Principal Typings for Explicit Substitution*

Daniel Lima Ventura & Mauricio Ayala-Rincón
Departamento de Matemática - UnB

Fairouz Kamareddine
ULTRA Group - Heriot-Watt University

4th International Workshop on Higher-Order Rewriting
June 25, 2007, Paris

* Research supported by the Brazilian Research Council - CNPq
Outline

1. Background

2. Principal Typing for $\lambda dB$

3. Principal Typing for ES
   - Principal Typing for $TA_{\lambda s_e}$
   - Principal Typing for $TA_{\lambda \sigma}$

4. Conclusion and Further Work
Principal Typing

Let $\Gamma \vdash b : A$ be a type judgement in some type system $S$

- $\tau = (\Gamma, A)$ is a typing of $b$ in $S$ ($S \triangleright b : \tau$).
Principal Typing

Let $\Gamma \vdash b : A$ be a type judgement in some type system $S$

- $\tau = (\Gamma, A)$ is a typing of $b$ in $S$ ($S \triangleright b : \tau$).
- $\tau$ is a **principal typing** (PT) of $b$ if $S \triangleright b : \tau$ and $\tau$ represents any other possible typing of $b$. 
Principal Typing

Let $\Gamma \vdash b : A$ be a type judgement in some type system $S$

- $\tau = (\Gamma, A)$ is a typing of $b$ in $S$ ($S \triangleright b : \tau$).
- $\tau$ is a **principal typing** (PT) of $b$ if $S \triangleright b : \tau$ and $\tau$ represents any other possible typing of $b$.
- PT property allows *compositional* type inference
Principal Typing vs. Principal Type [Jim96]

Given term $b$ and context $\Gamma$, $A$ is a principal type of $b$ if it represents any other possible type of $b$ in $\Gamma$. 
Principal Typing vs. Principal Type [Jim96]

Given term $b$ and context $\Gamma$, $A$ is a **principal type** of $b$ if it represents any other possible type of $b$ in $\Gamma$.

**Principal Type:** $\Gamma \vdash b : ?$
Principal Typing vs. Principal Type [Jim96]

Given term $b$ and context $\Gamma$, $A$ is a **principal type** of $b$ if it represents any other possible type of $b$ in $\Gamma$.

**Principal Type**: $\Gamma \vdash b : ?$

**Example**

Question: $y : A \to A \vdash \lambda_x.(y \ x) : ?$
Principal Typing vs. Principal Type [Jim96]

Given term $b$ and context $\Gamma$, $A$ is a **principal type** of $b$ if it represents any other possible type of $b$ in $\Gamma$.

**Principal Type**: $\Gamma \vdash b : ?$

**Example**

Question: $y : A \rightarrow A \vdash \lambda x. (y \ x) : ?$

Answer: $A \rightarrow A$
Principal Typing vs. Principal Type [Jim96]

Given term $b$ and context $\Gamma$, $A$ is a **principal type** of $b$ if it represents any other possible type of $b$ in $\Gamma$.

**Principal Type**: $\Gamma \vdash b : ?$

**Example**

Question: $y : A \rightarrow A \vdash \lambda x. (y \ x) : ?$

Answer: $A \rightarrow A$

Question: $y : A \rightarrow B \vdash \lambda x. (y \ x) : ?$
Principal Typing vs. Principal Type [Jim96]

Given term \( b \) and context \( \Gamma \), \( A \) is a **principal type** of \( b \) if it represents any other possible type of \( b \) in \( \Gamma \).

**Principal Type:** \( \Gamma \vdash b : ? \)

**Example**

Question: \( y : A \to A \vdash \lambda_x. (y \ x) : ? \)
Answer: \( A \to A \)

Question: \( y : A \to B \vdash \lambda_x. (y \ x) : ? \)
Answer: \( A \to B \)
Principal Typing vs. Principal Type

Principal Typing: $\vdash b : ?$
Principal Typing vs. Principal Type

Principal Typing: ? ⊢ b :?

Example
Question: ? ⊢ λx.(y x) :?
Principal Typing vs. Principal Type

**Principal Typing:** $? \vdash b : ?$

**Example**

**Question:** $? \vdash \lambda x. (y \ x) : ?$

**Answer:** $(y : A \rightarrow B, A \rightarrow B)$
Principal Typing vs. Principal Type

**Principal Typing:** \( \vdash b : ? \)

**Example**

Question: \( \vdash \lambda x.(y \ x) : ? \)

Answer: \((y : A \rightarrow B, A \rightarrow B)\)

Question: \( \vdash \lambda x.(y \ (y \ x)) : ? \)
Principal Typing vs. Principal Type

**Principal Typing:** $\vdash b : ?$

**Example**

Question: $\vdash \lambda x. (y \ x) : ?$

Answer: $(y : A \rightarrow B, A \rightarrow B)$

Question: $\vdash \lambda x. (y \ (y \ x)) : ?$

Answer: $(y : A \rightarrow A, A \rightarrow A)$
## Principal Typing vs. Principal Type

<table>
<thead>
<tr>
<th></th>
<th>Principal Type</th>
<th>Principal Typing</th>
</tr>
</thead>
<tbody>
<tr>
<td>STLC</td>
<td>✓ [Hi97]</td>
<td>✓ [Wells02]</td>
</tr>
<tr>
<td>Hindley/Milner</td>
<td>✓ [DM82]</td>
<td>X [Wells02]</td>
</tr>
<tr>
<td>System F</td>
<td>✓ [KW04]</td>
<td>✓ [KW04]</td>
</tr>
<tr>
<td>System II</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Principal Typing

Definition
For some typing $\tau$ in $S$ let $Terms_S(\tau) = \{ a | S \triangleright a:\tau \}$. 
Principal Typing

Definition
For some typing τ in S let $Terms_S(\tau) = \{ a | S \triangleright a : \tau \}$.

Definition (Typing’s Partial Order)
Let $\tau \leq_S \tau'$ iff $Terms_S(\tau) \subseteq Terms_S(\tau')$.
Principal Typing

Definition (Wells’ PT[Wells02])

A typing $\tau$ in system $S$ is principal for some term $a$ iff $S \triangleright a:\tau$ and $S \triangleright a:\tau'$ implies $\tau \leq_S \tau'$. 
Principal Typing

Definition (Wells’ PT[Wells02])
A typing $\tau$ in system $S$ is principal for some term $a$ iff $S \triangleright a : \tau$ and $S \triangleright a : \tau'$ implies $\tau \leq_S \tau'$.

Example
For $\tau_1 = (y : A \rightarrow B, A \rightarrow B)$ and $\tau_2 = (y : A \rightarrow A, A \rightarrow A)$ we have $\text{Terms}(\tau_1) \subset \text{Terms}(\tau_2)$
Principal Typing for STLC

Definition (Hindley’s PT [Wells02])

Let $\tau = (\Gamma, B)$, where $TA_\lambda \triangleright a : \tau$. $\tau$ is principal typing of $a$ iff for any typing $\tau' = (\Gamma', B')$ where $TA_\lambda \triangleright a : \tau'$, then exists some type substitution $s$ such that $s(\Gamma) \subseteq \Gamma'$ and $s(B) = B'$. 
Principal Typing for STLC

Definition (Hindley’s PT \[\text{Wells02}\])
Let \(\tau = (\Gamma, B)\), where \(\text{TA}_\lambda \triangleright a : \tau\). \(\tau\) is principal typing of \(a\) iff for any typing \(\tau' = (\Gamma', B')\) where \(\text{TA}_\lambda \triangleright a : \tau'\), then exists some type substitution \(s\) such that \(s(\Gamma) \subseteq \Gamma'\) and \(s(B) = B'\).

Theorem ([Wells02])
A typing \(\tau\) is principal according to Wells’ PT iff \(\tau\) is principal according to Hindley’s PT.
Simple Type System

**Definition (Simple Types and Contexts)**

**Types** \( A ::= K \mid A \rightarrow A \)

**Contexts** \( \Gamma ::= \text{nil} \mid A.\Gamma \)
Simple Type System

**Definition (Simple Types and Contexts)**

**Types** \( A ::= K \mid A \to A \)  

**Contexts** \( \Gamma ::= \text{nil} \mid A.\Gamma \)

- \( |\Gamma|: \Gamma \) length.
Simple Type System

Definition (Simple Types and Contexts)

**Types** \( A ::= K | A \to A \)

**Contexts** \( \Gamma ::= \text{nil} | A.\Gamma \)

- \(|\Gamma|\): \(\Gamma\) length.
- For \(\Gamma = A_1.A_2.\cdots.A_m\):
  \(\Gamma_{\leq n} = A_1.\cdots.A_n\)
  \(\Gamma_{\geq n} = A_n.\cdots.A_m\)
Simple Type System

Definition (Simple Types and Contexts)

**Types** \[ A ::= K | A \to A \]

**Contexts** \[ \Gamma ::= \text{nil} | A.\Gamma \]

- \(|\Gamma|\): \(\Gamma\) length.
- For \(\Gamma = A_1.A_2.\cdots.A_m\):
  - \(\Gamma_{\leq n} = A_1.\cdots.A_n\)
  - \(\Gamma_{\geq n} = A_n.\cdots.A_m\)
- \(\Gamma_{< n}\) and \(\Gamma_{> n}\) defined similar
The System $TA_{\lambda dB}$

Syntax

Terms $a ::= n | (a a) | \lambda a$

Typing Rules

$A.\Gamma \vdash 1 : A$ (Var)

$\Gamma \vdash n : B$

$A.\Gamma \vdash n + 1 : B$ (Varn)

$A.\Gamma \vdash b : B$

$\Gamma \vdash \lambda b : A \rightarrow B$ (Lambda)

$\Gamma \vdash a : A \rightarrow B$

$\Gamma \vdash b : A$

$\Gamma \vdash (a b) : B$ (App)
PT in $TA_{\lambda dB}$

Lemma (Weakening)

If $\Gamma \vdash a : B$, then $\Gamma . A \vdash a : B$ for any type $A$.

Definition (PT in $TA_{\lambda dB}$)

Let $\tau = (\Gamma, B)$, where $TA_{\lambda dB} \triangleright a : \tau$. $\tau$ is principal typing of $a$ iff for any typing $\tau' = (\Gamma', B')$ where $TA_{\lambda dB} \triangleright a : \tau'$, then exists some type substitution $s$ such that $s(\Gamma) = \Gamma' \leq |\Gamma|$ and $s(B) = B'$. 
PT for $TA_{\lambda dB}$

Theorem (Correspondence for $TA_{\lambda dB}$)

A typing $\tau$ is PT of $b$ in $TA_{\lambda dB}$ iff $\tau$ is principal of $b$ according to Wells’ PT
PT for $TA_{\lambda dB}$

Theorem (Correspondence for $TA_{\lambda dB}$)

A typing $\tau$ is PT of $b$ in $TA_{\lambda dB}$ iff $\tau$ is principal of $b$ according to Wells’ PT

Theorem (PT for $TA_{\lambda dB}$)

$TA_{\lambda dB}$ satisfies PT property
Type Inference for $TA_{\lambda dB}$ [AyMu2000]

1st Given term $a$, let $a'$ be its annotated version.

Ex: for $a = \lambda.(2 \ 1)$, $a' = (\lambda.(2 \ 1 \ 1 \ 1 \ 1) \ 1 \ 1 \ 1 \ 1) \ 1 \ 1 \ 1 \ 1$
Type Inference for $TA_{\lambda dB}$ [AyMu2000]

1st Given term $a$, let $a'$ be its annotated version.
Ex: for $a = \lambda.( 2 \ 1 )$, $a' = ( \lambda.( 2 \ \Gamma_1 1 \Gamma_2 )\Gamma_3 )\Gamma_4$

2nd Let $R_0$ be the set of subterms of $a'$. Start using the type inference algorithm on $\langle R_0, \emptyset \rangle$
Type Inference for $TA_{\lambda dB}$ [AyMu2000]

1st  Given term $a$, let $a'$ be its annotated version.
Ex: for $a = \lambda.(2 \ 1)$, $a' = (\lambda.(2 \ A_1 \ 1 \ A_2)A_3)A_4$

2nd  Let $R_0$ be the set of subterms of $a'$. Start using the type inference algorithm on $\langle R_0, \emptyset \rangle$

3rd  When $\langle \emptyset, E \rangle$ is reached, $E$ is the set of equations on type and context variables
Type Inference for $TA_{\lambda dB}$ [AyMu2000]

1st Given term $a$, let $a'$ be its annotated version.
Ex: for $a = \lambda.(2 \Gamma_1^1 1 \Gamma_2^2)\Gamma_3^3)\Gamma_4^4$, $a' = (\lambda.(2 \Gamma_1^1 1 \Gamma_2^2)\Gamma_3^3)\Gamma_4^4$

2nd Let $R_0$ be the set of subterms of $a'$. Start using the type inference algorithm on $\langle R_0, \varnothing \rangle$

3rd When $\langle \varnothing, E \rangle$ is reached, $E$ is the set of equations on type and context variables

4th Use a first order unification algorithm to give you the m.g.u.
Type Inference for $TA_{\lambda dB}$

(Var) $\langle R \cup \{\Gamma^A\}, E \rangle \rightarrow \langle R, E \cup \{\Gamma = A.\Gamma'\} \rangle$

(Varn) $\langle R \cup \{n^A\}, E \rangle \rightarrow \langle R, E \cup \{\Gamma = A'_1.\cdots.A'_{n-1}.A.\Gamma'\} \rangle$

(Lambda) $\langle R \cup \{(\lambda.a^\Gamma_1A_1^\Gamma_2)^\Gamma_2\}, E \rangle \rightarrow \langle R, E \cup \{A_2 = A^* \rightarrow A_1, \Gamma_1 = A^*.\Gamma_2\} \rangle$

(App) $\langle R \cup \{(a^\Gamma_1^A \ b^\Gamma_2^A)^\Gamma_3\}, E \rangle \rightarrow \langle R, E \cup \{\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_3, A_1 = A_2 \rightarrow A_3\} \rangle$

Obs: $A'$, $A^*$ and $\Gamma'$ are fresh variables
The System $TA_{\lambda s_e}$ [KR97]

Syntax

Terms $a ::= n \mid (a \ a) \mid \lambda . a \mid a^{\sigma i} a \mid \varphi^j_k a$

Typing Rules

\[
\frac{\Gamma \leq_k \Gamma \geq_{k+i} a : A}{\Gamma \vdash a : A} \quad (\text{Phi}) \quad \frac{\Gamma \geq_i b : B \quad \Gamma \leq_i B . \Gamma \geq_{i} a : A}{\Gamma \vdash a^{\sigma i} b : A} \quad (\text{Sigma})
\]
**PT in $TA_{\lambda s_e}$**

**Definition (PT in $TA_{\lambda s_e}$)**

Let $\tau = (\Gamma, B)$, where $TA_{\lambda s_e} \triangleright a : \tau$. $\tau$ is principal typing of $a$ iff for any typing $\tau' = (\Gamma', B')$ where $TA_{\lambda s_e} \triangleright a : \tau'$, then exists some type substitution $s$ such that $s(\Gamma) = \Gamma' \leq |\Gamma|$ and $s(B) = B'$.

**Theorem (Correspondence for $TA_{\lambda s_e}$)**

A typing $\tau$ is PT of $b$ in $TA_{\lambda s_e}$ iff $\tau$ is principal of $b$ according to Wells’ PT definition.
PT for $TA_{\lambda s_e}$

Theorem (PT for $TA_{\lambda s_e}$)

$TA_{\lambda s_e}$ satisfies PT property
PT for $TA_{\lambda s_e}$

**Theorem (PT for $TA_{\lambda s_e}$)**

$TA_{\lambda s_e}$ satisfies PT property

\[
\text{(Sigma)} \quad \langle R \cup \{ (a_{A_1}^{\Gamma_1} \sigma^i b_{A_2}^{\Gamma_2})_{A_3} \}, E \rangle \rightarrow \\
\langle R, E \cup \{ A_1 = A_3, \Gamma_1 = A'_1 \cdots A'_{i-1} \cdot A_2 \cdot \Gamma_2, \Gamma_3 = A'_1 \cdots A'_{i-1} \cdot \Gamma_2 \} \rangle,
\]

\[
\text{(Phi)} \quad \langle R \cup \{ (\phi_k^i a_{A_1}^{\Gamma_1})_{A_2} \}, E \rangle \rightarrow \\
\langle R, E \cup \{ A_1 = A_2, \Gamma_2 = A'_1 \cdots A'_{k+i-1} \cdot \Gamma', \Gamma_1 = A'_1 \cdots A'_k \cdot \Gamma' \} \rangle,
\]
The System $\mathcal{T}_{\lambda\sigma}[ACCL91]$  

Syntax

Terms $a ::= 1 \mid (a\ a) \mid \lambda\ a \mid a[s]$

Substitution $s ::= id \mid \uparrow \mid a.s \mid s \circ s$

Typing rules

Terms

\[
A.\Gamma \vdash 1 : A \quad (\text{var})
\]

\[
\Gamma \vdash a : A \rightarrow B \quad \Gamma \vdash b : A
\frac{}{\Gamma \vdash (a\ b) : B} \quad (\text{app})
\]

\[
A.\Gamma \vdash b : B
\frac{}{\Gamma \vdash \lambda\ b : A \rightarrow B} \quad (\text{lambda})
\]

\[
\Gamma \vdash s \triangleright \Gamma' \quad \Gamma' \vdash a : A
\frac{}{\Gamma \vdash a[s] : A} \quad (\text{clos})
\]

Substitutions

\[
\Gamma \vdash id \triangleright \Gamma \quad (\text{id})
\]

\[
\Gamma \vdash a : A \quad \Gamma \vdash s \triangleright \Gamma'
\frac{}{\Gamma \vdash a.s \triangleright \ A.\Gamma'} \quad (\text{cons})
\]

\[
A.\Gamma \vdash \uparrow \triangleright \Gamma \quad (\text{shift})
\]

\[
\Gamma \vdash s'' \triangleright \Gamma'' \quad \Gamma'' \vdash s' \triangleright \Gamma'
\frac{}{\Gamma \vdash s' \circ s'' \triangleright \Gamma'} \quad (\text{comp})
\]
**PT in \( TA_{\lambda\sigma} \)**

**Lemma (Weakening)**

*If \( \Gamma \vdash a : B \), then \( \Gamma.A \vdash a : B \) for any type \( A \). Particularly, if \( \Gamma \vdash s \triangleright \Gamma' \) then \( \Gamma.A \vdash s \triangleright \Gamma'.A \).*
**PT in TA}_{\lambda \sigma}**

**Lemma (Weakening)**

*If* $\Gamma \vdash a : B$, *then* $\Gamma . A \vdash a : B$ *for any type* $A$. *Particularly, if* $\Gamma \vdash s \triangleright \Gamma'$ *then* $\Gamma . A \vdash s \triangleright \Gamma'. A$.

**Definition (PT in TA}_{\lambda \sigma}**

Let $\tau = (\Gamma, \mathcal{T})$, *where* $\mathcal{T}_{\lambda \sigma} \triangleright a : \tau$.
PT in $TA_{\lambda\sigma}$

**Lemma (Weakening)**

If $\Gamma \vdash a : B$, then $\Gamma.A \vdash a : B$ for any type $A$. Particularly, if $\Gamma \vdash s \triangleright \Gamma'$ then $\Gamma.A \vdash s \triangleright \Gamma'.A$.

**Definition (PT in $TA_{\lambda\sigma}$)**

Let $\tau = (\Gamma, \mathcal{T})$, where $TA_{\lambda\sigma} \triangleright a : \tau.\tau$ is principal typing of $a$ iff for any typing $\tau' = (\Gamma', \mathcal{T}')$ where $TA_{\lambda\sigma} \triangleright a : \tau'$, then exists some type substitution $s$ such that $s(\Gamma) = \Gamma' \leq |\Gamma|$ and: if $\mathcal{T}$ is a type then $s(\mathcal{T}) = \mathcal{T}'$ otherwise $s(\mathcal{T}) = \mathcal{T}' \leq |\mathcal{T}|$. 


PT for $TA_{\lambda\sigma}$

Theorem (Correspondence for $TA_{\lambda\sigma}$)

A typing $\tau$ is PT of $b$ in $TA_{\lambda\sigma}$ iff $\tau$ is principal of $b$ according to Wells’ PT definition.
PT for $TA_{\lambda \sigma}$

Theorem (Correspondence for $TA_{\lambda \sigma}$)

A typing $\tau$ is PT of $b$ in $TA_{\lambda \sigma}$ iff $\tau$ is principal of $b$ according to Wells’ PT definition

Theorem (PT for $TA_{\lambda \sigma}$)

$TA_{\lambda \sigma}$ satisfies PT property
Type Inference for $TA_{\lambda\sigma}$ [Bo95]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Context</th>
<th>Post-condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Var)</td>
<td>$\langle R \cup {1^\Gamma_A}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma = A.\Gamma'} \rangle$</td>
</tr>
<tr>
<td>(Lambda)</td>
<td>$\langle R \cup {(\lambda.a^{\Gamma_1}<em>{A_1})^{\Gamma_2}</em>{A_2}}, E \rangle$</td>
<td>$\langle R, E \cup {A_2 = A^* \rightarrow A_1, \Gamma_1 = A^*.\Gamma_2} \rangle$</td>
</tr>
<tr>
<td>(App)</td>
<td>$\langle R \cup {(a^{\Gamma_1}<em>{A_1} b^{\Gamma_2}</em>{A_2})^{\Gamma_3}_{A_3}}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_3, A_1 = A_2 \rightarrow A_3} \rangle$</td>
</tr>
<tr>
<td>(Clos)</td>
<td>$\langle R \cup {(a^{\Gamma_1}<em>{A_1} [s^{\Gamma_2}</em>{A_2}]^{\Gamma_3}_{A_3})}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma_1 = \Gamma_3, \Gamma_2 = \Gamma_4, A_1 = A_2} \rangle$</td>
</tr>
<tr>
<td>(Id)</td>
<td>$\langle R \cup {id^{\Gamma_1}_{\Gamma_2}}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma_1 = \Gamma_2} \rangle$</td>
</tr>
<tr>
<td>(Shift)</td>
<td>$\langle R \cup {\uparrow^{\Gamma_1}_{\Gamma_2}}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma_1 = \Gamma_2} \rangle$</td>
</tr>
<tr>
<td>(Cons)</td>
<td>$\langle R \cup {(a^{\Gamma_1}<em>{A_1} s^{\Gamma_2}</em>{A_2})^{\Gamma_3}<em>{A_3} t^{\Gamma_4}</em>{A_4}}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_4, \Gamma_5 = A_1.\Gamma_3} \rangle$</td>
</tr>
<tr>
<td>(Comp)</td>
<td>$\langle R \cup {(s^{\Gamma_1}<em>{A_2} \circ t^{\Gamma_2}</em>{A_3})^{\Gamma_4}_{A_4}}, E \rangle$</td>
<td>$\langle R, E \cup {\Gamma_1 = \Gamma_4, \Gamma_2 = \Gamma_6, \Gamma_3 = \Gamma_5} \rangle$</td>
</tr>
</tbody>
</table>
Conclusion

- PT is not a trivial property
Conclusion

- PT is not a trivial property
- A definition of PT for $\lambda$-calculus in de Bruijn notation was proposed and proved correct
Conclusion

- PT is not a trivial property
- A definition of PT for $\lambda$-calculus in de Bruijn notation was proposed and proved correct
- The PT property for $TA_{\lambda dB}$ was proved
Conclusion

- PT is not a trivial property
- A definition of PT for $\lambda$-calculus in de Bruijn notation was proposed and proved correct
- The PT property for $TA_{\lambda dB}$ was proved
- A definition of PT for $\lambda s_e$ and $\lambda \sigma$ were proposed and proved correct
Conclusion

- PT is not a trivial property
- A definition of PT for $\lambda$-calculus in de Bruijn notation was proposed and proved correct
- The PT property for $TA_{\lambda dB}$ was proved
- A definition of PT for $\lambda s_e$ and $\lambda \sigma$ were proposed and proved correct
- The PT property were proved using a type inference algorithm for $\lambda s_e$ [AyMu2000] and $\lambda \sigma$ [Bo95].
Further Work

- Verify if the PT property holds for the corresponding systems with open terms.
Further Work

- Verify if the PT property holds for the corresponding systems with open terms.
- Research $\lambda\sigma$ and $\lambda s_e$ with intersection types and study PT for these new systems.
Further Work

- Verify if the PT property holds for the corresponding systems with open terms.
- Research $\lambda\sigma$ and $\lambda s_e$ with intersection types and study PT for these new systems.
- Type unification + substitution calls for expansion variables.
Explicit Substitutions.

M. Ayala-Rincón and C. Muñoz.
Explicit Substitutions and All That.

P. Borovanský.
Implementation of Higher-Order Unification Based on Calculus of Explicit Substitutions.

N.G. de Bruijn.
Lambda-Calculus Notation with Nameless Dummies, a Tool for Automatic Formula Manipulation, with Application to the Church-Rosser Theorem.

L. Damas and R. Milner.
Principal Type-Schemes for Functional Programs.

J. R. Hindley.
Basic Simple Type Theory.

T. Jim.
What are principal typings and what are they good for?
Background
Principal Typing for \( \lambda dB \)
Principal Typing for ES

Conclusion and Further Work

A.J. Kfoury and J.B. Wells
Principality and type inference for intersection types using expansion variables,

F. Kamareddine and A. Ríos.
Extending a \( \lambda \)-calculus with explicit substitution which preserves strong normalisation into a confluent calculus on open terms,

J.B. Wells
The essence of principal typings,