# Higher order complexity

Hugo Férée Mathieu Hoyrup

CCA 2013

Université de Lorraine
Loria
Laboratoire lorrain de recherche en informatique et ses applications

# Type two Theory of Effectivity

## Computability

- Represented spaces, admissibility (Weihrauch)
- Extended admissibility, on QCB-spaces (Schröder)

## Complexity

- Kawamura and Cook : $Reg \subseteq \Sigma^* \to \Sigma^*$
- Polynomial time complexity based on $\mathrm{BFF}_2$
- allows to define notions of complexity over non $\sigma-$compact spaces like $\mathcal{C}([0,1],\mathbb{R})$

# "Feasible" admissibility

## Definition (Polynomial reducibility)

$\delta \leq_P \delta'$ if $\delta = \delta' \circ f$ with $f$ polynomial time computable

## Theorem (Kawamura & Cook)

*$\delta_\square$ is the "largest" representation of $\mathcal{C}([0,1],\mathbb{R})$ making $Eval : \mathcal{C}([0,1],\mathbb{R}) \to [0,1] \to \mathbb{R}$ polynomial time computable.*

$\rightarrow$ For which spaces can we do the same?

# First order representations are not sufficient

## Theorem

*Let $X$ be a Polish space that is not $\sigma$-compact. Then there is no representation of $\mathcal{C}(X, \mathbb{R})$ making the time complexity of $Eval_{X,\mathbb{R}} : \mathcal{C}(X, \mathbb{R}) \times X \to \mathbb{R}$ well-defined.*

$(X = \mathcal{C}([0, 1], \mathbb{R})$ for example) b

## Lemma

*There is no surjective partial continuous function $\phi : (\mathbb{N} \to \mathbb{N}) \to \mathcal{C}(\mathbb{N} \to \mathbb{N}, \mathbb{N})$ bounded by a total continuous function.*

## Corollary

*"Higher order is required to define complexity-friendly representations."*

# Higher order complexity

Finite types: $\tau = \mathbb{N} \mid \tau_1 \times \ldots \times \tau_n \to \mathbb{N}$

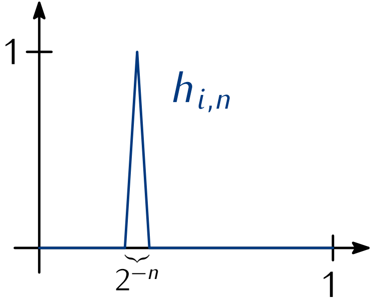
A notion of feasibility at all finite types: BFF.

Problem: Some intuitively feasible functionals are not in BFF.

## Example

$$\Gamma : (\mathcal{C}([0,1],\mathbb{R}) \to \mathbb{R}) \times \mathbb{N} \to \mathbb{R}$$

$$\Gamma(F, n) = \prod_{0 \leq i \leq 2^n} (1 + |F(h_{i,n})|)$$



# Higher order strategies

$$\overbrace{\underbrace{((\overset{x}{\mathbb{N}} \to \overset{g}{\mathbb{N}}) \to \overset{f}{\mathbb{N}})}_{\text{Opponent}} \to \overset{F}{\mathbb{N}}}^{\text{Player}}$$

Moves: $?^f$ or $!^f(v)$ + justifications.

## Definition

A strategy is a function which given a list of previous moves, outputs a valid move.

# Examples

- $x = 3$

$$\underline{\;?^x\;}\ !^x(3)$$

- $f(x) = 2x + 1$

$$\underline{\;?^f\;}\ ?^x\ \underline{\;!^x(n)\;}\ !^f(2n+1)$$

- $F(g) = g(\lambda x.x) + 1$

$$\begin{array}{llllll}
\underline{\;?^F\;}\ ?^g & \underline{\;!^g(n)\;}\ !^F(n+1) & & & & \\
 & \underline{\;?^h\;}\ ?^x\ \underline{\;!^x(n)\;}\ !^h(n) & \underline{\;!^g(m)\;}\ !^F(m+1) & & & \\
 & & \underline{\;?^h\;}\ ?^x\ \cdots & & &
\end{array}$$

# Examples

- $x = 3$

$$\underline{\;?^x\;}\ !^x(3)$$

- $f(x) = 2x + 1$

$$\underline{\;?^f\;}\ ?^x\ \underline{\;!^x(n)\;}\ ?^x\ \underline{\;!^x(n)\;}\ ?^x\ \underline{\;!^x(n)\;}\ !^f(2n+1)$$

- $F(g) = g(\lambda x.x) + 1$

$$\begin{array}{llllll}
\underline{\;?^F\;}\ ?^g & \underline{\;!^g(n)\;}\ !^F(n+1) & & & & \\
 & \underline{\;?^h\;}\ ?^x\ \underline{\;!^x(n)\;}\ !^h(n) & \underline{\;!^g(m)\;}\ !^F(m+1) & & & \\
 & & \underline{\;?^h\;}\ ?^x\ \cdots & & &
\end{array}$$

# Examples

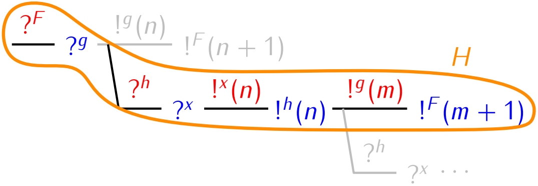
- $x = 3$

$$\frac{?^x}{\quad} \; !^x(3)$$

- $f(x) = 2x + 1$

$$\frac{?^f}{\quad} \; ?^x \; \frac{!^x(n)}{\quad} \; ?^x \; \frac{!^x(n)}{\quad} \; ?^x \; \frac{!^x(n)}{\quad} \; !^f(2n+1)$$

- $F(g) = g(\lambda x.x) + 1$



# Size of a strategy

## Definition

$$S_s(b_1, \ldots, b_n) = \max_{(s_1,\ldots s_n) \in K_{b_1} \times \cdots \times K_{b_n}} |H(s, s_1, \ldots s_n)|$$

$$K_b = \{s' \mid S_{s'} \preccurlyeq b\}$$

## Example

- $n \in \mathbb{N}$ has a strategy of size $\mathcal{O}(\log_2 n)$.
- $f : \mathbb{N} \to \mathbb{N}$ has a strategy of size $|f|(n) = \max_{|x| \leq n} |f(x)|$.
- The size of a strategy for $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is at least its modulus of continuity.

# Higher order Turing machines

## Definition (HOTM)

A HOTM is a kind of oracle Turing machine which plays a game versus its oracles.

## Property

*A strategy is computable* $\iff$ *it is represented by a HOTM.*

Run-time of a HOTM: same as for an OTM.

# Polynomial time complexity

## Definition (Higher type polynomials )

$HTP$ = simply-typed $\lambda-$calculus with $+, * : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

## Property

*HTP of type 1 and 2 are respectively the usual polynomials and the second-order polynomials.*

## Example

The complexity of $\Gamma$ is about $\mathcal{F}, n \mapsto \mathcal{F}(\lambda x.c) \times \mathcal{F}(\lambda y.P(y, n))$

# Higher order representations

## Definition (Kleene-Kreisel Spaces)

$KKS = [\mathbb{N}, \subseteq, \to, \times]$

## Definition (Representation)

A representation $\delta$ of a space $X$ with a $KKS$ $A$ is a surjective function from $A$ to $X$.

## Definition (Polynomial reduction)

$\delta_1 \leq_P \delta_2$ if $\delta_1 = \delta_2 \circ F$ for some polynomial time computable $F : A_1 \to A_2$.

# Standard representation of $\mathcal{C}(X, Y)$

## Definition

$\delta_{\mathcal{C}(X,Y)}(F) = f$ whenever $f \circ \delta_X = \delta_Y \circ F$

## Property

*$Eval : \mathcal{C}(X, Y) \times X \to Y$ is polynomial-time computable w.r.t. $(\delta_{\mathcal{C}(X,Y)}, \delta_X, \delta_Y)$*

## Theorem

*It is the largest representation making Eval polynomial.*

# Conclusion

- we have a definition of higher order complexity
- new representation spaces

- we need to understand the difference with BFF
- study the notion of admissibility of such representations (c.f. Schröder).