Normal forms, linearity, and prime algebraicity over nonflat domains

Basil A. Karádais

DRAFT OF 17 MAY 2017, 3:02 P.M.

Abstract

Using representations of nonflat Scott domains to model type systems, it is natural to wish that they be "linear", in which case the complexity of the fundamental test for entailment of information drops from exponential to linear, the corresponding mathematical theory becomes much simpler, and moreover has ties to models of computation arising in the study of sequentiality, concurrency, and linear logic. Earlier attempts to develop a fully nonflat semantics based on linear domain representations for a rich enough type system allowing inductive types, were designed in a way that felt rather artificial, as it featured certain awkward and counter-intuitive properties; eventually, the focus turned on general, nonlinear representations.

Here we try to turn this situation around, by showing that we can work linearly in a systematic way within the nonlinear model, and that we may even restrict to a fully linear model whose objects are in a bijective correspondence with the ones of the nonlinear and are easily seen to form a prime algebraic domain. To obtain our results we study mappings of finite approximations of objects that can be used to turn approximations into normal and linear forms.

1 Introduction

When we think about computability with an eye to actual practice, we strive to reason as finitarily as it gets. In domain-theoretic denotational semantics, we understand a higher-order program through a collection of *approximations*, that is, partial descriptions of its input-output behavior, embodying consistent and complete information about it. Trying to be as finitary as possible, we base our model on truly *finite* approximations of programs, that is, finite sets of information tokens, and then work with an appropriate domain representation, as pioneered by Dana Scott [29]: two finite approximations may give consistent information, and the information of one may entail the information of the other. The denotation of a program is then retrieved as a consistent and deductively closed set of tokens. Starting from consistent finite approximations, the *formal neighborhoods*, we obtain the domain-theoretical compact elements by taking their deductive closures, and a topological basis by taking their upper cones.

A type system in this context is set up over inductively generated *Scott information systems* serving as interpretations for the base types, where the tokens—the atomic approximations that make up formal neighborhoods—are generated as a free algebra by constructors. One of the fundamental choices here concerns how exactly we want to let partiality enter the model. A fairly mainstream approach is to simply introduce *partiality as a pseudotoken*, that in particular does not participate in the generation of the other tokens, and so to end up with *flat domains* as base types. The semantics here is “strict”: a constructor induces a non-injective mapping and different constructors have overlapping ranges. Nevertheless, as is well known, the resulting model was refined enough to start off the study of deep questions, like sequentiality, or full abstraction for PCF, in a fruitful way [22, 16, 5].

Another approach, which has been drawing growing interest in the recent past—a fact reflected for example on the advent of realistic non-strict programming languages, like Haskell—is to introduce *partiality as a pseudoconstructor*, therefore allowing it to participate in the generation of the rest of the tokens, and ending up with *nonflat domains* as base types. The constructors regain their injectivity and their non-overlapping ranges, at the cost of making us work, already at base types, within a nontrivial, and in fact pretty involved, preordered set instead of a flat tree. Things get very combinatorial very quickly, and old answers may need novel tools to reestablish in the nonflat case, especially if we’re interested in a constructive development.

In particular, a strand of this kind of research tries to exploit the extra structure, hoping for results that wouldn’t hold in the flat case: already in the early 90’s, Martín Escardó [9] shows that characteristic functions which fail to be computable in the flat domain of natural numbers, become computable when elevated to the corresponding nonflat domain. Another strand, advanced by the Munich logic group, involves adapting fundamental results, like computational adequacy, definabilility, and density, to the nonflat case [27, 24, 12, 13] (see also [28]), and also recasting previous approaches to important topics in the nonflat setting in an arguably more natural way—a recent example being work on exact real arithmetic by exploiting the inherent base-type nonstrictness coinductively [17, 18]. It turns out that these two strands occasionally meet: Davide Rinaldi and the author have independently observed that the typical Berger-like argument for the Kleene–Kreisel density theorem [4], can be recast in the nonflat setting in a way that provides finite witnesses [15, 25]. The pages that follow present some observations and techniques concerning formal neighborhoods, in type systems interpreted upon nonflat information systems, and have been largely developed to help attack certain general questions like the above.

Such general questions we leave though for follow-up work. Here we choose to present the material largely as a story about *linearity*, which has also been called *atomicity*: the entailment of a token by a neighborhood depends on only one token in the neighborhood. This might sound an esoteric subject of study, but as we will recount in section 5, linearity is tied quite naturally to the research on such topics as sequentiality, concurrency, and also linear logic, since it is a statement of *linearity of entailment* brought down to the level of domain representations. The practical significance of this property is already easy to appreciate: since only *singletons* of a neighborhood U may be tested against a given token for entailment, the brute-force entailment test under linearity features linear complexity, in contrast to the generally exponential complexity of having to test potentially *all* subneighborhoods of U. Moreover, a more terse logical representation of domains suggests itself, where entailment appears as a binary predicate. In the way of a concrete and deeper connection to the subjects mentioned above, we will also see how linearity gives rise to *prime algebraic domains*.

Our goal then is to develop tools which will help us expose the intrinsic linearity properties of the information systems at hand and indeed help us show that the respective domains are prime algebraic. We begin in section 2 with some basic facts regarding our chosen model. In section 3 we revisit the notion of *neighborhood mapping* [13] and discuss its appropriate notion of continuity. Then, in section 4, we use *information-preserving* neighborhood mappings, which, additionally, send equivalent arguments to the same value, thus providing us with *normal forms*. Starting gently with nonparametric finitary base types, we discuss four distinct normal forms, namely, the straightforward supremum and deductive closure, together with a *path* and a *tree form*. Then we move on to higher types, where we introduce a streamlined version of *eigen-neighborhood* [13], and use it to establish a *higher-type normal form theorem*: normal forms at lower types induce normal forms at higher types. We conclude the section by developing yet further the basics of neighborhood mappings in order to prove a *normal form theorem for arbitrary algebras*; this generalizes the results on nonparametric finitary base types, thus establishing the existence of normal forms at every type by virtue of mutual induction. In section 5, we use our results on neighborhood mappings and normal forms to extricate an *implicit* notion of linearity in our general model, and then show how to make this *explicit* by restricting the model to an appropriate submodel based on the idea of "paths". Finally we use this knowledge to provide a simple representation of linear coherent information systems, the *preordered tolerances*, and show that the domains induced by such systems are *prime algebraic*. We end in section 6 with a short discussion on future work.

2 Background

In this section we recount the basics of the setting on which we will base the rest. We outline our type system, which allows for parametric types and infinitary base types, we explain its interpretation through Scott information systems, and we close with some useful technical observations concerning simple base types.

Types

Our type system is based on the Schwichtenberg–Wainer approach [28, §6.1.4]. Types are built simultaneously by four basic rules, one for type parameters, one for constructor types, one for base types, and one for higher types. Let ξ and $\tau_1, \tau_2, \ldots$ be distinct type variables, where the former is to be used as a dummy variable and the latter are *type parameters*. Write $\vec{\rho} \to \sigma$ to mean $\rho_1 \to \cdots \to \rho_m \to \sigma$ (associated to the right).

- Every type parameter τ is a *type*.
- If $\tau_1, \ldots, \tau_p$ for $p \geqslant 0$ are type parameters and $\vec{\rho_1}, \ldots, \vec{\rho_n}$ for $n \geqslant 0$ are types, then

  $$\tau_1 \to \cdots \to \tau_p \to (\vec{\rho_1} \to \xi) \to \cdots \to (\vec{\rho_n} \to \xi) \to \xi$$

  is a *constructor type (of arity $p+n$)*. It is called *finitary* (possibly up to the parameters[1]) if all $\vec{\rho_i}$'s are empty for $i = 1, \ldots, n$, and *infinitary* otherwise (by convention, the case that both p and n are equal to zero yields a *nullary* constructor).
- If $\kappa_1, \ldots, \kappa_k$ are constructor types for $k > 0$ and at least one of them is nullary, then $\mu_\xi(\kappa_1, \ldots, \kappa_k)$ is a *type*. We think of such a type as an inductively defined *base type* or *algebra*, generated by *constructors* C_l corresponding to κ_l, $l = 1, \ldots, k$. If all of its constructor types are finitary then we speak of a *finitary algebra* (otherwise we call it *infinitary*), and if some of its constructor types are parametric, then we speak of a *parametric algebra*.

[1] Schwichtenberg–Wainer [28] call types that are finitary up to parameters *structure finitary*.

- If ρ, σ are types then $\rho \to \sigma$ is a *type*; these are the usual *higher types*.

Note that constructor types that involve free occurences of ξ are not official types themselves, but are just used to construct base types. Examples of base types are

- the *unit type* $\mathbb{U} := \mu_\xi(\xi)$ with a single nullary constructor,
- the *type of boolean values* $\mathbb{B} := \mu_\xi(\xi, \xi)$, with constructors for the truth $\mathtt{tt} : \mathbb{B}$ and the falsity $\mathtt{ff} : \mathbb{B}$,
- the *type of natural numbers* $\mathbb{N} := \mu_\xi(\xi, \xi \to \xi)$, with constructors for the zero $\mathtt{0} : \mathbb{N}$ and the successor $\mathtt{S} : \mathbb{N} \to \mathbb{N}$,
- the *type of (extended) derivations* $\mathbb{D} := \mu_\xi(\xi, \xi, \xi \to \xi, \xi \to \xi \to \xi)$, with constructors for an axiom $\mathtt{0} : \mathbb{D}$, another axiom $\mathtt{1} : \mathbb{D}$, a one-premise rule $\mathtt{S} : \mathbb{D} \to \mathbb{D}$, and a two-premise rule $\mathtt{B} : \mathbb{D} \to \mathbb{D} \to \mathbb{D}$ (this algebra is simple yet nontrivial enough to provide us with examples as we go along),
- and the *type of (countable) ordinal numbers* $\mathbb{O} := \mu_\xi(\xi, \xi \to \xi, (\mathbb{N} \to \xi) \to \xi)$, with constructors for the zero $\mathtt{0} : \mathbb{O}$, the successor $\mathtt{S} : \mathbb{O} \to \mathbb{O}$, and the limit $\mathtt{L} : (\mathbb{N} \to \mathbb{O}) \to \mathbb{O}$, which is a typical infinitary example.

If we allow parameters, we get a host of standard, useful parametric types, like

- *types of lists* $\mathbb{L}(\tau) := \mu_\xi(\xi, \tau \to \xi \to \xi)$ with constructors for the empty list $\mathtt{nil}_\tau : \mathbb{L}(\tau)$ and the prepending $\mathtt{cons}_\tau : \tau \to \mathbb{L}(\tau) \to \mathbb{L}(\tau)$,
- *product types* $\tau_1 \times \cdots \times \tau_p := \mu_\xi(\tau_1 \to \cdots \to \tau_p \to \xi)$ with just one constructor for p-tuples $\mathtt{T} : \tau_1 \to \cdots \to \tau_p \to \tau_1 \times \cdots \times \tau_p$,
- and *sum types* $\tau_1 + \cdots + \tau_p := \mu_\xi(\tau_1 \to \xi, \ldots, \tau_p \to \xi)$ with p constructors for the injections $\mathtt{I}_i : \tau_i \to \tau_1 + \cdots + \tau_p$ for $i = 1, \ldots, p$.

Such parametric types will play an important role when we discuss recursive neighborhood mappings in §4.3. By convention, we will use ι, η to denote arbitrary *base* types and ρ, σ to denote arbitrary types in general.[2]

Information systems

A *(Scott) information system* is a triple $(\mathrm{Tok}, \mathrm{Con}, \vdash)$, where Tok is a countable set of *tokens*, Con is a collection of finite sets of tokens, which we call *consistent sets* or *(formal) neighborhoods*, and $\vdash$ is a subset of $\mathrm{Con} \times \mathrm{Tok}$, the *entailment*. These are subject to the axioms

$$
\begin{aligned}
&\{a\} \in \mathrm{Con}, \\
&U \subseteq V \wedge V \in \mathrm{Con} \to U \in \mathrm{Con}, \\
&U \in \mathrm{Con} \wedge a \in U \to U \vdash a, \\
&U \vdash V \wedge V \vdash c \to U \vdash c, \\
&U \in \mathrm{Con} \wedge U \vdash b \to U \cup \{b\} \in \mathrm{Con},
\end{aligned}
$$

[2] This type system spans the whole spectrum of the type system of Schwichtenberg–Wainer [28], barring *mutually* (or *simultaneously*) *defined algebras*. It should be straightforward, albeit technical, to extend the results that follow to the case of mutually defined algebras. Here we omit them for the sake of simplicity.

where $U \vdash V$ stands for $U \vdash b$ for all $b \in V$. From the latter follows vacuously that $U \vdash \varnothing$ for all U, while $\varnothing \in \mathrm{Con}$ follows from the first two axioms, assuming inhabitedness of Tok. Occasionally we will use finite sets Γ of tokens that are not necessarily pairwise consistent, for which we write $\Gamma \in \mathrm{Fin}$, so $\mathrm{Con} \subseteq \mathrm{Fin}$. An information system is called *coherent* when in addition to the above it satisfies

$$\forall_{a,a' \in U} \{a,a'\} \in \mathrm{Con} \to U \in \mathrm{Con} \tag{1}$$

for all $U \in \mathrm{Fin}$. By the coherence and the second axiom above, it follows that the consistency of a token set is equivalent to the consistency of its pairs. Drawing on this property, we often write $a \asymp b$ for $\{a,b\} \in \mathrm{Con}$, and even $U \asymp V$ for $U \cup V \in \mathrm{Con}$.[3] In this paper we generally restrict our attention to coherent systems even if we don't mention it explicitly.

Given two coherent information systems ρ and σ,[4] we form their *function space* $\rho \to \sigma$: define its tokens by $\langle U,b\rangle \in \mathrm{Tok}$ if $U \in \mathrm{Con}_\rho$ and $b \in \mathrm{Tok}_\sigma$, its consistency by $\langle U,b\rangle \asymp \langle U',b'\rangle$ if $U \asymp_\rho U'$ implies $b \asymp_\sigma b'$, and its entailment by $W \vdash \langle U,b\rangle$ if $WU \vdash_\sigma b$, where

$$b \in WU := \exists_{U' \in \mathrm{Con}_\rho} \left(\langle U',b\rangle \in W \wedge U \vdash_\rho U'\right).$$

The last operation is called *neighborhood application*; one can show that it is monotone in both arguments, that is, that $U \vdash U'$ implies $WU \vdash WU'$ and that $W \vdash W'$ implies $WU \vdash W'U$, for all appropriate U,U',W,W'. The proof of the following can be found in [28, §6.1.6].

Fact 2.1. *The function space of two coherent information systems is itself a coherent information system.*

Information systems as representations of domains

An *ideal* (also *element* or *point*) of an information system ρ is a possibly infinite token set $x \subseteq \mathrm{Tok}$, such that $U \in \mathrm{Con}$ for every $U \subseteq_f x$ (consistency), and $U \vdash b$ for some $U \subseteq_f x$ implies $b \in x$ (deductive closure). If x is an ideal of ρ, we write $x \in \mathrm{Ide}_\rho$ or $x : \rho$. Note that in the generic setting, as for example in [30], where partiality is introduced by an extra token rather than by an extra constructor, the empty set is always an ideal of ρ, and is the natural candidate for the role of the bottom element, denoted by $\bot$.[5]

By a *(Scott–Ershov) domain* (with a countable basis) we mean a directed complete partial order that is additionally algebraic and bounded complete [1, 30, 2]. It is furthermore *coherent* [23, 21], if every set of compacts has a least upper bound exactly when each of its *pairs* has a least upper bound. Write $b \in \overline{U}$ if and only if $U \vdash b$ (in the generic setting we have $\overline{\varnothing} = \varnothing = \bot$). The following fact, based directly on the work of Scott [29], is fundamental to our approach (for the proofs see [30, §6.1] and [14, Theorem 8]).

Fact 2.2 (Representation theorem)**.** *Let* $\rho = (\mathrm{Tok}_\rho, \mathrm{Con}_\rho, \vdash_\rho)$ *be a coherent information system. Then* $(\mathrm{Ide}_\rho, \subseteq, \bot_\rho)$ *is a coherent domain with compacts given by*

[3] Instead of “$U \asymp V$”, the notation “$U \uparrow V$” is also used.

[4] With this notation we already anticipate the use of information systems as an interpretation of our type system, but for the moment the correspondence is not yet official.

[5] Shortly we will discuss our intepretation of types and we will see that, on this point, our setting diverges from the generic one, but without affecting what follows here.

$\{\overline{U} \mid U \in \mathrm{Con}_\rho\}$. *Conversely, every coherent domain can be represented by a coherent information system.*

An *approximable mapping* between two information systems ρ and σ is a relation $r \subseteq \mathrm{Con}_\rho \times \mathrm{Con}_\sigma$, that generalizes entailment in the following sense: $\langle \varnothing, \varnothing \rangle \in r$; if $\langle U, V_1 \rangle, \langle U, V_2 \rangle \in r$ then $\langle U, V_1 \cup V_2 \rangle \in r$; and if $U \vdash_\rho U'$, $\langle U', V' \rangle \in r$, and $V' \vdash_\sigma V$, then $\langle U, V \rangle \in r$. One can show [29] that there is a bijective correspondence between the approximable mappings from ρ to σ and the ideals of the function space $\rho \to \sigma$, and moreover establish the categorical equivalence between domains with Scott continuous functions and information systems with approximable mappings. The equivalence is preserved if we restrict ourselves to the coherent case [14].

Information systems as interpretations of types

We assign an information system to each type. Every higher type is naturally assigned a function space, so it suffices to discuss the information systems for base types, that is, for algebras. The definition that follows is to be understood as a mutual induction, since we allow for constructors with infinitary arguments.

We equip every algebra ι (having at least one nullary constructor) with an extra nullary pseudoconstructor $*_\iota$ (or just $*$) to denote partiality, ending up with what one could call *algebra with partiality* or $*$-*algebra*; for example, as we will readily see, the natural numbers for us, besides `0` and `S`, will always employ the partiality pseudoconstructor $*_\mathbb{N}$ too, thus comprising the tokens $*_\mathbb{N}$, `0`, $\mathtt{S}*_\mathbb{N}$, `S0`, $\mathtt{SS}*_\mathbb{N}$, `SS0` and so on. Since this is here a universal demand, we allow ourselves to focus on the proper constructors and keep partiality pseudoconstructors tacit when we work with algebras, and also to suppress the use of new terminology: we keep writing, e.g., $\mu_\xi(\xi, \xi \to \xi)$ instead of $\mu_\xi(\xi, \xi, \xi \to \xi)$ for $\mathbb{N}$ and we keep saying "algebra" where we actually mean "algebra with partiality"—just as we often say "algebra" where we actually mean "pointed algebra" (algebra with an extra element for the bottom) in the traditional flat setting. Our treatment of partiality marks a small but rather fundamental difference between our approach and that of Schwichtenberg–Wainer. In their exposition, the pseudotoken $*$ is a special *untyped* symbol, which is allowed to occur in place of a token at *any* base type, whereas here we demand that it be *typed* like any other term.

In the following, we rather loosely speak of "an r-ary constructor C" for a given algebra ι, to mean a constructor of some arity $\rho_1, \dots, \rho_r$, corresponding to the constructor type of C; some of these types may be parameters, or recursive calls (we give this correspondence rigorously in §4.3; following the notation we gave in the beginning of this section we should have $r = p + n$). A constructor may also be nullary (in particular it may be the pseudoconstructor $*_\iota$), in which case it will take no arguments.

- If C is an r-ary constructor for ι and $a_i \in \mathrm{Tok}_{\rho_i}$ for $i = 1, \dots, r$ then $Ca_1 \cdots a_r \in \mathrm{Tok}_\iota$.[6] For its *head constructor* write $\mathrm{hd}(Ca_1 \cdots a_r) = C$; for its *i-th component token* write $a(i)$, that is, $(Ca_1 \cdots a_r)(i) = a_i$ for $i = 1, \dots, r$.
- We have $a \asymp_\iota *$ and $* \asymp_\iota a$ for all $a \in \mathrm{Tok}_\iota$. Furthermore, if C is an r-ary constructor and $a_i \asymp_{\rho_i} b_i$ for $i = 1, \dots, r$ then $Ca_1 \cdots a_r \asymp_\iota Cb_1 \cdots b_r$. Finally, we have $U \in \mathrm{Con}_\iota$ if $a \asymp_\iota a'$ for all $a, a' \in U$.
- We have $U \vdash_\iota *$ for all $U \in \mathrm{Con}_\iota$. Furthermore, if C is an r-ary constructor, every $U_i \in \mathrm{Con}_{\rho_i}$ is inhabited and $U_i \vdash_{\rho_i} b_i$ for $i = 1, \dots, r$, then $U \vdash_\iota Cb_1 \cdots b_r$

[6] For typographical convenience, throughout the text, and particularly in the several involved examples that follow, we often adopt the polish notation when we write down tokens.

for all $U \in \mathrm{Con}_\iota$ that are *sufficient for C on $U_1, \ldots, U_r$*, in the sense that for each $i = 1, \ldots, r$ and each $a_i \in U_i$ there exists an $a \in U$ such that $\mathrm{hd}(a) = C$ and $a(i) = a_i$. Finally, if $U \vdash_\iota b$, then also $U \cup \{*\} \vdash_\iota b$.

There is a subtle peculiarity in the above: the definition of the predicate Con_ι incorporates coherence (1), so it follows that $\varnothing \vdash_\iota \{*\}$, which not only diverges from the model employed in [28] but also from the generic domain-theoretic setting that we saw above. The behavior of the empty neighborhood does not get too peculiar though. Firstly, the inhabitedness of U_i's in the last clause forbids the situation $\varnothing \vdash_\iota Cb_1 \cdots b_r$ for proper constructors C, so the common intuition for both $\varnothing$ and $*$ as standing for "least information" is faithfully backed up by the definition. Moreover, it is easy to see that Fact 2.2 continues to hold, as long as we let the singleton $\{*_\iota\}$ play the role of the bottom element $\bot_\iota$ at base types and the set $\{\langle U, b\rangle \mid U \in \mathrm{Con}_\rho \wedge b \in \bot_\sigma\}$ play the role of the bottom element $\bot_{\rho\to\sigma}$ at higher types (at every type, the empty set is not even an ideal anymore). We call tokens and neighborhoods of these bottom ideals *trivial* or *uninformative*, and we reserve the term *empty* for the uninhabited set; so trivial neighborhoods at type ρ are neighborhoods $U \in \mathrm{Con}_\rho$ such that $\varnothing_\rho \vdash_\rho U$, and likewise for trivial tokens. We write Con^1_ρ and Con^i_ρ for the inhabited and the informative neighborhoods at type ρ respectively.[7]

Concerning sufficiency, note that (a) in case C is a proper constructor, U is sufficient for C on $U_1, \ldots, U_r$ if and only if $U \cup \{*\}$ is, if and only if $U \setminus \{*\}$ is, and (b) we trivially have $U \vdash_\iota CU_1 \cdots U_r$, whenever U is sufficient for C on $U_1, \ldots, U_r$; here the *constructor application* is defined by

$$CU_1 \cdots U_r := \{Ca_1 \cdots a_r \mid a_1 \in U_1, \ldots, a_r \in U_r\},$$

which is consistent if and only if every U_i is consistent at its respective type. We can also achieve the other direction, in the sense that every neighborhood U that is nontrivial is equivalent to one of the form $CU_1 \cdots U_r$: if

$$U \setminus \{*\} = \{Ca_{11} \cdots a_{r1}, \ldots, Ca_{1m} \cdots a_{rm}\},$$

we gather all i-th component tokens into a neighborhood, the *i-th component neighborhood* $U(i) := \{a_{i1}, \ldots, a_{im}\}$ of U, and let $U_i := U(i)$ for every $i = 1, \ldots, r$; then we indeed have $U \sim_\iota CU_1 \cdots U_r$ (where $U \sim V$ abbreviates $U \vdash V \wedge V \vdash U$). We call this the *component form* of U.

It is straightforward, but tedious, to check that all these make sense.

Fact 2.3. *Let ι be an algebra given by constructors. The triple $(\mathrm{Tok}_\iota, \mathrm{Con}_\iota, \vdash_\iota)$ is a coherent information system (up to the parameters).*

Elementary facts concerning nonparametric finitary base types

The interpretation of any type system such as the above starts with the values of *nonparametric* and *finitary* algebras (like $\mathbb{N}$ and $\mathbb{B}$). Whenever the need arises for an infinitary algebra (like $\mathbb{O}$), the necessary higher type values appearing as argument types at constructors (like the type $\mathbb{N} \to \mathbb{O}$ for the limit ordinal constructor L) have to be constructed on the fly using already constructed types. Finally, every time that a

[7] Recall that in the original definition by Scott [29], a trivial token, denoted by "Δ", was demanded to exist in *every* information system; here we have a single trivial token at every base type and several such at higher types.

type parameter is involved, it is clear that whenever it is replaced, it should be, again, by an already constructed type. We will spell all this out rigorously when we'll need it in §4.3, but it should already be clear that the case of nonparametric finitary algebras lies at the foundation of our whole edifice; for these we gather here some basic tools.

At nonparametric finitary base types, if U is a consistent finite set, then also its deductive closure $\overline{U}$ is clearly consistent and, thanks to the absence of infinitary arguments to the constructors, finite too (to see this we can use the component form $CU_1 \cdots U_r$ of U and reason inductively). This fact will provide us with perhaps the easiest normal form for nonparametric finitary base types in section 4.1.

A short discussion of antisymmetry may further motivate the pursuit of normal forms of section 4 in general. By straightforward induction on the generation of tokens we can show that nonparametric finitary base types are antisymmetric *on tokens* (write $a \vdash_\rho b$ for $\{a\} \vdash_\rho b$).

Lemma 2.4 (Antisymmetry)**.** *Let ι be a nonparametric finitary algebra. For all tokens $a, b \in \mathrm{Tok}_\iota$, if $a \sim_\iota b$ then $a = b$.*

On the other hand, it is easy to see that nonparametric finitary base types are not antisymmetric *on neighborhoods*, since for example $\{\mathtt{S}{*}\} \sim_{\mathbb{N}} \{\mathtt{S}{*}, *\}$ and $\{\mathtt{B0}{*}, \mathtt{B}{*}\mathtt{1}\} \sim_{\mathbb{D}} \{\mathtt{B01}\}$. It follows that antisymmetry does not carry over to higher types, either for tokens or for neighborhoods, and therefore not to infinitary algebras either.

Now for some necessary technicalities. Let ι be any nonparametric finitary algebra. Define the *height* $|a|$ of $a \in \mathrm{Tok}_\iota$ by

$$
\begin{aligned}
|*_\iota| &:= 0, \\
|Ca_1 \cdots a_r| &:= 1 + \max\{|a_1|, \ldots, |a_r|\},
\end{aligned}
$$

and the *size* $\|a\|$ of $a \in \mathrm{Tok}_\iota$ to be the number of its proper constructors:

$$
\begin{aligned}
\|*\| &:= 0, \\
\|Ca_1 \cdots a_r\| &:= 1 + \|a_1\| + \cdots + \|a_r\|.
\end{aligned}
$$

Lemma 2.5. *Let ι be a nonparametric finitary algebra and $a, b, a_1, \ldots, a_r \in \mathrm{Tok}_\iota$.*

1. *If $a \vdash_\iota b$ then $|a| \geqslant |b|$ and $\|a\| \geqslant \|b\|$.*
2. *We have $|a| \leqslant \|a\|$. Moreover, we have $|a| = \|a\|$ if and only if*
$$\forall_{b \in \mathrm{Tok}_\iota} \big(|b| = |a| \to \|b\| \geqslant \|a\| \big). \tag{2}$$
3. *Let $m = \max\{|a_1|, \ldots, |a_r|\}$. If $m = \|a_1\| + \cdots + \|a_r\|$, then $|a_i| = \|a_i\|$ for all i among $1, \ldots, r$. Moreover, if $m > 0$, then there exists a unique i among $1, \ldots, r$, such that*
$$|a_i| = \|a_i\| = m \ \wedge \ \forall_{j \neq i} |a_j| = \|a_j\| = 0.$$

Proof. The formulas in 1 are shown by straightforward induction, as well as that $|a| \leqslant \|a\|$ in 2. We show that $|a| = \|a\|$ if and only if (2) holds. From left to right, let a be a token with height and size equal, and let b be such that $|b| = |a|$. We have

$$\|b\| \geqslant |b| = |a| = \|a\|.$$

For the other way around, if $a = *$, then it's immediate, while for $a = Ca_1 \cdots a_r$ with $|a| = n > 0$, consider the token $b_{C,n}$ defined inductively as follows:

$$b_{C,1} := C \underbrace{* \cdots *}_{r},$$
$$b_{C,n+1} := Cb_{C,n} \underbrace{* \cdots *}_{r-1};$$

we have $|b_{C,n}| = \|b_{C,n}\| = n$, so $\|a\| \leqslant |a|$ by (2); by the assumption we get $\|a\| = |a|$.

We show 3 by cases on m. If $m = 0$, then for all $i = 1, \ldots, r$ we have $|a_i| = 0$, that is, $a_i = *$, so also $\|a_i\| = 0$. If $m > 0$, then there are $i = 1, \ldots, r$, for which $|a_i| = m$; assume there are k such a_i's ($k > 0$), and let l be the sum of the heights of the rest, that is, of all a_j's with $|a_j| \neq m$; by (2) we have

$$\|a_1\| + \cdots + \|a_r\| \geqslant k \cdot m + l;$$

by the assumption we get $m \geqslant k \cdot m + l$, from which we obtain $k = 1$ and $l = 0$; which is exactly what we wanted. ☐

3 Neighborhood mappings

By way of heuristics, we'd rather avoid working with the whole class of approximable maps between two (coherent) information systems. The reason is that we would like to spare ourselves the trouble of having to check after the fact if the maps that we used were “finitary” enough. Instead, we concentrate on mappings that operate on finite sets and seem to fit our setting more naturally. The notion of “neighborhood mapping” will be our central tool in what follows, and clearly one could have a lot to ask about it, but we will follow a lazy tactic; in this section we introduce the basics and later expand only when we need to.

A *neighborhood mapping* from type ρ to type σ is a mapping $f : \mathrm{Con}_\rho \to \mathrm{Con}_\sigma$. Such a mapping is *compatible (with equientailment)* if $f(U_1) \sim_\sigma f(U_2)$, whenever $U_1 \sim_\rho U_2$. It is *entailment-preserving* or *monotone* if $U_1 \vdash_\rho U_2$ implies $f(U_1) \vdash_\sigma f(U_2)$, and *consistency-preserving* or just *consistent* if $U_1 \asymp_\rho U_2$ implies $f(U_1) \asymp_\sigma f(U_2)$.

All three of the above notions are fundamental to our development. Compatibility with equientailment is arguably a sine qua non, but, as should be expected, it is too weak to ensure either monotonicity or consistency; for example the mapping from $\mathrm{Con}_\mathbb{B}$ to $\mathrm{Con}_\mathbb{B}$ defined by

$$\varnothing_\mathbb{B}, \{*_\mathbb{B}\} \mapsto \{\mathtt{tt}\},$$
$$\{*_\mathbb{B}, \mathtt{tt}\}, \{\mathtt{tt}\} \mapsto \{\mathtt{ff}\},$$
$$\{*_\mathbb{B}, \mathtt{ff}\}, \{\mathtt{ff}\} \mapsto \{\mathtt{tt}\},$$

(recall that all algebras for us tacitly assume a nullary pseudoconstructor that has to be accounted for) is compatible but neither monotone nor consistent. Furthermore, there are consistent mappings that are not monotone, like the mapping from $\mathrm{Con}_\mathbb{B}$ to $\mathrm{Con}_\mathbb{B}$ defined by

$$\varnothing_\mathbb{B}, \{*_\mathbb{B}\} \mapsto \{\mathtt{tt}\},$$
$$\{*_\mathbb{B}, \mathtt{tt}\}, \{\mathtt{tt}\} \mapsto \{*_\mathbb{B}\},$$
$$\{*_\mathbb{B}, \mathtt{ff}\}, \{\mathtt{ff}\} \mapsto \{*_\mathbb{B}\},$$

and, moreover, there are consistent mappings that are not even compatible (see example below). Not surprisingly, monotone neighborhood mappings are the safest ones to work with.

Lemma 3.1. *Let* $f : \mathrm{Con}_\rho \to \mathrm{Con}_\sigma$ *be a neighborhood mapping.*

1. *It is monotone if and only if it is compatible and* $f(U_1 \cup U_2) \vdash_\sigma f(U_1) \cup f(U_2)$ *for every* $U_1, U_2 \in \mathrm{Con}_\rho$ *with* $U_1 \asymp_\rho U_2$.
2. *If it is monotone, then it is also consistent.*

Proof. For 1, assume that f is compatible and satisfies the above condition; let $U_1, U_2 \in \mathrm{Con}_\rho$ with $U_1 \vdash_\rho U_2$; then $U_1 \sim_\rho U_1 \cup U_2$, and by compatibility $f(U_1) \sim_\sigma f(U_1 \cup U_2)$, so the assumption yields $f(U_1) \vdash_\sigma f(U_2)$. Conversely, assume that f is monotone; then compatibility is immediate, and letting $U_1, U_2 \in \mathrm{Con}_\rho$ with $U_1 \asymp_\rho U_2$, we have $U_1 \cup U_2 \vdash_\rho U_i$ for $i = 1, 2$, so $f(U_1 \cup U_2) \vdash_\rho f(U_i)$, by monotonicity. The statement 2 follows immediately. □

Example. At each type ρ, easy examples of neighborhood mappings are the *identity mapping* id given by $U \mapsto U$, and the *constant mappings* given by $U \mapsto U_0$ for any fixed $U_0 \in \mathrm{Con}_\rho$; all of these are monotone.

Another example of a rather useful monotone mapping is the *partial height* mapping for nonparametric finitary base types. Consider the *token* mapping $\mathsf{ph} : \mathrm{Tok}_\mathbb{D} \to \mathrm{Tok}_\mathbb{D}$ given (intuitively) by $\mathsf{ph}(a) := \mathtt{S}^{|a|} *_\mathbb{D}$, and extend it to neighborhoods by letting

$$\begin{aligned} \varnothing_\mathbb{D} &\mapsto \{*_\mathbb{D}\}, \\ \{U\} &\mapsto \{\mathsf{ph}(a) \mid a \in U\}. \end{aligned}$$

An example of a consistent mapping that is not compatible would be the rather crude detotalizing mapping for nonparametric finitary algebras, $\mathsf{detot}(U) := U \backslash \{a \in U \mid a \text{ total}\}$ (a is *total* in a nonparametric finitary base type if it does not involve $*$); by using such a mapping we don't harm consistency, but we rather unwarrantedly *lose information*, since $\{\mathtt{S}*_\mathbb{D}, \mathtt{S}0\} \sim_\mathbb{D} \{\mathtt{S}0\}$, but $\mathsf{detot}(\{\mathtt{S}*_\mathbb{D}, \mathtt{S}0\}) = \{\mathtt{S}*_\mathbb{D}\} \not\sim_\mathbb{D} \varnothing_\mathbb{D} = \mathsf{detot}(\{\mathtt{S}0\})$. □

Ideals from neighborhood mappings

Despite the merits of monotonicity, it turns out that the weaker property of consistency suffices for a neighborhood mapping, because it is exactly what we need to naturally capture the notion of continuity for ideals.

Recall that an ideal is a possibly infinite set of tokens that is (a) consistent, that is, every two of its tokens are consistent to each other, and (b) deductively closed, that is, if some finite part of it entails a token, then this token must also belong to it. The idea of a neighborhood mapping f is obviously to achieve these two requirements by appropriately working on the level of neighborhoods: intuitively, the (right-flattened) graph of f should correspond to an ideal. To ensure that (a) holds, it is fitting that we require consistency from f, but what about (b); should we require something more? We show that we don't.

Define the *idealization* $\hat{f}$ of a neighborhood mapping $f : \mathrm{Con}_\rho \to \mathrm{Con}_\sigma$ to be the token set

$$\hat{f} := \{\langle U, b\rangle \in \mathrm{Tok}_{\rho\to\sigma} \mid \underset{U_1,\dots,U_m \in \mathrm{Con}_\rho}{\exists} \Big(U \vdash_\rho \bigcup_{j=1}^m U_j \wedge \bigcup_{j=1}^m f(U_j) \vdash_\sigma b \Big)\}.$$

Note that the term $\bigcup_{j=1}^{m} f(U_j)$ in the definition (which accounts for the essential nonlinearity of our setting) is implicitly required to be consistent—otherwise it wouldn't be allowed to appear on the left of an entailment.

Write $\langle U,V\rangle$ for $\{\langle U,b\rangle \mid b \in V\}$ (in particular, $\langle U,\varnothing\rangle = \varnothing$).

Proposition 3.2. *Let ρ, σ be types, and f be a neighborhood mapping from ρ to σ. Then $\hat{f}$ is an ideal if and only if f is consistent.*

Proof. Assume that $\hat{f}$ is an ideal, and let $U_1,U_2 \in \mathrm{Con}_\rho$, with $U_1 \asymp_\rho U_2$. Since $U_i \vdash_\rho U_i$ and $f(U_i) \vdash_\sigma f(U_i)$ for each $i = 1,2$, we have $\langle U_i, f(U_i)\rangle \subseteq \hat{f}$, by the definition of idealization, so the consistency of $\hat{f}$ yields $f(U_1) \asymp_\sigma f(U_2)$, and f is consistent.

Now assume that f is consistent. For the consistency of $\hat{f}$, let $\langle U_i,b_i\rangle \in \hat{f}$, with $U_1 \asymp_\rho U_2$. By the definition of idealization there exist $U_{11},\dots U_{1m_1},U_{21},\dots U_{2m_2} \in \mathrm{Con}_\rho$, such that

$$U_i \vdash_\rho \bigcup_{j_i=1}^{m_i} U_{ij_i} \wedge \bigcup_{j_i=1}^{m_i} f(U_{ij_i}) \vdash_\sigma b_i, \qquad (\star)$$

for each $i = 1,2$. Since U_1 and U_2 are consistent, the propagation of consistency at type ρ gives us $\bigcup_{j_1=1}^{m_1} U_{1j_1} \asymp_\rho \bigcup_{j_2=1}^{m_2} U_{2j_2}$, which in turn yields $\bigcup_{j_1=1}^{m_1} f(U_{1j_1}) \asymp_\sigma \bigcup_{j_2=1}^{m_2} f(U_{2j_2})$, due to the consistency of f; by the propagation at type σ we get $b_1 \asymp_\sigma b_2$.

For the deductive closure of $\hat{f}$, let $W \subseteq \hat{f}$ and $\langle U,b\rangle \in \mathrm{Tok}_{\rho\to\sigma}$ be such that $W \vdash_{\rho\to\sigma} \langle U,b\rangle$. By the definition of entailment, there are $\langle U_i,b_i\rangle \in W$, $i = 1,\dots,n$, such that $U \vdash_\rho \bigcup_{i=1}^{n} U_i$ and $\{b_1,\dots,b_n\} \vdash_\sigma b$. Now each $\langle U_i,b_i\rangle$ is in $\hat{f}$, so there exist neighborhoods $U_{i1},\dots,U_{im_i}$, such that $(\star)$ holds, but now for $i = 1,\dots,n$. By propagation at ρ, *all* U_{ij_i}'s are consistent; moreover, the consistency of f ensures that *all* $f(U_{ij_i})$'s are consistent; then, by the transitivity of entailment, we have

$$U \vdash_\rho \bigcup_{i=1}^{n} \bigcup_{j_i=1}^{m_i} U_{ij_i} \wedge \bigcup_{i=1}^{n} \bigcup_{j_i=1}^{m_i} f(U_{ij_i}) \vdash_\sigma b,$$

so $\langle U,b\rangle \in \hat{f}$, by the definition. □

Let us again note that not all ideals can be given by neighborhood mappings by way of Proposition 3.2: a counterexample at type $\mathbb{N} \to \mathbb{N}$ would be the "cototal" ideal given by $\{\langle \mathtt{0}, \mathtt{S}^n * \rangle \mid n = 0,1,\dots\}$ (it is possible though to pair every ideal with a *system* of monotone neighborhood mappings satisfying certain natural conditions, but we won't expand on this here as we won't need it). Having stressed that, this result would justify the term *continuous* for a *consistent* neighborhood mapping, but for reasons of clarity we will refrain from using the term.

4 Normal forms

We mentioned in the introduction that the complexity of nonflat base types can become unwieldy very early. To illustrate the point, consider the nonparametric finitary base type $\mathbb{N}$, where two neighborhoods are equivalent exactly when their highest tokens coincide, for example $\{\mathtt{SS0},\mathtt{S}*\} \sim_{\mathbb{N}} \{\mathtt{SS0},\mathtt{SS}*,*\}$. By an elementary combinatorial argument we can see that the number of equivalent neighborhoods whose highest token is some known a, is the number of all subsets of the set $\overline{a}\backslash\{a\}$; this means, for example, that merely for the natural number 9 (that is, for the numeral SSSSSSSSS0), there are already a thousand and twenty four equivalent neighborhood representations in the model. It is clear that we could use canonical ways to spot neighborhoods with the desired information, and work exclusively with them—one of these canonical forms will indeed be the singleton form $\{a\}$ (for nonparametric finitary algebras).

In our context we look at normal forms not so much as irreducible elements in a rewriting system [6], but rather as values of special consistent neighborhood *endo*mappings. Let ρ be a type; a neighborhood mapping $f : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$ is a *normal form mapping (at type ρ)* if $f(U) \sim_\rho U$ (preservation of information) and $U_1 \sim_\rho U_2$ implies $f(U_1) = f(U_2)$ (uniqueness) for all $U, U_1, U_2 \in \mathrm{Con}_\rho$. By the first requirement, it is clear that every normal form mapping is monotone, so by Lemma 3.1 also compatible and consistent. In this section we establish the following.

Theorem 4.1 (Normal forms)**.** *At every type there exists a normal form mapping.*

We begin gently, by first discussing normal form mappings at nonparametric finitary base types (§4.1). Then we develop a method of inducing normal forms at higher types, provided we have normal forms at their lower types (Theorem 4.17). Finally, we complete the picture by addressing the case of base types that may be infinitary or have parameters (Theorem 4.25). These results will prove instrumental in our way towards linearity and ultimately prime algebraicity in section 5.

4.1 Normal forms at nonparametric finitary base types

There are two normal forms at nonparametric finitary base types that are easy to spot: closures and suprema. Two slightly more intricate normal forms are "paths" and "trees"; path forms will play a crucial role in the study of linearity in section 5.2, while tree forms provide a more intuitive variation of paths.

Closures and suprema

Let ι be a nonparametric finitary algebra. Perhaps the normal forms that are easiest to recognize in this case are given by the deductive closure and the supremum.

We defined the deductive closure in section 2. Define the *supremum* $\mathsf{sup}(a,b)$ of two consistent tokens $a, b \in \mathrm{Tok}_\iota$ inductively over their structure by

$$\mathsf{sup}(a,*) = \mathsf{sup}(*,a) = a,$$
$$\mathsf{sup}(Ca_1 \cdots a_r, Cb_1 \cdots b_r) = C\mathsf{sup}(a_1,b_1)\cdots\mathsf{sup}(a_r,b_r),$$

for every constructor C of arity r. Further, for a neighborhood $U \in \mathrm{Con}_\iota$ define its *supremum* $\mathsf{sup}(U) \in \mathrm{Tok}_\iota$ by

$$\mathsf{sup}(\varnothing_\iota) := *_\iota,$$
$$\mathsf{sup}(\{a\}) := a,$$
$$\mathsf{sup}(\{a_1,\ldots,a_m\}) := \mathsf{sup}(\cdots\mathsf{sup}(a_1,a_2)\cdots,a_m).$$

Proposition 4.2 (Closure and supremum normal form)**.** *Let ι be a nonparametric finitary base type. The mappings $U \mapsto \overline{U}$ and $U \mapsto \{\mathsf{sup}(U)\}$ are normal form mappings at type ι.*

Proof. That the deductive closure provides a normal form is easy to show (recall that in nonparametric finitary algebras deductive closures are finite). We show that the supremum yields a normal form. We concentrate first on the case of two tokens $a, b \in \mathrm{Tok}_\iota$, where $a \asymp_\iota b$. If one of them is trivial, say $b = *$, then $\{\mathsf{sup}(a, *)\} = \{a\} \sim_\iota \{a, *\}$. If not, then $a = Ca_1 \cdots a_r$ and $b = Cb_1 \cdots b_r$, with $a_i \asymp_\iota b_i$ for all i's; the induction hypothesis is that $\{\mathsf{sup}(a_i, b_i)\} \sim_\iota \{a_i, b_i\}$ for all i's, so we have

$$\begin{aligned}
\{a, b\} &= \{Ca_1 \cdots a_r, Cb_1 \cdots b_r\} \\
&\sim_\iota C\{a_1, b_1\} \cdots \{a_r, b_r\} \\
&\sim_\iota C\{\mathsf{sup}(a_1, b_1)\} \cdots \{\mathsf{sup}(a_r, b_r)\} \\
&= \{C\mathsf{sup}(a_1, b_1) \cdots \mathsf{sup}(a_r, b_r)\} \\
&= \{\mathsf{sup}(a, b)\}.
\end{aligned}$$

Now let $U \in \mathrm{Con}_\iota$. If $U = \varnothing$ then $\mathsf{sup}(U) = *$. If $U = \{a_1, \ldots, a_m\}$ then

$$\begin{aligned}
U &\sim_\iota \{\mathsf{sup}(a_1, a_2), a_3, \ldots, a_m\} \\
&\sim_\iota \cdots \\
&\sim_\iota \{\mathsf{sup}(\cdots \mathsf{sup}(a_1, a_2) \cdots, a_m)\} \\
&= \{\mathsf{sup}(U)\},
\end{aligned}$$

based on the previous.

As for uniqueness, if $U_1 \sim_\iota U_2$, then $\mathsf{sup}(U_1) \sim_\iota \mathsf{sup}(U_2)$ by transitivity, and by Lemma 2.4 we get $\mathsf{sup}(U_1) = \mathsf{sup}(U_2)$. □

We may write cl for $U \mapsto \overline{U}$ and lub for $U \mapsto \{\mathsf{sup}(U)\}$.

Paths and trees

In terms of cardinality, the closure and supremum of a neighborhood would be its biggest and smallest normal form respectively. The following normal forms, “paths” and “trees”, fall in between the two. Paths will be most important for us, since we will need them in making linearity explicit and showing prime algebraicity in sections 5.2 and 5.3. The common idea behind both paths and trees is to consider the entailment diagram of the closure of a given neighborhood, and then eliminate the cycles that appear in it. We show that this does not cause information loss.

Call $a \in \mathrm{Tok}_\iota$ a *path*, and write $a \in \mathrm{Tok}_\iota^p$, if it is built inductively by the following clauses:

- $* \in \mathrm{Tok}_\iota^p$;
- if C is a constructor and $a \in \mathrm{Tok}_\iota^p$, then $C\vec{*}a\vec{*} \in \mathrm{Tok}_\iota^p$ (where the vectors $\vec{*}$ may be empty).

For example, in the algebra $\mathbb{D}$, the token B$*$(S0) is a path, whereas B0(S$*$) isn't. The choice of the name stems from the fact that a path's deductive closure contains no cycles (see Proposition 4.3.1 below). By convention, we write $C\vec{*}a^i\vec{*}$, if we want to indicate that a^i possesses the i-th position in the arity of C, that is, that $a^i = (C\vec{*}a^i\vec{*})(i)$ using the notation of section 2.

Proposition 4.3. *Let ι be a base type.*

1. *Let $a \in \mathrm{Tok}_\iota^p$ and $b_1, b_2 \in \mathrm{Tok}_\iota$. The following* path comparability *property holds: if $a \vdash_\iota b_1$ and $a \vdash_\iota b_2$, then $b_1 \vdash_\iota b_2$ or $b_2 \vdash_\iota b_1$.*

2. *Let $a \in \mathrm{Tok}_\iota^p$ and $b \in \mathrm{Tok}_\iota$. If $a \vdash_\iota b$, then $b \in \mathrm{Tok}_\iota^p$.*

3. *Let $U \in \mathrm{Con}_\iota^1$, and $b \in \mathrm{Tok}_\iota^p$. The following* path linearity *property holds: if $U \vdash_\iota b$ then there exists an $a \in U$, such that $\{a\} \vdash_\iota b$.*

Proof. For 1. By induction on a. If $a = *$ then $b_m = *$ for both $m = 1, 2$. If $a = C\vec{*}a^i\vec{*}$, with $a^i \in \mathrm{Tok}_\iota^p$, then for each $m = 1, 2$ we have $b_m = C\vec{*}b_m^i\vec{*}$, with $a^i \vdash_\iota b_m^i$. The induction hypothesis yields $b_1^i \vdash_\iota b_2^i$ or $b_2^i \vdash_\iota b_1^i$, and the definition of entailment does the rest.

For 2. By induction on the path a. If $a = *$ then also $b = *$. Let $a = C\vec{*}a^i\vec{*}$, with $a^i \in \mathrm{Tok}_\iota^p$ (for some i within the arity of C). If $b = *$ then again we have it, otherwise there exists a $b^i \in \mathrm{Tok}_\iota$ such that $b = C\vec{*}b^i\vec{*}$ and $a^i \vdash_\iota b^i$; the induction hypothesis yields $b^i \in \mathrm{Tok}_\iota^p$, so $b \in \mathrm{Tok}_\iota^p$ as well, by definition.

For 3. By induction on b. If $b = *$ then any element of U will do (there is at least one element since U is inhabited). If $b = C\vec{*}b^i\vec{*}$, with $b^i \in \mathrm{Tok}_\iota^p$, then $U \backslash \{*\}$ has the form $\{C\vec{a_1}, \ldots, C\vec{a_m}\}$, where $\{a_{1i}, \ldots, a_{mi}\} \vdash_\iota b^i$, by the definition of entailment. By the induction hypothesis there exists a $j = 1, \ldots, m$, such that $\{a_{ji}\} \vdash_\iota b^i$; it follows that $\{C\vec{a_j}\} \vdash_\iota b$. □

A nice characterization of paths comes from the minimality of their size.

Proposition 4.4. *A token $a \in \mathrm{Tok}_\iota$ is a path if and only if it has minimal size for its height, that is, if $|a| = \|a\|$.*

Proof. From left to right, let $a \in \mathrm{Tok}_\iota^p$. By induction on the information of a. If $a = *$, then both its height and its size are zero by definition. If $a = C\vec{*}b\vec{*}$ for some constructor C and $b \in \mathrm{Tok}_\iota^p$, then the induction hypothesis gives us $|b| = \|b\| = m$ for some $m \geqslant 0$; by the definition of height and size we obtain $|a| = 1 + m = \|a\|$.

For the other way around, let a be such that $|a| = \|a\| = m$ for some $m \geqslant 0$. We perform induction on m. For $m = 0$, we have $a = *$, which is a path by definition. For $m + 1 \geqslant 0$, we have $a = Ca_1 \cdots a_r$ with

$$m = \max\{|a_1|, \ldots, |a_r|\} = \|a_1\| + \cdots + \|a_r\|;$$

by Proposition 2.5.3, we have $|a_i| = \|a_i\| =: m_i$ for all $i = 1, \ldots, r$, and either $m = m_i = 0$, so $a = C\vec{*}$, or else there is exactly one i such that $m_i > 0$, so $a = C\vec{*}a_i\vec{*}$, with $a_i \in \mathrm{Tok}_\iota^p$ by the induction hypothesis; in both cases we have $a \in \mathrm{Tok}_\iota^p$ by definition. □

It turns out that paths facilitate a natural notion of “irredundant normal form” for neighborhoods: these are normal forms U where every inhabited subneighborhood is maximal within U, in the sense that it includes what it entails. If $U \in \mathrm{Con}_\rho$, its *maximal elements* $\mathsf{irr}(U)$ are those tokens $a \in U$, such that if $a' \in U$ is some other token with $a' \vdash_\rho a$, then $a' \sim_\rho a$. Say that a neighborhood U is *path reduced*, and write $U \in \mathrm{Con}_\iota^{pr}$, if every token in it is a path and is maximal in U. For example, $\{\mathtt{B00}\}$ and $\{\mathtt{B0*}, \mathtt{B*0}, \mathtt{B**}\}$ are not path reduced, but $\{\mathtt{B0*}, \mathtt{B*0}\}$ is.

Proposition 4.5 (Irredundancy)**.** *If $U \in \mathrm{Con}_\iota^{pr}$, $U_0 \subseteq U$ is inhabited and $a \in U$ then $U_0 \vdash_\iota a$ implies $a \in U_0$.*

Proof. Let $U \in \mathrm{Con}_\iota^{pr}$, $U_0 \subseteq U$ an inhabited subneighborhood, and $a \in U$. Since U consists of paths, Proposition 4.3.3 yields a single $a' \in U_0$ such that $a' \vdash_\iota a$. But U is reduced, so $a' = a$. □

A *path form* of a neighborhood is an equivalent neighborhood that is path reduced; for example, the finite set $\{\mathtt{B0*},\mathtt{B*0}\}$ in $\mathbb{D}$ is a path form of both $\{\mathtt{B00}\}$ and $\{\mathtt{B0*},\mathtt{B*0},\mathtt{B**}\}$.

Theorem 4.6 (Path normal form)**.** *Let ι be a nonparametric finitary base type. There exists a normal form mapping* $\mathsf{pth}:\mathrm{Con}_\iota\to\mathrm{Con}_\iota$ *such that for every $U\in\mathrm{Con}_\iota$ we have* $\mathsf{pth}(U)\in\mathrm{Con}_\iota^{pr}$.

Proof. We first consider path forms for tokens $a\in\mathrm{Tok}_\iota$ (and thus cover the singleton finite sets). Let $\mathsf{pth}^t:\mathrm{Tok}_\iota\to\mathrm{Fin}_\iota$ be the mapping defined recursively by the clauses

$$\mathsf{pth}^t(C)=\{C\}\quad\text{for } C \text{ nullary},$$
$$\mathsf{pth}^t(Ca_1\cdots a_r)=\bigcup_{i=1}^{r}C\overrightarrow{\{*\}}\mathsf{pth}^t_*(a_i)\overrightarrow{\{*\}},$$

where $\mathsf{pth}^t_*(a):=\mathsf{pth}^t(a)\setminus\{*\}$ (we always have $\mathsf{pth}^t_*(a)=\mathsf{pth}^t(a)$, when $a\neq *$). Note the use of the constructor neighborhood mapping (or else, the “constructor application”) in the inductive clause, and that in the first clause the pseudoconstructor $*$ is counted in.

It is straightforward to see that if $b\in\mathsf{pth}^t(a)$ then $b\in\mathrm{Tok}_\iota^p$. Such a b must also be maximal: if $a=*$, then also $b=*$; otherwise $a=Ca_1\cdots a_r$ and $b=C\vec{*}a^i\vec{*}$ for some $i=1,\dots,r$ and $a^i\in\mathsf{pth}^t(a_i)$ with $a^i\neq *$; assuming that there is a $b'\in\mathsf{pth}^t(a)$, such that $b'\vdash_\iota b$, it follows that $b'=Cb_1\cdots b_i\cdots b_r$, with $b_i\vdash_\iota a^i$; by the construction of pth^t and the induction hypothesis for $\mathsf{pth}^t(a_i)$, it must be $b_j=*$ for all $j\neq i$, and $b_i=a^i$, so $b'=b$. We’ve shown then that $\mathsf{pth}^t(a)$ is path reduced for every $a\in\mathrm{Tok}_\iota$. The preservation of information follows from the induction hypotheses $\mathsf{pth}^t(a_i)\sim_\iota\{a_i\}$ for each $i=1,\dots,r$, and the definition of entailment. As for uniqueness, it follows immediately from Lemma 2.4, since $a\sim_\iota b$ implies $a=b$, so $\mathsf{pth}^t(a)=\mathsf{pth}^t(b)$.

Moving on to neighborhoods $U\in\mathrm{Con}_\iota$, we may set $\mathsf{pth}(U):=\mathsf{pth}^t(\mathsf{sup}(U))$; this is a normal form mapping by the previous and Proposition 4.2. □

Remark. In previous approaches to normal forms [26, 13], the restriction to nonsuperunary constructors and binary entailment made it possible to avoid paths and obtain a normal form directly from the mapping $U\mapsto\mathsf{irr}(U)$, which was both “linear”, in the sense that $\mathsf{irr}(U)\vdash b$ implied $\{a\}\vdash b$ for some $a\in\mathsf{irr}(U)$ (see section 5), and irredundant in the sense of Proposition 4.5. This mapping doesn’t work in the general case of an algebra with superunary constructors and a full entailment predicate. Take for example the neighborhood $U=\{\mathtt{BB00*},\mathtt{BB0*0},\mathtt{BB*00}\}$, for which it already holds that $U=\mathsf{irr}(U)$; this is neither linear (it entails $\mathtt{BB000}$ with any two tokens, but not with any single one of them) nor irredundant (for the subset $U_0:=\{\mathtt{BB00*},\mathtt{BB0*0}\}$ we have $U_0\vdash_{\mathbb{D}}\mathtt{BB*00}$ but $\mathtt{BB*00}\notin U_0$, so it is not maximal within U). Moreover, even if we restricted entailment to its binary version and we tried to find *subsets* of U that do satisfy Proposition 4.5 and are themselves linear and irredundant, we would actually find three: every pair of tokens in U forms such a neighborhood; but there would be no natural reason to prefer one over the other as a normal form in order to have uniqueness. □

From the path form of a neighborhood we can easily obtain its “tree form” by taking atomic closures. Call a neighborhood $U\in\mathrm{Con}_\iota$ a *(full) tree*, and write $U\in\mathrm{Con}_\iota^{tr}$, if for every $a\in U$ we have $a\in\mathrm{Tok}_\iota^p$ and $\overline{a}\subseteq U$ (the name is justified by Proposition 4.3.1). For example, the neighborhood $\{\mathtt{B0*},\mathtt{B*0},\mathtt{B00}\}$ is no tree, but $\{\mathtt{B0*},\mathtt{B*0},\mathtt{B**},*\}$ is, consisting of the closures of the paths B0$*$ and B$*$0. So *the root* of the trees that we consider is always $*$ (that’s why we think of them as “full”), while their *leaves* are simply their tokens of maximal information. A *tree form* of a neighborhood is an equivalent neighborhood that is a tree; for example, a tree-form of $\{\mathtt{B0*},\mathtt{B*0},\mathtt{B00}\}$ is $\{\mathtt{B0*},\mathtt{B*0},\mathtt{B**},*\}$—actually, the only one.

Proposition 4.7 (Tree normal form)**.** *Let ι be a nonparametric finitary base type. There exists a normal form mapping* $\mathtt{tr} : \mathrm{Con}_\iota \to \mathrm{Con}_\iota$ *such that for every* $U \in \mathrm{Con}_\iota$ *we have* $\mathtt{tr}(U) \in \mathrm{Con}_\iota^{tr}$.

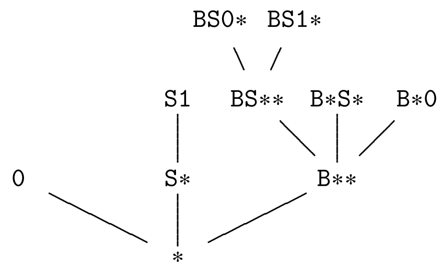
Proof. Let $U \in \mathrm{Con}_\iota$. By Theorem 4.6 we can assume that U is path reduced. Set $\mathtt{tr}(U) := \bigcup_{a\in U} \overline{a}$. It is clear by the construction that this is a tree. The preservation of information and the uniqueness are both straightforward. □

Remark. The tree form of a neighborhood can of course be generated without appeal to its path form. For tokens we can first set

$$\mathtt{tr}(Ca_1 \cdots a_r) := \{*\} \cup \bigcup_{i=1}^{r} \{C \overrightarrow{*} a_{0i} \overrightarrow{*} \mid a_{0i} \in \mathtt{tr}(a_i)\},$$

and then $\mathtt{tr}(U) := \bigcup_{a\in U} \mathtt{tr}(a)$. □

Example. At type $\mathbb{D}$, the singleton $\{\mathtt{0}\}$ has tree form $\{*,\mathtt{0}\}$, and the singleton $\{\mathtt{S1}\}$ has tree form $\{*,\mathtt{S*},\mathtt{S1}\}$. The singleton $\{\mathtt{B(S0)(S*)}\}$, which involves a binary constructor, has tree form $\{*,\mathtt{B**},\mathtt{B(S*)*},\mathtt{B(S0)*},\mathtt{B*(S*)}\}$, and similarly the singleton $\mathtt{B(S1)0}$ has tree form $\{*,\mathtt{B**},\mathtt{B(S*)*},\mathtt{B(S1)*},\mathtt{B*0}\}$. The union of these tree forms yields the following picture.



□

4.2 Normal forms at higher types

In this section we turn to higher types built over *arbitrary* (not necessarily nonparametric and finitary) base types, for which we assume that we can find normal forms. We will cover the latter in the next section, thus completing the general mutually inductive argument.

If $\Theta \in \mathrm{Fin}_{\rho\to\sigma}$ is a finite set with $\Theta = \{\langle U_j, b_j\rangle \mid j = 1,\dots,l\}$, write $L(\Theta)$ for $\bigcup_j U_j$ (notice that this is a *flattening*), and $R(\Theta)$ for $\bigcup_j \{b_j\}$. Furthermore, recall that for $U \in \mathrm{Con}_\rho$ and $\Delta \in \mathrm{Fin}_\sigma$, we write $\langle U,\Delta\rangle$ for $\{\langle U,b\rangle \mid b \in \Delta\}$; for nontrivial V'’s we have $\langle U,V\rangle \vdash_{\rho\to\sigma} \langle U',V'\rangle$ if and only if $U' \vdash_\rho U$ and $V \vdash_\sigma V'$ (this is not true for trivial ones, for example we have $\langle\{\mathtt{tt}\},\mathtt{0}\rangle \vdash \langle\{\mathtt{ff}\},*\rangle$ at type $\mathbb{B} \to \mathbb{N}$). Finally, we write $U \in \mathrm{Con}_\Gamma$ for $\Gamma \in \mathrm{Fin}_\rho$ when $U \subseteq \Gamma$ and $U \in \mathrm{Con}_\rho$.

Eigen-neighborhoods

Consider a neighborhood $W = \{\langle U_j, b_j\rangle \mid j = 1,\dots l\}$ at some higher type $\rho \to \sigma$, which we apply to some information, say U, of type ρ. If this U is above U_j and U_k for some $j,k = 1,\dots l$, then (a) U_j and U_k must be consistent, and (b) both b_j and b_k will belong to the value WU. Furthermore, if U is above U_j, which in turn is above some other U_k, then (a) U must be above U_k as well, and (b) both b_j and b_k will again belong to the value WU. These two basic facts regarding application motivate the definition of the *eigen-neighborhoods* of W, Eig_W.

At a base type ι, we will use here the convention that the only eigen-neighborhoods of a neighborhood U are $\varnothing_\iota$ and U. At a higher type $\rho \to \sigma$, a neighborhood H is an eigen-neighborhood of $W \in \mathrm{Con}_{\rho\to\sigma}$ if it is of the form $H = \langle U, V\rangle$ and it features the following properties of left and right closure:

$$U = \overline{U} \cap L(W) \ \wedge \ V = \overline{WU} \cap R(W).$$

Since the first requirement implies that $U \in \mathrm{Con}_{L(W)}$, it is clear that Eig_W is a finite set of neighborhoods for every W. Let us stress that the concept of eigen-neighborhoods is *not* given inductively over types; in the following we concentrate on eigen-neighborhoods at higher types, as it's there where they prove essential.

Given $W \in \mathrm{Con}_{\rho\to\sigma}$ as above, every $U \in \mathrm{Con}_\rho$ induces an eigen-neighborhood $W\restriction_U$, the *(eigen-)restriction of* W *to* U, in the following natural way:

$$W\restriction_U := \left\langle \overline{U} \cap L(W), \overline{WU} \cap R(W)\right\rangle;$$

the eigen-neighborhood $W\restriction_U$ is basically the "support" of W with respect to U, that is, the part of W that answers to the input U.[8] The following are straightforward from the definitions and the monotonicity of neighborhood application.

Lemma 4.8. *Let* ρ, σ *be types and* $W \in \mathrm{Con}_{\rho\to\sigma}$.

1. *For all* $U \in \mathrm{Con}_\rho$, *we have* $WU \vdash_\sigma b$ *if and only if* $R(W\restriction_U) \vdash_\sigma b$.
2. *For all* $U, U' \in \mathrm{Con}_\rho$, *if* $U \vdash_\rho U'$ *then* $P(W\restriction_{U'}) \subseteq P(W\restriction_U)$, *where* P *stands for* L *and* R.
3. *We have* $H \in \mathrm{Eig}_W$ *if and only if* $H = W\restriction_{L(H)}$.

Based on 3 above, we may establish an intuition of eigen-neighborhoods as "generalized tokens". Write eig for the mapping $U \mapsto \bigcup \mathrm{Eig}_U$; trivially $U = \mathsf{eig}(U)$ for all base-type U's, whereas at higher types we have

$$\mathsf{eig}(W) := \bigcup_{U\in\mathrm{Con}_{L(W)}} W\restriction_U = \bigcup_{U\in\mathrm{Con}_{L(W)}} \left\langle \overline{U} \cap L(W), \overline{WU} \cap R(W)\right\rangle.$$

We say that $\mathsf{eig}(U)$ is the *eigenform* of U, and if $U = \mathsf{eig}(U)$, we say that U *is in eigenform.*

Example. Every base-type neighborhood is in eigenform, and every empty set $\varnothing_{\rho\to\sigma}$ is in eigenform. As a further easy example, consider a trivial higher-type neighborhood

[8] Note that this is quite different than the more modest *restriction of* W *to* U, which would be just the subneighborhood $\{\langle U', b\rangle \in W \mid U \vdash_\rho U'\}$.

$\langle U, \{*_\sigma\}\rangle$, where we write $*_{\rho\to\sigma} := \left\langle \{*_\rho\}, *_\sigma \right\rangle$; it is easy to see that every inhabited eigen-neighborhood here has the form $\left\langle \overline{U_0} \cap U, \{*_\sigma\} \right\rangle$ where $U_0 \subseteq U$.

At type $\mathbb{D} \to \mathbb{D}$ consider the neighborhood

$$W = \{\langle\{\texttt{B0}*\},\texttt{B}*\texttt{1}\rangle,\langle\{\texttt{B}*\texttt{1}\},\texttt{B0}*\rangle,\langle\{\texttt{S}*\},\texttt{S0}\rangle,\langle\{\texttt{S0}\},\texttt{S}*\rangle\}.$$

Here $L(W) = R(W) = \{\texttt{B0}*,\texttt{B}*\texttt{1},\texttt{S}*,\texttt{S0}\}$. Letting U vary over $\mathrm{Con}_{L(W)}$, we get the following table:

$\overline{U} \cap L(W)$	$\overline{WU} \cap R(W)$
$\varnothing$	$\varnothing$
$\{\texttt{B0}*\}$	$\{\texttt{B}*\texttt{1}\}$
$\{\texttt{B}*\texttt{1}\}$	$\{\texttt{B0}*\}$
$\{\texttt{B0}*,\texttt{B}*\texttt{1}\}$	$\{\texttt{B}*\texttt{1},\texttt{B0}*\}$
$\{\texttt{S}*\}$	$\{\texttt{S0}\}$
$\{\texttt{S0}\}$	$\{\texttt{S0},\texttt{S}*\}$
$\{\texttt{S}*,\texttt{S0}\}$	$\{\texttt{S0},\texttt{S}*\}$

It follows that

$$\begin{aligned}\mathsf{eig}(W) = \{&\langle\{\texttt{B0}*\},\texttt{B}*\texttt{1}\rangle,\langle\{\texttt{B}*\texttt{1}\},\texttt{B0}*\rangle,\langle\{\texttt{B0}*,\texttt{B}*\texttt{1}\},\texttt{B}*\texttt{1}\rangle,\langle\{\texttt{B0}*,\texttt{B}*\texttt{1}\},\texttt{B0}*\rangle,\\ &\langle\{\texttt{S}*\},\texttt{S0}\rangle,\langle\{\texttt{S0}\},\texttt{S0}\rangle,\langle\{\texttt{S0}\},\texttt{S}*\rangle,\langle\{\texttt{S}*,\texttt{S0}\},\texttt{S0}\rangle,\langle\{\texttt{S}*,\texttt{S0}\},\texttt{S}*\rangle\}\end{aligned}$$

is the eigenform of W. □

Proposition 4.9 (Eigenform). *Let ρ and σ be types, and $W, W_1, W_2 \in \mathrm{Con}_{\rho\to\sigma}$.*

1. *We have $W \sim_{\rho\to\sigma} \mathsf{eig}(W)$. Moreover, $\mathrm{Eig}_{\mathsf{eig}(W)} = \mathrm{Eig}_W$, therefore the mapping eig is idempotent, that is, $\mathsf{eig}(\mathsf{eig}(W)) = \mathsf{eig}(W)$.*

2. *We have $W_1 \vdash_{\rho\to\sigma} W_2$ if and only if for every $H_2 \in \mathrm{Eig}_{W_2}$ there exists an $H_1 \in \mathrm{Eig}_{W_1}$ such that $H_1 \vdash_{\rho\to\sigma} H_2$. Similarly, we have $W_1 \asymp_{\rho\to\sigma} W_2$ if and only if for all $H_1 \in \mathrm{Eig}_{W_1}$ and $H_2 \in \mathrm{Eig}_{W_2}$ we have $H_1 \asymp_{\rho\to\sigma} H_2$.*

Proof. For 1. Let W be a neighborhood at type $\rho \to \sigma$. From left to right, let $\langle U, V\rangle$ be one of its eigen-neighborhoods. By the definition we have $V = \overline{WU} \cap R(W)$, from which we get that $WU \vdash_\sigma V$, that is, that $W \vdash_{\rho\to\sigma} \langle U, V\rangle$. For the other way around, let $\langle U, b\rangle \in W$. For the induced eigen-neighborhood $W\!\upharpoonright_U$ we have $W\!\upharpoonright_U U = \overline{WU} \cap R(W)$, and $b \in WU \cap R(W)$, so $W\!\upharpoonright_U \vdash_{\rho\to\sigma} \langle U, b\rangle$.

Before we deal with idempotence, let us notice that $L(W) = L(\mathsf{eig}(W))$ and $R(W) = R(\mathsf{eig}(W))$. Indeed, for the first equality, if $a \in L(W)$, then there is a $\langle U, b\rangle \in W$, with $a \in U$; then $a \in L(W\!\upharpoonright_U)$, where $W\!\upharpoonright_U \in \mathrm{Eig}_W$, so $a \in L(\mathsf{eig}(W))$; conversely, if $a \in L(\mathsf{eig}(W))$, then there is an $H \in \mathrm{Eig}_W$, such that $a \in L(H)$, which means that $a \in L(W)$ immediately. For the second equality, if $b \in R(W)$, then there is a $\langle U, b\rangle \in W$; then $b \in R(W\!\upharpoonright_U)$, where $W\!\upharpoonright_U \in \mathrm{Eig}_W$, so $b \in R(\mathsf{eig}(W))$; conversely, if $b \in R(\mathsf{eig}(W))$, then there's an $H \in \mathrm{Eig}_W$, such that $b \in R(H)$, which means that $b \in R(W)$ by the definition of eigen-neighborhoods.

For idempotence, it suffices to show that both $\mathsf{eig}(W)$ and W share the same eigen-neighborhoods. Let $H = \langle U, V\rangle$; by the definition of eigen-neighborhoods, we have $H \in \mathrm{Eig}_{\mathsf{eig}(W)}$ if and only if

$$U = \overline{U} \cap L(\mathsf{eig}(W)) \ \wedge \ V = \overline{\mathsf{eig}(W)U} \cap R(\mathsf{eig}(W)),$$

which, by the previous paragraphs (and the fact that $W \sim_{\rho\to\sigma} W'$ implies $WU \sim_\sigma W'U$ for all W, W', U), holds if and only if

$$U = \overline{U} \cap L(W) \ \wedge \ V = \overline{WU} \cap R(W);$$

this conjunction yields by definition $H \in \mathrm{Eig}_W$.

For 2. Let $W_1, W_2 \in \mathrm{Con}_{\rho\to\sigma}$. Concerning entailment, assume first that $W_1 \vdash_{\rho\to\sigma} W_2$, and let $H_2 \in \mathrm{Eig}_{W_2}$; by 1, it is clear that $W_1 \vdash_{\rho\to\sigma} H_2$, and setting $H_1 := W_1 \upharpoonright_{L(H_2)}$ we do get that $H_1 \vdash_{\rho\to\sigma} H_2$. For the other way around, assuming that every eigen-neighborhood of W_2 is entailed by some neighborhood of W_1, it follows that a part of $\bigcup \mathrm{Eig}_{W_1}$ suffices to entail all of $\bigcup \mathrm{Eig}_{W_2}$, so 1 yields what we want.

Concerning consistency, we have $W_1 \asymp_{\rho\to\sigma} W_2$ if and only if $\bigcup \mathrm{Eig}_{W_1} \asymp_{\rho\to\sigma} \bigcup \mathrm{Eig}_{W_2}$ thanks to 1; this holds exactly when $H_1 \asymp_{\rho\to\sigma} H_2$ for all $H_1 \in \mathrm{Eig}_{W_1}$ and $H_2 \in \mathrm{Eig}_{W_2}$. □

Note that although the property of idempotence justifies the use of the definite article for the eigenform of a neighborhood, the mapping eig still *does not* yield normal forms: two equivalent eigenforms are not necessarily equal, as one sees already at base types. Its utility is rather that it rearranges and tidies up the information of the given neighborhood by bringing it to a more explicit and manageable form. An example for the latter is the use of the eigenform of a neighborhood W to easily obtain *conservative extensions*: if $H \in \mathrm{Eig}_W$, then $W \sim_\rho W \cup W_H$ for every $W_H \in \mathrm{Con}_\rho$ with $H \vdash_\rho W_H$, a simple technique that often helps at higher types.

Furthermore, Proposition 4.9.2 brings into light an implicit *linear* behavior of higher-type entailment, since it says that single “generalized tokens” on the left suffice, and anticipates the facts that we establish in section 5.1.

Eigen-irredundancy

Based on the intuition of eigen-neighborhoods as generalized tokens, we may establish a form of a neighborhood, which is still no normal form, but at least does not feature informationally redundant eigen-neighborhoods.

Write Eig^i_W for the nontrivial eigen-neighborhoods of a given neighborhood W. If H is a nontrivial eigen-neighborhood of W then $R(H)$ is a nontrivial neighborhood. Furthermore, entailment between nontrivial eigen-neighborhoods reduces (contravariantly) to componentwise *inclusion*.

Lemma 4.10. *Let ρ, σ be types, $W \in \mathrm{Con}_{\rho\to\sigma}$, and $H_1, H_2 \in \mathrm{Eig}^i_W$ be such that $H_1 \vdash_{\rho\to\sigma} H_2$. Then $L(H_1) \subseteq L(H_2)$ and $R(H_2) \subseteq R(H_1)$. Moreover, we also have $R(H_2) = R(H_1)$.*

Proof. Since $H_1 \vdash_{\rho\to\sigma} H_2$, and they are nontrivial, we have $L(H_2) \vdash_\rho L(H_1)$ and $R(H_1) \vdash_\sigma R(H_2)$. The corresponding inclusions follow from the left and right closure properties that define eigen-neighborhoods. The last claim follows from the monotonicity of application (see also Lemma 4.8.2): since $L(H_2) \vdash_\rho L(H_1)$, we have $WL(H_2) \vdash_\sigma WL(H_1)$; then $\overline{WL(H_1)} \cap R(W) \subseteq \overline{WL(H_2)} \cap R(W)$, so $R(H_1) \subseteq R(H_2)$, and we're done. □

Call $W \in \mathrm{Con}_{\rho\to\sigma}$ *eigen-irredundant* if it is in eigenform and each of its nontrivial eigen-neighborhoods is maximal, that is, if $H \in \mathrm{Eig}^i_W$, then for all $H' \in \mathrm{Eig}_W$ with $H' \vdash_{\rho\to\sigma} H$, we have $H' \sim_{\rho\to\sigma} H$. Write $\mathrm{Eig}^{\max}_W$ for the set of maximal eigen-neighborhoods of W. Trivial neighborhoods are already eigen-irredundant, since each of their eigen-neighborhoods is maximal (being equivalent to the empty one). By Lemma 4.10 it follows that in an eigen-irredundant neighborhood, if H, H' are nontrivial and $H' \vdash_{\rho\to\sigma} H$ then $H' = H$.

Lemma 4.11. *Let ρ, σ be types. For every $W \in \mathrm{Con}_{\rho\to\sigma}$ there exists an eigen-irredundant $W' \in \mathrm{Con}_{\rho\to\sigma}$, such that $W \sim_{\rho\to\sigma} W'$.*

Proof. Let $W \in \mathrm{Con}_{\rho\to\sigma}$. We have $W \sim_{\rho\to\sigma} \bigcup \mathrm{Eig}_W$, by Proposition 4.9.1. Then $W' := \bigcup \mathrm{Eig}_W^{\max}$ is an equivalent eigen-irredundant neighborhood. □

An eigen-irredundant equivalent of the neighborhood W that we considered in the previous example is obtained by taking the union of all eigen-neighborhoods except $\langle\{\mathtt{S0}\},\{\mathtt{S0},\mathtt{S}*\}\rangle$ and $\{\{\mathtt{S}*,\mathtt{S0}\},\{\mathtt{S0},\mathtt{S}*\}\}$, since they are not maximal. There are in general several witnesses for the above lemma. One that we will be using a lot later is provided by the mapping $\mathsf{eirr} : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$, defined by $U \mapsto \bigcup \mathrm{Eig}_U^{\max} \cap \mathrm{Eig}_U^i$, which has the additional feature that, on trivial U’s, it keeps only the empty eigen-neighborhood.

Proposition 4.12 (Eigencorrespondence)**.** *Let $W_1, W_2 \in \mathrm{Con}_{\rho\to\sigma}$, and W_1 be eigen-irredundant. We have $W_1 \sim_{\rho\to\sigma} W_2$ if and only if for each $H_1 \in \mathrm{Eig}_{W_1}$ there is exactly one $H_2 \in \mathrm{Eig}_{W_2}$, up to equientailment, such that $H_2 \sim_{\rho\to\sigma} H_1$.*

Proof. Assume that $W_1 \sim_{\rho\to\sigma} W_2$, and let $H_1 \in \mathrm{Eig}_{W_1}$. By Proposition 4.9.2 there is an $H_2 \in \mathrm{Eig}_{W_2}$, such that $H_2 \vdash_{\rho\to\sigma} H_1$ for which, in turn, there is an $H_1' \in \mathrm{Eig}_{W_1}$, such that $H_1' \vdash_{\rho\to\sigma} H_2$. It follows that $H_1' \vdash_{\rho\to\sigma} H_1$, but W_1 is eigen-irredundant, so $H_1' = H_1$, and consequently $H_2 \sim_{\rho\to\sigma} H_1$. The uniqueness of H_2 up to equientailment is clear. As for the converse, it follows immediately from Proposition 4.9.2. □

It follows that if W_1 and W_2 are equivalent and both eigen-irredundant, then their eigen-neighborhoods are in a one to one correspondence (up to equientailment); as we will soon see, this will provide the crucial stepping stone towards our goal. Another nice property that follows from eigencorrespondence is the next one.

Corollary 4.13. *Let $W_1, W_2 \in \mathrm{Con}_{\rho\to\sigma}$ be eigen-irredundant. If $W_1 \sim_{\rho\to\sigma} W_2$ then $W_1 \upharpoonright_U \sim_{\rho\to\sigma} W_2 \upharpoonright_U$ for every $U \in \mathrm{Con}_\rho$.*

Proof. Let $U \in \mathrm{Con}_\rho$. By Proposition 4.12, there is exactly one $H \in \mathrm{Eig}_{W_1}$, such that $H \sim_{\rho\to\sigma} W_2 \upharpoonright_U$. If H is trivial then $W_2 \upharpoonright_U$ is also trivial, therefore $R(W_2 \upharpoonright_U)$ is trivial at type σ. Since W_1 and W_2 are equivalent, we get $W_1 U \sim_\sigma W_2 U$ by monotonicity of application, and then $R(W_1 \upharpoonright_U) \sim_\sigma R(W_2 \upharpoonright_U)$ by the definition of restriction; by the transitivity of entailment we have that $R(W_1 \upharpoonright_U)$ is also trivial at type σ, which means that $W_1 \upharpoonright_U$ is trivial at $\rho \to \sigma$. If H is nontrivial, then $L(H) \sim_\rho L(W_2 \upharpoonright_U)$ and $R(H) \sim_\sigma R(W_2 \upharpoonright_U)$. Now, on the one hand, since $U \vdash_\rho L(W_2 \upharpoonright_U)$, we have $U \vdash_\rho L(H)$ by transitivity, so Lemma 4.8 (items 3 and 2) yields $L(H) \subseteq L(W_1 \upharpoonright_U)$; on the other hand, since $W_1 \sim_{\rho\to\sigma} W_2$, we have $W_1 U \sim_\sigma W_2 U$, so $R(W_1 \upharpoonright_U) \sim_\sigma R(W_2 \upharpoonright_U)$, by Lemma 4.8.1. It follows that $H \vdash_{\rho\to\sigma} W_1 \upharpoonright_U$, but W_1 is eigen-irredundant, so $H = W_1 \upharpoonright_U$. This means that $W_1 \upharpoonright_U \sim_{\rho\to\sigma} W_2 \upharpoonright_U$, so we’re done. □

Eigenpowers of endomappings

Given two endomappings $f : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$ and $g : \mathrm{Con}_\sigma \to \mathrm{Con}_\sigma$, define their *eigenpower* g^f by

$$W \mapsto \bigcup_{H \in \mathrm{Eig}_W} \langle f(L(H)), g(R(H)) \rangle .$$

Obviously, this is not always a neighborhood mapping, in other words, it is not necessarily *consistently defined* for all f and g; in general we have $g^f : \mathrm{Con}_{\rho\to\sigma} \to \mathrm{Fin}_{\rho\to\sigma}$. A neighborhood mapping $f : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$ is called *inflationary* (or *expansive*) when $f(U) \vdash_\rho U$ and *deflationary* (or *contractive*) when $U \vdash_\rho f(U)$ for $U \in \mathrm{Con}_\rho$.

Lemma 4.14. *Let f and g be neighborhood endomappings at types ρ and σ respectively. If f is inflationary and g is deflationary, then g^f is a deflationary neighborhood endomapping at type $\rho \to \sigma$. Dually, if f is deflationary, g inflationary, and g^f consistently defined, then g^f is an inflationary neighborhood mapping at type $\rho \to \sigma$.*

Proof. Let $W \in \mathrm{Con}_{\rho\to\sigma}$ and $\langle U,b\rangle \in g^f(W)$. By definition there exists some $H \in \mathrm{Eig}_W$ such that $U = f(L(H))$ and $b \in g(R(H))$. The assumptions yield immediately $H \vdash_{\rho\to\sigma} \langle U,b\rangle$, so $\mathsf{eig}(W) \vdash_{\rho\to\sigma} g^f(W)$, and from Proposition 4.9.1 we're done. For the dual case, which similarly leads to $g^f(W) \vdash_{\rho\to\sigma} \mathsf{eig}(W)$, we just use the extra assumption that $g^f(W) \in \mathrm{Con}_{\rho\to\sigma}$, and we're done. □

Since a normal form mapping is information-preserving, that is, simultaneously inflationary and deflationary, it follows that the eigenpower of two normal form mappings is also deflationary (hence consistently defined) and inflationary, in other words, itself information-preserving.

Corollary 4.15. *If $f : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$ and $g : \mathrm{Con}_\sigma \to \mathrm{Con}_\sigma$ are information-preserving, then so is $g^f : \mathrm{Con}_{\rho\to\sigma} \to \mathrm{Con}_{\rho\to\sigma}$.*

To achieve a normal form, according to the definition, we additionally need to collapse every equientailment class of neighborhoods to a singleton, but the eigenpower construction alone does not ensure this. To see why, consider at type $\mathbb{N} \to \mathbb{B}$ the neighborhoods $W_1 := \{\langle\{\mathtt{S}*\},\mathtt{tt}\rangle\}$ and $W_2 := \{\langle\{\mathtt{S0}\},\mathtt{tt}\rangle,\langle\{\mathtt{S}*\},\mathtt{tt}\rangle\}$, for which we have $W_1 \sim W_2$; the corresponding eigen-neighborhoods are easily seen to be $\varnothing_{\mathbb{N}\to\mathbb{B}}$ and $\langle\{\mathtt{S}*\},\{\mathtt{tt}\}\rangle$ for W_1 and $\varnothing_{\mathbb{N}\to\mathbb{B}}$, $\langle\{\mathtt{S}*\},\{\mathtt{tt}\}\rangle$, and $\langle\{\mathtt{S0},\mathtt{S}*\},\{\mathtt{tt}\}\rangle$ for W_2; now using, say, the normal form mappings $\mathsf{lub}_\mathbb{B}$ and $\mathsf{lub}_\mathbb{N}$, we get

$$\mathsf{lub}_\mathbb{B}^{\mathsf{lub}_\mathbb{N}}(W_1) = \{\langle\{*_\mathbb{N}\},*_\mathbb{B}\rangle,\langle\{\mathtt{S}*\},\mathtt{tt}\rangle\}$$

but

$$\mathsf{lub}_\mathbb{B}^{\mathsf{lub}_\mathbb{N}}(W_2) = \{\langle\{*_\mathbb{N}\},*_\mathbb{B}\rangle,\langle\{\mathtt{S}*\},\mathtt{tt}\rangle,\langle\{\mathtt{S0}\},\mathtt{tt}\rangle\}.$$

Proposition 4.16. *Let f and g be normal form mappings at types ρ and σ respectively. Then their eigenpower is a normal form mapping at type $\rho \to \sigma$, when restricted to eigen-irredundant neighborhoods.*

Proof. As we already mentioned above, by Lemma 4.14 we have $g^f(W) \sim_{\rho\to\sigma} W$ for every $W \in \mathrm{Con}_{\rho\to\sigma}$ (even for redundant ones), so there remains to show that for any eigen-irredundant $W_1, W_2 \in \mathrm{Con}_{\rho\to\sigma}$ that satisfy $W_1 \sim_{\rho\to\sigma} W_2$, we have $g^f(W_1) = g^f(W_2)$. So let W_1 and W_2 be two such neighborhoods. By Proposition 4.12 we know that every equientailment class of eigen-neighborhoods of W_1 corresponds to exactly one equientailment class of eigen-neighborhoods of W_2; under f and g, which are normal form mappings, each such class collapses to a singleton, so we obtain

$$\bigcup_{H_1 \in \mathrm{Eig}_{W_1}} \langle f(L(H_1)), g(R(H_1))\rangle =_{\rho\to\sigma} \bigcup_{H_2 \in \mathrm{Eig}_{W_2}} \langle f(L(H_2)), g(R(H_2))\rangle,$$

and therefore $g^f(W_1) = g^f(W_2)$. □

Theorem 4.17 (Normal forms for function spaces)**.** *If there exist normal form mappings at types ρ and σ respectively, then there exists a normal form mapping at type $\rho \to \sigma$.*

Proof. Let f and g be normal form mappings at types ρ and σ respectively. From Proposition 4.16 and the fact that eirr is a witness for Lemma 4.11 we immediately have that the composition $g^f \circ \mathsf{eirr}$ is a normal form mapping at type $\rho \to \sigma$. □

Note that different choices of normal forms at the constituent types result in different normal forms at the higher type. In view of the smooth behavior of the eigenpower on eigen-irredundant neighborhoods, we will always bundle together the composition of eigenpower and eirr, and—for want of a better notation—write ${}^f g$ instead of $g^f \circ \mathsf{eirr}$.

4.3 Normal forms at arbitrary base types

In §4.1 we examined nonparametric finitary algebras, which are in a sense pure, in that their elements are constructed inductively but independently of other types. In general, though, the algebras that we use may be mixed, that is, they may depend on other given types, just like higher types do. This mixing forces us to discuss normal forms at arbitrary base types in a more modular way.

Suppose for example that we are interested in the algebra

$$\mathbb{A} = \mu_\xi(\xi, \xi \to \xi, \tau \to (\mathbb{N} \to \xi) \to \xi),$$

which we can think of as a variation of $\mathbb{O}$ where the limit constructor is parametrized by values in τ (officially, we should have written $\mathbb{A}(\tau)$). Assuming that we already have a normal form mapping $\mathsf{nf}_\mathbb{N}$ and that we can readily provide a normal form mapping nf_τ every time τ is specified, it seems canonical to try and construct a normal form mapping $\mathsf{nf}_\mathbb{A}$ by pattern-matching, or *by recursion* on $\mathbb{A}$, loosely as follows:

$$\begin{aligned}
&\mathsf{nf}_\mathbb{A}(\{*_\mathbb{A}\}) := U_* \text{ for a fixed } U_* \in \mathrm{Con}_\mathbb{A},\\
&\mathsf{nf}_\mathbb{A}(\{0_\mathbb{A}\}) := U_0 \text{ for a fixed } U_0 \in \mathrm{Con}_\mathbb{A},\\
&\mathsf{nf}_\mathbb{A}(\mathtt{S}_\mathbb{A} U) := \mathtt{S}_\mathbb{A}\mathsf{nf}_\mathbb{A}(U),\\
&\mathsf{nf}_\mathbb{A}(\mathtt{L}_\mathbb{A} U W) := \mathtt{L}_\mathbb{A}\mathsf{nf}_\tau(U)\mathsf{nf}_{\mathbb{N}\to\mathbb{A}}(W).
\end{aligned}$$

To this end we need to develop the theory of neighborhood mappings in a fairly straightforward but nonetheless much more careful and detailed manner.

Products, sums, and compositions of neighborhood mappings

We first turn to the case of neighborhood mappings that accept arguments from *various* types, either at once (product types), or by case distinction (sum types).

Define the *(separated) sum* and the *lifted product* of two finite sets A and B to be the sets

$$\begin{aligned}
A \dotplus B &:= \{\#_+\} \cup (\{0\} \times A) \cup (\{1\} \times B),\\
A \dot\times B &:= \{\#_\times\} \cup A \times B,
\end{aligned}$$

respectively, where $\#_+$ and $\#_\times$ are new symbols to both A and B.

Lemma 4.18. *Let ρ and σ be two types. We have*

$$\mathrm{Con}^1_\rho \dot\times \mathrm{Con}^1_\sigma \simeq \mathrm{Con}^1_{\rho\times\sigma} \quad \textit{and} \quad \mathrm{Con}_\rho \dotplus \mathrm{Con}_\sigma \simeq \mathrm{Con}^1_{\rho+\sigma}.$$

Proof. An isomorphism pair for products is given by

$$\begin{aligned}
&\mathrm{Con}^1_\rho \dot\times \mathrm{Con}^1_\sigma \hookrightarrow \mathrm{Con}^1_{\rho\times\sigma},\\
&\quad \#_\times \mapsto \{*_{\rho\times\sigma}\},\\
&\quad (U^\rho, V^\sigma) \mapsto \{(a,b) \mid a \in U \wedge b \in V\},\\
&\mathrm{Con}^1_{\rho\times\sigma} \hookrightarrow \mathrm{Con}^1_\rho \dot\times \mathrm{Con}^1_\sigma,\\
&\quad \{*_{\rho\times\sigma}\} \mapsto \#_\times,\\
&\quad \{(a_i,b_i) \mid i=1,\dots,m\} \mapsto (\{a_i \mid i=1,\dots,m\}, \{b_i \mid i=1,\dots,m\}),
\end{aligned}$$

and for sums by

$$\begin{aligned}
&\mathrm{Con}_\rho \dot+ \mathrm{Con}_\sigma \hookrightarrow \mathrm{Con}^1_{\rho+\sigma},\\
&\quad \#_+ \mapsto \{*_{\rho+\sigma}\},\\
&\quad (i,U) \mapsto \begin{cases} \mathtt{I}_\rho U & \text{if } i=0,\\ \mathtt{I}_\sigma U & \text{if } i=1,\end{cases}\\
&\mathrm{Con}^1_{\rho+\sigma} \hookrightarrow \mathrm{Con}_\rho \dot+ \mathrm{Con}_\sigma,\\
&\quad \{*_{\rho+\sigma}\} \mapsto \#_+,\\
&\quad \mathtt{I}_\tau U \mapsto \begin{cases} (0,U) & \text{if } \tau=\rho,\\ (1,U) & \text{if } \tau=\sigma.\end{cases}
\end{aligned}$$

The details are straightforward. □

This lemma is useful in two ways. Firstly, it shows how all notions and arguments concerning general neighborhood mappings carry over automatically to the cases where a neighborhood mapping takes vectors as arguments or is defined by cases. Secondly, it allows us a certain degree of sloppiness when writing down mappings on several arguments, a freedom that we readily exploit.

To comment on the handling of empty neighborhoods, as we will see, the main reason of introducing mappings defined on product types it to facilitate an interplay with the constructor application of section 2, where empty neighborhoods are not allowed anyway; nevertheless, we can of course consider mappings that assign values to empty arguments, but we just have to keep in mind that Lemma 4.18 will not apply unless we restrict the domain to inhabited neighborhoods (in what follows we will try to avoid this nitpicking attitude). On the other hand, when we define a neighborhood-mapping f on a sum type $\rho+\sigma$, we may in general define it also on the empty neighborhood, with a reasonable demand in this case being that $f(\varnothing_{\rho+\sigma}) \sim f(\{*_{\rho+\sigma}\})$.

Examples. Typical examples of mappings defined on a product are the *projection neighborhood mappings* $\mathsf{pr}_i : \mathrm{Con}_{\rho_1\times\rho_2} \to \mathrm{Con}_{\rho_i}$, defined by $(U_1,U_2) \mapsto U_i$ for $i=1,2$, and similarly for larger products (note already that we sloppily wrote (U_1,U_2) instead of $\mathtt{T}U_1U_2$). A further natural example comes from the application between neighborhoods: for types ρ and σ, the *application neighborhood mapping* $\cdot : \mathrm{Con}_{(\rho\to\sigma)\times\rho} \to \mathrm{Con}_\sigma$ is given by $(W,U) \mapsto WU$; as we mentioned already, again in section 2, this mapping is also monotone. An example for sums would be a mapping $f : \mathrm{Con}^1_{\rho+\sigma} \to \mathrm{Con}^1_{\rho+\sigma}$ with $U \mapsto U^\rho$ if $U \in \mathrm{Con}_\rho$ and $U \mapsto U^\sigma$ if $U \in \mathrm{Con}_\sigma$, for some fixed $U^\rho \in \mathrm{Con}_\rho$ and $U^\sigma \in \mathrm{Con}_\sigma$—but soon we will see a less naive example (again, we were sloppy when we wrote "$U \mapsto U^\rho$ if $U \in \mathrm{Con}_\rho$" instead of "$\mathtt{I}_\rho U \mapsto U^\rho$"). □

We will need a way to combine given mappings into a single mapping over the product or sum of the respective given types. Let ρ_m, σ_m, ρ_l, and σ be types for $m = 1,\dots,n$, $l = 1,\dots,k$. The *product* of the neighborhood mappings $f_m : \mathrm{Con}_{\rho_m} \to \mathrm{Con}_{\sigma_m}$ is the neighborhood mapping $f_1 \times \cdots \times f_n : \mathrm{Con}_{\rho_1\times\cdots\times\rho_n} \to \mathrm{Con}_{\sigma_1\times\cdots\times\sigma_n}$ given by

$$\begin{aligned}(U_1,\dots,U_n) &\mapsto (f_1(U_1),\dots,f_n(U_n)),\\ \varnothing,\{*\} &\mapsto \{*\},\end{aligned}$$

for all $U_m \in \mathrm{Con}_{\rho_m}$. To define a notion of sum for mappings, we will use the component form (see "Information systems as interpretations of types" in section 2) of a base-type neighborhood, to use in the various clauses of sum definitions: if ι is some algebra and $U \in \mathrm{Con}_\iota$, we simply set $\mathsf{cf}(U) := \{*_\iota\}$ if $U \sim_\iota \{*_\iota\}$, and $\mathsf{cf}(U) := CU(1)\cdots U(r)$ if there is some $Ca_1\cdots a_r \in U$ (with $U(i)$ being the i-th component neighborhood of U). Now the *sum* of the mappings $f_l : \mathrm{Con}_{\rho_l} \to \mathrm{Con}_\sigma$, for $l = 1,\dots,k$, is the neighborhood mapping $f_1 + \cdots + f_k : \mathrm{Con}_{\rho_1+\cdots+\rho_k} \to \mathrm{Con}_\sigma$ given by

$$U \mapsto \begin{cases} f_1(U_0), & \mathsf{cf}(U) = \mathtt{I}_1 U_0,\\ \quad\vdots & \\ f_k(U_0), & \mathsf{cf}(U) = \mathtt{I}_k U_0,\\ \bot_\sigma, & \mathsf{cf}(U) = \{*\},\end{cases}$$

for all $U \in \mathrm{Con}_{\rho_1+\cdots+\rho_k}$, where $\bot_\iota := \{*_\iota\}$ for every algebra ι and $\bot_\sigma := \varnothing_\sigma$ for every other type σ; note that the use of the symbol "$\bot$" here refers to a neighborhood, and not to an ideal as in section 2. For reasons of typographical economy we may also use the notations $\prod_m f_m$ and $\sum_l f_l$, respectively.[9]

Lemma 4.19. *A product neighborhood mapping is compatible (respectively, monotone or consistent) if and only if each one of its component neighborhood mappings is compatible (respectively, monotone or consistent). The same holds for a sum neighborhood mapping.*

Proof. For monotonicity in the product case, without harming generality let $(U_1,\dots,U_m),(U_1',\dots,U_m') \in \mathrm{Con}_{\rho_1\times\cdots\times\rho_m}$ be such that $(U_1,\dots,U_m) \vdash_{\rho_1\times\cdots\times\rho_m} (U_1',\dots,U_m')$. The value $(f_1\times\cdots\times f_m)(U_1,\dots,U_m)$ is by definition equal to the vector $(f_1(U_1),\dots,f_m(U_m))$, which, by the assumption and the definition of entailment at product types, entails in $\sigma_1\times\cdots\times\sigma_m$ the vector $(f_1(U_1'),\dots,f_m(U_m'))$; but this is exactly the value $(f_1\times\cdots\times f_m)(U_1,\dots,U_m)$, as we wanted. Compatibility and consistency are shown similarly.

Let's show consistency for the sum case. Let $U,U' \in \mathrm{Con}^1_{\rho_1+\cdots+\rho_k}$ be such that $U \asymp_{\rho_1+\cdots+\rho_k} U'$. By the definition of consistency at sum types, and without harming generality, U and U' must draw from the same constituent type, say $\mathsf{cf}(U) = \mathtt{I}_l U_0$ and $\mathsf{cf}(U') = \mathtt{I}_l U_0'$, and satisfy $U_0 \asymp_{\rho_l} U_0'$; by assumption, the respective mapping f_l is consistent, so $f_l(U_0) \asymp_\sigma f_l(U_0')$, therefore $(f_1+\cdots+f_k)(U) \asymp_\sigma (f_1+\cdots+f_k)(U')$. The argument works backwards in an analogous way, and the arguments for monotonicity and compatibility work similarly. ☐

[9] Following category-theoretical conventions [3], we should perhaps use "$[f_1,\dots,f_k]$" instead of "$f_1 + \cdots + f_k$"—which would rather stand for a mapping of the sort $\mathrm{Con}_{\rho_1+\cdots+\rho_k} \to \mathrm{Con}_{\sigma_1+\cdots+\sigma_k}$, the "coproduct" of $f_l : \mathrm{Con}_{\rho_l} \to \mathrm{Con}_{\sigma_l}$ for every $l = 1,\dots,k$. In the present framework though we do not have to thematize the latter notion, so we indulge our need for typographical convenience without risking clarity.

Let us lastly officially address the case of composition, which we actually already tacitly used in Theorem 4.17. Let ρ, σ, τ be types, and $f : \mathrm{Con}_\tau \to \mathrm{Con}_\sigma$, $g : \mathrm{Con}_\rho \to \mathrm{Con}_\tau$ be neighborhood mappings. As expected, the *composition* $f \circ g : \mathrm{Con}_\rho \to \mathrm{Con}_\sigma$ is defined by

$$U \mapsto f(g(U))$$

for every $U \in \mathrm{Con}_\rho$.

Lemma 4.20. *Let ρ, σ, τ be types. If $f : \mathrm{Con}_\tau \to \mathrm{Con}_\sigma$ and $g : \mathrm{Con}_\rho \to \mathrm{Con}_\tau$ are compatible (respectively, monotone or consistent) then their composition $f \circ g$ is also compatible (and respectively, monotone or consistent).*

Proof. Let $U, U' \in \mathrm{Con}_\rho$. For consistency, if $U \asymp_\rho U'$, then $g(U) \asymp_\tau g(U')$ by the consistency of g, and $f(g(U)) \asymp_\sigma f(g(U'))$ by the consistency of f. Monotonicity and compatibility are shown in the same way. □

Eigenpowers of general neighborhood mappings

We already defined the eigenpower between two neighborhood *endo*mappings in section 4.2, and here we generalize this operation for general mappings. Let ρ_1, ρ_2, σ_1, and σ_2 be types, and $f : \mathrm{Con}_{\rho_1} \to \mathrm{Con}_{\rho_2}$, $g : \mathrm{Con}_{\sigma_1} \to \mathrm{Con}_{\sigma_2}$ two neighborhood mappings. The *eigenpower* of f and g is the mapping $g^f : \mathrm{Con}_{\rho_1 \to \sigma_1} \to \mathrm{Fin}_{\rho_2 \to \sigma_2}$ given by

$$W \mapsto \bigcup_{H \in \mathrm{Eig}_W} \langle f(L(H)), g(R(H)) \rangle$$

for every $W \in \mathrm{Con}_{\rho_1 \to \sigma_1}$. As was the case with eigenpowers of endomappings, the eigenpower in general need not produce a consistent set.

To work out a criterion for consistency in view of the governing contravariance, and since the notions of inflationary and deflationary mappings don't apply anymore as in Lemma 4.14, it helps to consider, along with consistency-preserving neighborhood mappings, also *in*consistency-preserving ones: a neighborhood mapping $f : \mathrm{Con}_\rho \to \mathrm{Con}_\sigma$ is called *consistency-reflecting* if $f(U_1) \asymp_\sigma f(U_2)$ implies $U_1 \asymp_\rho U_2$ for all $U_1, U_2 \in \mathrm{Con}_\rho$. Apparent but important examples of consistency-reflecting mappings are identity mappings as well as normal form mappings, and more generally all information-preserving mappings.

Lemma 4.21. *Let $f : \mathrm{Con}_{\rho_1} \to \mathrm{Con}_{\rho_2}$ and $g : \mathrm{Con}_{\sigma_1} \to \mathrm{Con}_{\sigma_2}$ be two neighborhood mappings. If f is consistency-reflecting and g is consistency-preserving, then g^f is a neighborhood mapping.*

Proof. Let $W \in \mathrm{Con}_{\rho_1 \to \sigma_1}$ and $\langle U_1, b_1 \rangle, \langle U_2, b_2 \rangle \in g^f(W)$ with $U_1 \asymp_{\rho_2} U_2$. By the definition of eigenpower there exist inhabited eigen-neighborhoods H_1, H_2 of W, such that $U_i = f(L(H_i))$ and $b_i \in g(R(H_i))$ for each $i = 1, 2$. So $U_1 \asymp_{\rho_2} U_2$ means $f(L(H_1)) \asymp_{\rho_2} f(L(H_2))$, which implies $L(H_1) \asymp_{\rho_1} L(H_2)$ since f is consistency-reflecting; since W is a neighborhood we get $R(H_1) \asymp_{\sigma_1} R(H_2)$, and the consistency of g yields $g(R(H_1)) \asymp_{\sigma_2} g(R(H_2))$, which finally gives us $b_1 \asymp_{\sigma_2} b_2$, as we wanted. □

As in the case of endomappings, also here it makes sense to write ${}^f g$ for $g^f \circ \mathsf{eirr}$, where $f : \mathrm{Con}_{\rho_1} \to \mathrm{Con}_{\rho_2}$ and $g : \mathrm{Con}_{\sigma_1} \to \mathrm{Con}_{\sigma_2}$, meaning that we perform the eigenpower operation after we have brought the initial $(\rho_1 \to \sigma_1)$-neighborhood to an equivalent eigen-irredundant form.

Lemma 4.22. *Let ρ_1, ρ_2, σ_1, and σ_2 be types, and $f : \mathrm{Con}_{\rho_1} \to \mathrm{Con}_{\rho_2}$, $g : \mathrm{Con}_{\sigma_1} \to \mathrm{Con}_{\sigma_2}$ two neighborhood mappings. The following hold.*

1. *If f and g are compatible then so is ${}^f g$.*
2. *If f is consistency-reflecting and g consistency-preserving, then ${}^f g$ is consistency-preserving.*
3. *If f and g are monotone, then so is ${}^f g$.*

Proof. The mapping eirr is information-preserving by Lemma 4.11, so it is a fortiori monotone, consistency-preserving, and compatible. Furthermore, by Lemma 4.20, composition preserves compatibility, consistency, and monotonicity, so it suffices to show the lemma for the mapping g^f applied to two eigen-irredundant neighborhoods $W_1, W_2 \in \mathrm{Con}_{\rho_1 \to \sigma_1}$.

For compatibility, assume that $W_1 \sim_{\rho_1 \to \sigma_1} W_2$. By eigencorrespondence (Proposition 4.12), every $H_1 \in \mathrm{Eig}_{W_1}$ has exactly one equivalent $H_2 \in \mathrm{Eig}_{W_2}$, and vice versa. If H_1 and H_2 are such, then by compatibility of f and g we have

$$f(L(H_1)) \sim_{\rho_2} f(L(H_2)) \wedge g(R(H_1)) \sim_{\sigma_2} g(R(H_2)),$$

from which we get

$$\langle f(L(H_1)), g(R(H_1))\rangle \sim_{\rho_2 \to \sigma_2} \langle f(L(H_2)), g(R(H_2))\rangle$$

so $g^f(W_1) \sim_{\rho_2 \to \sigma_2} g^f(W_2)$ by the definition of eigenpower. For consistency, the argument is analogous to the one of the proof of Lemma 4.21. Finally, for monotonicity, assume that $W_1 \vdash_{\rho_1 \to \sigma_1} W_2$, and let $H_2 \in \mathrm{Eig}_{W_2}$; there is some $H_1 \in \mathrm{Eig}_{W_1}$ such that $H_1 \vdash_{\rho_1 \to \sigma_1} H_2$, which means that

$$L(H_2) \vdash_{\rho_1} L(H_1) \wedge R(H_1) \vdash_{\sigma_1} R(H_2)$$

(here we assume without loss of generality that H_2 is nontrivial); by the monotonicity of both f and g we get

$$f(L(H_2)) \vdash_{\rho_2} f(L(H_1)) \wedge g(R(H_1)) \vdash_{\sigma_2} g(R(H_2)),$$

which is sufficient to yield what we want. □

Recursive mappings

Consider an algebra $\iota = \mu_\xi(\kappa_1, \dots, \kappa_k)$, where the l-th constructor type κ_l is

$$\vec{\tau_l} \to (\overrightarrow{\rho_{1l}} \to \xi) \to \dots \to (\overrightarrow{\rho_{n_l l}} \to \xi) \to \xi$$

for $l = 1, \dots, k$. For each l write $\tilde{\kappa}_l(\sigma)$ for

$$\bar{\tau}_l \times (\overline{\rho_{1l}} \to \sigma) \times \dots \times (\overline{\rho_{n_l l}} \to \sigma),$$

where σ may be any type. Here we use the bar notation for *products* of types: $\bar{\rho}$ means $\rho_1 \times \dots \times \rho_m$ for $m > 0$ and $\mathbb{U}$ for $m = 0$. In particular, in the case of a nullary constructor, the degenerate product type that emerges as $\tilde{\kappa}_l(\sigma)$, is just $\mathbb{U}$. Now call *construction type (from ι to σ)*, and write $\tilde{\iota}(\sigma)$ for

$$\tilde{\kappa}_1(\sigma) + \dots + \tilde{\kappa}_k(\sigma);$$

in case σ is ι we just write $\tilde{\iota}$.

Lemma 4.23. *Let ι be an algebra. There exist neighborhood mappings* $\mathsf{dst}_\iota : \mathrm{Con}_\iota \to \mathrm{Con}_{\tilde\iota}$ *and* $\mathsf{cst}_\iota : \mathrm{Con}_{\tilde\iota} \to \mathrm{Con}_\iota$ *such that*

$$(\mathsf{cst}_\iota \circ \mathsf{dst}_\iota)(U) \sim_\iota U \wedge (\mathsf{dst}_\iota \circ \mathsf{cst}_\iota)(\tilde U) \sim_{\tilde\iota} \tilde U,$$

for every $U \in \mathrm{Con}_\iota$ *and every* $\tilde U \in \mathrm{Con}_{\tilde\iota}$.

Proof. Define

$$\mathsf{dst}_\iota(U) := \begin{cases} \{*_{\tilde\iota}\}, & \mathsf{cf}(U) = \{*_\iota\}, \\ \mathtt{I}_l U^1, & \mathsf{cf}(U) = C_l U^0, \end{cases}$$

$$\mathsf{cst}_\iota(\tilde U) := \begin{cases} \{*_\iota\}, & \mathsf{cf}(\tilde U) = \{*_{\tilde\iota}\}, \\ C_l \tilde U^1, & \mathsf{cf}(\tilde U) = \mathtt{I}_l \tilde U^0 \end{cases}$$

(recall that, if C is nullary, then CV means that V is an empty vector, and that the resulting neighborhood is just $\{C\}$). It is easy to check that these do what we want them to do. □

We say that a mapping f *is defined by recursion on ι*, or just that it is *recursive*, if $f : \mathrm{Con}_\iota \to \mathrm{Fin}_\sigma$, and for each $l = 1, \ldots, k$ there are mappings

$$g_l : \mathrm{Con}_{\tilde\kappa_l(\sigma)} \to \mathrm{Fin}_\sigma, \quad g_{\overline{\tau}_l} : \mathrm{Con}_{\overline{\tau}_l} \to \mathrm{Con}_{\overline{\tau}_l}, \quad g_{\overline{\rho_{m_l l}}} : \mathrm{Con}_{\overline{\rho_{m_l l}}} \to \mathrm{Con}_{\overline{\rho_{m_l l}}},$$

such that

$$f = \Big(\sum_{l=1}^{k} g_l \circ g_{\overline{\tau}_l} \times \prod_{m_l=1}^{n_l} {}^{g_{\overline{\rho_{m_l l}}}} f \Big) \circ \mathsf{dst}_\iota, \tag{R}$$

where $\times$ binds stronger than $\circ$. Notice how dst operates as a *selector*, sending the input to the appropriate constructor, and also that a mapping with the unit type as source type morally stands for a *constant* of the target type.

It is crucial for our purposes that a recursive mapping be allowed to not be consistently defined in general. For example, if some of the mappings $g_{\overline{\rho_{m_l l}}}$ do not reflect consistency, the preservation of consistency for f is in question. Another reason is that one of our basic building blocks for such recursions is the *union* $f_1 \cup f_2 : \mathrm{Con}_\rho \to \mathrm{Fin}_\sigma$ of two neighborhood mappings $f_1, f_2 : \mathrm{Con}_\rho \to \mathrm{Con}_\sigma$, defined by

$$U \mapsto f_1(U) \cup f_2(U).$$

Obviously, this is consistently defined only when the two neighborhood mappings are *mutually consistent*, that is, such that $f_1(U) \asymp_\sigma f_2(U)$ for every $U \in \mathrm{Con}_\rho$; an obvious characterization of mutual consistency in the case of mappings defined on an algebra ι given by k constructors, is that they are *mutually consistent at each constructor*, meaning that $f_1(U) \asymp_\sigma f_2(U)$ for $U \sim_\iota \{*_\iota\}$ as well as $f_1(C_l U) \asymp_\sigma f_2(C_l U)$ for $U \in \mathrm{Con}_{\tilde\kappa_l(\iota)}$, $l = 1, \ldots, k$.

Examples. All four normal form mappings that we already encountered in section 4.1 (the case being a nonparametric finitary algebra $\iota = \mu_\xi(\kappa_1, \ldots, \kappa_k)$) are recursive in the above sense. Indeed, we can see each of them as an instance of a mapping $\mathsf{nf}_\iota : \mathrm{Con}_\iota \to \mathrm{Con}_\iota$ with

$$\mathsf{nf}_\iota = \Big(\sum_{l=1}^{k} g_l \circ \prod_{m_l=1}^{n_l} \mathsf{nf}_\iota \Big) \circ \mathsf{dst}_\iota$$

for appropriate $g_l : \mathrm{Con}_{\tilde{\kappa}_l(\iota)} \to \mathrm{Con}_\iota$, $l = 1, \ldots, k$. In particular, these g_l would be defined by

$$(U_1, \ldots, U_{n_l}) \mapsto C_l U_1 \cdots U_{n_l},$$
$$(U_1, \ldots, U_{n_l}) \mapsto \{*_\iota\} \cup C_l U_1 \cdots U_{n_l}$$

for the supremum and the closure mapping respectively, and by

$$(U_1, \ldots, U_{n_l}) \mapsto \bigcup_{m_l=1}^{n_l} C_l \overrightarrow{\{*_\iota\}} U_{m_l} \overrightarrow{\{*_\iota\}},$$
$$(U_1, \ldots, U_{n_l}) \mapsto \{*_\iota\} \cup \bigcup_{m_l=1}^{n_l} C_l \overrightarrow{\{*_\iota\}} U_{m_l} \overrightarrow{\{*_\iota\}}$$

for the path and the tree mappings respectively.

Similarly, we can view the partial height mapping as a recursive mapping $\mathsf{ph}_\iota : \mathrm{Con}_\iota \to \mathrm{Con}_\mathbb{N}$ (with ι being finitary), given by

$$\mathsf{ph}_\iota = \Big(\sum_{l=1}^{k} g_l \circ \prod_{m_l=1}^{n_l} \mathsf{ph}_\iota\Big) \circ \mathsf{dst}_\iota$$

where the g_l's are given by

$$(U_1, \ldots, U_{n_l}) \mapsto \{\mathsf{S}*_\mathbb{N}\} \cup \mathsf{S}\Big(\bigcup_{m_l=1}^{n_l} U_{m_l}\Big),$$

for $l > 0$. Notice that these g_l's are in themselves not necessarily consistently defined, but their composition with ph_ι trivially ensures the overall consistency. □

Proposition 4.24. *Let $f : \mathrm{Con}_\iota \to \mathrm{Fin}_\sigma$ be a recursive mapping given by the equation* (R)*. If the mappings g_l are consistently defined for $l = 1, \ldots, k$, then so is f. Furthermore:*

1. *if the mappings g_l, $g_{\bar{\tau}_l}$, and $g_{\overline{\rho_{m_l l}}}$ are compatible, then so is f;*
2. *if the mappings g_l and $g_{\bar{\tau}_l}$ are consistency-preserving and $g_{\overline{\rho_{m_l l}}}$ are consistency-reflecting, then f is consistency-preserving;*
3. *if the mappings g_l, $g_{\bar{\tau}_l}$, and $g_{\overline{\rho_{m_l l}}}$ are monotone, then so is f.*

Proof. All statements are easy to accept by mere inspection of (R), based on Lemma 4.19, Lemma 4.20, and Lemma 4.22. We will nevertheless elaborate a bit, just to get a feeling of how one should perform induction on this equation.

For the first statement, assume that all g_l's are consistently defined (that is, neighborhood mappings), and let $U \in \mathrm{Con}_\iota$. In case $\mathsf{cf}(U) = \{*_\iota\}$, we have $f(U) = \bot_\sigma$ by the definition of the sum of neighborhood mappings, and of course $\bot_\sigma \in \mathrm{Con}_\sigma$ for any type σ. In case $\mathsf{cf}(U) = C_l U_0 U_1 \cdots U_{n_l}$ for some $l = 1, \ldots, k$, with $U_0 \in \mathrm{Con}_{\bar{\tau}_l}$ and $U_{m_l} \in \mathrm{Con}_{\overline{\rho_{m_l l}} \to \iota}$ for $m_l = 1, \ldots, n_l$, we have: (a) $g_{\bar{\tau}_l}(U_0) = V_0 \in \mathrm{Con}_{\bar{\tau}_l}$, (b) $({}^{g_{\overline{\rho_{m_l l}}}} f)(U_{m_l}) = V_{m_l} \in \mathrm{Con}_{\overline{\rho_{m_l l}} \to \sigma}$ by the induction hypothesis for every m_l, and then (c) $f(U) = g_l(V_0, V_1, \ldots, V_{n_l}) \in \mathrm{Con}_\sigma$ by the assumption.

Let's see the argument for the preservation of consistency (the other arguments are carried out similarly). Let $U, U' \in \mathrm{Con}_\iota$ be such that $U \asymp_\iota U'$. If, say, $\mathsf{cf}(U') = \{*_\iota\}$, then $f(U) \asymp_\sigma \bot_\sigma = f(U')$. Otherwise, there will be an l between 1 and k, and neighborhoods $U_0, U_1, \ldots, U_{n_l}$ and $U'_0, U'_1, \ldots, U'_{n_l}$ of the appropriate types, such that $\mathsf{cf}(U) = C_l U_0 U_1 \cdots U_{n_l}$ and $\mathsf{cf}(U') = C_l U'_0 U'_1 \cdots U'_{n_l}$, with

$$U_0 \asymp_{\bar{\tau}_l} U'_0 \wedge \forall_{m_l=1}^{n_l} U_{m_l} \asymp_{\overline{\rho_{m_l}l} \to \iota} U'_{m_l};$$

assuming that every $g_{\bar{\tau}_l}$ preserves and every $g_{\overline{\rho_{m_l}l}}$ reflects consistency, and together with the induction hypothesis that f preserves the consistency of the arguments at hand, we get

$$g_{\bar{\tau}_l}(U_0) \asymp_{\bar{\tau}_l} g_{\bar{\tau}_l}(U'_0) \wedge \forall_{m_l=1}^{n_l} ({}^{g_{\overline{\rho_{m_l}l}}} f)(U_{m_l}) \asymp_{\overline{\rho_{m_l}l} \to \sigma} ({}^{g_{\overline{\rho_{m_l}l}}} f)(U'_{m_l});$$

assuming further that g_l preserves consistency, we get

$$g_l(g_{\bar{\tau}_l}(U_0), \ldots, ({}^{g_{\overline{\rho_{m_l}l}}} f)(U_{m_l}), \ldots) \asymp_\sigma g_l(g_{\bar{\tau}_l}(U'_0), \ldots, ({}^{g_{\overline{\rho_{m_l}l}}} f)(U'_{m_l}), \ldots),$$

so $f(U) \asymp_\sigma f(U')$, as we wanted. □

We are now ready to prove the normal form theorem for the general case of base types.

Theorem 4.25 (Normal forms for base types)**.** *Let* $\iota = \mu_\xi(\kappa_1, \ldots, \kappa_k)$ *be an arbitrary base type and* $l = 1, \ldots, k$*. If* $g_{\bar{\tau}_l} : \mathrm{Con}_{\bar{\tau}_l} \to \mathrm{Con}_{\bar{\tau}_l}$ *and* $g_{\overline{\rho_{m_l}l}} : \mathrm{Con}_{\overline{\rho_{m_l}l}} \to \mathrm{Con}_{\overline{\rho_{m_l}l}}$ *are normal form mappings and* $g_l : \mathrm{Con}_{\tilde{\kappa}_l(\iota)} \to \mathrm{Fin}_\iota$ *are consistently defined and such that*

$$g_l(U_0, U_1, \ldots, U_{n_l}) \sim_\iota C_l U_0 U_1 \cdots U_{n_l}, \tag{A}$$

for all $U_0, U'_0, U_1, U'_1, \ldots, U_{n_l}, U'_{n_l}$ *of appropriate types, then the recursive mapping defined by the equation* (R) *is a normal form mapping at type* ι*.*

Proof. By Proposition 4.24, such an f is consistently defined. We have to show that it preserves information and identifies equivalent neighborhoods (uniqueness).

To show the preservation of information, let $U \in \mathrm{Con}_\iota$. In case $\mathsf{cf}(U) = \{*_\iota\}$ we have $f(U) = \{*_\iota\} \sim_\iota U$. In case $\mathsf{cf}(U) = C_l U_0 U_1 \cdots U_{n_l}$ for some l and $U_0, U_1, \ldots, U_{n_l}$ of appropriate types, we have

$$\begin{aligned} f(U) &= g_l(g_{\bar{\tau}_l}(U_0), ({}^{g_{\overline{\rho_{1l}}}} f)(U_1), \ldots, ({}^{g_{\overline{\rho_{n_l}l}}} f)(U_{n_l})) \\ &\overset{(\star)}{\sim_\iota} g_l(U_0, U_1, \ldots, U_{n_l}) \\ &\overset{(A)}{\sim_\iota} C_l U_0 U_1 \cdots U_{n_l} \\ &\sim_\iota U, \end{aligned}$$

where at step $(\star)$ we used the preservation of information of $g_{\bar{\tau}_l}$ and ${}^{g_{\overline{\rho_{m_l}l}}} f$ for $m_l = 1, \ldots, n_l$ (here is where we need the induction hypothesis for the arguments at hand), as well as the *compatibility of* g_l *in nontrivial arguments*, which in turn follows from (A): if $U, U' \in \mathrm{Con}_{\tilde{\kappa}_l(\iota)}$ are such that $U \sim_{\tilde{\kappa}_l(\iota)} U'$, then

$$g_l(U) \overset{(A)}{\sim_\iota} C_l U \sim_\iota C_l U' \overset{(A)}{\sim_\iota} g_l(U'),$$

as we wanted.

Now let $U, U' \in \mathrm{Con}_\iota$ be such that $U \sim_\iota U'$. In case both neighborhoods are trivial, we have $f(U_1) = \{*_\iota\} = f(U_2)$. In case they're not, there will be an $l = 1, \ldots, k$ and neighborhoods $U_0, U_0', U_1, U_1', \ldots, U_{m_l}, U_{m_l}'$ of appropriate types, such that $\mathsf{cf}(U) = C_l U_0 U_1 \cdots U_{n_l}$ and $\mathsf{cf}(U') = C_l U_0' U_1' \cdots U_{n_l}'$, as well as

$$U_0 \sim_{\bar{\tau}_l} U_0' \wedge \bigvee_{m_l=1}^{n_l} U_{m_l l} \sim_{\overline{\rho_{m_l l}} \to \iota} U_{m_l l}';$$

by the uniqueness property of $g_{\bar{\tau}_l}$ and ${}^{g_{\overline{\rho_{m_l l}}}} f$ for $m_l = 1, \ldots, n_l$ (here again we need the induction hypothesis for the arguments at hand), we get

$$g_{\bar{\tau}_l}(U_0) = g_{\bar{\tau}_l}(U_0') \wedge \bigvee_{m_l=1}^{n_l} ({}^{g_{\overline{\rho_{m_l l}}}} f)(U_{m_l}) = ({}^{g_{\overline{\rho_{m_l l}}}} f)(U_{m_l}');$$

then we immediately get

$$g_l(U_0, U_1, \ldots, U_{n_l}) = g_l(U_0', U_1', \ldots, U_{n_l}'),$$

so $f(U) = f(U')$, as we wanted. □

This indeed concludes our study of normal forms, since the general statement of Theorem 4.1, which we promised, is a direct corollary of Theorems 4.25 and 4.17.

Example. For the algebra $\mathbb{O}$ of ordinal numbers we have $\tilde{\kappa}_1(\mathbb{O}) = \mathbb{U}$, $\tilde{\kappa}_2(\mathbb{O}) = \mathbb{O}$, and $\tilde{\kappa}_3(\mathbb{O}) = \mathbb{N} \to \mathbb{O}$, so its construction type is $\tilde{\mathbb{O}} = \mathbb{U} + \mathbb{O} + (\mathbb{N} \to \mathbb{O})$, and the general form of a recursive normal form mapping on $\mathbb{O}$ is

$$\mathsf{nf}_{\mathbb{O}} = (g_1 + g_2 \circ \mathsf{nf}_{\mathbb{O}} + g_3 \circ {}^{\mathsf{nf}_{\mathbb{N}}}\mathsf{nf}_{\mathbb{O}}) \circ \mathsf{dst}_{\mathbb{O}},$$

with $\mathsf{nf}_{\mathbb{N}} : \mathrm{Con}_{\mathbb{N}} \to \mathrm{Con}_{\mathbb{N}}$ a normal form mapping on $\mathbb{N}$, and $g_1 : \mathrm{Con}_{\mathbb{U}} \to \mathrm{Con}_{\mathbb{O}}$, $g_2 : \mathrm{Con}_{\mathbb{O}} \to \mathrm{Con}_{\mathbb{O}}$, $g_3 : \mathrm{Con}_{\mathbb{N} \to \mathbb{O}} \to \mathrm{Con}_{\mathbb{O}}$ satisfying the following:

$$\bigvee_{U \in \mathrm{Con}_{\mathbb{U}}} g_1(U) \sim_{\mathbb{O}} \{0_{\mathbb{O}}\},$$

$$\bigvee_{U \in \mathrm{Con}_{\mathbb{O}}} g_2(U) \sim_{\mathbb{O}} \mathtt{S}_{\mathbb{O}} U,$$

$$\bigvee_{W \in \mathrm{Con}_{\mathbb{N} \to \mathbb{O}}} g_3(W) \sim_{\mathbb{O}} \mathtt{L}_{\mathbb{O}} W$$

(we keep in mind that the notations $L(W)$ and $\mathtt{L}_{\mathbb{O}} W$ stand for different things!). Let's choose the easy ones:

$$\mathsf{nf}_{\mathbb{N}}(U) := \mathsf{lub}_{\mathbb{N}}(U), \quad g_1(U) := \{0_{\mathbb{O}}\}, \quad g_2(U) := \mathtt{S}_{\mathbb{O}} U, \quad g_3(W) := \mathtt{L}_{\mathbb{O}} W.$$

Now consider the neighborhood

$$U = \{*_{\mathbb{O}}, \mathtt{L}\langle\{0_{\mathbb{N}}, *_{\mathbb{N}}\}, \mathtt{S}*_{\mathbb{O}}\rangle, \mathtt{L}\langle\{0_{\mathbb{N}}\}, \mathtt{SL}\langle\{0_{\mathbb{N}}\}, 0_{\mathbb{O}}\rangle\rangle,$$
$$\mathtt{L}\langle\{\mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, \mathtt{SSS0}_{\mathbb{N}}\}, \mathtt{SS}*_{\mathbb{O}}\rangle, \mathtt{L}\langle\{\mathtt{SSS}*_{\mathbb{N}}, \mathtt{SS}*_{\mathbb{N}}\}, \mathtt{SSS0}_{\mathbb{O}}\rangle\}.$$

We have $\mathsf{dst}_{\mathbb{O}}(U) = \mathtt{I}_3 W$, with

$$W = \{\langle\{0_{\mathbb{N}}, *_{\mathbb{N}}\}, \mathtt{S}*_{\mathbb{O}}\rangle, \langle\{0_{\mathbb{N}}\}, \mathtt{SL}\langle\{0_{\mathbb{N}}\}, 0_{\mathbb{O}}\rangle\rangle,$$
$$\langle\{\mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, \mathtt{SSS0}_{\mathbb{N}}\}, \mathtt{SS}*_{\mathbb{O}}\rangle, \langle\{\mathtt{SSS}*_{\mathbb{N}}, \mathtt{SS}*_{\mathbb{N}}\}, \mathtt{SSS0}_{\mathbb{O}}\rangle\},$$

so the computation of the normal form begins with the selection of the third branch of its definition, and we get

$$\mathsf{nf}_{\mathbb{O}}(U) = g_3({}^{\mathsf{nf}_{\mathbb{N}}}\mathsf{nf}_{\mathbb{O}}(W)) = g_3(\mathsf{nf}_{\mathbb{O}}^{\mathsf{nf}_{\mathbb{N}}}(\mathsf{eirr}(W))). \tag{3}$$

We need to determine the maximal eigen-neighborhoods of W, for which

$$\begin{aligned} L(W) &= \{\mathtt{0}_{\mathbb{N}}, *_{\mathbb{N}}, \mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, \mathtt{SSS0}_{\mathbb{N}}, \mathtt{SSS}*_{\mathbb{N}}\}, \\ R(W) &= \{\mathtt{S}*_{\mathbb{O}}, \mathtt{SL}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle, \mathtt{SS}*_{\mathbb{O}}, \mathtt{SSS0}_{\mathbb{O}}\}; \end{aligned}$$

letting $U' \in \mathrm{Con}_{L(W)}$, the following table sums up the results of the relevant computations:

$\overline{U'} \cap L(W)$	$\overline{WU'} \cap R(W)$	$\mathrm{Eig}_W^{\max}$
$\varnothing_{\mathbb{N}}$	$\varnothing_{\mathbb{O}}$	
$\{*_{\mathbb{N}}\}$	$\varnothing_{\mathbb{O}}$	
$\{\mathtt{0}_{\mathbb{N}}, *_{\mathbb{N}}\}$	$\{\mathtt{SL}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle, \mathtt{S}*_{\mathbb{O}}\}$	✓
$\{\mathtt{S}*_{\mathbb{N}}, *_{\mathbb{N}}\}$	$\varnothing_{\mathbb{O}}$	
$\{\mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, *_{\mathbb{N}}\}$	$\varnothing_{\mathbb{O}}$	
$\{\mathtt{SSS}*_{\mathbb{N}}, \mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, *_{\mathbb{N}}\}$	$\{\mathtt{SSS0}_{\mathbb{O}}, \mathtt{SS}*_{\mathbb{O}}, \mathtt{S}*_{\mathbb{O}}\}$	✓
$\{\mathtt{SSS0}_{\mathbb{N}}, \mathtt{SSS}*_{\mathbb{N}}, \mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, *_{\mathbb{N}}\}$	$\{\mathtt{SSS0}_{\mathbb{O}}, \mathtt{SS}*_{\mathbb{O}}, \mathtt{S}*_{\mathbb{O}}\}$	

Since

$$\begin{aligned} &\mathsf{lub}_{\mathbb{N}}(\{\mathtt{0}_{\mathbb{N}}, *_{\mathbb{N}}\}) = \{\mathtt{0}_{\mathbb{N}}\}, \\ &\mathsf{lub}_{\mathbb{N}}(\{\mathtt{SSS}*_{\mathbb{N}}, \mathtt{SS}*_{\mathbb{N}}, \mathtt{S}*_{\mathbb{N}}, *_{\mathbb{N}}\}) = \{\mathtt{SSS}*_{\mathbb{N}}\}, \end{aligned}$$

the computation (3) continues as

$$\begin{aligned} \mathsf{nf}_{\mathbb{O}}(U) = g_3\big(&\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathsf{nf}_{\mathbb{O}}(\{\mathtt{SL}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle, \mathtt{S}*_{\mathbb{O}}\})\rangle \\ &\cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, \mathsf{nf}_{\mathbb{O}}(\{\mathtt{SSS0}_{\mathbb{O}}, \mathtt{SS}*_{\mathbb{O}}, \mathtt{S}*_{\mathbb{O}}\})\rangle\big), \end{aligned} \tag{4}$$

so we have to recurse in $\mathsf{nf}_{\mathbb{O}}$ twice. We finish here the computation from (4) without explaining any further details.

$$\begin{aligned} \mathsf{nf}_{\mathbb{O}}(U) &= g_3\big(\langle\{\mathtt{0}_{\mathbb{N}}\}, g_2(\mathsf{nf}_{\mathbb{O}}(\{\mathtt{L}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle\}))\rangle \\ &\qquad \cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, g_2(\mathsf{nf}_{\mathbb{O}}(\{\mathtt{SS0}_{\mathbb{O}}, \mathtt{S}*_{\mathbb{O}}\}))\rangle\big) \\ &= g_3\Big(\Big\langle\{\mathtt{0}_{\mathbb{N}}\}, g_2(g_3(\mathsf{nf}_{\mathbb{O}}^{\mathsf{nf}_{\mathbb{N}}}(\{\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle\})))\Big\rangle \\ &\qquad \cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, g_2(g_2(\mathsf{nf}_{\mathbb{O}}(\{\mathtt{S0}_{\mathbb{O}}\})))\rangle\big) \\ &= g_3\big(\langle\{\mathtt{0}_{\mathbb{N}}\}, g_2(g_3(\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathsf{nf}_{\mathbb{O}}(\{\mathtt{0}_{\mathbb{O}}\})\rangle))\rangle \\ &\qquad \cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, g_2(g_2(g_2(\mathsf{nf}_{\mathbb{O}}(\{\mathtt{0}_{\mathbb{O}}\}))))\rangle\big) \\ &= g_3\big(\langle\{\mathtt{0}_{\mathbb{N}}\}, g_2(g_3(\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathsf{nf}_{\mathbb{O}}(\{\mathtt{0}_{\mathbb{O}}\})\rangle))\rangle \\ &\qquad \cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, g_2(g_2(g_2(\mathsf{nf}_{\mathbb{O}}(\{\mathtt{0}_{\mathbb{O}}\}))))\rangle\big) \\ &= g_3\big(\langle\{\mathtt{0}_{\mathbb{N}}\}, g_2(g_3(\{\langle\{\mathtt{0}_{\mathbb{N}}\}, g_1(\{\mathtt{0}_{\mathbb{U}}\})\rangle\}))\rangle \\ &\qquad \cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, g_2(g_2(g_2(g_1(\{\mathtt{0}_{\mathbb{U}}\}))))\rangle\big) \\ &= \ldots \\ &= g_3\big(\langle\{\mathtt{0}_{\mathbb{N}}\}, \{\mathtt{SL}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle\}\rangle \cup \langle\{\mathtt{SSS}*_{\mathbb{N}}\}, \{\mathtt{SSS0}_{\mathbb{O}}\}\rangle\big) \\ &= \{\mathtt{L}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{SL}\langle\{\mathtt{0}_{\mathbb{N}}\}, \mathtt{0}_{\mathbb{O}}\rangle\rangle, \mathtt{L}\langle\{\mathtt{SSS}*_{\mathbb{N}}\}, \mathtt{SSS0}_{\mathbb{O}}\rangle\} \end{aligned}$$

Now it is straightforward to check, for example, that we indeed have $\mathsf{nf}_{\mathbb{O}}(U) \sim_{\mathbb{O}} U$. □

5 Linearity and prime algebraicity

In a similar way that we can reduce the consistency to a binary predicate by coherence (1), we can reduce entailment to a binary predicate by an appropriate property. Call a neighborhood U *linear* if it satisfies

$$U \vdash b \leftrightarrow \underset{a \in U}{\exists} \{a\} \vdash b \tag{5}$$

for all b; in this case, write $U \vdash^{\mathsf{lin}} b$. For example every singleton forms a linear neighborhood. Call an information system, as well as its corresponding type, *linear* if (5) holds for all inhabited neighborhoods U and tokens b.

From a computational point of view, linearity ensures that in order to decide $U \vdash b$, we don't have to check $U_0 \vdash b$ for *all* $U_0 \subseteq U$, it suffices to just check it for the singleton ones; so a potentially exponential search is replaced by a linear one. A further technical advantage that linearity provides is in terms of presentation: we will see later that linear coherent information systems afford a representation by a structure with two predicates that are both merely binary. These can be seen as further simplifications of certain *preorders with consistency* that have appeared in the past [27, 33].

But apart from these relatively obvious technical advantages, there are deeper reasons for someone to want to work with linear systems. Since Gordon Plotkin [22] elaborated on the role of inherently nonsequential functionals like the "parallel or" in Scott's model, a lot of work focused on finding restricted models where this problem would not arise—this is the well known quest for a "fully abstract" model for Plotkin's PCF [8]. Gérard Berry noticed early that if a functional is to be called "sequential", it should at least be *stable* [5], that is, apart from being Scott continuous it should preserve consistent infima. Based on the work of Berry, the appropriate domains for stability are the so called "dI-domains", and, as Guo-Qiang Zhang [35, 36, 37] showed, in order to represent stable domains by information systems, it is necessary to require linearity. On the level of domains, this requirement translates to the distributivity of infima over suprema, a property that Glynn Winskel [32] showed to be equivalent to the notion of "prime algebraicity". In a different direction, such "prime algebraic" (and coherent) directed complete partial orders had already been employed by Winskel, together with Mogens Nielsen and Gordon Plotkin [20, 21], in the study of concurrency. Furthermore, since Jean-Yves Girard based his linear logic among other things on Berry's stability notion [10], linearity is a property that we encounter naturally when pursuing models of linear logic [38, 33, 7].

Now, it is easy to see that *flat* information systems induced by algebras are linear. In our *nonflat* setting, the base type $\mathbb{N}$ of natural numbers is linear: for example, the entailment $\{\mathtt{S}*, \mathtt{SS0}\} \vdash_{\mathbb{N}} \mathtt{SS}*$ is achieved by $\{\mathtt{SS0}\} \vdash_{\mathbb{N}} \mathtt{SS}*$. Similarly for the type $\mathbb{B}$ of booleans. Moreover, in Proposition 5.2 we will see that linearity, like coherence, is a property that function spaces preserve, so one could naturally argue that the choice of *linear* nonflat information systems should be perfectly legitimate. Indeed, one can establish several fundamental results in the nonflat setting based on linear systems alone: Helmut Schwichtenberg dealt with density, preservation of values, and adequacy [27], and we have also shown definability for the type system based on $\mathbb{B}$ and $\mathbb{N}$ [13, 12]; similar results were also obtained earlier by Fritz Müller [19].

Notwithstanding the above, linearity seems to be natural, or at least explicit, only for those systems that are built by constructors with at most unary arity, like $\mathbb{B}$ and $\mathbb{N}$. In contrast, in $\mathbb{D}$ for example, we would have

$$\{\mathtt{B0}*, \mathtt{B}*\mathtt{0}\} \vdash \mathtt{B00} \text{ but } \{\mathtt{B0}*, \mathtt{B}*\mathtt{0}\} \nvdash^{\mathsf{lin}} \mathtt{B00};$$

it is rather awkward, and certainly counter-intuitive, to have a notion of entailment where possession of the information `B0*` and `B*0` fails to provide the information `B00`.[10] It would then seem that linear systems might not be appropriate for a more inclusive theory of higher-type computability after all, and that one would need to deal with the generally nonlinear entailment in order to be on the safe side.

We see in this section that this is actually not the case. On the one hand, as was already foreshadowed in Proposition 4.9.2, nonlinear systems turn out to be “implicitly linear”: every neighborhood is equivalent to a linear one (Theorem 5.1). More importantly, we can make this hidden linearity explicit by restricting our models to linear subsystems, without losing in expressivity at all (Theorem 5.4). Based on these observations we will conclude by simplifying the presentation of the information systems at hand (Proposition 5.5) and showing that the domains induced are prime algebraic (Theorem 5.8).

5.1 Implicit linearity

Call a type *implicitly linear*, when every neighborhood has a *linear form*, that is, an equivalent neighborhood that is linear. It is not hard to see that all of our nonparametric finitary base types are implicitly linear, since there are normal form mappings, like cl or lub (see section 4.1), that take linear values; generally, we call a mapping $f : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$ a *linear form mapping (at type ρ)* when it preserves information and its values satisfy (5).

Just as we showed in section 4 that every type has normal forms by showing that there exist normal form mappings at every type, we will show now that every type is implicitly linear by showing the existence of linear form mappings at all types.

Theorem 5.1 (Implicit linearity)**.** *Every type is implicitly linear.*

Proof by induction over types. Let ρ be an arbitrary type. We show that there exists a neighborhood mapping $\mathsf{lin}_\rho : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$, such that $\mathsf{lin}_\rho(U)$ is linear and equivalent to U for all $U \in \mathrm{Con}_\rho$.

First we consider the case of a higher type $\rho \to \sigma$. Let $W \in \mathrm{Con}_{\rho\to\sigma}$; by Lemma 4.11, we may safely assume that it is eigen-irredundant. Set

$$\mathsf{lin}_{\rho\to\sigma}(W) := \mathsf{lin}_\sigma^{g_\rho}(W),$$

where $g_\rho : \mathrm{Con}_\rho \to \mathrm{Con}_\rho$ is some information-preserving mapping (for example, the identity), and lin_σ is provided by the induction hypothesis at σ. This is obviously a finite set that is equivalent to W by the induction hypothesis at σ and Corollary 4.15. It remains to show the implicit linearity property. Let $\langle U,b\rangle \in \mathrm{Tok}_{\rho\to\sigma}$ be such that $\mathsf{lin}_{\rho\to\sigma}(W) \vdash_{\rho\to\sigma} \langle U,b\rangle$. Then also $W \vdash_{\rho\to\sigma} \langle U,b\rangle$, which means that $W\upharpoonright_U \vdash_{\rho\to\sigma} \langle U,b\rangle$, or, equivalently, $U \vdash_\rho L(W\upharpoonright_U)$ and $R(W\upharpoonright_U) \vdash_\sigma b$; now, lin_σ maps to equivalent neighborhoods, so $\mathsf{lin}_\sigma(R(W\upharpoonright_U)) \vdash_\sigma b$; by the induction hypothesis at σ, there exists a single token $b_0 \in \mathsf{lin}_\sigma(R(W\upharpoonright_U))$ such that $\{b_0\} \vdash_\sigma b$, so for the token $c := \langle g_\rho(L(W\upharpoonright_U)), b_0\rangle$, which belongs to $\mathsf{lin}_{rho\to\sigma}(W)$ by definition, we have $\{c\} \vdash_{\rho\to\sigma} \langle U,b\rangle$, and we're done.

Now let $\iota = \mu_\xi(\kappa_1,\dots,\kappa_k)$ be a base type, with $\tilde{\kappa}_l(\iota)$ being

$$\overline{\tau_l} \times (\overline{\rho_{1l}} \to \iota) \times \cdots \times (\overline{\rho_{n_l l}} \to \iota)$$

[10] Linearity is one of several things that become subtly tricky when one decides to go nonflat. Compare with the issue of sequentiality, as discussed for example by Winskel [31, pp. 340–341].

for $l = 1,\dots,k$. Assume that $\mathsf{lin}_{\overline{\tau_l}}$ is a linear form mapping at $\overline{\tau_l}$ and that $g_{\overline{\rho_{m_l l}}}$ preserve the information at $\overline{\rho_{m_l l}}$ for all $m_l = 1,\dots,n_l$ (again, these could all be the respective identities), and consider the recursive mapping given by

$$\mathsf{lin}_\iota = \Big(\sum_{l=1}^{k} C_l \circ \mathsf{lin}_{\overline{\tau_l}} \times \prod_{m_l=1}^{n_l} {}^{g_{\overline{\rho_{m_l l}}}}\mathsf{lin}_\iota\Big) \circ \mathsf{dst}_\iota,$$

where C_l is here meant to stand for the mapping defined by constructor application. Let $U \in \mathrm{Con}_\iota$ and $b \in \mathrm{Tok}_\iota$. In case $\mathsf{cf}(U) = \{*_\iota\}$, we have $\mathsf{lin}_\iota(U) = \{*_\iota\} \sim_\iota U$, so information is preserved; furthermore, we have $\{*_\iota\} \vdash_\iota b$ if and only if $b = *_\iota$, so choosing $a := *_\iota \in \mathsf{lin}_\iota(U)$ witnesses linearity. In case $\mathsf{cf}(U) = C_l U_0 U_1 \cdots U_{n_l}$ for some appropriately typed neighborhoods $U_0, U_1, \dots, U_{n_l}$, then

$$\mathsf{lin}_\iota(U) = C_l \mathsf{lin}_{\overline{\tau_l}}(U_0)({}^{g_{\overline{\rho_{1l}}}}\mathsf{lin}_\iota)(U_1)\cdots({}^{g_{\overline{\rho_{n_l l}}}}\mathsf{lin}_\iota)(U_{n_l}) \overset{(\star)}{\sim_\iota} C_l U_0 U_1 \cdots U_{n_l} = U,$$

where at step $(\star)$ we used the hypothesis for $g_{\overline{\tau_l}}$ and the induction hypothesis for every ${}^{g_{\overline{\rho_{m_l l}}}}\mathsf{lin}_\iota$, $m_l = 1,\dots,n_l$, so information is preserved. To show linearity, assume without harming generality that $b = C_l b_0 b_1 \cdots b_{n_l}$, for appropriately typed tokens $b_0, b_1, \dots, b_{n_l}$; we have

$$\begin{aligned}
\mathsf{lin}_\iota(U) \vdash_\iota b &\Leftrightarrow C_l \mathsf{lin}_{\overline{\tau_l}}(U_0)({}^{g_{\overline{\rho_{1l}}}}\mathsf{lin}_\iota)(U_1)\cdots({}^{g_{\overline{\rho_{n_l l}}}}\mathsf{lin}_\iota)(U_{n_l}) \vdash_\iota C_l b_0 b_1 \cdots b_{n_l} \\
&\Leftrightarrow \mathsf{lin}_{\overline{\tau_l}}(U_0) \vdash_{\overline{\tau_l}} b_0 \wedge \forall_{m_l=1}^{n_l} ({}^{g_{\overline{\rho_{m_l l}}}}\mathsf{lin}_\iota)(U_{m_l}) \vdash_{\overline{\rho_{m_l l}} \to \iota} b_{m_l} \\
&\overset{(\star)}{\Rightarrow} \exists_{a_0 \in \mathsf{lin}_{\overline{\tau_l}}(U_0)} \{a_0\} \vdash_{\overline{\tau_l}} b_0 \wedge \forall_{m_l=1}^{n_l} \exists_{a_{m_l} \in ({}^{g_{\overline{\rho_{m_l l}}}}\mathsf{lin}_\iota)(U_{m_l})} \{a_{m_l}\} \vdash_{\overline{\rho_{m_l l}} \to \iota} b_{m_l} \\
&\overset{(\star\star)}{\Leftrightarrow} \exists_{a \in \mathsf{lin}_\iota(U)} \{a\} \vdash_\iota b;
\end{aligned}$$

at step $(\star)$ we use the linearity hypothesis of $\mathsf{lin}_{\overline{\tau_l}}$ as well as the induction hypothesis for every $({}^{g_{\overline{\rho_{m_l l}}}}\mathsf{lin}_\iota)$, and at step $(\star\star)$ we choose $a := C_l a_0 a_1 \cdots a_{n_l}$ as a witness. $\square$

Example. At type $\mathbb{D} \to \mathbb{D}$ consider the neighborhood

$$W = \{\langle\{\mathtt{B0*}\}, \mathtt{B1*}\rangle, \langle\{\mathtt{B*1}\}, \mathtt{B*0}\rangle\}.$$

It is nonlinear, since it needs both of its tokens to entail the token $c = \langle\{\mathtt{B01}\}, \mathtt{B10}\rangle$. To linearize it, the proof of Theorem 5.1 suggests that we use $\mathsf{lin}_{\mathbb{D}\to\mathbb{D}} = {}^{\mathsf{id}_\mathbb{D}}\mathsf{lin}_\mathbb{D}$, with

$$\mathsf{lin}_\mathbb{D} = (\mathtt{0}_\mathbb{D} + \mathtt{1}_\mathbb{D} + \mathtt{S}_\mathbb{D} \circ \mathsf{lin}_\mathbb{D} + \mathtt{B}_\mathbb{D} \circ \mathsf{lin}_\mathbb{D} \times \mathsf{lin}_\mathbb{D}) \circ \mathsf{dst}_\mathbb{D}.$$

So we get

$$\begin{aligned}
\mathsf{lin}_{\mathbb{D}\to\mathbb{D}}(W) &= {}^{\mathsf{id}_\mathbb{D}}\mathsf{lin}_\mathbb{D}(W) = \mathsf{lin}_\mathbb{D}^{\mathsf{id}_\mathbb{D}}(\mathsf{eirr}(W)) \\
&= \mathsf{lin}_\mathbb{D}^{\mathsf{id}_\mathbb{D}}(\langle\{\mathtt{B0*}\}, \{\mathtt{B1*}\}\rangle \cup \langle\{\mathtt{B*1}\}, \{\mathtt{B*0}\}\rangle \\
&\qquad \cup \langle\{\mathtt{B0*}, \mathtt{B*1}\}, \{\mathtt{B1*}, \mathtt{B*0}\}\rangle) \\
&= \langle\{\mathtt{B0*}\}, \mathsf{lin}_\mathbb{D}(\{\mathtt{B1*}\})\rangle \cup \langle\{\mathtt{B*1}\}, \mathsf{lin}_\mathbb{D}(\{\mathtt{B*0}\})\rangle \\
&\qquad \cup \langle\{\mathtt{B0*}, \mathtt{B*1}\}, \mathsf{lin}_\mathbb{D}(\{\mathtt{B1*}, \mathtt{B*0}\})\rangle \\
&= \langle\{\mathtt{B0*}\}, \{\mathtt{B1*}\}\rangle \cup \langle\{\mathtt{B*1}\}, \{\mathtt{B*0}\}\rangle \\
&\qquad \cup \langle\{\mathtt{B0*}, \mathtt{B*1}\}, \{\mathtt{B10}\}\rangle \\
&= \{\langle\{\mathtt{B0*}\}, \mathtt{B1*}\rangle, \langle\{\mathtt{B*1}\}, \mathtt{B*0}\rangle, \langle\{\mathtt{B0*}, \mathtt{B*1}\}, \mathtt{B10}\rangle\}.
\end{aligned} \tag{6}$$

This is now linear: it entails the above token c with the singleton $\{\langle\{\mathtt{B0*},\mathtt{B*1}\},\mathtt{B10}\rangle\}$.

For the sake of a slightly more involved example, consider the variation of ordinals given by the algebra $\mathbb{A} := \mu_\xi(\xi, \xi \to \xi, (\mathbb{D} \to \mathbb{D}) \to (\mathbb{N} \to \xi) \to \xi)$ (where the limit constructor is parametrized by values in $\mathbb{D} \to \mathbb{D}$), and then the type $\mathbb{L}(\mathbb{A}) = \mu_\xi(\xi, \mathbb{A} \to \xi \to \xi)$—which in this example we just write $\mathbb{L}$ for typographic convenience—with $\tilde{\mathbb{L}} = \mathbb{U} + \mathbb{A} \times \mathbb{L}$. The witness in the proof of Theorem 5.1 dictates the use of a recursive mapping

$$\mathsf{lin}_\mathbb{L} = (\mathtt{nil}_\mathbb{A} + \mathtt{cons}_\mathbb{A} \circ \mathsf{lin}_\mathbb{A} \times \mathsf{lin}_\mathbb{L}) \circ \mathsf{dst}_\mathbb{L},$$

with $\mathsf{lin}_\mathbb{A}$ (in the parameter position) in turn taken to be given recursively by

$$\mathsf{lin}_\mathbb{A} = \left(\mathtt{0}_\mathbb{A} + \mathtt{S}_\mathbb{A} \circ \mathsf{lin}_\mathbb{A} + \mathtt{L}_\mathbb{A} \circ \mathsf{lin}_{\mathbb{D}\to\mathbb{D}} \times {}^{\mathsf{id}_\mathbb{N}}\mathsf{lin}_\mathbb{A}\right) \circ \mathsf{dst}_\mathbb{A};$$

here, as in the proof of Theorem 5.1, $\mathtt{nil}_\mathbb{A}$, $\mathtt{cons}_\mathbb{A}$, $\mathtt{0}_\mathbb{A}$, $\mathtt{S}_\mathbb{A}$, $\mathtt{L}_\mathbb{A}$ stand for the respective neighborhood applications, $\mathsf{lin}_{\mathbb{D}\to\mathbb{D}}$ is the one that we already handled above, and $\mathsf{id}_\mathbb{D}$ is just the easiest choice of an information-preserving mapping that the proof demands. At type $\mathbb{L}$, the neighborhood $U = \{\mathtt{cons}_\mathbb{A} a_i l_i \mid i = 1,2,3\}$, with

$$\begin{aligned} a_1 &= \mathtt{L}_\mathbb{A}\langle U_1, b_1\rangle\langle U_1', b_1'\rangle, & l_1 &= *_\mathbb{L},\\ a_2 &= \mathtt{L}_\mathbb{A}\langle U_2, b_2\rangle\langle U_2', b_2'\rangle, & l_2 &= *_\mathbb{L},\\ a_3 &= *_\mathbb{A}, & l_3 &= \mathtt{nil}_\mathbb{A}, \end{aligned}$$

and

$$\begin{aligned} U_1 &= \{\mathtt{B}_\mathbb{D}\mathtt{0}_\mathbb{D}*_\mathbb{D}\}, & b_1 &= \mathtt{B}_\mathbb{D}\mathtt{1}_\mathbb{D}*_\mathbb{D}, & U_1' &= \{\mathtt{0}_\mathbb{N}\}, & b_1' &= *_\mathbb{A},\\ U_2 &= \{\mathtt{B}_\mathbb{D}*_\mathbb{D}\mathtt{1}_\mathbb{D}\}, & b_2 &= \mathtt{B}_\mathbb{D}*_\mathbb{D}\mathtt{0}_\mathbb{D}, & U_2' &= \{\mathtt{S}_\mathbb{N}*_\mathbb{N}\}, & b_2' &= \mathtt{0}_\mathbb{A}, \end{aligned}$$

is not linear, since it needs all of its elements to entail the token

$$b = \mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B}_\mathbb{D}\mathtt{0}_\mathbb{D}\mathtt{1}_\mathbb{D}\}, \mathtt{B}_\mathbb{D}\mathtt{1}_\mathbb{D}\mathtt{0}_\mathbb{D}\rangle\langle\{\mathtt{0}_\mathbb{N}\}, *_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}.$$

We have

$$\begin{aligned} \mathsf{lin}_\mathbb{L}(U) &= \mathtt{cons}_\mathbb{A}(\mathsf{lin}_\mathbb{A}(a_1,a_2,a_3), \mathsf{lin}_\mathbb{L}(l_1,l_2,l_3))\\ &= \mathtt{cons}_\mathbb{A}(\mathtt{L}_\mathbb{A}(\mathsf{lin}_{\mathbb{D}\to\mathbb{D}}(W), \mathsf{lin}_{\mathbb{N}\to\mathbb{A}}(W')), \mathtt{nil}_\mathbb{A}) \end{aligned}$$

(we write $\mathsf{lin}_\rho(a_1,\dots,a_m)$ for $\mathsf{lin}_\rho(\{a_1,\dots,a_m\})$), and we found the linear form of the neighborhood $W = \{\langle U_1,b_1\rangle, \langle U_2,b_2\rangle\}$ to be the one in (6), while the linear form of the neighborhood $W' = \{\langle U_1',b_1'\rangle, \langle U_2',b_2'\rangle\}$ is similarly found to be itself, that is, $\mathsf{lin}_{\mathbb{A}\to\mathbb{A}}(W') = W'$ (indeed, W' is already in linear form); so we get

$$\begin{aligned} \mathsf{lin}_\mathbb{L}(U) = \{ &\mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B0*}\},\mathtt{B1*}\rangle\langle\{\mathtt{0}_\mathbb{N}\},*_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}\\ &\mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B0*}\},\mathtt{B1*}\rangle\langle\{\mathtt{S}*_\mathbb{N}\},\mathtt{0}_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}\\ &\mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B*1}\},\mathtt{B*0}\rangle\langle\{\mathtt{0}_\mathbb{N}\},*_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}\\ &\mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B*1}\},\mathtt{B*0}\rangle\langle\{\mathtt{S}*_\mathbb{N}\},\mathtt{0}_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}\\ &\mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B0*},\mathtt{B*1}\},\mathtt{B10}\rangle\langle\{\mathtt{0}_\mathbb{N}\},*_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}\\ &\mathtt{cons}_\mathbb{A}\mathtt{L}_\mathbb{A}\langle\{\mathtt{B0*},\mathtt{B*1}\},\mathtt{B10}\rangle\langle\{\mathtt{S}*_\mathbb{N}\},\mathtt{0}_\mathbb{A}\rangle\mathtt{nil}_\mathbb{A}\}, \end{aligned}$$

which is linear: it entails the above token b with its second last element. ☐

5.2 Explicit linearity

The witness we provided in the proof of Theorem 5.1 fails to be a normal form mapping because of the identity, but the argument would work just fine with any other mapping instead of id (which we used for simplicity), as long as it sends a neighborhood to an equivalent neighborhood; so there are plenty of witnesses of implicit linearity that are actually normal form mappings. In particular, there are witnesses that ensure that not only the entailments at $\rho \to \sigma$, but also all entailments of lower type will be linear (see our last example). One may naturally wonder: can we not just restrict ourselves to appropriate normal forms at every type, and work exclusively in a linear setting?

Indeed we can, just not in a downright naive way. First of all we observe that if we achieve linearity at base types then we're done, since all function spaces will also be linear, by the following known result [27].

Proposition 5.2. *Let ρ, σ be coherent information systems, with σ being linear. Then their function space $\rho \to \sigma$ is a linear coherent information system.*

Proof. Let $W \in \mathrm{Con}^1_{\rho\to\sigma}$ and $\langle U, b\rangle \in \mathrm{Tok}_{\rho\to\sigma}$ be such that $W \vdash_{\rho\to\sigma} \langle U, b\rangle$, which means that $WU \vdash_\sigma b$. If WU is empty then b must be trivial, so any token of W is a witness. If WU is inhabited, by the linearity at σ we get some $b_0 \in WU$, such that $\{b_0\} \vdash_\sigma b$, so there exists a token $\langle U_0, b_0\rangle \in W$, such that $\{U_0, b_0\} \vdash_{\rho\to\sigma} \langle U, b\rangle$. □

Now in order to obtain linear information systems for our base types, we naturally turn to linear normal forms such as the ones we encountered in section 4.1—let's restrict ourselves to the nonparametric finitary case for the time being. The naive way to go about the problem is to consider a class $\mathrm{Con}^\star$ of neighborhoods in some normal form that we know is linear, and check if the triple $(\mathrm{Tok}, \mathrm{Con}^\star, \vdash)$ will do. The only two choices that we have from our previous discussion is to have $\mathrm{Con}^\star$ consisting of either the suprema or the deductive closures; it turns out that both are bad choices. If we restrict Con to the neighborhoods that can serve as suprema, that is, to *singletons*, we lose the propagation of consistency by entailment, and if we restrict it to deductively closed neighborhoods, then we lose the consistency of singletons and the closure of consistency to subsets. Trying to mend these shortcomings by tweaking, for example, the definition of entailment, only seems to lead further from intuition.

A less naive idea is to capitalize on our results on *path normal forms*, namely, by restricting the *carrier* set to paths, and then adapting consistency and entailment accordingly. Obviously, we have to adapt the relevant main notions and results from the nonparametric finitary case to the general one. We need predicates Tok^p_ρ and Con^p_ρ for every type ρ.

- Let $\iota = \mu_\xi(\kappa_1, \ldots, \kappa_k)$ be an arbitrary algebra. We have $*_\iota \in \mathrm{Tok}^p_\iota$, and for every $l = 1, \ldots, k$ with κ_l having $\rho_1, \ldots, \rho_r$ as argument types (parameters or recursive ones) and $a \in \mathrm{Tok}^p_{\rho_i}$ for $i = 1, \ldots, r$, we have $C_l \overrightarrow{*_\iota} a \overrightarrow{*_\iota} \in \mathrm{Tok}^p_\iota$ (where the vectors $\overrightarrow{*_\iota}$ may be empty).
- Let ρ and σ be types. If $U \in \mathrm{Con}^p_\rho$ and $b \in \mathrm{Tok}^p_\sigma$, then we have $\langle U, b\rangle \in \mathrm{Tok}^p_{\rho\to\sigma}$.
- Let ρ be any type. If $U \in \mathrm{Con}_\rho$ is such that $a \in \mathrm{Tok}^p_\rho$ for every $a \in U$, then we have $U \in \mathrm{Con}^p_\rho$.

Notice that we write above Con^p, and not Con^{pr}: in the last clause we also include neighborhoods of paths which are not "path reduced" in the sense of Proposition 4.5

and Theorem 4.6; obviously, $\mathrm{Con}^{pr} \subseteq \mathrm{Con}^{p}$. It is straightforward to see that Proposition 4.3.3 can be generalized to the case of an arbitrary algebra, that is, that even an infinitary algebra ι satisfies the property of *path linearity*: if $U \in \mathrm{Con}^1_\iota$ (not necessarily in Con^p_ι) and $b \in \mathrm{Tok}^p_\iota$, then $U \vdash_\iota b$ implies $U \vdash^{\mathrm{lin}}_\iota b$.

Write $\rho \cong \sigma$, if the ideals of ρ and the ideals of σ are in a bijective correspondence. Moreover, if $x, y : \rho$, write $x \vdash_\rho y$ to mean that for every $V \subseteq_f y$ there is a $U \subseteq_f x$ with $U \vdash_\rho V$. The following in a way expresses that the *non*-path tokens of an algebra form a "redundant" set [38].

Proposition 5.3. *Let ι be a base type. There exists a linear coherent information system η, such that $\eta \cong \iota$.*

Proofsketch. Given a base type ι, define ι^p by letting Tok_{ι^p} be Tok^p_ι, Con_{ι^p} be Con^p_ι, and $\vdash_{\iota^p}$ be $\vdash_\iota \cap (\mathrm{Con}_{\iota^p} \times \mathrm{Tok}_{\iota^p})$. It is straightforward to check that ι^p is indeed a coherent information system; we call it the *path subsystem* of ι.

To see that it is linear, let $U \in \mathrm{Con}^1_{\iota^p}$ and $b \in \mathrm{Tok}_{\iota^p}$ be such that $U \vdash_{\iota^p} b$. Since b is a path, by path linearity there is an $a \in U$ with $\{a\} \vdash_\iota b$. But a is itself a path, so $\{a\} \vdash_{\iota^p} b$.

For the equivalence of the ideals, we just have to observe that an ideal in ι also contains the path forms of each of its neighborhoods, and, vice versa, that by an ideal in ι^p we recover the non-path tokens by taking the ι-closures of its neighborhoods. So we consider the maps $F : \iota \to \iota^p$ and $G : \iota^p \to \iota$, defined by

$$F(x) := \bigcup_{U \subseteq_f x} \{a \in \mathrm{Tok}_{\iota^p} \mid U \vdash_\iota a\},$$
$$G(x) := \bigcup_{U \subseteq_f x} \{a \in \mathrm{Tok}_\iota \mid U \vdash_\iota a\};$$

using Theorem 5.1 and Theorem 4.25 for the recursive mapping

$$\mathsf{pth}_\iota = \Big(\sum_{l=1}^{k} g_l \circ \mathsf{pth}_{\overline{\tau_l}} \times \prod_{m_l=1}^{n_l} {}^{\mathsf{pth}_{\overline{\rho_{m_l l}}}}\mathsf{pth}_\iota\Big) \circ \mathsf{dst}_\iota$$

with

$$g_l(U_1, \ldots, U_{r_l}) := \bigcup_{i=1}^{r_l} C_l \overrightarrow{\{*_\iota\}} U_i \overrightarrow{\{*_\iota\}}$$

(r_l counting all parameters and all recursive arguments of C_l), it is by a tedious but straightforward mutual induction that we show F and G to be well defined, injective (note that $F(x) \sim_\iota x$ and $G(x) \sim_\iota x$), and mutually inverse, thus $\iota \cong \iota^p$. □

It follows that if we are willing to dispose of succinct representations like `B00`, and restrict to their path-neighborhood representations like $\{$`B0*`$,$`B*0`$\}$, we obtain a model of information systems, which are not only coherent, but also linear in an explicit way. More precisely, from Propositions 5.2 and 5.3, by letting ι' be ι^p for every algebra and $(\rho \to \sigma)'$ be $\rho' \to \sigma'$ for arbitrary types ρ and σ, we obtain the following.

Theorem 5.4 (Linearity)**.** *Let ρ be a type. There exists a linear coherent information system ρ', such that $\rho' \cong \rho$.*

5.3 Preordered tolerances and prime algebraicity

A direct consequence of linearity is the practical property of *prime algebraicity*. Since the late 70’s, when this notion was first introduced by Nielsen, Plotkin and Winskel [20, 21], the connection of linearity on the level of information systems and prime algebraicity on the level of domains has been made by various authors, for example by Zhang [37, 38], but also by Winskel [33]—who has otherwise studied prime algebraicity for domains arising from event structures [32], and also showed that prime algebraicity is in fact distributivity in disguise (see also [37]).

We will first simplify the path representation further, by choosing to work with a binary consistency predicate as a primary notion. Call a triple $(\mathrm{Tok}, \asymp, \geqslant)$ a *preordered tolerance* if $\geqslant$ is a preorder, that is, reflexive and transitive, $\asymp$ is a *tolerance relation* [34], that is, reflexive and symmetric, and furthermore

$$a \asymp b \wedge b \geqslant c \rightarrow a \asymp c$$

is satisfied for all $a, b, c \in \mathrm{Tok}$. An *ideal* in such a structure is to be understood as follows: for all $a, b \in x$ we have $a \asymp b$; for all $a \in x$ and all $b \in \mathrm{Tok}$, if $a \geqslant b$ then $b \in x$. It is straightforward to check the following.

Proposition 5.5. *If* $(\mathrm{Tok}, \mathrm{Con}, \vdash)$ *is a given linear coherent information system, we obtain a preordered tolerance by defining* $a \asymp b$ *if* $\{a, b\} \in \mathrm{Con}$ *and* $a \geqslant b$ *if* $\{a\} \vdash b$. *Conversely, if* $(\mathrm{Tok}, \asymp, \geqslant)$ *is a given preordered tolerance, we obtain a linear coherent information system by setting* $U \in \mathrm{Con}$ *if* $a \asymp b$ *for all* $a, b \in U$, *and* $U \vdash b$ *if there exists* $a \in U$ *such that* $a \geqslant b$, *or else* b *is trivial (meaning,* $a \geqslant b$ *for all* a*’s in the preordered tolerance). Moreover, the ideals of a linear coherent information system coincide with the ideals of its induced preordered tolerance, and vice versa.*

So the domains that we get from linear systems are exactly the domains that we get from preordered tolerances. Preordered tolerances have been used in the past instead of information systems [19, 7, 13]—it is for the purpose of clarity that we use the generic term “preordered tolerance” here, since this is in reality the quintessence of a linear coherent information system, and it would deserve the same name in another context. We should also note that Nielsen, Plotkin, and Winskel [21] use instead a dual preordered *in*tolerance, which they call “event structure”, to prove prime algebraicity.

Now consider a type ρ. Based on Theorem 5.4, the coherent information system, also denoted by ρ, that we use to interpret the type in section 2 is equivalent to a linear coherent information system ρ^p; this, in turn, induces an equivalent preordered tolerance, based on Proposition 5.5, which we denote here by $\dot{\rho}$. For the needs of this section we say *standard model* for the collection of all ρ’s and *path model* for the collection of their respective $\dot{\rho}$’s.

An element p of a domain is called *completely prime*, or just *prime*, if whenever $p \sqsubseteq \bigsqcup X$ for a bounded set X, there exists an $x \in X$ such that $p \sqsubseteq x$. It is easy to check that, in the standard model of type $\mathbb{D}$, the ideals $x_1 := \overline{\{\mathtt{B0*}\}}$ and $x_2 := \overline{\{\mathtt{B*0}\}}$ are prime, but this is not the case for the ideal $x_0 := \overline{\{\mathtt{B00}\}}$. The domain is called *prime algebraic* if every element is the least upper bound of its prime approximations—notice the symmetry with the standard “compact” algebraicity required of domains.

We now observe that the least upper bound of a bounded set of ideals in the path model is given by their union. In the standard model this is the case only for directed sets of ideals, while for bounded ones we need to take the deductive closure of the union. For example, since $\{\mathtt{B0*}, \mathtt{B*0}\} \vdash_{\mathbb{D}} \mathtt{B00}$, the least upper bound of x_1 and x_2 from above is not just $x_1 \cup x_2$ but $\overline{x_1 \cup x_2} = x_0$ (we extend here the use of the bar notation to potentially infinite consistent sets). Write $x \asymp y$ to mean $U \asymp V$ for all neighborhoods $U \subseteq x, V \subseteq y$.

Lemma 5.6. *Let ρ be some type, and x and y be ideals of $\dot{\rho}$. If $x \asymp_{\dot{\rho}} y$ then the set $x \cup y$ is the least upper bound of x and y.*

Proof. The less straightforward thing to show is that the union is indeed an ideal of $\dot{\rho}$, for which we need to check only the deductive closure. Let b be an arbitrary path at type ρ and assume that for some $U \subseteq x \cup y$ we have $U \vdash_{\dot{\rho}} b$; by linearity there exists some path $a \in U$ such that $\{a\} \vdash_{\dot{\rho}} b$. Regardless of $a \in x$ or $a \in y$, by the deductive closure of either x or y we have $b \in x \cup y$. □

Write $\overline{a}$ instead of $\overline{\{a\}}$ for deductive closures generated by single tokens; in the case of a preordered tolerance we actually mean the sets $\{b \mid a \geqslant b\}$. Write also Tok^i for the collection of informative, that is, nontrivial tokens. We will not elaborate on it here, but let us at least note that for the next two results we need a choice principle that enables us to find an element $x \in X$ when X is an inhabited bounded set of ideals.

Lemma 5.7. *At a type ρ, an ideal of $\dot{\rho}$ is prime if and only if it is of the form $\overline{\{a\}}$ for some $a \in \mathrm{Tok}^i_{\dot{\rho}}$.*

Proof. Let a be a nontrivial path, and X be a bounded family of ideals of $\dot{\rho}$, such that $\overline{a} \subseteq \bigsqcup X$. By Lemma 5.6 we get $\overline{a} \subseteq \bigcup X$, that is, $a \in \bigcup X$. Since a is nontrivial, the family X must be nonempty, so we have some $x \in X$ with $a \in x$; but x is deductively closed, so $\overline{a} \subseteq x$, which means that $\overline{a}$ is prime. Conversely, let p be a prime ideal of $\dot{\rho}$, and consider the family $\{a \mid a \in p\}$. We obviously have $p \subseteq \bigcup\{\overline{a} \mid a \in p\}$, which by primeness of p yields a single $a_p \in p$ such that $p \subseteq \overline{a_p}$; since also $\overline{a_p} \subseteq p$, we have $p = \overline{a_p}$. □

Theorem 5.8 (Prime algebraicity)**.** *At type ρ, the ideals of $\dot{\rho}$ form a prime algebraic coherent domain.*

Proof. It suffices to show the prime algebraicity. Let x be an arbitrary ideal of $\dot{\rho}$ and $X := \{\overline{a} \mid \overline{a} \subseteq x\}$, which by Lemma 5.7 contains all of its prime approximations; we show that $x = \bigcup X$. For some path b, if $b \in x$ then $\overline{b} \subseteq x$ by the closure of x, therefore $\overline{b} \in X$ by definition, and then $b \in \bigcup X$. Conversely, if $b \in \bigcup X$, then there exists some path a such that $b \in \overline{a} \subseteq x$, that is, $b \in x$. □

6 Discussion

We showed how we can circumvent the inherent combinatorial complexity of nonflat information systems by working with canonical and even normal forms, and we used these results to show that we can fully recover linearity in a setting with a full, nonlinear entailment predicate. As an application of the latter, we introduced preordered tolerances, simple versions of information systems, and used them to show that in the presence of linearity our domains are prime algebraic. There are several things that suggest themselves as next steps.

On the technical side, we made heavy use of neighborhood mappings between two information systems, preferring them to the whole class of approximable maps, but we did this on demand, in a lazy and almost haphazard way, guided primarily by the needs of the present paper. One could look soberly into the *theory of neighborhood mappings* in its own sake, from a systematic and wider viewpoint, say type- or category-theoretic.

In the case of normal forms, as well, there is a prospect of generalization, if one is interested in their totality. It is easy to see that the class of all normal forms at a given type can be arranged as a semigroup in two natural ways, namely both under composition, and under consistent union $U \mapsto f(U) \cup g(U)$, whenever $f(U) \asymp g(U)$ for every U; both of these structures lack their natural neutral element, namely the identity map and the constant $U \mapsto \varnothing$, respectively. Moreover, both of these semigroups are actually “bands” (every element is idempotent), the former is particularly “rectangular” (the equation $f \circ g \circ h = f \circ h$ holds for all f, g, h) and even “left-zero” (all elements are left-neutral), while the latter is commutative. We list this array of terms to merely indicate that the *theory of semigroups* [11] is rich enough to have thematized such concepts, which suggests that the deceivingly meager structure of a semigroup might still give good answers to general questions regarding the navigation in our quite complex nonflat domains of choice.

Last, but most relevant for our interest in higher-type computability, the main message of the paper in our view is that we can work linearly even with a nonlinear (that is, not necessarily binary) entailment, and therefore widen the scope of previous work [19, 27, 13] in a substantial way. There are two further points to stress. On the one hand, the manifestations of linearity that we have witnessed point to a possible way of internalizing lines of study such as *sequentiality, concurrency, and linear semantics in a nonflat framework*. There is an enormous amount of work invested in these matters by a lot of people already (see section 5 for some references), and to even adapt the basic ideas to our framework should be rewarding. The prime algebraicity that we demonstrated is already such a reward, which we earned fairly quickly and, given the relevant literature, quite expectedly, after having proved linearity. On the other hand, as we mentioned already in the introduction, we may hope that adapting such work to the nonflat case may even lead to unexpected and more powerful results. In pursuit of such aims, it seems likely that the tools we have used in this work, in particular the various sorts of neighborhood mappings that we encountered, may well prove to be indispensable.

Dues

For their lending an ear, their feedback, their suggestions, their clarifications, thanks to Helmut Schwichtenberg, Davide Rinaldi, Matthias Hofer, Rhea Marstaller, and Kenji Miyamoto. Matthias Hofer has also provided a proof for a special case of Theorem 4.17, using an earlier variant of eigen-neighborhood. The referees have helped and urged me to present a clearer, fuller and more correct picture of the results; thank you.

References

[1] Samson Abramsky and Achim Jung. Domain theory. In *Handbook of logic in computer science, Volume 3*, pages 1–168. Oxford University Press, 1994.

[2] Roberto M. Amadio and Pierre-Louis Curien. *Domains and lambda-calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1998.

[3] Steve Awodey. *Category theory*. Oxford University Press, 2nd edition, 2010.

[4] Ulrich Berger. Total sets and objects in domain theory. *Annals of Pure and Applied Logic*, 60(2):91–117, 1993.

[5] Gérard Berry. Stable models of typed λ-calculi. In *Automata, languages and programming, 5th Colloquium, Udine 1978*, volume 62 of *Lecture Notes in Computer Science*, pages 72–89, 1978.

[6] Marc Bezem, Jan Willem Klop, and Roel de Vrijer, editors. *Term rewriting systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[7] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Antonino Salibra. On linear information systems. In *Linearity, First International Workshop, Coimbra 2009*, volume 22 of *Electronic Proceedings in Theoretical Computer Science*, pages 38–48. Open Publishing Association, 2010.

[8] Pierre-Louis Curien. Definability and full abstraction. In *Computation, meaning, and logic. Articles dedicated to Gordon Plotkin*, volume 172 of *Electronic Notes in Theoretical Computer Science*, pages 301–310. Elsevier, 2007.

[9] Martín H. Escardó. On lazy natural numbers with applications to computability theory and functional programming. *ACM SIGACT News*, 24(1):61–67, 1993.

[10] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[11] John M. Howie. *Fundamentals of semigroup theory*, volume 12 of *London Mathematical Society Monographs*. Clarendon Press, 1995.

[12] Simon Huber, Basil A. Karádais, and Helmut Schwichtenberg. Towards a formal theory of computability. In *Ways of Proof Theory*, volume 2 of *Ontos Mathematical Logic*, pages 257–282. Ontos Verlag, 2010.

[13] Basil A. Karádais. *Towards an Arithmetic with Approximations*. PhD thesis, Ludwig-Maximilians-Universität München, 2013.

[14] Basil A. Karádais. Atomicity, coherence of information, and point-free structures. *Annals of Pure and Applied Logic*, 167(9):753–769, 2016.

[15] Basil A. Karádais. Nonflatness and totality. Submitted, 2016.

[16] Robin Milner. Fully abstract models of typed λ-calculi. *Theoretical Computer Science*, 4:1–22, 1977.

[17] Kenji Miyamoto. *Program Extraction from Coinductive Proofs and its Application to Exact Real Arithmetic*. PhD thesis, Ludwig-Maximilians-Universität München, 2013.

[18] Kenji Miyamoto and Helmut Schwichtenberg. Program extraction in exact real arithmetic. *Mathematical Structures in Computer Science*, 25(8):1692–1704, 2015.

[19] Fritz Müller. Full abstraction for a recursively typed lambda calculus with parallel conditional. Technical Report SFB 124, FB 14, Universität des Saarlandes, 1993.

[20] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains. In *Semantics of concurrent computation, International Symposium, Evian 1979*, volume 70 of *Lecture Notes in Computer Science*, pages 266–284, 1979.

[21] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains. i. *Theoretical Computer Science*, 13:85–108, 1981.

[22] Gordon D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, 1977/78.

[23] Gordon D. Plotkin. T^ω as a universal domain. *Journal of Computer and System Sciences*, 17(2):209–236, 1978.

[24] Florian Ranzi. Recursion and pcf-definability over the partial continuous functionals. Master's thesis, Ludwig-Maximilians-Universität München, 2008.

[25] Davide Rinaldi. *Formal methods in the theories of rings and domains*. PhD thesis, Ludwig-Maximilians-Universität München, 2014.

[26] Helmut Schwichtenberg. Eine Normalform für endliche Approximationen von partiellen stetigen Funktionalen. In *Logik und Grundlagenforschung, Festkolloquium zum 100. Geburtstag von Heinrich Scholz*, pages 89–95, 1986.

[27] Helmut Schwichtenberg. Recursion on the partial continuous functionals. In *Logic Colloquium '05*, volume 28 of *Lecture Notes in Logic*, pages 173–201, 2006.

[28] Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and computations*. Perspectives in Logic. Cambridge University Press, 2012.

[29] Dana S. Scott. Domains for denotational semantics. In *Automata, languages and programming, Aarhus 1982*, volume 140 of *Lecture Notes in Computer Science*, pages 577–613. Springer, 1982.

[30] Viggo Stoltenberg-Hansen, Ingrid Lindström, and Edward R. Griffor. *Mathematical theory of domains*, volume 22 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1994.

[31] Glynn Winskel. Event structures. In *Advances in Petri nets 1986, Part II, Bad Honnef 1986*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer, 1987.

[32] Glynn Winskel. An introduction to event structures. In *Linear time, branching time and partial order in logics and models for concurrency, Noordwijkerhout 1988*, volume 354 of *Lecture Notes in Computer Science*, pages 364–397, 1989.

[33] Glynn Winskel. Prime algebraicity. *Theoretical Computer Science*, 410(41):4160–4168, 2009.

[34] E. Christopher Zeeman. The topology of the brain and visual perception. In *Topology of 3-manifolds and related topics, University of Georgia Institute 1961*, pages 240–256. Prentice Hall, 1962.

[35] Guo-Qiang Zhang. dI-Domains as information systems. Technical Report DAIMI-PB-282, Aarhus University, 1989.

[36] Guo-Qiang Zhang. *Logic of domains*. Progress in Theoretical Computer Science. Birkhäuser, 1991.

[37] Guo-Qiang Zhang. dI-Domains as prime information systems. *Information and Computation*, 100(2):151–177, 1992.

[38] Guo-Qiang Zhang. Quasi-prime algebraic domains. *Theoretical Computer Science*, 155(1):221–264, 1996.