SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Constructing Linear Types in Isabelle/HOL

Felix Krayer

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Constructing Linear Types in Isabelle/HOL

Konstruktion linearer Typen in Isabelle/HOL

Author:	Felix Krayer
Examiner:	Florian Bruse
Supervisor:	Dmitriy Traytel, Tobias Nipkow
Submission Date:	13-11-2025

I confirm that this master’s thesis is my own work and I have documented all sources and material used.

Munich, 13-11-2025 Felix Krayer

Acknowledgments

Abstract

Contents

1 Introduction

- Datatypes in general
  - Datatypes in Isabelle/HOL are built on Bounded Natural Functors (BNFs) (defined in [TPB12])
  - Structure of the Thesis

2 Background

This Chapter serves to introduce BNFs and their generalization to Map-Restricted Bounded Natural Functors (MRBNFs). As described in Chapter 1, datatypes in Isabelle are implemented using BNFs, meaning that the type constructors of polymorphic types (types with type variables) can be described through the properties of a BNF.

Examples of one such types are α list and $\alpha\ \beta$ prod (infix notation $\alpha \times \beta$). These are unary and binary type constructors, meaning that they can be applied to other types to form a new type. Thus, type constructors are *functors*.

Furthermore, they have a well-defined set and map function. The exact properties are listed below. Thus they behave *natural*. Lastly, they are *bounded*, since the set of elements that the type can describe is bounded by a (possibly transfinite) cardinal.

$$\textit{map\_id}\colon\ \mathsf{map}_F\ id\ x = x \tag{2.1}$$

$$\textit{map\_comp}\colon\ \mathsf{map}_F\ g\ (\mathsf{map}_F\ f\ x) = \mathsf{map}_F\ (g \circ f)\ x \tag{2.2}$$

$$\textit{map\_cong}\colon\ (\forall z \in \mathsf{set}_F\ z.\ f\ z = g\ z) \Longrightarrow \mathsf{map}_F\ f\ x = \mathsf{map}_F\ g\ x \tag{2.3}$$

$$\textit{set\_map}\colon\ \mathsf{set}_F(\mathsf{map}_F\ f\ x) = f\,`\,\mathsf{set}_F\ x \tag{2.4}$$

$$\textit{infinite\_regular\_card\_order}\colon\ \mathsf{infinite\ bound}_F \wedge \mathsf{regular\ bound}_F \wedge \mathsf{cardinal\_order\ bound}_F \tag{2.5}$$

$$\textit{set\_bd}\colon\ |\mathsf{set}_F x| <_o \mathsf{bound}_F \tag{2.6}$$

$$\textit{rel\_compp\_leq}\colon\ \mathsf{rel}_F\ R\ \mathsf{OO}\ \mathsf{rel}_F\ Q\ = \mathsf{rel}_F\ (R\ \mathsf{OO}\ Q) \tag{2.7}$$

$$\textit{in\_rel}\colon\ \text{Weak Pullback Preservation WP} \tag{2.8}$$

Where \` is the image function on sets and OO is the composition of relations.

While most of these properties are straightforward, we want to explain the preservation of weak pullbacks in more detail.

$$\mathsf{rel}_F\ R\ x\ y = \exists \mathsf{z}.\ \mathsf{set}_F\ z \subseteq \{(a,b).\ R\ a\ b\} \wedge \mathsf{map}_F\ fst\ z = x \wedge \mathsf{map}_F\ snd\ z = y \tag{2.9}$$

The idea is that two elements x and y of the type $\alpha\ F$ are related through a relation R iff there exists a z that acts as a "zipped" version of x and y. The atoms of this z are the atoms of x and y, that are organized in pairs of R-related with the x as the first and y as the second position in the pair.

- Explain all the BNF theorems
- cite [Bla+19]

3 Linearizing MRBNFs

3.1 Linearization of MRBNFs (In theory)

In this section we define the linearization of an MRBNF on a subset of it's *live* variables. The result of the linearization is a new MRBNF with the same variable types (*live*, *dead*, *bound*, *free*), except for the linearized variables that change their type from *live* to *bound*. This means that the map function is now restricted to only allow bijective and small-support functions on these variables. Apart from this change, it is ensured that the MRBNF is *nonrepetitive* with respect to the linearized variables. We give a definition nonrepetitiveness in the following Subsection 3.1.1. Intuitively it means that the atoms of that type cannot occur multiple times in an element of the type.

3.1.1 Nonrepetitiveness

At the core of linearization lies the notion of *nonrepetitiveness*. We think of an element x of a type $\alpha\ F$ as being nonrepetitive if all its α atoms are distinct from another. We give an exact definition nonrep of nonrepetitiveness in relation to all other elements of $\alpha\ F$ with the *same shape*:

$$\textsf{same shape}_F\ x\ y = \textsf{rel}_F\ top\ x\ y \tag{3.1}$$

$$\textsf{nonrep}_F\ x = \forall y.\ \textsf{same shape}\ x\ y \longrightarrow (\exists f.\ y = \textsf{map}_F\ f\ x) \tag{3.2}$$

We consider two $\alpha\ F$ elements x and y to have the *same shape*, if they are related with the *top* relation, i.e., the relation that relates all α atoms to all others. This is true, if the relator $\textsf{rel}_F$ can relate the elements. In the case of list, this is the case when two lists have the same length but possibly different content.

Based on this, x is a nonrepetitive element, if for all other elements y with the same shape, a function exists through which x can be mapped to y. In our example of list, this holds for all lists with distinct elements (given a second list, one can easily define a function mapping the distinct elements of x to that list). It does not hold for lists with repeating elements, because no f exists that could map two same elements at different positions in this list to distinct elements in an arbitrary second list.

For MRBNFs with more than one live variable, we can give a definitions of *nonrepetitiveness* and having the *same shape* on a subset of the live variables. In that case, we consider x and y of type $(\alpha, \beta)\ G$ to have the same shape with respect to α, iff they are *equal* in their β atoms and are related with *top* in α as before. Consequently for the map in the nonrep definition, the *id* function is applied to the β atoms, since they are already required to be equal.

$$\mathsf{same\ shape}^1_G\ x\ y = \mathsf{rel}_G\ top\ (=)\ x\ y \tag{3.3}$$

$$\mathsf{nonrep}^1_G\ x = \forall y.\ \mathsf{same\ shape}\ x\ y \Longrightarrow (\exists f.\ y = \mathsf{map}_G\ f\ id\ x) \tag{3.4}$$

3.2 Required properties

A MRBNFs has to fulfill two properties to be linearized. First, to ensure that the resulting type constructor is non-empty, it is required, that there exists a nonrepetitive element (with respect to the linearized variables): $\exists x.\ \mathsf{nonrep}\ x$

Furthermore, even though MRBNFs are already required to perserve weak pullbacks as defined in Equation 2.9, for the linearization it is required that they perserve all pullbacks (P). Formalized this means that the existance of z in the equation has to be fulfilled uniquely, i.e., for each R-related x and y there existes *exactly one* z fulfilling the property in Equation 2.9. For example the full pullback preservation (P) is fulfilled by the α list and $\alpha\ \beta$ prod functors but not by α set.

3.3 Intermediate lemmas

F strong From the pullback preservation with uniqueness we can prove the following lemma. In fact this notion of strongness is equivalent to pullback preservation:

$$[\![\mathsf{rel}_F\ R\ x\ y;\ \mathsf{rel}_F\ Q\ x\ y]\!] \Longrightarrow \mathsf{rel}_F\ (inf\ R\ Q)\ x\ y$$

where the infimum inf of two relations R and Q relates all elements that are related by both R and Q.

rel exchange TODO: only interesting for functors with more than one variable

$$[\![\mathsf{rel}_G\ R_1\ R_2\ x\ y;\ \mathsf{rel}_G\ Q_1\ Q_2\ x\ y]\!] \Longrightarrow \mathsf{rel}_G\ R_1\ Q_2\ x\ y$$

map peresrving nonrepetitiveness

$$[\![\mathsf{nonrep}^1_G\ x;\ \mathsf{bijective} f]\!] \Longrightarrow \mathsf{nonrep}^1_G\ (\mathsf{map}_G\ f\ g\ x)$$

3.4 Proving the MRBNF axioms

3.5 Linearization of MRBNFs (In Isabelle)

We implement a command that allows the user to linearize an existing MRBNF or BNF on one or multiple of it's live variables: The syntax of the command is given in the following:

```
linearize_mrbnf ('a :: var, 'b) lin_type = ('a :: var, 'b) type on 'a
```

Abbreviations

BNF Bounded Natural Functor

MRBNF Map-Restricted Bounded Natural Functor

List of Figures

List of Tables

Bibliography

[Bla+19] J. C. Blanchette, L. Gheri, A. Popescu, and D. Traytel. “Bindings as bounded natural functors.” In: *Proceedings of the ACM on Programming Languages* 3.POPL (2019), pp. 1–34.

[TPB12] D. Traytel, A. Popescu, and J. C. Blanchette. “Foundational, compositional (co) datatypes for higher-order logic: Category theory applied to theorem proving.” In: *2012 27th Annual IEEE Symposium on Logic in Computer Science*. IEEE. 2012, pp. 596–605.