Linear approximation and Taylor expansion of λ-terms

F. Olimpieri

Aix-Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France

The pure λ-calculus

λ-terms

We inductively define Λ :

- if $x \in \mathcal{V}$ then $x \in \Lambda$;
- If $M \in \Lambda$, then $\lambda x.M \in \Lambda$;
- if $M, N \in \Lambda$, then $MN \in \Lambda$.

- $\lambda x.M$ stands for $x \mapsto M$.
- We can model *functional evaluation*:

$$(\lambda x.M)N \to M[N/x]$$

The pure λ-calculus

λ-terms

We inductively define Λ :

- if $x \in \mathcal{V}$ then $x \in \Lambda$;
- If $M \in \Lambda$, then $\lambda x.M \in \Lambda$;
- if $M, N \in \Lambda$, then $MN \in \Lambda$.

- $\lambda x.M$ stands for $x \mapsto M$.
- We can model *functional evaluation*:

$$(\lambda x.M)N \to M[N/x]$$

The pure λ-calculus

λ-terms

We inductively define Λ :

- if $x \in \mathcal{V}$ then $x \in \Lambda$;
- If $M \in \Lambda$, then $\lambda x.M \in \Lambda$;
- if $M, N \in \Lambda$, then $MN \in \Lambda$.

- $\lambda x.M$ stands for $x \mapsto M$.
- We can model *functional evaluation*:

$$(\lambda x.M)N \to M[N/x]$$

Linearity

Intuitive Definition

A function f is *linear* when it uses only once its argument during the computation.

Linearity for functional evaluation:

- The *identity function* is linear. Let $M \in \Lambda$, then $(\lambda x.x)M \to M$.
- The *copy function* is non-linear. Let $M \in \Lambda$, then $(\lambda x.xx)M \to MM$.

Linearity

Intuitive Definition

A function f is *linear* when it uses only once its argument during the computation.

Linearity for functional evaluation:

- The *identity function* is linear. Let $M \in \Lambda$, then $(\lambda x.x)M \to M$.
- The *copy function* is non-linear. Let $M \in \Lambda$, then $(\lambda x.xx)M \to MM$.

Linearity

Intuitive Definition

A function f is *linear* when it uses only once its argument during the computation.

Linearity for functional evaluation:

- The *identity function* is linear. Let $M \in \Lambda$, then $(\lambda x.x)M \to M$.
- The *copy function* is non-linear. Let $M \in \Lambda$, then $(\lambda x.xx)M \to MM$.

Linear approximation of λ-terms

Linear logic leads to the introduction of a *resource sensitive* approximation of programs.

Intuitively, a n-linear approximant of a term M is a version of it that uses exactly n times the argument under evaluation.

We denote as $T(M)$ the set of linear approximants of M.

Lemma

Let $M \in \Lambda$ and $s \in T(M)$. If $s \to t$ then there exists $N \in \Lambda$ such that $t \in T(N)$ and $M \to N$.

Linear approximation of λ-terms

Linear logic leads to the introduction of a *resource sensitive* approximation of programs.
Intuitively, a n-linear approximant of a term M is a version of it that uses exactly n times the argument under evaluation.
We denote as $T(M)$ the set of linear approximants of M.

Lemma

Let $M \in \Lambda$ and $s \in T(M)$. If $s \to t$ then there exists $N \in \Lambda$ such that $t \in T(N)$ and $M \to N$.

Linear approximation of λ-terms

Linear logic leads to the introduction of a *resource sensitive* approximation of programs.
Intuitively, a n-linear approximant of a term M is a version of it that uses exactly n times the argument under evaluation.
We denote as $T(M)$ the set of linear approximants of M.

Lemma

Let $M \in \Lambda$ and $s \in T(M)$. If $s \to t$ then there exists $N \in \Lambda$ such that $t \in T(N)$ and $M \to N$.

Some results

Theorem

Let $M \in \Lambda$. M is computationally meaningful iff the computation for some $s \in T(M)$ ends.

We can define a *Taylor expansion* for λ-terms:

Taylor formula

$$\Theta(M) = \sum_{s \in T(M)} \frac{1}{m(s)} s$$

Some results

Theorem

Let $M \in \Lambda$. M is computationally meaningful iff the computation for some $s \in T(M)$ ends.

We can define a *Taylor expansion* for λ-terms:

Taylor formula

$$\Theta(M) = \sum_{s \in T(M)} \frac{1}{m(s)} s$$