Higher-order Interpretations for Higher-order Complexity

Emmanuel Hainry & Romain Péchoux

LPAR, 11 may 2017

CNRS, INRIA, Université de Lorraine & LORIA, Nancy, France

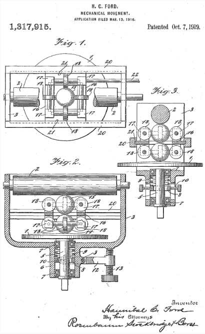
Introduction

First-order computability and complexity

- **Computability** is well understood:
  - Definitions, hierarchies (Turing degree)
  - Church-Turing's thesis
- **Computational Complexity** is well understood:
  - Definitions, classes
  - Various characterizations:
    - machine based characterizations
    - machine independent characterizations
      $\rightarrow$ Implicit Computational Complexity

Higher-order computability and complexity

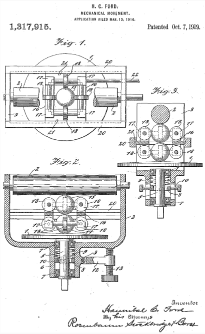
- **Computability** is (well) understood:
  - Order 2 = computations over reals.
  - **No** Church-Turing's thesis!
    - General Purpose Analog Computer by Shannon,
    - Blum-Shub-Smale model,
    - Computable Analysis (CA) by Weihrauch,
    - Oracle TM, ...



Higher-order computability and complexity

- **Computability** is (well) understood:
    - Order 2 = computations over reals.
    - **No** Church-Turing's thesis!
        - General Purpose Analog Computer by Shannon,
        - Blum-Shub-Smale model,
        - Computable Analysis (CA) by Weihrauch,
        - Oracle TM, ...
- **Complexity** is **not** well understood.
    - Polytime complexity on OTM = Basic Feasible Functions (BFF) by Constable, Melhorn
    - Polytime complexity in CA = $P(\mathbb{R})$ by Ko
    - No homogeneous theory for higher-order:

$$P(\mathbb{R}) \neq BFF$$



Objectives of this talk:

- **not developping** a new complexity theory for higher-order,
- **adapting first-order tools** for program complexity analysis,
- **validating** the theory by capturing existing higher-order complexity classes

Objectives of this talk:

- **not developping** a new complexity theory for higher-order,
- **adapting first-order tools** for program complexity analysis,
- **validating** the theory by capturing existing higher-order complexity classes

Framework:

- tool = (polynomial) **interpretations**
- target = $\mathbf{BFF}_i$, the Basic Feasible Functionals at any order.

First-order interpretations

First-order interpretations of TRS

- Defined in the 70s for showing TRS **termination**:
  - $\forall b$ of arity n, $(\!|b|\!) : \mathbb{N}^n \to^{\uparrow} \mathbb{N}$
  - $\forall l \to r \in R,\ (\!|l|\!) > (\!|r|\!)$

additive: for any constructor symbol c, $(\!|c|\!)(X) = X + k, \in \mathbb{N}$.

Let PI_{add} be the set of functions computed by TRS admitting an additive polynomial interpretation.

Theorem (Bonfante et al.)

$$PI_{add} \equiv FPTIME$$

First-order interpretations of TRS

Example

$$double(\epsilon) \to \epsilon$$
$$double(s(x)) \to s(s(double(x)))$$

First-order interpretations of TRS

Example

$$double(\epsilon) \to \epsilon$$
$$double(s(x)) \to s(s(double(x)))$$

$$(\!|\epsilon|\!) = 0, \ (\!|s|\!)(X) = X + 1, \ (\!|double|\!)(X) = 3X + 1$$

First-order interpretations of TRS

Example

$$double(\epsilon) \to \epsilon$$
$$double(s(x)) \to s(s(double(x)))$$

$$(\!|\epsilon|\!) = 0, \ (\!|s|\!)(X) = X + 1, \ (\!|double|\!)(X) = 3X + 1$$

$$(\!|double\ \epsilon|\!) = 1 > 0 = (\!|\epsilon|\!)$$
$$(\!|double\ s(x)|\!) = 3X + 4 > 3X + 3 = (\!|s(s(double(x)))|\!)$$

First-order interpretations of TRS

Example

$$double(\epsilon) \to \epsilon$$
$$double(s(x)) \to s(s(double(x)))$$

$$(\!|\epsilon|\!) = 0,\ (\!|s|\!)(X) = X + 1,\ (\!|double|\!)(X) = 3X + 1$$

$$(\!|double\ \epsilon|\!) = 1 > 0 = (\!|\epsilon|\!)$$
$$(\!|double\ s(x)|\!) = 3X + 4 > 3X + 3 = (\!|s(s(double(x)))|\!)$$

Additivity $\Rightarrow [\![double]\!] : x \mapsto 2x \in FPTIME$

Higher-order interpretations of TRS: State of the art

- **Termination:**
  - Van De Pol (1993) adapted interpretations for showing termination of **higher-order TRS.**

Higher-order interpretations of TRS: State of the art

- **Termination:**
  - Van De Pol (1993) adapted interpretations for showing termination of **higher-order TRS.**

- **Complexity:**
  - Férée et al. (2010) adapted interpretations to **first-order stream programs** for characterizing BFF (BFF_2) and $P(\mathbb{R})$.

Higher-order interpretations of TRS: State of the art

- **Termination:**
  - Van De Pol (1993) adapted interpretations for showing termination of **higher-order TRS.**
- **Complexity:**
  - Férée et al. (2010) adapted interpretations to **first-order stream programs** for characterizing BFF (BFF_2) and $P(\mathbb{R})$.
  - Baillot & Dal Lago (2016) adapted interpretations to **higher-order Simply Typed TRS** for characterizing FPtime.
    $\rightarrow$ a first step towards a better expressivity

Higher-order language

Higher Order Programming Language

Definition (Functional Language)

$$
\begin{aligned}
M \quad := \quad & x \\
\mid \quad & c \\
\mid \quad & op \\
\mid \quad & M_1 \; M_2 \\
\mid \quad & \lambda x.M \\
\mid \quad & \text{case } M \text{ of } c_1 \rightarrow M_1 | c_2 \rightarrow M_2 | ... | c_n \rightarrow M_n \\
\mid \quad & \text{letRec } f = M
\end{aligned}
$$

+ Inductive Typing

Example

Example

$$\text{letRec map} = \lambda\text{g}.\lambda\text{x.case x of } \mathbf{c} \text{ y z} \to \mathbf{c} \text{ (g y) (map g z)}$$
$$| \ \mathbf{nil} \to \mathbf{nil}$$

$List(\alpha) ::= \mathbf{nil} \mid \mathbf{c}\ \alpha\ List(\alpha)$

map: $(A \to B) \to List(A) \to List(B)$

Semantics

Four kinds of reductions:

- β reduction:

$$\lambda x.M\ N \longrightarrow_{\beta} M\{N/x\}$$

- case reduction:

$$\text{case } c_j N_j \quad \text{of } c_1 \rightarrow M_1 | ... | c_n \rightarrow M_n \longrightarrow_{\text{case}} M_j\ N_j$$

- letRec reduction:

$$\text{letRec } f = M \longrightarrow_{\text{letRec}} M\{\text{letRec } f = M/f\}$$

- Operator reduction (total functions over terms):

$$op\ M \rightarrow_{op} [\![op]\!](M)$$

+Left-most outermost reduction strategy

Higher-order interpretations

Interpretations of types

Definition

- $\llparenthesis b \rrparenthesis = \bar{\mathbb{N}} = \mathbb{N} \cup \{\top\}$
- $\llparenthesis T \to T' \rrparenthesis = \llparenthesis T \rrparenthesis \to^{\uparrow} \llparenthesis T' \rrparenthesis$

Definition

- $f : A \to^{\uparrow} B$ a monotonic function from A to B.
- $x <_{\bar{\mathbb{N}}} y$ iff $x < y$ or $y = \top$
- $f <_{A \to^{\uparrow} B} g$ iff $\forall x \in A,\ f(x) <_B g(x)$

Example (map: $(A \to B) \to List(A) \to List(B)$)

$\llparenthesis map \rrparenthesis$ is in $(\bar{\mathbb{N}} \to^{\uparrow} \bar{\mathbb{N}}) \to^{\uparrow} \bar{\mathbb{N}} \to^{\uparrow} \bar{\mathbb{N}}$.

Lattices

$$\bot_{\bar{\mathbb{N}}} = 0 \qquad \top_{\bar{\mathbb{N}}} = \top$$

$$\bot_{(\!|T \to T'|\!)} = \Lambda X^{(\!|T|\!)}.\bot_{(\!|T'|\!)} \qquad \top_{(\!|T \to T'|\!)} = \Lambda X^{(\!|T|\!)}.\top_{(\!|T'|\!)}$$

$$\sqcup^{(\!|T \to T'|\!)}(F, G) = \Lambda X^{(\!|T|\!)}. \sqcup^{(\!|T'|\!)} (F(X), G(X))$$

$$\sqcap^{(\!|T \to T'|\!)}(F, G) = \Lambda X^{(\!|T|\!)}. \sqcap^{(\!|T'|\!)} (F(X), G(X))$$

Lemma

For any type T, $((\!|T|\!), \leq, \sqcup, \sqcap, \top, \bot)$ is a complete lattice.

Interpretations of terms

$$n \oplus_{\bar{\mathbb{N}}} = \Lambda X.(n + X)$$

$$n \oplus_{(\!|T \to T'|\!)} : \Lambda F.\Lambda X.(n \oplus_{(\!|T'|\!)} F(X))$$

Definition (Interpretations)

$$\begin{array}{rcl} (\!|x|\!) & = & X \\ (\!|c|\!) & = & 1 \oplus (\Lambda X_1. \dots .\Lambda X_n. \sum_{i=1}^{n} X_i) \\ (\!|M\ N|\!) & = & (\!|M|\!)(\!|N|\!) \end{array}$$

Interpretations of terms

$$n \oplus_{\bar{\mathbb{N}}} = \Lambda X.(n + X)$$

$$n \oplus_{(\!|T \to T'|\!)} : \Lambda F.\Lambda X.(n \oplus_{(\!|T'|\!)} F(X))$$

Definition (Interpretations)

$$\begin{array}{rcl} (\!|x|\!) & = & X \\ (\!|c|\!) & = & 1 \oplus (\Lambda X_1. \ldots . \Lambda X_n. \sum_{i=1}^{n} X_i) \\ (\!|M\ N|\!) & = & (\!|M|\!)(\!|N|\!) \\ (\!|\lambda x.M|\!) & = & 1 \oplus (\Lambda (\!|x|\!).(\!|M|\!)) \end{array}$$

Interpretations of terms

$$n\oplus_{\bar{\mathbb{N}}} = \Lambda X.(n + X)$$

$$n\oplus_{(\!|T \to T'|\!)} : \Lambda F.\Lambda X.(n \oplus_{(\!|T'|\!)} F(X))$$

Definition (Interpretations)

$$\begin{array}{rcl}
(\!|x|\!) & = & X \\
(\!|c|\!) & = & 1 \oplus (\Lambda X_1. \ldots .\Lambda X_n. \sum_{i=1}^{n} X_i) \\
(\!|M\ N|\!) & = & (\!|M|\!)(\!|N|\!) \\
(\!|\lambda x.M|\!) & = & 1 \oplus (\Lambda (\!|x|\!).(\!|M|\!)) \\
(\!|case\ M\ of\ \ \ldots c_i \to M_i \ldots|\!) & = & 1 \oplus \sqcup_i \{(\!|M_i|\!) R_i \mid (\!|c_i|\!) R_i \leq (\!|M|\!)\}
\end{array}$$

Interpretations of terms

$$n \oplus_{\bar{\mathbb{N}}} = \Lambda X.(n + X)$$

$$n \oplus_{(\!|T \to T'|\!)} : \Lambda F.\Lambda X.(n \oplus_{(\!|T'|\!)} F(X))$$

Definition (Interpretations)

$$\begin{array}{rcl}
(\!|x|\!) & = & X \\
(\!|c|\!) & = & 1 \oplus (\Lambda X_1.\ldots.\Lambda X_n.\sum_{i=1}^{n} X_i) \\
(\!|M\ N|\!) & = & (\!|M|\!)(\!|N|\!) \\
(\!|\lambda x.M|\!) & = & 1 \oplus (\Lambda(\!|x|\!).(\!|M|\!)) \\
(\!|case\ M\ of\ \ldots c_i \to M_i \ldots|\!) & = & 1 \oplus \sqcup_i \{(\!|M_i|\!)R_i \mid (\!|c_i|\!)R_i \leq (\!|M|\!)\} \\
(\!|letRec\ f = M|\!) & = & \sqcap\{F \mid F \geq 1 \oplus (\Lambda(\!|f|\!).(\!|M)|\!)F\}
\end{array}$$

Interpretations of terms

$$n\oplus_{\bar{\mathbb{N}}} = \Lambda X.(n + X)$$

$$n\oplus_{(\!|T \to T'|\!)} : \Lambda F.\Lambda X.(n \oplus_{(\!|T'|\!)} F(X))$$

Definition (Interpretations)

$$\begin{array}{rcl}
(\!|x|\!) & = & X \\
(\!|c|\!) & = & 1 \oplus (\Lambda X_1.\ldots.\Lambda X_n. \sum_{i=1}^{n} X_i) \\
(\!|M\ N|\!) & = & (\!|M|\!)(\!|N|\!) \\
(\!|\lambda x.M|\!) & = & 1 \oplus (\Lambda (\!|x|\!).(\!|M|\!)) \\
(\!|case\ M\ of\ \ \ldots c_i \to M_i \ldots|\!) & = & 1 \oplus \sqcup_i \{(\!|M_i|\!) R_i \mid (\!|c_i|\!) R_i \leq (\!|M|\!)\} \\
(\!|letRec\ f = M|\!) & = & \sqcap \{F \mid F \geq 1 \oplus (\Lambda (\!|f|\!).(\!|M|\!)) F\}
\end{array}$$

$$(\!|op\ \mathtt{M}|\!) \geq (\!|[\![op]\!](\mathtt{M})|\!)$$

Properties of interpretations

Theorem

Any term M has an interpretation.

Knaster-Tarski: $lfp(\Lambda X.1 \oplus ((\Lambda (\!| f |\!).(\!| M |\!))X))$

Lemma

If $M \longrightarrow N$, then $(\!| M |\!) \geq (\!| N |\!)$.

If $M \longrightarrow_{\alpha} N$, $\alpha \neq op$, then $(\!| M |\!) > (\!| N |\!)$.

Lemma

If $M :: B$ and $(\!| M |\!) \neq \top$ then M terminates in time $O((\!| M |\!))$.

Example of Interpretation

$$(\!|\mathit{letRec}\ \mathit{map} = \lambda g.\lambda x.\mathit{case}\ x\ \mathit{of}\ \mathbf{c}\ y\ z \to \mathbf{c}\ (g\ y)\ (\mathit{map}\ g\ z) | \mathbf{nil} \to \mathbf{nil}|\!)$$

$$= \dots$$

$$\vdots$$

$$= \dots$$

$$= \sqcap \{F \mid F \geq 5 \oplus (\Lambda G.\Lambda X. \sqcup \{((G\ Y) \oplus (F\ G\ Z)) | \ X \geq 1 \oplus Y \oplus Z\}\}$$

with 1 (letRec), 2 (Lambda), 1 (Case), 2 (Cons **c**), 2 (Cons **nil**)

Relaxing interpretations

$$(\!|\mathit{letRec}\ \mathit{map} = \lambda g.\lambda x.\mathit{case}\ x\ \mathit{of}\ \mathbf{c}\ y\ z \rightarrow \mathbf{c}\ (g\ y)\ (\mathit{map}\ g\ z)|\!)$$

$$= \sqcap\{F \mid F \geq 5 \oplus (\Lambda G.\Lambda X. \sqcup \{((G\ Y) \oplus (F\ G\ Z)) \mid X \geq 1 \oplus Y \oplus Z\}\}$$

Relaxing interpretations

$$(\!|letRec\ map = \lambda g.\lambda x.case\ x\ of\ \mathbf{c}\ y\ z \rightarrow \mathbf{c}\ (g\ y)\ (map\ g\ z)|\!)$$

$$= \sqcap\{F \mid F \geq 5 \oplus (\Lambda G.\Lambda X. \sqcup \{((G\ Y) \oplus (F\ G\ Z)) | \ X \geq 1 \oplus Y \oplus Z\}\}$$

$$\leq \sqcap\{F \mid F \geq 5 \oplus (\Lambda G.\Lambda X.((G\ (X-1)) \oplus (F\ G\ (X-1))))\}$$

(constraint upper bound)

Relaxing interpretations

$$(\!|\mathit{letRec}\ \mathit{map} = \lambda g.\lambda x.\mathit{case}\ x\ \mathit{of}\ \mathbf{c}\ y\ z \rightarrow \mathbf{c}\ (g\ y)\ (\mathit{map}\ g\ z)|\!)$$

$$= \sqcap\{F \mid F \geq 5 \oplus (\Lambda G.\Lambda X. \sqcup \{((G\ Y) \oplus (F\ G\ Z)) | \ X \geq 1 \oplus Y \oplus Z\}\}$$

$$\leq \sqcap\{F \mid F \geq 5 \oplus (\Lambda G.\Lambda X.((G\ (X-1)) \oplus (F\ G\ (X-1))))\}$$

(constraint upper bound)

$$\leq \Lambda G.\Lambda X.(5 + G\ X)) \times X$$

(min upper bound)

A characterization of BFF_i

A BTLP is a non-recursive and well-formed procedure P defined by:

$$
\begin{aligned}
P \quad & ::= v^{\tau_1\times\ldots\times\tau_n\to\mathbb{N}}(v_1^{\tau_1},\ldots,v_n^{\tau_n}) P^* V I^* \textbf{ Return } v_r^{\mathbb{N}} \textbf{ End} \\
V \quad & ::= \textbf{var } v_1^{\mathbb{N}},\ldots,v_n^{\mathbb{N}}; \\
I \quad & ::= v^{\mathbb{N}} := E; \mid \textbf{Loop } v_0^{\mathbb{N}} \textbf{ with } v_1^{\mathbb{N}} \textbf{ do } I^* \textbf{ EndLoop ;} \\
E \quad & ::= 1 \mid v^{\mathbb{N}} \mid v_0^{\mathbb{N}} + v_1^{\mathbb{N}} \mid v_0^{\mathbb{N}} - v_1^{\mathbb{N}} \mid v_0^{\mathbb{N}} \# v_1^{\mathbb{N}} \mid \\
& \qquad v^{\tau_1\times\ldots\times\tau_n\to\mathbb{N}}(A_1^{\tau_1},\ldots,A_n^{\tau_n}) \\
A \quad & ::= v \mid \lambda v_1,\ldots,v_n.v(v'_1\ldots,v'_m) \quad \text{with } v \notin \{v_1,\ldots,v_n\}
\end{aligned}
$$

$order(b) = 0 \quad order(T \to T') = \max(order(T)+1, order(T'))$

BFF$_i$ is the class of order i functionals computable by a BTLP program.

Higher-order polynomial

$$P_1 ::= c \in \mathbb{N} | X_0 | P_1 + P_1 | P_1 \times P_1$$
$$P_{i+1} ::= P_i | P_{i+1} + P_{i+1} | P_{i+1} \times P_{i+1} | X_i(P_i)$$

Definition

Let FP_i, $i > 0$, be the class of polynomial functionals at order i that consist in functionals computed by closed terms M such that:

- $order(M) = i$
- $(\!| M |\!)$ is bounded by an order i polynomial ($\exists P_i,\ (\!| M |\!) \leq P_i$).

Results

Define the Safe Feasible Functionals at order i, SFF_i by:

$$SFF_1 = BFF_1,$$
$$\forall i \geq 1,\ SFF_{i+1} = BFF_{i+1 \restriction SFF_i}$$

Theorem (Hainry Péchoux)
For any order i, $FP_i = SFF_i$.

In particular, FP_1 is FPtime and FP_2 is BFF with FPtime oracles.

Conclusion

Conclusion

Results

- An interpretation theory for higher-order functional languages
- A characterization of well-known classes: BFF_i

Issues and future work

- BFF_i is known to be restricted
  - $\rightarrow$ see Férée's phD manuscript (2014)
- The interpretation synthesis problem is very hard.
- Interpretations for complexity analysis of real operators and real-based languages.
- Adapt the results to space: does it make sense?
- Adapt ICC techniques to characterize $P(\mathbb{R})$.