Abstract. In 1994 Gerard Huet formalized in Coq the cube property of \( \lambda \)-calculus residuals. His development is based on a clever idea, a beautiful inductive definition of residuals. However, in his formalization there is a lot of noise concerning the representation of terms with binders. We re-interpret his work in Abella, a recent proof assistant based on higher-order abstract syntax and provided with a nominal quantifier. By revisiting Huet’s approach and exploiting the features of Abella, we get a strikingly compact and natural development, which makes Huet’s idea really shine.

1 Introduction

The confluence or Church-Rosser theorem of \( \lambda \)-calculus has been formalized in several proof assistants, and it is probably the theorem with the highest number of formalized proofs \([26,31,21,32,27,30,17,15,16]\). In \([17]\) Huet formalizes in Coq also a deeper result, the cube property of \( \lambda \)-calculus residuals (due to Jean-Jacques Lévy \([20,4]\)). This paper presents a new, simple formalization of this result, developed in Abella \([9,8]\), a recent proof-assistant based on higher-order abstract syntax (HOAS) \([7,23]\) and provided with a nominal quantifier \([13,12,11,24,10,2]\).

Residual systems are a standard tool in rewriting \([20,18,19,15,34]\). In particular, they are at the basis of the advanced rewriting theory of \( \lambda \)-calculus and orthogonal rewriting systems (standardization, neededness, Lévy’s families and optimality, inside-out reductions, see \([20,18,19,34]\)). Roughly, one first introduces a mechanism to track redexes along reductions (typically using positions or underlinings), so that it is possible to say which are the residuals of a redex \( r \) after another redex. Then, one shows that any given span \( u_2 \leftarrow t \rightarrow u_1 \) can be closed by simply reducing on both sides the residuals of one redex after the other. The idea is that residuals refine confluence providing a minimal closure of confluence diagrams. The abstract theory of residual systems—which is independent from \( \lambda \)-calculus—is based on three axioms, the cube property plus two other minor axioms (see \([34]\), Chapter 8.7). The development in this paper essentially proves that \( \lambda \)-calculus admits a residual system, but we will not enter into the details of the abstract theory.
A delicate point is how to define residuals for a given calculus. In [17] Huet presents an elegant and compact solution for $\lambda$-calculus: he uses a simple ternary relation over terms with underlinings, defined by induction on the structure of the first term. However, Huet represents (marked) $\lambda$-terms using de Bruijn indexes, and a relevant part of his development deals with the properties of indexes, substitution and lifting.

Initially, we repeated Huet’s development to see how much Abella—being a proof assistant based on HOAS—could help in simplifying Huet’s work. All the troubles about indexes, lifting and substitution disappear, this was expected. However, along the way we realized that other simplifications were possible (the first two are independent from Abella):

1. **Marks**: Huet underlines applications, which requires to introduce a notion of well-formed term (regular terms in [17]) and to show that various operations preserve well-formedness. According to common practice we rather mark redexes (as in [3], for instance), so that any marked term is well-formed, and some lemmas disappear.

2. **Rewriting**: by analyzing inductions and dependencies between lemmas we simplify the statements and the number of lemmas required to prove the cube property. In particular, Huet recognizes the so-called prism property as more fundamental than the cube property, but we show that the direct proof of the cube property is not harder than the proof of the prism property. This also agrees with the clean abstract theory in [34] (which did not exist at the time of [17]), where there are examples of residual systems enjoying the cube property but not the prism property. Actually, we show that for $\lambda$-calculus it is possible to prove both properties with the same induction.

3. **Contexts**: in HOAS-based proof assistants $\alpha$-equivalence and substitution are primitive notions, but induction usually requires to consider predicates inside contexts of local assumptions (called worlds in Twelf [28], schemas in Beluga [29,6]) and prove properties about them. With respect to the untyped $\lambda$-calculus these contexts are artifacts, since they do not belong to the informal theory. The nominal quantifier $\nabla$ (nabra) of Abella, not available in other HOAS settings, allows to formalize the untyped $\lambda$-calculus circumventing the use of contexts.

The final result is quite striking: we formalize a property subsuming both the cube and prism properties using only two definitions and one auxiliary lemma. Moreover, the development follows exactly the informal, pen-and-paper reasoning: there is no need to care about indexes, $\alpha$-equivalence, substitution or contexts.

The next section contains an introduction to residuals and the way Huet represents them. In Section 3 we present the formal development, also explaining the representation of $\lambda$-terms. Section 4 discusses some variations over our development.

The sources of the development can be found on-line [1].