Quasi Morphisms for Almost Full Relations

Dominique Larchey-Wendling

May 7, 2024

Quasi Morphisms for Almost Full Relations

Dominique Larchey-Wendling

Université de Lorraine, CNRS, LORIA, France
dominique.larchey-wendling@loria.fr

Abstract

In Coq, we mechanize two morphisms for transferring the almost full property between relations.

1 Introduction

The study of almost full relations [11] (constructive well quasi orders) mainly consists in establishing closure properties of the `af` predicate (see Fig. 1). For instance, Higman's lemma [3, 1, 9] states its closure under the homeomorphic embedding of lists, and Kruskal's theorem [4, 10], closure under the homeomorphic embedding of rose trees. Our former Coq constructive proof of Kruskal's tree theorem [5] suffers from being quite monolithic, a property unfortunately inherited from Veldman's [10] pen&paper proof of which it derives. In the process of a major refactoring effort aimed at modularity, removal of code duplication, and readability, we have identified two important tools to transfer `af` from one relation R to another T, i.e. to establish entailments of shape $\mathtt{af}\,R \to \mathtt{af}\,T$.

We present these tools independently of the context of intricate developments. The first one is simple but versatile: it is sufficient to provide a surjective relational morphism from R to T. The second one, more specialized, but instrumental in the constructive proofs of Higman/Kruskal's results [1, 10], aims at transfers of shape $\mathtt{af}\,R \to \mathtt{af}\,T{\uparrow}y_0$; see Fig. 1 for $(\cdot{\uparrow}\cdot)$. In that case, it is sufficient to provide a quasi morphism to enable the transfer. When assuming decidability of relations as in [9], a quasi morphism can be turned into a surjective relational morphism, allowing for a short proof of transfer. In the general case, the transfer is much more involved. The two bricks that compose this tool, the FAN theorem and a combinatorial principle, can be traced back to [1], and are repeatedly inlined in [10]. However, the quasi morphism result is never stated in a general setting to be established independently, hence this abstract.

We only present the main results and the ingredients to obtain them, sticking to an informal presentation, without giving justifications. Strict preciseness is deferred to the available Coq artifact [7] that is both standalone, compact with less than 1k loc, commented and designed for human readability.[1] See also [6, 8] for a presentation on how these results are used e.g. to establish Higman's/Veldman's results.

$$\frac{R\,x\,y}{R{\uparrow}a\,x\,y} \qquad \frac{R\,a\,x}{R{\uparrow}a\,x\,y} \qquad\Big|\qquad \frac{\forall x y, R\,x\,y}{\mathtt{af}\,R} \qquad \frac{\forall a, \mathtt{af}\,R{\uparrow}a}{\mathtt{af}\,R}$$

$$\frac{}{\mathtt{Forall}_2\,R\,[]\,[]} \qquad \frac{R\,x\,y \qquad \mathtt{Forall}_2\,R\,l\,m}{\mathtt{Forall}_2\,R\,(x::l)\,(y::m)}$$

$$\frac{R\,y\,x \qquad y \in l}{\mathtt{good}\,R\,(x::l)} \qquad \frac{\mathtt{good}\,R\,l}{\mathtt{good}\,R\,(x::l)} \qquad\Big|\qquad \frac{P\,l}{\mathtt{bar}\,P\,l} \qquad \frac{\forall x, \mathtt{bar}\,P\,(x::l)}{\mathtt{bar}\,P\,l}$$

Figure 1: Inductive rules for $(\cdot{\uparrow}\cdot)$, `af`, $\mathtt{Forall}_2$, `good` and `bar`, with $R : \mathtt{rel}_2\,\_\,\_$ and $P : \mathtt{rel}_1\,(\mathtt{list}\,\_)$.

[1] In this abstract, the results are `Prop`-bounded but the artifact itself is generic in `Prop`-bounded vs `Type`-bounded alternatives.

2 Surjective relational morphisms

Below we write $\mathbb{P}$ for `Prop` and we use $\texttt{rel}_1\,X \coloneqq X \to \mathbb{P}$ (resp. $\texttt{rel}_2\,X \coloneqq X \to X \to \mathbb{P}$) to represent unary (resp. binary) relations, denoting $\subseteq$ for relations inclusion. For $R : \texttt{rel}_2\,X$ and $P : \texttt{rel}_1\,X$, we write $R{\Downarrow}P : \texttt{rel}_2\,\{x \mid P\,x\}$ for the restriction of R to the subtype. We adopt the usual notations for lists: $[]$ for the empty list, :: for the cons(tructor), and $\in$ for list membership. The product embedding for lists is defined inductively as $\texttt{Forall}_2\,R : \texttt{list}\,X \to \texttt{list}\,Y \to \mathbb{P}$ by the two rules of Fig. 1.

Following [11], a binary relation $R : \texttt{rel}_2\,X$ is *almost-full* (AF) if it satisfies the predicate $\texttt{af}\,R : \mathbb{P}$ defined inductively by the two rules of Fig. 1. There, we define the *lifted relation* $R{\uparrow}a$ by $(R{\uparrow}a)\ x\ y \coloneqq R\ x\ y \vee R\ a\ x$, and we extend lifting to lists by $R{\uparrow\uparrow}[a_1;\ldots;a_n] \coloneqq R{\uparrow}a_n \ldots {\uparrow}a_1$. Intuitively, R is AF if it is bound to become a full relation, whatever sequence of liftings is applied to it. An alternative formulation uses the inductive `bar` predicate and $\texttt{good}\,R$ sequences/lists as defined in Fig. 1. For any $l : \texttt{list}\,X$, we establish the equivalence $\texttt{af}\,(R{\uparrow\uparrow}l) \leftrightarrow \texttt{bar}\,(\texttt{good}\,R)\ l$, and in particular we get $\texttt{af}\,R \leftrightarrow \texttt{bar}\,(\texttt{good}\,R)\ []$. This result allows for an easy application of the FAN theorem (see below).

Already in [11], monotonicity is present as a tool to transfer `af` from one relation to another, i.e. $R \subseteq T \to \texttt{af}\,R \to \texttt{af}\,T$, but R and T must share the same carrier type. Also mentioned in [11], one can transport `af` using a map $f : X \to Y$ with $\texttt{af\_comap} : \texttt{af}\,R \to \texttt{af}\,\big(\lambda\,x_1\,x_2, R\,(f\,x_1)\,(f\,x_2)\big)$, but this tool is quite cumbersome to use as the target `af` relation has to be put first in this restrictive shape.[2]

Instead, we introduce the notion of *surjective relational morphism* to transport `af` from $R : \texttt{rel}_2\,X$ to $T : \texttt{rel}_2\,Y$. This is a *relational* map $f : X \to Y \to \mathbb{P}$ with the two following properties:

1. $\forall y, \exists x, f\ x\ y$ (surjective); 2. $\forall x_1\,x_2\,y_1\,y_2, f\ x_1\ y_1 \to f\ x_2\ y_2 \to R\ x_1\ x_2 \to T\ y_1\ y_2$ (morphism).

Under these assumptions we establish $\texttt{af}\,R \to \texttt{af}\,T$. This formulation is more versatile: a) there is no constraint on the shape of the target T, b) it does not restrict morphisms to total functions, hence they can be *partial*, c) but also critically, they can map to *several outputs*. For instance, the entailment $\texttt{af}\,R \to \texttt{af}\,R{\Downarrow}P$ is trivial to establish using such a morphism. But without some strong hypotheses, like e.g. P is Boolean, there is no surjective functional map *onto* the carrier type $\{x \mid P\,x\}$ of $R{\Downarrow}P$.

We use relational morphisms extensively in this development, e.g. for short proofs of the transfer $\texttt{af}\,R{\uparrow}a \to \texttt{af}\,R{\Downarrow}(\neg R\,a)$ and the converse $\texttt{af}\,R{\Downarrow}(\neg R\,a) \to \texttt{af}\,R{\uparrow}a$. But the later requires the *decidability* of $(R\,a)$ as an additional hypothesis. Notice that using negations like in $\neg R\,a$ (as done in [9]) allows for equivalences between $\texttt{af}\,R$ and (inductive) well-foundedness of sequences/lists expansion restricted to bad R-sequences, but be aware that this approach usually restricts the study to decidable relations.

3 Quasi morphisms

We switch to the central transfer tool used in our inductive mechanizations of Higman's [6] and Veldman's [8] results, the notion of *quasi morphism*. It allows to establish the entailment $\texttt{af}\,R \to \texttt{af}\,T{\uparrow}y_0$ for $R : \texttt{rel}_2\,X$, $T : \texttt{rel}_2\,Y$ and $y_0 : Y$. For this, one needs the following data: a map $ev : X \to Y$ from analyses to evaluations and a predicate $E : \texttt{rel}_1\,X$ characterizing *exceptional* analyses satisfying:[3]

1. $\forall y, \texttt{fin}\,(ev^{-1}\,y)$; 2. $\forall x_1\,x_2, R\,x_1\ x_2 \to T\ (ev\ x_1)\ (ev\ x_2) \vee E\,x_1$; 3. $\forall y, (ev^{-1}\,y) \subseteq E \to T\ y_0\ y$;

where we denote $ev^{-1}y \coloneqq (\lambda x, ev\,x = y)$ and call them *analyses* of (the evaluation) y. They are assumed finitely many by Item 1; Item 2 states that ev is a morphism unless applied to exceptional analyses; and Item 3 states that y embeds y_0 when all its analyses are exceptional. One can "quickly" justify quasi

[2] Coquand's constructive version of Ramsey's theorem $\texttt{af}\,R \to \texttt{af}\,T \to \texttt{af}\,(R \cap T)$ is their main focus, but we do not need it.

[3] The analysis/evaluation terminology follows [10, page 241], and an exceptional analysis "contains a disappointing sub-tree."

morphisms by further assuming the decidability of both $T\,y_0$ and E. Indeed, in that case ev becomes a surjective relational morphism from $R{\Downarrow}(\neg E)$ to $T{\Downarrow}(\neg T\,y_0)$. Yet the statement of the quasi morphism result carefully avoids negation, and we establish it without those decidability assumptions. Nonetheless in that general case, the proof uses two non-trivial tools (also mechanized in the artifact), related to the choice sequences for $ll : \texttt{list}\ (\texttt{list}\ X)$, i.e. the inhabitants of $\texttt{FAN}\ ll \coloneqq \lambda\, c, \texttt{Forall}_2\ (\cdot \in \cdot)\ c\ ll$:[4]

- the FAN theorem for inductive bars: for $P : \texttt{rel}_1\ (\texttt{list}\ X)$ *monotone*, i.e. $\forall x\, l, P\, l \to P(x :: l)$, we have $\texttt{bar}\ P\ [] \to \texttt{bar}\ (\lambda\, ll, \texttt{FAN}\ ll \subseteq P)\ []$;[5]
- a finite combinatorial principle: for $P : \texttt{rel}_1\ (\texttt{list}\ X)$, $B : \texttt{rel}_1\ X$, and $ll : \texttt{list}\ (\texttt{list}\ X)$, assuming $\forall c, \texttt{FAN}\ ll\ c \to P\, c \vee \exists x, x \in c \wedge B\, x$ (any choice sequence satisfies P or meets B), we have either $\exists c, \texttt{FAN}\, ll\, c \wedge P\, c$ (P contains a choice sequence), or $\exists l, l \in ll \wedge \forall x, x \in l \to B\, x$ (there is a list in ll which is included in B).[6]

References

[1] Daniel Fridlender. Higman's lemma in type theory. In Eduardo Giménez and Christine Paulin-Mohring, editors, *Types for Proofs and Programs*, pages 112–133, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[2] Daniel Fridlender. An Interpretation of the Fan Theorem in Type Theory. In Thorsten Altenkirch, Bernhard Reus, and Wolfgang Naraschewski, editors, *Types for Proofs and Programs*, pages 93–105, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[3] Graham Higman. Ordering by Divisibility in Abstract Algebras. *Proceedings of the London Mathematical Society*, s3-2(1):326–336, 1952.

[4] Joseph B. Kruskal. Well-Quasi-Ordering, The Tree Theorem, and Vazsonyi's Conjecture. *Transactions of the American Mathematical Society*, 95(2):210–225, 1960.

[5] Dominique Larchey-Wendling. A mechanized inductive proof of Kruskal's tree theorem. https://members.loria.fr/DLarchey/files/Kruskal, 2015.

[6] Dominique Larchey-Wendling. Higman's lemma in the Almost Full library. https://github.com/DmxLarchey/Kruskal-Higman, 2024.

[7] Dominique Larchey-Wendling. Quasi Morphisms for Almost Full relations (artifact). https://github.com/DmxLarchey/Quasi-Morphisms, 2024.

[8] Dominique Larchey-Wendling. Veldman's theorem in the Almost Full library. https://github.com/DmxLarchey/Kruskal-Veldman, 2024.

[9] Helmut Schwichtenberg, Monika Seisenberger, and Franziskus Wiesnet. *Higman's Lemma and Its Computational Content*, pages 353–375. Springer International Publishing, Cham, 2016.

[10] Wim Veldman. An intuitionistic proof of Kruskal's theorem. *Archive for Mathematical Logic*, 43(2):215–264, Feb 2004.

[11] Dimitrios Vytiniotis, Thierry Coquand, and David Wahlstedt. Stop When You Are Almost-Full. In Lennart Beringer and Amy Felty, editors, *Interactive Theorem Proving*, pages 250–265, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[4] Intuitively, $\texttt{FAN}\ [l_1;\ldots;l_n]$ spans the (finitely many) lists $[c_1;\ldots;c_n]$ such that $c_1 \in l_1, \ldots, c_n \in l_n$.

[5] Compared to [1, 2], this FAN theorem has a shorter proof because it avoids the explicit construction of the FAN as a list.

[6] Classically (with excluded middle and choice), the combinatorial principle is trivial and not limited to finite fans.