Taylor Expansion as a Finitary Approximation Framework for the Infinitary \(\lambda\)-Calculus

Rémy Cerda, Aix-Marseille Université, CNRS, I2M
Joint work with Lionel Vaux Auclair

**What is it all about?**

**Infinite \(\lambda\)-terms...**

The usual, finite \(\lambda\)-terms are:
- a mathematical representation of programs
- terms on a certain signature (or the corresponding syntactic trees) equipped with a rewriting rule ("execution" of the programs).

Just as programs can have infinite loops, \(\lambda\)-terms can reduce infinitely:

\[
\begin{align*}
Yf & \to_p \varnothing \to_p \varnothing \to_p \varnothing \to_p \ldots \\
& \to fYf \to f \varnothing \to f \varnothing \to f \varnothing \ldots \\
& \to fYf \to f \varnothing \to f \varnothing \to f \varnothing \ldots \\
& \to fYf \to f \varnothing \to f \varnothing \to f \varnothing \ldots
\end{align*}
\]

This behaviour can be represented by an infinitary \(\lambda\)-calculus: (possibly) infinite \(\lambda\)-terms, with (possibly) infinite reductions.

\[
Yf \to_p \varnothing \to_p \varnothing \to_p \varnothing \to_p \ldots
\]

Formally, it is defined by metric completion on the syntactic trees [Kennaway et al. 1997] (the original definition) or by coinduction [Endrullis and Polonsky 2013] — we use the latter.

Our setting is a particular infinitary \(\lambda\)-calculys called \(\Lambda^{01}_m\) (not all infinite terms and reductions are authorized).

**... and finite approximants**

Idea: a program \(p : x \mapsto p(x)\) will be approximated by a sum of multilinear programs

\[
[x_1; \ldots; x_n] \mapsto \sum_{i \in N} \nu(X_i) x_1^{\nu(X_1)} \cdots x_n^{\nu(X_n)}
\]

where each linear variable \(x_i\) is used exactly once. For example:

\[
\begin{align*}
\text{function } (x) \{ & \text{ if } (x == 0) \{ \text{ return } x + 2; \} \text{ else } \} \mapsto \sum_{i \in N} \nu(X_i) x_1^{\nu(X_1)^i} \cdots x_n^{\nu(X_n)^i} \\
\text{function } (x_1, x_2) \{ & \text{ if } (x_1 == 0) \{ \text{ return } x_2 + 2; \} \text{ else } \} \mapsto \sum_{i \in N} \nu(X_1)^i x_1^{\nu(X_1)^i} \cdot x_2^{\nu(X_2)^i} \\
\text{function } (x_1, x_2) \{ & \text{ if } (x_2 == 0) \{ \text{ return } x_1 + 2; \} \text{ else } \} \mapsto \sum_{i \in N} \nu(X_2)^i x_1^{\nu(X_1)^i} \cdot x_2^{\nu(X_2)^i}
\end{align*}
\]

This is the Taylor expansion \(\mathcal{T}(\cdot)\) of \(\lambda\)-terms [Ehrhard and Regnier 2008]. It originates from linear logic, and is defined as a "real" Taylor expansion in the differential \(\lambda\)-calculus [Ehrhard and Regnier 2003]. Formally, approximants look like:

\[
\begin{align*}
\begin{array}{c}
\text{Great properties} \\
\text{The reduction } \longrightarrow, \text{ on approximants is weakly normalising and strongly confluent}
\end{array}
\end{align*}
\]

**The infinitary reduction is simulated by the finitary approximation!**

**The key theorem**

In [C. and Vaux Auclair, under review], we show:

**Theorem (simulation)**

For all \(M, N \in \Lambda_m^{01}\),
if \(M \longrightarrow^* N\) then \(\mathcal{T}(M) \longrightarrow^* \mathcal{T}(N)\).

This enables us to retrieve the crucial computation theorem that existed in the finite setting [Ehrhard and Regnier 2006] and has been fruitfully exploited in many situations [Barbarossa and Manzonetto 2020].

**Corollary (commutation)**

For all \(M \in \Lambda_m^{01}\), nf(\(\mathcal{T}(M)\)) = \(\mathcal{T}(\text{nf}(M))\).

**New characterisations**

Using our approximation framework, we are able to adapt two characterisations of normalisation properties ("termination" properties) to the infinitary setting.

**Characterisation of head-normalisability**

\(M \in \Lambda_m^{01}\) is head-normalisable if there exists \(s \in \mathcal{T}(M)\) such that \(\text{nf}(s) \neq 0\).

**Characterisation of normalisability**

\(M \in \Lambda_m^{01}\) is (infinitarily) normalisable if for all \(d \in \mathbb{N}\), there exists \(s \in \mathcal{T}(M)\) such that \(\text{nf}(s)\) contains a \(d\)-positive term.

**Reassuring corollaries**

As corollaries, classical \(\lambda\)-calculus results can be extended to the infinitary setting (which was already known, with more complicated proofs). This shows that \(\Lambda_m^{01}\) "behaves well" and is a reasonable setting!

**Corollary (solvable terms)**

\(M \in \Lambda_m^{01}\) is solvable iff it is head-normalisable.

**Corollary (properties of \(\longrightarrow^*\))**

\(\longrightarrow^*\) is confluent and has unique normal forms.

**Corollary (genericity)**

If \(M \in \Lambda_m^{01}\) is unsolvable, \(C(\cdot)\) is a context and \(C(M)\) has a normal form \(C^*\), then for any \(N \in \Lambda_m^{01}\), \(C(N) \longrightarrow^* C^*\).

**What comes next?**

**Further work 1.** We work in the \(\Lambda_m^{01}\) fragment, which is well-suited to Taylor approximation: what about wilder settings (\(\Lambda_m^{01}\) or \(\Lambda_m^{11}\))? For this, we need to design a new language of approximants...

**Further work 2.** What about the converse of the simulation theorem?

**Conjecture (conservativity)**

For all \(M, N \in \Lambda_m^{01}\), if \(\mathcal{T}(M) \longrightarrow^* \mathcal{T}(N)\) then \(M \longrightarrow^* \lambda N\).

**A few words of conclusion**

**Summary**

**Lesson 1.** The Taylor expansion provides a powerful approximation theory for the infinitary \(\lambda\)-calculus (ie. for the study of the limit behaviour of looping programs).

**Lesson 2.** The \(\Lambda_m^{01}\) infinitary \(\lambda\)-calculus is a "natural" setting to define the Taylor expansion of (finitary) \(\lambda\)-terms.

See all the details in the paper: arXiv:221105688...