Substitution for Non-Wellfounded Syntax with Binders through Monoidal Categories

Benedikt Ahrens

jww Ralph Matthes and Kobe Wullaert

FICS 2024

2024-02-07

Summary

Goals

1. Construct non-wellfounded syntax, untyped and simply-typed,
2. Construct monadic substitution operation on it — variable binding à la de Bruijn, with well-scopedness through typing
3. Formally verify the construction in a computer proof assistant and make it available for further use.

Motivation

- Non-wellfounded syntax is used, for instance, in proof search
- Coinductive types and mutually inductive-coinductive types are not well integrated in all proof assistants

Approach

- Construct syntax via a suitable limit construction
- Construct substitution via (categorical) corecursion scheme

Wellfounded Vs Non-Wellfounded Syntax

Wellfounded Syntax

- Initial Semantics: we look for initial object in a category of "models"
- Syntax is specified by a notion of signature
- Substitution is given by monad/monoid structure

Non-Wellfounded Syntax

- **Not** the dual of Initial Semantics
- Syntax is specified by a notion of signature
- Underlying syntax can be constructed as terminal coalgebra, but we are interested in its algebra structure
- Substitution still given by monoid structure
- Object we construct is not specified by a universal property

Outline

1. Overview of Related Work

2. Details of the Construction of Substitution

Related Work: Wellfounded Syntax à la Fiore

- Simple notion of signatures, e.g., $\{[0,0],[1]\}$ for LC
- Syntax as a functor $\Lambda : \mathbb{F} \to \mathsf{Set}$
- Substitution structure given via monoidal structure on $[\mathbb{F}, \mathsf{Set}]$
- Only wellfounded syntax is considered

Related Work: Wellfounded Syntax à la Hirschowitz and Maggesi

- Sophisticated notion of signature: “parametrized module”, e.g., $T \mapsto T \times T + T^*$
  - Signatures allow for expression of equations between terms
  - Do not automatically admit initial objects (syntax)
- Syntax as a functor $\Lambda : \mathsf{Set} \to \mathsf{Set}$
- Substitution structure given by monad structure
- Only wellfounded syntax is considered
- Formalization of syntax and substitution in a computer proof assistant

Related Work: Substitution for Non-Wellfounded Syntax à la Matthes and Uustalu

- Signatures: endofunctors with strength, e.g., $H(X) := X \times X + X \circ \mathsf{Maybe}$
- Syntax as a functor $\mathsf{C} \to \mathsf{C}$, for suitable C
- Substitution structure given by monad structure
- Non-wellfounded syntax is considered
- Explains in detail the construction of monad structure via categorical (co)recursion à la Mendler, axiomatized via “heterogeneous substitution system”

This Work: Pushout of the Aforementioned

- Generalize the results of Matthes and Uustalu to the level of monoidal categories
- Implement constructions and proofs in a computer proof assistant
- Instantiate the constructions to concrete categories to actually construct syntax

Summary

Tool chain for non-wellfounded syntax

input	multi-sorted binding signature
output	syntax and certified monadic substitution

Outline

The Tool Chain

- Signature
  - "Combinatorial" signatures for easy specification
  - "Semantic" signatures (endofunctors) for construction of the syntax

$\rightsquigarrow$ Non-wellfounded syntax as (inverse of) terminal coalgebra
  - Via Adámek's Theorem
  - Uses ω-continuity of the semantic signature

$\rightsquigarrow$ Monad structure on the syntax via coiteration
  - Construction of a "substitution system" on syntax
  - From substitution system, derive monoid structure

Signatures

"Combinatorial" signatures over a fixed type of sorts

- $\mathsf{ar} : I \to \mathsf{list}(\mathsf{list}(\mathsf{S}) \times \mathsf{S}) \times \mathsf{S}$.
- STLC with sorts $\Rightarrow : \mathsf{S} \to \mathsf{S} \to \mathsf{S}$ and $I = (\mathsf{S} \times \mathsf{S}) + (\mathsf{S} \times \mathsf{S})$

$$\mathsf{ar}(\mathsf{inl}\langle s, t\rangle) :\equiv \big\langle [\langle [\,], s \Rightarrow t\rangle, \langle [\,], s\rangle], t \big\rangle$$
$$\mathsf{ar}(\mathsf{inr}\langle s, t\rangle) :\equiv \big\langle [\langle [s], t\rangle], s \Rightarrow t \big\rangle$$

Translation to "semantic" signature: functor with strength

- Functor $\mathsf{C} \to \mathsf{C}$, for suitably chosen C (e.g., $[\mathbb{F}, \mathsf{Set}]$)
- Strength indicates how to do "substitution in subterms": it specifies what **more** has to be done than just having substitution commute with the term constructors

Constructing (Non-Wellfounded) Syntax

Theorem (Adámek)

If C *has limits of shape* $\omega = 0 \leftarrow 1 \leftarrow 2 \leftarrow \cdots$ *and a terminal object* 1*, and* $H : \mathsf{C} \to \mathsf{C}$ *is* ω*-continuous, then the limit of* $1 \leftarrow H1 \leftarrow H^2 1 \leftarrow \cdots$ *is a terminal* H*-coalgebra.*

- We are instead interested in functors of shape $\mathsf{Id} + H(\_)$, where Id models the inclusion of variables into terms.
- H is usually a sum (one summand per constructor of the language); ω-continuity can be proved modularly from continuity of the summands.

(Co)Recursion: Substitution Systems

- $(\mathcal{V}, I, \otimes)$ a monoidal category
- $H : \mathcal{V} \to \mathcal{V}$ with a pointed tensorial strength θ for H.

(t, η, τ) with $t : \mathcal{V}$, $\eta : I \to t$ and $\tau : Ht \to t$ is a *substitution system* if, for all (z, e, f) with $z : \mathcal{V}$, $e : I \to z$ and $f : z \to t$, there is a unique morphism $h : z \otimes t \to t$ such that:

$$
\begin{array}{ccccc}
z \otimes I & \xrightarrow{1_z \otimes \eta} & z \otimes t & \xleftarrow{1_z \otimes \tau} & z \otimes Ht \\
 & & & & \downarrow{\scriptstyle \theta_{(z,e),t}} \\
\downarrow{\scriptstyle \rho_z} & & \downarrow{\scriptstyle h} & & H(z \otimes t) \\
 & & & & \downarrow{\scriptstyle Hh} \\
z & \xrightarrow{f} & t & \xleftarrow{\tau} & Ht
\end{array}
$$

Theorem

Any substitution system (t, η, τ) is an (H, θ)-monoid.

Non-Wellfounded Syntax and Substitution

Theorem

- *$\mathcal{V}$ monoidal category with binary coproducts*
- *$H : \mathcal{V} \to \mathcal{V}$ with pointed monoidal strength θ*
- *(t, out) final coalgebra of $I + H(\_)$*
- *set $[\eta, \tau] := \mathsf{out}^{-1}$*

Then (t, η, τ) is a substitution system.

- A quite simple proof can be given using the notion of "completely iterative algebra".
- Alternatively, use primitive corecursion.

Conclusion

- Construction of non-wellfounded syntax and substitution mostly on the level of monoidal categories
- Formalization of context and variable binding requires commitment to specific monoidal category; different choices possible
- Full paper with the same title: arXiv:2308.05485

Conclusion

- Construction of non-wellfounded syntax and substitution mostly on the level of monoidal categories
- Formalization of context and variable binding requires commitment to specific monoidal category; different choices possible
- Full paper with the same title: arXiv:2308.05485

Thanks for your attention!