A positive perspective on term representation: work in progress

Jui-Hsuan Wu (Ray) and Dale Miller

Inria Saclay & LIX, Institut Polytechnique de Paris

LFMTP 2022, Haifa, Israel

1 August 2022

Outline

Terms

- Terms (or expressions) exist in various settings.

Terms

- Terms (or expressions) exist in various settings.

  Mathematics (equations, formulas, proofs, etc) / Programming languages (compilers, interpreters, etc) / Proof assistants.

Terms

- Terms (or expressions) exist in various settings.

  Mathematics (equations, formulas, proofs, etc) / Programming languages (compilers, interpreters, etc) / Proof assistants.

- There are different formats for terms:

```
(1 + 2) + (1 + (1 + 2))
let x = 1 + 2 in let y = (1 + (1 + 2)) in x + y
let x = 1 + 2 in let y = 1 + x in x + y
```

Terms

- Terms (or expressions) exist in various settings.

  Mathematics (equations, formulas, proofs, etc) / Programming languages (compilers, interpreters, etc) / Proof assistants.

- There are different formats for terms:

  ```
  (1 + 2) + (1 + (1 + 2))
  let x = 1 + 2 in let y = (1 + (1 + 2)) in x + y
  let x = 1 + 2 in let y = 1 + x in x + y
  ```

- Even some graphical representations:

  (Labelled) Trees, Directed acyclic graphs (DAGs), etc.

Terms

- Terms (or expressions) exist in various settings.

  Mathematics (equations, formulas, proofs, etc) / Programming languages (compilers, interpreters, etc) / Proof assistants.

- There are different formats for terms:

  ```
  (1 + 2) + (1 + (1 + 2))
  let x = 1 + 2 in let y = (1 + (1 + 2)) in x + y
  let x = 1 + 2 in let y = 1 + x in x + y
  ```

- Even some graphical representations:

  (Labelled) Trees, Directed acyclic graphs (DAGs), etc.

- What to do with terms? Equality, substitution, evaluation, etc.

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - ▷ Possible interaction between different formats for terms!

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - Possible interaction between different formats for terms!
- Why proof theory?

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - ▷ Possible interaction between different formats for terms!
- Why proof theory?
  - ▷ Highly principled and mathematically sound means for describing syntactic structures.

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - ▷ Possible interaction between different formats for terms!
- Why proof theory?
  - ▷ Highly principled and mathematically sound means for describing syntactic structures.
  - ▷ Proofs-as-terms, but not proofs-as-programs!

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - ▷ Possible interaction between different formats for terms!
- Why proof theory?
  - ▷ Highly principled and mathematically sound means for describing syntactic structures.
  - ▷ Proofs-as-terms, but not proofs-as-programs!
- Which proof system to use?

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - ▷ Possible interaction between different formats for terms!
- Why proof theory?
  - ▷ Highly principled and mathematically sound means for describing syntactic structures.
  - ▷ Proofs-as-terms, but not proofs-as-programs!
- Which proof system to use?
  - Sequent calculus? Too little structure, too much non-essential information (rule permutation).

Proof theory for term representations

- A framework for describing/unifying/justifying different term structures.
  - ▷ Possible interaction between different formats for terms!
- Why proof theory?
  - ▷ Highly principled and mathematically sound means for describing syntactic structures.
  - ▷ Proofs-as-terms, but not proofs-as-programs!
- Which proof system to use?
  - Sequent calculus? Too little structure, too much non-essential information (rule permutation).
  - Focused proof system *LJF*:
    - ▷ Focusing: large-scale rules.
    - ▷ Polarization: flexibility on forms of proofs (terms).

Focusing

- Introduced by Andreoli (1992) to reduce non-determinism in proof search for LL.

Focusing

- Introduced by Andreoli (1992) to reduce non-determinism in proof search for LL.
-
Rule	invertible	$\leftrightarrow$	non-invertible
Information	non-essential	$\leftrightarrow$	essential
Phase	negative $\Uparrow$	$\leftrightarrow$	positive $\Downarrow$

$\Rightarrow$ Two-phase structure of focused proofs.

Focusing

- Introduced by Andreoli (1992) to reduce non-determinism in proof search for *LL*.
Rule	invertible	$\leftrightarrow$	non-invertible
Information	non-essential	$\leftrightarrow$	essential
Phase	negative $\Uparrow$	$\leftrightarrow$	positive $\Downarrow$

  $\Rightarrow$ Two-phase structure of focused proofs.
- Applied to LJ and LK: LJT, LJQ, LKT, LKQ, etc.

Focusing

- Introduced by Andreoli (1992) to reduce non-determinism in proof search for *LL*.
-

Rule	invertible	$\leftrightarrow$	non-invertible
Information	non-essential	$\leftrightarrow$	essential
Phase	negative $\Uparrow$	$\leftrightarrow$	positive $\Downarrow$

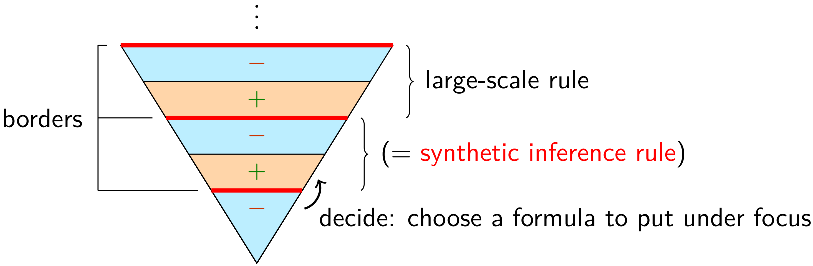
  $\Rightarrow$ Two-phase structure of focused proofs.
- Applied to LJ and LK: LJT, LJQ, LKT, LKQ, etc.
- Polarization: *LJF* and *LKF* by Liang and Miller (2009).

Focusing

- Introduced by Andreoli (1992) to reduce non-determinism in proof search for *LL*.
Rule	invertible	$\leftrightarrow$	non-invertible
Information	non-essential	$\leftrightarrow$	essential
Phase	negative $\Uparrow$	$\leftrightarrow$	positive $\Downarrow$

  $\Rightarrow$ Two-phase structure of focused proofs.
- Applied to LJ and LK: LJT, LJQ, LKT, LKQ, etc.
- Polarization: *LJF* and *LKF* by Liang and Miller (2009).
- Large-scale rules (not phases!): *synthetic inference rules*.

Two-phase structure, borders, and large-scale rules



$$
\dfrac{
\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\,ax \qquad \dfrac{}{\vdash B, B^\perp}\,ax}{\vdash A^\perp, B^\perp, A \otimes B}\,\otimes}{\vdash A^\perp, B^\perp \oplus (C^\perp \otimes D^\perp), A \otimes B}\,\oplus_1
\qquad
\dfrac{\dfrac{}{\vdash A^\perp, A}\,ax \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash C, C^\perp}\,ax \qquad \dfrac{}{\vdash D^\perp, D}\,ax}{\vdash C, D, C^\perp \otimes D^\perp}\,\otimes}{\vdash C, D, B^\perp \oplus (C^\perp \otimes D^\perp)}\,\oplus_2}{\vdash C \parr D, B^\perp \oplus (C^\perp \otimes D^\perp)}\,\parr}{\vdash A^\perp, B^\perp \oplus (C^\perp \otimes D^\perp), A \otimes (C \parr D)}\,\otimes
}{
\dfrac{\vdash A^\perp, B^\perp \oplus (C^\perp \otimes D^\perp), (A \otimes B) \,\&\, (A \otimes (C \parr D))}{\vdash A^\perp \parr (B^\perp \oplus (C^\perp \otimes D^\perp)), (A \otimes B) \,\&\, (A \otimes (C \parr D))}\,\parr
}\,\&
$$

The *LJF* system with only implication

- Formulas are built using atomic formulas and implication.

The *LJF* system with only implication

- Formulas are built using atomic formulas and implication.
- We work with polarized formulas.
  - Implications are negative.
  - Atomic formulas are either positive or negative.
    (forward-chaining / backchaining)

The *LJF* system with only implication

- Formulas are built using atomic formulas and implication.
- We work with polarized formulas.
  - Implications are negative.
  - Atomic formulas are either positive or negative.
    (forward-chaining / backchaining)
- A polarized formula (resp. theory) is a formula (resp. theory) together with an atomic bias assignment $\delta : \mathcal{A} \to \{+, -\}$.

The *LJF* system with only implication

- Formulas are built using atomic formulas and implication.
- We work with polarized formulas.
  - Implications are negative.
  - Atomic formulas are either positive or negative.
    (forward-chaining / backchaining)
- A polarized formula (resp. theory) is a formula (resp. theory) together with an atomic bias assignment $\delta : \mathcal{A} \to \{+, -\}$.
- Different polarizations do not affect provability, but give different forms of proofs.
  - ▷ If a sequent is provable in *LJF* for some polarization, then it is provable for all such polarizations.

The *LJF* system with only implication

Decide, Release, and Store Rules

$$\frac{N, \Gamma \Downarrow N \vdash A}{N, \Gamma \vdash A}\ D_l \qquad \frac{\Gamma \vdash P \Downarrow}{\Gamma \vdash P}\ D_r \qquad \frac{\Gamma \Uparrow P \vdash A}{\Gamma \Downarrow P \vdash A}\ R_l \qquad \frac{\Gamma \vdash N \Uparrow}{\Gamma \vdash N \Downarrow}\ R_r$$

$$\frac{\Gamma, C \Uparrow \Theta \vdash \Delta' \Uparrow \Delta}{\Gamma \Uparrow \Theta, C \vdash \Delta' \Uparrow \Delta}\ S_l \qquad \frac{\Gamma \Uparrow \Theta \vdash A}{\Gamma \Uparrow \Theta \vdash A \Uparrow}\ S_r$$

Initial Rules

$$\frac{A \text{ positive}}{A, \Gamma \vdash A \Downarrow}\ I_r \qquad \frac{A \text{ negative}}{\Gamma \Downarrow A \vdash A}\ I_l$$

Introduction Rules for Implication

$$\frac{\Gamma \vdash B \Downarrow \quad \Gamma \Downarrow B' \vdash A}{\Gamma \Downarrow B \supset B' \vdash A} \supset L \qquad \frac{\Gamma \Uparrow \Theta, B \vdash B' \Uparrow}{\Gamma \Uparrow \Theta \vdash B \supset B' \Uparrow} \supset R$$

Synthetic inference rules

Synthetic inference rule = large-scale rule = $\Downarrow$-phase + $\Uparrow$-phase

Definition

A *left synthetic inference rule* for B is an inference rule of the form

$$\frac{\Gamma_1 \vdash A_1 \quad \dots \quad \Gamma_n \vdash A_n}{\Gamma \vdash A}\ B$$

justified by a derivation (in LJF) of the form

$$\begin{array}{c} \Gamma_1 \vdash A_1 \qquad \dots \qquad \Gamma_n \vdash A_n \\ \vdots\ \Uparrow \text{phase} \\ \\ \vdots\ \Downarrow \text{phase} \\ \dfrac{\Gamma \Downarrow B \vdash A}{\Gamma \vdash A}\ D_l \end{array}$$

Axioms as rules

Definition

Let $\mathcal{T}$ be a finite polarized theory of order 2 or less, We define $LJ\langle\mathcal{T}\rangle$ to be the extension of LJ with the left synthetic inference rules for $\mathcal{T}$. More precisely, for every left synthetic inference rule

$$\frac{B, \Gamma_1 \vdash A_1 \qquad \ldots \qquad B, \Gamma_n \vdash A_n}{B, \Gamma \vdash A}\ B$$

with $B \in \mathcal{T}$, the inference rule

$$\frac{\Gamma_1 \vdash A_1 \qquad \ldots \qquad \Gamma_n \vdash A_n}{\Gamma \vdash A}\ B$$

is added to $LJ\langle\mathcal{T}\rangle$.

Axioms as rules

Definition

Let $\mathcal{T}$ be a finite polarized theory of order 2 or less, We define $LJ\langle\mathcal{T}\rangle$ to be the extension of LJ with the left synthetic inference rules for $\mathcal{T}$. More precisely, for every left synthetic inference rule

$$\frac{B, \Gamma_1 \vdash A_1 \quad \dots \quad B, \Gamma_n \vdash A_n}{B, \Gamma \vdash A}\ B$$

with $B \in \mathcal{T}$, the inference rule

$$\frac{\Gamma_1 \vdash A_1 \quad \dots \quad \Gamma_n \vdash A_n}{\Gamma \vdash A}\ B$$

is added to $LJ\langle\mathcal{T}\rangle$.

Theorem

$\mathcal{T}, \Gamma \vdash B$ provable in $LJ \Leftrightarrow \Gamma \vdash B$ provable in $LJ\langle\mathcal{T}\rangle$.

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

If a_i are all given the negative bias,

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

If a_i are all given the negative bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma \vdash a_0 \qquad \cdots \qquad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n}$$

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

If a_i are all given the negative bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma \vdash a_0 \qquad \cdots \qquad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n}$$

"backchaining"

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

If a_i are all given the negative bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma \vdash a_0 \qquad \cdots \qquad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n}$$

"backchaining"

If a_i are all given the positive bias,

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

If a_i are all given the negative bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma \vdash a_0 \quad \cdots \quad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n}$$

"backchaining"

If a_i are all given the positive bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma, a_0, \cdots, a_{n-1}, a_n \vdash A}{\Gamma, a_0, \cdots, a_{n-1} \vdash A}$$

An example

Let $\mathcal{T}$ be the collection of formulas
$D_1 = a_0 \supset a_1, \cdots, D_n = a_0 \supset \cdots \supset a_n, \cdots$ where a_i are atomic.

If a_i are all given the negative bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma \vdash a_0 \qquad \cdots \qquad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n}$$

"backchaining"

If a_i are all given the positive bias, the inference rules in $LJ\langle\mathcal{T}\rangle$ include

$$\frac{\Gamma, a_0, \cdots, a_{n-1}, a_n \vdash A}{\Gamma, a_0, \cdots, a_{n-1} \vdash A}$$

"forward-chaining"

Backchaining and Forward-chaining

What are the proofs of $a_0 \vdash a_n$?

Backchaining and Forward-chaining

What are the proofs of $a_0 \vdash a_n$?

When a_i are all given the negative bias, we have:

$$\frac{\Gamma \vdash a_0}{\Gamma \vdash a_1} \qquad \frac{\Gamma \vdash a_0 \quad \Gamma \vdash a_1}{\Gamma \vdash a_2} \qquad \ldots \qquad \frac{\Gamma \vdash a_0 \quad \cdots \quad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n} \qquad \ldots$$

- ▷ a unique proof of exponential size

Backchaining and Forward-chaining

What are the proofs of $a_0 \vdash a_n$?

When a_i are all given the negative bias, we have:

$$\frac{\Gamma \vdash a_0}{\Gamma \vdash a_1} \qquad \frac{\Gamma \vdash a_0 \quad \Gamma \vdash a_1}{\Gamma \vdash a_2} \qquad \cdots \qquad \frac{\Gamma \vdash a_0 \quad \cdots \quad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n} \qquad \cdots$$

- ▷ a unique proof of exponential size

When a_i are all given the positive bias, we have:

$$\frac{\Gamma, a_0, a_1 \vdash A}{\Gamma, a_0 \vdash A} \qquad \frac{\Gamma, a_0, a_1, a_2 \vdash A}{\Gamma, a_0, a_1 \vdash A} \qquad \cdots \qquad \frac{\Gamma, a_0, \ldots, a_{n-1}, a_n \vdash A}{\Gamma, a_0, \ldots, a_{n-1} \vdash A} \qquad \cdots$$

- ▷ a shortest proof of linear size

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma \vdash a_0}{\Gamma \vdash a_1} \qquad \frac{\Gamma \vdash a_0 \quad \Gamma \vdash a_1}{\Gamma \vdash a_2} \qquad \cdots$$

$$\frac{\Gamma \vdash a_0 \quad \cdots \quad \Gamma \vdash a_{n-1}}{\Gamma \vdash a_n}$$

Consider the proofs of $a_0 \vdash a_4$.

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma \vdash t_0 : a_0}{\Gamma \vdash E_1\, t_0 : a_1} \qquad \frac{\Gamma \vdash t_0 : a_0 \qquad \Gamma \vdash t_1 : a_1}{\Gamma \vdash E_2\, t_0\, t_1 : a_2} \qquad \cdots$$

$$\frac{\Gamma \vdash t_0 : a_0 \qquad \cdots \qquad \Gamma \vdash t_{n-1} : a_{n-1}}{\Gamma \vdash E_n\, t_0 \cdots t_{n-1} : a_n}$$

Consider the proofs of $a_0 \vdash a_4$.

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma \vdash t_0 : a_0}{\Gamma \vdash E_1\, t_0 : a_1} \qquad \frac{\Gamma \vdash t_0 : a_0 \qquad \Gamma \vdash t_1 : a_1}{\Gamma \vdash E_2\, t_0\, t_1 : a_2} \qquad \cdots$$

$$\frac{\Gamma \vdash t_0 : a_0 \qquad \cdots \qquad \Gamma \vdash t_{n-1} : a_{n-1}}{\Gamma \vdash E_n\, t_0 \cdots t_{n-1} : a_n}$$

Consider the proofs of $d_0 : a_0 \vdash t : a_4$.

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma \vdash t_0 : a_0}{\Gamma \vdash E_1\, t_0 : a_1} \qquad \frac{\Gamma \vdash t_0 : a_0 \qquad \Gamma \vdash t_1 : a_1}{\Gamma \vdash E_2\, t_0\, t_1 : a_2} \qquad \cdots$$

$$\frac{\Gamma \vdash t_0 : a_0 \qquad \cdots \qquad \Gamma \vdash t_{n-1} : a_{n-1}}{\Gamma \vdash E_n\, t_0 \cdots t_{n-1} : a_n}$$

Consider the proofs of $d_0 : a_0 \vdash t : a_4$.

The term annotating the unique proof is

$$\begin{array}{l}
(E_4\ (E_3\ (E_2\ (E_1\ d_0)\ (E_1\ d_0)) \\
\qquad\qquad (E_2\ (E_1\ d_0)\ (E_1\ d_0))) \\
\qquad (E_3\ (E_2\ (E_1\ d_0)\ (E_1\ d_0)) \\
\qquad\qquad (E_2\ (E_1\ d_0)\ (E_1\ d_0))))
\end{array}$$

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma, a_0, a_1 \vdash A}{\Gamma, a_0 \vdash A} \qquad \frac{\Gamma, a_0, a_1, a_2 \vdash A}{\Gamma, a_0, a_1 \vdash A} \qquad \cdots$$

$$\frac{\Gamma, a_0, \cdots, a_{n-1}, a_n \vdash A}{\Gamma, a_0, \cdots, a_{n-1} \vdash A}$$

Consider the proofs of $a_0 \vdash a_4$.

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma, x_0 : a_0, x_1 : a_1 \vdash t : A}{\Gamma, x_0 : a_0 \vdash F_1 x_0 (\lambda x_1.t) : A} \qquad \frac{\Gamma, x_0 : a_0, x_1 : a_1, x_2 : a_2 \vdash t : A}{\Gamma, x_0 : a_0, x_1 : a_1 \vdash F_2 x_0 x_1 (\lambda x_2.t) : A} \qquad \cdots$$

$$\frac{\Gamma, x_0 : a_0, \cdots, x_{n-1} : a_{n-1}, x_n : a_n \vdash t : A}{\Gamma, x_0 : a_0, \cdots, x_{n-1} : a_{n-1} \vdash F_n x_0 \cdots x_{n-1} (\lambda x_n.t) : A}$$

Consider the proofs of $d_0 : a_0 \vdash t : a_4$.

Annotating rules and proofs

Consider the inference rules in the previous example and annotate them.

$$\frac{\Gamma, x_0 : a_0, x_1 : a_1 \vdash t : A}{\Gamma, x_0 : a_0 \vdash F_1 x_0 (\lambda x_1.t) : A} \qquad \frac{\Gamma, x_0 : a_0, x_1 : a_1, x_2 : a_2 \vdash t : A}{\Gamma, x_0 : a_0, x_1 : a_1 \vdash F_2 x_0 x_1 (\lambda x_2.t) : A} \qquad \cdots$$

$$\frac{\Gamma, x_0 : a_0, \cdots, x_{n-1} : a_{n-1}, x_n : a_n \vdash t : A}{\Gamma, x_0 : a_0, \cdots, x_{n-1} : a_{n-1} \vdash F_n x_0 \cdots x_{n-1} (\lambda x_n.t) : A}$$

Consider the proofs of $d_0 : a_0 \vdash t : a_4$.

The term annotating the shortest proof is

$$\begin{array}{ll}
(F_1\ d_0 & (\lambda x_1. \\
(F_2\ d_0\ x_1 & (\lambda x_2. \\
(F_3\ d_0\ x_1\ x_2 & (\lambda x_3. \\
(F_4\ d_0\ x_1\ x_2\ x_3 & (\lambda x_4.\ x_4))))))))
\end{array}$$

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the negative bias, we have the following synthetic inference rules:

$$\frac{\Gamma \vdash D \qquad \Gamma \vdash D}{\Gamma \vdash D}\ \Phi$$

$$\frac{\Gamma, D \vdash D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the negative bias, we have the following synthetic inference rules:

$$\frac{\Gamma \vdash t : D \qquad \Gamma \vdash u : D}{\Gamma \vdash D} \; \Phi$$

$$\frac{\Gamma, D \vdash D}{\Gamma \vdash D} \; \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the negative bias, we have the following synthetic inference rules:

$$\frac{\Gamma \vdash t : D \qquad \Gamma \vdash u : D}{\Gamma \vdash \Phi\ t\ u : D}\ \Phi$$

$$\frac{\Gamma, D \vdash D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the negative bias, we have the following synthetic inference rules:

$$\frac{\Gamma \vdash t : D \qquad \Gamma \vdash u : D}{\Gamma \vdash \Phi\ t\ u : D}\ \Phi$$

$$\frac{\Gamma, x : D \vdash t : D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the negative bias, we have the following synthetic inference rules:

$$\frac{\Gamma \vdash t : D \qquad \Gamma \vdash u : D}{\Gamma \vdash \Phi\ t\ u : D}\ \Phi$$

$$\frac{\Gamma, x : D \vdash t : D}{\Gamma \vdash \Psi\ (\lambda x.t) : D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the positive bias, we have the following synthetic inference rules:

$$\frac{\Gamma, D, D, D \vdash D}{\Gamma, D, D \vdash D}\ \Phi$$

$$\frac{\Gamma, D \vdash D \qquad \Gamma, D \vdash D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the positive bias, we have the following synthetic inference rules:

$$\frac{\Gamma, x : D, y : D, z : D \vdash t : D}{\Gamma, D, D \vdash D}\ \Phi$$

$$\frac{\Gamma, D \vdash D \qquad \Gamma, D \vdash D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the positive bias, we have the following synthetic inference rules:

$$\frac{\Gamma, x : D, y : D, z : D \vdash t : D}{\Gamma, x : D, y : D \vdash \Phi\ x\ y\ (\lambda z.t) : D}\ \Phi$$

$$\frac{\Gamma, D \vdash D \qquad \Gamma, D \vdash D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the positive bias, we have the following synthetic inference rules:

$$\frac{\Gamma, x : D, y : D, z : D \vdash t : D}{\Gamma, x : D, y : D \vdash \Phi\ x\ y\ (\lambda z.t) : D}\ \Phi$$

$$\frac{\Gamma, x : D \vdash t : D \qquad \Gamma, y : D \vdash u : D}{\Gamma \vdash D}\ \Psi$$

and the initial rule.

Encodings of untyped λ-terms

We use a primitive type (atomic formula) D for untyped λ-terms.

We fix a theory $\mathcal{T} = \{\Phi : D \supset D \supset D, \Psi : (D \supset D) \supset D\}$ and consider proofs of sequents of the form $\mathcal{T}, x_1 : D, \cdots, x_k : D \vdash t : D$

When D is given the positive bias, we have the following synthetic inference rules:

$$\frac{\Gamma, x : D, y : D, z : D \vdash t : D}{\Gamma, x : D, y : D \vdash \Phi\ x\ y\ (\lambda z.t) : D}\ \Phi$$

$$\frac{\Gamma, x : D \vdash t : D \qquad \Gamma, y : D \vdash u : D}{\Gamma \vdash \Psi\ (\lambda x.t)\ (\lambda y.u) : D}\ \Psi$$

and the initial rule.

Two formats for untyped λ-terms

Two different polarity assignments give **two different term structures**:

- D is negative:

x	`nvar x`	x
$\Phi\ t\ u$	`napp t u`	tu
$\Psi\ (\lambda x.t)$	`nabs (x\ t)`	$\lambda x.t$

$\rightarrow$ Top-down / tree-like structure

- D is positive:

x	`pvar x`	x
$\Phi\ x\ y\ (\lambda z.t)$	`papp x y (z\ t)`	$\mathit{name}\ z = xy\ \mathit{in}\ t$
$\Psi\ (\lambda x.t)\,(\lambda y.s)$	`pabs (x\ t) (y\ s)`	$\mathit{name}\ y = \lambda x.t\ \mathit{in}\ s$

$\rightarrow$ Bottom-up / DAG structure

Some examples for the positive-bias syntax

```
name y = app x x in name z = app y y in z
```

- Arguments of `app` are all names

Some examples for the positive-bias syntax

```
name y = app x x in name z = app y y in z
```

- Arguments of `app` are all names

```
name y1 = app x x in name y2 = app x x in
name z = app y1 y2 in z
```

- Redundant naming

Some examples for the positive-bias syntax

```
name y = app x x in name z = app y y in z
```

- Arguments of app are all names

```
name y1 = app x x in name y2 = app x x in
name z = app y1 y2 in z
```

- Redundant naming

```
name y1 = app x x in name y2 = app y y in
name z = app y1 y1 in z
```

- Vacuous naming

Some examples for the positive-bias syntax

```
name y = app x x in name z = app y y in z
```

▷ Arguments of app are all names

```
name y1 = app x x in name y2 = app x x in
name z = app y1 y2 in z
```

▷ Redundant naming

```
name y1 = app x x in name y2 = app y y in
name z = app y1 y1 in z
```

▷ Vacuous naming

```
name y1 = app x x in name y2 = app y y in
name z = app y1 y2 in z
```

```
name z = abs (x\ name y1 = app y y in y1) in z
```

▷ Parallel naming

Some examples for the positive-bias syntax

```
name y = app x x in name z = app y y in z
```

- Arguments of app are all names

```
name y1 = app x x in name y2 = app x x in
name z = app y1 y2 in z
```

- Redundant naming

```
name y1 = app x x in name y2 = app y y in
name z = app y1 y1 in z
```

- Vacuous naming

```
name y1 = app x x and y2 = app y y in
name z = app y1 y2 in z
name z = abs (x\ name y1 = app y y in y1) in z
```

- Parallel naming

Some examples for the positive-bias syntax

```
name y = app x x in name z = app y y in z
```

▷ Arguments of app are all names

```
name y1 = app x x in name y2 = app x x in
name z = app y1 y2 in z
```

▷ Redundant naming

```
name y1 = app x x in name y2 = app y y in
name z = app y1 y1 in z
```

▷ Vacuous naming

```
name y1 = app x x and y2 = app y y in
name z = app y1 y2 in z
```

```
name y1 = app y y in name z = abs (x\ y1) in z
```

▷ Parallel naming

Cut-elimination for $LJ\langle\mathcal{T}\rangle$

The following theorem[1] states that cut is admissible for the extensions of LJ with polarized theories based on synthetic inference rules.

Theorem (Cut admissibility for $LJ\langle\mathcal{T}\rangle$)

Let $\mathcal{T}$ be a finite polarized theory of order 2 or less. Then the cut rule is admissible for the proof system $LJ\langle\mathcal{T}\rangle$.

[1] Sonia Marin, Dale Miller, Elaine Pimentel, and Marco Volpe. **From axioms to synthetic inference rules via focusing.** *Annals of Pure and Applied Logic 173(5).*

Cut-elimination for $LJ\langle\mathcal{T}\rangle$

The following theorem[1] states that cut is admissible for the extensions of LJ with polarized theories based on synthetic inference rules.

Theorem (Cut admissibility for $LJ\langle\mathcal{T}\rangle$)

Let $\mathcal{T}$ be a finite polarized theory of order 2 or less. Then the cut rule is admissible for the proof system $LJ\langle\mathcal{T}\rangle$.

The proof is based on a cut elimination procedure for LJF

- ▹ This defines the notion of substitution for terms.

[1] Sonia Marin, Dale Miller, Elaine Pimentel, and Marco Volpe. **From axioms to synthetic inference rules via focusing.** *Annals of Pure and Applied Logic 173(5).*

Cut-elimination for $LJ\langle\mathcal{T}\rangle$

The following theorem[1] states that cut is admissible for the extensions of *LJ* with polarized theories based on synthetic inference rules.

Theorem (Cut admissibility for $LJ\langle\mathcal{T}\rangle$)

Let $\mathcal{T}$ be a finite polarized theory of order 2 or less. Then the cut rule is admissible for the proof system $LJ\langle\mathcal{T}\rangle$.

The proof is based on a cut elimination procedure for *LJF*

- This defines the notion of substitution for terms.

When we restrict to atomic cut formulas, the cut elimination procedure can be presented in a big-step style.

- Cuts are permuted with synthetic rules instead of *LJF* rules.

[1] Sonia Marin, Dale Miller, Elaine Pimentel, and Marco Volpe. **From axioms to synthetic inference rules via focusing.** *Annals of Pure and Applied Logic 173(5).*

Untyped λ-terms (substitution)

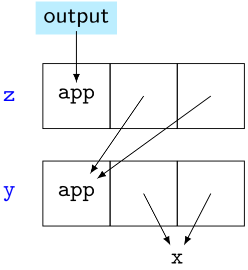
The cut-elimination procedure of *LJF* gives us the following definitions of substitutions.

```
type nsubst, psubst   tm -> (val -> tm) -> tm -> o.
```

```
nsubst T (x\ nvar x) T.
nsubst T (x\ nvar Y) (nvar Y).
nsubst T (x\ napp (R x) (S x)) (napp R' S') :-
   nsubst T R R', nsubst T S S'.
nsubst T (x\ nabs y\ R x y) (nabs y\ R' y) :-
   pi y\ nsubst T (x\ R x y) (R' y).
```

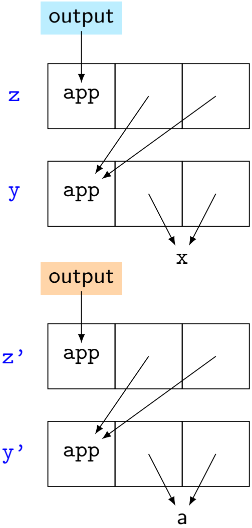
```
psubst (papp U V K) R (papp U V H) :- pi x\ psubst
    (K x) R (H x).
psubst (pabs S K)   R (pabs S H)   :- pi x\ psubst
    (K x) R (H x).
psubst (pvar U)     R (R U).
```

An example




```
name y = app x x in
name z = app y y in z
```

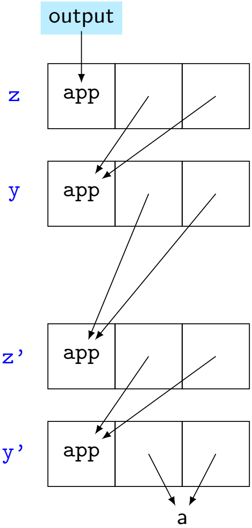
An example




```
name y = app x x in
name z = app y y in z
```

```
name y' = app a a in
name z' = app y' y' in z'
```

An example




```
name y = app x x in
name z = app y y in z
```

```
name y' = app a a in
name z' = app y' y' in
name y = app z' z' in
name z = app y y in z
```

```
name y' = app a a in
name z' = app y' y' in z'
```

Equality on terms

We have now two different formats for untyped λ-terms.

When should two such expressions be considered the same?

Equality on terms

We have now two different formats for untyped λ-terms.

When should two such expressions be considered the same?

"White box" approach:

- ▷ Look at the actual syntax of proof expressions.
  $\Rightarrow$ not working since we have two different sets of synthetic inference rules.

Equality on terms

We have now two different formats for untyped λ-terms.

When should two such expressions be considered the same?

"White box" approach:

- ▷ Look at the actual syntax of proof expressions.
  $\Rightarrow$ not working since we have two different sets of synthetic inference rules.

"Black box" approach:

- ▷ Describe *traces* by probing a term: exponential cost.
  $\hookrightarrow$ Bisimulation on graphical representations.

Graphical representations

The positive-bias syntax is closely related to some graphical representations.

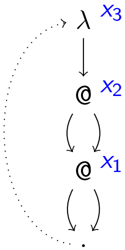
- `name` introduces new nodes and gives them a label.

Graphical representations

The positive-bias syntax is closely related to some graphical representations.

- `name` introduces new nodes and gives them a label.

Here is an example:



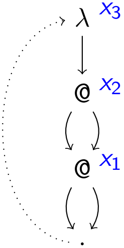

```
name x3 =
abs (x\ name x1 = app x x in
        name x2 = app x1 x1 in x2) in x3
```

Graphical representations

The positive-bias syntax is closely related to some graphical representations.

- ▷ `name` introduces new nodes and gives them a label.

Here is an example:



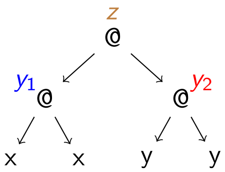

```
name x3 =
abs (x\ name x1 = app x x in
        name x2 = app x1 x1 in x2) in x3
```

Bisimulation on graphs allows to check sharing equality in linear time[2].

[2] Andrea Condoluci, Beniamino Accattoli, and Claudio Sacerdoti Coen. **Sharing equality is linear.** *PPDP 2019*.

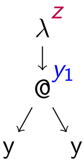
Graphical representations and parallel naming

Parallel naming can be captured by graphical representations:




```
name y1 = app x x  in  name y2 = app y y  in
name z = app y1 y2 in z

name y2 = app y y  in  name y1 = app x x  in
name z = app y1 y2 in z
```




```
name z = abs  (x\ name y1 = app y y  in y1) in
z

name y1 = app y y  in  name z = abs  (x\ y1) in
z
```

Related and future work

- Generalize to full LJF.

Related and future work

- Generalize to full LJF.
- Multi-focusing:
  - Parallel actions (parallel naming).
  - Maximal multi-focused proofs $\leftrightarrow$ graphical structures.
  - Conjecture: MMF proofs are isomorphic to some graphical structure in the case for untyped λ-terms.

Related and future work

- Generalize to full *LJF*.
- Multi-focusing:
  - ▷ Parallel actions (parallel naming).
  - ▷ Maximal multi-focused proofs $\leftrightarrow$ graphical structures.
  - ▷ Conjecture: MMF proofs are isomorphic to some graphical structure in the case for untyped λ-terms.
- Big-step cut-elimination for arbitrary cut formulas.
  - ▷ At the level of synthetic rules (not phases!).

Related and future work

- Generalize to full LJF.
- Multi-focusing:
  - ▷ Parallel actions (parallel naming).
  - ▷ Maximal multi-focused proofs $\leftrightarrow$ graphical structures.
  - ▷ Conjecture: MMF proofs are isomorphic to some graphical structure in the case for untyped λ-terms.
- Big-step cut-elimination for arbitrary cut formulas.
  - ▷ At the level of synthetic rules (not phases!).
- Connection with the literature in programming language theory (A-normal form, etc).

Related and future work

- Generalize to full LJF.
- Multi-focusing:
  - ▷ Parallel actions (parallel naming).
  - ▷ Maximal multi-focused proofs $\leftrightarrow$ graphical structures.
  - ▷ Conjecture: MMF proofs are isomorphic to some graphical structure in the case for untyped λ-terms.
- Big-step cut-elimination for arbitrary cut formulas.
  - ▷ At the level of synthetic rules (not phases!).
- Connection with the literature in programming language theory (A-normal form, etc).
- There exist some other frameworks for term structures, such as terms-as-graphs by Grabmayer. Are there some connections or overlaps?

Related and future work

- Generalize to full *LJF*.
- Multi-focusing:
  - ▷ Parallel actions (parallel naming).
  - ▷ Maximal multi-focused proofs $\leftrightarrow$ graphical structures.
  - ▷ Conjecture: MMF proofs are isomorphic to some graphical structure in the case for untyped λ-terms.
- Big-step cut-elimination for arbitrary cut formulas.
  - ▷ At the level of synthetic rules (not phases!).
- Connection with the literature in programming language theory (A-normal form, etc).
- There exist some other frameworks for term structures, such as terms-as-graphs by Grabmayer. Are there some connections or overlaps?
- Proof-theoretic methods for checking term equality.