New decreasing measures for typed λ-calculi

Cristian Sottile

Universidad Nacional de Quilmes
&
Universidad de Buenos Aires
Argentina

SCALP, Loria, Nancy
October 20, 2026



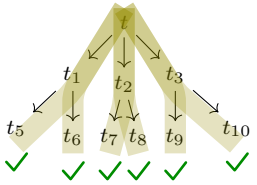






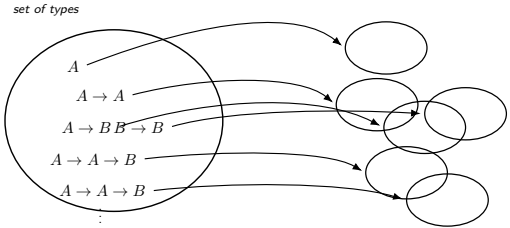
Termination in the STLC

Strong normalization (SN)



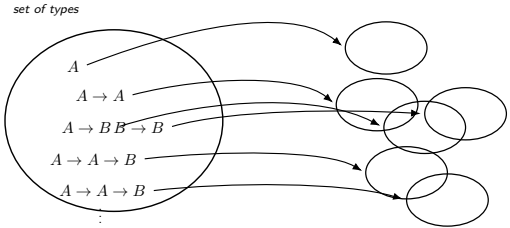
Reducibility (candidates) [Tait'67, Girard'71/72]

- Interpreting types as sets of terms
- Capturing "well-behaviour" w.r.t. termination



Reducibility (candidates) [Tait'67, Girard'71/72]

- Interpreting types as sets of terms
- Capturing "well-behaviour" w.r.t. termination



abstraction

concise — **adaptable**

lacking intuition — **why it works?**

The complexity of reducibility

Abstraction

de Vrijer'87

The *loose valuation* only gives an upper bound for the height. But the proof of strong normalization that corresponds to it—here called the "quick proof"—is simple and transparent. Presumably more transparent than the ones involving a computability predicate. As a matter of fact the quick proof can be viewed as providing some extra insight into the nature of the computability proofs. The computability predicate should then be analyzed as abstracting from all the information—still explicit in the labeled functionals—that is not strictly needed in an induction on terms which successfully establishes strong normalization.

Insight on *why*

Gallier'94

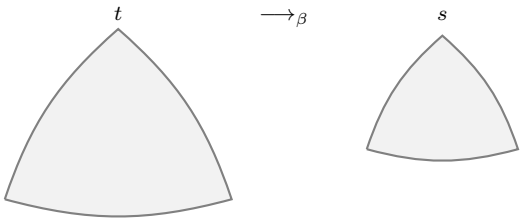
This paper provides some answers to the above questions. But before explaining our results, we would like to explain our motivations and our point of view a little more. Reducibility proofs are seductive and thrilling, but also elusive. Following these proofs step-by-step, we see that they "work" (when they are not wrong!), but I claim that most of us would still admit that they are not sure *why* these proofs work! The situation is somewhat comparable to driving a Ferrari (I suppose): the feeling of power is tremendous, but what exactly is under the hood? What kind of carburator, what kind of valve mechanism, gives such power and flexibility?

Decreasing measures

Decreasing measure

$\#(\cdot) : \lambda\text{-terms} \to \text{well-founded order}$ $\qquad t \to_\beta s \implies \#(t) > \#(s)$

Intuitively

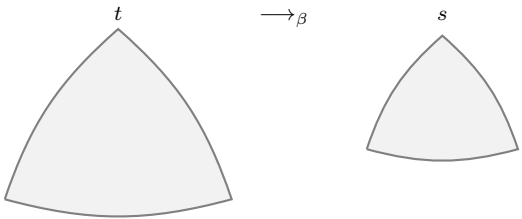


Decreasing measures

Decreasing measure

$\#(\cdot) : \lambda\text{-terms} \to \text{well-founded order}$ $\qquad t \to_\beta s \implies \#(t) > \#(s)$

Intuitively



Why?

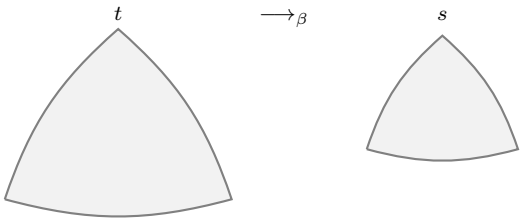
- Information
- Insight

Decreasing measures

Decreasing measure

$\#(\cdot) : \lambda\text{-terms} \to \text{well-founded order} \qquad t \to_\beta s \implies \#(t) > \#(s)$

Intuitively



Why?

- Information
- Insight

Existing measures

- Gandy'80
- de Vrijer'87

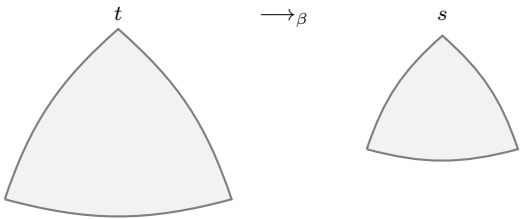
translation to λ-I
+
interpretation on HSMF
(hereditarily strictly monotonic functions)

Decreasing measures

Decreasing measure

$\#(\cdot) : \lambda\text{-terms} \to \text{well-founded order}$ $\qquad t \to_\beta s \implies \#(t) > \#(s)$

Intuitively



Why?

- Information
- Insight

Existing measures

- Gandy'80
- de Vrijer'87

translation to λ-I
+
interpretation on HSMF
(hereditarily strictly monotonic functions)

More syntactic (staying within LC)?

Works

Four decreasing measures for typed λ-calculi

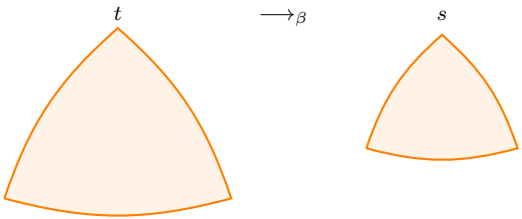
- $\mathcal{W}$: simple measure for STLC [Barenbaum, S., FSCD'23]
- $\mathcal{W}_\cap$: extension to IIT [Barenbaum, Ronchi Della Rocca, & S., MFPS'25]
- $\hat{\mathcal{W}}$: refinement to exactness STLC [To be submitted]
- $\mathcal{T}^{\mathbb{m}}$: complex measure for STLC [Barenbaum & S., FSCD'23]
- $\hat{\mathcal{W}}_\cap$: refinement to exactness for $\Lambda_\cap$ (WIP)

Gödel's koan (open question)

[TLCA #26, RTA #19]

Besides *merely* finding new ways of proving termination... we have Gödel's koan

To find an "easy" mapping from simply typed λ–calculus to ordinals



Koan?

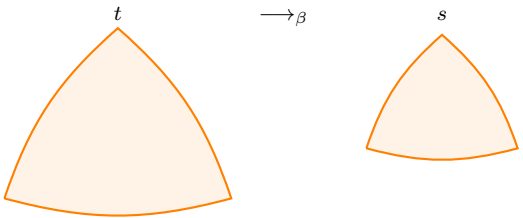
- "Easy" is not well defined
- Solutions space unclear
- A solution should be clearly recognizable

Gödel's koan (open question)

[TLCA #26, RTA #19]

Besides *merely* finding new ways of proving termination... we have Gödel's koan

To find an "easy" mapping from simply typed λ–calculus to ordinals



Koan?

- "Easy" is not well defined
- Solutions space unclear
- A solution should be clearly recognizable

Complexity dimensions

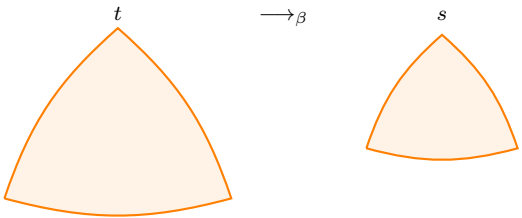
- codomain
- define
- compute
- prove
- understand

Gödel's koan (open question)

[TLCA #26, RTA #19]

Besides *merely* finding new ways of proving termination... we have Gödel's koan

To find an "easy" mapping from simply typed λ–calculus to ordinals



Koan?

- "Easy" is not well defined
- Solutions space unclear
- A solution should be clearly recognizable

Complexity dimensions

- codomain
- define
- compute
- prove
- understand

The koan in these works

- We did not solve it
- Easy at some dimensions, but not all
- We contributed new ideas
- Remains as a long-term motivation

$\mathcal{W}$: A simple decreasing measure for STLC

[Barenbaum & Sottile, FSCD'23]

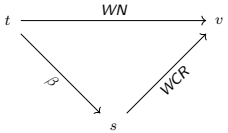
Concretized one from known ideas

DECrease from INCrease

Nederpelt-Klop's lemma $INC \wedge WCR \wedge WN \implies SN \wedge CR$

DECrease from INCrease

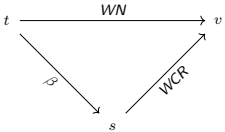
Nederpelt-Klop's lemma



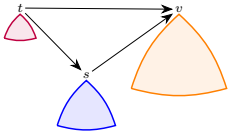
$$INC \wedge WCR \wedge WN \implies SN \wedge CR$$

DECrease from INCrease

Nederpelt-Klop's lemma

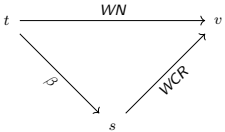


$INC \wedge WCR \wedge WN \implies SN \wedge CR$

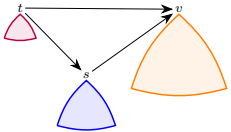


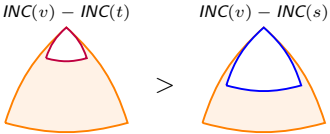
DECrease from INCrease

Nederpelt-Klop's lemma



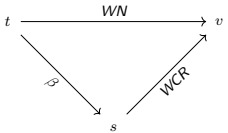
$INC \wedge WCR \wedge WN \implies SN \wedge CR$



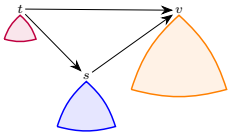


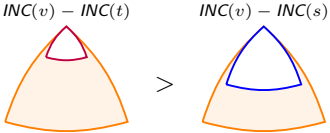
DECrease from INCrease

Nederpelt-Klop's lemma



$INC \wedge WCR \wedge WN \implies SN \wedge CR$





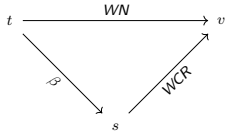
Recipe for DM!

But... not for STLC: not INC

What about a "superset" of STLC?

DECrease from INCrease

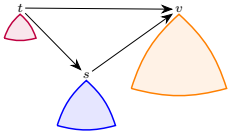
Nederpelt-Klop's lemma



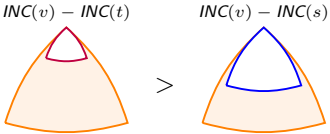
Recipe for DM!

The memory calculus $\lambda^{\mathbf{m}}$

$INC \wedge WCR \wedge WN \implies SN \wedge CR$



But... not for STLC: not INC



What about a "superset" of STLC?

Avoid erasure

$$(\lambda x.y)t \to_{\mathbf{m}} y\langle t\rangle$$

- Known idea (Church, Nederpelt, Klop)
- INC (more memories)
- WCR (by usual means)

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\, s$$

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\, s$$

Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\ s$$

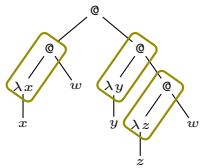
Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

A *weak* DM [Turing, Gandy, Prawitz]



Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\ s$$

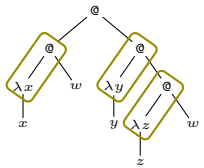
Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

A *weak* DM [Turing, Gandy, Prawitz]



Multiset of degrees $[d, d, d, ...] \to_\beta [d, d, ...]$

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\ s$$

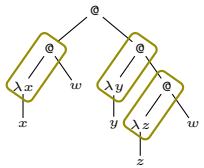
Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

A *weak* DM [Turing, Gandy, Prawitz]



Multiset of degrees $[d, d, d, ...] \to_\beta [d, d, ...]$

Simplification by degree

- Structural recursive function
- Result of bottom-up contracting all d-redexes

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\, s$$

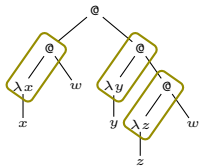
Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

A *weak* DM [Turing, Gandy, Prawitz]



Multiset of degrees $[d, d, d, ...] \to_\beta [d, d, ...]$

Simplification by degree

- Structural recursive function
- Result of bottom-up contracting all d-redexes

Properties Sound $t \to^*_{\mathbf{m}} \mathtt{S}_d(t)$ D-decreaser $\mathtt{S}_D(t) < D$

Note: partially imply Turing's obs. for $\lambda^{\mathbf{m}}$

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\ s$$

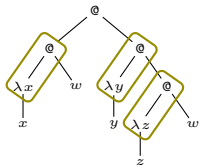
Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

A *weak* DM [Turing, Gandy, Prawitz]



Multiset of degrees $[d, d, d, ...] \to_\beta [d, d, ...]$

Simplification by degree

- Structural recursive function
- Result of bottom-up contracting all d-redexes

Properties Sound $t \to^*_{\mathbf{m}} \mathtt{S}_d(t)$ D-decreaser $\mathtt{S}_D(t) < D$

Note: partially imply Turing's obs. for $\lambda^{\mathbf{m}}$

Full simplification

d-simplify from D-max downto 1

Redex degrees and Turing's observation

Degree of a redex

Height of type of abstraction

$$\underbrace{(\lambda x.t)}_{A \to B}\ s$$

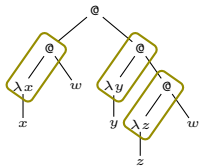
Turing's observation

Contracting redex $R := M \to_\beta N$ creates redex S

$\Longrightarrow$

degree of R is **greater** than degree of S

A *weak* DM [Turing, Gandy, Prawitz]



Multiset of degrees $[d, d, d, ...] \to_\beta [d, d, ...]$

Simplification by degree

- Structural recursive function
- Result of bottom-up contracting all d-redexes

Properties

Sound $t \to^*_{\mathbf{m}} \mathtt{S}_d(t)$

D-decreaser $\mathtt{S}_D(t) < D$

Note: partially imply Turing's obs. for $\lambda^{\mathbf{m}}$

Full simplification

d-simplify from D-max downto 1

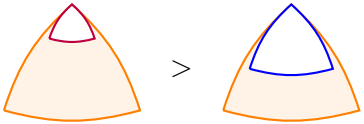
Properties

$\mathtt{S}_*$ normal form $t \to^*_{\mathbf{m}} \mathtt{S}_*(t) \not\to_{\mathbf{m}}$

Measure $\mathcal{W}$

$$t \to_{\mathbf{m}} s$$

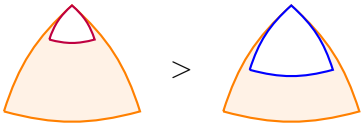
$\mathsf{w}(\mathsf{S}_*(t)) - \mathsf{w}(t)$ $\quad$ $\mathsf{w}(\mathsf{S}_*(s)) - \mathsf{w}(s)$



Measure $\mathcal{W}$

$t \to_{\mathbf{m}} s$

$\mathsf{w}(\mathsf{S}_*(t)) - \mathsf{w}(t) \qquad \mathsf{w}(\mathsf{S}_*(s)) - \mathsf{w}(s)$



$\lambda^{\mathbf{m}}$ measure, not STLC measure

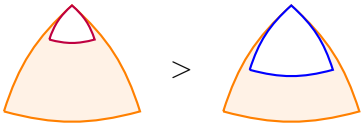
$\lambda^{\to} \subseteq \lambda^{\mathbf{m}}$ $\qquad \mathsf{w}(M) = 0$

$\mathsf{w}(\mathsf{S}_*(M)) - 0$

Measure $\mathcal{W}$

$t \to_{\mathbf{m}} s$

$\mathsf{w}(\mathtt{S}_*(t)) - \mathsf{w}(t)$ $\quad$ $\mathsf{w}(\mathtt{S}_*(s)) - \mathsf{w}(s)$



$\lambda^{\mathbf{m}}$ measure, not STLC measure

$\lambda^{\to} \subseteq \lambda^{\mathbf{m}}$ $\quad$ $\mathsf{w}(M) = 0$

$\mathsf{w}(\mathtt{S}_*(M)) - 0$

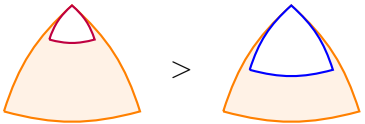
$\mathcal{W}$: a STLC measure

$$\mathcal{W}(M) = \mathsf{w}(\mathtt{S}_*(M))$$

Measure $\mathcal{W}$

$t \to_{\mathbf{m}} s$

$\mathsf{w}(\mathsf{S}_*(t)) - \mathsf{w}(t) \qquad \mathsf{w}(\mathsf{S}_*(s)) - \mathsf{w}(s)$



$\lambda^{\mathbf{m}}$ measure, not STLC measure

$\lambda^{\to} \subseteq \lambda^{\mathbf{m}}$ $\qquad \mathsf{w}(M) = 0$

$\mathsf{w}(\mathsf{S}_*(M)) - 0$

$\mathcal{W}$: a STLC measure

$$\mathcal{W}(M) = \mathsf{w}(\mathsf{S}_*(M))$$

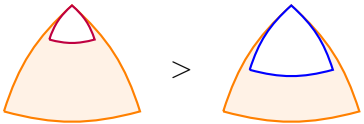
Main theorem

$$M \to_\beta N \implies \mathcal{W}(M) > \mathcal{W}(N)$$

Measure $\mathcal{W}$

$t \to_{\mathbf{m}} s$

$\mathsf{w}(\mathtt{S}_*(t)) - \mathsf{w}(t)$ $\quad$ $\mathsf{w}(\mathtt{S}_*(s)) - \mathsf{w}(s)$



$\lambda^{\mathbf{m}}$ measure, not STLC measure

$\lambda^{\to} \subseteq \lambda^{\mathbf{m}}$ $\quad$ $\mathsf{w}(M) = 0$

$\mathsf{w}(\mathtt{S}_*(M)) - 0$

$\mathcal{W}$: a STLC measure

$$\mathcal{W}(M) = \mathsf{w}(\mathtt{S}_*(M))$$

Main theorem

$$M \to_\beta N \implies \mathcal{W}(M) > \mathcal{W}(N)$$

Forgetful reduction

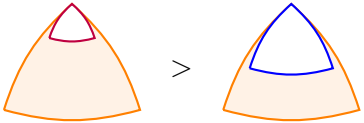
Erases a memory $\quad t\langle s\rangle \rhd t$

- Relate $\lambda^{\mathbf{m}}$ and STLC terms
- Commutes with $\to_{\mathbf{m}}$

Measure $\mathcal{W}$

$t \to_{\mathbf{m}} s$

$\mathsf{w}(\mathtt{S}_*(t)) - \mathsf{w}(t)$ $\quad$ $\mathsf{w}(\mathtt{S}_*(s)) - \mathsf{w}(s)$



$\lambda^{\mathbf{m}}$ measure, not STLC measure

$\lambda^{\to} \subseteq \lambda^{\mathbf{m}}$ $\quad$ $\mathsf{w}(M) = 0$

$\mathsf{w}(\mathtt{S}_*(M)) - 0$

$\mathcal{W}$: a STLC measure

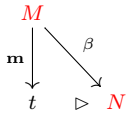
$$\mathcal{W}(M) = \mathsf{w}(\mathtt{S}_*(M))$$

Main theorem

$$M \to_\beta N \implies \mathcal{W}(M) > \mathcal{W}(N)$$

Forgetful reduction

Erases a memory $\quad t\langle s\rangle \rhd t$

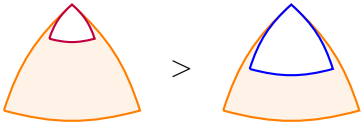


- Relate $\lambda^{\mathbf{m}}$ and STLC terms
- Commutes with $\to_{\mathbf{m}}$

Measure $\mathcal{W}$

$t \to_{\mathbf{m}} s$

$\mathtt{w}(\mathtt{S}_*(t)) - \mathtt{w}(t)$ $\quad$ $\mathtt{w}(\mathtt{S}_*(s)) - \mathtt{w}(s)$



$\lambda^{\mathbf{m}}$ measure, not STLC measure

$\lambda^{\to} \subsetneq \lambda^{\mathbf{m}}$ $\quad$ $\mathtt{w}(M) = 0$

$\mathtt{w}(\mathtt{S}_*(M)) - 0$

$\mathcal{W}$: a STLC measure

$$\mathcal{W}(M) = \mathtt{w}(\mathtt{S}_*(M))$$

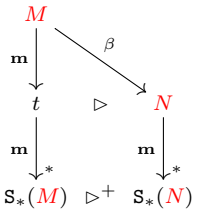
Main theorem

$$M \to_\beta N \implies \mathcal{W}(M) > \mathcal{W}(N)$$

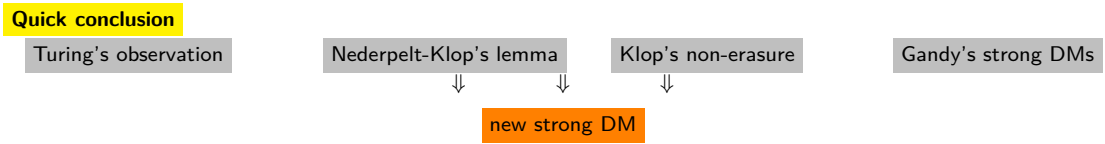
Forgetful reduction

Erases a memory $\quad t\langle s\rangle \rhd t$

- Relate $\lambda^{\mathbf{m}}$ and STLC terms
- Commutes with $\to_{\mathbf{m}}$



Quick conclusion



$\mathcal{W}_\cap$: A simple measure for $\Lambda_\cap$

[Barenbaum, Ronchi Della Rocca & Sottile, MFPS'25]

Adapted $\mathcal{W}$ to $\Lambda_\cap$

Introduction

Extra motivation

Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Introduction

Extra motivation

Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Idempotent intersection types

$$N : \{A_1, ..., A_n\}$$

Introduction

Extra motivation

Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Idempotent intersection types

$N : \{A_1, ..., A_n\}$ $\mathtt{S}_d$ relies on typing info

Introduction

Extra motivation

Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Idempotent intersection types

$N : \{A_1, ..., A_n\}$ $\mathtt{S}_d$ relies on typing info $\mathtt{S}_d$ requires Church-style

Introduction

Extra motivation

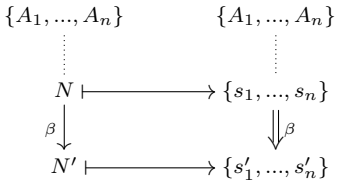
Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Idempotent intersection types

$N : \{A_1, ..., A_n\}$ $\mathtt{S}_d$ relies on typing info $\mathtt{S}_d$ requires Church-style

Existing Church version

[Liquori & Ronchi Della Rocca]



Introduction

Extra motivation

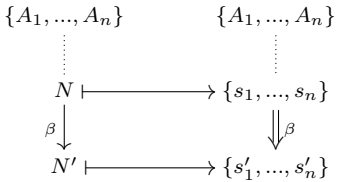
Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Idempotent intersection types

$N : \{A_1, ..., A_n\}$ S_d relies on typing info S_d requires Church-style

Existing Church version

[Liquori & Ronchi Della Rocca]



S_d: $\neq$ type, $\neq$ degree

$\{s_1, \quad ..., \quad s_n\}$

β

$\{s'_1, \quad ..., \quad s_n\}$

Introduction

Extra motivation

Existing measures weak or STLC-translation based $\Longrightarrow$ no native strong measure

Idempotent intersection types

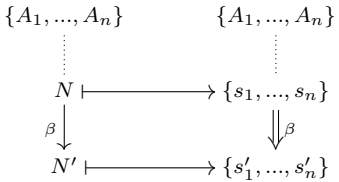
$N : \{A_1, ..., A_n\}$

$\mathtt{S}_d$ relies on typing info

$\mathtt{S}_d$ requires Church-style

Existing Church version

[Liquori & Ronchi Della Rocca]



$\mathtt{S}_d$: $\neq$ type, $\neq$ degree



$\mathtt{S}_d$ unsound

$\Lambda_\cap^{\mathsf{Ch}}$: Idempotent intersection types à la Church

$\Lambda_\cap^{\mathsf{Ch}}$: Idempotent intersection types à la Church

Linear reduction Inspired by Kfoury's

$$\{s_1, ..., t_n\} \underset{\mathsf{Ch}}{\longrightarrow} ... \longrightarrow \underset{\mathsf{Ch}}{\{s_1, ..., s_n\}}$$

$\Lambda^{\mathsf{Ch}}_\cap$: Idempotent intersection types à la Church

Linear reduction Inspired by Kfoury's

$$\{s_1, ..., t_n\} \to_{\mathsf{Ch}} ... \to_{\mathsf{Ch}} \{s_1, ..., s_n\}$$

Injectivity conditions in rules

$$\{A_1, ..., A_n\} (A_i \neq A_j \text{ if } i \neq j)$$

$$\{t_1, ..., t_n\} (t_i \neq t_j \text{ if } i \neq j)$$

$$\frac{(\Gamma \vdash_{\mathsf{Ch}} s_i : A_i)_{i \in 1..n} \qquad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{Ch}} \{s_1, \ldots, s_n\} : \{A_1, \ldots, A_n\}}$$

$\Lambda^{\mathsf{Ch}}_\cap$: Idempotent intersection types à la Church

Linear reduction Inspired by Kfoury's

$$\{s_1, ..., t_n\} \to_{\mathsf{Ch}} ... \to_{\mathsf{Ch}} \{s_1, ..., s_n\}$$

Injectivity conditions in rules

$$\{A_1, ..., A_n\} (A_i \neq A_j \text{ if } i \neq j)$$

$$\{t_1, ..., t_n\} (t_i \neq t_j \text{ if } i \neq j)$$

$$\frac{(\Gamma \vdash_{\mathsf{Ch}} s_i : A_i)_{i \in 1..n} \qquad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{Ch}} \{s_1, \ldots, s_n\} : \{A_1, \ldots, A_n\}}$$

Bijection between set-term $\vec{s}$ and set-type $\vec{A}$

$$\Gamma \Vdash \vec{s} : \vec{A}$$

$\Lambda^{\mathsf{Ch}}_\cap$: Idempotent intersection types à la Church

Linear reduction Inspired by Kfoury's

$$\{s_1, ..., t_n\} \to_{\mathsf{Ch}} ... \to_{\mathsf{Ch}} \{s_1, ..., s_n\}$$

Injectivity conditions in rules

$$\{A_1, ..., A_n\} (A_i \neq A_j \text{ if } i \neq j)$$

$$\{t_1, ..., t_n\} (t_i \neq t_j \text{ if } i \neq j)$$

$$\frac{(\Gamma \vdash_{\mathsf{Ch}} s_i : A_i)_{i \in 1..n} \qquad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{Ch}} \{s_1, \ldots, s_n\} : \{A_1, \ldots, A_n\}}$$

Bijection between set-term $\vec{s}$ and set-type $\vec{A}$

$$\Gamma \Vdash \vec{s} : \vec{A}$$

Substitution parallel

$$(\lambda x^{\{A_1,\ldots,A_n\}}.t)\{s_1^{A_1}, ..., s_n^{A_n}\}$$

$$\downarrow_{\mathsf{Ch}}$$

$$t[s_1^{A_1}/x^{A_1}, \ldots, s_n^{A_n}/x^{A_n}]$$

$\Lambda^{\mathsf{Ch}}_\cap$: Idempotent intersection types à la Church

Linear reduction Inspired by Kfoury's

$$\{s_1, ..., t_n\} \underset{\mathsf{Ch}}{\longrightarrow} ... \underset{\mathsf{Ch}}{\longrightarrow} \{s_1, ..., s_n\}$$

Injectivity conditions in rules

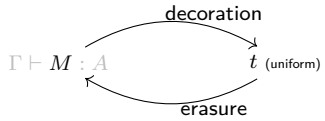
$$\{A_1, ..., A_n\}(A_i \neq A_j \text{ if } i \neq j)$$

$$\{t_1, ..., t_n\}(t_i \neq t_j \text{ if } i \neq j)$$

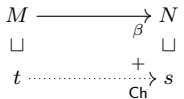
$$\frac{(\Gamma \vdash_{\mathsf{Ch}} s_i : A_i)_{i \in 1..n} \qquad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{Ch}} \{s_1, \ldots, s_n\} : \{A_1, \ldots, A_n\}}$$

Theorems relating $\Lambda^{\mathsf{Cu}}_\cap$ and $\Lambda^{\mathsf{Ch}}_\cap$

Correspondence



Simulation



Bijection between set-term $\vec{s}$ and set-type $\vec{A}$

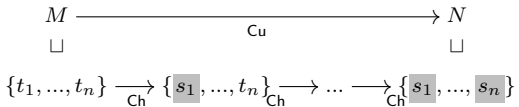
$$\Gamma \Vdash \vec{s} : \vec{A}$$

Substitution parallel

$$(\lambda x^{\{A_1,\ldots,A_n\}}.t)\{s_1^{A_1}, ..., s_n^{A_n}\}$$

$$\downarrow_{\mathsf{Ch}}$$

$$t[s_1^{A_1}/x^{A_1}, \ldots, s_n^{A_n}/x^{A_n}]$$



Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$
- One more step: lifting to $\Lambda_\cap^{\mathsf{Cu}}$

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$
- One more step: lifting to $\Lambda_\cap^{\mathsf{Cu}}$

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$
- One more step: lifting to $\Lambda_\cap^{\mathsf{Cu}}$

Curry-style measure

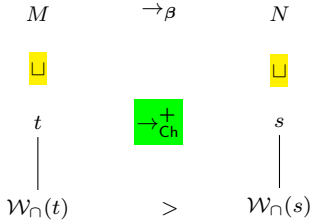
measure of the Church-correspondent

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$
- One more step: lifting to $\Lambda_\cap^{\mathsf{Cu}}$

Curry-style measure

measure of the Church-correspondent

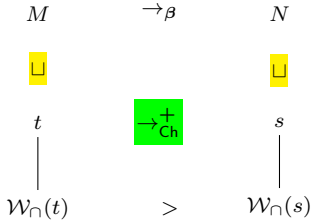


Measure

- Extend $\Lambda^{\mathsf{Ch}}_\cap$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}^{\mathsf{Ch}}_\cap$
- One more step: lifting to $\Lambda^{\mathsf{Cu}}_\cap$

Curry-style measure

measure of the Church-correspondent



Main theorem

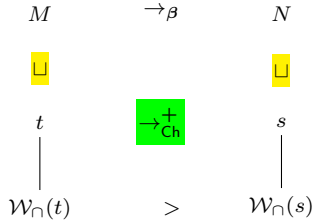
$$M \to_\beta N \implies \mathcal{W}_\cap(M) > \mathcal{W}_\cap(N)$$

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$
- One more step: lifting to $\Lambda_\cap^{\mathsf{Cu}}$

Curry-style measure

measure of the Church-correspondent



Main theorem

$$M \to_\beta N \implies \mathcal{W}_\cap(M) > \mathcal{W}_\cap(N)$$

Completing the picture: typable iff SN

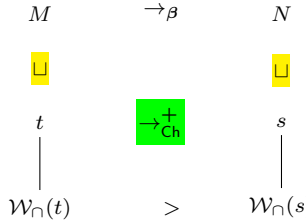
$$\Gamma \vdash M : A \iff M \in SN$$

Measure

- Extend $\Lambda_\cap^{\mathsf{Ch}}$ to memory $\Lambda_\cap$
- Replicate $\mathcal{W}$: obtain $\mathcal{W}_\cap^{\mathsf{Ch}}$
- One more step: lifting to $\Lambda_\cap^{\mathsf{Cu}}$

Curry-style measure

measure of the Church-correspondent



Main theorem

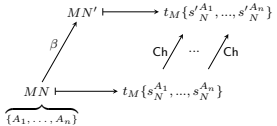
$$M \to_\beta N \implies \mathcal{W}_\cap(M) > \mathcal{W}_\cap(N)$$

Completing the picture: typable iff SN

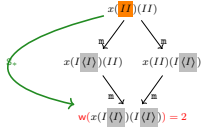
$$\Gamma \vdash M : A \iff M \in SN$$

Quick conclusion

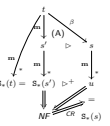
Church-style IIT



First native strong DM



Simplest codomain: $\mathbb{N}$



$\hat{\mathcal{W}}$ & $\hat{\mathcal{W}}_\cap$: Refinement to exactness of $\mathcal{W}$ & $\mathcal{W}_\cap$

[Barenbaum, Ronchi Della Rocca & Sottile, WIP]

Obtaining the tightest possible value

$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?**

$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\Longrightarrow$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

Central idea: **refine $\mathcal{W}$ to exactness**

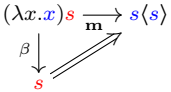
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\Longrightarrow$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\rightarrow_{\mathbf{m}}$ always remembers



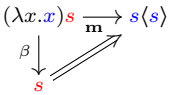
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\Longrightarrow$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

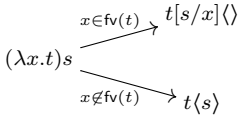
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



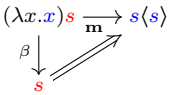
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

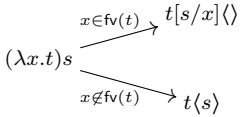
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



Definition

$$\hat{\mathcal{W}}(t) \;=\; \mathsf{w}(\mathtt{S}_*(t))$$

Decrease proof

$$M \to_\beta N \quad\Longrightarrow\quad \hat{\mathcal{W}}(N) > \hat{\mathcal{W}}(N)$$

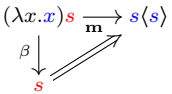
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

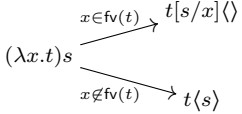
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



Definition

$$\hat{\mathcal{W}}(t) \;=\; \mathrm{w}(\mathtt{S}_*(t))$$

Decrease proof

$$M \to_\beta N \quad\Longrightarrow\quad \hat{\mathcal{W}}(N) > \hat{\mathcal{W}}(N)$$

Exactness theorem

$$\hat{\mathcal{W}}(M) = |M|$$

ISN induction + $\hat{\mathcal{W}}$ decomposition + maximal strategy

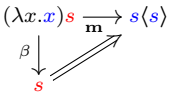
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

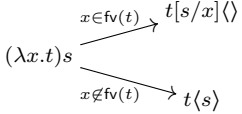
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



Definition

$$\hat{\mathcal{W}}(t) \;=\; \mathrm{w}(\mathtt{S}_*(t))$$

Decrease proof

$$M \to_\beta N \quad\Longrightarrow\quad \hat{\mathcal{W}}(N) > \hat{\mathcal{W}}(N)$$

Exactness theorem

$$\hat{\mathcal{W}}(M) = |M|$$

ISN induction + $\hat{\mathcal{W}}$ decomposition + maximal strategy

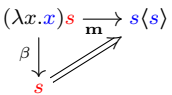
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

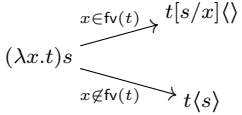
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



Definition

$$\hat{\mathcal{W}}(t) \;=\; \mathrm{w}(\mathtt{S}_*(t))$$

Decrease proof

$$M \to_\beta N \quad\implies\quad \hat{\mathcal{W}}(N) > \hat{\mathcal{W}}(N)$$

Exactness theorem

$$\hat{\mathcal{W}}(M) = |M|$$

ISN induction + $\hat{\mathcal{W}}$ decomposition + maximal strategy

WIP: exactness for $\mathcal{W}_\cap$

Extra motivation no exact DM

Extra problems Curry vs. Church gap

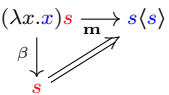
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

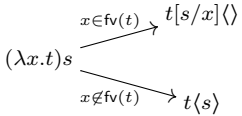
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



Definition

$$\hat{\mathcal{W}}(t) \;=\; \mathrm{w}(\mathtt{S}_*(t))$$

Decrease proof

$$M \to_\beta N \quad\Longrightarrow\quad \hat{\mathcal{W}}(N) > \hat{\mathcal{W}}(N)$$

Exactness theorem

$$\hat{\mathcal{W}}(M) = |M|$$

ISN induction + $\hat{\mathcal{W}}$ decomposition + maximal strategy

$$\{A_1, ..., A_n, A_{n+1}\} \qquad \{A_1, ..., A_n, A_{n+1}\}$$
$$\vdots \qquad\qquad\qquad \vdots$$
$$N \longmapsto \{s_1, ..., s_n, s_{n+1}\}$$

WIP: exactness for $\mathcal{W}_\cap$

Extra motivation no exact DM

Extra problems Curry vs. Church gap

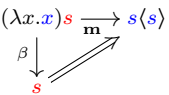
$\hat{\mathcal{W}}$: exact refinement for $\mathcal{W}$

$\mathbb{N}$ measures $\implies$ upper bound for $|M|$ **Is the upper bound exact?** Not for $\mathcal{W}$

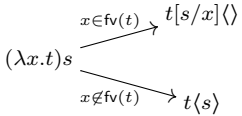
Central idea: **refine $\mathcal{W}$ to exactness**

Fixing overmeasurement

Problem $\to_{\mathbf{m}}$ always remembers



Solution refine $\to_{\mathbf{m}}$ only remember **on erasure**



Definition

$$\hat{\mathcal{W}}(t) \;=\; \mathrm{w}(\mathtt{S}_*(t))$$

Decrease proof

$$M \to_\beta N \quad\Longrightarrow\quad \hat{\mathcal{W}}(N) > \hat{\mathcal{W}}(N)$$

Exactness theorem

$$\hat{\mathcal{W}}(M) = |M|$$

ISN induction + $\hat{\mathcal{W}}$ decomposition + maximal strategy

$$\{A_1, ..., A_n, A_{n+1}\} \qquad \{A_1, ..., A_n, A_{n+1}\}$$
$$\vdots \qquad\qquad\qquad \vdots$$
$$N \longmapsto \{s_1, ..., s_n, s_{n+1}\}$$

WIP: exactness for $\mathcal{W}_\cap$

Extra motivation no exact DM

Extra problems Curry vs. Church gap

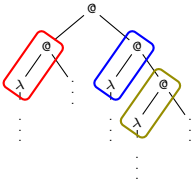
$\mathcal{T}^{\mathbb{m}}$: A complex decreasing measure for STLC

[Barenbaum & Sottile, FSCD'23]

Extending Turing's measure

Measure $\mathcal{T}^{\mathbb{m}}$: Extending Turing's result

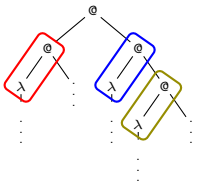
Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Measure $\mathcal{T}^{\mathbb{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

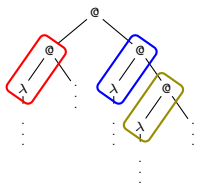
Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Measure $\mathcal{T}^{\mathfrak{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Plan

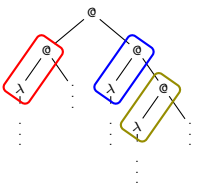
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3, ?), (2, ?), (1, ?)]$$
$$\mathcal{T}_2(t) = [(2, ?), (1, ?)]$$
$$\mathcal{T}_1(t) = [(1, ?)]$$

Lower measure: *e.g.*, $\mathcal{T}_2^{\mathfrak{m}}(t)$
when defining $\mathcal{T}_3^{\mathfrak{m}}(t)$

Measure $\mathcal{T}^{\mathfrak{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Plan

1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

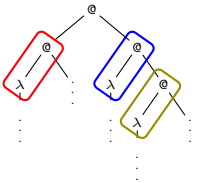
Lower measure: *e.g.*, $\mathcal{T}_2^{\mathfrak{m}}(t)$
when defining $\mathcal{T}_3^{\mathfrak{m}}(t)$

Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$

Measure $\mathcal{T}^{\mathtt{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Plan

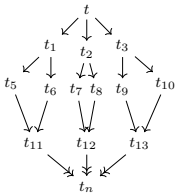
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

Lower measure: *e.g.*, $\mathcal{T}_2^{\mathtt{m}}(t)$ when defining $\mathcal{T}_3^{\mathtt{m}}(t)$

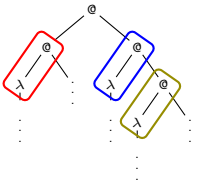
Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$



Measure $\mathcal{T}^{\mathtt{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Plan

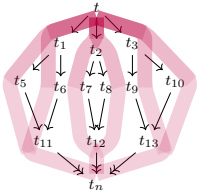
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

Lower measure: *e.g.*, $\mathcal{T}_2^{\mathtt{m}}(t)$ when defining $\mathcal{T}_3^{\mathtt{m}}(t)$

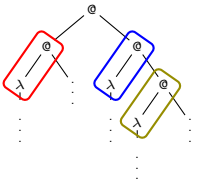
Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$



Measure $\mathcal{T}^{\mathfrak{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Plan

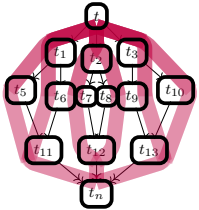
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

Lower measure: *e.g.*, $\mathcal{T}_2^{\mathfrak{m}}(t)$ when defining $\mathcal{T}_3^{\mathfrak{m}}(t)$

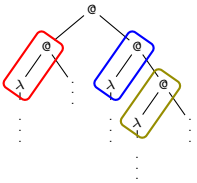
Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$



Measure $\mathcal{T}^{\mathfrak{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

Plan

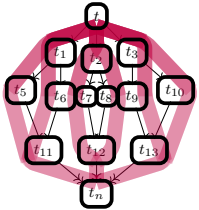
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

Lower measure: *e.g.*, $\mathcal{T}_2^{\mathfrak{m}}(t)$ when defining $\mathcal{T}_3^{\mathfrak{m}}(t)$

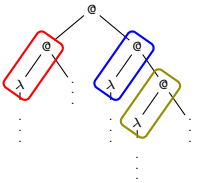
Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$



Measure $\mathcal{T}^{\mathfrak{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

$$\underbrace{[\mathcal{T}_{d-1}(t), \mathcal{T}_{d-1}(t_1), ..., \mathcal{T}_{d-1}(t_n)]}_{[\ \mathcal{T}^{\mathfrak{m}}_{d-1}(t')\ ||\ \rho{:}t \xrightarrow{d}_{\mathbf{m}}{}^{*} t'\ ]}$$

Plan

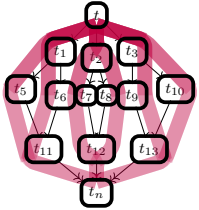
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

Lower measure: *e.g.*, $\mathcal{T}^{\mathfrak{m}}_2(t)$ when defining $\mathcal{T}^{\mathfrak{m}}_3(t)$

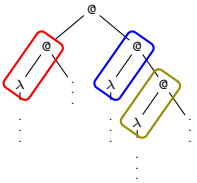
Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$



Measure $\mathcal{T}^{\mathtt{m}}$: Extending Turing's result

Turing's WN measure



$$\mathcal{T}(t) \quad = \quad [\ 3\ ,\ 2\ ,\ 1\ ]$$

Central idea

Adapt it to prove SN

Problem

redex contraction
can copy greater redexes
$\therefore$ growing MS

$$\underbrace{[\mathcal{T}_{d-1}(t), \mathcal{T}_{d-1}(t_1), ..., \mathcal{T}_{d-1}(t_n)]}_{[\ \mathcal{T}^{\mathtt{m}}_{d-1}(t')\ ||\ \rho{:}t \xrightarrow{d}_{\mathbf{m}}{}^{*} t'\ ]}$$

$$(d, [\ \mathcal{T}^{\mathtt{m}}_{d-1}(t')\ ||\ \rho : t \xrightarrow{d}_{\mathbf{m}}{}^{*} t'\ ])$$

Plan

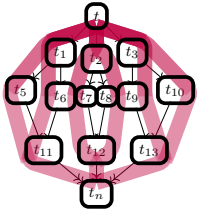
1. Indexed family of measures
2. Collect more information

$$\mathcal{T}_3(t) = [(3,?),(2,?),(1,?)]$$
$$\mathcal{T}_2(t) = [(2,?),(1,?)]$$
$$\mathcal{T}_1(t) = [(1,?)]$$

Lower measure: *e.g.*, $\mathcal{T}^{\mathtt{m}}_2(t)$ when defining $\mathcal{T}^{\mathtt{m}}_3(t)$

Consider:

1. d-red: $t \xrightarrow{d} s$ (Conf. & SN)
2. d-devs: $t \xrightarrow{d}{}^{*} s$



Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathbb{m}}(M) > \mathcal{T}^{\mathbb{m}}(N)$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathbb{m}}(M) > \mathcal{T}^{\mathbb{m}}(N)$$

What each redex contribute before and after reduction?

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathbb{m}}(M) > \mathcal{T}^{\mathbb{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathbb{m}}(M) > \mathcal{T}^{\mathbb{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathbb{m}}(t) \quad = \quad \underbrace{[..., (i, [..., \mathcal{T}^{\mathbb{m}}_{i-1}(t'), ...]), ...]}_{>} \quad + \quad \underbrace{[..., (j, [..., \mathcal{T}^{\mathbb{m}}_{j-1}(t'), ...]), ...]}_{=} \quad + \quad \underbrace{[..., (k, [..., \mathcal{T}^{\mathbb{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathbb{m}}(M) > \mathcal{T}^{\mathbb{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathbb{m}}(t) = \underbrace{[..., (i, [..., \mathcal{T}^{\mathbb{m}}_{i-1}(t'), ...]), ...]}_{>} + \underbrace{[..., (j, [..., \mathcal{T}^{\mathbb{m}}_{j-1}(t'), ...]), ...]}_{=} + \underbrace{[..., (k, [..., \mathcal{T}^{\mathbb{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathbb{m}}(M) > \mathcal{T}^{\mathbb{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathbb{m}}(t) = \underbrace{[..., (i, [..., \mathcal{T}^{\mathbb{m}}_{i-1}(t'), ...]), ...]}_{>} + \underbrace{[..., (j, [..., \mathcal{T}^{\mathbb{m}}_{j-1}(t'), ...]), ...]}_{=} + \underbrace{[..., (k, [..., \mathcal{T}^{\mathbb{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Greater redexes $>$ For each target below, greater lower measure above

$$\begin{array}{ccc} t & & \mathcal{T}^{\mathbb{m}}(t) = [..., (3, [\mathcal{T}^{\mathbb{m}}_2(t_1), ..., \mathcal{T}^{\mathbb{m}}_2(t_n)]), ...] \\ \mathbf{m} \big\downarrow j & & \\ t' & & \mathcal{T}^{\mathbb{m}}(t') = [..., (3, [\mathcal{T}^{\mathbb{m}}_2(t'_1), ..., \mathcal{T}^{\mathbb{m}}_2(t'_m)]), ...] \end{array}$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathtt{m}}(M) > \mathcal{T}^{\mathtt{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = \underbrace{[..., (i, [..., \mathcal{T}^{\mathtt{m}}_{i-1}(t'), ...]), ...]}_{>} + \underbrace{[..., (j, [..., \mathcal{T}^{\mathtt{m}}_{j-1}(t'), ...]), ...]}_{=} + \underbrace{[..., (k, [..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Similar redexes = Upwards injection + 1

$$t \quad \mathcal{T}^{\mathtt{m}}(t) = [..., (j, [\mathcal{T}^{\mathtt{m}}_{j-1}(t'), \mathcal{T}^{\mathtt{m}}_{j-1}(t_1), ..., \mathcal{T}^{\mathtt{m}}_{j-1}(t_n)]), ...]$$

$$\mathtt{m} \downarrow j$$

$$t' \quad \mathcal{T}^{\mathtt{m}}(t') = [..., (j, [\mathcal{T}^{\mathtt{m}}_{j-1}(t_1), ..., \mathcal{T}^{\mathtt{m}}_{j-1}(t_n)]), ...]$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathtt{m}}(M) > \mathcal{T}^{\mathtt{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = \underbrace{[..., (i, [..., \mathcal{T}^{\mathtt{m}}_{i-1}(t'), ...]), ...]}_{>} + \underbrace{[..., (j, [..., \mathcal{T}^{\mathtt{m}}_{j-1}(t'), ...]), ...]}_{=} + \underbrace{[..., (k, [..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Little redexes $>$ Non-erasure

$$t \;\Big|\; \mathtt{m}\; j \;\downarrow\; t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = [..., (k, [\mathcal{T}^{\mathtt{m}}_{k-1}(t_1), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t_n)]), ...]$$

$$\mathcal{T}^{\mathtt{m}}(t') = [..., (k, [\mathcal{T}^{\mathtt{m}}_{k-1}(t'_1), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'_n), \mathcal{T}^{\mathtt{m}}_{k-1}(t'_{n+1}), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'_m)]), ...]$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathtt{m}}(M) > \mathcal{T}^{\mathtt{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = \underbrace{[..., (i, [..., \mathcal{T}^{\mathtt{m}}_{i-1}(t'), ...]), ...]}_{>} + \underbrace{[..., (j, [..., \mathcal{T}^{\mathtt{m}}_{j-1}(t'), ...]), ...]}_{=} + \underbrace{[..., (k, [..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Little redexes $>$ Non-erasure

$$t \;\Big|\; \mathtt{m}\; j \;\downarrow\; t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = [..., (k, [\mathcal{T}^{\mathtt{m}}_{k-1}(t_1), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t_n)]), ...]$$

$$\mathcal{T}^{\mathtt{m}}(t') = [..., (k, [\mathcal{T}^{\mathtt{m}}_{k-1}(t'_1), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'_n), \mathcal{T}^{\mathtt{m}}_{k-1}(t'_{n+1}), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'_m)]), ...]$$

Decrease proof

Main theorem

$$M \to_\beta N \qquad \Longrightarrow \qquad \mathcal{T}^{\mathtt{m}}(M) > \mathcal{T}^{\mathtt{m}}(N)$$

What each redex contribute before and after reduction?

$$t \xrightarrow{j} t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = \underbrace{[..., (i, [..., \mathcal{T}^{\mathtt{m}}_{i-1}(t'), ...]), ...]}_{>} + \underbrace{[..., (j, [..., \mathcal{T}^{\mathtt{m}}_{j-1}(t'), ...]), ...]}_{=} + \underbrace{[..., (k, [..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'), ...]), ...]}_{<}$$

Little redexes $>$ Non-erasure

$$t \;\big|_{\mathtt{m}}\; j \;\downarrow\; t'$$

$$\mathcal{T}^{\mathtt{m}}(t) = [..., (k, [\mathcal{T}^{\mathtt{m}}_{k-1}(t_1), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t_n)]), ...]$$

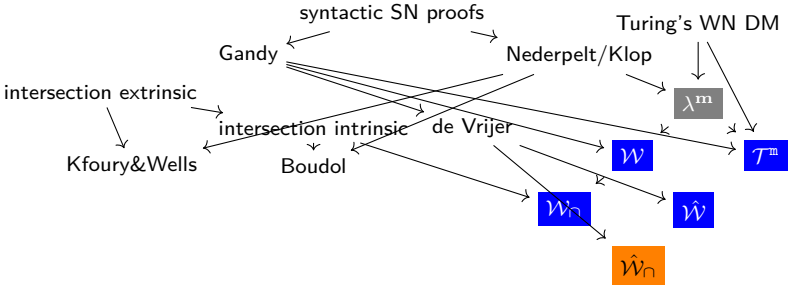
$$\mathcal{T}^{\mathtt{m}}(t') = [..., (k, [\mathcal{T}^{\mathtt{m}}_{k-1}(t'_1), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'_n), \mathcal{T}^{\mathtt{m}}_{k-1}(t'_{n+1}), ..., \mathcal{T}^{\mathtt{m}}_{k-1}(t'_m)]), ...]$$

Quick conclusions **How redex degrees could be *complemented* to make the DM strong**

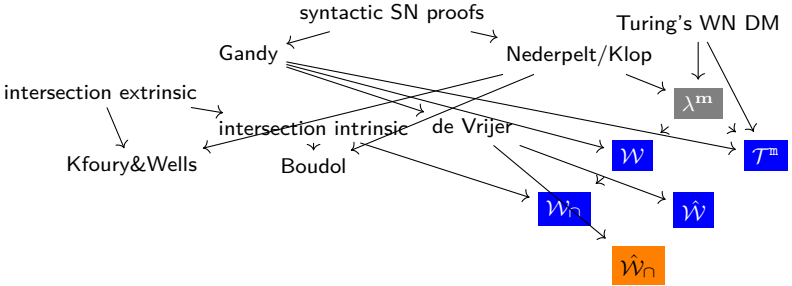
- *Easeness* dimensions
- Reducibility "opposite"
- Future distillation

Conclusions and future work

Conclusions and future work

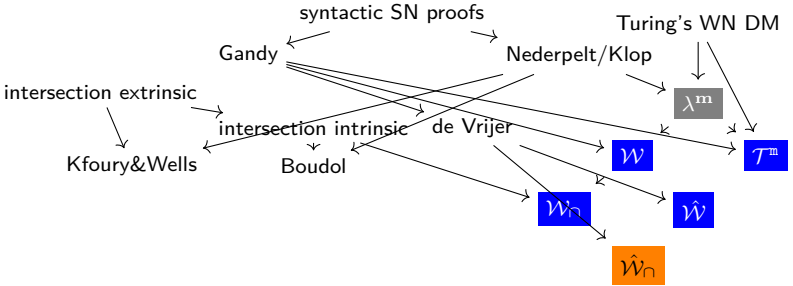


Conclusions and future work



- New elements
- New execution-decreasing info "inside" terms
- New techniques for proving SN
- Following a fully syntactic approach relying on $\lambda^{\mathbf{m}}$

Conclusions and future work



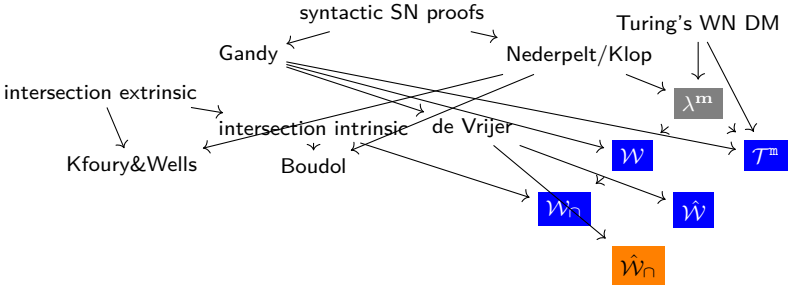
- New elements
- New execution-decreasing info "inside" terms
- New techniques for proving SN
- Following a fully syntactic approach relying on $\lambda^{\mathtt{m}}$

Depth path

more insight from $\mathcal{W}$ and $\mathcal{T}^{\mathtt{m}}$

- exact $\hat{\mathcal{W}}_\cap$
- simplified $\mathcal{T}^{\mathtt{m}}$

Conclusions and future work



- New elements
- New execution-decreasing info "inside" terms
- New techniques for proving SN
- Following a fully syntactic approach relying on $\lambda^{\mathtt{m}}$

Depth path

more insight from $\mathcal{W}$ and $\mathcal{T}^{\mathtt{m}}$

- exact $\hat{\mathcal{W}}_\cap$
- simplified $\mathcal{T}^{\mathtt{m}}$

Expansion path

adapting $\mathcal{W}$ to other systems

- HN $\Lambda_\cap$
- $\Lambda_{\equiv\setminus\mathsf{iso}}$
- System F

Thanks!