# Closed Inductive-Inductive Types are Reducible to Indexed Inductive Types

Ambrus Kaposi[1], András Kovács[1], and Ambroise Lafont[2]

[1] Eötvös Loránd University, Budapest, Hungary
{akaposi, kovacsandras}@inf.elte.hu
[2] IMT Atlantique, Inria, LS2N CNRS, France
ambroise.lafont@inria.fr

A mutual Inductive-inductive type [4, 2] (IITs) consists in a mutual definition of a type $\mathsf{Con} : \mathcal{U}$ and a type family $\mathsf{Ty} : \mathsf{Con} \to \mathcal{U}$ indexed over that type. Extended with quotients, this allows to formalize the syntax of a dependent type theory [1]. We show that IITs are reducible to indexed inductive types in a setting with uniqueness of identity proofs and function extensionality.

### Signatures and initial algebras for IITs

In [3], a syntax for signatures of quotient inductive-inductive types (QIITs) is given as a domain-specific type theory, where every typing context can be viewed as a listing of type, value and equality constructors. It is also shown that initial algebras for all QIITs can be constructed from terms of the syntax of signatures.

We can get inductive-inductive signatures by taking the equality-free fragment of QIIT signatures. We also restrict signatures so that they are closed (cannot refer to types external to a signature). In this case it also holds that all IITs are constructible from terms of the syntax of II signatures.

However, this syntax of II signatures is given as a QIIT, and thus at this stage we can only show that if we assume the existence of this particular QIIT, then all IITs exist. We strengthen this result by giving a construction of the syntax of II signatures from indexed inductive types, thereby showing that all IITs are in fact constructible from indexed inductive types.

### Constructing the syntax of II signatures

This can be viewed as an instance of the initiality problem popularized by Voevodsky: we have an intrinsically typed categorical notion of model for a particular type theory (the theory of II signatures), and we aim to construct the initial model, using inductively defined preterms and well-formedness relations.

However, the initiality proof in our case is simpler than in general, because the theory of II signatures does not contain $\beta$-rules, and thus it is possible to construct the initial model using only $\beta$-normal preterms, avoiding the use of quotients.

Hence, we first define normal preterms and typing relations on them, using indexed inductive types. Then, we use well-typed preterms to construct a model of the theory of II signatures. Lastly, we show that the constructed model is initial. We explain the last step in more detail.

### Initiality of the term model

Initiality means that we have a *recursion principle* for II signatures, and recursors are also unique. This is less convenient in practice than *induction*, but [3] shows unique recursion to be equivalent to induction, and in our case it is easier to consider the former.

To show initiality, we consider an arbitrary model for the theory of II signatures, and exhibit a unique morphism from the previously given term model to it. We do this in the following steps. Each step is preformed by induction on the presyntax.

1. We define a relation between the presyntax and the given model.
2. We show that the relation is right-unique.
3. We show that the relation is preserved by substitution.
4. We show that the relation is left-total on well-typed presyntax.
5. Since now the relation is shown to be functional, we can use it to build a model morphism.
6. We show that the model morphism is unique.

Streicher's previous construction [5] used a family of partial functions from the presyntax to a given model, which functions are later shown to be total on well-formed presyntax. In contrast, we use a *relation* which is later shown to be functional. For our use case, the relational approach seemed to be more convenient in a mechanized setting, and it could be worthwhile to try it in initiality proofs for other type theories as well.

**Formalization**

We formalized in Agda (`https://github.com/amblafont/UniversalII/blob/cwf-syntax/Cwf/`) the construction of the syntax of II signatures. Separately, there is an Agda formalization for [3] in `https://bitbucket.org/akaposi/finitaryqiit`, which includes the construction of IITs from the syntax of II signatures.

Note that the [3] formalization assumed definitional computation rules for induction on signatures, while our current construction of signatures only provides propositional computation rules. However, we already assume uniqueness of identity proofs (UIP) along with function extensionality, and also use limited equality reflection in the form of Agda rewrite rules. Hence, our formalization is largely in an extensional setting, and thus the propositional-definitional mismatch is not essential. Repeating these constructions without UIP and equality reflection is a potential line of future work. Another future research would be to extend the current result to open and infinitary signatures and QIITs.

# References


[1] Thorsten Altenkirch and Ambrus Kaposi. Type theory in type theory using quotient inductive types. In Rastislav Bodik and Rupak Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 18–29. ACM, 2016.

[2] Gabe Dijkstra. *Quotient inductive-inductive definitions.* PhD thesis, University of Nottingham, 2017.

[3] Ambrus Kaposi, András Kovács, and Thorsten Altenkirch. Constructing quotient inductive-inductive types. *Proceedings of the ACM on Programming Languages*, 3(POPL):2, 2019.

[4] Fredrik Nordvall Forsberg. *Inductive-inductive definitions.* PhD thesis, Swansea University, 2013.

[5] Thomas Streicher. *Semantics of type theory: correctness, completeness and independence results.* Springer Science & Business Media, 2012.