Towards Corecursion Without Corecursion in Coq

Vlad Rusu, David Nowak

Towards Corecursion Without Corecursion in Coq

Vlad Rusu
Inria, Lille, France
Vlad.Rusu@inria.fr

David Nowak
CNRS, Lille, France
David.Nowak@univ-lille.fr

Coinduction is an important concept in functional programming. To formally prove properties of corecursive functions one can try to define them in a proof assistant such as Coq [1]. But there are limitations on the functions that can be defined. In particular, corecursive calls must occur directly under a call to a constructor, without any calls to other recursive functions in between. In this paper we show how a partially ordered set endowed with a notion of approximation can be organized as a Complete Partial Order. This makes it possible to define corecursive functions without using Coq's corecursion, as the unique solution of a fixpoint equation, thereby escaping Coq's builtin limitations.

1 Introduction

Coinduction is an important concept in functional programming. Coinductive types are the types of possibly infinite data structures such as the lists of Haskell, and corecursive functions compute values in coinductive types. An example is the Sieve of Eratosthenes that produces the infinite stream of primes.

To formally define corecursive functions one can try to use a proof assistant such as Coq. But then one quickly stumbles on the limitations of Coq on this matter. Indeed the language of this proof assistant is total, meaning that a function must be defined on all its domain (which is not requested for functional programming languages in general). Totality is ensured in Coq by a syntactical check of so-called guardedness. This criterion consists in checking that any corecursive call occurs directly under a call to a constructor of the coinductive type that constitutes the codomain (a.k.a. range) of the function under definition [5]. For example, Rose trees are coinductively defined as the constructor *tree* applied to a *forest*, which is an inductively-defined list of Rose trees. A *mirror* function would be corecursively defined by *mirror t* = *tree*(*map mirror*(*reverse*(*forest t*))), but this is rejected by Coq because the corecursive call is not directly under the constructor *tree*, but under *map*, a recursive function defined on lists.

In a previous paper [7] we propose a method for defining such functions by replacing the syntactical guardedness criterion by a semantical proof obligation of productiveness: for each input, an arbitrarily close approximation of the corresponding output is eventually produced. When a functional (i.e., a higher-order function, which constitutes the "blueprint" of a corecursive function under definition) is monotonic and satisfies the productiveness requirement, a corecursive function can be defined as the unique fixpoint of the functional in question. This technique requires that the codomain of the function under definition be organized as a Complete Partial Order (CPO). For some simpler coinductive types such as streams or possibly infinite lists, the CPO can be directly defined using Coq's builtin mechanisms. But for mixed coinductive-inductive types such as Rose trees these mechanisms no longer work, and [7] replaces the native coinductive types of Coq with inductive types, completed with equivalence classes of "ascending" sequences of elements in inductive types, which are the limits of the sequences in question.

However, the construction we ended up with in [7] imposes some not-so-practical conditions. The order had to be *weakly total*, thus excluding the natural prefix order of Rose trees. Although we were able to exhibit a weakly total order, the constructor *tree* is not monotonic w.r.t. this order. As a consequence, functionals that employ this constructor are not monotonic either, which excludes them for being used in

fixpoint equations for the definition of corecursive functions. For example, the *mirror* function cannot be defined in the natural manner, as a fixpoint of the functional $Mirror = \lambda f.\, \lambda t.\, tree(map\ f\ (reverse(forest\ t)))$, but in an indirect way that makes it harder to prove that the obtained function is indeed the intended one.

Contributions. In this paper we replace the weak totality condition on orders by a notion of approximation meeting some quite natural conditions. These conditions are easier to satisfy than the (comparatively arbitrary) weak totality. We illustrate with the example of finite trees that the natural (i.e. structural) order on inductive types — quite importantly, this is the least constrained order under which constructors are monotonic — does satisfy the conditions on approximations whereas it did not satisfy weak totality. We then describe the construction of a CPO under the said conditions. This construction is substantially more involved than that in [7], but has to be done only once and can then be instantiated on any inductive type with an adequate notion of approximation in order to generate coinductive types. For Rose trees, the effect of the new construction is that the *tree* constructor is now monotonic w.r.t. their natural *prefix* order. It is now possible to define mixed recursive-corecursive functions in the most natural manner, as the unique solution of the fixpoint equation induced by a monotonic, productive functional of the function.

Outline. Section 2 presents preliminary notions, chief among which an abstract notion of approximations and a number of assumptions over them. We show that the assumptions have a model - finite trees with their natural prefix order and a "cut" function. Some technical developments are also included.

Section 3 defines the completion operation that produces a CPO from a a partially ordered set with a least element. Completion is done in several steps: first, one completes an initial set of "finite" elements with new elements that are equivalence classes of ascending sequences of finite elements, thereby providing such sequences with "limits" (least upper bounds) required by CPOs. Second, the order and approximations are extended to the new elements. Third, we show using a "diagonalization" technique that ascending sequences of both old and new elements have limits among the new elements, meaning that the construction of the CPO does not require further completion steps.

Section 4 applies completion to produce the mixed inductive-coinductive type of Rose trees from finite trees. The roadmap for defining the *mirror* recursive-corecursive function as the unique fixpoint of its functional is presented. Section 5 concludes and presents related and future work. This paper will be formalized in Coq, but we keep it language-agnostic and use standard mathematics for better readability.

2 Preliminaries

2.1 Sequences and Complete Partial Orders

Definition 1 *Consider a set C and a partial order $\preceq$ on C. We denote by $\prec$ the relation defined by $t \prec t'$ iff $t \preceq t'$ and $t \neq t'$. A sequence $(s_i)_{i\in\mathbb{N}}$ of elements of C is*

- increasing *whenever for all $i \in \mathbb{N}$, $s_i \preceq s_{i+1}$;*
- strictly increasing*, whenever for all $i \in \mathbb{N}$, $s_i \prec s_{i+1}$;*
- stabilizing to $c \in C$ *whenever there exist $m \in \mathbb{N}$ such that for all $i \geq m$, $s_i = c$, and* stabilizing *whenever it is stabilizing to some $c \in C$;*
- ascending *whenever it is increasing and non-stabilizing.*

Remark. A sequence is ascending iff it is increasing and has a strictly increasing subsequence, and each increasing and stabilizing sequence is stabilizing to a unique value.

Definition 2 *A Complete Partial Order (CPO) is a tuple $(C, \preceq, \bot)$, where $\preceq$ is a partial order on C, $\bot$ is the least element for $\preceq$, and such that each increasing sequence $(s_n)_{n\in\mathbb{N}}$ over C has a least upper bound.*

Hereafter we call least upper bounds *limits* and denote the limit of a sequence $(s_n)_{n\in\mathbb{N}}$ by $lim[(s_n)_{n\in\mathbb{N}}]$.

An important feature of CPOs is that the order $\preceq$ should be interpreted as a definition order. In this sense, $\bot$ is the *least defined* element, and elements that are maximal with respect to the order are interpreted as are "fully defined". Intermediate, non-maximal elements are therefore "partially defined".

2.2 Partially Ordered Sets with Least Element and Approximations

Assumption 1 *In the sequel we assume:*

1. *a partially ordered set $(C^\circ, \preceq^\circ)$ with a least element $\bot$. Elements of C° are often called* finite*. It is required that no finite element be an upper bound for an ascending sequence of finite elements;*
2. *for each $N \in \mathbb{N}$, an approximation function $\lceil\cdot\rceil_N : C^\circ \to C^\circ$. For any $c \in C^\circ$, $\lceil c\rceil_N$ is called the N-th approximation of c, and satisfies the following properties:*
   (a) *for all $c \in C^\circ$ and $N \in \mathbb{N}$, $\lceil c\rceil_N \preceq^\circ c$;*
   (b) *for all $c, c' \in C^\circ$ and $N \in \mathbb{N}$, $c \preceq c'$ implies $\lceil c\rceil_N \preceq^\circ \lceil c'\rceil_N$;*
   (c) *for all $c \in C^\circ$ and $N, N' \in \mathbb{N}$, $N \leq N'$ implies $\lceil c\rceil_N \preceq^\circ \lceil c\rceil_{N'}$;*
   (d) *for all $c \in C^\circ$ there exists $N \in \mathbb{N}$ such that $\lceil c\rceil_N = c$;*
   (e) *for all $c \in C^\circ$ and $N, N' \in \mathbb{N}$, $N \leq N'$ implies $\lceil\lceil c\rceil_{N'}\rceil_N = \lceil c\rceil_N$;*
3. *for any $N \in \mathbb{N}$ and any ascending sequence $(s_n)_{n\in\mathbb{N}}$, $(\lceil s_n\rceil_N)_{n\in\mathbb{N}}$ is stabilizing.*

Remark. The requirements in Assumption 1 are quite natural. Item 1 holds if finite elements have finitely many finite elements smaller than them, a property which may be expected from finite elements. Item 2.(a) says that approximations do approximate w.r.t. the order, items 2.(b) and 2.(c) says that approximations are monotonic in both arguments; item 2.(d) says that for finite elements approximation can be exact, and item 2.(e) says that approximating first at some order $N' \geq N$ and then approximating at order N amounts to approximating at order N. Finally, item 3 says that mapping approximations to ascending sequences makes them stabilize - the sequence of approximations cannot grow forever.

Example 1 *Let C° be the set T of trees inductively defined by the rules $\bot \in T$ and for each list l over T, tree $l \in T$. Let $\preceq^\circ$ be the prefix relation on T, inductively defined by $\bot \preceq^\circ t$ for each $t \in T$, and, for every pair of lists l, l' over T having the same length, say, m, tree $l \preceq^\circ$ tree l' whenever $l[i] \preceq^\circ l'[i]$ for each $i < n$ (where $l[i]$ denotes the i-th element of the list l). Let $\lceil\cdot\rceil_N$ ("cut at height N") denote the function $\lambda\, t.$ if $N = 0 \vee t = \bot$ then $\bot$ else tree$(map(\lceil\cdot\rceil_{N-1})($forest $t)$ where forest$($tree $l) \triangleq l$ and map is the mapping function on lists. We have proved in Coq that Assumption 1 holds for these definitions. The code is available at* `https://project.inria.fr/wpte2022/`.

Remark. Item 2.(b) in the above Assumption implies that for increasing sequence $(s_n)_{n\in\mathbb{N}}$, the sequence $(\lceil s_n\rceil_N)_{n\in\mathbb{N}}$ is increasing. Hence, for ascending sequences $(s_n)_{n\in\mathbb{N}}$ the sequence $(\lceil s_n\rceil_N)_{n\in\mathbb{N}}$ is increasing and stabilizing (cf. Item 3 of the Assumption) and therefore has a limit.

Lemma 1 *Assume two increasing and stabilizing sequences $(s_n)_{n\in\mathbb{N}}, (s'_n)_{n\in\mathbb{N}}$ over C°, such that for all $n \in \mathbb{N}$, $s_n \preceq^\circ s'_n$. Then $lim[(s_n)_{n\in\mathbb{N}}] \preceq^\circ lim[(s'_n)_{n\in\mathbb{N}}]$.*

Proof. Per the above observations, $lim[(s_n)_{n\in\mathbb{N}}]$ is the value at which $(s_n)_{n\in\mathbb{N}}$ stabilizes, hence, there exists $i \in \mathbb{N}$ such that for all $j \geq i$, $s_j = lim[(s_n)_{n\in\mathbb{N}}]$. Similarly, there exists $i' \in \mathbb{N}$ such that for all $j \geq i'$, $s'_j = lim[(s'_n)_{n\in\mathbb{N}}]$. Set $i'' := max\ i\ i'$. Then, $lim[(s_n)_{n\in\mathbb{N}}] = s_{i''}$; by hypothesis of the lemma $s_{i''} \preceq^\circ s'_{i''}$ holds; and $s_{i''} = lim[(s'_n)_{n\in\mathbb{N}}]$. By transitivity, $lim[(s_n)_{n\in\mathbb{N}}] \preceq^\circ lim[(s'_n)_{n\in\mathbb{N}}]$, which proves the lemma.

2.3 An Equivalence Relation on Sequences

Similar sequences eventually reach elements that are pairwise equal up to arbitrary close approximations:

Definition 3 *Two sequences $(s_n)_{n\in\mathbb{N}}, (s'_n)_{n\in\mathbb{N}}$ over C° are similar, written $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$, whenever for all $N \in \mathbb{N}$ there exists $i \in \mathbb{N}$ such that for all $j \geq i$, $\lceil s_j \rceil_N = \lceil s'_j \rceil_N$.*

Lemma 2 *The similarity relation $\sim$ is an equivalence relation on sequences over C°.*

Proof. Reflexivity and symmetry are immediate. For transitivity, assume $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ and $(s'_n)_{n\in\mathbb{N}} \sim (s''_n)_{n\in\mathbb{N}}$. Choose any $N \in \mathbb{N}$. From $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ we obtain $i \in \mathbb{N}$ such that for all $j \geq i$, $\lceil s_j \rceil_N = \lceil s'_j \rceil_N$, and from $(s'_n)_{n\in\mathbb{N}} \sim (s''_n)_{n\in\mathbb{N}}$ we obtain $i' \in \mathbb{N}$ such that for all $j \geq i'$, $\lceil s'_j \rceil_N = \lceil s''_j \rceil_N$. Setting $i'' := max\ i\ i'$ we get: for all $j \geq i''$, $\lceil s_j \rceil_N = \lceil s'_j \rceil_N = \lceil s''_j \rceil_N$; which proves $(s_n)_{n\in\mathbb{N}} \sim (s''_n)_{n\in\mathbb{N}}$ and establishes transitivity.

Lemma 3 *For an ascending sequence $(s_n)_{n\in\mathbb{N}}$, let s be the (unique) value at which $(\lceil s_n \rceil_N)_{n\in\mathbb{N}}$ stabilizes. If $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ then $(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}$ also stabilizes at s.*

Proof. From the stabilization hypothesis we obtain $i \in \mathbb{N}$ such that for all $j \geq i$, $\lceil s_j \rceil_N = s$. From $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ we obtain i' such that for all $j \geq i'$, $\lceil s_j \rceil_N = \lceil s'_j \rceil_N$. Setting $i'' := max\ i\ i'$ we obtain that for all $j \geq i''$, $\lceil s'_j \rceil_N = \lceil s_j \rceil_N = s$. Hence, $(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}$ stabilizes at s; which proves the lemma.

Lemma 3 helps defining approximations of equivalence classes $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N$ of ascending sequences:

Definition 4 *For an ascending sequence over C°, $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N = lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}]$.*

Indeed, Lemma 3 says that the above definition is independent on the chosen representative in the class. This is essential in the rest of the paper. Unless otherwise stated, sequences are assumed to be over C°.

Lemma 4 *For ascending sequences $(s_n)_{n\in\mathbb{N}}$ and $(s'_n)_{n\in\mathbb{N}}$ it holds that $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ if and only if for all $N \in \mathbb{N}$, $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N = \lceil [(s'_n)_{n\in\mathbb{N}}]_\sim \rceil_N$.*

Proof. ($\Rightarrow$): per the above remarks, $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N = lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}]$ and $\lceil [(s'_n)_{n\in\mathbb{N}}]_\sim \rceil_N = lim[(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}]$. If $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ we know by Lemma 3 that the value $lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}]$ at which $(\lceil s_n \rceil_N)_{n\in\mathbb{N}}$ stabilizes equals the value $lim[(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}]$ at which $(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}$ stabilizes. Hence the conclusion $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N = \lceil [(s'_n)_{n\in\mathbb{N}}]_\sim \rceil_N$.

($\Leftarrow$) : Choose an arbitrary $N \in \mathbb{N}$. Per the above remarks, from the hypothesis $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N = \lceil [(s'_n)_{n\in\mathbb{N}}]_\sim \rceil_N$ we obtain that the value $lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}]$ at which $\lceil (s_n)_{n\in\mathbb{N}} \rceil_N$ stabilizes equals the value $lim[(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}]$ at which $\lceil (s'_n)_{n\in\mathbb{N}} \rceil_N$ stabilizes. Let i be the least natural number such that for all $j \geq i$, $\lceil s_j \rceil_N = lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}]$ and let i' be the least natural number such that for all $j \geq i'$, $\lceil s'_j \rceil_N = lim[(\lceil s'_n \rceil_N)_{n\in\mathbb{N}}]$. Set $i'' := max\ i\ i'$. Hence, for all $j \geq i''$, $\lceil s_j \rceil_N = \lceil s'_j \rceil_N$ and since N was chosen arbitrarily, by Definition 3, $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$; which proves the ($\Leftarrow$) implication and the lemma.

2.4 An Order Relation on Equivalence Classes of Sequences

Notation. We denote by K the set of equivalence classes of ascending sequences of elements in C°.

Definition 5 *For $k, k' \in K$, $k \precsim k'$ whenever for all $N \in \mathbb{N}$, $\lceil k \rceil_N \leq^\circ \lceil k' \rceil_N$.*

Lemma 5 *The relation $\precsim$ is a partial order on K.*

Proof. Reflexivity and transitivity of $\precsim$ easily result from the corresponding properties of $\preceq^\circ$. For antisymmetry: let $k = [(s_n)_{n\in\mathbb{N}}]_\sim$ and $k' = [(s'_n)_{n\in\mathbb{N}}]_\sim$. Then, $k \precsim k'$ means that for all $N \in \mathbb{N}$, $\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil_N \preceq^\circ \lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil_N$, and $k' \precsim k$ means that for all $N \in \mathbb{N}$, $\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil_N \preceq^\circ \lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil_N$. As $\preceq^\circ$ is antisymmetric, for all $N \in \mathbb{N}$, $\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil_N = \lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil_N$, and by Lemma 4, $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$, i.e., $[(s_n)_{n\in\mathbb{N}}]_\sim = [(s'_n)_{n\in\mathbb{N}}]_\sim$: the lemma is proved.

We give an equivalent characterization of the order $\precsim$, expressed as Lemma 7 below. To prove that lemma we need the monotonicity of approximations of equivalence classes:

Lemma 6 *If $k \in K$ and $N \leq N'$ then $\lceil k\rceil_N \preceq^\circ \lceil k\rceil_{N'}$.*

Proof. Let $k = [(s_n)_{n\in\mathbb{N}}]_\sim$, thus, $\lceil k\rceil_N = lim[(\lceil s_n\rceil_N)_{n\in\mathbb{N}}]$ and $\lceil k\rceil_{N'} = lim[(\lceil s_n\rceil_{N'})_{n\in\mathbb{N}}]$. Using Assumption 1 item 2.(c), for all $n \in \mathbb{N}$, $\lceil s_n\rceil_N \preceq^\circ \lceil s_n\rceil_{N'}$. By Lemma 1 we obtain $lim[(\lceil s_n\rceil_N)_{n\in\mathbb{N}}] \preceq^\circ lim[(\lceil s_n\rceil_{N'})_{n\in\mathbb{N}}]$. Hence the conclusion $\lceil k\rceil_N \preceq^\circ \lceil k\rceil_{N'}$.

Lemma 7 *$k \precsim k'$ if and only if for all $N \in \mathbb{N}$ there is $N' \in \mathbb{N}$ such that $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_{N'}$.*

Proof. ($\Rightarrow$): this direction is trivial, as by Definition 5 $k \precsim k'$ means that for all $N \in \mathbb{N}$ there does exist $N' := N$ such that $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_{N'}$.

($\Leftarrow$): Choose an arbitrary $N \in \mathbb{N}$. For the corresponding $N' \in \mathbb{N}$ we have two cases:

- either $N' < N$, in which case by Lemma 6, $\lceil k'\rceil_{N'} \preceq^\circ \lceil k'\rceil_N$ and by transitivity using hypothesis $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_{N'}$ we obtain $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_N$;
- or $N \geq N'$, and from $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_{N'}$, using Assumption 1 item 2.(b) we obtain $\lceil\lceil k\rceil_N\rceil_N \preceq^\circ \lceil\lceil k'\rceil_{N'}\rceil_N$, which, by using item 2.(e), amounts to $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_N$.

Since $N \in \mathbb{N}$ has been arbitrarily chosen and in all cases $\lceil k\rceil_N \preceq^\circ \lceil k'\rceil_N$, by Definition 5 we obtain $k \precsim k'$, which proves the ($\Leftarrow$) implication and the lemma.

3 Completion

We show how a partially ordered set with a least element $(C^\circ, \preceq^\circ, \bot)$ satisfying Assumption 1 can be *completed* in order to obtain a CPO $(C, \preceq, \bot)$. We have already noted that increasing and stabilizing sequences have limits. Hence, limits have to be constructed for the ascending sequences.

Definition 6 *Given a triple $(C^\circ, \preceq^\circ, \bot)$ satisfying Assumption 1, the* completion *operation consists in*

1. *extending the base set C° to a set $C = C^\circ \cup K$ where K is the set of equivalence classes $[(s_n)_{n\in\mathbb{N}}]_\sim$ of ascending sequences over C°;*
2. *defining the limits of each ascending sequence $(s_n)_{n\in\mathbb{N}}$ over C° to be the equivalence class of the sequence: $lim[(s_n)_{n\in\mathbb{N}}] \triangleq [(s_n)_{n\in\mathbb{N}}]_\sim$.*
3. *extending the order $\preceq^\circ$ on C° to a relation $\preceq$ on C as follows:*
   (a) *for $c, c' \in C^\circ$, $c \preceq c$ whenever $c \preceq^\circ c'$;*
   (b) *for $c \in C$ and $k \in K$, $c \preceq k$ whenever there exists $N \in \mathbb{N}$ such that $c \preceq^\circ \lceil k\rceil_N$;*
   (c) *for $k, k' \in K$, $k \preceq k'$ whenever $k \precsim k'$.*

Remark. Definition 6 says nothing about limits of increasing sequences containing elements that are themselves limits (from K). Such sequences are dealt with ahead in the paper and shown to have limits among the new elements. Hence, the CPO construction does not require further completion steps.

A key difference with the earlier completion operation in [7] is that equivalence classes there were all maximal with respect to the order; this was required in order to exclude limit elements (in K) from occurring in ascending sequences. In the new construction, limit elements are not excluded any more from ascending sequences, which enables us to have a nontrivial order on K, in which maximal elements are seen as "fully defined", but in which "partially defined", non-maximal limit elements also exist.

We now show that the relation $\preceq$ is an order and that the constructed limits are least upper bounds of the respective sequences. This is then used in the next subsection in order to provide limits to sequences that include both finite elements and equivalence classes, and to prove that those limits are least upper bounds as well, thereby completing the organization of old and new elements and their order as a CPO.

Lemma 8 *In the context of Definition 6, the relation $\preceq$ is an order relation on C.*

Proof. The reflexivity of $\preceq$ follows from those of $\preceq^\circ$ and of $\precsim$. For antisymmetry: on C° it follows from the antisymmetry of $\preceq^\circ$. On K, it follows from the antisymmetry of $\precsim$. These are the only situations to consider, as for $c \in C^\circ$ and $k \in K$ having $k \preceq c$ is impossible according to Definition 6.

There remains to prove that $\preceq$ is transitive. There are only 4 possible cases:

1. $c, c', c'' \in C^\circ$ with $c \preceq c'$, $c' \preceq c''$: by Definition 6 of completion item 3.(a), here $\preceq$ is $\preceq^\circ$ and the required $c \preceq c''$ results from the transitivity of $\preceq^\circ$;
2. $c, c' \in C^\circ$ and $k \in K$ with $c \preceq c'$ and $c' \preceq k$: by Definition 6 item 3.(b), the latter means there exists $N \in \mathbb{N}$ such that $c' \preceq^\circ \lceil k \rceil_N$, and since $c \preceq c'$, the same N ensures $c \preceq^\circ \lceil k \rceil_N$ and thus $c \preceq k$;
3. $c \in C^\circ$ and $k, k' \in K$ with $c \preceq k$ and $k \preceq k'$: by Definition 6 item 3.(b), there exists $N \in \mathbb{N}$ such that $c \preceq^\circ \lceil k \rceil_N$, and by Definition 5 and item 3.(c), $\lceil k \rceil_N \preceq^\circ \lceil k' \rceil_N$, thus, by transitivity, there exists $N \in \mathbb{N}$ such that $c \preceq^\circ \lceil k' \rceil_N$, meaning that $c \preceq k'$;
4. $k, k', k'' \in K$ with $k \preceq k'$ and $k' \preceq k''$: by Definition 6 item 3.(c), here $\preceq$ is $\precsim$ and the required $k \preceq k''$ results from the transitivity of $\precsim$. This concludes the proof of transitivity for $\preceq$ and of the lemma as a whole.

Lemma 9 *In the context of Definition 6, the limit $lim[(s_n)_{n\in\mathbb{N}}]$ of an ascending sequence $(s_n)_{n\in\mathbb{N}}$ over C° is a least upper bound for $(s_n)_{n\in\mathbb{N}}$.*

Proof. By Definition 6, given an ascending sequence $(s_n)_{n\in\mathbb{N}}$, it limit $lim[(s_n)_{n\in\mathbb{N}}]$ is defined to be the equivalence class $[(s_n)_{n\in\mathbb{N}}]_\sim$ of the sequence.

We first prove that $[(s_n)_{n\in\mathbb{N}}]_\sim$ is an upper bound for the sequence $(s_n)_{n\in\mathbb{N}}$. That is, we must prove that for all $n \in \mathbb{N}$, $s_n \preceq [(s_n)_{n\in\mathbb{N}}]_\sim$. Choose an arbitrary $n \in \mathbb{N}$, and choose N such that $\lceil s_n \rceil_N = s_n$. This is made possible by Assumption 1 item 2.(d). We then have $s_n = \lceil s_n \rceil_N \preceq^\circ lim[(\lceil s_n \rceil_N)_{n\in N}] = \lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N$. Hence, $s_n \preceq^\circ \lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N$ and, by Definition 6 item 3.(b), $s_n \preceq [(s_n)_{n\in\mathbb{N}}]_\sim$. The upper-bound claim of $[(s_n)_{n\in\mathbb{N}}]_\sim$ with respect to $(s_n)_{n\in\mathbb{N}}$ is proved.

We now prove that $[(s_n)_{n\in\mathbb{N}}]_\sim$ is the least upper bound of $(s_n)_{n\in\mathbb{N}}$. Assume $b \in C$ such that for all $n \in \mathbb{N}$, $s_n \preceq b$. There are two cases to consider: $b \in C^\circ$ or $b \in K$. If $b \in C^\circ$ then for all $n \in \mathbb{N}$, $s_n \preceq^\circ b$. But this is impossible by Assumption 1 item 1 since $(s_n)_{n\in\mathbb{N}}$ is ascending.

If $b \in K$, then $b = [(s'_n)_{n\in\mathbb{N}}]_\sim$ for some ascending sequence $(s'_n)_{n\in\mathbb{N}}$. From the fact that for all $n \in \mathbb{N}$, $s_n \preceq [(s'_n)_{n\in\mathbb{N}}]_\sim$, using Assumption 1 item 2.(b) we obtain that for all $n \in \mathbb{N}$, there is $N' \in \mathbb{N}$ such that $s_n \preceq^\circ \lceil [(s'_n)_{n\in\mathbb{N}}]_\sim \rceil_{N'}$. Choose an arbitrary $N \in \mathbb{N}$. Since for all $n \in \mathbb{N}$, $\lceil s_n \rceil_N \preceq^\circ s_n$ we further obtain that for all $n \in \mathbb{N}$ there exists $N' \in \mathbb{N}$ such that $\lceil s_n \rceil_N \preceq^\circ \lceil [(s'_n)_{n\in N}]_\sim \rceil_{N'}$. Now, the sequence $(\lceil s_n \rceil_N)_{n\in\mathbb{N}}$ is increasing and stabilizing, and $lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}]$ is an element, say, $\lceil s_k \rceil_N$ of the sequence. We thus obtain that for the arbitrarily chosen $N \in \mathbb{N}$, there exists $N' \in \mathbb{N}$ such that $lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}] \preceq^\circ \lceil [(s'_n)_{n\in N}]_\sim \rceil_{N'}$.

Moreover, we know that $lim[(\lceil s_n \rceil_N)_{n\in\mathbb{N}}] = \lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N$. We thus obtain that for all $N \in \mathbb{N}$, there exists $N' \in \mathbb{N}$ such that $\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil_N \preceq^\circ \lceil [(s'_n)_{n\in N}]_\sim \rceil_{N'}$. By Lemma 7 we obtain $[(s_n)_{n\in\mathbb{N}}]_\sim \precsim [(s'_n)_{n\in\mathbb{N}}]_\sim$. Then, by Definition 6 item 3.(c), $[(s_n)_{n\in\mathbb{N}}]_\sim \preceq [(s'_n)_{n\in\mathbb{N}}]_\sim = b$. Thus, in the only possible case ($b \in K$) we have obtained that $[(s_n)_{n\in\mathbb{N}}]_\sim$ is $\preceq$-ordered with b. This concludes the proof of the lemma.

3.1 Completion + Diagonalization = CPO

In this section we show how to "finish" the completion operation in the previous section in order to obtain a CPO. We add what is missing: limits for ascending sequences that include equivalence classes. The main idea is to use a diagonalization technique to extract an ascending sequence of finite elements from an ascending sequence that includes equivalence classes. We need a few technical lemmas. The first lemma adapts Assumption 1 items 2.(a)-(c), which hold for elements of C°, to the elements of K.

Lemma 10

1. *for all $k \in K$ and $N \in \mathbb{N}$, $\lceil k \rceil_N \preceq k$;*
2. *for all $k, k' \in K$ and $N \in \mathbb{N}$, $k \preceq k'$ implies $\lceil k \rceil_N \preceq \lceil k' \rceil_N$;*
3. *for all $k \in K$ and $N, N' \in \mathbb{N}$, $N \leq N'$ implies $\lceil k \rceil_N \preceq \lceil k \rceil_{N'}$.*

Proof.

1. The approximation $\lceil k \rceil_N$ is an element of C°, hence, by Definition 6 item 3.(b), for $\lceil k \rceil_N \preceq k$ it is enough to show that there exists $N' \in \mathbb{N}$ such that $\lceil k \rceil_N \preceq \lceil k \rceil_{N'}$. Setting $N' := N$ ensures this;
2. By Definition 6 item 3.(c), $k \preceq k'$ is just $k \precsim k'$, which by Definition 5 is just the conclusion of this item - that for all $N \in \mathbb{N}$, $\lceil k \rceil_N \preceq \lceil k' \rceil_N$;
3. Both $\lceil k \rceil_N$ and $\lceil k \rceil_{N'}$ are elements of C°, thus, $\preceq = \preceq^\circ$, and the conclusion follows by Lemma 6.

Notation. For equivalence classes $k, k' \in K$ we write $k \prec k'$ for $k \precsim k'$ and $k \neq k'$. We denote by C^ω the set of sequences over a set C. We call the *diagonal* of a $s \in (C^\circ \cup K)^\omega$ the sequence $(\lceil s_n \rceil_n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$.

Lemma 11 *$k \prec k'$ implies that for all $N \in \mathbb{N}$, $\lceil k \rceil_N \prec^\circ \lceil k' \rceil_N$.*

Proof. Choose an arbitrary $N \in \mathbb{N}$. From Lemma 7 we obtain $\lceil k \rceil_N \preceq^\circ \lceil k' \rceil_N$. There remains to show that the previous inequality is strict. For, assuming the contrary, we would have that for all $N \in \mathbb{N}$, $\lceil k \rceil_N = \lceil k' \rceil_N$. Then, $k \precsim k'$ and $k' \precsim k$ follow by Definition 7, implying $k = k'$, in contradiction to $k \prec k'$.

Lemma 12 *If $s \in (C^\circ \cup K)^\omega$ is increasing, the sequence $(\lceil s_n \rceil_n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$ is increasing as well.*

Proof. Any two consecutive elements $\lceil s_{i-1} \rceil_{i-1}$ and $\lceil s_i \rceil_i$ satisfy $\lceil s_{i-1} \rceil_{i-1} \preceq \lceil s_i \rceil_i$ by Lemma 10 items 2&3.

Lemma 13 *If $s \in (C^\circ \cup K)^\omega$ is ascending and there is $i \in \mathbb{N}$ such that $s_i \in K$, then $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$ is ascending.*

Proof. By Lemma 12 the sequence $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$ is increasing. We only need to prove that it has a strictly increasing subsequence. Since s is ascending it has a strictly increasing subsequence $(s_{i_n})_{n\in\mathbb{N}}$. Since $s_i \in K$, we obtain that for all $j \geq i, s_j \in K$. Consider the subsequence $(s_{i_n})_{n\in\mathbb{N}, i_n \geq i}$ of $(s_{i_n})_{n\in\mathbb{N}}$. As a subsequence of an strictly increasing subsequence, $(s_{i_n})_{n\in\mathbb{N}, i_n \geq i}$ is strictly increasing, thus, for all $n \in \mathbb{N}$ s.t. $i_n \geq i$, $s_{i_n} \prec s_{i_{n+1}}$. Fix such an $n \in \mathbb{N}$. Using Lemma 11 with $N := i_n$ we obtain $\lceil s_{i_n} \rceil_{i_n} \prec^\circ \lceil s_{i_{n+1}} \rceil_{i_n}$. By Assumption 1 item 2.(c), $\lceil s_{i_{n+1}} \rceil_{i_n} \preceq^\circ \lceil s_{i_{n+1}} \rceil_{i_{n+1}}$. Overall, $\lceil s_{i_n} \rceil_{i_n} \prec^\circ \lceil s_{i_{n+1}} \rceil_{i_{n+1}}$; and since n was chosen arbitrarily, $(\lceil s_{i_n} \rceil_{i_n})_{n\in\mathbb{N}, i_n \geq i}$ is a strictly increasing subsequence of $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$, proving the lemma.

The last lemma in this section provides limits to sequences $s \in (C^\circ \cup K)^\omega$ with at least one element in K.

Lemma 14 *If $s \in (C^\circ \cup K)^\omega$ is ascending and $s_i \in K$, then $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$ is the least upper bound for s.*

Proof. We first prove that $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$ is an upper bound for s. Thus, we need to show that for all $j \in \mathbb{N}$, $s_j \preceq [(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$. Since s is increasing, it is enough to show the above inequality for $j \geq i$, i.e., when $s_j \in K$. By Lemma 7 we only need to show (†): *for all $N \in \mathbb{N}$, there exists $N' \in \mathbb{N}$ such that* $\lceil s_j \rceil_N \preceq^\circ \lceil [(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim \rceil_{N'}$. Choose an arbitrary $N \in \mathbb{N}$. Let $p = max\ j\ N$. Thus, by Assumption 1 item 2.(b), $\lceil s_j \rceil_N \preceq^\circ \lceil s_p \rceil_N$. Using Assumption 1 item 2.(c), $\lceil s_p \rceil_N \preceq^\circ \lceil s_p \rceil_p$. By transitivity, $\lceil s_j \rceil_N \preceq^\circ \lceil s_p \rceil_p$. But since (by Lemma 13) $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]$ is ascending, we know by Lemma 9 that $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$ is an upper bound for $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$. In particular, $\lceil s_p \rceil_p \preceq^\circ [(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$, and by transitivity, $\lceil s_j \rceil_N \preceq^\circ [(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$, which by Definition 6 item 3.(b) implies that there exists $N' \in \mathbb{N}$ such that $\lceil s_j \rceil_N \preceq^\circ \lceil [(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim \rceil_{N'}$, i.e., (†).

We now prove that $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$ is a least upper bound for s. Thus, assuming an upper bound $b \in C^\circ \cup K$ for s, we need to show that $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim \preceq b$. Now, by Lemma 9, $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim$ is the least upper bound for $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$. Hence in order to show $[(\lceil s_n \rceil_n)_{n\in\mathbb{N}}]_\sim \preceq b$ it is enough to show that b is an upper bound for $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$. Choose an arbitrary $n \in \mathbb{N}$. We have, by Lemma 10 item 1, $\lceil s_n \rceil_n \preceq s_n$ and since b is an upper bound for s, $s_n \preceq b$. Hence, for an arbitrary $n \in \mathbb{N}$, $\lceil s_n \rceil_n \preceq b$, which establishes that b is an upper bound for $(\lceil s_n \rceil_n)_{n\in\mathbb{N}}$, which completes the proof of the lemma.

4 Rose Trees as CPOs

In this section we focus on the definition of Rose trees, i.e., trees with finite breadth and possibly infinite depth, using the finite trees defined in Example 1 and the CPO construction in the previous section.

Remember from Example 1 that finite trees C° have constructors $\bot$ and *tree a l* for finite lists l of finite trees. The prefix order $\preceq^\circ$ is such that for all $t \in C^\circ$, $\bot \preceq^\circ t$, and for all lists l, l' of trees having the same length m, *tree* $l \preceq^\circ$ *tree* l' if and only if for all $i < m$, $l[i] \preceq^\circ l'[i]$. Together with their approximations they satisfy Assumption 1 and therefore constitute a CPO $(C^\circ \cup K, \preceq, \bot)$ where K is the set of equivalence classes of ascending sequences of finite trees. Such ascending sequences require trees having at least one branch that grows "forever", and their limits (in K) have at least one infinite branch.

Remark. Thanks to diagonalization, all sequences considered below are constituted only of finite trees.

Overall, the set K together with the set C° of finite trees constitute the set $C^\circ \cup K$ of Rose trees - with finite breadth and possibly infinite depth. Among Rose trees there are those who do not have $\bot$ as a subtree. Such trees are *maximal* for the $\preceq$ order and, since this order is a definition order, the maximal elements are interpreted as "completely defined". There are also elements that have some infinite subtrees and some $\bot$ subtrees. Those are non-maximal for the definition order $\preceq$ and hence not fully defined.

4.1 Extending the *tree* Constructor of Finite Trees to Rose Trees

We now define a function *tree* that takes a list of Rose trees (defined above) and produces a Rose tree.

Remark. In the sequel we identify the *equivalence class of an increasing and stabilizing sequence* with *the value at which all sequence in the class stabilize*, i.e. their common limit. Hence, for increasing and stabilizing sequences, $[(s_n)_{n\in\mathbb{N}}]_\sim = lim[(s_n)_{n\in\mathbb{N}}]$. Since this identity also holds for ascending sequences (cf. Definition 6), it holds for all increasing sequences, which enables us to treat them uniformly.

Consider now a list $l = [t^1, \ldots, t^m]$ of elements in $C = C^\circ \cup K$. Thanks to the above identification, each t^i equals the equivalence class of some increasing sequence $(t^i_n)_{n\in\mathbb{N}}$ that converges to it: for all $i = 1, \ldots m$, $t^i = [(t^i_n)_{n\in\mathbb{N}}]_\sim = lim[(t^i_n)_{n\in\mathbb{N}}]$. Let us consider the sequence of lists $([t^1_n, \ldots, t^m_n])_{n\in\mathbb{N}}$. This sequence is

"pointwise" increasing: for all $n \in \mathbb{N}$ and $i = 1,\dots m$, $t^i_n \preceq^\circ t^i_{n+1}$, and, thanks to the monotony of the $\preceq^\circ$ (prefix) order, the sequence $(tree[t^1_n,\dots,t^m_n])_{n\in\mathbb{N}}$ is increasing. This justifies the following definition:

Definition 7 (*tree* function on Rose Trees) *tree* $[[(t^1_n)_{n\in\mathbb{N}}]_\sim,\dots,[(t^m_n)_{n\in\mathbb{N}}]_\sim] \triangleq lim[(tree[t^1_n,\dots,t^m_n])_{n\in\mathbb{N}}]$.

We now show that the above definition does not depend on the choice of representatives in the equivalence classes $[(t^1_n)_{n\in\mathbb{N}}]_\sim,\dots,[(t^m_n)_{n\in\mathbb{N}}]_\sim$. Thus, it is required to prove that for each pair of lists of increasing sequences $[(t^1_n)_{n\in\mathbb{N}},\dots(t^m_n)_{n\in\mathbb{N}}]$ and $[(t'^1_n)_{n\in\mathbb{N}},\dots(t'^m_n)_{n\in\mathbb{N}}]$ satisfying $(t^i_n)_{n\in\mathbb{N}} \sim (t'^i_n)_{n\in\mathbb{N}}$ for all $i = 1,\dots,m$, it holds that $(tree[t^1_n,\dots t^m_n])_{n\in\mathbb{N}} \sim (tree[t'^1_n,\dots t'^m_n])_{n\in\mathbb{N}}$. Fix an arbitrary $N \in \mathbb{N}$. Now, from $(t^i_n)_{n\in\mathbb{N}} \sim (t'^i_n)_{n\in\mathbb{N}}$ for all $i = 1,\dots,m$, we obtain, for each $i = 1,\dots,m$, some $j_i \in \mathbb{N}$ such that for all $j \geq j_i, \lceil t^i_j\rceil_N = \lceil t'^i_j\rceil_N$. Setting J to be the maximum of the j_i and by using the definition of approximations for trees, we obtain for all $j \geq J$, $\lceil tree[t^1_j,\dots t^m_j]\rceil_{N+1} = tree[\lceil t^1_j\rceil_N,\dots,\lceil t^m_j\rceil_N] = tree[\lceil t'^1_j\rceil_N,\dots,\lceil t'^m_j\rceil_N] = \lceil tree[t'^1_j,\dots t'^m_j]\rceil_{N+1}$. Combined with the fact that for all j (in particular for $j \geq J$), $\lceil tree[t^1_j,\dots t^m_j]\rceil_0 = \bot = \lceil tree[t'^1_j,\dots t'^m_j]\rceil_0$, for the arbitrarily chosen $N \in \mathbb{N}$ there is $J \in \mathbb{N}$ such for all $j \geq M, \lceil tree[t^1_j,\dots t^m_j]\rceil_N = (\lceil tree[t'^1_j,\dots t'^m_j])\rceil_N$. But by definition of $\sim$ this is just $(tree[t^1_n,\dots t^m_n])_{n\in\mathbb{N}} \sim (tree[t'^1_n,\dots t'^m_n])_{n\in\mathbb{N}}$, which is what we had to prove in order to show the independence from representatives in Definition 7 of the *tree* function for Rose trees. There remains to prove that the function satisfies the following properties of its restriction to finite trees:

- approximations: $\lceil tree\ l\rceil_0 = \bot$ and $\lceil tree\ l\rceil_{N+1} = tree(map\ \lceil\cdot\rceil_N)\ l)$;
- monotonicity : $tree\ l \preceq tree\ l'$ iff for some $m \in \mathbb{N}$, $length\ l = length\ l' = m$ and for all $i < m$, $l[i] \preceq l'[i]$;
- surjectiveness for nonempty trees: for all $t \in C \setminus \{\bot\}$, there exists l such that $t = tree\ l$;
- injectiveness: for all lists of Rose trees l, l', $tree\ l = tree\ l'$ implies $l = l'$.

In particular, the last two properties will enable us to define the *forest* accessor for nonempty trees as: *forest t* is the (unique) list l such that $t = tree\ l$.

4.2 Towards Defining Corecursive Functions Without Corecursion

Definition 7 and its properties enables us to define corecursive functions based on some results from [7].

4.2.1 Productive Functionals and Unique Fixpoints

Assume a CPO $(C, \preceq, \bot)$. We extend the order $\preceq$ to functions $D \to C$ by $f_1 \preceq f_2$ iff $f_1\, x \preceq f_2\, x$ for all $x \in D$.

Definition 8 *A functional* $F : (D \to C) \to D \to C$ *is* increasing *if for all* $f_1, f_2 : D \to C$, $f_1 \preceq f_2$ *implies* $F f_1 \preceq F f_2$.

Consider a functional $F : (D \to C) \to D \to C$ as above. Let $\perp\!\!\!\perp : D \to C$ be the constant function such that $\perp\!\!\!\perp\, x = \bot$, for all $x \in D$, and let $F^n : (D \to C) \to D \to C$ be the functional inductively defined by $F^0 f = f$ and, for all $n \in \mathbb{N}$, $F^{n+1} f = F(F^n f)$.

Definition 9 *A functional* $F : (D \to C) \to D \to C$ *is productive whenever it is increasing and for all* $x \in D$, *the limit of the (increasing) sequence* $(F^n \perp\!\!\!\perp x)_{n\in\mathbb{N}}$ *is maximal w.r.t. the order* $\preceq$.

Theorem 1 ([7]) *If a functional* F *is productive then* $lim[(F^n \perp\!\!\!\perp)_{n\in\mathbb{N}}]$ *is the unique fixpoint of* F.

4.2.2 Towards Defining a Mirror Function for Rose Trees

The *mirror* function for Rose trees, which we intend to define via its fixpoint equation, has the following functional: $Mirror = \lambda f.\lambda t.\, if\ t = \bot\ then \bot\ else\ map\ f\ (reverse\ (forest\ t))$, where *reverse* is the standard function that reverses lists. For defining the *mirror* function via the equation $mirror = Mirror\ mirror$ we shall use Theorem 1, which requires us to prove that the functional *Mirror* is productive. We conjecture that this requires us to first define $mirror^\circ = \lambda t.\, if\ t = \bot\ then \bot\ else\ map\ mirror^\circ (reverse\ (forest\ t))$ for finite trees t, and then prove by induction on $n \in \mathbb{N}$ that for all Rose trees t, $Mirror^n \Lsh\, t = mirror^\circ(\lceil t \rceil_n)$. The latter identity is useful because $\lceil t \rceil_n$ has the property that all its $\bot$ subtrees are at the same "distance" from the root of t, which coincides with the tree's height. Hence, for infinite maximal trees t, the sequence $(Mirror^n \Lsh\, t)_{n\in\mathbb{N}}$ is ascending and "fair", in the sense that there remain no "undeveloped" $\bot$ subtree.

Hence, in the limit, there is no $\bot$ subtree, which makes the limit maximal and the function itself well-defined: when given as input a completely defined Rose tree it returns a tree of the same nature.

5 Conclusion and Related Work

We have shown a way to build an CPO from a partially ordered set with least element satisfying a certain assumption, and have shown that the assumption holds for finites trees. The resulting CPO of Rose trees enabled us to extend the *tree* constructor of finite trees to a homonymous function on Rose trees and to enable the possibility of defining a mixed recursive-corecursive function - the *mirror* function - as the unique fixpoint of its functional. This is the most natural way to define such a function and earlier attempts at this in [7] failed. The modified completion operation in this paper gets most of the credit for this progress. Although we have focused in this paper on an example — Rose trees and their mirror — we believe that the example is paradigmatic, and that understanding it opens the way to a general approach, perhaps partially automatable, for naturally defining corecursive functions without using corecursion.

Related Work. The completion operation has similarities with the classical construction of real numbers based on completing rationals with equivalence classes of Cauchy sequences of rationals. However, Cauchy sequences require metric spaces, with a distance function satisfying certain properties, and it does not seem possible to organize Rose trees and their prefix order as a metric space. The reason is that the distance requires a "weak totality" property that the prefix order does not satisfy. Defining corecursive functions based on Cauchy sequences is mentioned in [6]. Using Banach's fixpoint theorem, corecursive functions are defined as unique fixpoints of *eventually contracting* functionals. However, their work is not concerned with extending the definition of corecursive functions beyond the guarded ones.

The fixpoint theorem that we are using is a stronger version of *Kleene's theorem*, stating that continuous functionals over a CPO have a least fixpoint. Another classical result is *completion by ideals* that transforms partial orders with least element into CPOs and monotonic functions between partial orders into continuous functions between CPOs. This textbook result [2](Chapter 1) can be used for defining partial (co)recursive functions in, e.g., the semantics of Haskell. By contrast, we focus on Coq and its total corecursive functions, for which productiveness and unique, maximal fixpoints are essential.

Corecursion is present in several major proof assistants. In Coq, corecursive function definitions have to satisfy a guardedness-by-constructors criterion. In some cases, a function that is not guarded can be transformed into an equivalent, guarded function [3]. Their idea is to use an ad-hoc predicate stating that the definition under study is, in some sense, productive. However, they do not handle the case where corecursive calls are guarded by non-constructor functions, such as our mirror function for Rose trees.

Agda [9] is a proof assistant with an underlying type theory close to that of Coq. It also offers support

for corecursive function definition. In the core tool there is a guardedness checker similar to that of Coq, but somewhat more liberal. Extensions of Agda include sized types [8] that provide users with a uniform, automatic way of handling termination an productiveness. The implementation of sized types is currently unsound (cf. https://github.com/agda/agda/issues/3026).

Isabelle/HOL [10] is also major proof assistant which supports corecursive functions. It accept function that go beyond guarded corecursion [4], provided the functions are *friendly* (a friendly function needs to destruct at most one constructor of input to produce one constructor of output). Unguarded corecursive calls are also accepted, provided they eventually produce a constructor of output. Like in our case, the user needs to prove the conditions ensuring the well-formedness of corecursive functions.

References

[1] : *The Coq Proof Assistant*. https://coq.inria.fr/.

[2] Roberto M. Amadio & Pierre-Louis Curien (1998): *Domains and lambda-calculi*. *Cambridge tracts in theoretical computer science* 46, Cambridge University Press.

[3] Y. Bertot & E. Komendantskaya (2008): *Inductive and Coinductive Components of Corecursive Functions in Coq*. In: *Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science, CMCS 2008, Budapest, Hungary, April 4-6, 2008*, *Electronic Notes in Theoretical Computer Science* 203, pp. 25–47.

[4] J. C. Blanchette, A. Bouzy, A. Lochbihler, A. Popescu & D. Traytel: *Defining Nonprimitively (Co)recursive Functions in Isabelle/HOL*. https://isabelle.in.tum.de/dist/Isabelle2021/doc/corec.pdf.

[5] E. Giménez (1994): *Codifying Guarded Definitions with Recursive Schemes*. In: *Types for Proofs and Programs, International Workshop TYPES'94, Båstad, Sweden, June 6-10, 1994, Selected Papers*, *Lecture Notes in Computer Science* 996, Springer, pp. 39–59.

[6] D. Kozen & N. Ruozzi (2009): *Applications of Metric Coinduction*. *Log. Methods Comput. Sci.* 5(3).

[7] Vlad Rusu & David Nowak (2022): *Defining Corecursive Functions in Coq Using Approximations*. In: *ECOOP*, Berlin, Germany. Available at https://hal.inria.fr/hal-03671876.

[8] N. Veltri & N. van der Weide (2019): *Guarded Recursion in Agda via Sized Types*. In: *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, *LIPIcs* 131, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 32:1–32:19.

[9] : *The Agda Proof Assistant*. https://agda.readthedocs.io.

[10] : *The Isabelle/HOL Proof Assistant*. https://isabelle.in.tum.de/.