

Intersection Types for Implicit Computational Complexity

Kazushige Terui

`terui@nii.ac.jp`

National Institute of Informatics, Tokyo

Degree of intensional expressivity

- Nowadays there are plenty of Ptime systems.

Degree of intensional expressivity

- Nowadays there are **plenty** of Ptime systems.
- They are all **extensionally** complete (enough functions), but **intensionally** too poor (few algorithms) ...

Degree of intensional expressivity

- Nowadays there are **plenty** of Ptime systems.
- They are all **extensionally** complete (enough functions), but **intensionally** too poor (few algorithms) ...
- Usually poverty/richness of a system is exemplified by giving an example/counterexample program.

Degree of intensional expressivity

- Nowadays there are **plenty** of Ptime systems.
- They are all **extensionally** complete (enough functions), but **intensionally** too poor (few algorithms) ...
- Usually poverty/richness of a system is exemplified by giving an example/counterexample program.
- Is there more **principled** way to estimate the **degree of intensional expressivity**?

Degree of intensional expressivity

- Nowadays there are **plenty** of Ptime systems.
- They are all **extensionally** complete (enough functions), but **intensionally** too poor (few algorithms) ...
- Usually poverty/richness of a system is exemplified by giving an example/counterexample program.
- Is there more **principled** way to estimate the **degree of intensional expressivity**?
- **Intersection type systems** (or **implicit typing systems**): provide **static** characterizations of **dynamic** properties.

Degree of intensional expressivity

- Nowadays there are **plenty** of Ptime systems.
- They are all **extensionally** complete (enough functions), but **intensionally** too poor (few algorithms) ...
- Usually poverty/richness of a system is exemplified by giving an example/counterexample program.
- Is there more **principled** way to estimate the **degree of intensional expressivity**?
- **Intersection type systems** (or **implicit typing systems**): provide **static** characterizations of **dynamic** properties.
- Especially, **linear (non-idempotent)** systems are relevant for computational complexity.

Degree of intensional expressivity

- Nowadays there are **plenty** of Ptime systems.
- They are all **extensionally** complete (enough functions), but **intensionally** too poor (few algorithms) ...
- Usually poverty/richness of a system is exemplified by giving an example/counterexample program.
- Is there more **principled** way to estimate the **degree of intensional expressivity**?
- **Intersection type systems** (or **implicit typing systems**): provide **static** characterizations of **dynamic** properties.
- Especially, **linear (non-idempotent)** systems are relevant for computational complexity.

Outline

- Though **practically** useless (cf. Møller-Neergaard and Mairson 04), they could be **theoretically** useful for the above purpose...

Outline

- Though **practically** useless (cf. Møller-Neergaard and Mairson 04), they could be **theoretically** useful for the above purpose...
- In this talk, we only consider **pure lambda calculus**.

Outline

- Though **practically** useless (cf. Møller-Neergaard and Mairson 04), they could be **theoretically** useful for the above purpose...
- In this talk, we only consider **pure lambda calculus**.
 1. Background: how **hopeless** our situation is

Outline

- Though **practically** useless (cf. Møller-Neergaard and Mairson 04), they could be **theoretically** useful for the above purpose...
- In this talk, we only consider **pure lambda calculus**.
 1. Background: how **hopeless** our situation is
 2. Explicit vs implicit typings: some observations

Outline

- Though **practically** useless (cf. Møller-Neergaard and Mairson 04), they could be **theoretically** useful for the above purpose...
- In this talk, we only consider **pure lambda calculus**.
 1. Background: how **hopeless** our situation is
 2. Explicit vs implicit typings: some observations
 3. Polynomial time strongly normalizable terms: a static characterization

Outline

- Though **practically** useless (cf. Møller-Neergaard and Mairson 04), they could be **theoretically** useful for the above purpose...
- In this talk, we only consider **pure lambda calculus**.
 1. Background: how **hopeless** our situation is
 2. Explicit vs implicit typings: some observations
 3. Polynomial time strongly normalizable terms: a static characterization
 4. Beyond PSN: real challenge

Preliminary

- Λ : the set of untyped lambda terms.
- $W = \{0, 1\}^*$.
- Fix a Church-coding of binary words in W , eg.

$$\overline{01101} \equiv \lambda f_0 f_1 z. f_0(f_1(f_1(f_0(f_1 z))))).$$

- Any $M \in \Lambda$ **represents** a partial function $f_M : W \longrightarrow W$:

$$\begin{aligned} f_M(w) &= w' && \text{if } M\bar{w} \longrightarrow^* \bar{w}' \\ &= \uparrow && \text{otherwise.} \end{aligned}$$

- A **type system** \mathbf{L} (such as System F) determines a set $\mathcal{T} \subseteq \Lambda$ of typable terms.
- Usually, such \mathcal{T} is **r.e.**

Lambda-characterizations of FP

• $\mathcal{T} \subseteq \Lambda$ is P-sound if

$$M \in \mathcal{T} \implies f_M \in \mathbf{FP}.$$

Examples: \mathbf{MLL} , \emptyset

Lambda-characterizations of FP

- $\mathcal{T} \subseteq \Lambda$ is **P-sound** if

$$M \in \mathcal{T} \implies f_M \in \mathbf{FP}.$$

Examples: **MLL**, \emptyset

- \mathcal{T} is **extensionally P-complete** if in addition

$$f \in \mathbf{FP} \implies \exists M \in \mathcal{T} (f = f_M).$$

Examples: $\Lambda\mathbf{BC}$ (Bellantoni-Cook 92) translated into Λ ,
 $1\lambda^p(\mathbf{W})$ (Leivant-Marion 94), **LLL** (Girard 98),
SLL (Lafont 04), **DLAL** (Baillot-Terui 04)

Lambda-characterizations of FP

- $\mathcal{T} \subseteq \Lambda$ is **P-sound** if

$$M \in \mathcal{T} \implies f_M \in \mathbf{FP}.$$

Examples: **MLL**, \emptyset

- \mathcal{T} is **extensionally P-complete** if in addition

$$f \in \mathbf{FP} \implies \exists M \in \mathcal{T} (f = f_M).$$

Examples: $\Lambda\mathbf{BC}$ (Bellantoni-Cook 92) translated into Λ ,
 $1\lambda^p(\mathbf{W})$ (Leivant-Marion 94), **LLL** (Girard 98),
SLL (Lafont 04), **DLAL** (Baillot-Terui 04)

- \mathcal{T} is **intensionally P-complete** if in addition

$$f_M \in \mathbf{FP} \implies M \in \mathcal{T}.$$

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.
- Extensional P-completeness is very easy to achieve: it is basically sufficient if the following are typable:

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.
- Extensional P-completeness is very easy to achieve: it is basically sufficient if the following are typable:
 - polynomials

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.
- Extensional P-completeness is very easy to achieve: it is basically sufficient if the following are typable:
 - polynomials
 - one-step transitions of TM

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.
- Extensional P-completeness is very easy to achieve: it is basically sufficient if the following are typable:
 - polynomials
 - one-step transitions of TM
 - (restricted) iteration scheme

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.
- Extensional P-completeness is very easy to achieve: it is basically sufficient if the following are typable:
 - polynomials
 - one-step transitions of TM
 - (restricted) iteration scheme
- But so what?

Extensional vs Intensional P-completeness

- Intensional P-completeness implies extensional one.
- Extensional P-completeness is very easy to achieve: it is basically sufficient if the following are typable:
 - polynomials
 - one-step transitions of TM
 - (restricted) iteration scheme
- **But so what?**
- Intensional P-completeness is desired...

Intensional P-completeness is not r.e.

- $\Lambda\mathbf{P}$: the set of terms representing a polynomial time function:

$$M \in \Lambda\mathbf{P} \iff f_M \in \mathbf{FP}.$$

Intensional P-completeness is not r.e.

- $\Lambda\mathbf{P}$: the set of terms representing a polynomial time function:

$$M \in \Lambda\mathbf{P} \iff f_M \in \mathbf{FP}.$$

- **Fact:** $\Lambda\mathbf{P}$ is neither r.e. nor co-r.e.

Intensional P-completeness is not r.e.

- **$\Lambda\mathbf{P}$** : the set of terms representing a polynomial time function:

$$M \in \Lambda\mathbf{P} \iff f_M \in \mathbf{FP}.$$

- **Fact**: $\Lambda\mathbf{P}$ is neither r.e. nor co-r.e.
- **Proof**: Reduction of Hilbert's 10th problem.

Intensional P-completeness is not r.e.

- **$\Lambda\mathbf{P}$** : the set of terms representing a polynomial time function:

$$M \in \Lambda\mathbf{P} \iff f_M \in \mathbf{FP}.$$

- **Fact**: $\Lambda\mathbf{P}$ is neither r.e. nor co-r.e.
- **Proof**: Reduction of Hilbert's 10th problem.
- For any polynomial $P(X_1, \dots, X_n)$ with integer coefficients, there is M_P that works on unary integers as follows:

$$\begin{aligned} M_P(0) &= 1 \\ M_P(x+1) &= 1 && \text{if } \exists z_1, \dots, z_n. (-x \leq z_1, \dots, z_n \leq x \\ & && \wedge P(z_1, \dots, z_n) = 0) \\ &= 2 \cdot M_P(x) && \text{otherwise.} \end{aligned}$$

Intensional P-completeness is not r.e.

$$\begin{aligned}M_P(0) &= 1 \\M_P(x+1) &= 1 && \text{if } \exists z_1, \dots, z_n. (-x \leq z_1, \dots, z_n \leq x \\ & && \wedge P(z_1, \dots, z_n) = 0) \\ &= 2 \cdot M_P(x) && \text{otherwise.}\end{aligned}$$

- M_P can be considered as a program on binary words by:

$$\text{u2b}(n) = \underbrace{11 \dots 111}_{n \text{ times}} \quad \text{b2u}(\underbrace{011 \dots 100}_{n \text{ times}}) = n.$$

- $M_P \in \Lambda\mathbf{P}$ iff $P(X_1, \dots, X_n) = 0$ admits an integer solution.
- Hence $\Lambda\mathbf{P}$ is not r.e (nor co-r.e).

Intensional P-completeness is not r.e.

- Intensional P-completeness cannot be achieved via standard type systems.

Intensional P-completeness is not r.e.

- Intensional P-completeness cannot be achieved via standard type systems.
- The same holds even if Λ is replaced by, say, Λ_F (System F typable terms).

Intensional P-completeness is not r.e.

- Intensional P-completeness cannot be achieved via standard type systems.
- The same holds even if Λ is replaced by, say, Λ_F (System F typable terms).
- Cf. Given $M \in \Lambda_F$, it is decidable in Ptime whether M is typable in DLAL (Atassi-Baillet-Terui 2006).

Intensional P-completeness is not r.e.

- Intensional P-completeness cannot be achieved via standard type systems.
- The same holds even if Λ is replaced by, say, Λ_F (System F typable terms).
- Cf. Given $M \in \Lambda_F$, it is decidable in Ptime whether M is typable in DLAL (Atassi-Baillet-Terui 2006).
- We are looking for a better approximation of Λ_P .

Subclasses of ΛP

- ΛP_{SN} : the class of **Ptime strongly normalizable** lambda terms:

$$M \in \Lambda P_{SN} \iff$$

$\exists d \in N. \forall w \in W.$ **for any** reduction sequence

$$M\bar{w} \longrightarrow M_1 \longrightarrow \dots \longrightarrow M_k$$

$$k = O(|w|^d); \quad |M_i| = O(|w|^d) \text{ for any } 1 \leq i \leq k.$$

Subclasses of ΛP

- R : **feasible reduction strategy** (such as leftmost, innermost): given M , R picks up a redex of M , if any, in **Ptime**.
- ΛP_R : the class of **Ptime R -normalizable** lambda terms M :
 $\exists d \in N. \forall w \in W$. **for the R -reduction sequence**

$$M\bar{w} \longrightarrow_R M_1 \longrightarrow_R \cdots \longrightarrow M_k$$

with M_k in normal form

$$k = O(|w|^d); \quad |M_i| = O(|w|^d) \text{ for any } 1 \leq i \leq k.$$

- Finally,

$$\Lambda P_{WN} := \bigcup_{R:\text{feasible strategy}} \Lambda P_R.$$

Subclasses of ΛP

• $\Lambda P_{SN} = \Lambda P_{perp} \subseteq \Lambda P_R \subseteq \Lambda P_{WN} \subseteq \Lambda P$

Subclasses of $\Lambda\mathbf{P}$

- $\Lambda\mathbf{P}_{SN} = \Lambda\mathbf{P}_{perp} \subseteq \Lambda\mathbf{P}_R \subseteq \Lambda\mathbf{P}_{WN} \subseteq \Lambda\mathbf{P}$
- $\Lambda\mathbf{P}_{SN}, \Lambda\mathbf{P}_{WN}$ are neither r.e. nor co-r.e.

Subclasses of ΛP

- $\Lambda P_{SN} = \Lambda P_{perp} \subseteq \Lambda P_R \subseteq \Lambda P_{WN} \subseteq \Lambda P$
- $\Lambda P_{SN}, \Lambda P_{WN}$ are neither r.e. nor co-r.e.
- $DLAL \subset \Lambda P_{SN}$

Subclasses of ΛP

- $\Lambda P_{SN} = \Lambda P_{perp} \subseteq \Lambda P_R \subseteq \Lambda P_{WN} \subseteq \Lambda P$
- $\Lambda P_{SN}, \Lambda P_{WN}$ are neither r.e. nor co-r.e.
- $DLAL \subset \Lambda P_{SN}$
- $\Lambda BC \not\subset \Lambda P_{SN}, \Lambda BC \subset \Lambda P_{innermost}$.

Subclasses of ΛP

- $\Lambda P_{SN} = \Lambda P_{perp} \subseteq \Lambda P_R \subseteq \Lambda P_{WN} \subseteq \Lambda P$
- $\Lambda P_{SN}, \Lambda P_{WN}$ are neither r.e. nor co-r.e.
- $DLAL \subset \Lambda P_{SN}$
- $\Lambda BC \not\subset \Lambda P_{SN}, \Lambda BC \subset \Lambda P_{innermost}$.
- Intersection types provide **static** (i.e. reduction-free) characterization of Λ_{PSN} (though not r.e.).

Intersection types

- First introduced by (Coppo-Dezani 80)

Intersection types

- First introduced by (Coppo-Dezani 80)
- Later on, various systems have been considered:

Intersection types

- First introduced by (Coppo-Dezani 80)
- Later on, various systems have been considered:
 - With ω (CDV, Coppo-Dezani-Venneri 81),

Intersection types

- First introduced by (Coppo-Dezani 80)
- Later on, various systems have been considered:
 - With ω (CDV, Coppo-Dezani-Venneri 81),
 - With subtyping (BCD, Barendregt-Coppo-Dezani 83),

Intersection types

- First introduced by (Coppo-Dezani 80)
- Later on, various systems have been considered:
 - With ω (CDV, Coppo-Dezani-Venneri 81),
 - With subtyping (BCD, Barendregt-Coppo-Dezani 83),
 - With expansion variables (System I, Kfouly-Wells 03), etc.

Intersection types

- First introduced by (Coppo-Dezani 80)
- Later on, various systems have been considered:
 - With ω (CDV, Coppo-Dezani-Venneri 81),
 - With subtyping (BCD, Barendregt-Coppo-Dezani 83),
 - With expansion variables (System I, Kfouly-Wells 03), etc.
- Most systems satisfy (Pottinger 80, CDV 81):
 - M is typable without using $\omega \iff M$ is strongly normalizable.
 - M is typable using ω but not in the result type $\iff M$ is weakly normalizable.

Intersection types

- First introduced by (Coppo-Dezani 80)
- Later on, various systems have been considered:
 - With ω (CDV, Coppo-Dezani-Venneri 81),
 - With subtyping (BCD, Barendregt-Coppo-Dezani 83),
 - With expansion variables (System I, Kfouly-Wells 03), etc.
- Most systems satisfy (Pottinger 80, CDV 81):
 - M is typable without using $\omega \iff M$ is strongly normalizable.
 - M is typable using ω but not in the result type $\iff M$ is weakly normalizable.
- As a consequence, type inference is as difficult as normalization.

Intersection types and linearity

• Idempotency: $\sigma = \sigma \cap \sigma$.

Intersection types and linearity

- Idempotency: $\sigma = \sigma \cap \sigma$.
- Contraction: $\sigma \leq \sigma \cap \sigma$.

Intersection types and linearity

- **Idempotency:** $\sigma = \sigma \cap \sigma$.
- **Contraction:** $\sigma \leq \sigma \cap \sigma$.
- **Weakening:** $\sigma \cap \tau \leq \sigma$.

Intersection types and linearity

- **Idempotency**: $\sigma = \sigma \cap \sigma$.
- **Contraction**: $\sigma \leq \sigma \cap \sigma$.
- **Weakening**: $\sigma \cap \tau \leq \sigma$.
- **Linear intersection types** satisfy neither of them.

Intersection types and linearity

- **Idempotency**: $\sigma = \sigma \cap \sigma$.
- **Contraction**: $\sigma \leq \sigma \cap \sigma$.
- **Weakening**: $\sigma \cap \tau \leq \sigma$.
- **Linear intersection types** satisfy neither of them.
- **Affine intersection types** satisfy Weakening but not Idempotency/Contraction. Typability=SN.

Intersection types and linearity

- **Idempotency**: $\sigma = \sigma \cap \sigma$.
- **Contraction**: $\sigma \leq \sigma \cap \sigma$.
- **Weakening**: $\sigma \cap \tau \leq \sigma$.
- **Linear intersection types** satisfy neither of them.
- **Affine intersection types** satisfy Weakening but not Idempotency/Contraction. Typability=SN.
- Relationship with linear logic is suggested by (Regnier 92, Mairson-Møller 04, Carrier-Wells 04).

Explicit vs Implicit typings

- **Explicit typing:** introduces constructor/destructor to the subject.
- **Implicit typing:** does not change the subject.
- **Most** logical connectives admit both interpretations.
- **Explicit Quantifier:**

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \Lambda \alpha. M : \forall \alpha. A}$$

$$\frac{\Gamma \vdash M : \forall \alpha. A}{\Gamma \vdash MB : A[B/\alpha]}$$

- **Implicit Quantifier:**

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M : \forall \alpha. A}$$

$$\frac{\Gamma \vdash M : \forall \alpha. A}{\Gamma \vdash M : A[B/\alpha]}$$

Explicit vs Implicit typings

- Explicit Plus:

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \text{inl}(M) : A \oplus B}$$

$$\frac{x : A, \Gamma \vdash M : C \quad y : B, \Gamma \vdash N : C}{z : A \oplus B, \Gamma \vdash \text{case } z \text{ of } \text{inl}(x) \Rightarrow M \mid \text{inr}(y) \Rightarrow N : C}$$

- Implicit Plus (Union Type):

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M : A \oplus B}$$

$$\frac{x : A, \Gamma \vdash M : C \quad x : B, \Gamma \vdash M : C}{x : A \oplus B, \Gamma \vdash M : C}$$

Explicit vs Implicit typings

- Explicit Tensor:

$$\frac{\Gamma \vdash M : A \quad \Delta \vdash N : B}{\Gamma, \Delta \vdash M \otimes N : A \otimes B}$$

$$\frac{x : A, y : B, \Gamma \vdash M : C}{z : A \otimes B, \Gamma \vdash \text{let } z = x \otimes y \text{ in } M : C}$$

- Implicit Tensor (Non-idempotent intersection type):

$$\frac{\Gamma \vdash M : A \quad \Delta \vdash M : B}{\Gamma, \Delta \vdash M : A \otimes B}$$

$$\frac{x : A, x : B, \Gamma \vdash M : C}{x : A \otimes B, \Gamma \vdash M : C}$$

Explicit vs Implicit typings

- Explicit One:

$$\frac{}{\vdash * : \mathbf{1}} \quad \frac{\Gamma \vdash M : C}{x : \mathbf{1}, \Gamma \vdash \text{let } z = * \text{ in } M : C}$$

- Implicit One (Omega):

$$\frac{M \text{ arbitrary}}{\vdash M : \mathbf{1}} \quad \frac{\Gamma \vdash M : C}{x : \mathbf{1}, \Gamma \vdash M : C}$$

- Contraction, Weakening...

Explication

- For each system \mathbf{L} , $e\mathbf{L}$ (explicit), $i\mathbf{L}$ (implicit)
- Explication map: $(\bullet)^e : i\mathbf{L} \longrightarrow e\mathbf{L}$

$$\frac{\Gamma \vdash M : A \quad \Delta \vdash M : B}{\Gamma, \Delta \vdash M : A \otimes B} \implies \frac{\tilde{\Gamma} \vdash M^e : A \quad \tilde{\Delta} \vdash M^e : B}{\tilde{\Gamma}, \tilde{\Delta} \vdash M^e \otimes M^e : A \otimes B}$$

$$\frac{x : A, x : B, \Gamma \vdash M : C}{x : A \otimes B, \Gamma \vdash M : C} \implies \frac{x_1 : A, x_2 : B, \tilde{\Gamma} \vdash M^e : C}{x_0 : A \otimes B, \tilde{\Gamma} \vdash \text{let } x_0 = x_1 \otimes x_2 \text{ in } M^e}$$

- We only consider **strict types**: No $A \multimap B \otimes C$, $A \multimap B \oplus C$, $A \multimap \mathbf{1}$, etc.

Explication

- **Theorem:** For any \mathbf{L} ,

$$\frac{i\mathbf{L} \quad \longrightarrow \quad e\mathbf{L}}{\Gamma \vdash M : A \iff \tilde{\Gamma} \vdash M^e : A}$$
$$\begin{array}{ccc} M & & M^e \\ \downarrow * & \iff & \downarrow * \\ N & & N^e \end{array}$$

- **Corollary:** Subject reduction/Normalization for $e\mathbf{L} \implies$ the same for $i\mathbf{L}$.
- **$e\mathbf{L}$ term-nonuniform:** $M \otimes N, M \otimes N \otimes K, \dots$
- **$i\mathbf{L}$ type-nonuniform:** $M : A, M : A \otimes B, M : A \oplus B, \dots$

Implicit implication

- Is implicit implication useful?

$$\frac{x : A, \Gamma \vdash M : B \quad x \notin FV(M)}{\Gamma \vdash M : A \multimap B} \qquad \frac{\Gamma \vdash M : A \multimap B \quad x : A \vdash x : A}{x : A, \Gamma \vdash M : B}$$

- Yes, when **implicating LFPL** (Hofmann):

$$\frac{\frac{\frac{}{\vdash cons : \diamond \multimap A \multimap L(A) \multimap L(A)}}{x : \diamond \vdash cons : A \multimap L(A) \multimap L(A)} \quad \frac{}{a : A \vdash a : A} \quad \frac{}{l : L(A) \vdash l : L(A)}}{x : \diamond, a : A, l : L(A) \vdash cons(a, l) : L(A)}$$

$$\frac{x : \diamond, a : A, l : L(A) \vdash cons(a, l) : L(A)}{x : \diamond, l : L(A) \vdash \lambda a. cons(a, l) : A \multimap L(A)} \quad \frac{}{l : L(A) \vdash \lambda a. cons(a, l) : \diamond \multimap A \multimap L(A)}$$

- **Explication:** $i\mathbf{LFPL} \longrightarrow e\mathbf{LFPL}$ makes it the original system.

Linearization

1. $M \in SN \iff M \in i\mathbf{MAL}$ ($= i\mathbf{MLL} + \textit{Weak}$).

2. $M \in WN \iff M \in i\mathbf{MLL} + \textit{Negative 1}$

- We will sharpen 1. (The following results are also found by Ronchi Della Rocca and Gaboardi independently)
- 2. will be sharpened by Daniel de Calvalho's talk.

Strength of logic

- What is the strength of a logic?
- Explicit typing: How many terms it types.
 - $\text{MAL}_{\rightarrow} \subsetneq \text{SimTyp}_{\rightarrow} \subsetneq \text{SysF}_{\rightarrow\forall}$
- Implicit typing: How short typing derivations are.
 - In $i\text{MAL}$, normalization cost \approx derivation size
 - In $i\text{SimTyp}$, non-elementary terms have very short proofs.
 - In $i\text{SysF}$, ...
- Connection with propositional proof systems?

Type system *iMAL*

- Variables: α, β
- Intersections: $D, E ::= A_1 \otimes \dots \otimes A_n \quad (n \geq 1)$
- Types: $A, B, C ::= \alpha \mid D \multimap A$
- Environments: $\Gamma, \Delta, \Sigma ::= \{x_1 : A_1, \dots, x_n : A_n\}$

(Multiset. x_1, \dots, x_n not necessarily distinct.)

- Type inference rules:

$$\frac{}{\Gamma, x : A \vdash x : A} \text{ (var)} \quad \frac{\Gamma, x : A_1, \dots, x : A_n \vdash M : B \quad x \notin \text{Var}(\Gamma)}{\Gamma \vdash \lambda x.M : A_1 \otimes \dots \otimes A_n \multimap B} \text{ (}\multimap I\text{)}$$

$$\frac{\Gamma_0 \vdash M : A_1 \otimes \dots \otimes A_n \multimap B \quad \Gamma_1 \vdash N : A_1 \quad \dots \quad \Gamma_n \vdash N : A_n}{\Gamma_0, \Gamma_1, \dots, \Gamma_n \vdash MN : B} \text{ (}\multimap E\text{)}$$

Type system $iMAL$

- $\mathcal{D} \triangleright \Gamma \vdash M : A \iff \mathcal{D}$ is a derivation for $\Gamma \vdash M : A$.
- $\mathcal{D} \triangleright M \iff$ there are Γ, A such that $\mathcal{D} \triangleright \Gamma \vdash M : A$.
- $|M|$: the number of λ and applications in the term M .
- $|\mathcal{D}|$: the number of $(-\circ I)$ and $(-\circ E)$ in the derivation \mathcal{D} .
- **Lemma:** $\mathcal{D} \triangleright M \implies |\mathcal{D}| \geq |M|$.
- **Lemma:** M is in nf \implies there is \mathcal{D} such that $\mathcal{D} \triangleright M$ and $|\mathcal{D}| = |M|$.

Derivation size bounds normalization cost

- Subject Reduction Theorem:** $\mathcal{D} \triangleright \Gamma \vdash M : B$ and $M \longrightarrow N \implies \mathcal{D}' \triangleright \Gamma \vdash N : B$ and $|\mathcal{D}| > |\mathcal{D}'|$.

$$\frac{
 \begin{array}{c}
 [x : A_1] \cdots [x : A_n] \\
 \vdots \\
 M : B \\
 \hline
 \lambda x.M : A_1 \otimes \cdots \otimes A_n \multimap B
 \end{array}
 \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}
 }{
 (\lambda x.M)N : B
 }
 \implies
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}$$

Derivation size bounds normalization cost

- **Subject Reduction Theorem:** $\mathcal{D} \triangleright \Gamma \vdash M : B$ and $M \longrightarrow N \implies \mathcal{D}' \triangleright \Gamma \vdash N : B$ and $|\mathcal{D}| > |\mathcal{D}'|$.

$$\frac{
 \begin{array}{c}
 [x : A_1] \cdots [x : A_n] \\
 \vdots \\
 M : B
 \end{array}
 }{
 \lambda x. M : A_1 \otimes \cdots \otimes A_n \multimap B
 }
 \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}
 \implies
 \frac{
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}
 }{
 (\lambda x. M)N : B
 }$$

- **Corollary:** If $\mathcal{D} \triangleright M$, then

Derivation size bounds normalization cost

- **Subject Reduction Theorem:** $\mathcal{D} \triangleright \Gamma \vdash M : B$ and $M \longrightarrow N \implies \mathcal{D}' \triangleright \Gamma \vdash N : B$ and $|\mathcal{D}| > |\mathcal{D}'|$.

$$\frac{
 \begin{array}{c}
 [x : A_1] \cdots [x : A_n] \\
 \vdots \\
 M : B
 \end{array}
 }{
 \lambda x. M : A_1 \otimes \cdots \otimes A_n \multimap B
 }
 \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}
 \implies
 \frac{
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}
 }{
 (\lambda x. M)N : B
 }$$

- **Corollary:** If $\mathcal{D} \triangleright M$, then
 1. M strongly normalizes in $|\mathcal{D}|$ steps.

Derivation size bounds normalization cost

- **Subject Reduction Theorem:** $\mathcal{D} \triangleright \Gamma \vdash M : B$ and $M \longrightarrow N \implies \mathcal{D}' \triangleright \Gamma \vdash N : B$ and $|\mathcal{D}| > |\mathcal{D}'|$.

$$\frac{
 \begin{array}{c}
 [x : A_1] \cdots [x : A_n] \\
 \vdots \\
 M : B
 \end{array}
 }{
 \lambda x. M : A_1 \otimes \cdots \otimes A_n \multimap B
 }
 \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}
 \implies
 \frac{
 \lambda x. M : A_1 \otimes \cdots \otimes A_n \multimap B \quad N : A_1 \quad \cdots \quad N : A_n
 }{
 (\lambda x. M)N : B
 }$$

- **Corollary:** If $\mathcal{D} \triangleright M$, then
 1. M strongly normalizes in $|\mathcal{D}|$ steps.
 2. $M \longrightarrow^* N \implies |N| \leq |\mathcal{D}|$.

Normalization cost bounds derivation size

- Subject expansion does not hold in general. We refine Møller-Neergaard's simple proof of $\text{SN} \Rightarrow \text{Typability}$.

Normalization cost bounds derivation size

- Subject expansion does not hold in general. We refine Møller-Neergaard's simple proof of $SN \Rightarrow$ Typability.
- **Lemma 1:** If $\mathcal{D} \triangleright \Gamma \vdash M[N/x] : B$ and $x \in FV(M)$, then there is \mathcal{D}' such that $\mathcal{D}' \triangleright \Gamma \vdash (\lambda x.M)N : B$ and $|\mathcal{D}'| = |\mathcal{D}| + 2$.

$$\frac{
 \begin{array}{c}
 [x : A_1] \cdots [x : A_n] \\
 \vdots \\
 M : B
 \end{array}
 }{
 \lambda x.M : A_1 \otimes \cdots \otimes A_n \multimap B
 }
 \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots
 \end{array}
 \quad \Longrightarrow \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}$$

$$\frac{
 \lambda x.M : A_1 \otimes \cdots \otimes A_n \multimap B \quad N : A_1 \quad \cdots \quad N : A_n
 }{
 (\lambda x.M)N : B
 }
 \quad \Longrightarrow \quad
 \begin{array}{c}
 \vdots \\
 N : A_1 \quad \cdots \quad N : A_n \\
 \vdots \\
 M[N/x] : B
 \end{array}$$

Normalization cost bounds derivation size

- Lemma 2:** If $\mathcal{D} \triangleright \Gamma \vdash M : B$, $x \notin FV(M)$ and N is in nf, then there are \mathcal{D}' , Δ such that $\mathcal{D}' \triangleright \Gamma, \Delta \vdash (\lambda x.M)N : B$ and $|\mathcal{D}'| = |\mathcal{D}| + |N| + 1$.

$$\frac{\frac{\frac{\Gamma}{\vdots} M : B}{\lambda x.M : A \multimap B} \quad \frac{\Delta}{\vdots} N : A}{(\lambda x.M)N : B} \Longrightarrow \frac{\Gamma}{\vdots} M : B$$

- Consider the following **perpetual reduction strategy**:

$$\frac{x \in FV(M)}{(\lambda x.M)N\vec{L} \longrightarrow_P M[N/x]\vec{L}} \quad \frac{x \notin FV(M) \quad N_1 \longrightarrow_P N_2}{(\lambda x.M)N_1\vec{L} \longrightarrow_P (\lambda x.M)N_2\vec{L}}$$

$$\frac{x \notin FV(M) \quad N \text{ in nf}}{(\lambda x.M)N\vec{L} \longrightarrow_P M\vec{L}} \quad \frac{M_1 \longrightarrow_P M_2}{\lambda x.M_1 \longrightarrow_P \lambda x.M_2} \quad \frac{N_1 \longrightarrow_P N_2}{x\vec{M}N_1\vec{K} \longrightarrow_P x\vec{M}N_2\vec{K}}$$

Normalization cost bounds derivation size

- Weak Subject Expansion Theorem:** If $\mathcal{D} \triangleright N$ (with \mathcal{D} “canonical”) and $M \longrightarrow_P N$, then there is a “canonical” \mathcal{D}' such that $\mathcal{D}' \triangleright M$ and $|\mathcal{D}'| \leq |\mathcal{D}| + |M| + 1$.
- Two delicate points:** In so far as expanding along \longrightarrow_P ,
 - the previous lemmas are sufficient to obtain the type for M (Møller);
 - P -redex is not located **above** intersections, so that the increase can be bounded by $|M| + 1$:

$$\begin{array}{c}
 R : B_1 \quad \quad R : B_1 \\
 \vdots \quad \quad \quad \vdots \\
 L : A_1 \quad \cdots \quad L : A_n \\
 \hline
 L : A_1 \otimes \cdots \otimes A_n \\
 \vdots
 \end{array}
 \not\Rightarrow
 \begin{array}{c}
 R' : B_1 \quad \quad R' : B_n \\
 \vdots \quad \quad \quad \vdots \\
 L : A_1 \quad \cdots \quad L : A_n \\
 \hline
 L : A_1 \otimes \cdots \otimes A_n \\
 \vdots
 \end{array}$$

Characterization of ΛP_{SN}

- **Theorem 1:** $\mathcal{D} \triangleright M \implies M$ strongly normalizes in length $|\mathcal{D}|$ and size $|\mathcal{D}|$.

Characterization of ΛP_{SN}

- **Theorem 1:** $\mathcal{D} \triangleright M \implies M$ strongly normalizes in length $|\mathcal{D}|$ and size $|\mathcal{D}|$.
- **Theorem 2:** M strongly normalizes in length m and size $n \implies$ there is \mathcal{D} such that $\mathcal{D} \triangleright M$ and $|\mathcal{D}| \leq n(m + 1)$.

Characterization of Λ_{PSN}

- **Theorem 1:** $\mathcal{D} \triangleright M \implies M$ strongly normalizes in length $|\mathcal{D}|$ and size $|\mathcal{D}|$.
- **Theorem 2:** M strongly normalizes in length m and size $n \implies$ there is \mathcal{D} such that $\mathcal{D} \triangleright M$ and $|\mathcal{D}| \leq n(m + 1)$.
- **Static Characterization of Λ_{PSN} :** $M \in \Lambda_{PSN} \iff \exists d \in \mathbb{N}. \forall w \in W. \text{ there is } \mathcal{D}_w \text{ such that}$

$$\mathcal{D}_w \triangleright M\bar{w} \text{ and } |\mathcal{D}_w| = O(|w|^d).$$

Characterization of $\Lambda\mathbf{P}_{SN}$

- **Theorem 1:** $\mathcal{D} \triangleright M \implies M$ strongly normalizes in length $|\mathcal{D}|$ and size $|\mathcal{D}|$.
- **Theorem 2:** M strongly normalizes in length m and size $n \implies$ there is \mathcal{D} such that $\mathcal{D} \triangleright M$ and $|\mathcal{D}| \leq n(m + 1)$.
- **Static Characterization of Λ_{PSN} :** $M \in \Lambda_{PSN} \iff \exists d \in \mathbb{N}. \forall w \in W. \text{ there is } \mathcal{D}_w \text{ such that}$

$$\mathcal{D}_w \triangleright M\bar{w} \text{ and } |\mathcal{D}_w| = O(|w|^d).$$

- **In short:** $M \in \Lambda\mathbf{P}_{SN} \iff \{M\bar{w}\}_{w \in W}$ have polynomial size derivations.

Degree of intensional expressivity

- Let $M \in \Lambda\mathbf{P}_{SN}$. How complex are the derivations $\{\mathcal{D}_w \triangleright M\bar{w}\}_{w \in W}$?
- $w \mapsto \mathcal{D}_w$ is at least **Ptime** computable.
- Let $M : \mathbf{W}^n \multimap \mathbf{B} \in \mathbf{SLL}$.
- $\{(\mathcal{D}_n)\mathcal{E}_w \triangleright M\bar{w}\}_{w \in W, |w|=n}$
- The main part $\{\mathcal{D}_n\}_{n \in \mathbf{N}}$ only depends on the input length. Moreover, $n \mapsto \mathcal{D}_n$ is **Logspace** computable.
- Proposal: Degree of intensional expressivity = complexity of derivations.
- Open Problem: What about other systems such as DLAL?

Parallelization

- $M \in SN \implies M \in i\mathbf{MAL} \implies M^e \in e\mathbf{MAL}$
- In (Terui 05), we have shown:

$e\mathbf{MLL}$ Proof nets	\approx	Boolean circuits
Size	\approx	Size
Logical depth of cut-formulas	\approx	Circuit depth
Poly-size, Polylog-depth PNs	\approx	NC circuits

- If the logical depth is **shallow**, $M \mapsto M^e$ leads to **effective parallelization**.

Beyond ΛP_{SN}

- ΛP_{SN} is too small to be of practical interest.

Beyond ΛP_{SN}

- ΛP_{SN} is too small to be of practical interest.
- Iteration of if-then-else not allowed in ΛP_{SN} .

Beyond ΛP_{SN}

- ΛP_{SN} is too small to be of practical interest.
- Iteration of if-then-else not allowed in ΛP_{SN} .
- Let $M \equiv \lambda x.(\text{if } x = 0 \text{ then } x \text{ else } x)$. Then

$$M^n \bar{0} \longrightarrow_{CBV}^* M^{n-1} \bar{0} \longrightarrow_{CBV}^* \cdots \longrightarrow_{CBV}^* \bar{0}$$

takes $O(n)$ steps.

$$M \bar{0} \longrightarrow_{CBV}^* \text{if } M^{n-1} \bar{0} = \bar{0} \text{ then } M^{n-1} \bar{0} \text{ else } M^{n-1} \bar{0} \\ \longrightarrow_{CBV}^* \cdots \longrightarrow_{CBV}^* \bar{0}$$

takes $O(2^n)$ steps.

Beyond ΛP_{SN}

- ΛP_{SN} is too small to be of practical interest.
- Iteration of if-then-else not allowed in ΛP_{SN} .
- Let $M \equiv \lambda x.(\text{if } x = 0 \text{ then } x \text{ else } x)$. Then

$$M^n \bar{0} \longrightarrow_{CBV}^* M^{n-1} \bar{0} \longrightarrow_{CBV}^* \cdots \longrightarrow_{CBV}^* \bar{0}$$

takes $O(n)$ steps.

$$M \bar{0} \longrightarrow_{CBV}^* \text{if } M^{n-1} \bar{0} = \bar{0} \text{ then } M^{n-1} \bar{0} \text{ else } M^{n-1} \bar{0} \\ \longrightarrow_{CBV}^* \cdots \longrightarrow_{CBV}^* 0$$

takes $O(2^n)$ steps.

- In particular, $\Lambda BC \not\subset \Lambda P_{SN}$.

Beyond ΛP_{SN}

- Let R be a feasible reduction strategy.
- An (abstract) size function $s : \text{Derivations} \rightarrow \mathbb{N}$ is **admissible for (M, R)** if for any M_1 such that $M\bar{w} \xrightarrow{*}_R M_1$,
 1. $\mathcal{D} \triangleright M_1 \implies s(\mathcal{D}) \geq |M_1|$
 2. $(\mathcal{D}_1 \triangleright M_1 \xrightarrow{R} \mathcal{D}_2 \triangleright M_2) \implies s(\mathcal{D}_1) > s(\mathcal{D}_2)$
- **Theorem:** $M \in \Lambda P_R \iff$
 1. there is a size function s admissible for (M, R)
 2. $\{M\bar{w}\}_{w \in \mathbf{W}}$ has polynomial size derivations w.r.t. s .
- In practice, we have to find suitable size function s for each term M to be analyzed.

Case Study: a ramified Ptime system

- Recall (Leivant-Marion 94) characterizes **FP** based on lambda-calculus with pairing and some constants, where
- **Lower tier:** word algebra terms $\epsilon, 0(\epsilon), 1(0(\epsilon)), \dots$ of base type o
- **Higher tier:** Church words: $\lambda f x. f0(f1(f1(x))), \dots$
- $f \in \mathbf{FP} \iff f$ represented by

$$M_f : ((A \rightarrow A) \rightarrow A \rightarrow A) \rightarrow o$$

where $A \equiv o \times \dots \times o$.

Case Study: a ramified Ptime system

- We consider a variant characterization based on pure lambda-calculus with linear polymorphism and linear recursive types, where
- **Lower tier:** Scott words of type $\forall\alpha.\mu\beta.(\mathbf{B} \multimap \beta \multimap \alpha) \multimap \alpha \multimap \alpha$
- **Higher tier:** Church words of type $\forall\alpha.(\mathbf{B} \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow \alpha \Rightarrow \alpha$
(with $A \Rightarrow B \equiv !A \multimap B$)

The system $\mathbf{AL}_{!-\circ\forall\mu}$

- $\mathbf{AL}_{!-\circ\forall\mu}$: Intuitionistic affine linear logic with

$$\frac{\Gamma \vdash M : A \quad \alpha \notin FV(\Gamma)}{\Gamma \vdash M : \forall\alpha.A} \qquad \frac{\Gamma \vdash M : \forall\alpha.A}{\Gamma \vdash M : A[L/\alpha]}$$

$$\frac{\Gamma \vdash M : L[\mu\alpha.L/\alpha]}{\Gamma \vdash M : \mu\alpha.L} \qquad \frac{\Gamma \vdash M : \mu\alpha.L}{\Gamma \vdash M : L[\mu\alpha.L/\alpha]}$$

- where L is **purely linear**, i.e., without !.

Scott numerals

- For simplicity, we consider unary numerals rather than words.
- **Church numerals** of type $\mathbf{N}_C \equiv \forall \alpha. (\alpha \Rightarrow \alpha) \Rightarrow \alpha \Rightarrow \alpha$
nonlinear, support iteration
- **Scott numerals** of type $\mathbf{N}_S \equiv \forall \alpha. \mu \beta. (\beta \multimap \alpha) \multimap \alpha \multimap \alpha$
linear, support basic functions

$$\bar{0} \equiv \lambda x y. y \quad : \mathbf{N}_S$$

$$\overline{n+1} \equiv \lambda x y. x \bar{n} \quad : \mathbf{N}_S$$

$$suc \equiv \lambda z. \lambda x y. x z \quad : \mathbf{N}_S \multimap \mathbf{N}_S$$

$$prd \equiv \lambda z. z (\lambda x. x) \bar{0} \quad : \mathbf{N}_S \multimap \mathbf{N}_S$$

$$cond \equiv \lambda z_1 z_2 z_3. z_1 (\lambda w. z_2) z_3 \quad : \mathbf{N}_S^3 \multimap \mathbf{N}_S$$

Representation of FP

- **Theorem:** Any $f \in \mathbf{FP}$ can be represented by a closed term $M_f : \mathbf{W}_C \Rightarrow \mathbf{W}_S$.
- **Proof:** Any $M'_f : ((A \rightarrow A) \rightarrow A \rightarrow A) \rightarrow o$ with $A \equiv o \times \dots \times o$ in (LM94) can be easily translated into $\mathbf{AL}_{!-\circ\forall\mu}$.
- **Exercise:** Give a characterization of PSPACE in $\mathbf{AL}_{!-\circ\forall\mu}$.

Pruned size of derivations

- Pruned size $s(\mathcal{D})$ defined by:
- When \mathcal{D} is (var) , $s(\mathcal{D}) = 0$.
- $s(\mathcal{D}) = s(\mathcal{D}_0) + 1$ when \mathcal{D} is

$$\frac{\begin{array}{c} \vdots \mathcal{D}_0 \\ \Gamma, x : A_1, \dots, x : A_n \vdash M : B \end{array}}{\Gamma \vdash \lambda x. M : A_1 \otimes \dots \otimes A_n \multimap B} \quad (-\circ I)$$

- $s(\mathcal{D}) = s(\mathcal{D}_0) + \min(s(\mathcal{D}_1), \dots, s(\mathcal{D}_n)) + |\Gamma_1, A_1, \dots, \Gamma_n, A_n|$,
when \mathcal{D} is

$$\frac{\begin{array}{c} \vdots \mathcal{D}_0 \\ \Delta_0, \Gamma_0 \vdash M : A_1 \otimes \dots \otimes A_n \multimap B \end{array} \quad \begin{array}{c} \vdots \mathcal{D}_1 \\ \Delta_1, \Gamma_1 \vdash N : A_1 \end{array} \quad \dots \quad \begin{array}{c} \vdots \mathcal{D}_n \\ \Delta_n, \Gamma_n \vdash N : A_n \end{array}}{\Delta_0, \Gamma_0, \Delta_1, \Gamma_1, \dots, \Delta_n, \Gamma_n \vdash MN : B} \quad (-\circ E)$$

where Δ_i consists of **redundant types** (to be explained later).

Normalization bound

- Given $M : \mathbf{N}_C \Rightarrow \mathbf{N}_S$ and $\bar{n} : \mathbf{N}_C$:
- Linearize** $\mathcal{D} \triangleright M\bar{n}$ in $\mathbf{AL}_{!-\circ\forall\mu}$ into $\mathcal{D}_0 \triangleright M\bar{n} : \mathcal{L}^k -\circ \mathbf{N}_C$ in $i\mathbf{MAL}_{\forall\mu}$:

$$L \quad \mapsto \quad L$$

$$!L -\circ L \quad \mapsto \quad \mathbf{L} \equiv L \otimes \cdots \otimes L -\circ L$$

$$!(!L -\circ L) -\circ !L -\circ L \quad \mapsto \quad \mathcal{L} \equiv \mathbf{L}_1 \otimes \cdots \otimes \mathbf{L}_k -\circ \mathbf{L}_{k+1}$$

- Call the images of !-types **redundant** types.

Normalization bound

- **Strategy R :** First reduce all redices of type \mathcal{L} and \mathbf{L} to obtain M_1 , then reduce the **rightmost** $(\lambda x.M)N$ with N in nf.
- **After M_1 ,** the pruned size s is admissible for R :
 - $\mathcal{D}_i \triangleright M_i \implies s(\mathcal{D}_i) \geq |M_i|$
 - $(\mathcal{D}_i \triangleright M_i \longrightarrow_R \mathcal{D}_{i+1} \triangleright M_{i+1}) \implies s(\mathcal{D}_i) > s(\mathcal{D}_{i+1})$
- the latter because of
 - If $(\lambda x.M)N$ is the redex to be reduced, N is in nf and does not contain a free variable of redundant type.

Normalization bound

- If $\mathcal{D}_1 \triangleright M_1$, $s(M_1)$ is polynomial in N .
- By finite dispatching, the size function s' admissible for (M, R) can be obtained.
- **Theorem:** $M : \mathbf{W}_C \Rightarrow \mathbf{W}_S$ in $\mathbf{AL}_{!-\circ\forall\mu}$
 $\implies M \in \Lambda\mathbf{P}_R \subseteq \Lambda\mathbf{P}_{WN}$.

Logspace

- Let $\mathbf{W}'_C \equiv \forall \alpha. (\alpha \multimap \alpha) \Rightarrow \alpha \multimap \alpha$.
- Any $f \in \mathbf{BC}^{--}$ can be represented by $M_f : \mathbf{W}'_C \multimap \mathbf{W}_S$.
- **BC**: safe recursion

$$\begin{aligned}f(0, x; y) &= g(x; y) \\f(n + 1, x; y) &= h(n, x; f(n), y)\end{aligned}$$

- **BC⁻**: safe recursion with linear safe arguments

$$\begin{aligned}f(0, x : y) &= g(x : y) \\f(n + 1, x : y) &= h(n, x : f(n, x : y))\end{aligned}$$

- A slight extension of **BC⁻** captures LOGSPACE (Mairson and Møller-Neergaard)

Logspace

- **BC⁻**: safe recursion with linear safe arguments and without regressive arguments

$$\begin{aligned}f(0, x : y) &= g(x : y) \\f(n + 1, x : y) &= h(x : f(n))\end{aligned}$$

- Given $M : \mathbf{W}'_C \multimap \mathbf{W}_S$ and $\bar{w} : \mathbf{W}'_C$ in $\mathbf{AL}_{!-\circ\forall\mu}$,
- $M\bar{w}$ can be **linearized** into $i\mathbf{MAL}$ and **explicated** into $e\mathbf{MAL}$ and **η -expanded** as $(M\bar{w})_\eta$.
- $(M\bar{w})_\eta$ has size polynomial in $|w|$.
- **Logspace normalization procedure** for η -expanded \mathbf{MLL} (Terui 2002, Mairson 2004) could be applied...
- Seems to suggest a connection between two systems associated to Logspace.

Summary

- $\Lambda\mathbf{P}$, $\Lambda\mathbf{P}_{SN}$, $\Lambda\mathbf{P}_{WN}$ not r.e.
- Implicit vs Explicit typings:
 - Explication: $i\mathbf{L} \longrightarrow e\mathbf{L}$
 - Linearization: $SN = i\mathbf{MAL}$
- $\Lambda\mathbf{P}_{SN}$ = terms having polysize derivations in $i\mathbf{MAL}$.
- Degree of intensional expressivity = Complexity of derivations ?
- $\mathbf{W}_C \Rightarrow \mathbf{W}_S$ in $\mathbf{AL}_{!-\circ\forall\mu}$ captures FP
- Showed $M : \mathbf{W}_C \Rightarrow \mathbf{W}_S$ belongs to $\Lambda\mathbf{P}_{WN}$ by using an abstract size function.
- Suggested a connection between recursion theoretic and logical characterizations of LOGSPACE.