence of alternating polarities (coh/incoh)

$$|X| = S_0 \supsetneq S_1 \supsetneq \cdots \supsetneq S_n$$

where each S_i is a *maximal* subset of the right polarity of S_{i-1}.
Towers are *plays*, and their elements are *positions*. **Depth** of $X :=$ maximum possible n.

Conjecture (which would suffice for our implicit complexity purposes)

The iteration problem becomes easier for finite hypercoherences *of bounded depth*.

But wait, we didn't define implication yet... as usual $X \multimap Y = X^\perp \parr Y$
Definition: $S \subseteq |X| \times |Y|$ strictly coherent in $X \parr Y \iff \exists i \in \{1,2\} : \pi_i(S)$ strictly coherent
Explanation: Opponent to play in $X \parr Y \iff$ Opponent to play in either X or Y

Hypercoherences as games (2)

Ehrhard calls a *tower* of X any sequence of alternating polarities (coh/incoh)

$$|X| = S_0 \supsetneq S_1 \supsetneq \cdots \supsetneq S_n$$

where each S_i is a *maximal* subset of the right polarity of S_{i-1}.
Towers are *plays*, and their elements are *positions*. **Depth** of $X :=$ maximum possible n.

Conjecture (which would suffice for our implicit complexity purposes)

The iteration problem becomes easier for finite hypercoherences *of bounded depth*.

But wait, we didn't define implication yet... as usual $X \multimap Y = X^\perp \parr Y$
Definition: $S \subseteq |X| \times |Y|$ strictly coherent in $X \parr Y \iff \exists i \in \{1,2\} : \pi_i(S)$ strictly coherent
Explanation: Opponent to play in $X \parr Y \iff$ Opponent to play in either X or Y

Concurrency? For $|X|$ and $|Y|$ incoh $\begin{cases} |X| \times |Y| \supsetneq S_1 \times |Y| \supsetneq S_2 \times |Y| \supsetneq S_2 \times S'_1 \supsetneq S_2 \times S'_2 \\ \\ \end{cases}$

Hypercoherences as games (2)

Ehrhard calls a *tower* of X any sequence of alternating polarities (coh/incoh)

$$|X| = S_0 \supsetneq S_1 \supsetneq \cdots \supsetneq S_n$$

where each S_i is a *maximal* subset of the right polarity of S_{i-1}.
Towers are *plays*, and their elements are *positions*. **Depth** of $X :=$ maximum possible n.

Conjecture (which would suffice for our implicit complexity purposes)

The iteration problem becomes easier for finite hypercoherences *of bounded depth*.

But wait, we didn't define implication yet... as usual $X \multimap Y = X^\perp \parr Y$
Definition: $S \subseteq |X| \times |Y|$ strictly coherent in $X \parr Y \iff \exists i \in \{1,2\} : \pi_i(S)$ strictly coherent
Explanation: Opponent to play in $X \parr Y \iff$ Opponent to play in either X or Y

Concurrency? For $|X|$ and $|Y|$ incoh $\begin{cases} |X| \times |Y| \supsetneq S_1 \times |Y| \supsetneq S_2 \times |Y| \supsetneq S_2 \times S_1' \supsetneq S_2 \times S_2' \\ |X| \times |Y| \supsetneq |X| \times S_1' \supsetneq |X| \times S_2' \supsetneq |X| \supsetneq S_1 \times S_2' \supsetneq S_2 \times S_2' \end{cases}$

Conjecture (with more details)

For all $k \in \mathbb{N}_{\geq 1}$, there is a deterministic algorithm that, given X *of depth* $\leq k$, $x, y \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$, runs in space $O(\log(\mathrm{Card}(|X|)) + \log(n) + \log(\text{number of positions of } X))$ and decides whether $x, y \in c_n \circ \cdots \circ c_1$.

(using a sparse representation of $\Gamma(X)$ by the set of positions)

Theorem: this holds for $k = 3$ (maybe $k = 4$).

The iteration problem at depth 1

Conjecture (with more details)

For all $k \in \mathbb{N}_{\geq 1}$, there is a deterministic algorithm that, given X *of depth* $\leq k$, $x, y \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$, runs in space $O(\log(\mathrm{Card}(|X|)) + \log(n) + \log(\text{number of positions of } X))$ and decides whether $x, y \in c_n \circ \cdots \circ c_1$.

(using a sparse representation of $\Gamma(X)$ by the set of positions)

Theorem: this holds for $k = 3$ (maybe $k = 4$).

What is a depth 1 hypercoherence? Assume w.l.o.g. $|X| \notin \Gamma(X)$
For any $S \in \Gamma(X)$, the tower $|X| \supsetneq S$ is already maximal

Conjecture (with more details)

For all $k \in \mathbb{N}_{\geq 1}$, there is a deterministic algorithm that, given X *of depth* $\leq k$, $x, y \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$, runs in space $O(\log(\mathrm{Card}(|X|)) + \log(n) + \log(\text{number of positions of } X))$ and decides whether $x, y \in c_n \circ \cdots \circ c_1$.

(using a sparse representation of $\Gamma(X)$ by the set of positions)

Theorem: this holds for $k = 3$ (maybe $k = 4$).

What is a depth 1 hypercoherence? Assume w.l.o.g. $|X| \notin \Gamma(X)$, then $\Gamma(X) = \{\text{singletons}\}$
For any $S \in \Gamma(X)$, the tower $|X| \supsetneq S$ is already maximal $\longrightarrow$ S is a singleton

The iteration problem at depth 1

Conjecture (with more details)

For all $k \in \mathbb{N}_{\geq 1}$, there is a deterministic algorithm that, given X *of depth* $\leq k$, $x, y \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$, runs in space $O(\log(\mathrm{Card}(|X|)) + \log(n) + \log(\text{number of positions of } X))$ and decides whether $x, y \in c_n \circ \cdots \circ c_1$.

(using a sparse representation of $\Gamma(X)$ by the set of positions)

Theorem: this holds for $k = 3$ (maybe $k = 4$).

What is a depth 1 hypercoherence? Assume w.l.o.g. $|X| \notin \Gamma(X)$, then $\Gamma(X) = \{\text{singletons}\}$
For any $S \in \Gamma(X)$, the tower $|X| \supsetneq S$ is already maximal $\longrightarrow$ S is a singleton

Cliques of $X \multimap X$ = *partial functions* $f : |X| \rightharpoonup |X|$
Logspace algorithm: compute $z_1 = f(x)$, $z_2 = f(z_1)$, ... and check that $z_n = y$

The iteration problem at depth 2

Assume now that X has depth 2 (and w.l.o.g. $|X| \notin \Gamma(X)$), let $x \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$

- $\pi_2(c_1 \cap (\{x\} \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_1 \in \Gamma(X)$ be a *position* that contains it. (We can store positions in $O(\log(\text{number of positions of } X))$ space, but not $\pi_2(\ldots))$

The iteration problem at depth 2

Assume now that X has depth 2 (and w.l.o.g. $|X| \notin \Gamma(X)$), let $x \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$

- $\pi_2(c_1 \cap (\{x\} \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_1 \in \Gamma(X)$ be a *position* that contains it. (We can store positions in $O(\log(\text{number of positions of } X))$ space, but not $\pi_2(\ldots))$
- $\pi_2(c_2 \cap (P_1 \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_2 \in \Gamma(X)$ be a position that contains it. Morally: once a move has been played on the left of $X \multimap X$, the strategy c_2 has enough information to play on the right

The iteration problem at depth 2

Assume now that X has depth 2 (and w.l.o.g. $|X| \notin \Gamma(X)$), let $x \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$

- $\pi_2(c_1 \cap (\{x\} \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_1 \in \Gamma(X)$ be a *position* that contains it. (We can store positions in $O(\log(\text{number of positions of } X))$ space, but not $\pi_2(\ldots))$
- $\pi_2(c_2 \cap (P_1 \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_2 \in \Gamma(X)$ be a position that contains it. Morally: once a move has been played on the left of $X \multimap X$, the strategy c_2 has enough information to play on the right
- Let $P_3 \supseteq \pi_2(c_2 \cap (P_1 \times |X|))$ such that $P_3 \in \Gamma(X)$ (assuming non-emptiness). Etc.

The iteration problem at depth 2

Assume now that X has depth 2 (and w.l.o.g. $|X| \notin \Gamma(X)$), let $x \in |X|$ and $c_1, \ldots, c_n \sqsubset X \multimap X$

- $\pi_2(c_1 \cap (\{x\} \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_1 \in \Gamma(X)$ be a *position* that contains it. (We can store positions in $O(\log(\text{number of positions of } X))$ space, but not $\pi_2(\ldots))$
- $\pi_2(c_2 \cap (P_1 \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_2 \in \Gamma(X)$ be a position that contains it. Morally: once a move has been played on the left of $X \multimap X$, the strategy c_2 has enough information to play on the right
- Let $P_3 \supseteq \pi_2(c_2 \cap (P_1 \times |X|))$ such that $P_3 \in \Gamma(X)$ (assuming non-emptiness). Etc.

Important: if there exist $x = z_0, \ldots, z_n = y$ with $(z_{i-1}, z_i) \in c_i$, then $z_i \in P_i$
in particular if $\pi_2(\ldots) = \varnothing$ at some point then $(x, y) \notin c_n \circ \cdots \circ c_1$

The iteration problem at depth 2

Assume now that X has depth 2 (and w.l.o.g. $|X| \notin \Gamma(X)$), let $x \in |X|$ and $c_1, \dots, c_n \sqsubset X \multimap X$

- $\pi_2(c_1 \cap (\{x\} \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_1 \in \Gamma(X)$ be a *position* that contains it. (We can store positions in $O(\log(\text{number of positions of } X))$ space, but not $\pi_2(\dots)$)
- $\pi_2(c_2 \cap (P_1 \times |X|)) \in \Gamma(X) \cup \{\varnothing\}$. If non-empty, let $P_2 \in \Gamma(X)$ be a position that contains it. Morally: once a move has been played on the left of $X \multimap X$, the strategy c_2 has enough information to play on the right
- Let $P_3 \supseteq \pi_2(c_2 \cap (P_1 \times |X|))$ such that $P_3 \in \Gamma(X)$ (assuming non-emptiness). Etc.

Important: if there exist $x = z_0, \dots, z_n = y$ with $(z_{i-1}, z_i) \in c_i$, then $z_i \in P_i$
in particular if $\pi_2(\dots) = \varnothing$ at some point then $(x, y) \notin c_n \circ \dots \circ c_1$

This reduces the problem to the *depth 1 case*

$$c'_n \sqsubset X^{\perp}_{\restriction P_n} \multimap X^{\perp}_{\restriction P_{n-1}}, \dots, c'_1 \sqsubset X^{\perp}_{\restriction P_1} \multimap X^{\perp}_{\restriction \{x\}}$$

(indeed the sequence $P_1, \dots, P_n$ can be recomputed on the fly in logspace)

The iteration problem at depth 3

- depth 1: forward propagation of information $z_i = f(z_{i-1})$ with $f : |X| \rightharpoonup |X|$
- depth 2: forward pass followed by (depth 1) backwards pass
- depth 3 is trickier

The iteration problem at depth 3

- depth 1: forward propagation of information $z_i = f(z_{i-1})$ with $f : |X| \rightharpoonup |X|$
- depth 2: forward pass followed by (depth 1) backwards pass
- depth 3 is trickier

In general, for $|X| \notin \Gamma(X)$, given a position $P \in \Gamma(X)$ and $c \sqsubset X \multimap X$,

$$\pi_2(c \cap (P \times |X|)) \in \Gamma(X) \cup \{\varnothing\} \qquad \textbf{or} \qquad \pi_1(c \cap (P \times |X|)) \in \Gamma(X^\perp) \cup \{\varnothing\}$$

That is, when Opponent plays a move on the left of $X \multimap X$, the strategy c can react:

- either by playing on the right,
- or by answering on the left.

$\longrightarrow$ need to handle back-and-forth movement of information

Conclusion

- We saw that intuitions from game semantics could be read into hypercoherences (Ehrhard 2000)
- The "game depth" seems to be a relevant parameter for computational complexity
    - As shown through an algorithm for the iteration problem at low depth
    - This might help us with our ultimate goal in implicit complexity (conjecture from N. & Pradic 2019)

Conclusion

- We saw that intuitions from game semantics could be read into hypercoherences (Ehrhard 2000)
- The "game depth" seems to be a relevant parameter for computational complexity
  - As shown through an algorithm for the iteration problem at low depth
  - This might help us with our ultimate goal in implicit complexity (conjecture from N. & Pradic 2019)

So why use hypercoherences instead of some other game model? In my case:

- finitary semantics of 2nd order MALL / affine system F
- simple combinatorial description $\Longrightarrow$ helpful for algorithmics

Anyway all this is still rather speculative...

Conclusion

- We saw that intuitions from game semantics could be read into hypercoherences (Ehrhard 2000)
- The "game depth" seems to be a relevant parameter for computational complexity
  - As shown through an algorithm for the iteration problem at low depth
  - This might help us with our ultimate goal in implicit complexity (conjecture from N. & Pradic 2019)

So why use hypercoherences instead of some other game model? In my case:

- finitary semantics of 2nd order MALL / affine system F
- simple combinatorial description $\implies$ helpful for algorithmics

Anyway all this is still rather speculative...

Thanks for your attention! Any questions?