Research Topic:
The Epsilon Calculus in Proof Search

Supervisor: Lutz Straßburger

Research Team: Partout

Research Lab: LIX

Summary

Hilbert's epsilon calculus is a quantifier-free formulation of first order logic. Due to its cumbersome syntax, it has not seen much interest in formal verification. However, the epsilon calculus can have a non-elementary speed-up over more mainstream calculi. With this project I want to bring this speed-up to the world of automated and interactive theorem proving.

1 Project description

The *epsilon-calculus* has been introduced by Hilbert [9] in the beginning of the 20th century for his famous program on proving the consistency of mathematics. It provides an alternative syntax for predicate logic in which all quantifier information in a proof is encoded by *epsilon-terms*: the formula $\exists x.A(x)$ would be represented as $A(\varepsilon_x A(x))$, where $\varepsilon_x A(x)$ stands for a term t that would make $A(t)$ true, and if no such term exists, then $\varepsilon_x A(x)$ is an arbitrary term. This quantifier-free formulation simplifies the reasoning about proofs, which led to some applications in the foundations of mathematics. However, due to its cumbersome syntax and unwieldy nature, the epsilon-calculus has received little attention from the broader proof theory community, and it is not used at all in formal verification. Just to give a simple example, the first-order formula $\exists x.\exists y.\exists z.A(x,y,z)$ would be written in the epsilon-calculus as

$$
\begin{array}{l}
A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),\\
\varepsilon_y A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),y,\\
\varepsilon_z A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),y,z)),\\
\varepsilon_z A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),\\
\varepsilon_y A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z)),y,\\
\varepsilon_z A(\varepsilon_x A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),\varepsilon_z A(x,\varepsilon_y A(x,y,\varepsilon_z A(x,y,z)),z))y,z)),z))
\end{array}
$$

This could be the end of the story.

However, surprisingly, the epsilon calculus can provide a non-elementary speed-up over more standard formulations of first-order logic [4]. This means that proofs carried out in the epsilon calculus can be shorter by a non-elementary factor than what we are used to from traditional proof systems, like the cut-free sequent calculus. Given that first-order logic is undecidable in general, it would be highly desirable, if we could have proof search algorithms that have access to this speed-up, i.e., are able to find these shorter proofs. Even though there has been some investigation from the proof complexity viewpoint [12, 1], there is no modern proof theoretical treatment of the epsilon calculus.

In this project, I would like to make the advantages of epsilon calculus accessible to practitioners in formal methods, without having to suffer from the unreadable syntax mentioned above.

For this, I am planning to use the methods that have been developed within the last years around the area of *deep inference*, which has proven to be successful in the area of propositional logics, admitting complexity benefits over traditional proof formalisms, such as exponential proof speed-ups for certain tautologies [6] and a quasipolynomial complexity cut-elimination procedure [7, 11] (in contrast to such procedures in the traditional sequent calculus, which have exponential complexity), as well as novel normalization procedures which are not exhibited by traditional proof formalisms, such as new decomposition results [2, 14] and a confluent cut-elimination procedure [13].

Similar results have been obtained for first-order logic, including a simple statement of Herbrand's Theorem, a central theorem of first-order proof theory, in the form of a decomposition theorem which is not admitted by the sequent calculus [5, 13]. Despite these promising foundations, the deep-inference proof theory of quantified logics has received comparatively little research attention.

But recently Allett [3] added the *falsifier rule* to a deep inference system for first order logic. This rule allows to introduce epsilon-terms into a proof without the need of the cumbersome encodings of quantifiers by epsilon-terms as used in the traditional epsilon-calculus, and the cut-free new proof system still has the complexity benefits of the epsilon calculus.

What is needed now is a thorough proof theoretical investigation of the new *falsifier calculus* and its potential for implementations. Furthermore, the compositionality admitted by deep-inference systems may also be able to yield a simpler syntax for the epsilon-calculus, overcoming one of the primary

shortfalls of the traditional epsilon-calculus and potentially providing a new semantics for the epsilon-calculus and first-order predicate logic. All of this might lead to a paradigm change of how we see proof search in first-order logic.

2 Prerequisites

The successful candidate needs to be familiar with the proof theory of the quantifiers in first-order logic.

References

[1] Juan P. Aguilera and Matthias Baaz. Unsound Inferences Make Proofs Shorter. *The Journal of Symbolic Logic*, 84(1):102–122, 2019.

[2] Andrea Aler Tubella, Alessio Guglielmi, and Benjamin Ralph. Removing cycles from proofs. In Valentin Goranko and Mads Dam, editors, *26th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 82 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–17. Schloss Dagstuhl–Leibniz-Zentrum fÃ¼r Informatik, 2017.

[3] Cameron Allett. Non-elementary compression of first-order proofs in deep inference using epsilon-terms. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '24, New York, NY, USA, 2024. Association for Computing Machinery.

[4] Matthias Baaz and Anela Lolic. Epsilon calculus provides shorter cut-free proofs, 2024.

[5] Kai Brünnler. Cut elimination inside a deep inference system for classical predicate logic. *Studia Logica*, 82(1):51–71, 2006.

[6] Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):1–34, 2009. Article 14.

[7] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A quasipolynomial cut-elimination procedure in deep inference via atomic flows and threshold formulae. In *LPAR-16*, volume 6355 of *LNCS*, pages 136–153. Springer-Verlag, 2010.

[8] Alessio Guglielmi, Tom Gundersen, and Lutz Straßburger. Breaking paths in atomic flows for classical logic. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 284–293. IEEE Computer Society, 2010.

[9] David Hilbert and Paul Bernays. *Grundlagen der Mathematik*, volume 2. Springer Verlag, 2 edition, 1968.

[10] Dominic J. D. Hughes, Lutz Straßburger, and Jui-Hsuan Wu. Combinatorial proofs and decomposition theorems for first-order logic. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE, 2021.

[11] Emil Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 19(2):323–339, 2009.

[12] Georg Moser and Richard Zach. The epsilon calculus and herbrand complexity. *Studia Logica*, 82:133–155, 2006.

[13] Benjamin Ralph. *Modular Normalisation of Classical Proofs*. PhD thesis, University of Bath, 2019.

[14] Lutz Straßburger and Alessio Guglielmi. A system of interaction and structure IV: The exponentials and decomposition. *ACM Trans. Comput. Log.*, 12(4):23, 2011.