Characterizations of Polynomial Complexity Classes with a Better Intensionality

Romain Péchoux and Jean-Yves Marion

Carte project, Loria, Nancy

PPDP 2008
Motivations

- Resource control, resource certificates by static analysis

- Applications to Term Rewriting Systems / Functional programs

- Using:
  - Sup-interpretation by Marion and Péchoux[2006]
  - Dependency pairs by Arts and Giesl[2000]

- Our concern is intensionality rather than extensionality
Syntax and semantics of First Order Language

A program is a quadruplet $\langle X, C, F, R \rangle$

Constructors: $c \in C$, functions: $f \in F$, variables: $x \in X$.

(Terms) $T \ni t ::= x \mid c(t_1, \cdots, t_n) \mid f(t_1, \cdots, t_n)$

(Patterns) $P \ni p ::= c(p_1, \cdots, p_n) \mid x$

(Rules) $R \ni r ::= f(p_1, \cdots, p_n) \rightarrow t$

(Values) $V \ni v ::= c(v_1, \cdots, v_n)$

- The computational domain is the set of values $V^* = V \cup \{\text{Err}\}$
- Define $[e] = w$ iff $e \rightarrow^* w$ and $w \in V^*$
- Linear and disjoint patterns (Confluence: Huet[1980])
Program example

Example (Unary logarithm)

\[
\begin{align*}
\text{half}(0) & \rightarrow 0 \\
\text{half}(S(0)) & \rightarrow 0 \\
\text{half}(S(S(y))) & \rightarrow S(\text{half}(y)) \\
\log(0) & \rightarrow 0 \\
\log(S(0)) & \rightarrow 0 \\
\log(S(S(y))) & \rightarrow S(\log(S(\text{half}(y))))
\end{align*}
\]

Semantics:

- \([\log(S^n(0))] = S^{\lfloor \log(n) \rfloor}(0)\)
- \([\frac{1}{2}\text{half}(S^n(0))] = S^{\lfloor n/2 \rfloor}(0)\) \quad \text{with} \quad S^n(0) = S(\ldots S(0) \ldots) \quad \text{n times} \ S
Quasi-interpretations (QI)

Definition (Quasi-interpretation)

A program $p$ admits a quasi-interpretation if there is a
- total: $\forall b, \langle b \rangle$ is a function from $(\mathbb{R}^+)^n$ to $\mathbb{R}^+$,
- monotonic: $X_i \geq Y_i \Rightarrow \langle b \rangle(\ldots, X_i, \ldots) \geq \langle b \rangle(\ldots, Y_i, \ldots)$,
- and subterm: $\langle b \rangle(X_1, \cdots, X_n) \geq X_i$

assignment $\langle - \rangle$ such that for each rule $l \rightarrow r$:

$$\langle l \rangle \geq \langle r \rangle$$
Example of quasi-interpretation

Example (Quasi-interpretation of logarithm)

\[
\begin{align*}
\langle \text{half}(0) \rangle & \geq \langle 0 \rangle \\
\langle \text{half}(S(0)) \rangle & \geq \langle 0 \rangle \\
\langle \text{half}(S(S(y))) \rangle & \geq \langle S(\text{half}(y)) \rangle \\
\langle \log(0) \rangle & \geq \langle 0 \rangle \\
\langle \log(S(0)) \rangle & \geq \langle 0 \rangle \\
\langle \log(S(S(y))) \rangle & \geq \langle S(\log(S(\text{half}(y)))) \rangle \\
\langle 0 \rangle & = 0 \\
\langle S \rangle(X) & = X + 1 \\
\langle \log \rangle(X) & = \langle \text{half} \rangle(X) = X
\end{align*}
\]
Example \(|l| \geq |r|\)

Example (Quasi-interpretation of logarithm)

\[
\text{half}(S(S(y))) \rightarrow S(\text{half}(y))
\]

\[
\langle \text{half}(S(S(y))) \rangle = \langle \text{half} \rangle(\langle S \rangle(\langle S \rangle(Y))) = \langle y \rangle = Y
\]

\[
= \langle \text{half} \rangle(Y + 2) = \langle S \rangle(X) = X + 1
\]

\[
= Y + 2 \geq Y + 1
\]

\[
= \langle S(\text{half}(y)) \rangle
\]
Fundamental Lemma

**Definition (Polynomial quasi-interpretation)**

A quasi-interpretation is polynomial if $\forall b \ (b) \text{ is a max-polynomial.}$

**Definition (Additive quasi-interpretation)**

A quasi-interpretation is additive if $\forall c \in Cns$ of arity $n$:
- if $n > 0$ then $(c)(X_1, \ldots, X_n) = \sum_{i=1}^{n} X_i + \alpha_c$ with $\alpha_c \geq 1$
- if $n = 0$ then $(c) = 0$

**Fundamental Lemma**

If a program admits a polynomial and additive QI then there is a polynomial $P$ s.t. $\forall b$ and $\forall v_1, \ldots, v_n \in Values$:

$$\text{if } \llbracket b(v_1, \ldots, v_n) \rrbracket \in \mathcal{V}, \text{ then } P(\max_{i=1,n}(|v_i|)) \geq \| \llbracket b(v_1, \ldots, v_n) \rrbracket \|$$
Characterizations of \( \text{FP}_{\text{TIME}} \) and \( \text{FP}_{\text{SPACE}} \)

**Theorem (Bonfante, Marion and Moyen[2000,2001])**

The set of functions computed by programs having an additive and polynomial QI and:

- ordered by product \( \prec_{\text{rpo}} \) is exactly \( \text{FP}_{\text{TIME}} \).
- ordered by product and lexicographic \( \prec_{\text{rpo}} \) is exactly \( \text{FP}_{\text{SPACE}} \).
Quasi-interpretations fail on natural algorithms

Example (A motivating example: Division)

\[
\begin{align*}
\text{minus}(0, y) & \rightarrow 0 \\
\text{minus}(S(x), 0) & \rightarrow S(x) \\
\text{minus}(S(x), S(y)) & \rightarrow \text{minus}(x, y) \\
q(0, S(y)) & \rightarrow 0 \\
q(S(x), S(y)) & \rightarrow S(q(\text{minus}(x, y), S(y)))
\end{align*}
\]

\[
[j_{q(S^n(0), S^m(0))}] = S^{\lceil n/m \rceil}(0)
\]
Is there a quasi-interpretation of division?

Answer: NO!

Example (Division)

Suppose that $\langle - \rangle$ is an additive QI s.t. $\langle S \rangle(X) = X + k$.

$$
\langle q(S(x), S(y)) \rangle \geq \langle S(q(\text{minus}(x, y), S(y))) \rangle \\
\geq k + \langle q \rangle(\langle \text{minus} \rangle(X, Y), Y + k) \quad \langle S \rangle(X) = X + k \\
> \langle q \rangle(\max(X, Y), Y + k) \quad \text{Subterm property} \\
\geq \langle q \rangle(X + k, Y + k) \quad \text{For } Y \geq X + k \\
= \langle q(S(x), S(y)) \rangle
$$

Develop a tool with more intensionality:
- Partial instead of total assignments
- No more subterm property
Sup-interpretations (Marion and Péchoux[2006,2008])

Definition (Sup-interpretation)

A program admits a sup-interpretation if there is a partial assignment \( \theta \) such that:

1. \( \theta \) is monotonic.
2. \( \forall v \in \mathcal{V}, \ \theta(v) \geq |v| \)
3. \( \forall b \in \text{dom}(\theta) \) of arity \( n \) and for each values \( v_1, \ldots, v_n \), if \( \llbracket b(v_1, \ldots, v_n) \rrbracket \in \mathcal{V} \), then:

\[
\theta(b(v_1, \ldots, v_n)) \geq \theta(\llbracket b(v_1, \ldots, v_n) \rrbracket)
\]
Dependency pairs as a way to synthesize SI

**Definition (Dependency pair and dependency pair graph)**

- \( \langle f(p), g(e) \rangle \) is a dependency pair if \( f(p) \rightarrow C[g(e)] \in \mathcal{R} \) and \( g \in Fct \)
- The dependency pair graph \( G = (V, E) \) is defined by:
  - \( V \) is the set of dependency pairs
  - \( (\langle u, v \rangle, \langle w, z \rangle) \in E \) iff there is \( \sigma \) such that \( v\sigma^* \rightarrow w\sigma \)

**Example (Dependency pair of logarithm)**

\[
\langle q(s(x), s(y)), q(minus(x, y), s(y)) \rangle \quad \langle q(s(x), s(y)), minus(x, y) \rangle \quad \langle minus(s(x), s(y)), minus(x, y) \rangle
\]
Dependency pairs as a way to synthesize SI

**Theorem (Dependency pair (Arts and Giesl[2000]))**

A program is terminating if there is a reduction pair $(\succ, \succeq)$ s.t.:

- For each rule $l \to r$, $l \succeq r$
- For each dependency pair $\langle s, t \rangle$, $s \succeq t$
- For each cycle of the dependency pair graph, $\exists \langle s, t \rangle$ s.t. $s \succ t$

**Definition**

If $p$ terminates wrt the reduction pair reduction pair $(\succ, \succeq)$, we say that the program is in $DP(\succeq, \succ)$.

**Definition (Polynomial space interpretation)**

A program admits a polynomial space interpretation $\langle - \rangle$ if $\langle - \rangle$ is a monotonic, additive and polynomial assignment such that:

- For each rule $l \to r$, $\langle |l| \rangle \succeq \langle |r| \rangle$
- For each dependency pair $\langle s, t \rangle$ s.t. $\langle |s| \rangle \succeq \langle |t| \rangle$
Polynomial space interpretations

**Theorem**

*Every polynomial space interpretation is a sup-interpretation*

**Example**

\[
\begin{align*}
\langle 0 \rangle &= 0 \\
\langle S \rangle(X) &= X + 1 \\
\langle \text{half} \rangle(X) &= X/2 \\
\langle \log \rangle(X) &= X
\end{align*}
\]

In particular:

\[
\begin{align*}
\langle \text{half}(0) \rangle &\geq \langle 0 \rangle \\
\langle \text{half}(S(0)) \rangle &\geq \langle 0 \rangle \\
\langle \text{half}(S(S(x))) \rangle &\geq \langle S(\text{half}(x)) \rangle \\
\langle \text{half}(S(S(x))) \rangle &\geq \langle \text{half}(x) \rangle
\end{align*}
\]
Some restrictions on space

Definition (Bounded Recursive Calls)

A program is in \( BRC(\langle - \rangle) \) if for every dependency pair 
\((f(p_1, \cdots, p_n), g(e_1, \cdots, e_m))\) involved in a cycle of the DP graph, we have:

\[
\sum_{j \in \{1, \ldots, n\}} \langle p_j \rangle \geq \sum_{i \in \{1, \ldots, m\}} \langle e_i \rangle
\]

Definition (\( DP(\text{SPACE}) \))

A program is in \( DP(\text{SPACE}) \) if there is an additive, monotonic and max-polynomial assignment \( \langle - \rangle \) and a constant \( \delta > 0 \) such that:

\[
p \in DP(\geq_{\langle - \rangle}, >_{\delta_{\langle - \rangle}}) \cap BRC(\langle - \rangle)
\]

where \( x >_{\delta_{\langle - \rangle}} y \) if \( \langle x \rangle > \langle y \rangle + \delta \) (Lucas[2005])
Examples

Example (Bounded Recursive Calls)

Two dependency pairs are involved in cycles:

\[ \langle \text{minus}(S(x), S(y)), \text{minus}(x, y) \rangle : \|S(x)\| + \|S(y)\| \geq \|x\| + \|y\| \]
\[ \langle q(S(x), S(y)), q(\text{minus}(x, y), S(y)) \rangle : \|S(x)\| + \|S(y)\| \geq \|\text{minus}(x, y)\| + \|S(y)\| \]

Example (\(DP(\text{SPACE})\))

To show that \( p \in DP(\geq\langle-\rangle, >^{\langle-\rangle} \delta) \cap BRC(\langle-\rangle) \), we have to find \( \delta \) s.t.:

\[ \langle \text{minus}(S(x), S(y)), \text{minus}(x, y) \rangle : \|\text{minus}(S(x), S(y))\| >^{\langle-\rangle} \delta \|\text{minus}(x, y)\| \]
\[ \langle q(S(x), S(y)), q(\text{minus}(x, y), S(y)) \rangle : \|q(S(x), S(y))\| >^{\langle-\rangle} \delta \|q(\text{minus}(x, y), S(y))\| \]
\[ \langle q(S(x), S(y)), \text{minus}(x, y) \rangle : \|q(S(x), S(y))\| \geq \|\text{minus}(x, y)\| \]

\( \|S\|(X) = X + 1, \|\text{minus}\|(X, Y) = X, \|q\|(X, Y) = X + Y \) and \( \delta = 1 \)
Some restrictions on time

**Definition (Neighborhood)**

\[ N(l \rightarrow r) = \{(l, t), \text{ such that } (l, t) \text{ is involved in a cycle of the SDP graph and there is a context } C[\diamond] \text{ such that } r = C[t] \} \]

**Definition (**\(DP(\text{TIME})\))

A program \(p\) belongs to \(DP(\text{TIME})\) if there is an additive, monotonic and max-polynomial assignment \(\langle - \rangle\) and a constant \(\delta > 0\) such that:

- \(p \in DP(\geq \langle - \rangle, >_{\delta} \langle - \rangle) \cap BRC(\langle - \rangle)\)
- and, for each neighborhood \(\{(l, t_1), \ldots, (l, t_n)\}\):

\[
|l| >_{\delta} \sum_{i=1}^{n} |t_i|
\]
**Examples**

**Example (Neighborhood)**

There are two neighborhoods:

\[ N(\text{minus}(S(x), S(y))) \rightarrow S(\text{minus}(x, y))) = \{\langle \text{minus}(S(x), S(y)), \text{minus}(x, y)\rangle\} \]
\[ N(q(S(x), S(y))) \rightarrow S(q(\text{minus}(x, y), S(y)))) = \{\langle q(S(x), S(y)), q(\text{minus}(x, y), S(y))\rangle\} \]

**Example (\text{DP}(\text{TIME}))**

We have already shown that \( p \in \text{DP}(\geq\langle-\rangle, >\langle-\rangle_1) \cap \text{BRC}(\langle-\rangle) \)

It remains to check that for the neighborhoods, we have:

\[ \langle \text{minus}(S(x), S(y))\rangle \geq \langle \text{minus}(x, y)\rangle \]
\[ \langle q(S(x), S(y))\rangle \geq \langle q(\text{minus}(x, y), S(y))\rangle \]

\[ \langle S\rangle(X) = X + 1, \langle \text{minus}\rangle(X, Y) = X, \langle q\rangle(X, Y) = X + Y \]
Conclusion

**Theorem**

- $DP(\text{TIME}) = \text{FP}_\text{TIME}$
- $DP(\text{SPACE}) = \text{FP}_\text{SPACE}$

We obtain a better intensionality (tight upper bounds)

- with non-subterm assignments
- and dependency pairs

We can extend the result to RPO characterizations:

**Theorem**

1. $DP(\preceq, \succ^\text{prod}_{\text{rpo}}) \cap QI^{\text{add}}_{\text{poly}}$ is exactly $\text{FP}_\text{TIME}$
2. $DP(\preceq, \succ_{\text{rpo}}) \cap QI^{\text{add}}_{\text{poly}}$ is exactly $\text{FP}_\text{SPACE}$

whenever $(\preceq, \succ_{\text{rpo}})$ is a reduction pair satisfying both $\preceq \circ \succ_{\text{rpo}} \subseteq \succ_{\text{rpo}}$ and $\prec_{\text{rpo}} \circ \preceq \subseteq \succ_{\text{rpo}}$