Continuous and linear approximations for the λ-calculus

- Me
- Feel free to interrupt
- Feel free to ask questions later
- Notes are online

Outline

1. Böhm trees & continuous approximation
2. Back to intersection types
3. Linear approximation, aka Taylor expansion

1 Böhm trees & continuous approximation

Recall that λ-terms are either

→ $\lambda x_1 \dots x_n . y M_1 \dots M_m$, hnf ⤳ idea: stable prefix

→ $\lambda x_1 \dots x_m . \underbrace{(\lambda x.P) Q}_{\text{head redex.}} M_1 \dots M_n$

Def: For $M \in \Lambda$, its Böhm tree $BT(M)$ is def. by: [Barendregt 1977]

$$BT(M) := \begin{cases} \lambda \vec{x}. y\, BT(M_1) \dots BT(M_n) & \text{if } M \to_\beta^* \lambda \vec{x}. y \vec{M} . \\ \perp & \text{otherwise.} \end{cases}$$

Remarks:

* Tree-like representation
* it's a $\Lambda\perp$-term (add a constant $\perp$ to the syntax, denote by $\Lambda_\perp$ the obtained set).

Examples: Böhm trees of

- I
- any normal term
- Ω
- Y
- YK

* it can be infinite!
* $\to_\beta^*$ makes the definition ambiguous ... but it's fine:

Thm (head normalization): M has a hnf through $\to_\beta$ iff it has a hnf through $\to_h$

... so we can replace $\to_\beta^*$ with $\to_h^*$ in the definition.

Theorem: $\mathcal{B} := \{M = N \mid BT(M) = BT(N)\}$ is a λ-theory,

i.e. $=_\beta \subseteq \mathcal{B}$ and $\underbrace{M =_{\mathcal{B}} N \Rightarrow C[M] =_{\mathcal{B}} C[N]}_{\text{this isn't trivial!}}$.

*

Digression: Consider $\to_{\beta\bot}$ defined by

(the same rules as for $\to_\beta$) + $\dfrac{M \text{ has no hnf}}{M \to_{\beta\bot} \bot}$ $\quad \dfrac{}{\lambda x.\bot \to_{\beta\bot} \bot}$ $\quad \dfrac{}{\bot M \to_{\beta\bot} \bot}$

and $\to^\infty_{\beta\bot}$ defined coinductively by:

$$\frac{M \to^*_{\beta\bot} \lambda\vec{x}.\, P Q_1 \dots Q_n \quad P \to^*_{\beta\bot} P' \quad \triangleright Q_i \to^\infty_{\beta\bot} Q_i'}{M \to^\infty_{\beta\bot} \lambda\vec{x}.\, P' Q_1' \dots Q_n'} \qquad \frac{Q \to^\infty_{\beta\bot} Q'}{\triangleright Q \to^\infty_{\beta\bot} Q'}$$

(i.e. infinite derivations allowed provided infinite branches cross a double rule infinitely often)

then

Thm [Kennaway et al. 1997]
$\to^\infty_{\beta\bot}$ is confluent and all M has $BT(M)$ as a unique normal form.

Exercise (if time permits): Derive $Y \to^\infty_{\beta\bot} \lambda f.\, f f f \dots$

Remark: This is the modern presentation.
In the 90s the Dutch School did it with ordinal-indexed sequences of reductions + (topology) convergence properties

*

Def The approximation order $\sqsubseteq$ is defined on $\Lambda_\bot$ by
$\bot \sqsubseteq M$ (for all M) + monotonicity of constructors.

Def The set $\mathcal{A}$ of approximants is $\mathcal{A} := \{P \in \Lambda_\bot \text{ in } \beta\bot\text{-nf}\}$. Explicitly,
$$\mathcal{A} \ni P, \dots ::= \bot \mid \lambda\vec{x}.\, y\vec{P}.$$

Def The set $\mathcal{A}(M)$ of the approximants of $M \in \Lambda$ is
$$\mathcal{A}(M) := \{P \in \mathcal{A} \mid \exists M',\ M \to^*_\beta M' \sqsupseteq P\}.$$

Ex: Compute $\mathcal{A}(Y)$, $\mathcal{A}(\Omega)$.

Continuous approximation thm [Wadsworth, Lévy, Hyland, Barendregt]:
$\mathcal{A}(M)$ is directed and $\bigsqcup \mathcal{A}(M) \cong BT(M)$.

This is ill-defined ... unless we work in the ideal completion of $(\Lambda_\bot, \sqsubseteq)$.

this is the important part! Recall that directed = has binary joins.

Why "continuous"?

→ Each $P \in \mathcal{A}(M)$ defines an open set
→ $BT(M)$ is the intersection of these open sets = the limit

$$M \xrightarrow[\text{computation}]{\text{total}} {}^{\infty}_{\beta\bot}\ BT(M)$$

— approxim. ↓ - - - - - - - - - - - - ↑ limit — — —

$$M \xrightarrow[\text{partial comput.}]{} {}^{*}_{\beta}\ M' \sqsupseteq P$$

defines open sets

2 Back to intersection types ...

Recall our intersection type system:

Types: $A, B, \ldots ::= \alpha \in \mathbb{A} \mid \bar{A} \to B$ for a set $\mathbb{A}$ of atoms

$\bar{A}, \bar{B}, \ldots ::= \{A_1, \ldots, A_n\}$ for $n \in \mathbb{N}$

Contexts: Γ: Variables → Sets of types. $\Gamma, x:\bar{A}$ is such that $\Gamma(x) = \emptyset$ and $(\Gamma, x:\bar{A})(x) = \bar{A}$.

Typing rules:

$$\frac{A \in \bar{A}}{\Gamma, x:\bar{A} \vdash x:A}\ (ax) \qquad \frac{\Gamma \vdash M:A_1 \quad \ldots \quad \Gamma \vdash M:A_n}{\Gamma \vdash M:\{A_1, \ldots, A_n\}}\ (!)$$

$$\frac{\Gamma, x:\bar{A} \vdash M:B}{\Gamma \vdash \lambda x.M : \bar{A} \to B}\ (\lambda) \qquad \frac{\Gamma \vdash M:\bar{A} \to B \quad \Gamma \vdash N:\bar{A}}{\Gamma \vdash MN:B}\ (@)$$

Exercise: What are all the possible types of Y?

Easier exercise: For all $\alpha_0, \ldots, \alpha_n \in \mathbb{A}$, derive

$$\vdash Y : \{\{\} \to \alpha_n, \{\alpha_n\} \to \alpha_{n-1}, \ldots, \{\alpha_1\} \to \alpha_0\} \to \alpha_0 .$$

$\boxed{\text{For } n=0:}$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{f:[[]\multimap\alpha_0]\vdash f:[]\multimap\alpha_0}(ax)\qquad \dfrac{}{f\vdash xx:[]}(!)}{f,x:[]\vdash f(xx):\alpha_0}(@)}{f\vdash \lambda x.f(xx):[]\multimap\alpha_0}(\lambda)\qquad \dfrac{}{f\vdash \lambda x.f(xx):[]}(!)}{\dfrac{f:[[]\multimap\alpha_0]\vdash(\lambda x.f(xx))\lambda x.f(xx):\alpha_0}{\vdash Y:[[]\multimap\alpha_0]\multimap\alpha_0}(\lambda)}(@)$$

this context is equal to just $f:[[]\multimap\alpha_0]$.

I already use the non-idempotent notation, sorry!

$\boxed{\text{For } n>0:}$ Define $\overline{B}_n := []$ and $\forall 0<i\leq n$, $\overline{B}_{i-1} := \overline{B}_i + [\overline{B}_i\multimap\alpha_i]$.

$$\dfrac{\dfrac{\dfrac{[\alpha_1]\multimap\alpha_0\in\overline{A}}{f,x\vdash f:[\alpha_1]\multimap\alpha_0}(ax)\qquad \dfrac{\dfrac{\dfrac{\overline{B}_1\multimap\alpha_1\in\overline{B}_0}{f,x\vdash x:\overline{B}_1\multimap\alpha_1}(ax)\qquad\dfrac{\overline{B}_1\subset\overline{B}_0}{f,x\vdash x:\overline{B}_1}(ax,!)}{f,x\vdash xx:\alpha_1}(@)}{f,x\vdash xx:[\alpha_1]}(!)}{f:\overline{A},x:\overline{B}_0\vdash f(xx):\alpha_0}(@)}{f:\overline{A}\vdash\lambda x.f(xx):\overline{B}_0\multimap\alpha_0\quad(*)}(\lambda)$$

$$\dfrac{\dfrac{(*)}{f\vdash\lambda x.f(xx):\overline{B}_1\multimap\alpha_1}\qquad \dfrac{\vdots}{f\vdash\lambda x.f(xx):\overline{B}_1}}{f:\overline{A}\vdash\lambda x.f(xx):\overline{B}_0}\left(\text{def. of }\overline{B}_0\right)$$

where

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[]\multimap\alpha_n\in\overline{A}}{f\vdash f:[]\multimap\alpha_n}(ax)\qquad\dfrac{}{f\vdash xx:[]}(!)}{f\vdash f(xx):\alpha_n}(@)}{f\vdash\lambda x.f(xx):[]\multimap\alpha_n}(\lambda)}{f\vdash\lambda x.f(xx):\overline{B}_{n-1}}(!)}{\vdots}$$

$$\dfrac{\dfrac{f:\overline{A}\vdash\lambda x.f(xx):\overline{B}_0\multimap\alpha_0\quad(*)\qquad f:\overline{A}\vdash\lambda x.f(xx):\overline{B}_0}{f:\overline{A}\vdash(\lambda x.f(xx))\lambda x.f(xx):\alpha_0}(@)}{\vdash Y:\underbrace{[[]\multimap\alpha_n,[\alpha_n]\multimap\alpha_{n-1},\dots,[\alpha_1]\multimap\alpha_0]}_{\overline{A}}\multimap\alpha_0}(\lambda)$$

One cut-elimination step replaces with in .
By denoting $\Delta_f := \lambda x.fxx$, we obtain:

$$\dfrac{\dfrac{\dfrac{[\alpha_1]\multimap\alpha_0\in\overline{A}}{f\vdash f:[\alpha_1]\multimap\alpha_0}(ax)\qquad\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{f,x\vdash f(xx):\alpha_1}}{f\vdash\Delta_f:\overline{B}_1\multimap\alpha_1}\qquad\dfrac{\vdots}{f\vdash\Delta_f:\overline{B}_1}}{f\vdash\Delta_f\Delta_f:\alpha_1}(@)}{f\vdash\Delta_f\Delta_f:[\alpha_1]}(!)}{f:\overline{A}\vdash f(\Delta_f\Delta_f):\alpha_0}(@)}{\vdash\lambda f.f(\Delta_f\Delta_f):\overline{A}\multimap\alpha_0}(\lambda)$$

if we eliminate all cuts, we obtain:

$$\dfrac{\dfrac{[\,] \multimap \alpha_n \in \bar{A}}{f:\bar{A} \vdash f:[\,]\multimap\alpha_m}\,(\mathrm{ax}) \qquad \dfrac{}{f:\bar{A} \vdash \Delta_f\Delta_f : [\,]}\,(!)}{f:\bar{A} \vdash f(\Delta_f\Delta_f) : \alpha_m}\,(@)$$

$$\vdots$$

$$\dfrac{\dfrac{\dfrac{[\alpha_1]\multimap\alpha_0 \in \bar{A}}{f:\bar{A} \vdash f:[\alpha_1]\multimap\alpha_0}\,(\mathrm{ax}) \qquad \dfrac{}{f:\bar{A} \vdash f^{n}(\Delta_f\Delta_f) : \alpha_1}}{f:\bar{A} \vdash f^{n+1}(\Delta_f\Delta_f):\alpha_0}\,(@,!)}{\vdash \lambda f.\, f^{n+1}(\Delta_f\Delta_f) : \bar{A}\multimap\alpha_0}\,(\lambda)$$

This is exactly "the" typing corresponding to the approximant $\lambda f.\, f^{n+1}\bot$ of Y! in fact:

> **Thm** [Coppo, Dezani et al.]
> There is a derivation $\Gamma \vdash M:A$ iff $\exists P \in \mathcal{A}(M)$, $\Gamma \vdash P : A$.

in the previous example, the $\bot$ in $\lambda f.\, f^{n+1}\bot \in \mathcal{A}(Y)$
corresponds to $\vdash \Delta_f\Delta_f : [\,]$ in the (normalised) derivation $\vdash \lambda f.\, f^{n+1}(\Delta_f\Delta_f) : \bar{A}\multimap\alpha_0$,
which originates from $\vdash xx : [\,]$ in the (original) derivation $\vdash Y : \bar{A}\multimap\alpha_0$.

This is not always possible. Example:

$$(\lambda x.\, fxx)\,\lambda x.\, gx \;\to_\beta\; f(\lambda x.\, gx)(\lambda x.\, gx) \;\sqsupseteq\; f(\lambda x.\, gx)\,\bot$$

$$\dfrac{\dfrac{\dfrac{f,g,x \vdash f:[\alpha]\to[\alpha]\to\alpha \quad f,g,x\vdash x:[\alpha] \quad f,g,x\vdash x:[\alpha]}{f,g,x:[\alpha] \vdash fxx : \alpha}}{f,g \vdash \lambda x.\, fxx : [\alpha]\to\alpha} \qquad \dfrac{???}{f,g \vdash \lambda x.\, gx : \alpha}}{f,g \vdash (\lambda x.\, fxx)\,\lambda x.\, gx : \alpha}$$

$\leftsquigarrow$

$$\dfrac{\dfrac{f,g \vdash f:[\alpha]\to[\alpha]\to\alpha \quad \dfrac{\dfrac{f,g,x\vdash g:[\beta]\to\beta \quad f,g,x\vdash x:\beta}{f,g,x:[\beta]\vdash gx:\beta}}{f,g\vdash\lambda x.\, gx:\alpha}}{f,g \vdash f(\lambda x.\, gx) : [\alpha]\to\alpha} \qquad \dfrac{\dfrac{f,g\vdash g:[\,]\to\beta \quad f,g\vdash x:[\,]}{f,g\vdash gx:\beta}}{f,g\vdash\lambda x.\, gx:\alpha}}{f:[\alpha]\to[\alpha]\to\alpha,\ g:\begin{bmatrix}[\,]\to\beta,\\ [\beta]\to\beta\end{bmatrix} \vdash f(\lambda x.\, gx)(\lambda x.\, gx):\alpha}$$

with $\alpha = [\beta]\multimap\beta$

This is why we need <u>non-idempotent</u> types!

Recall the typing system:

Types: $A, B, \dots ::= \alpha \in \mathbb{A} \mid \bar{A} \multimap B$

$\bar{A}, \bar{B}, \dots ::= [M_1, \dots, M_n]$ (a multiset)

Contexts, rules: adapt the idempotent system.

Let us try to extract a syntax of terms from non-idempotent derivations... The key observation is that @ rules (i.e. applications) can receive several different derivations of the same argument judgement:

$$\dfrac{\dfrac{\dfrac{}{\vdash x:\alpha}\qquad\dfrac{}{\vdash x:\alpha}}{\vdots}}{f,g\vdash \lambda x.fxx:[\alpha]\to\alpha}\qquad \dfrac{}{f,g\vdash \lambda x.gx:\alpha}\qquad \dfrac{}{f,g\vdash \lambda x.gx:\alpha}$$

$$\overline{\qquad f,g \vdash (\lambda x.fxx)(\lambda x.gx) : \alpha \qquad}\ (@,!)$$

$$\to_\beta \quad \dfrac{\dfrac{}{}\qquad\dfrac{}{}}{\dfrac{\vdots}{f,g\vdash f(\lambda x.gx)\,\lambda x.gx:\alpha}}$$

Let's do this in our term syntax:

$$(\lambda x.\, f[x][x])\,[\lambda x.g[x], \lambda x.g[\,]] \longrightarrow f[\lambda x.g[x]][\lambda x.g[\,]]$$

$$\downarrow$$

$$f(\lambda x.gx)(\underbrace{\lambda x.\bot}_{=\bot})$$

it's the idea behind the linear approximation of the λ-calculus!

3 The linear approximation (aka Taylor expansion)

$$\begin{tikzcd}
M \arrow[rr, "\text{total computation}", "\infty"' pos=1] \arrow[d, leftrightarrow, "\text{approx. via Taylor series}"'] & & BT(M) \arrow[d, leftrightarrow, "\text{Taylor series}"] \\
s \arrow[rr, "\text{partial computation}", "*"' pos=1] & & nf(s)
\end{tikzcd}$$

These guys live in the resource λ-calculus. [Ehrhard-Regnier 2008]

(simple) terms $\quad \Lambda_r \ni s,t,\dots ::= x \mid \lambda x.s \mid s\bar{t} \qquad x \in \mathcal{V}$

$\Lambda_r^! \ni \bar{s},\bar{t},\dots ::= [s_1,\dots,s_n] \qquad n \in \mathbb{N}$

finite resource sums $\quad \mathbb{N}[\Lambda_r] := \{$ formal sums of elements of Λ_r with coefficients in $\mathbb{N}$ $\}$

multilinear substitution:

$$s\left\langle [t_1,\dots,t_n]/x \right\rangle := \begin{cases} \sum_{\sigma\in\mathfrak{S}(n)} s\left[t_1/x_{\sigma 1},\dots,t_n/x_{\sigma n}\right] & \text{if } (*) \\ 0 & \text{otherwise} \end{cases}$$

(*) if n = the number of occurrences of x in s
then $s\left[t_1/x_{\sigma 1},\dots\right]$ is the term obtained by replacing the $\sigma(i)^{th}$ occurrence with t_i, $\forall i$.

resource reduction

$$\frac{}{(\lambda x.s)\bar{t} \to_r s\langle \bar{t}/x \rangle} \quad + \text{ lifting to context}$$

$$+ \quad \frac{s \to_r S'}{s+T \to_r S'+T}$$

Prop: $\to_r$ is confluent and strongly normalising.

Exercise: know the multiset ordering? then try to prove normalisation!

if $(X,\leq)$ is a well-founded order, then so is $(!X, \leq_!)$ where
- $!X$ is the set of multisets of elements of X
- $\bar{x} \leq_! \bar{y}$ if $\bar{x}$ is obtained from $\bar{y}$ by deleting at least one element, and adding smaller ones.

Consequence: for $S \in \mathbb{N}[\Lambda_r]$ we can define $nf(S)$

Def: For $M \in \Lambda_\perp$ we define its Taylor expansion by:

$$\mathcal{T}(x) := \{x\}$$

$$\mathcal{T}(\lambda x.M) := \lambda x.\mathcal{T}(M) := \{\lambda x.s \mid s \in \mathcal{T}(M)\}$$

$$\mathcal{T}(MN) := \mathcal{T}(M)\,\mathcal{T}(N)^! := \left\{ s[t_1 \dots t_n] \;\middle|\; \begin{array}{l} s \in \mathcal{T}(M) \\ n \in \mathbb{N} \\ t_1 \dots t_n \in \mathcal{T}(N) \end{array} \right\}.$$

$$\mathcal{T}(\perp) := \emptyset.$$

This was originally presented as

$$\mathcal{T}(M) \cdot \sum_{n \in \mathbb{N}} \frac{\mathcal{T}(N)^!}{n!} \dots$$

in fact it is really a Taylor expansion!

How to reduce sets of resource terms?

if $\mathcal{X} = \bigcup_{i \in I} \{s_i\}$ and $\mathcal{Y} = \bigcup_{i \in I} |S_i'|$

with $\forall i,\ s_i \rightarrow_r^* S_i'$

then we write $\mathcal{X} \tilde{\rightarrow}_r \mathcal{Y}$.

the elements of the sum S_i'

We write $\tilde{nf}\left(\bigcup_{i \in I} \{s_i\}\right) := \bigcup_{i \in I} |nf(s_i)|$.

Thm: [Ehrhard–Regnier 2006, 2008 ; C.–Vaux 2023]

if $M \rightarrow_{\beta\perp}^{\infty} N$ then $\mathcal{T}(M) \tilde{\rightarrow}_r \mathcal{T}(N)$.

in particular, $\tilde{nf}(\mathcal{T}(M)) = \mathcal{T}(BT(M))$.

not really defined so far...

You can take $\mathcal{T}(BT(M)) = \bigcup_{P \in \mathcal{A}(M)} \mathcal{T}(P)$.

Corollaries: All the statements presented as "theorems" in this lecture!!!

Example:

$$\Upsilon(BT(Y)) = \bigcup_{P \in \mathcal{A}(Y)} \Upsilon(P) = \underbrace{\Upsilon(\bot)}_{=\emptyset} \cup \underbrace{\bigcup_{n \in \mathbb{N}} \Upsilon(\lambda f. f^{n+1} \bot)}$$

$n=0$: $\{\lambda f. f[\,]\}$

$n=1$: $\{\lambda f. f[\,], \lambda f. f[f[\,]], \lambda f. f[f[\,], f[\,]], \ldots\}$

$\vdots$

in particular the intersection type derivations we saw before can also type the following approximants:

$$\lambda f.\left(\lambda x. f[x\, x^{n-1}]\right)\left[\lambda x. f[x x^{n-2}], \ldots, \lambda x. f[x x^0], \lambda x. f[\,]\right] \to_r^* \lambda f. \underbrace{f[f[\ldots f[\,] \ldots]]}_{n+1 \text{ times}}$$

where x^n denotes $[\underbrace{x, \ldots, x}_{n \text{ times}}]$.

in fact,

Thm [de Carvalho 2007]: For $M \in \Lambda$,

$$\left\{ \begin{array}{l} \text{derivations of } \Gamma \vdash M : A \\ \text{in a non-idempotent system} \end{array} \right\} \Big/ \begin{array}{l} \text{some equivalence} \\ \text{relation induced} \\ \text{by permutations} \end{array} \cong \left\{ \begin{array}{l} s \in \Upsilon(M) \text{ such that} \\ nf(s) \neq 0 \end{array} \right\}$$

a small adaption would be useful, do you see which one?
Hint: there's a problem with weakening.

if i had to write an exam: do the whole lecture, but lazily (i.e. replacing head with weak head reductions!)