Formalization in Constructive Type Theory of the Standardization Theorem for the Lambda Calculus using Multiple Substitution

LFMTP 2018

M. Copes, N. Szasz, A. Tasistro

Universidad ORT Uruguay

7th of July 2018

Outline

1. Introduction
2. Preliminaries
3. Proof of the Standardization Theorem
4. Proof of the Leftmost Reduction Theorem

Previous work: Formal metatheory of the Lambda Calculus using Stoughton's substitution

E. Copello, N. Szasz, and A. Tasistro

- Formalization of the Lambda Calculus in Agda using one sort of names for both free and bound variables.
- Multiple substitution based on Stoughton's paper (1988).
- Structural inductive proofs for the Church-Rosser theorem and Subject Reduction.
- Library with definitions and lemmas for manipulating substitution. Fully checked in Agda.

Present work

Our goals

- Extend these metatheoretical results by proving:
  - Standardization Theorem for β-reduction
  - Leftmost Reduction Theorem
- Assess the extent at which the library can be reused for this development.
- Attempt to use structural induction only.

The Standardization Theorem

Definition (Standard reduction sequence)

A reduction sequence is said to be standard if successive redexes are contracted from left to right, possibly with some jumps.

Theorem (Standardization)

If a term M β-reduces to a term N, then there exists a standard β-reduction sequence from M to N.

Corollary (Leftmost reduction)

If a term has a β normal form, then the leftmost-outermost reduction strategy will find this normal form

Proofs of the Standardization Theorem

- Barendregt 1982
  - Uses residuals to define standard reductions.
  - Distinguishes between internal and head reductions.
  - Based on the FD and FD!
- Takahashi 1995
  - Follows a similar structure to Barendregt's.
  - Relies on Martin-Löf's parallel reductions to represent the reduction of a set of redexes.
  - Inductive structure.

Ryo Kashima

2000

- Inductive definition of β-reducibility with a standard sequence.
- Uses neither residuals nor the separation between internal and head reductions.
- All of the definitions and proofs follow an inductive structure.

Plan

1. Introduction
2. Preliminaries
3. Proof of the Standardization Theorem
4. Proof of the Leftmost Reduction Theorem

Lambda terms

```
data Λ : Set where
  v     : V → Λ
  _·_   : Λ → Λ → Λ
  ƛ     : V → Λ → Λ
```

- One set of names for both bound and free variables without identifying alpha-equivalent terms.

Multiple Substitution

$\Sigma = V \longrightarrow \Lambda$

- Functions mapping every variable to a term.
- Constructed from the identity substitution $\iota : \Sigma$ and an update operator $\prec\!+ : \Sigma \longrightarrow V\ x\ \Lambda \longrightarrow \Sigma$
- The application of a substitution σ to a term M is noted as $M \bullet \sigma$ and defined by structural recursion on M.
- The case for the abstraction renames the abstraction variable according to χ which guarantees certain choice axioms: $(\lambda x.M) \bullet \sigma = \lambda y.(M \bullet \sigma \prec\!+ (x, y))$, where $y = \chi(\sigma, \lambda x.M)$, is the first variable not free in $\sigma \downharpoonright M$.

Alpha Conversion

```
data _~α_ : Λ → Λ → Set where
  ~v  : {x : V} → (v x) ~α v x
  ~·  : {M M' N N' : Λ} → M ~α M' → N ~α N'
      → M · N ~α M' · N'
  ~ƛ  : {M M' : Λ}{x x' y : V}
      → y # ƛ x M → y # ƛ x' M'
      → M [ x := v y ] ~α M' [ x' := v y ]
      → ƛ x M ~α ƛ x' M'
```

- Alpha equivalent terms become equivalent when submitted to the same substitution.

Alpha Reflexive Transitive Closure

```
data α-star (⇝ : Λ -> Λ -> Set) : Λ -> Λ -> Set where
    refl : ∀{M} → α-star ⇝ M M
    α-step : ∀{M N N'} → α-star ⇝ M N' → N' ~α N → α-star ⇝ M N
    append : ∀ {M N K} → α-star ⇝ M K → ⇝ K N → α-star ⇝ M N
```

- One-step and transitivity can be proven from the previous definition.

Beta reducibility

```
data _β_@_ : Λ -> Λ -> ℕ -> Set where
  outer-redex : ∀ {x A B} -> ((ƛ x A) · B) β (A [ x := B ]) @ 0
  appNoAbsL : ∀ {n A B C} -> A β B @ n -> ¬ isAbs A
    -> (A · C) β (B · C) @ n
  appAbsL : ∀ {n A B C} -> A β B @ n -> isAbs A
    -> (A · C) β (B · C) @ (suc n)
  appNoAbsR : ∀ {n A B C} -> A β B @ n -> ¬ isAbs C
    -> (C · A) β (C · B) @ (n + countRedexes C)
  appAbsR : ∀ {n A B C} -> A β B @ n -> isAbs C
    -> (C · A) β (C · B) @ (suc (n + countRedexes C))
  abs : ∀ {n x A B} -> A β B @ n -> (ƛ x A) β (ƛ x B) @ n
```

```
_⟶β_ : Λ -> Λ -> Set
M ⟶β N = Σx ℕ (\n -> M β N ↧ n)

_↠β_ : Λ -> Λ -> Set
_↠β_ = α-star _⟶β_
```

- Equivalent to the classical inductive definition of beta reducibility.

Plan

1. Introduction
2. Preliminaries
3. Proof of the Standardization Theorem
4. Proof of the Leftmost Reduction Theorem

Standard reduction sequence

- A sequence of β-reductions $A_0 \underset{n_1}{\longrightarrow} A_1 \underset{n_2}{\longrightarrow} \ldots \underset{n_k}{\longrightarrow} A_k$ is called standard if $n_1 \leq n_2 \leq \cdots \leq n_k$

```
data seqβ-st (M : Λ) : (N : Λ) -> ℕ -> Set where
  nil : seqβ-st M M 0
  α-step : ∀ {n K N} -> seqβ-st M K n -> K ~α N -> seqβ-st M N n
  β-step : ∀ {K n n₀ N} -> seqβ-st M K n -> K β N @ n₀ -> n₀ ≥ n -> seqβ-st M N n₀
```

- We add an index to represent the lower bound of subsequent reductions, i.e. the number of the last redex reduced.
- Allows performing explicit α-conversion steps inside a reduction sequence.

Theorem (Standardization)

$(\forall M,\ N)\ (M \twoheadrightarrow_\beta N \implies (\exists n)\ (seq\beta st\ M\ N\ n))$

Head reduction in application

$$(\lambda x.A_0)\ A_1\ A_2\ \ldots\ A_n \ \longrightarrow_{hap}\ A_0[x := A_1]\ A_2 \ldots\ A_n$$

```
data _⟶hap_ : Λ -> Λ -> Set where
  hap-head : ∀{x A B} -> (ƛ x A) · B ⟶hap (A [ x := B ])
  hap-chain : ∀{C A B} -> A ⟶hap B -> (A · C) ⟶hap (B · C)

_↠hap_ : Λ -> Λ -> Set
_↠hap_ = α-star _⟶hap_
```

Lemma

$$(\forall M, N, \sigma)\ (M \twoheadrightarrow_{hap} N \implies M \bullet\ \sigma \twoheadrightarrow_{hap} N \bullet\ \sigma)$$

Key Idea: Standard Reduction Relation

Kashima defines an inductive relation that captures the existence of a Stardard Reduction Sequence between two terms.

```
data _↠st_ (L : Λ) : Λ -> Set where
  st-var : ∀{x} -> L ↠hap (v x) -> L ↠st (v x)
  st-app : ∀{A B C D} -> L ↠hap (A · B) -> A ↠st C -> B ↠st D -> L ↠st (C · D)
  st-abs : ∀{x A B} -> L ↠hap (ƛ x A) -> A ↠st B -> L ↠st (ƛ x B)
  st-α   : ∀{K B} -> L ↠st K -> K ~α B -> L ↠st B
```

We now prove that:

$$M \twoheadrightarrow_{\beta} N \implies M \twoheadrightarrow_{st} N \implies (\exists n)\ (seq\beta st\ M\ N\ n)$$

Standard compatibility with substitution

Lemma

$(\forall M, N, \sigma, \sigma')\ (M \twoheadrightarrow_{st} N\ \wedge\ \sigma \rightarrow_{st} \sigma' \implies M \bullet\ \sigma \twoheadrightarrow_{st} N \bullet\ \sigma')$

- By induction on $M \twoheadrightarrow_{st} N$
- The case for the abstraction requires the use of multiple substitution in order to use the induction hypothesis.

- $(\forall x,\ M,\ A,\ B)\ (M \twoheadrightarrow_{st} (\lambda x A)\ B \implies M \twoheadrightarrow_{st} A[x := B])$
- $(\forall M,\ N)\ (M \twoheadrightarrow_{st} N \ \wedge\ N \longrightarrow_{\beta} P \implies M \twoheadrightarrow_{st} P)$

Lemma

$(\forall M,\ N)\ (M \twoheadrightarrow_{\beta} N \implies M \twoheadrightarrow_{st} N)$

Standard $\Longrightarrow$ Standard Sequence

- $(\forall M,\ N)\ (M \twoheadrightarrow_{hap} N \implies seq\beta st\ M\ N\ 0)$
- $(\forall M,\ N,\ n,\ x)\ (seq\beta st\ M\ N\ n \implies seq\beta st\ (\lambda x M)\ (\lambda x N)\ n)$

Lemma

$(\forall M,\ N)\ (M \twoheadrightarrow_{st} N \implies (\exists n)\ (seq\beta st\ M\ N\ n))$

Notice that the converse holds as well.

Standardization

Theorem (Standardization)

$(\forall M,\ N)\ (M \twoheadrightarrow_\beta N \implies (\exists n)\ (seq\beta st\ M\ N\ n))$

- Follows directly from the previous lemmas.

Plan

1 Introduction

2 Preliminaries

3 Proof of the Standardization Theorem

4 Proof of the Leftmost Reduction Theorem

Leftmost Reduction Theorem

As a corollary of the Standardization Theorem

Theorem

If M has a normal form, then the leftmost-outermost reduction strategy always finds it.

- Interesting metatheoretical result about reduction strategies.
- Beta-equality is decidable for normalizing terms.

Leftmost Reduction Theorem

Formalization in Agda

```
_⟶l_ : Λ -> Λ -> Set
M ⟶l N =  M β N ↧ 0

_→→l_ : Λ -> Λ -> Set
_→→l_ = α-star _⟶l_

nf : Λ -> Set
nf M = countRedexes M ≡ 0
```

Theorem

$(\forall M, \ N) \ (M \twoheadrightarrow_\beta N \ \wedge \ nf \ N \implies M \twoheadrightarrow_l N)$

Leftmost Reduction Theorem

Proof

Lemma

$(\forall M,\ N,\ n)\ (M\ \beta\ N\ @\ n\ \wedge\ nf\ N\ \Longrightarrow\ n \equiv 0)$

- By induction on $M\ \beta\ N\ @\ n$

Lemma

$(\forall M,\ N,\ n)\ (seq\beta st\ M\ N\ n \wedge\ nf\ N\ \Longrightarrow\ M \twoheadrightarrow_l N)$

- By induction on $seq\beta st\ M\ N\ n$ using the previous lemma for the case $\beta - step$.
- Now the *L*eftmost Reduction Theorem follows directly from $M \twoheadrightarrow_\beta N \Longrightarrow (\exists n)\ (seq\beta st\ M\ N\ n) \Longrightarrow M \twoheadrightarrow_l N$, for N in normal form.

Conclusions

- Kashima's proof is correct! (completely certified in Agda).
- Using Stoughton's substitution, the theorem only requires structural induction. Novel in relation to previous approaches:
  - McKinna and Pollack (1999)
  - Guidi (2012)
  - Emerich and Ignas Vysniauskas (2014)
- Only a few lemmas had to be added to the substitution library in order to prove the theorem.
- Proof of equivalence between Kashima's notion of beta-reducibility and the classical one.
- Introduction of a new inductive definition of a standard reduction sequence, namely $seq\beta st$.
- Leftmost Reduction Theorem

Thank you!