Infinite Sequences as Intersection Types: a New Answer to Klop's Problem

Pierre VIAL
Journées de rentrée
$\perp$-IRIF

October 12, 2017

KLOP'S PROBLEM

- **Head** or **weak normalization** etc. have been *statically* characterized by various **intersection type systems (ITS)**.
- **Klop's Problem:** can the set of ∞-WN terms be characterized by an ITS ?
  Def: t is ∞-WN iff its Böhm tree does not contain $\perp$
-
  - **Tatsuta [07]:** an inductive ITS cannot do it.
  - Can a coinductive ITS characterize the set of ∞-WN terms?

KLOP'S PROBLEM

- **Head** or **weak normalization** etc. have been *statically* characterized by various **intersection type systems (ITS)**.
- **Klop's Problem:** can the set of ∞-WN terms be characterized by an ITS ?
  Def: t is ∞-WN iff its Böhm tree does not contain $\bot$
-
  - **Tatsuta** [**07**]**:** an inductive ITS cannot do it.
  - Can a coinductive ITS characterize the set of ∞-WN terms?
- **YES**, when inter. types = **sequences** and w/ a **validity criterion**.

Plan

Infinitary Normalization

Finite Intersection Type Systems

Back to Klop's Problem

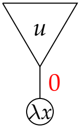
INFINITARY NORMALIZATION

FINITE INTERSECTION TYPE SYSTEMS

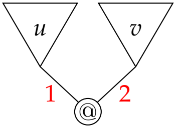
BACK TO KLOP'S PROBLEM

WEAK NORMALIZATION (FINITE CASE)

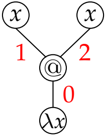
Variable x



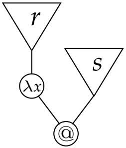
Abstraction $\lambda x.u$



Application $u\,v$

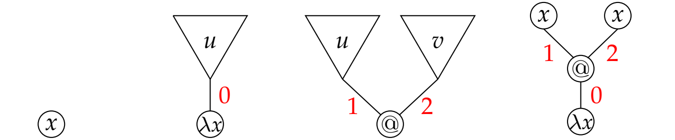


Example: $\Delta = \lambda x.xx$

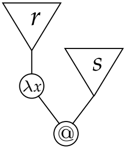


Redex $(\lambda x.r)s$

WEAK NORMALIZATION (FINITE CASE)



Variable x **Abstraction** $\lambda x.u$ **Application** $u\,v$ *Example:* $\Delta = \lambda x.xx$



Redex $(\lambda x.r)s$

β**-reduction:** $(\lambda x.r)s \to r[s/x]$ (= exec. step): replacing x with s in r

Normal form (NF) (= final state): a term without redexes.

Definition: A term t is (finitary) **weakly normalizing (WN)** if there is a (finite) *reduction path* from t to a **normal form**.

PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

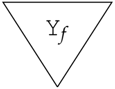
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\quad$ $\mathtt{Y}_f := \Delta_f \Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \dots \to f^n(\mathtt{Y}_f) \to \dots \to^\infty f^\omega$



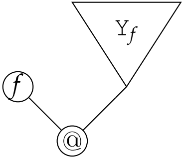
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\quad \mathtt{Y}_f := \Delta_f \Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \dots \to f^n(\mathtt{Y}_f) \to \dots \to^\infty f^\omega$



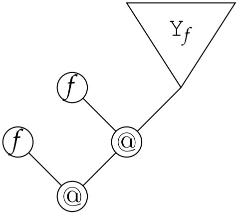
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \ldots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\qquad \mathtt{Y}_f := \Delta_f \Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \ldots \to f^n(\mathtt{Y}_f) \to \ldots \to^\infty f^\omega$



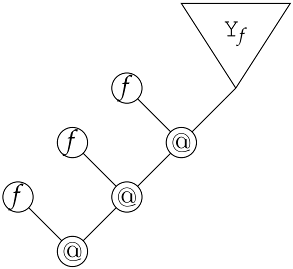
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\quad \mathtt{Y}_f := \Delta_f \Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \dots \to f^n(\mathtt{Y}_f) \to \dots \to^\infty f^\omega$



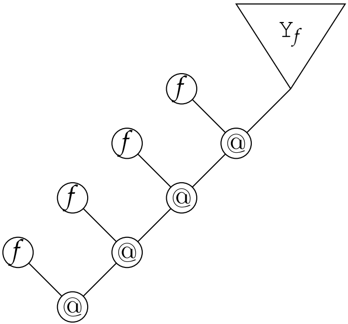
Productive vs. Unproductive Reduction

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\quad$ $\mathtt{Y}_f := \Delta_f \Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \dots \to f^n(\mathtt{Y}_f) \to \dots \to^\infty f^\omega$



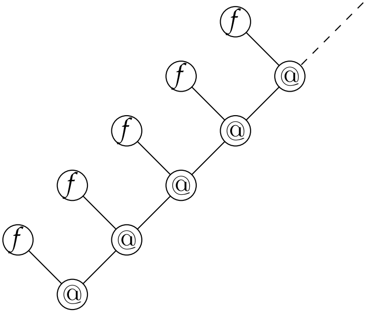
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\mathtt{Y}_f := \Delta_f\Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \dots \to f^n(\mathtt{Y}_f) \to \dots \to^\infty f^\omega$



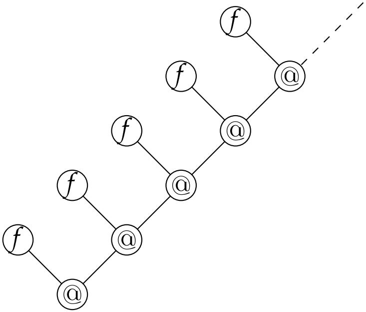
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \ldots$

Productive reduction: $\Delta_f := \lambda x.f(xx) \qquad \mathtt{Y}_f := \Delta_f\Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \ldots \to f^n(\mathtt{Y}_f) \to \ldots \to^\infty f^\omega$



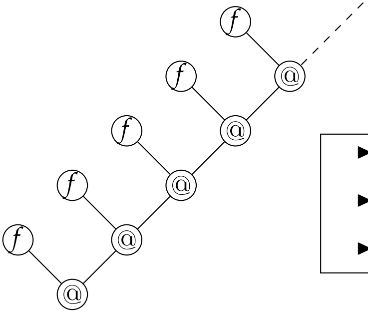
PRODUCTIVE VS. UNPRODUCTIVE REDUCTION

Unproductive reduction: $\Delta = \lambda x.x\,x,\ \Omega = \Delta\,\Delta$

$\Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \Omega \to \dots$

Productive reduction: $\Delta_f := \lambda x.f(xx)$ $\qquad \mathtt{Y}_f := \Delta_f \Delta_f$ "Curry f"

$\mathtt{Y}_f \to f(\mathtt{Y}_f) \to f^2(\mathtt{Y}_f) \to f^3(\mathtt{Y}_f) \to f^4(\mathtt{Y}_f) \to \dots \to f^n(\mathtt{Y}_f) \to \dots \to^{\infty} f^{\omega}$



- $\mathtt{Y}_f$ not WN
- $\mathtt{Y}_f$ is ∞-WN
- ∞-NF: $f^{\omega} = f(f^{\omega})$

INFINITE TERMS

- Infinite λ-term (infinite labelled trees).
- Infinite NF: an infinite tree without redex *e.g.* f^ω.
- *Productive* reduction sequence of infinite length (**strongly converging reduction sequence)**
  e.g. $\mathtt{Y}_f \to f(\mathtt{Y}_f)\ldots$, not $\Omega \to \Omega \ldots$
- A term t is ∞**-WN** if there is a reduction path to an ∞-NF.
- *Remark:* several variants of the infinitary λ-calculi.
  The one here: Λ^{001} in [Klop et al., 96] = **restriction**.

INFINITARY NORMALIZATION

FINITE INTERSECTION TYPE SYSTEMS

BACK TO KLOP'S PROBLEM

INTERSECTION TYPES

- Simple type systems (STS): Typable $\Rightarrow$ Normalizing
  vs. intersection t.s. (ITS): Typable $\Leftrightarrow$ Normalizing.

- STS: a variable x can be assigned only one type (that can be used several times)
  vs. ITS: a variable can be typed with several types
  e.g. $x : A \wedge B \wedge C \wedge B$.

- Typing an application in a STS (without contexts):

$$\frac{t : A \to B \qquad u : A}{t\,u : B}\ \texttt{app}$$

Intuition: t is a function from A to B, that can be fed with an arg. u of type A.

- *Example:* • STS: xx usually *not* typable.
  • ITS: if x assigned $A \wedge (A \to B)$, then $x\,x : B$ derivable.

SUBJECT REDUCTION AND SUBJECT EXPANSION

A good intersection type system should enjoy:

- **Subject Reduction (SR)**:
  Typing is stable under reduction.
- **Subject Expansion (SE)**:
  Typing is stable under anti-reduction.
  SE is usually not verified by Simple Type Systems (even Higher Order)

SR/SE pivotal to prove that normalization is characterized by the i.t.s.
(*e.g.* HN next slides).

HEAD NORMALIZATION

- Every term has a *head variable* or a *head redex*.
  - 1st case: **head normal form (HNF)**
  - 2nd case: head reducible term.
- HNF = $\lambda x_1 \ldots \lambda x_p . x\, t_1 \ldots t_q$
  vs. h. red. term $\lambda x_1 \ldots \lambda x_p . (\lambda x.r) s\, t_1 \ldots t_q$

  Intuition: head variable/redex = most "shallow" variable/redex.
- t is **head normalizing (HN)** if $\exists$ reduction path from t to a HNF.
  $\rightsquigarrow$ *HN^{ion}: eliminating (only) one redex.*
- The **head reduction strategy**: reducing head redexes (while it is possible).

EXAMPLE: CHARACTERIZING HN

In order to prove, that, in an ITS (satisfying SR and SE), a term t is typable iff it is **head normalizing (HN)**:

- Using Subject Reduction, we prove a Termination Property: if t is typable, then the head reduction strategy terminates for t.
  $\rightsquigarrow$ If t is typable, then t is HN.
- We type the **head normal forms (HNF)** and proceed by expansion
  $\rightsquigarrow$ If t is HN, then t is typable.

NON-IDEMPOTENT INTERSECTION

- In this talk, intersection is associative, commutative, but non-idempotent (**linearity**):

$$A \wedge A \wedge B = A \wedge B \wedge A \neq A \wedge B$$

- Paradigm: **multisets**
  - $[a, b, a] = [a, a, b] \neq [a, b]$
  - $[a, b, b] + [a, c] := [a, a, b, b, c]$
  - $[a]_3 := [a, a, a]$
  - $[a_i]_{i \in \{1,\dots,n\}}$ stands for $[a_1, \dots, a_n]$.
  - $[\,]$ is the **empty multiset** (neutral for $+$).

- *Metavariables:* When intersection is non-idempotent, the types A, B become σ, τ and $A \wedge B \wedge A$ becomes $[\sigma, \tau, \sigma]$.

TYPING RULES OF $\mathscr{R}_0$ (GARDNER/DE CARVALHO)

Strict Types (τ, σ_i): $\tau, \sigma_i \quad := \quad o \in \mathscr{O} \;\mid\; [\sigma_i]_{i \in I} \to \tau$

Multiset types: $[\sigma_i]_{i \in I}$ (= inter. type)

Contexts (Γ, Δ): assign *intersection* types to variables.

TYPING RULES OF $\mathscr{R}_0$ (GARDNER/DE CARVALHO)

Strict Types (τ, σ_i)*:* $\tau, \sigma_i \ := \ o \in \mathscr{O} \ | \ [\sigma_i]_{i \in I} \to \tau$

Multiset types: $[\sigma_i]_{i \in I}$ (= inter. type)

Contexts (Γ, Δ)*:* assign *intersection* types to variables.

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma, x : [\sigma_i]_{i \in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i \in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i \in I} \to \tau \qquad (\Delta_i \vdash u : \sigma_i)_{i \in I}}{\Gamma +_{i \in I} \Delta_i \vdash t\, u : \tau}\ \texttt{app}$$

- `app`: u may be typed 1, 2, 3,… or 0 times (*cf.* $\#I$).
- No weakening $\rightsquigarrow$ **relevance**.

TYPING RULES OF $\mathscr{R}_0$ (GARDNER/DE CARVALHO)

Strict Types (τ, σ_i)*:* $\tau, \sigma_i \ := \ o \in \mathscr{O} \ | \ [\sigma_i]_{i \in I} \to \tau$

Multiset types: $[\sigma_i]_{i \in I}$ (= inter. type)

Contexts (Γ, Δ)*:* assign *intersection* types to variables.

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma, x : [\sigma_i]_{i \in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i \in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i \in I} \to \tau \qquad (\Delta_i \vdash u : \sigma_i)_{i \in I}}{\Gamma +_{i \in I} \Delta_i \vdash t\, u : \tau}\ \texttt{app}$$

- `app`: u may be typed 1, 2, 3,… or 0 times (*cf.* $\# I$).
- No weakening $\rightsquigarrow$ **relevance**.

Ex:

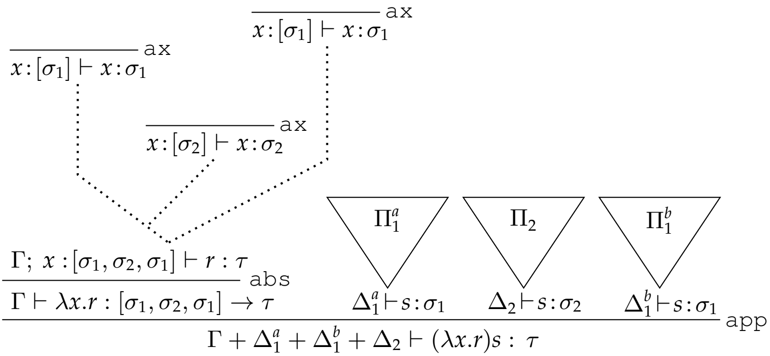
$$\frac{\dfrac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax}}{\vdash \lambda x.x : [\tau] \to \tau}\ \texttt{abs} \qquad \frac{\dfrac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax}}{x : [\tau] \vdash \lambda y.x : [\,] \to \tau}\ \texttt{abs}$$

FEATURES

- System $\mathscr{R}_0$ enjoys both Subject Reduction and Expansion.
  (SR and SE easy to prove thanks to strictness and relevance)
- If t is $\mathscr{R}_0$-typable, then the head reduction starting at t terminates.
  (Very easy thanks to the non-idempotency of intersection)
- Head Normal Forms are $\mathscr{R}_0$-typable.
- **Theorem[de Carvalho, 07]:** a term is typable in $\mathscr{R}_0$ iff it is head normalizing.

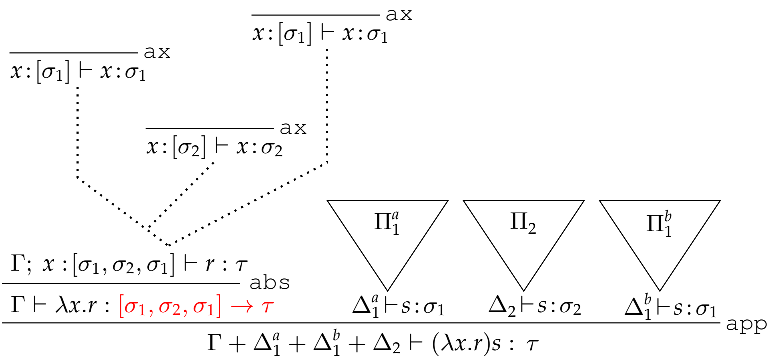
WEIGHTED SUBJECT REDUCTION (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



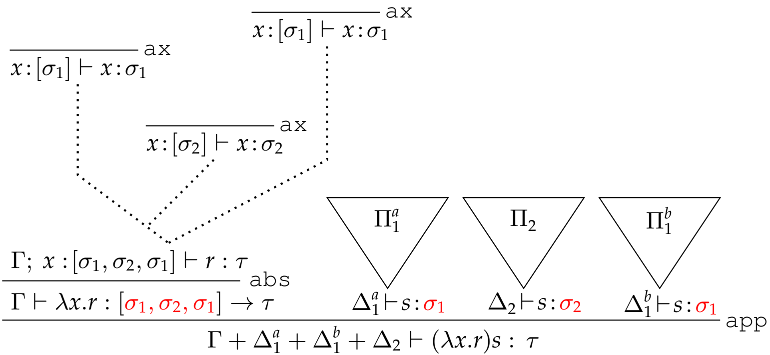
Weighted Subject Reduction (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



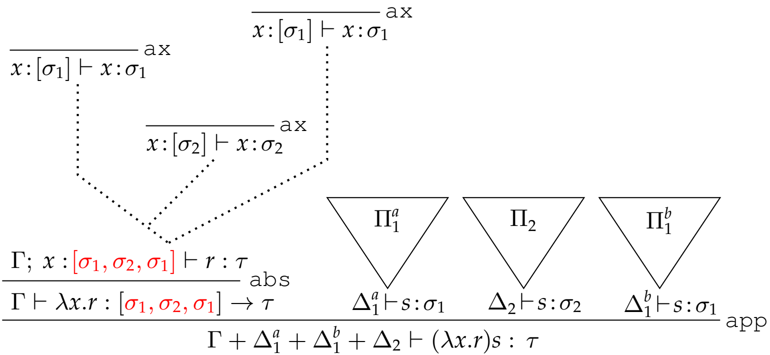
WEIGHTED SUBJECT REDUCTION (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



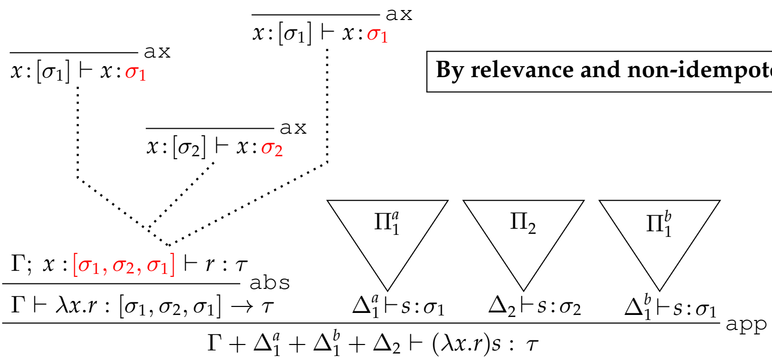
WEIGHTED SUBJECT REDUCTION (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



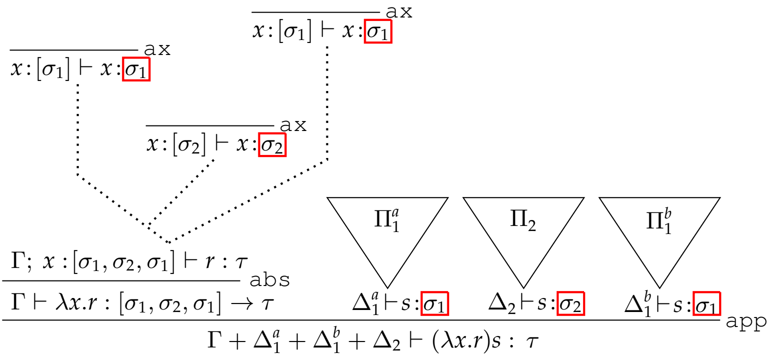
WEIGHTED SUBJECT REDUCTION (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



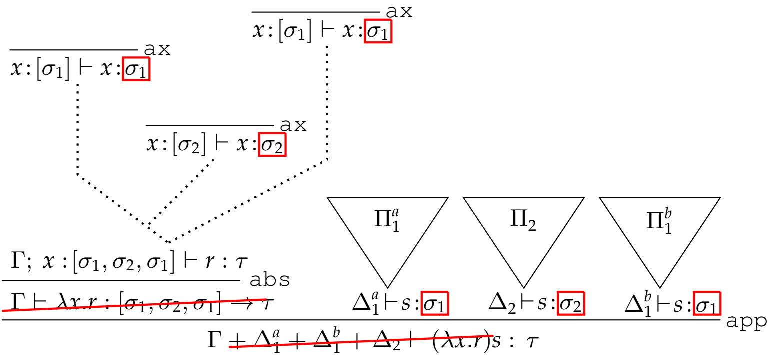
Weighted Subject Reduction (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



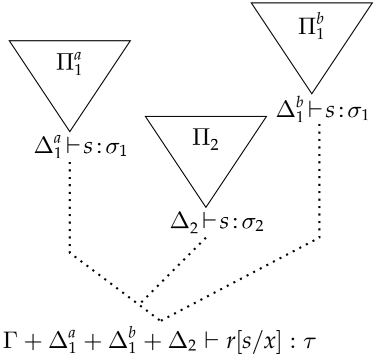
WEIGHTED SUBJECT REDUCTION (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



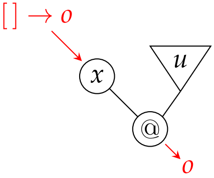
WEIGHTED SUBJECT REDUCTION (HN λ)

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



UNTYPED SUBTERMS

- The argument u of an `app`-rule may be left **untyped**:



i.e.

$$\dfrac{\dfrac{}{x : [[\,] \to o] \vdash x : [\,] \to o}\ \texttt{ax}}{x : [[\,] \to o] \vdash x\, u : o}\ \texttt{app}$$

$t : [\,] \to \tau$ *means "t is a function to* τ *that does not look at its argument".*

- u is not erasable in $x\,u$ (whether u norm. or not).
  $\rightsquigarrow$ a $\mathscr{R}_0$-derivation is *not* a certificate of Weak Normalization.

- To characterize WN, we need to consider **unforgetful** derivations/judgments
  Unforgetfulness of $\Gamma \vdash t : \tau$*: conditions on the parity of the nesting levels of the occ. of* $[\,]$ *in* Γ *and* τ*.*

- **Theorem[folklore]:** t is WN if t is unforgetfully typable in $\mathscr{R}_0$.

Infinitary Normalization

Finite Intersection Type Systems

Back to Klop's Problem

TOWARDS INFINITARY TYPING

- *Idea:* to characterize ∞-WN, let us unforgetfully type infinite normal forms $\rightsquigarrow$ no part of an ∞-NF must be left untyped…
- Need to consider infinite derivations with a coinductive type grammar ($\mathscr{R}_0 \rightsquigarrow \mathscr{R}$).

TOWARDS INFINITARY TYPING

- *Idea:* to characterize ∞-WN, let us unforgetfully type infinite normal forms $\rightsquigarrow$ no part of an ∞-NF must be left untyped...
- Need to consider infinite derivations with a coinductive type grammar ($\mathscr{R}_0 \rightsquigarrow \mathscr{R}$).
- **Problem 1:** how do we perform infinite subject reduction/expansion ?
  Actually, this is difficult only for SE (extra-slide available)
- **Problem 2:** the coinductive type grammar allows to define for instance ρ by $\rho = [\rho]_\omega \rightarrow o$.
  Using ρ, we may type $\Omega = \Delta\,\Delta$ with o (*unsound* derivations).

TOWARDS INFINITARY TYPING

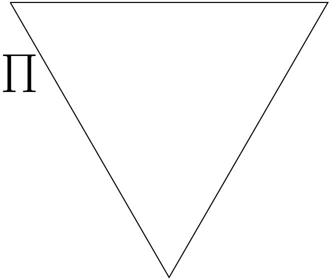
- *Idea:* to characterize ∞-WN, let us unforgetfully type infinite normal forms $\rightsquigarrow$ no part of an ∞-NF must be left untyped...
- Need to consider infinite derivations with a coinductive type grammar ($\mathscr{R}_0 \rightsquigarrow \mathscr{R}$).
- **Problem 1:** how do we perform infinite subject reduction/expansion ?
  Actually, this is difficult only for SE (extra-slide available)
- **Problem 2:** the coinductive type grammar allows to define for instance ρ by $\rho = [\rho]_\omega \rightarrow o$.
  Using ρ, we may type $\Omega = \Delta\,\Delta$ with o (*unsound* derivations).
- **Solution (for both problems):** resort to a *validity criterion* called **approximability**.

APPROXIMABILITY (INTUITIONS)

- A derivation is a set of symbols, that satisfies some grammar.
- Some derivations are included in others

$$\dfrac{\dfrac{}{x : [[\ ] \to o] \vdash x : [\ ] \to o}}{x : [[o] \to o] \qquad \vdash x\,y : o}$$

- **Informal Definition [V,LICS17]:** a derivation Π is approximable if, for all *finite* selection of symbols B_0, there is a *finite* derivation Π_f included in Π and containing B_0.

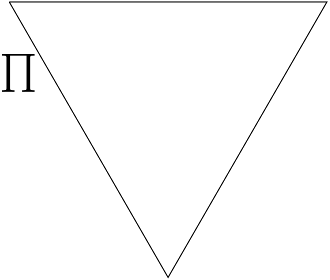


APPROXIMABILITY (INTUITIONS)

- A derivation is a set of symbols, that satisfies some grammar.
- Some derivations are included in others

$$\dfrac{\dfrac{}{x : [[o] \to o] \vdash x : [o] \to o} \qquad \dfrac{}{y : [o] \vdash y : o}}{x : [[o] \to o]; y : [o] \vdash x\, y : o}$$

- **Informal Definition [V,LICS17]:** a derivation Π is approximable if, for all *finite* selection of symbols B_0, there is a *finite* derivation Π_f included in Π and containing B_0.

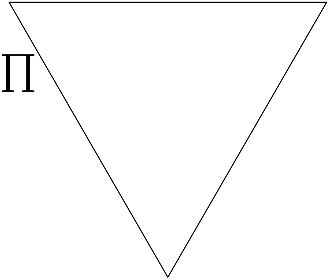


Approximability (intuitions)

- A derivation is a set of symbols, that satisfies some grammar.
- Some derivations are included in others (black $\subseteq$ black+red)

$$\dfrac{\dfrac{}{x : [[o] \to o] \vdash x : [o] \to o} \qquad \dfrac{}{y : [o] \vdash y : o}}{x : [[o] \to o]; y : [o] \vdash x\, y : o}$$

- **Informal Definition [V,LICS17]:** a derivation Π is approximable if, for all *finite* selection of symbols B_0, there is a *finite* derivation Π_f included in Π and containing B_0.

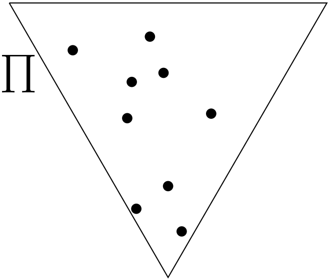


APPROXIMABILITY (INTUITIONS)

- A derivation is a set of symbols, that satisfies some grammar.
- Some derivations are included in others (black $\subseteq$ black+red)

$$\dfrac{\overline{x : [[o] \to o] \vdash x : [o] \to o} \qquad \overline{y : [o] \vdash y : o}}{x : [[o] \to o]; y : [o] \vdash x\,y : o}$$

- **Informal Definition [V,LICS17]:** a derivation Π is approximable if, for all *finite* selection of symbols B_0, there is a *finite* derivation Π_f included in Π and containing B_0.

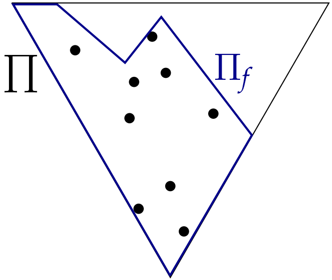


APPROXIMABILITY (INTUITIONS)

- A derivation is a set of symbols, that satisfies some grammar.
- Some derivations are included in others (black ⊆ black+red)

$$\dfrac{\overline{x : [[o] \to o] \vdash x : [o] \to o} \qquad \overline{y : [o] \vdash y : o}}{x : [[o] \to o]; y : [o] \vdash x\, y : o}$$

- **Informal Definition [V,LICS17]:** a derivation Π is approximable if, for all *finite* selection of symbols B_0, there is a *finite* derivation Π_f included in Π and containing B_0.

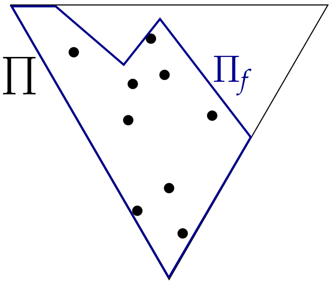


Approximability (intuitions)

- A derivation is a set of symbols, that satisfies some grammar.
- Some derivations are included in others (black $\subseteq$ black+red)

$$\frac{x : [[o] \to o] \vdash x : [o] \to o \qquad y : [o] \vdash y : o}{x : [[o] \to o]; y : [o] \vdash x\,y : o}$$

- **Informal Definition [V,LICS17]:** a derivation Π is approximable if, for all *finite* selection of symbols B_0, there is a *finite* derivation Π_f included in Π and containing B_0.



Problem 3: Approximability cannot be expressed with multisets.

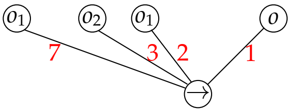
SEQUENTIAL INTERSECTION

- **Strict Types:**

$$S_k, T \ ::= \ o \in \mathscr{O} \mid (S_k)_{k\in K} \to T$$

- **Sequence Types** $(S_k)_{k\in K}$:
  - Intersection type replacing multiset types.
  - We also write $(S_k)_{k\in K} = (k \cdot S_k)_{k\in K}$.
  - The indexes k are called **tracks** (=labels of *edges*).

- *Example:* $(7 \cdot o_1, 3 \cdot o_2, 2 \cdot o_1) \to o$



- *Tracking:* $(3 \cdot S, 5 \cdot T, 9 \cdot S) = (3 \cdot S, 5 \cdot T) \uplus (9 \cdot S)$

  vs. $[\sigma, \tau, \sigma] = [\sigma, \tau] + [\sigma]$

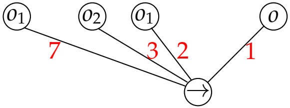
SEQUENTIAL INTERSECTION

- **Strict Types:**

$$S_k, T \ ::= \ o \in \mathcal{O} \mid (S_k)_{k\in K} \to T$$

- **Sequence Types** $(S_k)_{k\in K}$:
  - Intersection type replacing multiset types.
  - We also write $(S_k)_{k\in K} = (k \cdot S_k)_{k\in K}$.
  - The indexes k are called **tracks** (=labels of *edges*).

- *Example:* $(7 \cdot o_1, 3 \cdot o_2, 2 \cdot o_1) \to o$



- *Tracking:* $(3 \cdot S, 5 \cdot T, 9 \cdot S) = (3 \cdot S, 5 \cdot T) \uplus (9 \cdot S)$

  vs. $[\sigma, \tau, \sigma] = [\sigma, \tau] + [\sigma]$

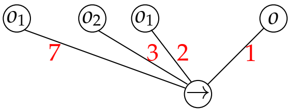
SEQUENTIAL INTERSECTION

- **Strict Types:**

$$S_k, T \ ::= \ o \in \mathscr{O} \mid (S_k)_{k \in K} \to T$$

- **Sequence Types** $(S_k)_{k \in K}$:
  - Intersection type replacing multiset types.
  - We also write $(S_k)_{k \in K} = (k \cdot S_k)_{k \in K}$.
  - The indexes k are called **tracks** (=labels of *edges*).

- *Example:* $(7 \cdot o_1, 3 \cdot o_2, 2 \cdot o_1) \to o$



- *Tracking:* $(3 \cdot S, 5 \cdot T, 9 \cdot S) = (3 \cdot S, 5 \cdot T) \uplus (9 \cdot S)$
  vs. $[\sigma, \tau, \sigma] = [\sigma, \tau] + [\sigma]$

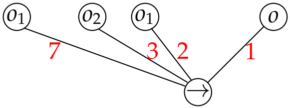
SEQUENTIAL INTERSECTION

- **Strict Types:**

$$S_k, T \ ::= \ o \in \mathscr{O} \mid (S_k)_{k \in K} \to T$$

- **Sequence Types** $(S_k)_{k \in K}$:
  - Intersection type replacing multiset types.
  - We also write $(S_k)_{k \in K} = (k \cdot S_k)_{k \in K}$.
  - The indexes k are called **tracks** (=labels of *edges*).

- *Example:* $(7 \cdot o_1, 3 \cdot o_2, 2 \cdot o_1) \to o$



- *Tracking:* $(3 \cdot S, 5 \cdot T, 9 \cdot S) = (3 \cdot S, 5 \cdot T) \uplus (9 \cdot S)$

  vs. $[\sigma, \tau, \sigma] = [\sigma , \tau] + [\sigma]$

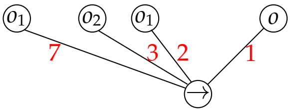
SEQUENTIAL INTERSECTION

- **Strict Types:**

$$S_k, T \ ::= \ o \in \mathscr{O} \mid (S_k)_{k \in K} \rightarrow T$$

- **Sequence Types** $(S_k)_{k \in K}$:
  - Intersection type replacing multiset types.
  - We also write $(S_k)_{k \in K} = (k \cdot S_k)_{k \in K}$.
  - The indexes k are called **tracks** (=labels of *edges*).

- *Example:* $(7 \cdot o_1, 3 \cdot o_2, 2 \cdot o_1) \rightarrow o$



- *Tracking:* $(3 \cdot S, 5 \cdot T, 9 \cdot S) = (3 \cdot S, 5 \cdot T) \uplus (9 \cdot S)$

  vs. $[\sigma, \tau, \sigma] = [\underset{?}{\sigma}, \tau] + [\underset{?}{\sigma}]$

Derivations of S

$$\dfrac{}{x:\ (k\cdot T)\vdash x:\ T}\ \texttt{ax} \qquad \dfrac{C;\ x:(S_k)_{k\in K}\vdash t:T}{C\vdash \lambda x.t:(S_k)_{k\in K}\to T}\ \texttt{abs}$$

$$\dfrac{C\vdash t:\ (S_k)_{k\in K}\to T \qquad (D_k\vdash u:\ S_k)_{k\in K}}{C\uplus(\uplus_{k\in K}D_k)\vdash t\,u:\ T}\ \texttt{app}$$

- Beware that $C\uplus(\uplus_{k\in K}D_k)$ may be not defined

  e.g. $x:(3\cdot o,4\cdot o')\uplus x:(3\cdot o,8\cdot o')$ undefined.

- System S features **pointers** (called **bipositions**)
  $\rightsquigarrow$ Approximability is definable in S (problem 3 solved).

- Every S-derivation collapses on a $\mathscr{R}$-derivation.

Proposition[V]: Given t, the set of the S-derivations typing t is a complete partial order (c.p.o.).

Characterization of infinitary WN

Proposition[V,LICS17]: In System $\mathtt{S}$:

- *SR:* typing is stable by productive ∞-reduction.
- *SE: approximable* typing stable by productive ∞-expansion.

Theorem[V,LICS17] A ∞-term t is ∞-WN iff t is unforgetfully typable by means of an approximable derivation.

Proof scheme.

- Using SR, we prove that if t is suitably typable, then the *Böhm stategy* starting at t produces an ∞-NF.
  $\rightsquigarrow$ if t is suitably typable, then t is ∞-WN.
- We suitably type the ∞-NF and proceed by infinitary SE.
  $\rightsquigarrow$ If t is ∞-WN, then t is suitably typable

SUMMARY

- HN: standard ITS (allowing untyped arguments).
- WN (finite case): unforgetfulness.
- ∞**-WN**: demands infinite types and proofs.
  - Unsoundness: need for a validity criterion (**approximability**).
  - Use **sequence** types (pointing, tracking, etc).

Thus, Klop's Problem is solved (main theorem):

∞-WN characterized by an i.t.s.

RELATED AND FUTURE WORK

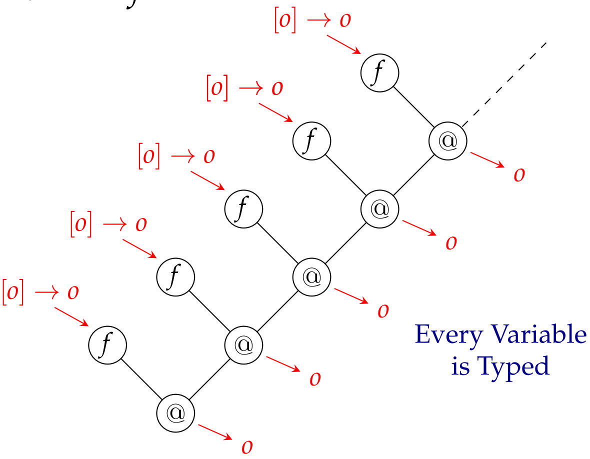
- Bonus: positive answer to TLCA Problem # 20 (characterizing hereditary permutations).
- The collapse of Type System $\mathtt{S}$ on type System $\mathscr{R}$ (coinductive multiset inter.) is surjective [V., 2015].
- Every term is $\mathtt{S}$ or $\mathscr{R}$-typable (complete unsoundness) [V., 2016] $\rightsquigarrow$ semantical info. on λ-terms
- Categorical generalization of intersection type systems *á la* Melliès-Zeilberger [Pellissier,Mazza,V,POPL18].
- Caracterization of 001-SN ? Of WN/SN in Λ^{101} ?
- Relations with Grellois-Melliès' infinitary model of LL?

QUESTIONS

Thank you for your attention !

TRUNCATION (FIGURES)

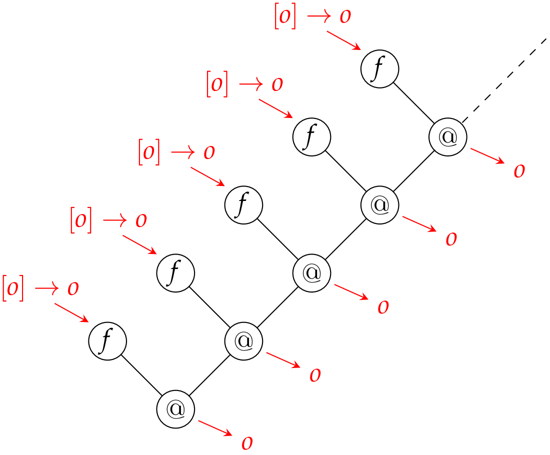
$$\Pi' \triangleright \Gamma \vdash f^\omega : o$$



$$\Gamma = f : [[o] \to o]_\omega \text{ (infinite multiplicity)}$$

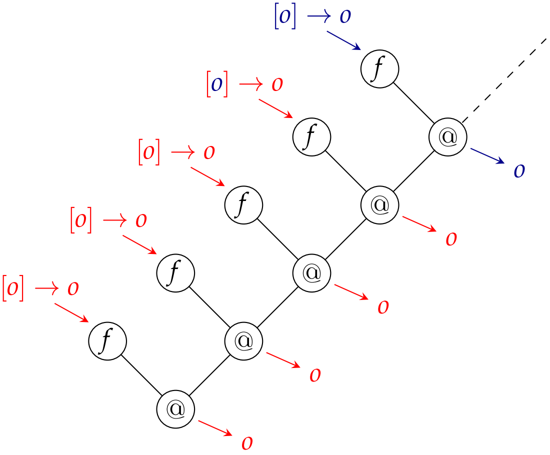
TRUNCATION (FIGURES)

$\Pi' \rhd f : [[o] \to o]_\omega \vdash f^\omega : o$ can be **truncated** into Π'_4



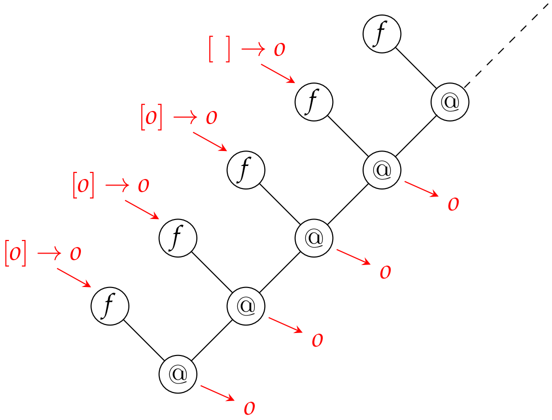
TRUNCATION (FIGURES)

$\Pi' \rhd f : [[o] \to o]_\omega \vdash f^\omega : o$ can be **truncated** into Π'_4



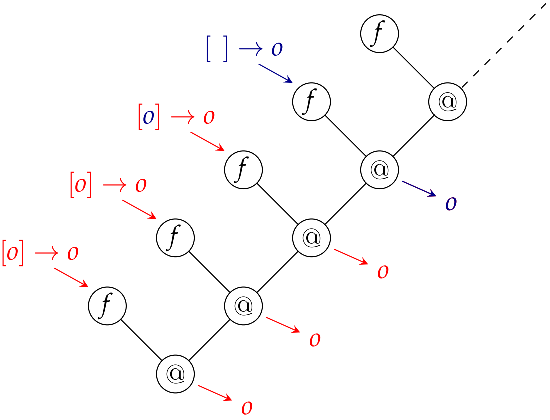
TRUNCATION (FIGURES)

$\Pi' \rhd f : [[o] \to o]_\omega \vdash f^\omega : o$ can be **truncated** into Π'_4



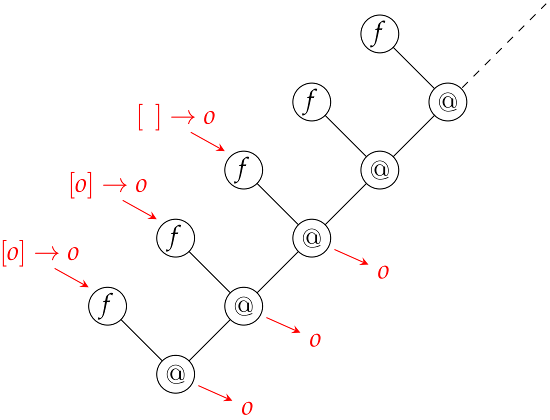
Truncation (Figures)

$\Pi' \rhd f : [[o] \to o]_\omega \vdash f^\omega : o$ can be **truncated** into Π'_3



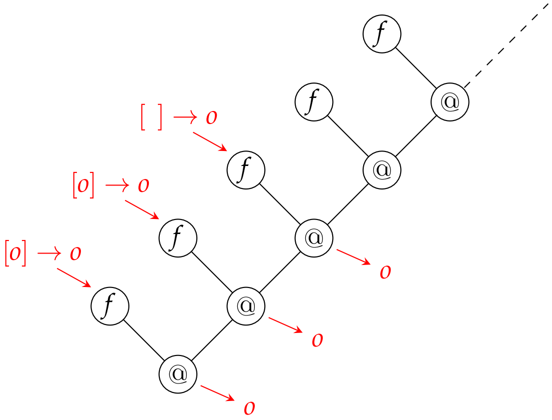
TRUNCATION (FIGURES)

$\Pi' \rhd f : [[o] \to o]_\omega \vdash f^\omega : o$ can be **truncated** into Π'_3



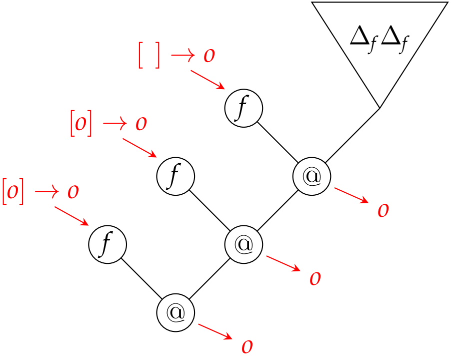
TRUNCATION (FIGURES)

f^ω may be replaced by $f^3(\Delta_f \Delta_f)$ in Π'_3,
yielding Π^3_3 :



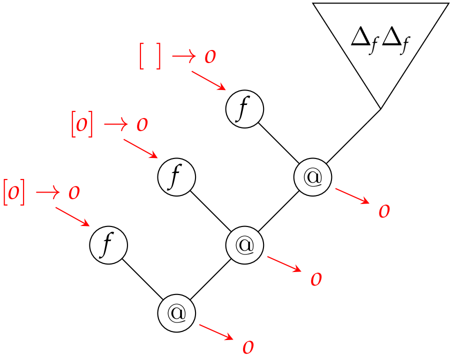
Truncation (Figures)

f^ω may be replaced by $f^3(\Delta_f\Delta_f)$ in Π_3',
yielding Π_3^3 :



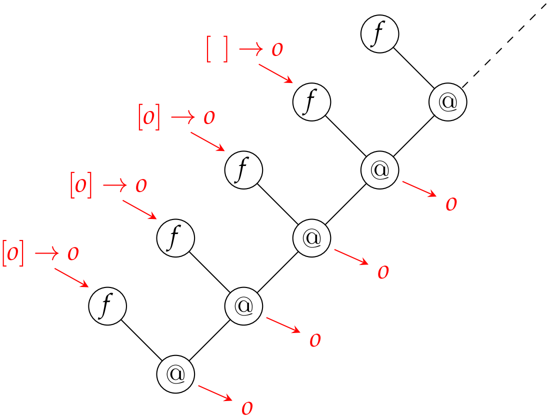
TRUNCATION (FIGURES)

Π_3^3 may be expanded 3 times,
yielding $\Pi_3 \triangleright \Delta_f \Delta_f$:



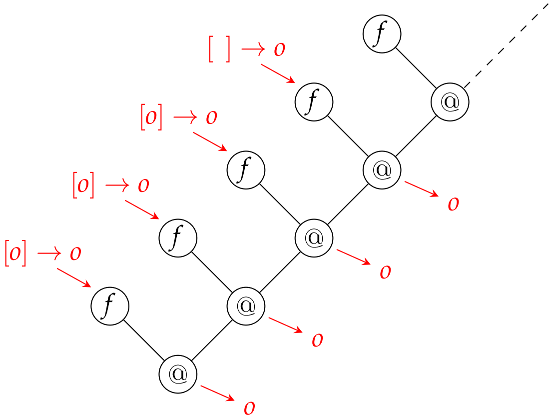
TRUNCATION (FIGURES)

Back to Π'_4, level 4 truncation of Π' :



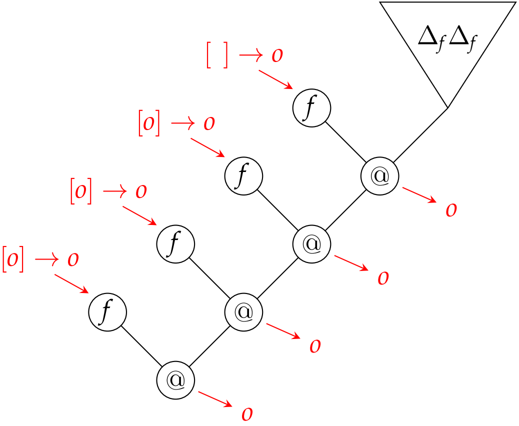
Truncation (Figures)

f^ω may be replaced by $f^4(\Delta_f \Delta_f)$ in Π'_3, yielding Π_4^4 :



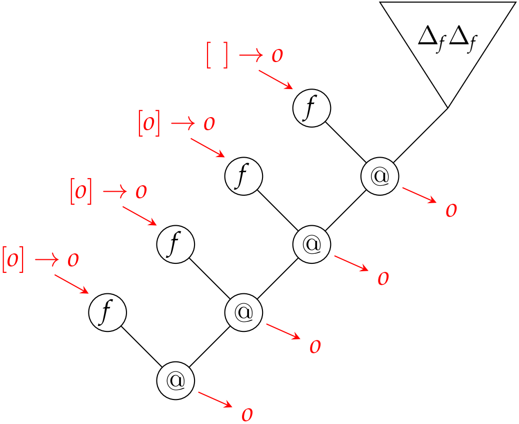
TRUNCATION (FIGURES)

f^ω may be replaced by $f^4(\Delta_f\Delta_f)$ in Π_3', yielding Π_4^4 :

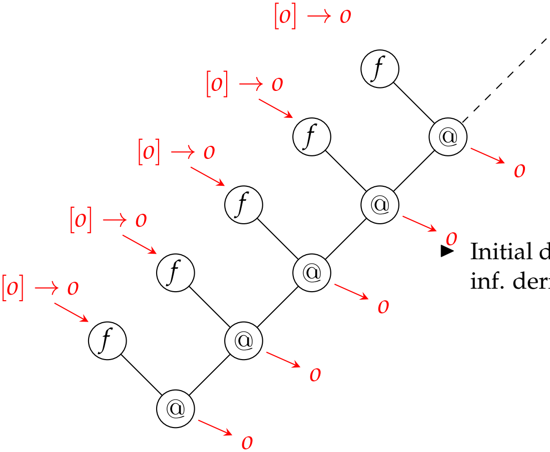


TRUNCATION (FIGURES)

Π_4^4 may be expanded 4 times,
yielding $\Pi_4 \rhd \Delta_f \Delta_f$:

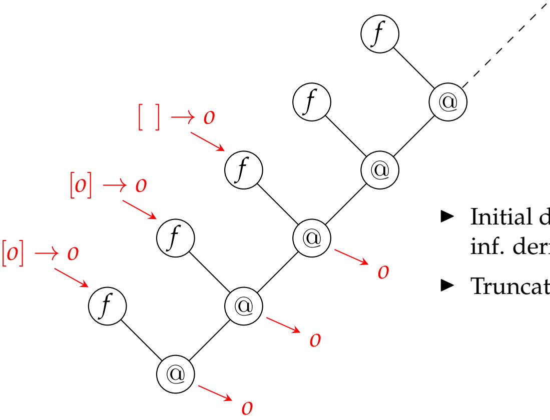


Truncation (Figures)



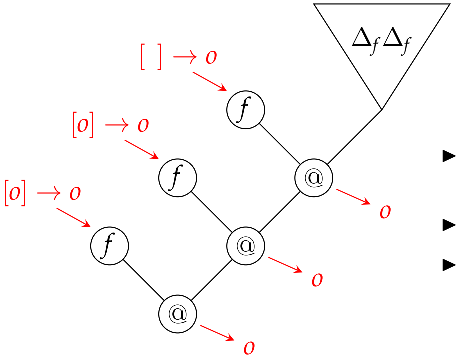
- Initial derivation (inf. term, inf. deriv.)

TRUNCATION (FIGURES)



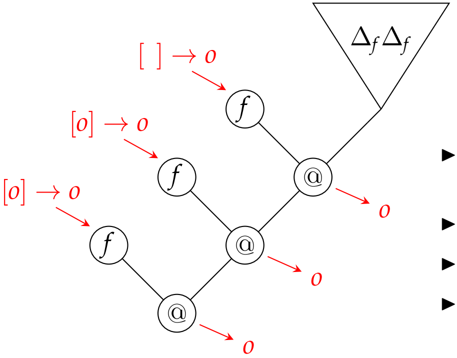
- Initial derivation (inf. term, inf. deriv.)
- Truncation (inf t., f. deriv.)

TRUNCATION (FIGURES)



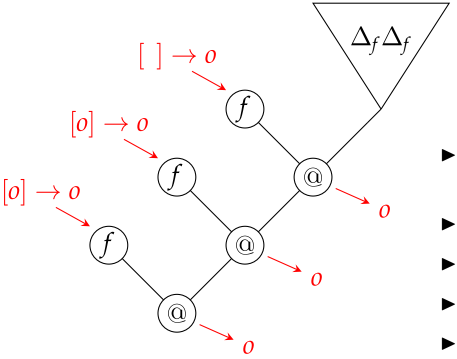
- Initial derivation (inf. term, inf. deriv.)
- Truncation (inf t., f. deriv.)
- Subject subst. (fin. t., fin. d.)

TRUNCATION (FIGURES)



- Initial derivation (inf. term, inf. deriv.)
- Truncation (inf t., f. deriv.)
- Subject subst. (fin. t., fin. d.)
- Expansion

TRUNCATION (FIGURES)



- Initial derivation (inf. term, inf. deriv.)
- Truncation (inf t., f. deriv.)
- Subject subst. (fin. t., fin. d.)
- Expansion
- Take the join