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 this proof assistant. 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 with certain extensions - approximations and definedness measures - can be organised as Complete Partial Orders endowed with the same extensions. This makes it possible to define *total* corecursive functions without using Coq's corecursion, as unique solutions of fixpoint equations, 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 [6]. 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 [9] 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 Omega-Complete Partial Order (ωCPO). For some coinductive types such as streams or possibly infinite lists, the ωCPO can be directly defined using Coq's builtin mechanisms, and particular corecursive functions, which do not admit a direct definition in Coq, can be indirectly defined as instances of limits of sequences of approximating functions in the ωCPO in question. But, being nonetheless based on Coq's builtin coinduction features, the ωCPO construction does not work for all coinductive types. In particular, for mixed inductive-coinductve types such as Rose trees the construction of *limits* in the corresponding ωCPO is rejected by Coq as unguarded. The solution we proposed in [9] is to replace the construction of ωCPOs of coinductive types by a completion mechanism,

by which inductive types are completed with equivalence classes of "ascending" sequences of elements in inductive types, which are limits for the sequences in question and play the role of corecursive values.

However, the construction we ended up with in [9] imposes some not-so-practical conditions. The order had to be *weakly total*, thus excluding the natural order induced by the constructors of the inductive types. Although we were able to exhibit a weakly total order for the particular case of Rose tees, 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 partial orders by a notion of approximation of a given precision 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 an ωCPO under the said conditions. This construction is substantially more involved than that in [9], 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. By further eriching the ωCPOs with aapproximations with a measure of "definedness" (inherited from a definedness measure on partial orders with approximations) it becomes to define total corecursive functions (i.e., functions that return "total" values) in the most natural manner, as the unique solution of the fixpoint equation induced by a monotonic, productive functional of the function. We also give practical sufficient conditions ensuring the productiveness requirement.

Outline. Section 2 presents preliminary notions. These include the definition of ωCPOs and a recapp of results from [9] that enable the definition of total corecursive functions as unique fixpoints of certain "productive" functionals involving ωCPOs. We then introduce an abstract notion of approximations of a given precision and some 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 an ωCPO 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 ωCPOs. 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 ωCPO does not require further completion steps.

In Section 4 we give practically usable sufficient conditions for proving the productiveness of functionals, which is required for defining total corecursive functions as unique fixpoints of those functionals. A special attention is given to *maximal* elements in the underlying ωCPO. These are shown to coincide with total coinductive values - the values that a total corecursive function can return. Total values are characterized by having a "definedness" equal to infinity, according to a definedness measure, which we assume on the underlying partial order and its approximations and extend to the ωCPO.

Section 5 applies completion to produce Rose trees from finite trees. The whole approach is in-

stantiated in Section 6 by defiing the *mirror* function as the unique fixpoint of its functional, whose productiveness is proved by using the sufficient conditions in the previous section.

Section 7 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 *An ω-Complete Partial Order (ωCPO) 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 $lim(\!|(s_n)_{n\in\mathbb{N}}|\!)$.

Remark. An important feature of ωCPOs 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 "totally defined". Intermediate, non-maximal elements are therefore "partially defined". In this paper we are interested in defining *total* corecursive functions, which only return totally defined values. Such values play an important role in the rest of the paper.

We now recapp some results from [9] that enable the definition of total corecursive functions as unique fixpoints of "productive" functionals.

Assume an ωCPO $(C, \preceq, \bot)$. We extend $\preceq$ 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 3 *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 4 *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 $\preceq$.*

Remark. The relationship between the productiveness of a functional and the informal notion of productiveness for corecursive functions is established in [9].

Theorem 1 ([9]) *If a functional F is productive then $lim(\!|(F^n \perp\!\!\!\perp)_{n\in\mathbb{N}}|\!)$ is the unique fixpoint of F.*

Remark. Hence, in order to define a total function one needs to (1) organize the codomain C of the function as an ωCPO; (2) prove that the functional of the function in question is productive.

2.2 Partially Ordered Sets with Least Elements and Quantified 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 $\llbracket\cdot\rrbracket^N : C^\circ \to C^\circ$. For any $c \in C^\circ$, $\llbracket c\rrbracket^N$ is called thee* approximation of c having precision N*, and satisfies the following properties:*
   (a) *for all $c \in C^\circ$ and $N \in \mathbb{N}$, $\llbracket c\rrbracket^N \preceq^\circ c$;*
   (b) *for all $c, c' \in C^\circ$ and $N \in \mathbb{N}$, $c \preceq c'$ implies $\llbracket c\rrbracket^N \preceq^\circ \llbracket c'\rrbracket^N$;*
   (c) *for all $c \in C^\circ$ and $N, N' \in \mathbb{N}$, $N \leq N'$ implies $\llbracket c\rrbracket^N \preceq^\circ \llbracket c\rrbracket^{N'}$;*
   (d) *for all $c \in C^\circ$ there exists $N \in \mathbb{N}$ such that $\llbracket c\rrbracket^N = c$;*
   (e) *for all $c \in C^\circ$ and $N, N' \in \mathbb{N}$, $N \leq N'$ implies $\llbracket\llbracket c\rrbracket^{N'}\rrbracket^N = \llbracket c\rrbracket^N$;*
3. *for any $N \in \mathbb{N}$ and any ascending sequence $(s_n)_{n\in\mathbb{N}}$, $(\llbracket s_n\rrbracket^N)_{n\in\mathbb{N}}$ 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 precision, 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 precision $N' \geq N$ and then approximating at precision N amounts to approximating at precision N. Finally, item 3 says that mapping approximations to ascending sequences makes them stabilize - the sequence of approximations cannot grow forever.

Example 1 (model for Assumption 1) *Let C° be the set of trees inductively defined by the rules $\bot \in C^\circ$ and for each list l over T, tree$^\circ$ $l \in T$. Let $\preceq^\circ$ be the prefix relation on C°, inductively defined by $\bot \preceq^\circ t$ for each $t \in C^\circ$, and, for every pair of lists l, l' over T having the same length, say, m, tree$^\circ$ $l \preceq^\circ$ tree$^\circ$ l' whenever $l[i] \preceq^\circ l'[i]$ for each $i < m$ (where $l[i]$ denotes the i-th element of the list l). Let $\llbracket\cdot\rrbracket^N$ ("cut at height N") denote the function $\lambda\, t$. if $N = 0 \vee t = \bot$ then $\bot$ else tree$^\circ$(map($\llbracket\cdot\rrbracket^{N-1}$)(forest$^\circ$ t)) where forest$^\circ$(tree$^\circ$ 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/`.

The $\preceq^\circ$ order and the approximation $\llbracket\cdot\rrbracket^N$ at precision $N \in \mathbb{N}$ can equivalently defined by rewriting, which will be useful ahead in the paper. We thus recall some notions about rewriting. A position *is a string over $\mathbb{N}$, with an associative concatenation operation $\_\cdot\_$ having the empty string ϵ as neutral element. The set of all positions is denoted $\mathbb{N}^*$. Given a tree t, the set $\pi\ t$ of* positions of t *is inductively defined by $\pi\ \bot = \{\epsilon\}$ and $\pi(\text{tree}^\circ[t_0, \cdots, t_{n-1}]) = \bigcup_{i=0}^{n-1}(\pi\ t_i) \cup \bigcup_{i=0}^{n-1}\bigcup_{p\in(\pi\ t_i)}\{i \cdot p\}$. For t a tree and $p \in \pi\ t_i$, the* subtree $t_{|p}$ of t at position p *is defined by $t_{|\epsilon} := t$ and, for all $0 \leq i < n$ and $p \in (\pi\ t_i)$, $(\text{tree}^\circ[t_0, \cdots, t_{n-1}])_{|(i\cdot p)} := (t_i)_{|p}$. A position $p \in \pi\ t$ is a $\bot$-position whenever $t_{|p} = \bot$. The* substutition $t[t']_{|p}$ *of a tree t' at a position p of a tree t is inductively defined by $t[t']_{|\epsilon} = t'$ and, for all $0 \leq i < n$ and $p \in (\pi\ t_i)$, $(\text{tree}[t_0, \ldots, t_i, \ldots, t_{n-1}])t']_{|(i\cdot p)} = \text{tree}^\circ[t_1, \ldots, t_i[t']_{|p}, \ldots, t_{n-1}]$. The* rewriting relation *is defined as follows: two trees t, t' are in the relation if there exists a $\bot$-position $p \in \pi\ t$ and a tree t'' such that $t' = t[t'']_{|p}$. Then, we prove that the $\preceq^\circ$ order is the reflexive-transitive closure of the rewriting relation; and the approximation $\llbracket t\rrbracket^N$ can be defined as substituting $t_{|p}$ with $\bot$ at all positions of length N in t.*

Another observation that will be useful ahead in the paper is related to independent *positions: two positions $p, p' \in \mathbb{N}^*$ are independent if none of them is a prefix of the other one. Given trees t, t_1, t_2 and two independent positions $p_1, p_2 \in \pi\ t$ the following equality holds: $(t[t_1]_{|p_1})[t_2]_{|p_2} = (t[t_2]_{|p_2})[t_1]_{|p_1}$, that is, the substitutions of t_1, t_2 in t at the independent positions p_1, p_2 can be done in any order.*

Refinements of trees as defined in Example 1 include, e.g., terms built over many-sorted first-order signatures and variations thereof [3], providing us with a rich set of examples satisfying the assumptions required for "completing" them to infinite terms and defining corecursive functions over them.

We now proceed with a some technical developments required for the completion operation. The presentation is independent of any model of Assumption 1, hence, it works for all such models.

Remark. Assumption 1 item 2.(b) implies that for increasing sequences $(s_n)_{n\in\mathbb{N}}$, the sequence $(\llbracket s_n \rrbracket^N)_{n\in\mathbb{N}}$ is increasing. Hence, for ascending sequences $(s_n)_{n\in\mathbb{N}}$ the sequence $(\llbracket s_n \rrbracket^N)_{n\in\mathbb{N}}$ is increasing and stabilizing (cf. Item 3 of the Assumption) and therefore has a limit: the value at which the sequence stabilizes.

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.

Another consequence of Assumption 1 is the following lemma.

Lemma 2 *For all $c \in C^\circ$ there exists $N \in \mathbb{N}$ such that for all $N' \geq N$, $\llbracket c \rrbracket^{N'} = c$.*

Proof. N such that $\llbracket c \rrbracket^N = c$ exists by Assumption 1 item 2.(d). Consider any $N' \geq N$. We have $c = \llbracket c \rrbracket^N$ by item 2.(d), hence, $c \preceq^\circ \llbracket c \rrbracket^{N'}$ using item 2.(c), and $\llbracket c \rrbracket^{N'} \preceq^\preceq c$ by item 2.(a). Using the antisymmetry of the $\preceq^\circ$ relation we obtain that for the arbitrarily chosen $N' \geq N$, $\llbracket c \rrbracket^{N'} = c$, 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 5 *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 $j \geq i$, $\llbracket s_j \rrbracket^N = \llbracket s'_j \rrbracket^N$.*

Lemma 3 *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 $j \geq i$, $\llbracket s_j \rrbracket^N = \llbracket s'_j \rrbracket^N$, 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 $j \geq i'$, $\llbracket s'_j \rrbracket^N = \llbracket s''_j \rrbracket^N$. Setting $i'' := max\ i\ i'$ we get: for all $j \geq i''$, $\llbracket s_j \rrbracket^N = \llbracket s'_j \rrbracket^N = \llbracket s''_j \rrbracket^N$; which proves $(s_n)_{n\in\mathbb{N}} \sim (s''_n)_{n\in\mathbb{N}}$ and establishes transitivity.

Lemma 4 *For an increasing sequence $(s_n)_{n\in\mathbb{N}}$, let s be the (unique) value at which $(\llbracket s_n \rrbracket^N)_{n\in\mathbb{N}}$ stabilizes. If $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ then $(\llbracket s'_n \rrbracket^N)_{n\in\mathbb{N}}$ also stabilizes at s.*

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

Lemma 4 helps defining approximations of equivalence classes $\llbracket [(s_n)_{n\in\mathbb{N}}]_\sim \rrbracket^N$ of ascending sequences:

Definition 6 *For any ascending sequence over C°, $\llbracket [(s_n)_{n\in\mathbb{N}}]_\sim \rrbracket^N = lim(\!|(\llbracket s_n \rrbracket^N)_{n\in\mathbb{N}}|\!)$.*

Indeed, Lemma 4 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 5 *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 $N \in \mathbb{N}$, $\lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N = \lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N$.*

Proof. ($\Rightarrow$): per the above remarks, $\lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N = lim(\!|(\lceil s_n\rceil^N)_{n\in\mathbb{N}}|\!)$ and $\lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N = lim(\!|(\lceil s'_n\rceil^N)_{n\in\mathbb{N}}|\!)$. If $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$ we know by Lemma 4 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, $\lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N = \lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N$.

($\Leftarrow$) : Choose an arbitrary $N \in \mathbb{N}$. Per the above remarks, from the hypothesis $\lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N = \lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N$ 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 5, $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$; 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 7 *For $k,k' \in K$, $k \precsim k'$ whenever for all $N \in \mathbb{N}$, $\lceil k\rceil^N \preceq^\circ \lceil k'\rceil^N$.*

Lemma 6 *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 $N \in \mathbb{N}$, $\lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N \preceq^\circ \lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N$, and $k' \precsim k$ means that for all $N \in \mathbb{N}$, $\lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N \preceq^\circ \lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N$. As $\preceq^\circ$ is antisymmetric, for all $N \in \mathbb{N}$, $\lceil\lceil[(s_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N = \lceil\lceil[(s'_n)_{n\in\mathbb{N}}]_\sim\rceil\rceil^N$, and by Lemma 5, $(s_n)_{n\in\mathbb{N}} \sim (s'_n)_{n\in\mathbb{N}}$, 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 8 below. To prove that lemma we need the monotonicity of approximations of equivalence classes:

Lemma 7 *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 8 *$k \precsim k'$ 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 7 $k \precsim k'$ 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 7, $\lceil k'\rceil^{N'} \preceq^\circ \lceil k'\rceil^N$ 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 7 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 an ωCPO $(C, \preceq, \bot)$. We have already noted that increasing and stabilizing sequences have limits. Hence, limits have to be constructed for the ascending sequences.

Definition 8 *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 $c \preceq^\circ \llbracket k \rrbracket^N$;
   (c) for $k, k' \in K$, $k \preceq k'$ whenever $k \precsim k'$.

Remark. Definition 8 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 ωCPO construction does not require further completion steps.

A key difference with the earlier completion operation in [9] 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 "totally 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 an ωCPO.

Lemma 9 *In the context of Definition 8, 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 8.

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 8 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 8 item 3.(b), the latter means there exists $N \in \mathbb{N}$ such that $c' \preceq^\circ \llbracket k \rrbracket^N$, and since $c \preceq c'$, the same N ensures $c \preceq^\circ \llbracket k \rrbracket^N$ 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 8 item 3.(b), there exists $N \in \mathbb{N}$ such that $c \preceq^\circ \llbracket k \rrbracket^N$, and by Definition 7 and item 3.(c), $\llbracket k \rrbracket^N \preceq^\circ \llbracket k' \rrbracket^N$, thus, by transitivity, there exists $N \in \mathbb{N}$ such that $c \preceq^\circ \llbracket k' \rrbracket^N$, meaning that $c \preceq k'$;
4. $k, k', k'' \in K$ with $k \preceq k'$ and $k' \preceq k''$: by Definition 8 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 10 *In the context of Definition 8, 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 8, 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 $s_n = \lceil s_n \rceil^N \preceq^\circ lim(\!|(\lceil s_n \rceil^N)_{n\in N}|\!) = \lceil\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil^N$. Hence, $s_n \preceq^\circ \lceil\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil^N$ and, by Definition 8 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 $lim(\!|(\lceil s_n \rceil^N)_{n\in\mathbb{N}}|\!) = \lceil\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil^N$. We thus obtain that for all $N\in\mathbb{N}$, there exists $N'\in\mathbb{N}$ such that $\lceil\lceil [(s_n)_{n\in\mathbb{N}}]_\sim \rceil^N \preceq^\circ \lceil [(s'_n)_{n\in N}]_\sim \rceil^{N'}$. By Lemma 8 we obtain $[(s_n)_{n\in\mathbb{N}}]_\sim \precsim [(s'_n)_{n\in\mathbb{N}}]_\sim$. Then, by Definition 8 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 = ωCPO

In this section we show how to "finish" the completion operation in the previous section in order to obtain an ωCPO. 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 11

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 8 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 8 item 3.(c), $k\preceq k'$ is just $k\precsim k'$, which by Definition 7 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 7.

The next lemma establishes that ascending sequences are similar to any of their subsequences.

Lemma 12 *Consider an ascending sequence $(s_n)_{n\in\mathbb{N}}$ and a subsequence $(s_{(\varphi\ n)})_{n\in\mathbb{N}}$ of it, for some strictly increasing $\varphi:\mathbb{N}\to\mathbb{N}$. Then $(s_n)_{n\in\mathbb{N}} \sim (s_{(\varphi\ n)})_{n\in\mathbb{N}}$.*

Proof. As a subsequence of an ascending sequence, $(s_{(\varphi\ n)})_{n\in\mathbb{N}}$ is ascending. Let $k = [(s_n)_{n\in\mathbb{N}}]_\sim$ and $k' = [(s_{(\varphi\ n)})_{n\in\mathbb{N}}]_\sim$. We observe that, by Lemma 8, showing $k = k'$ is equivalent to finding, for each $N\in\mathbb{N}$, some $N'\in\mathbb{N}$ such that $\lceil k\rceil^N = \lceil k'\rceil^{N'}$. Let $N' := N$. Using Definition 6, we have to show that $lim(\!|(\lceil s_n\rceil^N)_{n\in\mathbb{N}}|\!) = lim(\!|(\lceil s_{(\varphi\ n)}\rceil^N)_{n\in\mathbb{N}}|\!)$. Let n_0 be the least natural number such that $lim(\!|(\lceil s_n\rceil^N)_{n\in\mathbb{N}}|\!) = \lceil s_{n_0}\rceil^N$, and let $n_1\in\mathbb{N}$ be such that $lim(\!|(\lceil s_{(\varphi\ n)}\rceil^N)_{n\in\mathbb{N}}|\!) = \lceil s_{n_1}\rceil^N$. $n_1 < n_0$ contradicts the minimality of n_0, hence, $n_1 \geq n_0$, which implies $lim(\!|(\lceil s_n\rceil^N)_{n\in\mathbb{N}}|\!) = \lceil s_{n_1}\rceil^N = lim(\!|(\lceil s_{(\varphi\ n)}\rceil^N)_{n\in\mathbb{N}}|\!)$ and proves the lemma.

We next introduce a notion of "diagonalization" for an equivalence class of ascending sequences.

Definition 9 *For $k\in K$, the sequence $(\lceil k\rceil^n)_{n\in\mathbb{N}}$ is called the diagonal of k.*

The main interest of diagonalization is that it gives a "canonical" representative for the class.

Lemma 13 *For all $k\in K$, the sequence $(\lceil k\rceil^n)_{n\in\mathbb{N}}$ is ascending, and $lim(\!|(\lceil k\rceil^n)_{n\in\mathbb{N}}|\!) = k$.*

Proof. We first prove the following technical result about ascending sequences: (†) *if $(s_m)_{m\in\mathbb{N}}$ is an ascending sequence, then for all $j,n\in\mathbb{N}$ there exist $j'>n, n'>n$ such that $\lceil s_j\rceil^n \prec^\circ \lceil s_{j'}\rceil^{n'}$.*

Proof of (†): there exists $j'>k$ such that $s_j \prec^\circ s_{j'}$. Moreover, for large enough n' (in particular, $n' > Nn$) $\lceil s_{j'}\rceil^{n'} = s'_{j'}$ (by Assumption 1, items 2(d)(c)(a)). Hence, $\lceil s_j\rceil^n \preceq^\circ s_j \prec^\circ s'_{j'} = \lceil s_{j'}\rceil^{n'}$, i.e., (†).

We now prove that $(\lceil k\rceil^n)_{n\in\mathbb{N}}$ is ascending. By Lemma 11 item 3 it is increasing. Choose an ascending sequence $(s_n)_{n\in\mathbb{N}}\in k$. Fix an arbitrary $n\in\mathbb{N}$. It holds that $\lceil k\rceil^n = lim(\!|(\lceil s_m\rceil^n)_{m\in\mathbb{N}}|\!)$. Now, $lim(\!|(\lceil s_m\rceil^n)_{m\in\mathbb{N}}|\!) = \lceil s_j\rceil^n$ for some $j\in\mathbb{N}$. Using (†) there exists $j'>j$ and $n'>n$ such that $\lceil s_j\rceil^n \prec^\circ \lceil s_{j'}\rceil^{n'}$. But $\lceil s_{j'}\rceil^{n'} \preceq^\circ lim(\!|(\lceil s_m\rceil^{n'})_{m\in\mathbb{N}}|\!)$. Overall, for the arbitrarily chosen $n\in\mathbb{N}$, there exists $n'>n$ such that $\lceil k\rceil^n = lim(\!|(\lceil s_m\rceil^n)_{m\in\mathbb{N}}|\!) \prec^\circ lim(\!|(\lceil s_m\rceil^{n'})_{m\in\mathbb{N}}|\!) = \lceil k\rceil^{n'}$, which proves that $(\lceil k\rceil^n)_{n\in\mathbb{N}}$ is ascending.

Finally, we prove that $lim(\!|(\lceil k\rceil^n)_{n\in\mathbb{N}}|\!) = k$, i.e., k is the least upper bound of $(\lceil k\rceil^n)_{n\in\mathbb{N}}$. By Lemma 11 item 1, k is an upper bound of $(\lceil k\rceil^n)_{n\in\mathbb{N}}$. We now prove it is the least upper bound. For, assuming an upper bound $k'\in C^\circ\cup K$, from Assumption 1 item 1 and the ascending property of $(\lceil k\rceil^n)_{n\in\mathbb{N}}$ we know that $k'\in C^\circ$ is impossible, hence, $k'\in K$. Now, since k' is an upper bound, for all $n\in\mathbb{N}$, $\lceil k\rceil^n \precsim k'$. Choose an arbitrary $n\in\mathbb{N}$. Since $\lceil k\rceil^n\in C^\circ$ and $k'\in K$, by Definition 8 item 3 (b), there exists $n'\in\mathbb{N}$ such that $\lceil k\rceil^n \preceq^\circ \lceil k\rceil^{n'}$. By Lemma 8 we obtain $k\precsim k'$, which shows $lim(\!|(\lceil k\rceil^n)_{n\in\mathbb{N}}|\!) = k$ and completes the proof.

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 also call *diagonal* of $s\in(C^\circ\cup K)^\omega$ the sequence $(\lceil s_n\rceil^n)_{n\in\mathbb{N}}\in(C^\circ)^\omega$.

Lemma 14 *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 11 items 2&3.

Lemma 15 *For all $k,k'\in K$, $k\prec k'$ implies that there exists $j\in\mathbb{N}$ such that for all $j'\geq j$, $\lceil k\rceil^{j'} \prec^\circ \lceil k'\rceil^{j'}$.*

Proof. $k\prec k'$ implies in particular $k\precsim k'$. Assuming the negation of the conclusion, for each $j\in\mathbb{N}$ there exists $j'>j$ such that $\lceil k\rceil^{j'} \not\prec^\circ \lceil k'\rceil^{j'}$. But $k\precsim k'$ implies $\lceil k\rceil^{j'} \preceq^\circ \lceil k'\rceil^{j'}$, hence, for each $j\in\mathbb{N}$ there exists $j'>j$ such that $\lceil k\rceil^{j'} = \lceil k'\rceil^{j'}$, which means that there exists a strictly increasing function $\varphi:\mathbb{N}\to\mathbb{N}$ such that for all $j\in\mathbb{N}$, $\lceil k\rceil^{(\varphi\ j)} = \lceil k'\rceil^{(\varphi\ j)}$, i.e., $(\lceil k\rceil^{(\varphi\ j)})_{j\in\mathbb{N}} = (\lceil k'\rceil^{(\varphi\ j)})_{j\in\mathbb{N}}$. But $(\lceil k\rceil^{(\varphi\ j)})_{j\in\mathbb{N}}$

is a subsequence of the diagonal $(\llbracket k \rrbracket^j)_{j\in\mathbb{N}}$ of k, an ascending sequence according to Lemma 13, hence, by Lemma12, $(\llbracket k \rrbracket^j)_{j\in\mathbb{N}} \sim (\llbracket k \rrbracket^{(\varphi\ j)})_{j\in\mathbb{N}}$, and, again using Lemma 13, $k = lim(\!|(\llbracket k \rrbracket^{(\varphi\ j)})_{j\in\mathbb{N}}|\!)$. Similarly, $k' = lim(\!|(\llbracket k' \rrbracket^{(\varphi\ j)})_{j\in\mathbb{N}}|\!)$, i.e., $k = k'$, in contradiction with the hypothesis $k \prec k'$. The contradiction arises from negating the conclusion of the lemma, hence, the lemma is proved.

Lemma 16 *Consider a strictly increasing sequence $(k_n)_{n\in\mathbb{N}} \in K^\omega$. Then $(\llbracket k_n \rrbracket^n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$ is ascending.*

Proof. We build by induction a strictly increasing subsequence $(\llbracket k_{i_n} \rrbracket^{i_n})_{n\in\mathbb{N}}$ of $(\llbracket k_n \rrbracket^n)_{n\in\mathbb{N}}$. For the base case the subsequence is $\llbracket k_0 \rrbracket^0$. Assume now the subsequence of interest has been built up to a given length, hence, its last element is of the form $\llbracket k_{i_j} \rrbracket^{i_j}$. We have, by the strictly increasing nature of $(k_n)_{n\in\mathbb{N}}$, $k_{i_j} \prec k_{i_j+1}$, hence, using Lemma 11 item 1, $\llbracket k_{i_j} \rrbracket^{i_j} \prec k_{i_j+1}$. Using Lemma 15 and Assumption 1 items 2.(d) and 2.(c), we can find a $i' > i_j + 1$ which is large enough to ensure $\llbracket \llbracket k_{i_j} \rrbracket^{i_j} \rrbracket^{i'} = \llbracket k_{i_j} \rrbracket^{i_j}$, such that $\llbracket \llbracket k_{i_j} \rrbracket^{i_j} \rrbracket^{i'} \prec^\circ \llbracket k_{i_j+1} \rrbracket^{i'}$. Since $i' > i_j + 1$, by monotonicity of $(k_n)_{n\in\mathbb{N}}$, we obtain using Lemma 11 item 2, $\llbracket k_{i_j+1} \rrbracket^{i'} \preceq \llbracket k_{i'} \rrbracket^{i'}$, i.e., $\llbracket k_{i_j+1} \rrbracket^{i'} \preceq^\circ \llbracket k_{i'} \rrbracket^{i'}$ since the elements being compared are in C°. Hence, we have obtained $i' > i_j$ such that $\llbracket k_{i_j} \rrbracket^{i_j} \prec^\circ \llbracket k_{i'} \rrbracket^{i'}$, and therefore have built the strictly increasing subsequence of interest up to length $j+1$. An infinite strictly increasing subsequence $(\llbracket k_{i_n} \rrbracket^{i_n})_{n\in\mathbb{N}}$ of $(\llbracket k_n \rrbracket^n)_{n\in\mathbb{N}}$ is (obviously) defined as soon as all its elements are defined. In order to define its j-th element, we build the finite strictly increasing subsequence by induction as above up to length j and take its last element.

Lemma 17 *If $s \in (C^\circ \cup K)^\omega$ is ascending and there is $i \in \mathbb{N}$ such that $s_i \in K$, then $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$ is ascending.*

Proof. By Lemma 14 the sequence $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$ is increasing. We only need to prove that it has a strictly ascending 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 a strictly increasing subsequence, $(s_{i_n})_{n\in\mathbb{N}, i_n\geq i}$ is a strictly increasing sequence, and belongs to K^ω. We apply Lemma 16 to it and get an ascending subsequence of $(s_{i_n})_{n\in\mathbb{N}, i_n\geq i}$, which by transitivity of the subsequence relation is also an ascending subsequence of the original sequence s in the hypothesis.

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

Proof. We first prove that $[(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$ is an upper bound for s. Thus, we need to show that for all $j \in \mathbb{N}$, $s_j \preceq [(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$. 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 8 we only need to show (†): *for all $N \in \mathbb{N}$, there exists $N' \in \mathbb{N}$ such that* $\llbracket s_j \rrbracket^N \preceq^\circ \llbracket [(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim \rrbracket^{N'}$. Choose an arbitrary $N \in \mathbb{N}$. Let $p = max\ j\ N$. Thus, by Assumption 1 item 2.(b), $\llbracket s_j \rrbracket^N \preceq^\circ \llbracket s_p \rrbracket^N$. Using Assumption 1 item 2.(c), $\llbracket s_p \rrbracket^N \preceq^\circ \llbracket s_p \rrbracket^p$. By transitivity, $\llbracket s_j \rrbracket^N \preceq^\circ \llbracket s_p \rrbracket^p$. But since (by Lemma 17) $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$ is ascending, we know by Lemma 10 that $[(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$ is an upper bound for $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$. In particular, $\llbracket s_p \rrbracket^p \preceq^\circ [(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$, and by transitivity, $\llbracket s_j \rrbracket^N \preceq^\circ [(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$, which by Def. 8 item 3.(b) implies that there is $N' \in \mathbb{N}$ such that $\llbracket s_j \rrbracket^N \preceq^\circ \llbracket [(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim \rrbracket^{N'}$, i.e., (†).

We now prove that $[(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$ 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 $[(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim \preceq b$. Now, by Lemma 10, $[(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim$ is the least upper bound for the ascending sequence $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$. Hence in order to show $[(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}]_\sim \preceq b$ one only has to show that b is an upper bound for $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$. Choose an arbitrary $n \in \mathbb{N}$. We have, by Lemma 11 item 1, $\llbracket s_n \rrbracket^n \preceq s_n$ and since b is an upper bound for s, $s_n \preceq b$. Hence, for an arbitrary $n \in \mathbb{N}$, $\llbracket s_n \rrbracket^n \preceq b$, which establishes that b is an upper bound for $(\llbracket s_n \rrbracket^n)_{n\in\mathbb{N}}$, which completes the proof of the lemma.

This concludes the construction of the ωCPO by completion.

4 Proving Productiveness

As already noted in Section 2 obtaining an ωCPO is only the first step step towords our main goal - defining total corecursive functions by unique fixpoints of their *productive* functionals by means of Theorem 1. The second step is to prove the productiveness requirement for given functionals. As visible in Definition 4, productiveness is concerned with certain sequences converging towards *maximal* elements.

4.1 Maximal vs. Totally Defined Elements

Maximality is easy to define, but it is not easy to prove in practice, especially for equivalence classes denoting infinite elements in an ωCPO obtained by completion. In this section we provide an equivalent notion to maximality, called *total definability*, which is easier to use in practice because it is "computational" in nature. It is based on a "definedness" function which takes values in $\mathbb{N} \cup \{\infty\}$, a set in which one can perform computations: approximations, limits of sequences of numbers, etc. The definedness function is expected to satisfy some assumptions, stated below and justified after the statement.

Assumption 2 *We assume a* definedness *function $\delta : C^\circ \to \mathbb{N} \cup \{\infty\}$ satisfying the following properties:*

1. *for all $c, c' \in C^\circ$, $c \preceq^\circ c'$ implies $\delta\ c \leq \delta\ c'$;*
2. *for all $c \in C^\circ$, if $\delta\ c = \infty$ then c is maximal w.r.t. $\preceq^\circ$;*
3. *for all $c \in C^\circ$ and $N \in \mathbb{N}$, if $\delta\ c \geq N$ then $\delta\ (\llbracket c \rrbracket^N) = N$;*
4. *for all $c, c' \in C^\circ$ and $N \in \mathbb{N}$, if $c \preceq^\circ c'$ and $\delta\ c > N$ then $\llbracket c \rrbracket^N = \llbracket c' \rrbracket^N$;*
5. *for all ascending sequences $(s_n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$ and $N \in \mathbb{N}$ such that for all $n \in \mathbb{N}$, $\delta\ s_n \leq N$, there exists $i \in \mathbb{N}$ and an increasing sequence $(s'_j)_{j\geq i}$ such that for all $j \geq i$, $s_j \preceq^\circ s'_j$ and $N < \delta\ s'_j$.*

If $\delta\ c = n$ then we say that c is defined up to *n. If c is defined up to ∞ we also say that c is totally defined.*

A Model for Assumption 2 Like Assumption 1, Assumption 2 has a model in the set of tree defined in Example 1. Here, $\delta\ c$ is either: a natural number being *the minimal length of a $\bot$-position in c*, if c has such positions; or, c has no $\bot$-positions, $\delta\ c = \infty$. With the convention that the minimum of an empty list over $\mathbb{N} \cup \{\infty\}$ is ∞, i.e., $\min\ [] = \infty$, δ can be recursively defined as follows:

$\delta\ c = \mathit{if}\ c = \bot\ \mathit{then}\ 0\ \mathit{else}\ 1 + \min(\mathit{map}\ \delta\ (\mathit{forest}^\circ\ c))$.

We now justify the assumptions on the δ function.

1. *for all $c, c' \in C^\circ$, $c \preceq^\circ c'$ implies $\delta\ c \leq \delta\ c$*: This is monotonicity of δ. if $c \preceq^\circ c'$ then either $c = c'$, in which case $\delta\ c = \delta\ c'$, or c' is obtained from c' by rewriting some $\bot$ subtrees by non-$\bot$ ones. If c and c' still have at least one, common, minimal $\bot$ position then $\delta\ c' = \delta\ c$. Otherwise, due to rewriting, the $\bot$ positions of c' (if any) are strictly longer then those of c, which implies $\delta\ c < \delta\ c'$.
2. *for all $c \in C^\circ$, if $\delta\ c = \infty$ then c is maximal w.r.t. $\preceq^\circ$*: $\delta\ c = \infty$ means that c has no $\bot$-positions, i.e., it does not allow rewriting. If c were non-maximal there would be c' with $c \prec^\circ c'$, meaning rewriting would be possible from c. But we have just seen that it is not. Hence, c is maximal.
3. *for all $c \in C^\circ$ and $N \in \mathbb{N}$, if $\delta\ c \geq N$ then $\delta\ (\llbracket c \rrbracket^N) = N$*: $\delta\ c \geq N$ means that the are no $\bot$-positions of c of length $< N$. Now, the approximation $\llbracket c \rrbracket^N$ replaces all trees at positions of length N in c by $\bot$. These all the occurences of $\bot$ in $\llbracket c \rrbracket^N$, and their positions all have length N, i.e., $\delta\ (\llbracket c \rrbracket^N) = N$.

4. *for all $c, c' \in C^\circ$ and $m \in \mathbb{N}$, if $c \preceq^\circ c'$ and $\delta\ c > N$ then $[\![c]\!]^N = [\![c']\!]^N$*: here, all the $\bot$-positions of c, c' are of length $> N$, and c' is obtained from c by rewriting in some (possiby, in zero) positions of length $> N$. Hence, this does not affect positions of length $\leq N$. In particular, for all p of length $\leq N$, p is a position of c iff it is a position of c', and if this is the case then $c_{|p} = c'_{|p}$. Now, the approximation of precision N replaces all trees at positions of length N by $\bot$. After performing the approximations, it is still the case that for all positions p of length $\leq N$, p is a position of $[\![c]\!]^N$ iff it is a position of $[\![c']\!]^N$, and if this is the case then $([\![c]\!]^N)_{|p} = ([\![c']\!]^N)_{|p}$; which implies $[\![c]\!]^N = [\![c']\!]^N$.
5. *for all ascending sequences $(s_n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$ and $N \in \mathbb{N}$ such that for all $n \in \mathbb{N}$, $\delta\ s_n \leq N$, there exists $i \in \mathbb{N}$ and an increasing sequence $(s'_j)_{j\geq i}$ such that for all $j \geq i$, $s_j \preceq^\circ s'_j$ and $N < \delta\ s'_j$*: we shall need to further elaborate on the definition of $\preceq^\circ$ by rewriting: for $c, c' \in C^\circ$ and l a list of positions, we inductively define the statement $c \preceq^\circ_l c'$ by $c \preceq^\circ_{[]} c$ and $c \preceq^\circ_{(p;l)} c'$ if there exist c'' s.t. $(c[c'']_p) \preceq^\circ_l c'$, where "$p; l$" denotes adding the element p at the head of the list l. We prove that $c \preceq^\circ c'$ holds iff there exists a l of $\bot$-positions of c s.t. $c \preceq^\circ_l c'$; and if the positions in l are mutually independent from those in l' and $c \preceq^\circ_{l'} c'$ and $c \preceq^\circ_{l''} c''$ then there exists c'''' such that $c' \preceq^\circ_{l''} c'''$ and $c'' \preceq^\circ_{l'} c''''$.

   Back to the statement of interest, since $(s_n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$ is ascending and for all $n \in \mathbb{N}$, $\delta\ s_n \leq N$, there must exist a first index, say, i, after which all rewritings that generate the subsequent elements of the sequence $(s_j)_{j\geq i}$ occur at non-minimal $\bot$-positions. Otherwise, the minimal $\bot$-positions are rewritten infinitely many times and eventually their length becomes strictly greater than N, in contradiction with the hypothesis *for all $n \in \mathbb{N}$, $\delta\ s_n \leq N$*. We have the index i and, by the manner in which it was obtained,, for all $j \geq i$ $s_j \preceq^\circ_{l_j} s_{j+1}$ for a list l_j of non-minimal $\bot$-positions of s_j.

   We now inductively build an increasing sequence $(s'_j)_{j\geq i}$ such that $N < \delta\ s'_j$ for all $j \geq i$ and (more than what the current statement requires) there is a list l' of positions such that for all $j \geq i$, $s_j \preceq^\circ_{l'} s'_j$. In the base case, let l' be the list of all minimal $\bot$-positions of s_i. By replacing $\bot$ at all positions in l' with sufficiently tall trees, we obtain $s_i \prec^\circ_{l'} s'_i$ with $N < \delta\ s'_i$, which concludes this case.

   Assume now the sequence $(s'_j)_{j\geq i}$ has been built up to some $j \geq i$. From the induction hypothesis, $s_j \preceq^\circ_{l'} s'_j$ and $N < \delta\ s'_j$. We have also obtained above that $s_j \preceq^\circ_{l_j} s_{j+1}$ holds for a list l_j of non-minimal positions of s_j. Hence, the positions in l' and l_j are mutually independent. Thus, there exists s'_{j+1} with $s'_j \preceq^\circ_{l_j} s'_{j+1}$ and $s_{j+1} \preceq^\circ_{l'} s'_{j+1}$. Moreover, $N < \delta\ s'_{j+1}$ (which results from $N < \delta\ s'_j$ and mononotonicity of δ). This completes the induction step and the construction of $(s'_j)_{j\geq i}$ with the expected properties in the current statement. The model for Assumption 2 is also completed.

The remaining results in this section only rely on Assumption 1 and 2. They hold for any model of the assumption, including the trees from Example 1 and as well as terms over various kinds of signatures.

Lemma 19 *For all $c \in C^\circ$, c is totally defined if and only if c is maximal w.r.t. $\preceq^\circ$.*

Proof. ($\Rightarrow$): Suppose $\delta\ c = \infty$ but c is not maximal, i.e., there exists $c' \in C^\circ$ such that $c \prec^\circ c'$. Then, by the monotonicity of δ - item 1 in Assumption 2, $\delta\ c' = \infty$. Using Assumption 2 item 4, for all $N \in \mathbb{N}$, $[\![c]\!]^N = [\![c']\!]^N$. By choosing N large enough, using Lemma 2, we obtain $[\![c]\!]^N = c$ and $[\![c']\!]^m = c'$, meaning that $c = c'$, in contradiction to $c \prec^\circ c'$. The contradiction occurs due to the negation of the maximality of c; hence, the maximality holds. ($\Leftarrow$): this is just Assumption 2 item 2.

Definition 10 *The definedness function δ from Assumption 2 is extended over K by $\delta\ k = lim(\!|(\delta([\![k]\!]^m)_{m\in\mathbb{N}})|\!)$. The notions of definedness up to some $n \in \mathbb{N} \cup \{\infty\}$ and total definedness are extended to K as well.*

Remark. The limit (a.k.a. least upper bound) of a sequence of elements in $\mathbb{N} \cup \{\infty\}$ always exists, so referring to it it Definition 10 is legitimate.

Lemma 20 *For $k,k' \in K$, if $k \preceq k'$ then $\delta\ k \leq \delta\ k'$.*

Proof. From $k \preceq k'$ and Lemma 11 item 2, for all $n \in \mathbb{N}$, $\llbracket k \rrbracket^n \preceq \llbracket k' \rrbracket^n$, i.e., $\llbracket k \rrbracket^n \preceq^\circ \llbracket k' \rrbracket^n$, hence, by Assumption 2 item 1, for all $n \in \mathbb{N}$, $\delta\ (\llbracket k \rrbracket^n) \leq \delta\ (\llbracket k' \rrbracket^n)$. Thus, $lim(\!|(\delta(\llbracket k \rrbracket^n)_{n\in\mathbb{N}})|\!) \leq lim(\!|(\delta(\llbracket k' \rrbracket^n)_{n\in\mathbb{N}})|\!)$, i.e., $\delta\ k \leq \delta\ k'$.

Lemma 21 *For $k \in K$ and $n,m \in \mathbb{N}$, $n \geq m$ implies $\llbracket k \rrbracket^m = \llbracket \llbracket k \rrbracket^n \rrbracket^m$.*

Proof. Choose an arbitrary ascending sequence $(s_j)_{j\in\mathbb{N}} \in k$. Choose arbitrarily $n,m \in \mathbb{N}$ such that $n \geq m$. Then, by Definition 6, $\llbracket k \rrbracket^m = lim(\!|(\llbracket s_j \rrbracket^m)_{j\in\mathbb{N}}|\!)$ and $\llbracket k \rrbracket^n = lim(\!|(\llbracket s_j \rrbracket^n)_{j\in\mathbb{N}}|\!)$, hence, $\llbracket \llbracket k \rrbracket^n \rrbracket^m = \llbracket lim(\!|(\llbracket s_j \rrbracket^n)_{j\in\mathbb{N}}|\!) \rrbracket^m = lim(\!|(\llbracket \llbracket s_j \rrbracket^n \rrbracket^m)_{j\in\mathbb{N}}|\!)$. Using Assumption 1 item 2.(e), since $n \geq m$, $\llbracket \llbracket s_j \rrbracket^n \rrbracket^m = \llbracket s_j \rrbracket^m$, hence, $lim(\!|(\llbracket \llbracket s_j \rrbracket^n \rrbracket^m)_{j\in\mathbb{N}}|\!) = lim(\!|(\llbracket s_j \rrbracket^m)_{j\in\mathbb{N}}|\!) = \llbracket k \rrbracket^m$. That is, we have obtained $\llbracket \llbracket k \rrbracket^n \rrbracket^m = \llbracket k \rrbracket^m$ for arbitrarily chosen $n,m \in \mathbb{N}$ such that $n \geq m$, which proves the lemma.

Lemma 22 *For all $k \in K$, if $k \preceq k'$ and $\delta\ k = \infty$ then for all $m \in \mathbb{N}$, $\llbracket k \rrbracket^m = \llbracket k' \rrbracket^m$.*

Proof. From $k \preceq k'$ and $\delta\ k = \infty$ we obtain using Lemma 20 that $\delta\ k' = \infty$. We note that in the diagonal $(\llbracket k \rrbracket^n)_{n\in\mathbb{N}}$ of k no element can be totally defined; otherwise by Lemma 19 that element would be maximal, in contradiction with the ascending nature of $(\llbracket k \rrbracket^n)_{n\in\mathbb{N}}$ established in Lemma 13. The same reasoning holds for the diagonal of k'. Hence, for all $n \in \mathbb{N}$, $\delta(\llbracket k \rrbracket)_n \in \mathbb{N}$ and $\delta(\llbracket k' \rrbracket)_n \in \mathbb{N}$.

Fix an arbitrary $m \in \mathbb{N}$. Since $\delta\ k = \delta\ k' = \infty$, there exists n such that $n \geq m$, $\delta\ (\llbracket k \rrbracket)_n > m$. Let $c = \llbracket k \rrbracket^n$ and $c' = \llbracket k' \rrbracket^n$. Then, $c,c' \in C^\circ$, and $\delta\ c > m$. Moreover, from $k \preceq k'$ we obtain using Lemma 11 item 2, $\llbracket k \rrbracket^n \preceq \llbracket k' \rrbracket^n$, hence, $c \preceq c'$, i.e., $c \preceq^\circ c'$. Using Assumption 2 item 4 we obtain $\llbracket c \rrbracket^m = \llbracket c' \rrbracket^m$, thus, $\llbracket \llbracket k \rrbracket^n \rrbracket^m = \llbracket \llbracket k' \rrbracket^n \rrbracket^m$. Since $n \geq m$, using Lemma 21 we obtain $\llbracket k \rrbracket^m = \llbracket \llbracket k \rrbracket^n \rrbracket^m$ and $\llbracket k' \rrbracket^m = \llbracket \llbracket k' \rrbracket^n \rrbracket^m$, i.e., for the arbitrarily chosen $m \in \mathbb{N}$, $\llbracket k \rrbracket^m = \llbracket k' \rrbracket^m$, which proves the lemma.

Lemma 23 *For sequences $(s_n)_{n\in\mathbb{N}}$ and $(s'_n)_{n\in\mathbb{N}}$ in $(C^\circ)^\omega$, if $(s_n)_{n\in\mathbb{N}}$ is ascending, $(s'_n)_{n\in\mathbb{N}}$ is increasing, and for all $n \in \mathbb{N}$, $s_n \preceq s'_n$, then $(s'_n)_{n\in\mathbb{N}}$ is ascending and $[(s_n)_{n\in\mathbb{N}}]_\sim \preceq [(s'_n)_{n\in\mathbb{N}}]_\sim$.*

Proof. Assume $(s'_n)_{n\in\mathbb{N}}$ is stabilizing to some value $c \in C^\circ$. Then for all $n \in \mathbb{N}$, $s_n \preceq^\circ s'_n \preceq^\circ c$, and since $(s_n)_{n\in\mathbb{N}}$ is ascending this contradicts Assumption 1 item 1. Hence, $(s'_n)_{n\in\mathbb{N}}$ is ascending. Let $k = [(s_n)_{n\in\mathbb{N}}]_\sim$ and $k' = [(s'_n)_{n\in\mathbb{N}}]_\sim$. We have to prove $k \preceq k'$, i.e., $k \precsim k'$ following Definition 8 item 3.(c). By Definition 7 this amounts to showing that for all $N \in \mathbb{N}$, $\llbracket k_n \rrbracket^N \preceq^\circ \llbracket k'_n \rrbracket^N$. Choose an arbitrary $N \in \mathbb{N}$. By Definition 6, $\llbracket k_n \rrbracket^N \preceq^\circ \llbracket k'_n \rrbracket^N$ amounts to proving $lim(\!|(\llbracket s_n \rrbracket^N)_{n\in\mathbb{N}}|\!) \preceq^\circ lim(\!|(\llbracket s'_n \rrbracket^N)_{n\in\mathbb{N}}|\!)$. Thanks to the hypothesis *for all $n \in \mathbb{N}$, $s_n \preceq^\circ s'_n$*, by using Assumption 1 item 2.(b) we obtain that for all $n \in \mathbb{N}$, $\llbracket s_n \rrbracket^N \preceq^\circ \llbracket s'_n \rrbracket^N$. Lemma 1 then establishes what was left to prove: $lim(\!|(\llbracket s_n \rrbracket)_{n\in\mathbb{N}}|\!) \preceq^\circ lim(\!|(\llbracket s'_n \rrbracket)_{n\in\mathbb{N}}|\!)$.

Lemma 24 *Consider two ascending sequences $(s_n)_{n\in\mathbb{N}}$ and $(s'_n)_{n\in\mathbb{N}}$ in $(C^\circ)^\omega$, and $N \in \mathbb{N}$ such that $[(s_n)_{n\in\mathbb{N}}]_\sim \preceq [(s'_n)_{n\in\mathbb{N}}]_\sim$ and for all $n \in \mathbb{N}$, $\delta\ s_n < N \leq \delta\ s'_n$. Then $[(s_n)_{n\in\mathbb{N}}]_\sim \prec [(s'_n)_{n\in\mathbb{N}}]_\sim$.*

Proof. Let $k = [(s_n)_{n\in\mathbb{N}}]_\sim$ and $k' = [(s'_n)_{n\in\mathbb{N}}]_\sim$. We have the hypothesis $k \preceq k'$, i.e., $k \precsim k'$ following Definition 8 item 3.(c), and have to show $k \prec k'$. For, assuming the contrary, $k = k'$, i.e., for all $M \in \mathbb{N}$, $\llbracket k_n \rrbracket^M = \llbracket k'_n \rrbracket^M$, and in particular $\llbracket k_n \rrbracket^N = \llbracket k'_n \rrbracket^N$. By Definition 6, the latter amounts to proving $lim(\!|(\llbracket s_n \rrbracket^N)_{n\in\mathbb{N}}|\!) = lim(\!|(\llbracket s'_n \rrbracket^N)_{n\in\mathbb{N}}|\!)$, which implies that for some sufficiently large $n \geq \mathbb{N}$, $\llbracket s_n \rrbracket^N = \llbracket s'_n \rrbracket^N$. But by hypothesis, $\delta\ s_n < N$, hence, by the monotonicity of δ in Assumption 2, $\delta\ (\llbracket s_n \rrbracket^N) < N$. Also by hypothesis, $\delta\ s'_n \geq N$, hence, by Assumption 2 item 3, $\delta(\llbracket s'_n \rrbracket^M) = N$, leading to a contradiction. The origin of the contradiction is the assumption $k = k'$, hence, $k \prec k'$, which proves the lemma.

Lemma 25 *For all $k \in K$, k is totally defined if and only if k is maximal w.r.t. $\preceq$.*

Proof. ($\Rightarrow$): Let $k' \in K$ such that $k \preceq k'$. From $\delta\ k = \infty$ and Lemma 20 it follows that $\delta\ k' = \infty$, and from Lemma 22 if follows that for all $m \in \mathbb{N}$, $\llbracket k \rrbracket^m = \llbracket k' \rrbracket^m$. By Definition 7, $k' \precsim k$, i.e., $k' \preceq k$ according to Definition 8 item 3.(c). Hence, k is maximal w.r.t. $\preceq$.

($\Leftarrow$): We show the contrapositive of this implication: assuming $\delta\ k \in \mathbb{N}$, we show that k is not maximal. From $\delta\ k \in \mathbb{N}$, by Definition 10, $M := max\ [(\delta(\llbracket k \rrbracket^m)_{m\in\mathbb{N}})] \in \mathbb{N}$, thus, for all $m \in \mathbb{N}$, $\delta(\llbracket k \rrbracket^m) \leq M$.

Consider now Assumption 2 item 5 where $(s_m)_{m\in\mathbb{N}} := (\llbracket k \rrbracket^m)_{m\in\mathbb{N}}$ is an ascending sequence. We obtain $i \in \mathbb{N}$ and an increasing sequence $(s'_j)_{j\geq i}$ such that for all $j \geq i$, $\llbracket k \rrbracket^j \preceq^\circ s'_j$ and $N < \delta\ s'_j$. For all $j < i$ we let $s'_j := (\llbracket k \rrbracket^j$. Hence, for all $j \in$, $\llbracket k \rrbracket^j \preceq^\circ s'_j$. Next, by Lemma 23, the increasing sequence $(s'_m)_{m\in\mathbb{N}}$ is ascending, and $k \preceq lim(\!|(s'_m)_{m\in\mathbb{N}}|\!)$. Finally, by Lemma 24 instantiated with $N := M + 1$, $k \prec lim(\!|(s'_m)_{m\in\mathbb{N}}|\!)$. Hence, k is not maximal; which concludes the proof of the contrapositive of ($\Leftarrow$) and of the lemma.

4.2 Sufficient Conditions for Productiveness

We now use the results in the previous sections in order to give practical sufficient conditions for the productiveness of functionals and, by way of consequence, for defining total corecursive functions as unique fixpoints of the functionals in question. In this section we globally assume two partial orders with least elements $(D^\circ, \preceq^\circ_D, \bot_D)$ and $(C^\circ, \preceq^\circ_C, \bot_C)$. Let $(D, \preceq_D, \bot_D)$ and $(C, \preceq_C, \bot_C)$ be the respective ωCPOs built by completion, with K_D and K_C the respective sets of equivalence classes, $\llbracket\rrbracket_D$ and $\llbracket\rrbracket_C$ their approximation functions, and δ_D, δ_C their definedness functions.

Notation. $\widehat{D}$ denotes the set of maximal elements of D. A function $f^\circ : D^\circ \to C^\circ$ is *strictly monotonic* if for all $x, y \in D$, $x \prec^\circ_D y$ implies $f^\circ\ x \prec^\circ_C f^\circ\ y$. Note that a strictly monotonic function is also monotonic : $x \preceq^\circ_D y$ implies $f^\circ\ x \preceq^\circ_C f^\circ\ y$). The function f° *preserves definedness* if for all $x \in D$, $\delta_C(f^\circ\ x) \geq \delta_D\ x$.

Lemma 26 *Assume $f^\circ : D^\circ \to C^\circ$ is strictly monotonic and definedness preserving, and the functional $F : (\widehat{D} \to C) \to \widehat{D} \to C$ is monotonic and $\forall n \in \mathbb{N}, \forall x \in \widehat{D}, F^n \bot\!\!\!\bot\ x = f^\circ(\llbracket x \rrbracket^n_D)$. Then F is productive.*

Proof. Since F is monotonic, there remains to show the rest of the productiveness requirement: for all $x \in \widehat{D}$, $lim(\!|(F^n \bot\!\!\!\bot\ x)_{n\in\mathbb{N}}|\!)$ is maximal in C. Thanks to the hypothesis $\forall n \in \mathbb{N}, \forall x \in \widehat{D}, (F^n \bot\!\!\!\bot\ x) = f^\circ(\llbracket x \rrbracket^n_D)$, what we have to show is that for all $x \in \widehat{D}$, $lim(\!|(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}|\!)$ is maximal in C. There are two cases:

- if $x \in D^\circ \cap \widehat{D}$ then $(\llbracket x \rrbracket^n_D)_{n\in\mathbb{N}}$ is increasing and stabilizes at its limit x. By monotonocity of f°, $(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}$ is also increasing and stabilizes at its limit $f^\circ\ x$. Since x is maximal, by Lemma 19 $\delta_D\ x = \infty$ and by hypothesis, $\delta_C\ (f^\circ\ x) \geq \delta_D\ x = \infty$, hence, again by Lemma 19, $f^\circ\ x$ is maximal in C and since $f^\circ\ x = lim(\!|(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}|\!)$ we obtain the maximality of $lim(\!|(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}|\!)$;
- $x \in K_D \cap \widehat{D}$: by Lemma 13, $(\llbracket x \rrbracket^n_D)_{n\in\mathbb{N}}$ is ascending and its limit is x. Now, none of the elements in the sequence is maximal, otherwise the sequence would by stabilizing; hence, by Lemma 25, for all $n \in \mathbb{N}$, $\delta_D\ (\llbracket x \rrbracket^n_D) < \infty$. By (strict) monotonocity of f°, $(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}$ is also ascending. Since x is maximal, by Lemma 13 $\delta_D\ x = \infty$, which, by Definition 10, means $lim(\!|(\delta_D\ (\llbracket x \rrbracket^n_D))|\!)_{n\in\mathbb{N}}) = \infty$. Again, none of the elements in the sequence $(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}$ is maximal, otherwise the sequence would by stabilizing, Hence, by Lemma 25, for all $n \in \mathbb{N}$, $\delta_C\ (f^\circ(\llbracket x \rrbracket^n_D)) < \infty$. By hypothesis, $\delta_C\ (f^\circ(\llbracket x \rrbracket^n_D)) \geq \delta_D\ \llbracket x \rrbracket^n_D$ and therefore $\delta_C(lim(\!|(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}|\!)) \geq \delta_D(lim(\!|(\llbracket x \rrbracket^n_D)_{n\in\mathbb{N}}|\!)) = \delta_D\ x = \infty$, which, using Lemma 13, implies that $lim(\!|(f^\circ(\llbracket x \rrbracket^n_D))_{n\in\mathbb{N}}|\!)$ is maximal in C. The proof is done.

Lemma 26 is used in conjunction with Theorem 1 for defining total corecursive functions as unique fixpoints of their functionals. We next illustrate how everything comes together on Rose trees.

5 Rose Trees as an ωCPO and the *mirror* function as a unique fixpoint

In this section we define Rose trees using the finite trees in Example 1 and the ωCPO construction in Section 3. We also define the *mirror* function, a total corecursive function between Rose trees, as a unique fixpoint of its functional, proved to be productive using the techniques shown in Section 4.

Remember from Example 1 that finite trees C° have constructors $\perp$ and $tree^\circ\ l$ for finite lists l of finite trees. The prefix order $\preceq^\circ$ is such that for all $t \in C^\circ$, $\perp \preceq^\circ t$, and for all lists l, l' of trees having the same length m, $tree^\circ\ l \preceq^\circ tree^\circ\ l'$ if and only if for all $i < m$, $l[i] \preceq^\circ l'[i]$. The $forest^\circ$ accessor is defined from non-$\perp$ trees - i.e., trees of the form $tree^\circ\ l$ for l a list of finite trees, by $forest^\circ(tree^\circ l) = l$.

We have seen in the previous sections that, together with their approximations $\llbracket\ \rrbracket$) and definedness function δ, finite trees satisfy Assumptions 1 and 2 therefore constitute an ωCPO $(C^\circ \cup K, \preceq, \perp)$ where K is the set of equivalence classes of ascending sequences of finite trees, some of which are maximal - or, equivalently, totally defined. Ascending sequences require trees having at least one branch that grows "forever", and their limits (in K) have at least one infinite branch. Overall, the set K together with the set C° of finite trees constitute the set $C^\circ \cup K$ of Rose trees, of finite breadth and possibly infinite depth.

5.1 Extending the $tree^\circ$ Constructor and $forest^\circ$ Accessor from Finite Trees to Rose Trees

The ωCPO of Rose trees is not yet fully functional (for the purpose of defining total corecursive functions) until we extend the $tree^\circ$ constructor and $forest^\circ$ accessor from finite trees to Rose trees. Indeed, functionals for corecursive functions under definition use the extended constructors and accessors. Such is the functional $Mirror = \lambda\ f.\lambda\ t.tree\ (map\ f\ (reverse\ (forest\ t)))$ for the *mirror* function that shall we define.

We first extend he notion of diagonal, defined for equivalence classes in K (Def. 9), to $C^\circ \cup K$.

Definition 11 *For all $c \in C^\circ \cup K$, we define the diagonal of c to be the sequence $(\llbracket c \rrbracket^n)_{n\in\mathbb{N}}$.*

Lemma 27 *For all $c \in C^\circ \cup K$, the diagonal of c is an increasing sequence and its limit is c.*

Proof. For $c \in K$ we the result follows from Lemma 13. For $c \in C^\circ$, the sequence is increasing due to Assumption 1 item 2.(c). By Lemma 2, for large enough $n \in \mathbb{N}$, $\llbracket c \rrbracket^n = c$, meaning that $(\llbracket c \rrbracket^n)_{n\in\mathbb{N}}$ is increasing and stabilizes at c, hence its limt is c.

Consider a list $l = [t_0, \ldots, t_{m-1}]$ of elements in $C = C^\circ \cup K$. By Lemma 27, each t_i is the limit of its diagonal: $t_i = lim(\!|(\llbracket t_i \rrbracket^n)_{n\in\mathbb{N}}|\!)$. Let us consider the sequence of lists $([\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}$. This sequence is pointwise increasing: for all $n \in \mathbb{N}$ and $i < m$, $\llbracket t_i \rrbracket^n \preceq^\circ \llbracket t_i \rrbracket^{n+1}$. Thanks to the monotonicity of the $\preceq^\circ$ order, the sequence $(tree^\circ[\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}$ is increasing as well. This justifies the following definition:

Definition 12 $tree\ [t_0, \ldots, t_{m-1}] \triangleq lim(\!|(tree^\circ[\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}|\!)$.

We now give equivalent conditions under which $tree\ [t_0, \ldots, t_{m-1}] \in C^\circ$ and $tree\ [t_0, \ldots, t_{m-1}] \in K$.

Lemma 28 *$tree\ [t_0, \ldots, t_{m-1}] \in C^\circ$ if and only if for all $i < m$, $t_i \in C^\circ$. If this is the case then $tree\ [t_0, \ldots, t_{m-1}] = tree^\circ\ [t_0, \ldots, t_{m-1}]$.*

Proof.

($\Rightarrow$): if $tree\ [t_0, \ldots, t_{m-1}] \in C^\circ$ then the increasing sequence $(tree^\circ[\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}$ cannot be ascending; otherwise, its limit (and in particular upper bound) $lim(\!|(tree^\circ[\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}|\!)$, which by Definition 12 satisfies $lim(\!|(tree^\circ[\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}|\!) = tree\ [t_0, \ldots, t_{m-1}]$, is in C°, and this contradicts Assumption 1 item 1. Hence, the sequence $(tree^\circ[\llbracket t_0 \rrbracket^n, \ldots, \llbracket t_{m-1} \rrbracket^n])_{n\in\mathbb{N}}$ is stabilizing.

This implies that for all $i < m$, the increasing sequence $(\lceil\!\lceil t_i \rceil\!\rceil^n)_{n\in\mathbb{N}}$ is stabilizing (the contrary would imply that $(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}$ is non-stabilizing as well). The value at which $(\lceil\!\lceil t_i \rceil\!\rceil^n)_{n\in\mathbb{N}}$ is stabilizing is its limit t_i, hence, the latter is in C°. This implies that $(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_m - 1 \rceil\!\rceil^n])_{n\in\mathbb{N}}$ stabilizes at $tree^\circ\ [t_0, \ldots, t_{m-1}]$, meaning that $lim(\!|(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}|\!) = tree^\circ\ [t_0, \ldots, t_{m-1}]$ and using Definition 12, $tree^\circ\ [t_0, \ldots, t_{m-1}] = tree\ [t_0, \ldots, t_{m-1}]$.

($\Leftarrow$): if for all $i < m$, $t_i \in C^\circ$, then for all $i < m$, $(\lceil\!\lceil t_i \rceil\!\rceil^n)_{n\in\mathbb{N}}$ is stabilizing at its limit t_i. It follows that $(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}$ is stabilizing and its limit equals $tree^\circ\ [t_0, \ldots, t_{m-1}] \in C^\circ$. On the other hand, by Definition 12, $lim(\!|(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}|\!) = tree\ [t_0, \ldots, t_{m-1}]$. It follows that $tree\ [t_0, \ldots, t_{m-1}] \in C^\circ$ and $tree\ [t_0, \ldots, t_{m-1}] = tree^\circ\ [t_0, \ldots, t_{m-1}]$.

Lemma 29 *$tree\ [t_0, \ldots, t_{m-1}] \in K$ if and only if there exists $i < m$ such that $t_i \in K$. If this is the case then $tree\ [t_0, \ldots, t_{m-1}] = [(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}]_\sim$*

Proof. $tree\ [t_0, \ldots, t_{m-1}] \in K$ is equivalent to $tree\ [t_0, \ldots, t_{m-1}] \notin C^\circ$, which by Lemma 28 is equivalent to the existence of some $i < m$ such that $t_i \in K$. The statement $tree\ [t_0, \ldots, t_{m-1}] = [(tree^\circ[\lceil\!\lceil t_0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t_{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}]_\sim$ follows from Definition 12 and Definition 8 item 2 of limits of ascending sequences over C°.

The next result holds in general (i.e., not only for sequences of trees) but was not needed until now.

Lemma 30 *For any increasing sequence $(s_n)_{n\in\mathbb{N}} \in (C^\circ)^\omega$, $\lceil\!\lceil lim(\!|(s_n)_{n\in\mathbb{N}}|\!) \rceil\!\rceil^N = lim(\!|(\lceil\!\lceil s_n \rceil\!\rceil^N)_{n\in\mathbb{N}}|\!)$.*

Proof. If the sequence is ascending then the lemma follows from Definition 6 and Definition 8 item 2. If the sequence is stabilizing to some $c \in C^\circ$ then $\lceil\!\lceil lim(\!|(s_n)_{n\in\mathbb{N}}|\!) \rceil\!\rceil^N = \lceil\!\lceil c \rceil\!\rceil^N = lim(\!|(\lceil\!\lceil s_n \rceil\!\rceil^N)_{n\in\mathbb{N}}|\!)$.

The following lemma states that the function *tree* from Definition 12 is continuous in the sense of Scott:

Lemma 31 *For all $m \in \mathbb{N}$, if for all $i < m$, the sequences $(s_n^i)_{n\in N}$ are increasing, then it holds that $lim(\!|(tree^\circ[s_n^0, \ldots, s_n^{m-1}])_{n\in\mathbb{N}}|\!) = tree\ [(lim(\!|(s_n^0)_{n\in N}|\!)), \ldots, (lim(\!|(s_n^{m-1})_{n\in N}|\!))]$.*

Proof. We first note that $lim(\!|(tree^\circ[s_n^0, \ldots, s_n^{m-1}])_{n\in\mathbb{N}}|\!)$ exists due to monotonicity of the $tree^\circ$ constructor. We first prove the lemma for the *stabilizing case* - all sequences $(s_n^i)_{n\in N}$ stabilize. They stabilize to their limits $lim(\!|(s_n^i)_{n\in N}|\!)$, thus, $(tree^\circ[s_n^0, \ldots, s_n^{m-1}])_{n\in\mathbb{N}}$ stabilizes to $tree^\circ[(lim(\!|(s_n^0)_{n\in N}|\!)), \ldots, (lim(\!|(s_n^{m-1})_{n\in N}|\!))]$ which by Lemma 28 equals $tree\ [(lim(\!|(s_n^0)_{n\in N}|\!)), \ldots, (lim(\!|(s_n^{m-1})_{n\in N}|\!))]$; which proves the stabilizing case.

We now focus on the general case. Let $t^i = lim(\!|(s_n^i)_{n\in N}|\!)$ for $i < m$. By Def. 12, $tree[t^0, \ldots, t^{m-1}] = lim(\!|(tree^\circ[\lceil\!\lceil t^0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t^{m-1} \rceil\!\rceil^n])_{n\in\mathbb{N}}|\!)$. But using Lemma 30, for all $i < m$ and $n \in \mathbb{N}$, $\lceil\!\lceil t^i \rceil\!\rceil^n \triangleq \lceil\!\lceil lim(\!|(s_j^i)_{j\in N}|\!) \rceil\!\rceil^n = lim(\!|(\lceil\!\lceil s_j^i \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!)$, thus, $tree^\circ[\lceil\!\lceil t^0 \rceil\!\rceil^n, \ldots, \lceil\!\lceil t^{m-1} \rceil\!\rceil^n] = tree^\circ[(lim(\!|(\lceil\!\lceil s_j^0 \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!)), \ldots, (lim(\!|(\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!))]$ for all $n \in \mathbb{N}$. Hence, (†): $tree[t^0, \ldots, t^{m-1}] = lim(\!|(tree^\circ[(lim(\!|(\lceil\!\lceil s_j^0 \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!)), \ldots, (lim(\!|(\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!))])_{n\in\mathbb{N}}|\!)$.

In the right-hand side of the (†) equality, all sequences $(\lceil\!\lceil s_j^0 \rceil\!\rceil^n)_{j\in\mathbb{N}}, \ldots, (\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)_{j\in\mathbb{N}}$ stabilize. Using the *stabilizing case* estblished above we obtain $tree^\circ[(lim(\!|(\lceil\!\lceil s_j^0 \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!)), \ldots, (lim(\!|(\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!))] = lim(\!|(tree^\circ[(\lceil\!\lceil s_j^0 \rceil\!\rceil^n), \ldots, (\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)])_{j\in\mathbb{N}}|\!)$, for all $n \in \mathbb{N}$. By rewriting the latter equality in (†) we obtain (‡): $tree[t^0, \ldots, t^{m-1}] = lim(\!|(lim(\!|(tree^\circ[(\lceil\!\lceil s_j^0 \rceil\!\rceil^n), \ldots, (\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)])_{j\in\mathbb{N}}|\!))_{n\in\mathbb{N}}|\!)$. Using properties of the $\lceil\!\lceil\ \rceil\!\rceil$ function, for all $n, j \in \mathbb{N}$, $tree^\circ[(\lceil\!\lceil s_j^0 \rceil\!\rceil^n), \ldots, (\lceil\!\lceil s_j^{m-1} \rceil\!\rceil^n)] = \lceil\!\lceil tree^\circ[s_j^0, \ldots, s_j^{m-1}] \rceil\!\rceil^{n+1}$. Rewriting this equality in (‡) gives (♯): $tree[t^0, \ldots, t^{m-1}] = lim(\!|(lim(\!|(\lceil\!\lceil tree^\circ[s_j^0, \ldots, s_j^{m-1}] \rceil\!\rceil^{n+1})_{j\in\mathbb{N}}|\!))_{n\in\mathbb{N}}|\!)$. But the sequence $(\lceil\!\lceil lim(\!|(tree^\circ[s_j^0, \ldots, s_j^{m-1}])_{j\in\mathbb{N}}|\!) \rceil\!\rceil^{n+1})_{n\in\mathbb{N}}$ is just $(\lceil\!\lceil lim(\!|(tree^\circ[s_j^0, \ldots, s_j^{m-1}])_{j\in\mathbb{N}}|\!) \rceil\!\rceil^n)_{n\in\mathbb{N}}$ without its first element, hence, $lim(\!|(\lceil\!\lceil lim(\!|(tree^\circ[s_j^0, \ldots, s_j^{m-1}])|\!)_{j\in\mathbb{N}} \rceil\!\rceil^{n+1}|\!)_{n\in\mathbb{N}} = lim(\!|(\lceil\!\lceil lim(\!|(tree^\circ[s_j^0, \ldots, s_j^{m-1}])|\!)_{j\in\mathbb{N}} \rceil\!\rceil^n|\!)_{n\in\mathbb{N}}$, which together (♯) gives by transitivity (♭): $tree[t^0, \ldots, t^{m-1}] = lim(\!|(lim(\!|(\lceil\!\lceil tree^\circ[s_j^0, \ldots, s_j^{m-1}] \rceil\!\rceil^n)_{j\in\mathbb{N}}|\!))_{n\in\mathbb{N}}|\!)$. Using Lemma 30, $lim(\!|(\lceil\!\lceil tree^\circ[s_j^0, \ldots, s_j^{m-1}] \rceil\!\rceil^n)|\!)_{j\in\mathbb{N}} = \lceil\!\lceil lim(\!|(tree^\circ[s_j^0, \ldots, s_j^{m-1}])|\!)_{j\in\mathbb{N}} \rceil\!\rceil^n$ for all $n \in \mathbb{N}$. By rewriting the latter equality in (♭) we get (♮): $tree[t^0, \ldots, t^{m-1}] = lim(\!|(\lceil\!\lceil lim(\!|(tree^\circ[s_j^0, \ldots, s_j^{m-1}])|\!)_{j\in\mathbb{N}} \rceil\!\rceil^n)|\!)_{n\in\mathbb{N}}$.

By Lemma 27, $lim(\!|(\llbracket lim(\!|(tree^\circ[s_j^0,\ldots,s_j^{m-1}])\!|)_{j\in\mathbb{N}}\rrbracket^n)\!|)_{n\in\mathbb{N}} = lim(\!|(tree^\circ[s_j^0,\ldots,s_j^{m-1}])\!|)_{j\in\mathbb{N}}$, and by transitivity: (•): $tree[t^0,\ldots,t^{m-1}] = lim(\!|(tree^\circ[s_j^0,\ldots,s_j^{m-1}])\!|)_{j\in\mathbb{N}}$. Finally, $tree[lim(\!|(s_n^0)_{n\in N}|\!),\ldots,lim(\!|(s_n^{m-1})_{n\in N}|\!)]$, which is by definition $tree[t^0,\ldots,t^{m-1}]$, was proved by (•) equal to $lim(\!|(tree^\circ[s_j^0,\ldots,s_j^{m-1}])\!|)_{j\in\mathbb{N}}$: qed.

There remains to prove that *tree* satisfies the following properties of its restriction to finite trees:

- approximations: $\llbracket tree\ l\rrbracket^0 = \bot$ and $\llbracket tree\ l\rrbracket^{N+1} = tree(map\ \llbracket\cdot\rrbracket^N)\ l$;
- 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 infinite trees: for all $k\in K$ there exists l such that $k = tree\ l$;
- 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.

Lemma 32 *For any list l over $C^\circ\cup K$, $\llbracket tree\ l\rrbracket^0 = \bot$ and for all $N\in\mathbb{N}$, $\llbracket tree\ l\rrbracket^{N+1} = tree((map\ \llbracket\cdot\rrbracket^N)\ l)$.*

Proof. If all elements of l are in C° then the statements follows from the definition of $\llbracket\ \rrbracket$ for finite trees. Assume now there is at least one element of l which is in K, and let $l = [t_0,\ldots t_{m-1}]$. Then, $tree\ [t_0,\ldots t_{m-1}]\in K$ by Lemma 29, and by Def 12, $tree\ [t_0,\ldots t_{m-1}] = lim(\!|(tree^\circ[\llbracket t_0\rrbracket^n,\ldots,\llbracket t_{m-1}\rrbracket^n])_{n\in\mathbb{N}}|\!)$. Then, by Lemma 30, for all $M\in\mathbb{N}$, (†): $\llbracket tree\ [t_0,\ldots t_{m-1}]\rrbracket^M = lim(\!|(\llbracket tree^\circ[\llbracket t_0\rrbracket^n,\ldots,\llbracket t_{m-1}\rrbracket^n]\rrbracket^M)_{n\in\mathbb{N}}|\!)$.

- if $M = 0$ then (†) reduces to $\llbracket tree\ l\rrbracket^0$ being the limit of a sequence of $\bot$, which proves the first part of the lemma: $\llbracket tree\ l\rrbracket^0 = \bot$;
- if $M > 0$, i.e., $M = N+1$, the right-hand side of (†) becomes (by definition of $\llbracket\ \rrbracket$ for finite trees) $lim(\!|(tree^\circ[\llbracket\llbracket t_0^n\rrbracket\rrbracket^N,\ldots,\llbracket\llbracket t_{m-1}\rrbracket^n\rrbracket^N_{n\in\mathbb{N}}])\!|)$. For $n\geq N$, by Assumption 1 item 5 it holds that for all $i<m$, $\llbracket\llbracket t_i\rrbracket^n\rrbracket^N = \llbracket t_i\rrbracket^N$. Hence, the increasing sequence $(tree^\circ[\llbracket\llbracket t_0\rrbracket^n\rrbracket^N,\ldots,\llbracket\llbracket t_{m-1}\rrbracket^n\rrbracket^N])_{n\in\mathbb{N}}$ stabilizes to $tree^\circ[\llbracket t_0\rrbracket^N,\ldots,\llbracket t_{m-1}\rrbracket^N]$. Since for all $i<m$, $\llbracket t_i\rrbracket^N\in C^\circ$ we have, by Lemma 28, $tree^\circ[\llbracket t_0\rrbracket^N,\ldots,\llbracket t_{m-1}\rrbracket^N] = tree[\llbracket t_0\rrbracket^N,\ldots,\llbracket t_{m-1}\rrbracket^N]$. Remembering that $l = [t_0,\ldots t_{m-1}]$ and $M = N+1$, (†) proves the second statement of the lemma: $\llbracket tree\ l\rrbracket^{N+1} = tree((map\ \llbracket\cdot\rrbracket^N)\ l)$.

Lemma 33 *$tree\ l\preceq tree\ l'$ iff for some $m\in\mathbb{N}$, $length\ l = length\ l' = m$ and for all $i = 1,\ldots,m$, $l[i]\preceq l'[i]$.*

Proof.

($\Rightarrow$): if both l and l' consist of finite trees then the statement follows from the definition of $\preceq^\circ$. Otherwise, $tree\ l\preceq tree\ l'$ holds in two possible cases:

- $tree\ l\in C^\circ$ and $tree\ l'\in K$. Let $m = length\ l$. Using Lemma 28, for all $i<m$, $l[i]\in C^\circ$ and $tree\ l = tree^\circ\ l$. By Definition 8 item 3.(b), $tree^\circ\ l\preceq tree\ l$ implies there exists $M\in\mathbb{N}$ such that $tree^\circ\ l\preceq^\circ \llbracket tree\ l\rrbracket^M$. Now, $M = 0$ is impossible since, by Lemma 32, this would imply the impossible inequality $tree^\circ\ l\preceq^\circ\bot$. Hence, $M = N+1$ and then, using Lemma 32, we obtain $tree^\circ\ l\preceq^\circ tree(map\ (\llbracket\cdot\rrbracket^M)l')$. Now, all the elements of $map\ (\llbracket\cdot\rrbracket^M)l'$ are finite trees, thus, using Lemma 28, we obtain $tree^\circ\ l\preceq^\circ tree^\circ(map\ (\llbracket\cdot\rrbracket^N)l')$. Using the properties of the constructor $tree^\circ$, we obtain $length(map\ (\llbracket\cdot\rrbracket^M)\ l') = length\ l = m$ and then $length\ l' = m$ using the property that *map* preserves list length. Moreover, we obtain that for all $i<m$, $l[i]\preceq^\circ\llbracket l'[i]\rrbracket^N$, which, by Definition 8 item 3.(b) (if $l'[i]\in K$) resp. Assumption 1 item 2.(a) (if $l'[i]\in C^\circ$) implies that for all $i<m$, $l[i]\preceq l'[i]$, which settles this case.
- $tree\ l\in K$ and $tree\ l'\in K$. Then, using Definition 8 item 3.(c), for all $N\in\mathbb{N}$, $\llbracket tree\ l\rrbracket^{N+1}\preceq\llbracket tree\ l'\rrbracket^{N+1}$. Fix an arbitrary $N\in\mathbb{N}$. Uusing Lemma 32, $tree\ (map(\llbracket\cdot\rrbracket^N)l)\preceq tree\ (map(\llbracket\cdot\rrbracket^N)l)$, which, using Lemma 28 and Lemma 8 item 3.(a), becomes $tree^\circ\ (map(\llbracket\cdot\rrbracket^N)l)\preceq^\circ tree^\circ\ (map(\llbracket\cdot\rrbracket^N)l)$. Next, using properties of the $\preceq^\circ$ order, $length\ (map(\llbracket\cdot\rrbracket^N)l) = length\ (map(\llbracket\cdot\rrbracket^N)l)$ which implies

length l = *length* l' and let m denote the common value of the lengths. We obtain that for all $i < m$, $\llbracket l[i] \rrbracket^N \preceq^\circ \llbracket l'[i] \rrbracket^N$, and since $N \in \mathbb{N}$ has been chosen arbitrarily, by Definition 8 item 3.(b) (if $l'[i] \in K$) resp. Lemma 2 (if $l'[i] \in C^\circ$) using Definition 8 item 3.(c), implies that for all $i < m$ it holds that $l[i] \preceq l'[i]$, which settles this case as well.

($\Leftarrow$): again, if both l and l' consist of finite trees then the statement follows from the definition of $\preceq^\circ$. The case where l contains an element of K, say, at position i, why l' contains only elements of C° is impossible, since (by Definition 8 item 3), $l[i] \in K$ cannot be in $\preceq$ order with $l'[i] \in C^\circ$. The two remaining cases are:

- l only consists of elements in C° and l' contains at least one element in K: let $0 \leq i_1, i_2, \ldots i_n < m$ be the positions where $l'[i_j] \in K$ for $j = 1, \ldots, n$. For any such index i_j there is N_j such that $l[i_j] \preceq^\circ \llbracket l'[i_j] \rrbracket^{N_j}$. Let N' be the maximum of $\{N_j | j = 1, \ldots, n\}$. For the positions in the list *other than* $i_1, i_2, \ldots i_n$, the elements at those positions are in C°. Using Lemma 2, there exists N'' such that for all $i_p \in \{0, \ldots m-1\} \setminus \{i_1, \ldots i_n\}$, $\llbracket l'[i_p] \rrbracket^{N''} = l'[i_p]$. Take now $N = max\ N'\ N''$. Then, using Assumption 1 item 2.(c), Lemma 2 and the hypothesis of the implication, we obtain (†): for all $i < m$, $l[i] \preceq^\circ \llbracket l'[i] \rrbracket^N$. Next, since l contains only elements in C°, by Lemma 28, *tree* l = *tree*$^\circ$ $l \in C^\circ$. We now show (‡): *tree*$^\circ$ $l \preceq^\circ \llbracket$*tree* $l' \rrbracket^{N+1}$, which by Definition 8 item 3.(b) implies the desired conclusion *tree* $l \preceq$ *tree* l'. Now, (‡) amounts to *tree*$^\circ$ $l \preceq^\circ$ *tree*$(map(\llbracket \cdot \rrbracket^N)\ l')$, which (since *length* l = *length* l' implies *length* l = *length* $(map(\llbracket \cdot \rrbracket^N) l')$) reduces to (†): this case is proved.
- both l and l' contain at least one element of K. Then, by Lemma 29, *tree* $l \in K$ and *tree* $l' \in K$. In order to prove *tree* $l \preceq$ *tree* l', by Definition 8 item 3.(b), we need to prove *tree* $l \precsim$ *tree* l', which by Lemma 8 amounts to proving (†): for all $M \in \mathbb{N}$ there exists $M' \in \mathbb{N}$ such that $\llbracket$*tree* $l \rrbracket^M \preceq^\circ \llbracket$*tree* $l' \rrbracket^{M'}$. For $M = 0$ any $M' \in \mathbb{N}$ will do. Now consider an arbitrary $M > 0$, i.e., $M = N + 1$. Using Lemma 32 and the properties of $\preceq^\circ$, (†) reduces to proving (‡): *for all* $N \in \mathbb{N}$ *and* $i < m$, *there exists* $N' \in \mathbb{N}$ *such that* $\llbracket l[i] \rrbracket^N \preceq^\circ \llbracket l'[i] \rrbracket^{N'}$. Let $0 \leq i_1, i_2, \ldots i_n < m$ the positions at which $l'[i_j] \in K$ for $j = 1, \ldots, n$. For any such index i_j there is N_j such that $l[i_j] \preceq^\circ \llbracket l'[i_j] \rrbracket^{N_j}$. We choose $N \in \mathbb{N}$ and let N' be the maximum of $\{N\} \cup \{N_j | j = 1, \ldots, n\}$. We show (‡) for the arbitrary N, i and chosen N':
  - if $l[i] \in C^\circ$ and $l'[i] \in C^\circ$, $\llbracket l[i] \rrbracket^N \preceq^\circ \llbracket l'[i] \rrbracket^{N'}$ results from $l[i] \preceq^\circ l'[i]$ (obtained from the hypothesis $l[i] \preceq l'[i]$ and Definition 8 item 3.(a)), $N \leq N'$, and Assumption 1 items 2.(b), 2.(c);
  - if $l[i] \in C^\circ$ and $l[i] \in K$, we have $\llbracket l[i] \rrbracket^N \preceq^\circ l[i]$ by Assumption 1 item 2.(a), hence, by Definition 8 item 3.(a), $\llbracket l[i] \rrbracket^N \preceq l[i]$. Moreover, $l[i] \preceq^\circ \llbracket l'[i] \rrbracket^{N'}$, from the definition of N' and Assumption 1 item 2.(c), and then $l[i] \preceq \llbracket l'[i] \rrbracket^{N'}$ by Definition 8 item 3.(a). By transitivity, we obtain the desired $\llbracket l[i] \rrbracket^N \preceq \llbracket l'[i] \rrbracket^{N'}$;
  - if $l[i] \in K$ and $l[i] \in K$, $\llbracket l[i] \rrbracket^N \preceq^\circ \llbracket l'[i] \rrbracket^{N'}$ results from the hypothesis $l[i] \preceq l'[i]$, $N \leq N'$, and Lemma 11 items 2 and 3. This completes the proof of the current subcase and of the lemma.

Lemma 34 *For all* $k \in K$ *there exists a list* l *of Rose trees such that* k = *tree* l.

Proof. Consider the diagonal $(\llbracket k \rrbracket^n)_{n \in \mathbb{N}}$. By Lemma 13 it is an ascending sequence and its limit is k. Hence, there exists $M \in \mathbb{N}$ such that for all $j \geq M$, $\llbracket k \rrbracket^j \neq \bot$. We choose M minimal with the above property, hence, and for all $j < M$, $\llbracket k \rrbracket^j = \bot$. Thus, for all $j \geq M$, $\llbracket k \rrbracket^j$ = *tree*$^\circ$ l^j for some list l^j of finite trees. Due to the increasing nature of $(\llbracket k \rrbracket^n)_{n \in \mathbb{N}}$, all the lists l^j have the same length, say, m. Moreover, $m > 0$, because if $m = 0$ the sentence would be stabilizing. Thus (†): for all $j \geq M$, $\llbracket k \rrbracket^j$ = *tree*$^\circ$ $[t^j_0, \ldots t^j_{m-1}]$ with $m > 0$. For any arbitrarily chosen $i < m$, consider the sequence $(t_i^{n+M})_{n \in \mathbb{N}}$. Thanks to properties of $\preceq^\circ$ the sequence is increasing, and let t_i be its limit. There remains to prove

that $k = tree\ [t_0,\ldots,t_{m-1}]$. By Lemma 31, $tree\ [t_0,\ldots,t_{m-1}] = lim(\!|(tree^\circ[t_0^{n+M},\ldots,t_{m-1}^{n+M}])_{n\in\mathbb{N}}|\!)$. Using (†), $lim(\!|(tree^\circ[t_0^{n+M},\ldots,t_{m-1}^{n+M}])_{n\in\mathbb{N}}|\!) = lim(\!|(\lceil\!|k|\!\rceil^{n+M})_{n\in\mathbb{N}}|\!)$. But $(\lceil\!|k|\!\rceil^{n+M})_{n\in\mathbb{N}}$ is a subsequence of $(\lceil\!|k|\!\rceil^n)_{n\in\mathbb{N}}$, hence, using Lemma 12 the latter two sequences are similar and have the same limit k. This proves $tree\ [t^0,\ldots,t^{m-1}] = k$ and the lemma.

Lemma 35 *For all lists of Rose trees l, l', $tree\ l = tree\ l'$ implies $l = l'$.*

Proof. $tree\ l = tree\ l'$ implies $tree\ l \preceq tree\ l'$ and $tree\ l' \preceq tree\ l$. The former implies $length\ l = length\ l' = m$ for some $m \in \mathbb{N}$ and for all $i < m$, $l[i] \preceq l'[i]$. Symmetrically, for all $i < m$, $l'[i] \preceq l[i]$. It follows that $length\ l = length\ l' = m$ for some $m \in \mathbb{N}$ and for all $i < m$, $l[i] = l'[i]$, i.e., the equality of the two lists.

We can now define the *forest* function as an extension of the $forest^\circ$ accessor:

Definition 13 *Given $t \in (C^\circ \setminus \{\bot\}) \cup K$, we define forest t as follows:*

- *if $t \in C^\circ \setminus \{\bot\}$ then $forest\ t = forest^\circ\ t$;*
- *if $t \in K$ then we let $forest\ t = l$, where l is the unique list of trees such that $t = tree\ l$. (The existence and uniquencess of l is ensured by Lemmas 34 and 35.)*

The *tree* and *forest* functions satisfy the expected equations which they inherit from $tree^\circ$ and $forest^\circ$:

Lemma 36 *For all $t \in (C^\circ \setminus \{\bot\}) \cup K$, $tree\ (forest\ t) = t$, and for all lists l of Rose trees, $forest\ (tree\ l) = l$.*

Proof. We prove the first statement in the lemma. Let $t \in (C^\circ \setminus \{\bot\}) \cup K$. If $t \in (C^\circ \setminus \{\bot\})$ the statement holds due to properties of the $tree^\circ$ constructor and $forest^\circ$ accessor. If $t \in K$ then, by Definition 13, $forest\ t$ is the unique list of trees l such that $t = tree\ l$, i.e., $t = tree\ (forest\ t)$.

We focus on the second statement. If l is a list of finite trees then by Lemma 28 $tree\ l = tree^\circ\ l \in C^\circ$ and then $forest\ (tree\ l) = forest^\circ(tree^\circ l)$, and the statement follows from $forest^\circ\ (tree^\circ\ l) = l$. If l contains at least one infinite tree then by Lemma 29, $tree\ l \in K$. The statement $forest\ (tree\ l) = l$ follows from Definition 13.

We now have all the ingredients for writing functionals involving Rose trees and for defining functions over Rose trees as unique fixpoints of the functionals in question.

6 Defining a Total Mirror Function Between Rose Trees

The *mirror* function for totally defined (equivalently, maximal) Rose trees, which we shall define via its fixpoint equation, has the functional defined by $Mirror\ f\ \hat{t} = tree(map\ f\ (reverse\ (forest\ \hat{t})))$, for all functions $f : \widehat{Tree} \to Tree$ and $\hat{t} : \widehat{Tree}$, where $\widehat{Tree}$ denotes maximal elements in the ωCPO of Rose trees. For defining the *mirror* function via the equation $mirror = Mirror\ mirror$ we use Theorem 1, which requires us to prove that the functional *Mirror* is productive. The productiveness is established via Lemma 26. For this, we first prove that F it is monotonic, then inductively define a mirror function for finite trees t°, $mirror^\circ\ t^\circ = if\ t^\circ = \bot\ then \bot\ else\ tree^\circ(map\ mirror^\circ(reverse\ (forest\ t^\circ)))$. Next, we show that $mirror^\circ$ is strictly increasing and preserves definedness, which are routine‘ proofs by induction on finite trees. Finally we prove the last (and nontrivial) part of the sufficient condition in a separate lemma:

Lemma 37 *For all $n \in \mathbb{N}$ and $\hat{t} \in \widehat{Tree}$, $Mirror^n \perp\!\!\!\perp \hat{t} = mirror^\circ(\lceil\!|\hat{t}|\!\rceil^n)$*

Proof. By induction on n. The base case is trivial as it reduces to $\bot = \bot$. For the inductive step: assume that for all $\hat{t} \in \widehat{Tree}$, $Mirror^n \perp\!\!\!\perp \hat{t} = mirror^\circ(\lceil\!|\hat{t}|\!\rceil^n)$. Fix an arbitrary $\hat{t} \in \widehat{Tree}$. We have the equality chain:

$Mirror^{n+1} \perp\!\!\!\perp \hat{t}$ = (by definition of iteration of a functional)

$Mirror(Mirror^n \bot\!\!\!\bot)\ \hat{t}$ = (by induction hypothesis and properties of function composition $\circ$)
$Mirror\ (mirror^\circ \circ (\llbracket \cdot \rrbracket^n))\ \hat{t}$ = (using $\hat{t} = tree\ (forest\ \hat{t})$ that holds by Lemma 36)
$Mirror\ (mirror^\circ \circ (\llbracket \cdot \rrbracket^n))\ (tree\ (forest\ \hat{t}))$ = (by definition of *Mirror*)
$tree\ (map\ (mirror^\circ \circ (\llbracket \cdot \rrbracket^n)))(reverse(forest\ \hat{t}))$ = (by Lemma 28)
$tree^\circ(map\ (mirror^\circ \circ (\llbracket \cdot \rrbracket^n))(reverse(forest\ \hat{t})))$ = (by properties of *map* vs. $\circ$)
$tree^\circ(map\ mirror^\circ(map(\llbracket \cdot \rrbracket^n)(reverse(forest\ \hat{t}))))$ = (by properties of *map* vs. *reverse*)
$tree^\circ(map\ mirror^\circ(reverse(map(\llbracket \cdot \rrbracket^n)\ forest\ \hat{t})))$ = (by definition of $mirror^\circ$)
$mirror^\circ(tree^\circ(map(\llbracket \cdot \rrbracket^n)\ forest\ \hat{t}))$ = (by Lemma 28)
$mirror^\circ(tree\ (map(\llbracket \cdot \rrbracket^n)\ forest\ \hat{t}))$ = (by Lemma 32)
$mirror^\circ(\llbracket tree\ (forest\ \hat{t}) \rrbracket^{n+1})$ = (using $\hat{t} = tree\ (forest\ \hat{t})$ that holds by Lemma 36)
$mirror^\circ(\llbracket \hat{t} \rrbracket^{n+1})$

which concludes the proof of the induction step and of the lemma.

Then, Lemma 26 ensures that *Mirror* is productive and Theorem 1 says that there is a unique function $mirror : \widehat{Tree} \to \widehat{Tree}$ such that $mirror = Mirror\ mirror$, i.e., for all maximal (a.k.a. totally defined) Rose trees $\hat{t}$, $mirror\ \hat{t} = map\ mirror\ (reverse\ (forest\ \hat{t}\ ))$; which was our original goal.

7 Conclusion, Related Work and Future Work

We have shown a way to build an ωCPO from a partially ordered set satisfying certain assumptions regarding approximations and definedness measures. The assumptions have a model as finite trees, and more generally, as terms built over various classes of signatures. The ωCPO resulting by completion inherits the approximation and definedness measures and enables, in combination with earlier results [9], to define total corecursive functions as unique fixpoints of their productive functionals. The approximation and definedness measures are employed in the design of practically usable, liberal sufficient conditions for productiveness. We illustrate the resulting approach for defining the *mirror* function on Rose trees as the unique fixpoint of its functional. The approach will be implemented in Coq as it is mainly designed to increase the (currently very limited) expressiveness of corecursive functions definable in Coq.

Related Work. Our 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 tree-like datatypes such as Rose trees and their natural prefix order as a metric space. We have investigated this and found the reason: the distance requires a "weak totality" property that the prefix order does not satisfy. Defining corecursive functions based on Cauchy sequences is mentioned in [7]. 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 class of definable total corecursive functions beyond the guarded ones, whereas for us that is the major goal.

We now compare our work with some textbook results from elementary domain theory [2](Chapter 1). The fixpoint theorem that we are using is a specialized version of *Kleene's theorem*, stating that continuous functionals over CPOs (a generalization of ωCPOs) have a least fixpoint. Kleene's theorem can be used for defining partial functions, and does not guarantee in any way that the resulting function is total. For totality one needs productiveness, and our version of the theorem specifically uses productiveness for defining total functions. Another textbook result is *completion by ideals* that transforms partial orders into CPOs and monotonic functions between partial orders into continuous functions between CPOs. Moreover, completion by ideals is a *free* construction, in a categorical sense; it satisfies the fewest

possible properties in order to be a completion to a CPO. However, in order to define total corecursive functions with our approach one needs an extension of a restriction of CPOs, namely, ωCPOs extended with specific approximation and definedness measures. Hence, completion by ideals as such is not useful for our approach. However, an refinement of it that would transform partial orders + approximation and definedness measures to CPOs inheriting those measures could be useful, especially if it were a free construction. In that case, some tedious results that users need to prove "by hand" for each datatype (namely, the extension of constructors and accessors of inductive types to the completed types, and their properties - in our case, all of Section 5.1) are mere consequences of the (for now, hypothetical completion) being free. We are planning to investigate this idea in future work.

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 [4]. 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 [11] 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 [10] that provide users with a uniform, automatic way of handling termination and productiveness. The implementation of sized types is currently unsound (cf. `https://github.com/agda/agda/issues/3026`).

Isabelle/HOL [12] is also major proof assistant which supports corecursive functions. It accept function that go beyond guarded corecursion [5], 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.

Future Work. We have already mentioned the idea of presenting our approach as a free construction. This is theoretical work but has practical benefits since it eases the burden users have to go through with the current completion operation for each datatype on which they wish to define total corecursive functions. Another more practical line of work is to find applications where productive but unguarded corecursive functions are necessary. We found one in the semantics of synchronous reactive languages. Specifically, synchronous languages allowing the writing of nondeterministic programs have executions that are Rose trees, and the functions involved in their semantics are total, productive but unguarded - an interesting test-bed for our approach. Finally, the approach is targeting Coq but it is general enough to be immplemented in other proof assistants. An interesting candidate is Lean [8], a proof assistant from the same family as Coq, which to our best knowledge does not have include coinductive types nor corecursive functions, but does heave quotient types, which our approach would definitely benefit from.

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] Franz Baader & Tobias Nipkow (1998): *Term Rewriting and All That*. Cambridge University Press, doi:10.1017/CBO9781139172752.

[4] 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.

[5] 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`.

[6] 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.

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

[8] : *The Lean Proof Assistant*. `https://leanprover.github.io/`.

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

[10] 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.

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

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