

# Join inverse categories and reversible recursion

Robin Kaarsgaard

DIKU, Department of Computer Science, University of Copenhagen  
Copenhagen, Denmark  
robin@di.ku.dk

Many reversible functional programming languages (such as Theseus [8] and the combinator calculi  $\Pi$  and  $\Pi^0$  [3]) as well as categorical models thereof (such as  $\dagger$ -traced symmetric monoidal categories [3]) come equipped with a tacit assumption of totality, a property that is neither required [2] nor necessarily desirable as far as guaranteeing reversibility is concerned. Shedding ourselves of this assumption, however, requires us to handle partiality explicitly as additional categorical structure.

One approach which does precisely that is Cockett & Lack’s notion of *inverse categories* [4], a specialization of *restriction categories*, which have recently been suggested and developed by Giles [5] as models of reversible (functional) programming. In this paper, we will argue that assuming ever slightly more structure on these inverse categories, namely the presence of *countable joins* of parallel morphisms, gives rise to a number of additional properties useful for modelling reversible functional programming, notably reversible (tail) recursion and recursive data types (via  $\omega$ -algebraic compactness with respect to structure-preserving functors), which are not otherwise present in general. This is done by adopting two different, but complementary, views on inverse categories with countable joins as enriched categories – as CPO-categories, and as (specifically  $\Sigma$ Mon-enriched) unique decomposition categories.

**Background** In the framework of restriction categories, partiality is handled by equipping each morphism  $f : A \rightarrow B$  with a partial identity morphism  $\bar{f} : A \rightarrow A$  (the *restriction idempotent* of  $f$ , intuitively the identity defined precisely where  $f$  is defined) subject to a few axioms, notably that  $\bar{f}$  is the right-identity of  $f$  under composition. This definition provides a partial ordering on Hom sets by defining  $f \leq g$  for parallel morphisms  $f$  and  $g$  iff  $g \circ \bar{f} = f$ .

Joins on morphisms (see, e.g., Guo [6]) are then defined to be joins with respect to this partial order (subject to a few axioms), with the caveat that parallel morphisms  $f$  and  $g$  can only be joined if they are *join compatible*, which they are iff  $g \circ \bar{f} = f \circ \bar{g}$  (intuitively, if they agree on all points in their domain where they are both defined). This definition is then straightforwardly extended to sets (in this particular case, countable ones) of parallel morphisms by saying that a set  $S \subseteq \text{Hom}(A, B)$  is join compatible if all morphisms of  $S$  are pairwise join compatible. A restriction category is thus said to have (countable) joins if all (countable) join compatible sets have a join, and the category has a *restriction zero object*, that is, a zero object in the usual sense which additionally satisfies that the zero map  $0_{A,A} : A \rightarrow A$  is a restriction idempotent (i.e., that  $0_{A,A} = \overline{0_{A,A}}$ ) for all objects  $A$  (the zero map  $0_{A,B}$  is the unit for joins in  $\text{Hom}(A, B)$ ).

Perhaps more immediately important to our applications, restriction categories allow for a definition of a *partial isomorphism* as a morphism  $f : A \rightarrow B$  for which there exists a *partial inverse*  $f^* : B \rightarrow A$  such that  $f^* \circ f = \bar{f}$  and  $f \circ f^* = \overline{f^*}$ . An *inverse category* is then defined to be a restriction category in which all morphisms are partial isomorphisms; as such, inverse categories are “groupoids with partiality,” and can be canonically equipped with the structure of a  $\dagger$ -category by letting the  $\dagger$ -functor map each morphism to its partial inverse. Keeping with this canonical structure, we will use  $f^\dagger$  for the partial inverse of  $f$  from here on out. Inverse categories can be equipped with joins in the same way as general restriction categories can (with slightly more work, see Guo [6]).

**CPO-enrichment** Since inverse categories come equipped with partially ordered Hom sets, demonstrating CPO-enrichment reduces to producing suprema of  $\omega$ -chains and showing that composition is continuous and strict. Let  $\mathcal{C}$  be an inverse category with countable joins. For an  $\omega$ -chain  $\{f_i\}_{i \in \omega}$  of some  $\text{Hom}_{\mathcal{C}}(A, B)$ , we define its supremum by<sup>1</sup>

$$\sup \{f_i\}_{i \in \omega} = \bigvee_{i \in \omega} f_i .$$

That this join exists follows from the fact that  $f \leq g$  implies that  $f$  and  $g$  are join compatible. That composition is continuous follows directly by this definition since

$$g \circ \bigvee_{f \in F} f = \bigvee_{f \in F} (g \circ f) \quad \text{and} \quad \left( \bigvee_{f \in F} f \right) \circ h = \bigvee_{f \in F} (f \circ h)$$

are axioms of joins [6]; similarly, strictness of composition follows by the universal mapping property for the zero object, noting that the zero map  $0_{A,B}$  is least in the partial order on  $\text{Hom}_{\mathcal{C}}(A, B)$  for all objects  $A, B$ .

From this follows the existence of fixed points for all continuous *morphism schemes for recursion*, i.e., monotone and continuous functions of the form  $f : \text{Hom}_{\mathcal{C}}(A, B) \rightarrow \text{Hom}_{\mathcal{C}}(A, B)$  by Kleene's fixed point theorem, and can thus be used to model recursion. A further pleasant property is *locally continuity* of the canonical  $\dagger$ -functor on  $\mathcal{C}$ , i.e., the map  $\text{inv}_{A,B} : \text{Hom}_{\mathcal{C}}(A, B) \rightarrow \text{Hom}_{\mathcal{C}}(B, A)$  given by  $\text{inv}_{A,B}(f) = f^\dagger$  is monotone and continuous for all objects  $A, B$ .

Combining the two, we can show that each continuous morphism scheme for recursion  $f$  has a *fixed-point adjoint*  $f_{\ddagger}$  such that  $(\text{fix } f)^\dagger = \text{fix } f_{\ddagger}$ ; intuitively, that the partial inverse of a recursive function can be constructed recursively in a canonical way. This is done by defining

$$f_{\ddagger} = \text{inv}_{A,B} \circ f \circ \text{inv}_{B,A}$$

which is continuous since it is a continuous composition of continuous functions;  $\text{fix } f_{\ddagger} = (\text{fix } f)^\dagger$  can then be shown using local continuity of the  $\dagger$ -functor, and by noting that  $f_{\ddagger}^n = \text{inv}_{A,B} \circ f^n \circ \text{inv}_{B,A}$ . This gives us reversible recursion in the style of RFUN [9]: a recursive function is inverted by replacing recursive calls with calls to the inverse function, and then inverting the remainder of the function. Further, by considering more general morphism schemes, we can get a procedure for representing *parameterized functions* in the style of Theseus [8].

Another consequence is the fact that every inverse category with countable joins can be embedded (fully faithfully and in a join and restriction preserving manner) into an inverse category that is  $\omega$ -algebraically compact with respect to the class of join and restriction preserving functors. The proof of this theorem is somewhat involved: it relies on a coincidence between *restriction monics* in inverse categories (split monics that split a restriction idempotent) and *embeddings* in CPO-categories (morphisms  $f : A \rightarrow B$  with *projections*  $f^* : B \rightarrow A$  such that  $f^* \circ f = 1_A$  and  $f \circ f^* \leq 1_B$ ); on Guo's characterization of join restriction categories as partial map categories with certain stable colimits [6]; and on Adámek's fixed point theorem [1].

**$\Sigma$ Mon-enrichment and unique decomposition** Another way to approach inverse categories with countable joins is as  $\Sigma$ -monoid enriched categories in the sense of Haghverdi [7]. Briefly, a  $\Sigma$ -monoid consists of a set  $S$  equipped with a partial sum function  $\Sigma$  defined on

<sup>1</sup>This is a slight abuse of notation since joins in restriction categories are unordered, i.e., defined on sets rather than families. The join  $\bigvee_{i \in \omega} f_i$  should thus be taken to mean  $\bigvee_{f \in F} f$  where  $F = \{f_i \mid i \in \omega\}$ .

countable families of  $S$  (say that such a family is *summable* if its sum exists), subject to the axioms of *partition-associativity* (a family is summable iff any partitioning of it is piecewise summable, and the sum of the pieces coincide with the sum of the family) and *unary sum* (the sum of a singleton family is equal to its element). It is straightforwardly the countable joins in inverse categories satisfies these axioms, with summability coinciding with join compatibility.

If we, in addition, suppose that our inverse category  $\mathcal{C}$  with countable joins is equipped with a *disjoint sum tensor* in the sense of Giles [5] (a symmetric monoidal restriction functor  $\cdot \oplus \cdot$  with the restriction zero as unit, and equipped with jointly epic injections  $\Pi_1 : A \rightarrow A \oplus B$  and  $\Pi_2 : B \rightarrow A \oplus B$  and jointly monic coinjections  $\Pi_1^\dagger : A \oplus B \rightarrow A$  and  $\Pi_2^\dagger : A \oplus B \rightarrow B$ ), we get straightforwardly that  $\mathcal{C}$  is a *unique decomposition category* (a symmetric monoidal category with *quasi-injections*  $\iota_j : X_j \rightarrow \bigoplus_I X_i$  and *quasi-projections*  $\rho_j : \bigoplus_I X_i \rightarrow X_j$  for all  $j \in I$  where  $I$  a finite index set, subject to a two axioms [7]). Using join compatibility of disjoint morphisms, it follows by Haghverdi [7] that  $\mathcal{C}$  is traced, and that the trace can be constructed by

$$\text{Tr}_{A,B}^U(f) = f_{11} + \sum_{n \in \mathbb{N}} f_{21} \circ f_{22}^n \circ f_{12} = f_{11} \vee \bigvee_{n \in \mathbb{N}} f_{21} \circ f_{22}^n \circ f_{12}$$

for all  $f : A \oplus U \rightarrow B \oplus U$ , where  $f_{ij} = \rho_j \circ f \circ \iota_i = \Pi_j^\dagger \circ f \circ \Pi_i$ . In this special case, however, this is not just a trace, but a  $\dagger$ -trace (*i.e.*, it satisfies  $\text{Tr}_{B,A}^U(f^\dagger) = \text{Tr}_{A,B}^U(f)^\dagger$ ): this can be seen by realizing that  $(f_{ij})^\dagger = f_{ji}^\dagger$ , and by using  $(\bigvee_{f \in F} f)^\dagger = \bigvee_{f \in F} f^\dagger$  which follows directly from local continuity of the  $\dagger$ -functor in the CPO-view. This is significant given that  $\dagger$ -traces are used to model reversible tail recursion.

**Conclusion** The existence of countable joins in inverse categories provides us with a model of partial reversible functional programming with recursive types and general recursion in the style of RFUN. Further assuming the existence of a disjoint sum tensor allows us to extend the standard model of  $\dagger$ -traced symmetric monoidal categories to one with a notion of partiality.

## References

- [1] Jiří Adámek. Recursive Data Types in Algebraically  $\omega$ -Complete Categories. *Information and Computation*, 118:181–190, 1995.
- [2] Holger Bock Axelsen and Robert Glück. What do reversible programs compute? In *Foundations of Software Science and Computational Structures*, volume 6604, pages 42–56, 2011.
- [3] William J. Bowman, Roshan P. James, and Amr Sabry. Dagger traced symmetric monoidal categories and reversible programming. In *Reversible Computation*, volume 7165 of *LNCS*, pages 51–56, 2011.
- [4] J. R. B. Cockett and Stephen Lack. Restriction categories I: Categories of partial maps. *Theoretical Computer Science*, 270(2002):223–259, 2002.
- [5] Brett Gordon Giles. *An investigation of some theoretical aspects of reversible computing*. PhD thesis, University of Calgary, 2014.
- [6] Xiuzhan Guo. *Products, Joins, Meets, and Ranges in Restriction Categories*. PhD thesis, University of Calgary, 2012.
- [7] Esfandiar Haghverdi. Unique decomposition categories, Geometry of Interaction and combinatory logic. *Mathematical Structures in Computer Science*, 10(2):205–230, 2000.
- [8] Roshan P. James and Amr Sabry. Theseus: A High Level Language for Reversible Computing. Available at <http://www.cs.indiana.edu/~sabry/papers/theseus.pdf>, 2014.
- [9] Tetsuo Yokoyama, Holger Bock Axelsen, and Robert Glück. Towards a reversible functional language. In *Reversible Computation*, volume 7165 of *LNCS*, pages 14–29, 2012.