Parallelism in Realizability Models

Satoshi Nakata

Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan

Abstract. Study of parallel operations such as Plotkin's parallel-or has promoted the development of the theory of programming languages. In this paper, we consider parallel operations in the framework of categorical realizability. Given a partial combinatory algebra A equipped with an "abstract truth value" Σ (called predominance), we introduce the notions of Σ-or and Σ-and combinators in A. By choosing a suitable A and Σ, a form of parallel-or may be expressed as a Σ-or combinator. We then investigate the relationship between these combinators and the realizability model $\mathbf{Ass}(A)$ (the category of assemblies over A) and show the following: under a natural assumption on Σ, (i) A admits Σ-and combinator iff for any assembly $X \in \mathbf{Ass}(A)$ the Σ-subsets (canonical subassemblies) of X form a poset with respect to inclusion. (ii) A admits both Σ-and and Σ-or combinators iff for any $X \in \mathbf{Ass}(A)$ the Σ-subsets of X form a lattice with respect to intersection and union.

Keywords: Realizability · Partial combinatory algebra · Parallel-or function.

1 Introduction

Traditionally, the *realizability interpretation* has been introduced as semantics of intuitionistic arithmetic. It rigorously defines "what it means to justify a proposition by an algorithm." While it is originally formulated in terms of recursive functions [8], it is later generalized to a framework based on *Partial Combinatory Algebras* (PCAs), which include various computational models. The interpretation itself has been given a categorical generalization, such as the *realizability topos* and the category of *assemblies*. In particular, in the category $\mathbf{Ass}(A)$ of assemblies over PCA A, we can discuss implementation of mathematical structures and functions by algorithms [19]. Moreover, $\mathbf{Ass}(A)$ provide effective models to higher-order programming languages such as PCF [1,9,14].

In this paper, we will consider how the structure of the realizability model $\mathbf{Ass}(A)$ is affected by the choice of a computational model A. More specifically, we focus on the following two concepts.

I. Parallel operations in PCA:
Comparing Kleene's first algebra $\mathcal{K}_1$ and term models of lambda calculus as PCA, there is a difference in the degree of parallelism. For example, term models exclude Plotkin's *parallel-or* function [16], whereas $\mathcal{K}_1$ does not. While such a parallel operation has received a lot of attention in the

theory of programming languages, it also plays an implicit role in elementary recursion theory. For example, the union of two semi-decidable sets $U, V \subseteq \mathbb{N}$ is again semi-decidable precisely because a Turing machine can check whether input $n \in \mathbb{N}$ belongs to U or to V *in parallel*. In this paper, we first consider a pair of nonempty subsets $\Sigma = (T, F)$ of PCA A as an "abstract truth value" and define combinators Σ-or and Σ-and in A. In a suitable A, these notions may express a form of parallel operations.

II. Σ-subsets in $\mathbf{Ass}(A)$:
It is known that such a pair $\Sigma = (T, F)$ may be identified with a *predominance* $t : 1 \to \Sigma$ in $\mathbf{Ass}(A)$, which is a morphism obtained by weakening the condition for being a subobject classifier [9]. An important feature is that, for every assembly X over A, Σ induces a certain class of "canonical" subassemblies of X. It is called the class of *Σ-subsets* of X and is written $\mathrm{Sub}_\Sigma(X)$. Unlike the subobject lattice $\mathrm{Sub}(X)$, $\mathrm{Sub}_\Sigma(X)$ does not form a poset in general. When it does, Σ is called *dominance* and used to construct a subcategory of (internal) domains in the context of *Synthetic domain theory* [7,9,13,14,18].
Interestingly, considering a suitable Σ in $\mathbf{Ass}(\mathcal{K}_1)$, the Σ-subsets of a natural number object exactly correspond to the semi-decidable subsets of $\mathbb{N}$ [11]. That is, the notion of Σ-subset can be regarded as a generalization of semi-decidable set. From the discussion in I., we can expect that if A admits Σ-or, then $\mathrm{Sub}_\Sigma(X)$ is closed under union.

The purpose of this paper is to give a precise correspondence between these two concepts. In particular, we prove the following results. Under a natural assumption on a predominance Σ, A admits Σ-and combinator if and only if, for every assembly X, the Σ-subsets of X form a poset with respect to inclusion (Theorem 24). Furthermore, A admits both Σ-and and Σ-or combinators if and only if, for every assembly X, the Σ-subsets of X form a lattice with respect to intersection and union (Theorem 28).

Outline

The structure of this paper is as follows. In Section 2, we give some basic definitions and properties about PCAs. In Section 3, we introduce the notions of Σ-or and Σ-and combinators in a PCA relative to an "abstract truth value" (predominance) Σ. In Section 4, we proceed to the category $\mathbf{Ass}(A)$ of assemblies over A and the notion of Σ-subset. Lastly, in Section 5, we discuss the relationship between Σ-or and Σ-and combinators in A and the structure of the Σ-subsets in $\mathbf{Ass}(A)$.

2 Preliminary

We review some basic concepts and notations in realizability theory.

Definition 1 ([9]). *A* partial combinatory algebra (PCA) *is a set A equipped with a partial binary operation $\cdot : A \times A \rightharpoonup A$ such that there exist elements* k, $\mathrm{s} \in A$ *satisfying the conditions*

$$\mathrm{k} \cdot x \downarrow, \quad (\mathrm{k} \cdot x) \cdot y = x, \quad (\mathrm{s} \cdot x) \cdot y \downarrow, \quad ((\mathrm{s} \cdot x) \cdot y) \cdot z \cong (x \cdot z) \cdot (y \cdot z)$$

for any $x, y, z \in A$. Here $\downarrow$ is to be read as "defined" (and $\uparrow$ as "undefined") and $\cong$ means that if one side is defined, then so is the other and they are equal. We often write xy instead of $x \cdot y$, and axy instead of $(ax)y$. A PCA is called total *if its operation is total. Obviously, a singleton forms a total PCA, that is called a* trivial *PCA.*

PCA is often regarded as an "abstract machine" and there are many interesting examples: Turing machines, λ-calculus, the continuous functions of type $\omega^{\omega} \to \omega$, a reflexive object in any cartesian-closed category [19]. A common feature of PCAs is that they can imitate untyped λ-calculus as follows.

Notation 2 *Let $T(A)$ denote the set of terms generated by constants a, b, $\cdots \in A$, variables x, y, $\cdots$ and binary function symbol $\cdot$. We write $FV(t)$ for the set of free variables occurring in $t \in T(A)$.*

Given a term $t \in T(A)$ and a variable x, we define a new term $\lambda^* x.\, t$ by induction on the structure of t. For instance, $\lambda^* x.\, x$ is defined by skk, $\lambda^* x.\, t$ by kt if t is either a variable $y \neq x$ or a constant a, and $\lambda^* x.\, tt'$ by $\mathrm{s}(\lambda^* x.\, t)(\lambda^* x.\, t')$. By repetition, we obtain an element $\lambda^* \boldsymbol{x}.\, t(\boldsymbol{x})$ in A for any $\boldsymbol{x} = x_1, \cdots, x_n$.

Theorem 3 ([9,19]). *Let A be a PCA and $t(\boldsymbol{x}) \in T(A)$. Then, for any $a_1, \cdots, a_n \in A$, $(\lambda^* \boldsymbol{x}.\, t(\boldsymbol{x}))a_1 \cdots a_{n-1}$ is defined and $(\lambda^* \boldsymbol{x}.\, t(\boldsymbol{x}))a_1 \cdots a_n \cong t(a_1, \cdots, a_n)$ holds.*

Remark 4. In particular, $\lambda^* x.\, (ab) :\equiv \mathrm{s}(\mathrm{k}a)(\mathrm{k}b) \in A$ is always defined even if $a \cdot b \uparrow$. This dummy λ-abstraction is useful to lock the evaluation. It may be later unlocked by applying it to an arbitrary element c in A:

$$(\lambda^* x.\, ab) \cdot c \cong a \cdot b.$$

This technique is used in Sections 3 and 5.

Notation 5 *We use the following notations:* $\mathrm{i} :\equiv \lambda^* x.\, x$, $\mathrm{true} :\equiv \lambda^* xy.\, x$, $\mathrm{false} :\equiv \lambda^* xy.\, y$, $(\text{if } b \text{ then } x \text{ else } y) :\equiv bxy$, $\langle x, y \rangle :\equiv \lambda^* z.\, zxy$, $\mathrm{fst} :\equiv \lambda^* p.\, p(\mathrm{true})$, $\mathrm{snd} :\equiv \lambda^* p.\, p(\mathrm{false})$.

In this paper, we are mainly interested in the following examples.

Example 6. (i) **Kleene's first algebra** $\mathcal{K}_1$: Consider the set of natural numbers $\mathbb{N}$ with a partial operation $\cdot : \mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$ defined by $n \cdot m :\equiv [\![n]\!](m)$, where $[\![n]\!]$ is the n-th partial computable function (with respect to a fixed effective numbering of Turing machines). This PCA is called *Kleene's first algebra* and is denoted by $\mathcal{K}_1$. The undefinedness $\uparrow$ of $a \cdot b$ can be regarded as divergence of computation.

(ii) **λ-term models**: Let Λ^0 be the set of *closed* λ-terms and T a *λ-theory*, that is, a congruence relation on λ-terms which contains β-equivalence. Considering the quotient modulo T, we obtain a total PCA Λ^0/T equipped with the application operation.

Another variation of λ-term model is given based on the *call-by-value* reduction strategy on Λ^0. A *value* is either an abstraction $\lambda x.\, M$ or a variable x. Values are denoted by V, W and the set of closed values by Λ^0_v. According to [5, Definition 7], we define $\to_{cbv}$ by the following binary relation (where $\overline{N} \equiv N_1, \cdots, N_n$ with $n \geq 0$):

$$\frac{}{(\lambda x.\, M)V\overline{N} \to_{cbv} M[V/x]\overline{N}} \qquad \frac{M \to_{cbv} M'}{VM\overline{N} \to_{cbv} VM'\overline{N}}$$

That is, one reduces a term from left to right with the constraint that the β-reduction can be applied only when the argument is a value. The transitive reflexive closure of $\to_{cbv}$ is denoted by $\twoheadrightarrow_{cbv}$. Note that the above reduction is called the left reduction in Plotkin's seminal work [15].

Define a partial operation $\cdot : \Lambda^0_v \times \Lambda^0_v \rightharpoonup \Lambda^0_v$ by:

$$V_1 \cdot V_2 := W \;\text{ if } V_1 V_2 \twoheadrightarrow_{cbv} W \text{ and } W \in \Lambda^0_v.$$

Otherwise, $V_1 \cdot V_2$ is undefined. Together with combinators $S := \lambda xyz.\, xz(yz)$ and $K := \lambda xy.\, x$, we obtain a non-total PCA $(\Lambda^0_v, \cdot)$.

3 Parallel combinators in PCA

Recall that Plotkin's parallel-or function $\mathrm{por}^{\mathrm{p}}$, originally introduced in the context of PCF [16], behaves as follows:

$$\begin{array}{ll} \mathrm{por}^{\mathrm{p}} MN \Downarrow \mathrm{true} & \text{if } M \Downarrow \mathrm{true} \text{ or } N \Downarrow \mathrm{true}, \\ \mathrm{por}^{\mathrm{p}} MN \Downarrow \mathrm{false} & \text{if } M \Downarrow \mathrm{false} \text{ and } N \Downarrow \mathrm{false}, \\ \mathrm{por}^{\mathrm{p}} MN \Uparrow & \text{otherwise} \end{array}$$

(where M, N are terms and $M \Downarrow V$ means that M evaluates to a value V). The point is that evaluation of a term may diverge. Hence one has to evaluate the arguments M, N in parallel to check if $\mathrm{por}^{\mathrm{p}} MN \Downarrow \mathrm{true}$. Given $\mathrm{por}^{\mathrm{p}}$, we may define a term por such that

$$(1) \qquad \mathrm{por} MN \Downarrow \quad \text{iff} \quad M \Downarrow \text{ or } N \Downarrow,$$

that may be seen as a weaker form of parallel-or. We now consider such operations in a PCA A. To make things as general as possible, we define them relative to two nonempty subsets (T, F) of A, which stand for "true/termination" and "false/failure", respectively.

The idea of dealing with two nonempty subsets of A is due to Longley. Actually he considered a more general notion of divergence in [9,10]. As he pointed out, these data correspond to a *predominance* in the category $\mathbf{Ass}(A)$ of assemblies.

Definition 7. *Given S_0, $S_1 \subseteq A$, we define $S_0 \times S_1 \coloneqq \{ \langle a_0, a_1 \rangle \in A \mid a_0 \in S_0$ and $a_1 \in S_1 \}$.*

We call a pair $\Sigma = (T, F)$ of nonempty subsets of A, which need not be disjoint, a predominance *on A. An element $\mathrm{or}_\Sigma \in A$ is called a Σ-*or *combinator if it satisfies*

$$\mathrm{or}_\Sigma(T \times T) \subseteq T, \quad \mathrm{or}_\Sigma(T \times F) \subseteq T,$$
$$\mathrm{or}_\Sigma(F \times T) \subseteq T, \quad \mathrm{or}_\Sigma(F \times F) \subseteq F.$$

*To be precise, $\mathrm{or}_\Sigma(T \times T) \subseteq T$ means that for every $f, g \in T$, $\mathrm{or}_\Sigma\langle f, g\rangle$ is defined and belongs to T. Dually, an element $\mathrm{and}_\Sigma \in A$ is called a Σ-*and *combinator if it satisfies*

$$\mathrm{and}_\Sigma(T \times T) \subseteq T, \quad \mathrm{and}_\Sigma(T \times F) \subseteq F,$$
$$\mathrm{and}_\Sigma(F \times T) \subseteq F, \quad \mathrm{and}_\Sigma(F \times F) \subseteq F.$$

We say that A admits Σ-or *if there exists or_Σ in A, and similarly for Σ-and.*

Example 8. Let $\Sigma_\mathrm{d} \coloneqq (\{ \mathrm{true} \}, \{ \mathrm{false} \})$. Then, every PCA admits Σ_d-or and Σ_d-and because $\mathrm{or}_{\Sigma_\mathrm{d}}$ can be defined as

$$\lambda^* p.\, (\text{if } \mathrm{fst} \cdot p \text{ then true else } (\text{if } \mathrm{snd} \cdot p \text{ then true else false})),$$

and similarly for $\mathrm{and}_{\Sigma_\mathrm{d}}$.

Example 9. Berry showed the following sequentiality theorem. Consider a λ-theory $T_{\mathcal{BT}}$ that identifies λ-terms which have the same Böhm tree. In the PCA $\Lambda^0/T_{\mathcal{BT}}$, there is no term M such that

$$M\langle \mathrm{i}, \Omega\rangle = M\langle \Omega, \mathrm{i}\rangle = \mathrm{i}, \qquad M\langle \Omega, \Omega\rangle = \Omega,$$

where $\Omega \coloneqq (\lambda x.\, xx)(\lambda x.\, xx)$ (See [2]). Hence $\Lambda^0/T_{\mathcal{BT}}$ does not admit Σ-or with respect to $\Sigma = (\{ \mathrm{i} \}, \{ \Omega \})$.

We next introduce an important predominance, which works uniformly for all non-total PCAs. This example is essentially due to Mulry.

Definition 10 ([11]). *For a non-total A, define a predominance $\Sigma_\mathrm{sd} \coloneqq (T_\mathrm{sd}, F_\mathrm{sd})$ by*

$$T_\mathrm{sd} \coloneqq \{ a \in A \mid a \cdot \mathrm{i} \downarrow \}, \quad F_\mathrm{sd} \coloneqq \{ a \in A \mid a \cdot \mathrm{i} \uparrow \}.$$

By definition, every Σ_sd-or combinator satisfies $\mathrm{or}_{\Sigma_\mathrm{sd}}\langle f, g\rangle \downarrow$ and

$$\mathrm{or}_{\Sigma_\mathrm{sd}}\langle f, g\rangle \cdot \mathrm{i} \downarrow \quad \text{iff} \quad f \cdot \mathrm{i} \downarrow \text{ or } g \cdot \mathrm{i} \downarrow$$

for every $f, g \in A$. In analogy with (1), we simply call $\mathrm{or}_{\Sigma_\mathrm{sd}}$ a *parallel-or* combinator and dually call $\mathrm{and}_{\Sigma_\mathrm{sd}}$ a *parallel-and*. We have chosen i as the "key" to "unlock" the evaluation, but actually it can be anything.

Proposition 11. *For every non-total PCA A, A admits parallel-or $\mathrm{or}_{\Sigma_\mathrm{sd}}$ if and only if A has a combinator por^u that satisfies $\mathrm{por}^\mathrm{u}\langle f, g\rangle \downarrow$ and*

$$\mathrm{por}^\mathrm{u}\langle f, g\rangle \cdot a \downarrow \quad \text{iff} \quad f \cdot a \downarrow \text{ or } g \cdot a \downarrow$$

for any $f, g, a \in A$.

Let $\mathrm{dom}(f)$ denote the set $\{ a \in A \mid f \cdot a \downarrow \}$. Then we have $\mathrm{dom}(\mathrm{por}^{\mathrm{u}}\langle f, g\rangle) = \mathrm{dom}(f) \cup \mathrm{dom}(g)$. Since subsets of the form $\mathrm{dom}(f)$ are precisely the semi-decidable sets (computably enumerable sets) in $\mathcal{K}_1$, we may claim that our parallel-or combinator has a generalized ability to take the union of two semi-decidable sets.

Let us now examine which PCA admits parallel-and (resp. parallel-or). We may expect that any PCA has a combinator which behaves as follows: "evaluate $f \cdot \mathrm{i}$ first; if it terminates, evaluate $g \cdot \mathrm{i}$ next." If we try to express this by a λ-term, we get a Σ_{sd}-and combinator (parallel-and).

Theorem 12. *Every non-total PCA admits parallel-and.*

On the other hand, parallel-or is more subtle. It is certainly true that Turing machines can perform a computation like: "evaluate $f \cdot \mathrm{i}$ and $g \cdot \mathrm{i}$ in parallel until one of them terminates." However, such a computation cannot be performed in λ-calculus due to its sequential nature. Consequently,

Proposition 13. *$\mathcal{K}_1$ admits both parallel-and and parallel-or, while Λ_v^0 admits parallel-and but not parallel-or.*

4 Predominances in the category of assemblies

In the modern theory of realizability, one builds a category over a given PCA A, in such a way that elements of A are used to implement a function or to justify a proposition in the constructive sense. There are several examples such as the *realizability topos* $\mathbf{RT}(A)$, the category $\mathbf{Ass}(A)$ of *assemblies* and the category $\mathbf{Mod}(A)$ of *modest sets* [19]. In particular, considering $\mathbf{RT}(\mathcal{K}_1)$, we can obtain the *effective topos* of Hyland [6] and the standard interpretation of first-order number theory in $\mathbf{RT}(\mathcal{K}_1)$ precisely corresponds to Kleene's traditional realizability interpretation [8]. In this sense, such categories are called "realizability models" in the literature.

We here focus on $\mathbf{Ass}(A)$, a full subcategory of $\mathbf{RT}(A)$. Notably, the latter can be obtained from the former by the *exact completion* [3,9]. $\mathbf{Ass}(A)$ is more primitive than $\mathbf{RT}(A)$ and is sufficiently rich as semantics of programming languages [1,9,14].

Definition 14. *An* assembly *over A is a pair $X = (|X|, ||\cdot||_X)$, where $|X|$ is a set and $||\cdot||_X : |X| \to \mathcal{P}(A)$ is a function such that $||x||_X$ is nonempty for any $x \in |X|$. An element $a \in A$ is called a* realizer *of x if $a \in ||x||_X$. A morphism of assemblies $f : (|X|, ||\cdot||_X) \to (|Y|, ||\cdot||_Y)$ is a function $f : |X| \to |Y|$ which has a* realizer *$r_f \in A$, that is, for any $x \in |X|$ and $a \in ||x||_X$, $r_f a$ is defined and in $||f(x)||_Y$. We say that r_f* realizes *f.*

One can verify that the assemblies and morphisms over A form a category $\mathbf{Ass}(A)$ (whose composition and identity are inherited from the category of sets). It has a terminal object given by $1 := (\{ * \}, ||\cdot||_1)$ with $||*||_1 := A$. Furthermore, $\mathbf{Ass}(A)$ always has a natural number object (NNO) N. For example, a canonical

NNO in $\mathbf{Ass}(\mathcal{K}_1)$ is given by $N := (\mathbb{N}, ||\cdot||_N)$ with $||n||_N := \{\, n \,\}$. The hom-set on N exactly corresponds the set of total computable functions on $\mathbb{N}$.

$\mathbf{Ass}(A)$ is a finitely complete locally cartesian-closed category [9,19]. This is a common feature of toposes such as the category of sets and realizability toposes. Every topos, in addition, has a subobject classifier, while $\mathbf{Ass}(A)$ does not unless A is trivial. Nevertheless, as one can see in [9,14], there is a useful concept of a "restricted classifier". Recall that a morphism $t : 1 \rightarrowtail \Sigma$ in a finitely complete category is a *subobject classifier* if for every monomorphism $m : U \rightarrowtail X$ there is *exactly one* morphism $\chi_m : X \to \Sigma$ which gives a pullback diagram

$$\begin{array}{ccc} U & \xrightarrow{\;!\;} & 1 \\ {\scriptstyle m}\downarrow & \lrcorner & \downarrow{\scriptstyle t} \\ X & \xrightarrow[\chi_m]{} & \Sigma. \end{array}$$

χ_m is called the *characteristic map* of m. By slightly weakening the condition, we obtain the concept of predominance.

Definition 15 ([17]). *Let $\mathcal{C}$ be a finitely complete category and Σ an object of $\mathcal{C}$. A monomorphism $t : 1 \rightarrowtail \Sigma$ is a* predominance *if every monomorphism $m : U \rightarrowtail X$ has* at most one *characteristic map χ_m in the above sense.*

*A subobject $[m]$ of X (that is the equivalence class of a monomorphism $m : U \rightarrowtail X$) is called Σ-*subset *of X and written $U \subseteq_\Sigma X$ if m arises as a pullback of $1 \rightarrowtail \Sigma$. Let $\mathrm{Sub}_\Sigma(X)$ denote the set of Σ-subsets of X.*

By definition, $\mathrm{Sub}_\Sigma(X)$ is a subclass of $\mathrm{Sub}(X)$, the class of subobjects of X. If $t : 1 \rightarrowtail \Sigma$ is a subobject classifier, we have $\mathrm{Sub}_\Sigma(X) = \mathrm{Sub}(X)$ for every X. One can easily show that a predominance $t : 1 \rightarrowtail \Sigma$ is an isomorphism iff $\mathrm{Sub}_\Sigma(X)$ consists of the equivalence class of isomorphisms. Such a predominance is called *trivial*.

Longley discussed the above notions in $\mathbf{Ass}(A)$ [9]. Suppose that a monomorphism $t : 1 \rightarrowtail \Sigma$ in $\mathbf{Ass}(A)$ is a predominance. Then we can observe that the cardinality of the underlying set $|\Sigma|$ is no more than two. Further if $\mathrm{card}|\Sigma| = 1$, Σ is a terminal object in $\mathbf{Ass}(A)$, hence t is trivial. Thus the non-triviality of t implies that Σ has a doubleton $|\Sigma| = \{\, t, f \,\}$ as the underlying set, so it determines a predominance $(||t||_\Sigma, ||f||_\Sigma)$ on A. Conversely, each predominance (T, F) on A induces a non-trivial predominance $t : 1 \rightarrowtail \Sigma$ with $|\Sigma| := \{\, t, f \,\}, ||t||_\Sigma := T$ and $||f||_\Sigma := F$. To sum up:

Theorem 16 ([9, Subsection 4.2]). *The non-trivial predominances in $\mathbf{Ass}(A)$ are in bijective correspondence with the predominances on A.*

Moreover, every monomorphism $m : U \rightarrowtail X$ that arises as a pullback of $t : 1 \rightarrowtail \Sigma$ is isomorphic to the inclusion $U' \rightarrowtail X$ whose domain is a canonical subassembly defined below.

Definition 17. *Let X be an assembly in $\mathbf{Ass}(A)$. An assembly $U = (|U|, ||\cdot||_U)$ is a* canonical subassembly *of X if $|U| \subseteq |X|$ and $||x||_U = ||x||_X$ for any $x \in |U|$.*

As a convention, we identify each element of $\mathrm{Sub}_\Sigma(X)$ with the associated canonical subassembly of X and Σ-subset relation $U \subseteq_\Sigma X$ with the inclusion $|U| \subseteq |X|$.

Here we give two examples.

Example 18. 1. $\Sigma_\mathrm{d} = (\{\,\mathrm{true}\,\}, \{\,\mathrm{false}\,\})$: In this case, for a Σ_d-subset U of X and its characteristic map $\chi : X \to \Sigma_\mathrm{d}$, we have

$$x \in |U| \iff \chi(x) = t \iff \forall a \in ||x||_X \;\; r_\chi \cdot a = \mathrm{true},$$

where r_χ is a realizer of χ. When $A = \mathcal{K}_1$ and X is the canonical NNO N given above, $|U|$ is nothing but a decidable subset of $\mathbb{N}$. That is, $\mathrm{Sub}_{\Sigma_\mathrm{d}}(N)$ is equal to the set of decidable subsets of $\mathbb{N}$.

2. $\Sigma_\mathrm{sd} = (T_\mathrm{sd}, F_\mathrm{sd})$: Similarly to (1), we obtain

$$x \in |U| \iff \forall a \in ||x||_X \;\; r_\chi a \cdot \mathrm{i}\downarrow.$$

Thus when $A = \mathcal{K}_1$ and X is the canonical NNO, $|U|$ is the domain of a partial computable function $e_U := \lambda^* n.\,(r_\chi n) \cdot \mathrm{i}$. Hence $\mathrm{Sub}_{\Sigma_\mathrm{sd}}(N)$ coincides with the set of semi-decidable subsets of $\mathbb{N}$.

It is obvious that $\subseteq_\Sigma$ is a reflexive, antisymmetric relation on $\mathrm{Sub}_\Sigma(X)$ with the greatest element X and the least element $\emptyset$ (the empty assembly). But $\subseteq_\Sigma$ is not an order in general.

Definition 19 ([7,17]). *A* dominance *on A is a predominance Σ such that $\subseteq_\Sigma$ is transitive.*

Longley gave the following characterization of being a dominance in $\mathbf{Ass}(A)$.

Theorem 20 ([9, Proposition 4.2.7]). *Let $\Sigma = (T, F)$ be a predominance on A. The following are equivalent.*

1. *Σ is a dominance.*
2. *There exists a combinator $r_\mu \in A$ such that*

$$r_\mu(T \times (A \Rightarrow T)) \subseteq T, \quad r_\mu(T \times (A \Rightarrow F)) \subseteq F, \quad r_\mu(F \times A) \subseteq F,$$

where $S_0 \Rightarrow S_1$ denotes $\{\, e \in A \mid \text{whenever } a \in S_0,\ ea \in S_1 \,\}$.

Remark 21. The notion of predominance has been studied in the context of *Synthetic domain theory (SDT)*. It is one of the necessary pieces to construct a subcategory of "abstract domains" in a suitable category $\mathcal{C}$ (such as $\mathbf{Ass}(A)$, $\mathbf{Mod}(A)$). Various axioms for predominance have been investigated by Hyland, Phoa, Taylor and others, and being dominance is the first step towards SDT [7,13,14,18]. In fact, when a predominance t is a dominance, it induces a *lifting monad* $\bot$ on $\mathbf{Ass}(A)$. By using this monad, Longley concretely demonstrated how to construct a model of an extension of PCF. In this process, he showed that the predominance Σ_sd on an arbitrary non-total A is a dominance [9, Example 4.2.9 (ii)].

5 Parallel combinators with respect to Σ and Σ-subsets

In this section, we will make clear the correspondence between the parallel combinators on A considered in Section 3 and the structure of Σ-subsets in Section 4. Interestingly, under a natural assumption on a predominance, our notion of Σ-and and the condition (2) of Theorem 20 correspond perfectly, thus we obtain that if A admits Σ-and then the Σ-subsets form a poset with respect to inclusion. In addition, we show that A admits Σ-or iff the Σ-subsets are closed under union. This is a generalization of the correspondence between parallel-or and union of semi-decidable sets discussed in Section 1.

Lemma 22. *Let $\Sigma = (T, F)$ be a predominance on A. If Σ is a dominance, then A admits Σ-and.*

Proof. By Theorem 20, A has a combinator r_μ that satisfies

$$r_\mu(T \times (A \Rightarrow T)) \subseteq T, \quad r_\mu(T \times (A \Rightarrow F)) \subseteq F, \quad r_\mu(F \times A) \subseteq F.$$

Defining $\mathrm{and}_\Sigma \coloneqq \lambda^* p.\, r_\mu \langle \mathrm{fst}\, p, \mathrm{k}(\mathrm{snd}\, p)\rangle$, we obtain a Σ-and in A.

The converse holds under an additional assumption and we obtain the first characterization theorem:

Definition 23. *Given $a, b \in A$, we write $a \cong b$ if $a \cdot x \cong b \cdot x$ for every $x \in A$. A predominance $\Sigma = (T, F)$ is a called* Rice partition *of A if T is closed under $\cong$ and $F = A \setminus T$.*

Theorem 24. *Let $\Sigma = (T, F)$ be a Rice partition of A. Then A admits Σ-and iff Σ is a dominance iff $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma)$ is a poset for every $X \in \mathbf{Ass}(A)$.*

Proof. We only need to show the forward direction of the first equivalence. Suppose that A admits Σ-and. Letting $l \coloneqq \lambda^* xy.(x \cdot \mathrm{i} \cdot y)$, lb is always defined and $(b\mathrm{i}) \cdot y \cong (lb) \cdot y$ for any $b, y \in A$. Since (T, F) is a Rice partition, we have

$$\begin{cases} b \in (A \Rightarrow T) \implies b\mathrm{i} \in T \implies lb \in T \\ b \in (A \Rightarrow F) \implies b\mathrm{i} \in F \implies lb \in F \\ b \in A \implies lb \in T \cup F \end{cases}$$

for any $b \in A$. We thus have the following implications:

$$\begin{aligned} a \in T \text{ and } b \in (A \Rightarrow T) &\implies a \in T \text{ and } lb \in T \\ &\implies \mathrm{and}_\Sigma \langle a, lb\rangle \in T, \end{aligned}$$

$$\begin{aligned} a \in T \text{ and } b \in (A \Rightarrow F) &\implies a \in T \text{ and } lb \in F \\ &\implies \mathrm{and}_\Sigma \langle a, lb\rangle \in F, \end{aligned}$$

$$\begin{aligned} a \in F \text{ and } b \in A &\implies a \in F \text{ and } (lb \in T \text{ or } lb \in F) \\ &\implies \mathrm{and}_\Sigma \langle a, lb\rangle \in F. \end{aligned}$$

Therefore $r_\mu \coloneqq \lambda^* p.\, \mathrm{and}_\Sigma \langle \mathrm{fst}\, p, l\, (\mathrm{snd}\, p)\rangle$ satisfies condition (2) of Theorem 20.

Notice that if A is non-total, A naturally has a Rice partition, that is, $\Sigma_{\mathrm{sd}} = (T_{\mathrm{sd}}, F_{\mathrm{sd}})$. In conjunction with Theorem 12, we obtain Longley's result that Σ_{sd} is a dominance (See Remark 21).

Now suppose that Σ is a dominance. Then for every object X, $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma)$ is a poset with the least and greatest elements. Moreover, it is automatically equipped with binary meets (intersections).

Definition 25. *Let U and V be canonical subassemblies of X. $U \cap V$ denotes the canonical subassembly of X such that $|U \cap V| := |U| \cap |V|$ and $||x||_{U \cap V} := ||x||_X$ for any $x \in |U| \cap |V|$. Similarly for $U \cup V$.*

It is well-known that the set $\mathrm{Sub}(X)$ of subobjects of X forms a lattice in $\mathbf{Ass}(A)$. On the other hand:

Lemma 26. *If Σ is a dominance, then, for every assembly X, $\mathrm{Sub}_\Sigma(X)$ is closed under intersection $\cap$ and $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma, \cap)$ forms a meet-semilattice.*

Proof. Let U, V be canonical subassemblies of X and $m : U \rightarrowtail X$, $n : V \rightarrowtail X$ the inclusions, respectively. Then $U \cap V$ can be obtained as in the following pullback diagram:

$$\begin{array}{ccc} U \cap V & \rightarrowtail & U \\ {\scriptstyle n^{-1}(m)} \downarrow & \lrcorner & \downarrow {\scriptstyle m} \\ V & \underset{n}{\rightarrowtail} & X. \end{array}$$

If both U and V are Σ-subsets of X, then $U \cap V$ is a Σ-subset of V since $\mathrm{Sub}_\Sigma(X)$ is closed under pullback. Hence $U \cap V$ is a Σ-subset of X. Recalling the structure of the subobject lattice $\mathrm{Sub}(X)$, the binary meet appears as a pullback. Thus $\cap$ behaves as a meet with respect to $\subseteq_\Sigma$.

This means that $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma)$ is a sub-meet-semilattice of $\mathrm{Sub}(X)$ when Σ is a dominance.

Let us finally discuss the effect of having a Σ-or combinator in A. As we have already seen in Section 3, a parallel-or in $\mathcal{K}_1$ has the ability to take the join of two semi-decidable subsets. This fact can be generalized and refined as follows. Notice that the assumption of Rice partition implies that U is a Σ-subset of X iff there exists a characteristic map $\chi_U : X \to \Sigma$ with a realizer r_{χ_U} satisfying

$$x \in |U| \iff \chi_U(x) = t \iff r_{\chi_U}(||x||_X) \subseteq T.$$

The second equivalence is ensured by $T \cap F = \emptyset$. We are now ready to prove the second characterization theorem.

Theorem 27. *Let $\Sigma = (T, F)$ be a predominance with $T \cap F = \emptyset$. Then A admits Σ-or if and only if $\mathrm{Sub}_\Sigma(X)$ is closed under union $\cup$ for every assembly X.*

Proof. We first show the forward direction. Let U, V be Σ-subsets of X, χ_U, χ_V their characteristic maps and r_{χ_U}, r_{χ_V} their realizers, respectively. Then the canonical subassembly $U \cup V$ naturally induces a function $\chi_{U\cup V} : |X| \to |\Sigma|$ such that

$$x \in |U| \cup |V| \iff \chi_{U\cup V}(x) = t.$$

Since A admits Σ-or, we can define $r_{\chi_{U\cup V}}$ as $\lambda^* x.\, \mathrm{or}_\Sigma \langle r_{\chi_U}\, x, r_{\chi_V}\, x\rangle$ in A. Then $r_{\chi_{U\cup V}}$ behaves as follows:

$$\begin{aligned} r_{\chi_{U\cup V}}(||x||_X) \subseteq T &\iff r_{\chi_U}(||x||_X) \subseteq T \text{ or } r_{\chi_V}(||x||_X) \subseteq T \\ &\iff x \in |U| \text{ or } x \in |V| \\ &\iff x \in |U| \cup |V|. \end{aligned}$$

Thus $r_{\chi_{U\cup V}}$ is a realizer of $\chi_{U\cup V}$ and $U \cup V$ is a Σ-subset of X.

To show the backward direction, let us note the following two facts:

– Given two assemblies X and Y, the product $X \times Y$ in $\mathbf{Ass}(A)$ can be concretely described as

$$|X \times Y| \coloneqq |X| \times |Y|, \qquad ||(x,y)||_{X\times Y} \coloneqq ||x||_X \times ||y||_Y.$$

– Every subset S of A induces an assembly $\overline{S}$ such that

$$|\overline{S}| \coloneqq S, \qquad ||a||_{\overline{S}} \coloneqq \{\, a \,\}.$$

For example, there is an assembly $\overline{T \cup F} \times \overline{T \cup F}$ that corresponds to the set $\{\, \langle a, b\rangle \in A \mid a, b \in T \cup F \,\}$.

Let $H \coloneqq T \cup F$. Then we have $\overline{T} \subseteq_\Sigma \overline{H}$ because there is a characteristic map $\chi_T : \overline{H} \to \Sigma$ such that $\chi_T(a) = t$ iff $a \in T$, and it is realized by i. Similarly, one can easily verify the following relations:

$$\overline{T} \times \overline{H} \subseteq_\Sigma \overline{H} \times \overline{H}, \qquad \overline{H} \times \overline{T} \subseteq_\Sigma \overline{H} \times \overline{H}.$$

Lastly, since $\mathrm{Sub}_\Sigma(\overline{H} \times \overline{H})$ is closed under union, we obtain

$$\overline{T} \times \overline{H} \cup \overline{H} \times \overline{T} \subseteq_\Sigma \overline{H} \times \overline{H}.$$

This induces a characteristic map $\chi : \overline{H} \times \overline{H} \to \Sigma$ and a realizer r_χ such that for any $a, b \in T \cup F$,

$$a \in T \text{ or } b \in T \iff \chi((a,b)) = t \iff r_\chi(||(a,b)||_{\overline{H}\times\overline{H}}) \subseteq T.$$

Note that $||(a,b)||_{\overline{H}\times\overline{H}} = \{\, \langle a, b\rangle \,\}$. Hence r_χ satisfies the following property: for any $a, b \in T \cup F$,

– $r_\chi \cdot \langle a, b\rangle$ belongs to T if $a \in T$ or $b \in T$.
– Otherwise, $r_\chi \cdot \langle a, b\rangle$ belongs to F.

Thus r_χ is nothing but a Σ-or combinator.

By restricting to the case of Rice partition, we can summarize the role of Σ-and and Σ-or as follows.

Theorem 28. *Suppose that $\Sigma = (T, F)$ is a Rice partition of A. Then A admits both Σ-and and Σ-or if and only if $(\mathrm{Sub}_\Sigma(X), \subseteq_\Sigma, \cap, \cup)$ forms a lattice for every assembly X.*

Proof. The backward direction is obvious by Theorem 24 and Theorem 27.

For the forward direction, it remains to check that $\cup$ behaves as a join with respect to $\subseteq_\Sigma$. It is sufficient to verify the following claims: if $U, V \subseteq_\Sigma X$ then

$$U \subseteq_\Sigma U \cup V, \qquad U \cup V \subseteq_\Sigma X.$$

The latter is just closure under union, that is already established by Theorem 27. For the former, let $\chi_U : X \to \Sigma$ be the characteristic map of $U \rightarrowtail X$, which exists by $U \subseteq_\Sigma X$. Then $\chi_U|_{|U \cup V|} : U \cup V \to \Sigma$ is the characteristic map of $U \rightarrowtail U \cup V$, which is realized by any realizer of χ_U.

By recalling that a non-total PCA always has Rice partition Σ_{sd} that is a dominance, we finally conclude:

Corollary 29. *Let A be a non-total PCA. Then A admits parallel-or in A if and only if $(\mathrm{Sub}_{\Sigma_{\mathrm{sd}}}(X), \subseteq_{\Sigma_{\mathrm{sd}}}, \cap, \cup)$ forms a lattice for every object X in $\mathbf{Ass}(A)$.*

As we have stated in Proposition 13, Λ^0_v is an example of a non-total PCA that does not admit parallel-or. Therefore, one cannot always take a union of Σ_{sd}-subsets in $\mathbf{Ass}(\Lambda^0_v)$ unlike in $\mathbf{Ass}(\mathcal{K}_1)$.

6 Future work

In this paper we have focused on $\mathbf{Ass}(A)$ among other realizability models. In $\mathbf{Ass}(A)$, (non-trivial) predominances Σ are exactly those that arise from pairs (T, F) of nonempty subsets of A. This simplicity has led to a handy description of Σ-subsets as canonical subassemblies, and consequently a clear correspondence between Σ-and/or combinators and the structure of $\mathrm{Sub}_\Sigma(X)$. All the results in this paper hold for the category $\mathbf{Mod}(A)$ of modest sets over A too, that is a full subcategory of $\mathbf{Ass}(A)$.

On the other hand, the situation is entirely different if we consider the realizability topos $\mathbf{RT}(A)$, that is the exact completion of $\mathbf{Ass}(A)$. The predominances in $\mathbf{RT}(A)$ include the subobject clasifier as well as those associated with a *local operator* j (a.k.a. *Lawvere-Tierney topology*) such as the predominance classifying j-dense subobjects and the one classifying j-closed subobjects. Studying parallel operations in relation to these predominances could be interesting, since local operators in $\mathbf{RT}(A)$ correspond to subtoposes of $\mathbf{RT}(A)$ on one hand, and can be seen as "generalized Turing degrees" on the other [4,6,12]. It is left to future work.

Acknowledgment

I am grateful to my supervisor, Kazushige Terui, and Naohiko Hoshino for many useful discussions. This work is supported by JST Grant Number JPMJFS2123.

References

1. Samson Abramsky and Marina Lenisa. Linear realizability and full completeness for typed lambda-calculi. *Annals of Pure and Applied Logic*, 134(2-3):122–168, 2005.
2. Henk P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1984.
3. Aurelio Carboni, Peter J. Freyd, and Andre Scedrov. A categorical approach to realizability and polymorphic types. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Language Semantics*, pages 23–42, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.
4. Eric Faber and Jaap van Oosten. More on geometric morphisms between realizability toposes. *Theory and Applications of Categories*, 29:874–895, 2014.
5. Giulio Guerrieri. Head reduction and normalization in a call-by-value lambda-calculus. In Yuki Chiba, Santiago Escobar, Naoki Nishida, David Sabel, and Manfred Schmidt-Schauß, editors, *2nd International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE 2015)*, volume 46 of *OpenAccess Series in Informatics (OASIcs)*, pages 3–17, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
6. J. M. E. Hyland. The effective topos. In *The L. E. J. Brouwer Centenary Symposium*, volume 110 of Stud. Logic Foundations Math. North-Holland, pages 165–216, 1982.
7. J. M. E. Hyland. First steps in synthetic domain theory. In *Category Theory*, LNM, volume 1488, pages 131–156, 1991.
8. Stephen C. Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10:109–124, 1945.
9. John R. Longley. *Realizability Toposes and Language Semantics*. PhD thesis, University of Edinburgh, 1994.
10. John R. Longley and Alex K. Simpson. A uniform approach to domain theory in realizability models. *Mathematical Structures in Computer Science*, 7(5):469–505, 1997.
11. Philip S. Mulry. Generalized banach-mazur functionals in the topos of recursive sets. *Journal of Pure and Applied Algebra*, 26(1):71–83, 1982.
12. Wesley Phoa. Relative computability in the effective topos. *Mathematical Proceedings of the Cambridge Philosophical Society*, 106:419–422, 1989.
13. Wesley Phoa. *Domain Theory in Realizability Toposes*. PhD thesis, University of Cambridge, 1990.
14. Wesley Phoa. From term models to domains. *Information and Computation*, 109(1):211–255, 1994.
15. Gordon D. Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.
16. Gordon D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, 1977.
17. Giuseppe Rosolini. *Continuity and Effectiveness in Topoi*. PhD thesis, University of Oxford, 1986.
18. P. Taylor. The fixed point property in synthetic domain theory. In *Proceedings Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 152–160, 1991.
19. Jaap van Oosten. *Realizability: an introduction to its categorical side*, volume 152 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2008.