On the Intensional Expressive Power of Bounded Calculi

Work in Progress

Ugo Dal Lago

Dipartimento di Informatica
Università di Verona

FOLLIA workshop on Implcicit Complexity, January 18th 2005

Outline

Classical Implicit Computational Complexity

- Most results in Implicit Computational Complexity have the following shape:
  - Starting from a programming language or logical system P, they isolate a subclass P^* of P.
  - Every *program* in P^* is shown to be computable in time (or space) bounded by a function in a class C (**soundness**).
  - Every *function* computable in time bounded by a function in C is shown to be representable in P^* (**completeness**).
- C could be the class of polynomials, the class of elementary functions, the class of primitive recursive functions, etc.
- But there is a mismatch here! Soundness is proved *intensionally*, completeness is proved *extensionally*.

Towards Intensionality

- Suppose we replace the usual completeness statement:
  - Every *function* computable in time bounded by a function in C is shown to be representable in P^* (**extensional completeness**).

  with another one
  - P^* includes every *program* computable in time bounded by a function in C. (**intensional completeness**).
- Intensional completeness is more interesting from a programming perspective!
- Unfortunately, if P^* is sound and intensionally complete, then P^* is not recursively enumerable, provided P and C are nontrivial.
- What should we do?

A New Programme for ICC

- Why not analyzing P^* itself as opposed to the class of representable functions for P^*?
- This will help to:
  - **Compare** the intensional expressive power of existing ICC systems.
  - Understanding **how far we are** from intentionally complete systems.
  - Ultimately, understanding **the intrinsic limits** of ICC.
- I will present an example of what can be done. P will be recursion on notation, while P^* will be Cobham's bounded recursion on notation.

Programs

- We can give the notion of a **primitive recursive definition (PRD) with a integer arity** as follows, by induction:
  - The symbol nil is a PRD with arity 0.
  - The symbols $cons0$ and $cons1$ are PRDs with arity 1.
  - For every $n \geq 1$ and for every $1 \leq i \leq n$, the symbo π_i^n is a PRD with arity n.
  - If s is a PRD with arity $n \geq 0$ and $t_1, \ldots, t_n$ are PRDs with arity $m \geq 0$, then $comp(s, t_1, \ldots, t_n)$ is a PRD with arity m.
  - If s is PRD with arity $n \geq 0$ and t_0, t_1 are two PRDs with arity $n+2$, then $rec(s, t_0, t_1)$ is a PRD with arity $n+1$.

  $\mathcal{D}$ is the class of all primitive recursive definitions.
- The semantics of a PRD s with arity n is a **primitive recursive function** $[\![s]\!] : \mathbb{B}^n \to \mathbb{B}$

Program Evaluation

A **primitive recursive expression (PRE)** is either an element of $\mathbb{B}$ or has the form $s(e_1, \ldots, e_n)$, where s is a PRD with arity n and $e_1, \ldots, e_n$ are PREs. Evaluations of PREs can be described as a rewrite relation:

$$
\begin{aligned}
nil &\to \varepsilon \\
cons0(u) &\to 0 \cdot u \\
cons1(u) &\to 1 \cdot u \\
\pi_i^n(u_1, \ldots, u_n) &\to u_i \\
comp(s, t_1, \ldots, t_n)(u_1, \ldots, u_m) &\to s(t_1(u_1, \ldots, u_m), \ldots, t_n(u_1, \ldots, u_m)) \\
rec(s, t_0, t_1)(\varepsilon, u_1, \ldots, u_n) &\to s(u_1, \ldots, u_n) \\
rec(s, t_0, t_1)(0 \cdot u_0, u_1, \ldots, u_n) &\to t_0(rec(s, t_0, t_1)(u_0, \ldots, u_n), u_0, \ldots, u_n) \\
rec(s, t_0, t_1)(1 \cdot u_0, u_1, \ldots, u_n) &\to t_1(rec(s, t_0, t_1)(u_0, \ldots, u_n), u_0, \ldots, u_n)
\end{aligned}
$$

Program Evaluation and Complexity

- By the following theorem, the number of reduction steps induces an invariant cost model:

Theorem

For every PRD t with arity n, there is a polynomial $p : \mathbb{N}^{n+1} \to \mathbb{N}$ such that whenever $t(u_1, \ldots, u_n) \to^m e$, it holds that $|e| \leq p(m, |u_1|, \ldots, |u_n|)$

- t is **polytime** iff there is a polynomial $p : \mathbb{N}^n \to \mathbb{N}$ such that whenever $t(u_1, \ldots, u_n) \to^m v$, it holds that $m \leq p(|u_1|, \ldots, |u_n|)$. $\mathcal{PD}$ is the class of all primitive recursive definitions.
- A PRD t is **hereditarily polytime** iff t and every sub-definition of t are polytime. $\mathcal{HPD}$ is the class of all primitive recursive definitions.

Bounded Programs

- We can now define the class of **bounded primitive recursive definitions (BPRD)** as follows. Take the definition of a PRD and modify the last inductive clause. Replace
  - If s is PRD with arity $n \geq 0$ and t_0, t_1 are two PRDs with arity $n+2$, then $rec(s, t_0, t_1)$ is a PRD with arity $n+1$.

  by
  - if s is BPRD with arity $n \geq 0$ and t_0, t_1 are two BPRDs with arity $n+2$, then $t = rec(s, t_0, t_1)$ is a BPRD with arity $n+1$ provided there is a polynomial $p : \mathbb{N}^{n+1} \to \mathbb{N}$ such that $|[\![t]\!](u_0, \ldots, u_n)| \leq p(|u_0|, \ldots, |u_n|)$ whenever $u_0, \ldots, u_n \in \mathbb{B}$.
- $\mathcal{HPD}$ is the class of all bounded primitive recursive definitions.
- This is a slight variation on bounded recursion on notation as defined by Cobham.

On the Expressive Power of Bounded Programs

- The following is due to Cobham:

Theorem

$[\![\mathcal{BD}]\!] = \mathscr{P}$.

- What we would like, however, is a result on the **intensional** expressive power of bounded recursion. We can easily get it:

Theorem

$\mathcal{BD} = \mathcal{HPD}$.

- This is not too surprising, since the definition of a BPRD involves the existence of a polynomial, precisely as the definition of a polytime primitive recursive definition.

Polynomial Expressions

- The language of **polynomial expressions** is defined as follows:

$$p ::= 0 \mid 1 \mid x \mid p + p \mid sum(p, x, p).$$

  where x ranges over a countable class $\mathcal{X}$ of variables. The variable x acts as a binder on q in the polynomial expression $sum(p, x, q)$.
- The set $\mathcal{FV}(p)$ of free variables of the polynomial expression p is defined in the usual way.
- The semantic $[\![p]\!]_\rho$ of a polynomial expression p in an environment $\rho : \mathcal{X} \to \mathbb{N}$ is defined as follows:

$$\begin{aligned}
[\![0]\!]_\rho &= 0 \\
[\![1]\!]_\rho &= 1 \\
[\![x]\!]_\rho &= \rho(x) \\
[\![p + q]\!]_\rho &= [\![p]\!]_\rho + [\![q]\!]_\rho \\
[\![sum(p, x, q)]\!]_\rho &= \sum_{i < [\![p]\!]_\rho} [\![q]\!]_{\rho\{x \leftarrow i\}}
\end{aligned}$$

Polyomial Expressions and Fomulae

- $[\![p]\!]_\rho$ only depends on the values of ρ on variables in $\mathcal{FV}(p)$:

Theorem

Let p be a resource expression and let $\mathcal{FV}(p) = \{x_1, \ldots, x_n\}$. Then there is a polynomial $P : \mathbb{N}^n \to \mathbb{N}$ such that $[\![p]\!]_{\rho\{x_1 \leftarrow m_1, \ldots, x_n \leftarrow m_n\}} \leq P(m_1, \ldots, m_n)$ for every $m_1, \ldots, m_n \in \mathbb{N}$.

- The symbol $\leq$ denotes the pointwise order between polynomial expressions: $p \leq q$ iff $[\![p]\!]_\rho \leq [\![q]\!]_\rho$ for every ρ.
- A partial order $\sqsubseteq$ on polynomial expressions is a **polynomial order** iff $p \sqsubseteq q$ implies $p \leq q$.
- The language of **fomulae** is defined as follows:

$$A ::= \alpha(p_1, \ldots, p_n) \mid A \otimes A \mid A \multimap A \mid \forall \alpha.A \mid !_{x<p} A$$

where $x \notin \mathcal{FV}(p)$, α ranges over a countable class of atoms.

Bounded Linear Logic (BLL)

Axiom and Cut

$$\frac{A \sqsubseteq B}{A \vdash B}\ A \qquad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\ U$$

Structural Rules

$$\frac{\Gamma \vdash B}{\Gamma, !_{x<p} A \vdash B}\ W \qquad \frac{\Gamma, !_{x<p} A, !_{y<q} A\{p + y/x\} \vdash B \quad r \sqsupseteq p + q}{\Gamma, !_{x<r} A \vdash B}\ X$$

Multiplicative Logical Rules

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ R_{\multimap} \qquad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}\ L_{\multimap}$$

Exponential Rules

$$\frac{A_1, \ldots, A_n \vdash B \quad \forall 1 \leq i \leq n. p_i \sqsupseteq p}{!_{x<p_1} A_1, \ldots, !_{x<p_n} A_n \vdash !_{x<p} B}\ P_! \qquad \frac{A\{1/x\}, \Gamma \vdash B \quad p \sqsupseteq 1}{!_{x<p} A, \Gamma \vdash B}\ D_!$$

$$\frac{!_{x<p} !_{z<q} A\{z + sum(x, x, q)/y\}, \Gamma \vdash B \quad r \sqsupseteq sum(p, x, q)}{!_{y<r} A, \Gamma \vdash B}\ N_!$$

Some Intersting Facts

- For every Bounded Linear Logic proof π, there is a corresponding proof $\overline{\pi}$ in Multiplicative and Exponential Linear Logic (MELL). Moreover, skeletons of π and $\overline{\pi}$ are indentical. Surprisingly:

Theorem

For every cut-free MELL proof ρ, there is a (cut-free) bounded linear logic proof π such that $\rho = \overline{\pi}$.

- Soundness holds independently on the underlying reduction strategy.
- Completeness is proved by embedding bounded recursion on notation into Bounded Linear Logic.
- Can we sharply characterize the class of $\overline{\mathsf{BLL}}$?