INVESTIGATIONS ON RELATIVE DEFINABILITY IN PCF

by
Riccardo R. Pucella

School of Computer Science
McGill University
Montréal, Québec
Canada

June 2005

A dissertation
submitted to the Faculty of Graduate Studies and Research
of McGill University
in partial fulfillment of the requirements for
the Degree of Master of Science

Abstract

The focus of this thesis is the study of relative definability of first-order boolean functions with respect to the language PCF, a paradigmatic typed, higher-order language based on the simply-typed λ-calculus. The basic core language is sequential. We study the effect of adding construct that embody various notions of parallel execution. The resulting set of equivalence classes with respect to relative definability forms a sup-semilattice analoguous to the lattice of degrees in recursion theory. Recent results of Bucciarelli show that the lattice of degrees of parallelism has both infinite chains and infinite antichains. By considering a very simple subset of Sieber's sequentiality relations, we identify levels in the lattice and derive inexpressiblity results concerning functions on different levels. This allows us to explore further the structure of the lattice of degrees of parallelism and show the existence of new infinite hierarchies. We also identify four subsemilattices of this structure, all characterized by a simple property.

Résumé

Dans ce mémoire nous nous concentrons sur l'étude de la definition relative de fonctions booléennes de premier order à partir du langage PCF, un language typé d'ordre supérieur paradigmatique, basé sur le λ-calcul simplement typé. Le langage de base est fondamentalement séquentiel. Nous étudions les conséquences de l'ajout d'éléments implémentant différent degrés d'exécution parallèle. L'ensemble résultant de classes d'équivalence de fonctions forme une semilattice supérieure analogue à la lattice de degrés en théorie de la récursion. Des résultats récents de Bucciarelli montrent que la lattice de degrés de parallélisme possède à la fois des chaines infinies et des antichaines infinies. En considérant un sous-ensemble très simple de relations de séquentialité de Sieber, nous identifions des niveaux de la lattice et dérivons des résultats d'inexpressibilité concernant les fonctions sur différents niveaux. Ceci nous permet d'explorer plus en profondeur la structure de la lattice de degrés de parallélisme et de démontrer l'existence de nouvelles hiérarchies infinies. Nous identifions aussi quatre sous-semilattices de cette structure, toutes caractérisées par une propriété particulièrement simple.

Acknowledgments

I should start by thanking the person most responsible for the heavy mass of paper currently resting in your hands. Prakash, thanks for the coaching. Thanks for starting off on tangents sometimes, going off to explore strange ideas that are never boring. And most of all, thanks for pulling me into CompSci...

I must acknowledge the financial support of FCAR, that made surviving these past two years a manageable experience...

Orm, who slowly taught me the love of Formula 1 racing, thereby wrecking my Sunday mornings and probably messing with my sleep patterns until the day I die — thanks for the psychological support. With Liz, you guys made this last year an enjoyable one.

Saskia, late-night confidant, thanks for the companionship. A few cans of tuna are coming your way.

Pour le reste de ces remerciements, je vais revenir à ma langue natale. Je dois remercier mes parents. Ce furent eux qui m'inculquèrent cette curiosité parfois néfaste, plus souvent qu'autrement salvatrice, qui me permet de survivre dans ce bas-monde. Je ne crois pas que je puisse exprimer toute cette gratitude en une seule fois. Ce n'est pas grave. Cela viendra, une petite partie à la fois.

Et finalement, ma femme, Katia, qui est apparue dans ma vie le temps d'un clin d'oeil, et qui tout doucement s'est faite indispensable. Les mots me manquent pour m'exprimer, et si je dois aller puiser dans la littérature des mots satisfants et dignes d'elle, ce sera au moins Eluard qui s'exprimera pour moi:

Ses rêves en pleine lumière
Font s'évaporer les soleils,
Me font rire, pleurer et rire,
Parler sans avoir rien à dire.[1]

[1] Paul Eluard, *L'Amoureuse*.

Contents

List of Figures

Chapter 1

Introduction

The study of relative definability — which functions one needs to add to a programming language in order to be able to implement a given function — is a fundamental problem in theoretical computer science. The complexity of this problem is in direct proportion to the complexity of the programming language and the class of functions one desires to implement. In this thesis, languages are taken to be sequential, and the functions considered all embody a certain level of parallelism.

This work is the product of a research conducted to investigate such a problem in a particularly simple context — simple yet rich enough to yield interesting results. We study the relative definability of continuous first-order boolean functions with respect to Plotkin's language PCF [Plo77], which is fundamentally a simply-typed λ-calculus with recursion, over the ground types of integers and booleans.

Relative definability defines a partial ordering on continuous boolean functions, and this ordering yields an equivalence relation on continuous boolean functions. The object of our study will be the structure of the resulting partially ordered set of equivalence classes of functions (called degrees of parallelism). Work by Trakhtenbrot [Tra73, Tra75], Sazonov [Saz76] and Bucciarelli and Malacaria [Buc95, BM95] show that the structure of degrees of parallelism is highly non-trivial: the poset forms in fact a sup-semilattice — just like degrees of recursion in recursion theory — and contains a “two-dimensional” hierarchy of functions, both infinite chains and infinite antichains of functions.

This thesis consists mostly of a further exploration of the semilattice of degrees of parallelism. Our main structural results include the following:

- A partition of the class of continuous functions, forming two disjoint subsemilattices, STABLE and UNSTABLE.
- The identification of a third subsemilattice, MONO, orthogonal to the previous two and intersecting them both, containing the hierarchy discovered by Bucciarelli in [Buc95].
- The identification of a new “two-dimensional” hierarchy in STABLE, similar to (yet different from) Bucciarelli’s.
- Minimality results concerning non-sequential continuous functions.
- The identification of a new hierarchy in UNSTABLE.
- The characterization of functions in UNSTABLE in relation to the class of functions in STABLE that they dominate in the definability ordering. We identify a class of functions that can implement all the functions in STABLE, and show that they form a fourth subsemilattice.
- A complete characterization of a known class of functions, subsequential functions, by showing that they are equivalent to functions in the MONO subsemilattice.

In order to prove some of the results referred to above, we introduce a new technique that (among others) partitions degrees of parallelism in such a way that inexpressibility results are immediate between functions in various partitions. The technique is based on a theorem by Sieber [Sie92] relating relative definability of functions and sequentiality relations (cf. sections 2.3.1 and 3.2). By considering only a simple class of sequentiality relations, we define the notion of the p-level of a function, and show the inexpressibility of a function from another via a simple criterion on the p-levels of both functions. The class of sequentiality relations under consideration being fairly weak, the inexpressibility results we get are strong: the functions are fundamentally different.

We also discuss possible directions in which to extend the technique in order to handle the full set of sequentiality relations. This would yield a function-independent characterization of the semilattice of degrees of parallelism, and provide new insights into the complexity and structure of the semilattice.

1.1 Overview of the Thesis

- Chapter 2 presents the basic notions from the theory of domains and continuous functions one needs to follow the work herein. We also introduce the language PCF and define sequentiality relations.

- Chapter 3 defines the concept of relative definability of a function from another function via the language PCF. Relative definability forms a partial ordering on the set of all continuous boolean functions, and we form the poset of the equivalence classes of the induced equivalence relation. We present the known results concerning the structure of this poset.

- Chapter 4 introduces the techniques we will use in this thesis to prove some of our results. We define the concept of p-level, which partitions the class of continuous boolean functions. We present an inexpressibility criterion relating functions in different p-levels. Moreover, we characterize the p-level of functions in various important function classes.

- Chapter 5 presents our results concerning the structure of stable continuous boolean functions. We identify an important partition of stable functions, and exhibit a new "two-dimensional" hierarchy. We also discuss minimality results concerning non-sequential functions.

- Chapter 6 presents our results concerning the structure of unstable continuous boolean functions. We exhibit a new hierarchy of unstable functions. We also start an investigation on the class of stable functions dominated by a given unstable function. We identify a class of stable-dominating functions that can express any stable function, and study its structure.

- Chapter 7 indicates some directions in which we could extend the present work in order to fully characterize the relative definability structure. We note interesting questions that were raised during the course of this research, and give concluding remarks summarizing both the results obtained and the techniques developed in the course of the work.

Chapter 2

Preliminaries

This chapter introduces the background material necessary to follow this thesis. The material presented here is well-known, and mostly covers the theory of domains and continuous functions between domains, as well as the language PCF that we will use.

We assume the reader is familiar with basic mathematical notions like partially ordered sets (posets), λ-calculus and elementary semantics of programming languages.

2.1 Domains

This section presents the necessary basics of the theory of domains and continuous, stable and sequential functions. Most of the material can be found in introductory textbooks on programming language semantics, such as [Gun92].

2.1.1 Partial Orders

Given a poset $(D, \leq)$, and a subset $X \subseteq D$. We say X is *consistent* (or upperbounded) iff there is an element y in D such that $\forall x \in X(x \leq y)$. We say X is *directed* iff it is non-empty and every pair of elements of X has an upperbound in X. A poset D is called a *directed-complete partial order* (dcpo) — or simply a complete partial order (cpo) — iff every directed set has a least upperbound. A poset is *bounded-complete* iff every non-empty consistent set has a least upperbound. A poset D is *pointed* iff it has a least element $\perp_D$ (or simply $\perp$ when the poset is understood from the context) with $\forall x \in X, \perp_D \leq x$.

A *sup-semilattice* is a poset in which every finite subset has a least upperbound. Note that a sup-semilattice must be pointed, since the empty set must itself have a least upperbound.

Given a set A, we can construct a poset $A_\perp$ by adjoining a new element $\perp$ to A, and letting the partial order relation be defined simply by the equations $\forall x \in A(\perp \leq x)$. Such a poset is called a *flat poset* (or flat domain). The flat domain of booleans $\mathcal{B}_\perp$ (or simply $\mathcal{B}$ when the context is clear) and the flat domain of natural numbers $\mathcal{N}_\perp$ (or simply $\mathcal{N}$) are represented in figure 2.1.

Algebraicity

Given a poset D, we say an element $x \in D$ is *isolated* (finite of compact) iff whenever it is dominated by a least upperbound $\bigvee Y$ of some directed set Y, there exist some $y \in X$ that dominates x. A cpo D is *algebraic* iff for every $x \in D$, the set of isolated elements below x is directed, and has x as its least upperbound. A cpo D is ω-*algebraic* iff it is algebraic and the set of isolated elements of D is countable. A poset D has *property (I)* iff every isolated element dominates finitely many isolated elements. A poset is *distributive* iff it is bounded-complete and for every x and consistent pair

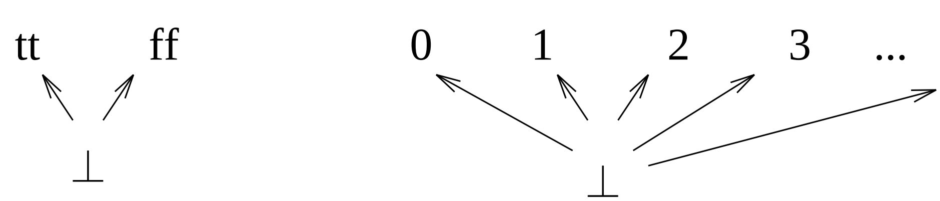

Figure 2.1: The flat domains of booleans and natural numbers

$\{y, z\}$,

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

A *Scott domain* is an ω-algebraic pointed bounded-complete and directed-complete poset. A *dI-domain* (or Berry domain) is a distributive Scott domain with property (I). One can easily check that any flat poset with countably many base elements is a dI-domain.

2.1.2 Continuous Functions

A function $f : D \to E$ between posets is *monotone* iff for every $x, y \in D$, $x \leq_D y$ implies that $f(x) \leq_E f(y)$. A function $f : D \to E$ between cpos is *Scott-continuous* (or simply continuous) iff it is monotone and it preserves directed least upperbounds: for every directed $X \subseteq D$, $f(\bigvee X) = \bigvee \{f(x) : x \in X\}$.

A pointed cpo D has the useful property that any continuous function $f : D \to D$ has a least fixed point. One can verify easily that

$$fix(f) = \bigvee \{f^n(\perp_D) : n \geq 0\}$$

is a fixed point of f and that it is the least such.

Note that a monotone function from a finite poset to an arbitrary poset must be continuous as well. Given two posets D, E, we can define a partial ordering over functions from D to E, called the *extensional ordering*, and that we will denote by $\sqsubseteq_{D,E}$ (dropping the subscripts when the posets are clear from the context), by

$$f \sqsubseteq g \Leftrightarrow \forall x \in D. f(x) \leq g(x)$$

The poset of functions from D to E equipped with the extensional ordering does not generally form a cpo, but if D and E are cpos, and we consider the poset of continuous functions from D to E equipped with the extensional ordering — poset denoted by $[D \to E]$ — we do indeed get a cpo, called the continuous function space cpo from D to E.

2.1.3 Stability

A function $f : D \to E$ between cpos is said to be *stable* [Ber76] iff it is continuous and for all $x \in D$ and all $y \leq f(x)$, there exists a unique $z \leq x$ such that

$$\forall x' \leq x(x' \geq z \Leftrightarrow f(x') \geq y).$$

A function $f : D \to E$ between cpos is said to be *completely multiplicative* iff for any consistent $X \subseteq D$, $f(\bigwedge X) = \bigwedge\{f(x) : x \in X\}$. It is not hard to show that a function f is stable iff it is completely multiplicative.

If the cpos D and E in question are in fact dI-domains, it is sufficient to have $f(x \wedge y) = f(x) \wedge f(y)$ for x and y consistent for f to be stable.

Not every continuous function is stable. A function which is not stable is sometimes called unstable. The textbook example of a continuous yet unstable function is the Parallel OR function (POR), defined as follows:

$$POR(x, y) = \begin{cases} tt & \text{if } x \text{ or } y \text{ is } tt \\ ff & \text{if } x \text{ and } y \text{ are } ff \\ \perp & \text{otherwise} \end{cases}$$

This function is easily seen to be unstable, since

$$POR((tt, \perp) \wedge (\perp, tt)) = POR(\perp, \perp) = \perp$$

but

$$POR(tt, \perp) \wedge POR(\perp, tt) = tt \wedge tt = tt.$$

This functions plays an important role in the remainder of this thesis.

2.1.4 Sequentiality

Sequentiality can be defined straightforwardly when one has a strategy for computing the value of a function — when one views functions as algorithms for computing values. The concept of sequentiality is well understood for first-order functions. On the other hand, defining sequentiality for higher-order functionals is a fundamental problem in semantics (refer for example to [Gev95]). Intuitively, a function f is sequential if it may be computed sequentially, according to some sequential computation strategy. Many definitions of sequentiality exist, all defining what it means for a first-order function to be sequential. Early definitions of sequentiality (M-sequentiality [Mil77] and V-sequentiality [Vui73]) have been given for functions from products of flat posets to a flat poset, and relied heavily on positional information of the arguments involved in subcomputations. Kahn and Plotkin's definition (KP-sequentiality [KP78]) used special structures called concrete domains (which are more general than flat posets) and defined sequential functions between concrete domains. Although Kahn and Plotkin's definition is more generally applicable, it is straightforward to show that for first-order functions whose arguments are taken in flat domains, all three sequentiality definitions are equivalent. In view of this, and because we will be working with flat posets, we shall use Milner's sequentiality definition, which is by far the simplest to state and use.

A function $f : \mathcal{B}^k \to \mathcal{B}$ is said to be *sequential in the sense of Milner* or *M-sequential* if it is constant or if there exists an integer i (called an *index of sequentiality*) with $1 \le i \le k$ such that $x_i = \perp$ implies that $f(x_1, \ldots, x_k) = \perp$ and such that for any fixed value x_i, the function of the remaining arguments is also M-sequential. An M-sequential function will from now on be referred to simply as a sequential function.

It is not too hard to show that any sequential function must be stable (in fact, stability was originally motivated by a desire to generalize the notion of sequentiality). However, not every stable function is sequential. The standard example in the literature is the so-called Gustave function [Ber76], defined as follows, and easily seen to be stable and not sequential:

$$GUST(x,y,z) = \begin{cases} tt & \text{if } x = tt \text{ and } y = \mathit{ff} \\ tt & \text{if } y = tt \text{ and } z = \mathit{ff} \\ tt & \text{if } z = tt \text{ and } x = \mathit{ff} \\ \perp & \text{otherwise} \end{cases}$$

2.2 The PCF Language

In [Plo77], Plotkin introduced PCF, a simply-typed λ-calculus over the ground types of integers and booleans. This section focuses on the semantical description of the language (the syntax is a standard λ-calculus syntax).

The set *Type* of types τ is defined as

$$\tau :: \iota \mid o \mid \tau_1 \rightarrow \tau_2$$

where ι is the integer ground type and o the boolean ground type. Let c range over a collection of constants and v range over a collection of variables. Let M range over λ-terms, defined to be the least collection that contains the constants and the variables, closed under application and abstraction: an application is a term of the form MM' and an abstraction is a term of the form $\lambda v^\tau.M$, for terms M and M', variable v and type τ.

The PCF language has denumerably many variables of each type τ, and the following constants (with their associated type):

$$\begin{array}{rll}
tt: & o & \\
f\!f: & o & \\
\underline{n}: & \iota & \text{for each } n \in \mathcal{N} \\
\supset_\iota: & o \rightarrow \iota \rightarrow \iota \rightarrow \iota & \\
\supset_o: & o \rightarrow o \rightarrow o \rightarrow o & \\
succ, pred: & \iota \rightarrow \iota & \\
\mathrm{Y}_\tau: & (\tau \rightarrow \tau) \rightarrow \tau & \text{for each } \tau \in Type
\end{array}$$

PCF-terms are defined as well-typed λ-terms over these variables and constants, via the standard typing rules. A PCF-program is a closed PCF-term of ground type.

To clarify the presentation of PCF-terms, we use some syntactic notation like **if** M **then** N **else** P **fi** instead of $\supset_\iota MNP$ or $\supset_o MNP$ (the type being clear from the context), a syntactic $\perp$ that can be taken to be the (diverging) term $\mathrm{Y}_\tau(\lambda x^\tau.x^\tau)$, and a syntactic conjunction $\wedge$ and disjunction $\vee$. Moreover, we will often present PCF-terms in uncurried form, as n-ary functions.

2.2.1 Operational Semantics of PCF

The operational semantics of PCF is given by a partial evaluation function taking programs and returning constants. The evaluation function, Eval, is defined using an immediate reduction relation $\rightarrow$ between terms:

$$\mathrm{Eval}(M) = c \iff M \xrightarrow{*} c, \text{ for any program } M \text{ and constant } c$$

where $\xrightarrow{*}$ is the transitive reflexive closure of $\rightarrow$. The $\rightarrow$ reduction relation is just the standard reduction relation of λ-calculus, along with some extra rules to handle the additional constants. Among others,

$$\begin{array}{rcl} \supset_\sigma tt M_\sigma N_\sigma & \rightarrow & M_\sigma \ (\sigma \text{ ground}) \\ \supset_\sigma f\!f M_\sigma N_\sigma & \rightarrow & N_\sigma \ (\sigma \text{ ground}) \\ Y_\sigma M & \rightarrow & M(Y_\sigma M) \end{array}$$

Please refer to [Plo77] for a complete description of the operational semantics.

2.2.2 Denotational Semantics of PCF

The goal of denotational semantics is to provide a realm of abstract values, some of which are denoted by programs. The standard denotational semantics for PCF associates with each type τ a cpo D^τ (this family of type-indexed cpos is called a λ-model). The simplest such model is the *continuous functions model*:

$$\begin{array}{rcll} D^\iota & = & \mathcal{N} & \text{(the flat cpo of natural numbers)} \\ D^o & = & \mathcal{B} & \text{(the flat cpo of truthvalues)} \\ D^{\tau_1 \rightarrow \tau_2} & = & (D^{\tau_1} \rightarrow D^{\tau_2}) & \text{(the cpo of continuous functions)} \end{array}$$

The constants are given their standard interpretation. If c is a constant of type τ, then $[\![c]\!]$ is in D^τ. In particular,

$$[\![tt]\!] = tt$$

$$
\begin{aligned}
[\![\mathit{ff}]\!] &= \mathit{ff} \\
[\![\supset_\sigma]\!](p)(x)(y) &= \begin{cases} x & \text{if } p = \mathit{tt} \\ y & \text{if } p = \mathit{ff} \\ \bot & \text{if } p = \bot \end{cases} \\
& \quad (p \in D^o,\ x, y \in D^\sigma \text{ and } \sigma \text{ a ground type}) \\
[\![\mathrm{Y}_\sigma]\!](f) &= \bigcup_{n \geq 0} f^n(\bot) \qquad (f \in D^{\sigma \to \sigma})
\end{aligned}
$$

2.2.3 The Full Abstraction Problem

We can define the *observable behavior* of a PCF-program P by:

$$beh(P) = \left\{ n : P \xrightarrow{*} \underline{n} \right\}.$$

We define two PCF-terms M, N to be *observationally equivalent*, $M \equiv N$, if for all program contexts $C[\cdot]$ (i.e. with $C[M]$ and $C[N]$ PCF-programs) we have $beh(C[M]) = beh(C[N])$.

A denotational semantics is called *computationally adequate* if semantic equality corresponds to observational equivalence: $[\![M]\!] = [\![N]\!]$ implies $M \equiv N$. This is a trivial result if the denotational semantics is defined by induction on the structure of expressions [Plo77]. A denotational semantics is *fully abstract* if the converse holds, expressing the fact that the denotational semantics does not make unobservable distinctions between any given terms: $M \equiv N$ implies $[\![M]\!] = [\![N]\!]$.

The denotational semantics we gave above for PCF — the standard continuous functions semantics — is not fully abstract. To see that, consider the function POR (section 2.1.3) which is continuous and hence in $\mathcal{B}^{o \to o \to o}$. No function f satisfying the following equations can however be PCF-definable:

$$
\begin{aligned}
f(\mathit{tt}, \bot) &= \mathit{tt} \\
f(\bot, \mathit{tt}) &= \mathit{tt} \\
f(\mathit{ff}, \mathit{ff}) &= \mathit{ff}
\end{aligned}
$$

(a fact we will prove in the next section). To see how full abstraction fails, consider the following term $PORTEST_i$ of type $(o \to o \to o) \to o$, with $i = tt, ff$:

$$\lambda y.\text{if } y(tt, \perp) \wedge y(\perp, tt) \wedge \neg y(ff, ff) \text{ then } i \text{ else } \perp \text{ fi}$$

Then

$$[\![PORTEST_i]\!] f = \begin{cases} i & \text{if } f \text{ satisfies the above equations} \\ \perp & \text{otherwise} \end{cases}$$

This means that $[\![PORTEST_{tt}]\!] \neq [\![PORTEST_{ff}]\!]$ because they differ on the function POR, but we must have $beh(PORTEST_{tt} M) = beh(PORTEST_{ff} M)$ for every closed PCF-term M.

2.3 Logical Relations

Plotkin [Plo77] used an *Activity Lemma* to prove that POR was not PCF-definable. Later in [Plo80], he introduced logical relations to identify non-definable continuous functions.

Definition 2.1 *An n-ary logical relation R on a λ-model $(D^\tau)_{\tau \in Type}$ is a family of relation $R^\tau \subseteq (D^\tau)^n$ such that for all types σ,τ and $f_1, \ldots, f_n \in D^{\sigma \to \tau}$,*

$$R^{\sigma \to \tau}(f_1, \ldots, f_n) \Leftrightarrow \forall d_1, \ldots, d_n . R^\sigma(d_1, \ldots, d_n) \Rightarrow R^\tau(f_1 d_1, \ldots, f_n d_n).$$

Definition 2.2 *An element $d \in D^\tau$ is called invariant under R (or with respect to R) if $R^\tau(d, \ldots, d)$ holds.*

When a function $f : \mathcal{B}^k \to \mathcal{B}$ is presented in uncurried form, the invariance of f under a n-ary logical relation R can be restated as follows: for any tuples

$$\begin{array}{rcl} (x_1^1, \ldots, x_n^1) & \in & R \\ & \vdots & \\ \left(x_1^k, \ldots, x_n^k\right) & \in & R \end{array}$$

one has

$$\big(f(x_1^1,\ldots,x_1^k),\ldots,f(x_n^1,\ldots,x_n^k)\big)\in R$$

Note that a logical relation R is uniquely determined by its definition on the ground types. The following theorem, known as the *Main Lemma for Logical Relations* [Plo80], is the fundamental theorem from which most of the techniques we will study are derived:

Theorem 2.3 *Let R be a logical relation on a λ-model such that $[\![c]\!]$ is invariant under R for every constant c. Then $[\![M]\!]$ is invariant under R for every closed λ-term M over these constants.*

We can now prove that POR is not PCF-definable. Let R be the following 3-ary logical relation:

$$R^o(d_1,d_2,d_3)\Leftrightarrow(d_1=\bot)\vee(d_2=\bot)\vee(d_1=d_2=d_3)$$

It is easy to see that $[\![c]\!]$ is invariant under R for each PCF constant c. Hence, $[\![M]\!]$ is invariant under R for every closed PCF-term M. But we check that there is not closed term M with POR $= [\![M]\!]$, since POR is not invariant under R: $(tt,\bot,ff)$ and $(\bot,tt,ff)$ are in R, but applying POR pointwise to those tuples yields (tt,tt,ff), which is not in R.

2.3.1 Sequentiality Relations

Sieber [Sie92] gave a semantic characterization of all logical relations under which the constants of PCF are invariant.

Definition 2.4 *For each $n\geq 0$ and each pair of sets $A\subseteq B\subseteq\{1,\ldots,n\}$, the presequentiality relation $S_n^{A,B}\subseteq(D^\tau)^n$, $\tau=\iota,o$, is an n-ary logical relation defined by*

$$S_n^{A,B}(d_1,\ldots,d_n)\Leftrightarrow(\exists i\in A.d_i=\bot)\vee(\forall i,j\in B.d_i=d_j)$$

An n-ary logical relation R is called a sequentiality relation if R is an intersection of presequentiality relations.

Note that the logical relation we used to determine that POR was not PCF-definable was a presequentiality relation, namely $S_3^{\{1,2\},\{1,2,3\}}$. The following theorem characterizes sequentiality relations:

Theorem 2.5 *The sequentiality relations are the only logical relations under which the meanings of all constants of PCF are invariant.*

Sequentiality relations, as we shall see in the next chapter, play a very important role in the study of relative definability of first-order functions. In chapter 4, we will focus our attention on presequentiality relations almost exclusively.

Chapter 3

The Poset of Degrees of Parallelism

In this chapter we introduce some notions that will allow us to study first-order boolean functions, and define what it means to say that a function can express another. We also define a partial ordering on functions derived from the expressibility of a function from another. We start our investigation of the structure of the resulting poset and identify and describe a hierarchy of functions in the poset discovered by Bucciarelli showing that the structure of the poset is highly non-trivial.

3.1 First-Order Boolean Continuous Functions

We present in this section various definitions that will help us discuss first-order boolean functions. Before defining concepts, let us first mention some abuse of notation we shall frequently use: given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function and $\underline{x} = (x_1, \ldots, x_k)$, then $f(\underline{x})$ stands for $f(x_1, \ldots, x_k)$, and given $A = \{\underline{x}_1, \ldots, \underline{x}_n\} \subseteq \mathcal{B}^k$, $f(A)$ is defined to be $\{f(\underline{x}_i) : \underline{x}_i \in A\}$. Moreover, π_1 and π_2 are defined respectively as the first and second projection functions associated to a product operation.

The *trace* of a function will be the central notion we will focus on to study boolean functions. The trace of a function is a representation of the minimum input needed for the function to produce a result.

Definition 3.1 *Given a continuous function $f : \mathcal{B}^k \to \mathcal{B}$, define the trace of f to be*

$$tr(f) = \Big\{(v, b) | v \in \mathcal{B}^k, b \in \mathcal{B}, b \neq \bot, f(v) = b \text{ and } \forall v' < v, f(v') = \bot\Big\}$$

With such a presentation, we see that a continuous function $f : \mathcal{B}^k \to \mathcal{B}$ is stable if and only if for all $v_1, v_2 \in \pi_1(tr(f))$, $v_1 \not\uparrow v_2$. Note that the monotonicity of f insures that if $v_1 \uparrow v_2$ then $f(v_1) = f(v_2)$.

Linear coherence is a concept first used to study first-order boolean functions by Bucciarelli and Erhard [BE91, BE94, Buc95].

Definition 3.2 *A subset $A = \{v_1, \ldots, v_k\}$ of $\mathcal{B}^n$ is linearly coherent (or simply coherent) if*

$$\forall j,\ 1 \leq j \leq n \left(\left(\forall l,\ 1 \leq l \leq k.\ v_l^j \neq \bot\right) \Rightarrow \forall l_1, l_2,\ 1 \leq l_1 \leq l_2 \leq k.\ v_{l_1}^j = v_{l_2}^j\right)$$

The set of coherent subsets of $M \subseteq \mathcal{B}^n$ is noted $\mathcal{C}(M)$. Recall that an *Egli-Milner lowerbound* of a set A is a set B such that

$$\forall x \in A \exists y \in B. y \leq x \text{ and } \forall y \in B \exists x \in A. y \leq x.$$

it is not hard to show that if $A \in \mathcal{C}(\mathcal{B}^n)$ and B is an Egli-Milner lowerbound of A, then $B \in \mathcal{C}(\mathcal{B}^n)$.

Closely related to the notion of coherence is the following:

Definition 3.3 *A subset $A = \{v_1, \ldots, v_k\}$ of $\mathcal{B}^n$ is $\bot$-covering if*

$$\forall j\ 1 \leq j \leq n (\exists i\ 1 \leq i \leq k\ v_i^j = \bot)$$

It is easy to see that A being $\bot$-covering implies that A is coherent.

Definition 3.4 *A subset $M \subseteq \mathcal{B}^n$ is completely $\bot$-covering if for any set $A \in \mathcal{C}(M)$, A is $\bot$-covering.*

Abusing the terminology, we will sometimes say that a continuous function $f : \mathcal{B}^k \to \mathcal{B}$ is $\perp$-covering or completely $\perp$-covering if $\pi_1(tr(f))$ has the corresponding property.

Finally, we will also require the notion of a monovalued function:

Definition 3.5 *A function $f : \mathcal{B}^k \to \mathcal{B}$ is monovalued if $|\pi_2(tr(f))| = 1$.*

By another abuse of terminology, we will say that a subset $A \subseteq \pi_1(tr(f))$ is monovalued if $|f(A)| = 1$. When a function f (or a subset of $\pi_1(tr(f))$) is not monovalued, it is said to be bivalued.

Our final remarks about first-order boolean functions concern two operations that will prove useful in later sections. Given a continuous boolean function $f : \mathcal{B}^k \to \mathcal{B}$, we define the function $neg(f) : \mathcal{B}^k \to \mathcal{B}$ to be the function returning *tt* when f returns *ff* and returning *ff* when f returns *tt*. As for the second operation, given two continuous boolean functions $f : \mathcal{B}^k \to \mathcal{B}$ and $g : \mathcal{B}^{k'} \to \mathcal{B}$, we will define a function $f + g : \mathcal{B}^{max(k,k')+1} \to \mathcal{B}$. Without loss of generality, we assume that there exists a $l \geq 0$ such that $k' = k - l$. Let the function $f + g$ be defined via the following trace:

$$\begin{aligned} tr(f+g) &= \{((tt, x_1, \ldots, x_k), b) : ((x_1, \ldots, x_k), b) \in tr(f)\} \bigcup \\ & \quad \{((\underbrace{ff, \ldots, ff}_{l+1}, x_1, \ldots, x_{k'}), b) : ((x_1, \ldots, x_{k'}), b) \in tr(g)\} \end{aligned}$$

3.2 Relative Definability

Relative definability refers to the ability of defining some function using another function. The intuition behind this statement is that a function can define another function if there exist some algorithm in some language that uses the former to compute the latter. The language we will use to specify relative definability will be the language PCF, introduced in 2.2. Formally,

Definition 3.6 *Given two continuous functions f and g, we say that f is PCF-expressible by g ($f \preceq g$) if there exists a PCF-term M such that $f = [\![M]\!] g$.*

An equivalent terminology one finds in the literature is to say that "f is less parallel than g" for $f \preceq g$, or that f is g-expressible. The $\preceq$ preorder induces an equivalence relation $\equiv$ on continuous function such that $f \equiv g$ iff $f \preceq g$ and $g \preceq f$. An equivalence class of the $\equiv$ equivalence relation is called a *degree of parallelism*, and two functions f and g with $f \equiv g$ are called *equiparallel*. The degree of parallelism of a given continuous function f is denoted $[f]$.

We note that given any function f, $neg(f) \equiv f$. Moreover, given functions $f \equiv f'$ and $g \equiv g'$, then we must have $f + g \equiv f' + g'$ ($f \preceq f' \preceq f' + g'$ and $g \preceq g' \preceq f' + g'$ implies $f + g \preceq f' + g'$, and symmetrically for the other direction).

This thesis is solely concerned with PCF-expressibility of first-order boolean functions, which is fully characterized by Sieber's sequentiality relations presented in section 2.3.1. The following theorem is fundamental and gives the full characterization of the $\preceq$ preorder.

Theorem 3.7 *For any $f : \mathcal{B}^n \to \mathcal{B}$ and $g : \mathcal{B}^k \to \mathcal{B}$ continuous functions, $f \preceq g$ if and only if for any sequentiality relation R, if g is invariant under R then f is also invariant under R.*

It is interesting to note that this characterization is effective and Stoughton [Sto94] implemented an algorithm that decides $f \preceq g$ given the functions f and g.

3.3 The Structure of Degrees of Parallelism

We are interested in studying the structure of first-order degrees of parallelism. Trakhtenbrot [Tra73, Tra75] and Sazonov [Saz76] first investigated the subject and pointed out finite subposets of degrees (though not necessarily first-order degrees). Some facts are further consequences of some well-known results: the poset of degrees of parallelism must have a top element, by Plotkin's full abstraction result for PCF+POR [Plo77]. Moreover, it is easy to check that any sequential function must be directly PCF-definable.

We can easily show that first-order degrees of parallelism form a sup-semilattice:

Proposition 3.8 *The poset of first-order degrees of parallelism is a sup-semilattice with a bottom element (the set of PCF-definable functions) and a top element (the degree of POR).*

Proof: The set of PCF-definable (sequential) functions is the bottom element of degrees by definition ,whereas [POR] is the top element of degrees by Plotkin's definability result [Plo77]. We now show that given $f : \mathcal{B}^k \to \mathcal{B}$ and $g : \mathcal{B}^{k'} \to \mathcal{B}$, $[f+g] = [f] \vee [g]$ (we know $[f+g]$ is well-defined, by an earlier argument). We need to first show that $f \preceq f+g$ and $g \preceq f+g$ (as before, assume that there exists a $l \geq 0$ such that $k' = k - l$). We notice that

$$f = [\![\lambda h \lambda x_1 \ldots x_k . h(tt, x_1, \ldots, x_k)]\!](f+g)$$

and

$$g = \left[\!\!\left[\lambda h \lambda x_1 \ldots x_{k'} . h(\underbrace{ff, \ldots, ff}_{l+1}, x_1, \ldots, x_{k'}) \right]\!\!\right](f+g)$$

Moreover, let $h : \mathcal{B}^l \to \mathcal{B}$ be such that $f, g \preceq h$. Then there exist M, N with $f = [\![M]\!] h$ and $g = [\![N]\!] h$. It is easy to see that

$$\begin{aligned} F &= \lambda p \lambda x_1 \ldots x_{k+1} . \text{if } x_1 \text{ then } M(p, x_2, \ldots, x_{k+1}) \text{ else} \\ &\quad\; \text{if } x_2 \vee \cdots \vee x_{l+1} \text{ then } \perp \text{ else } N(p, x_{l+2}, \ldots, x_{k+1}) \\ f+g &= [\![F]\!] h \end{aligned}$$

Hence, $[f+g] = [f] \vee [g]$. □

We will denote the semilattice of first-order degrees of parallelism by CONT (for first-order continuous functions).

Let us mention for the sake of completeness a first-order function that both Trakhtenbrot and Sazonov presented in their finite subposet of degrees — Trakhtenbrot's voting function V introduced in [Tra73] and defined by

$$\mathrm{V}(x_1, x_2, x_3) = \begin{cases} x_i & \text{if } x_i = x_j \text{ for } i \neq j \\ \perp & \text{if } \forall i \neq j,\, x_i \neq x_j \end{cases}$$

or equivalently by the following trace (presented in matrix form):

$$\begin{array}{|ccc|c|} \hline tt & tt & \perp & tt \\ tt & \perp & tt & tt \\ \perp & tt & tt & tt \\ ff & ff & \perp & ff \\ ff & \perp & ff & ff \\ \perp & ff & ff & ff \\ \hline \end{array}$$

It is not hard to show that POR and V are equiparallel:

Proposition 3.9 *V≡POR.*

Proof: V$\preceq$POR is immediate since POR is maximal. To show POR$\preceq$V, we note that

$$\mathrm{POR} = [\![\lambda f.\lambda x_1 x_2.f(x_1, x_2, tt)]\!] \mathrm{V}$$

□

3.4 Hypergraphs

In [Buc95, BM95] an attempt was made at finding a categorical counterpart to the semilattice of degrees. Without going into details, the goal was to define a category whose objects would correspond to first-order boolean functions, and a morphism between objects F,G of the category would exist iff the function corresponding to F were less parallel than the function corresponding to G. The selected category was the category of so-called hypergraphs. We will not describe here what hypergraphs are — since we will not be using them — we will only say that the hypergraph of a function records the number of elements in the trace of f as well as keeping track of the return value of each trace element and the linearly coherent subsets of the first projection of the trace, while forgetting about the arity of f and the actual individual arguments making up the trace of f. Morphisms are defined as suitable maps between hypergraphs.

However, this category does not exactly reflect the structure of the degrees of parallelism semilattice. In fact, the best that was achieved was the following theorem, proved in [BM95]:

Theorem 3.10 *Let f,g be two continuous functions, and H_f,H_g be the hypergraphs corresponding to f and g. Then $HOM(H_f, H_g) \neq \emptyset \Rightarrow f \preceq g$.*

In order to use this result, we will recast it in a non-hypergraph form — since we will not be using the hypergraph terminology in this thesis.

Corollary 3.11 *Let f, g be two continuous functions. If there exists a function $\alpha : tr(f) \to tr(g)$ such that*

1. *for all $A \subseteq tr(f)$, if $\pi_1(A)$ is non-singleton and linearly coherent, then $\pi_1(\alpha(A))$ is non-singleton and linearly coherent.*
2. *for all $A \subseteq tr(f)$ with $\pi_1(tr(f))$ non-singleton and linearly coherent, and for all $x, y \in A$, we have $\pi_2(x) \neq \pi_2(y) \Rightarrow \pi_2(\alpha(x)) \neq \pi_2(\alpha(y))$.*

then $f \preceq g$.

Proof: This corollary is simply a translation of theorem 3.10, with an explicit description of what hypergraphs and morphisms look like. $\square$

3.5 The Bucciarelli Hierarchy

In [Buc95], Bucciarelli exhibited a non-trivial hierarchy of functions in the CONT semilattice, thereby showing its highly non-trivial structure.

We define the functions $\mathrm{BUCC}_{(n,m)}$ via the following description. The trace of $\mathrm{BUCC}_{(n,m)}$ has m elements and each trace element returns *tt*. Moreover, for any subset of less than n elements (and at least two) of the first projection of the trace, there must exist a column which makes that subset incoherent. The arity of $\mathrm{BUCC}_{(n,m)}$ is $\sum_{i=2}^{n-1} \begin{pmatrix} m \\ i \end{pmatrix}$, and in the $\mathrm{j^{th}}$ column only elements corresponding to rows in the $\mathrm{j^{th}}$ subset (with respect to an arbitrary enumeration) will be defined: by *tt* for the first row in that subset and by *ff* for the other rows in the subset.

For example, the following matrix represents the trace of $\mathrm{BUCC}_{(3,4)}$:

tt	*tt*	*tt*	$\perp$	$\perp$	$\perp$	*tt*
ff	$\perp$	$\perp$	*tt*	*tt*	$\perp$	*tt*
$\perp$	*ff*	$\perp$	*ff*	$\perp$	*tt*	*tt*
$\perp$	$\perp$	*ff*	$\perp$	*ff*	*ff*	*tt*

and this matrix represents the trace of $\mathrm{BUCC}_{(4,4)}$:

tt	*tt*	*tt*	$\perp$	$\perp$	$\perp$	*tt*	*tt*	*tt*	$\perp$	*tt*
ff	$\perp$	$\perp$	*tt*	*tt*	$\perp$	*ff*	*ff*	$\perp$	*tt*	*tt*
$\perp$	*ff*	$\perp$	*ff*	$\perp$	*tt*	*ff*	$\perp$	*ff*	*ff*	*tt*
$\perp$	$\perp$	*ff*	$\perp$	*ff*	*ff*	$\perp$	*ff*	*ff*	*ff*	*tt*

If we define the numerical predicate $C^{m,n',m'}$ by

$$\begin{aligned} C^{m,n',m'} &= \min(n'-1, m \bmod m') * \left\lceil \frac{m}{m'} \right\rceil + \\ &\quad \max(0, (n'-1) - (m \bmod m')) * \left\lfloor \frac{m}{m'} \right\rfloor \end{aligned}$$

we get the following results, whose proof can be found in [Buc95]:

Proposition 3.12 *If* n, m, n', m' *are integers such that* $3 \leq n \leq m$, $3 \leq n' \leq m'$ *and* $n > C^{m,n',m'}$*,then* $BUCC_{(n,m)} \preceq BUCC_{(n',m')}$.

Proposition 3.13 *If* n, m, n', m' *are integers such that* $3 \leq n \leq m$, $3 \leq n' \leq m'$ *and* $n \leq C^{m,n',m'}$*,then* $BUCC_{(n,m)} \not\preceq BUCC_{(n',m')}$.

Computing various values of $C^{i,j,k}$ as in [Buc95], we end up with a hierarchy of functions as pictured in figure 3.1, where the directed edges represent strict relative definability results.

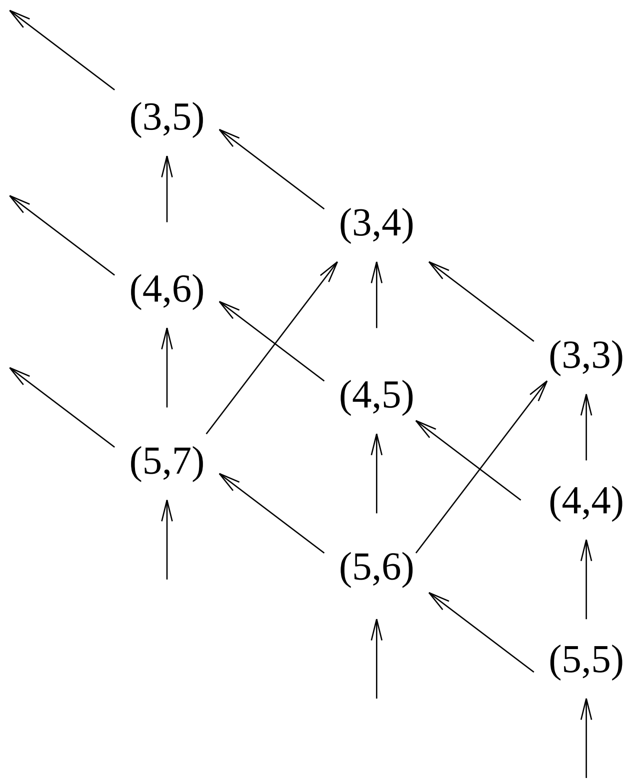

Figure 3.1: The Bucciarelli hierarchy

Chapter 4

Presequentiality Relations

In this chapter, we develop the basic techniques we will use to prove most of the results of the next chapters. Our focus will be on the presequentiality relations of section 2.3.1. We know, by theorem 3.7, that we can show that a function f is not g-expressible if we can exhibit a sequentiality relation R such that g is invariant under R but f is not. In some cases, the sequentiality relation R used to differentiate f and g is a presequentiality relation of the form $S_n^{A,B}$, and hence the functions f and g are differentiated using a very weak form of sequentiality relation. Our goal for this chapter is to provide the framework in which we will discuss *p-levels*, a numerical characteristic of functions partitioning the poset of degrees of parallelism into classes of functions invariant under the same presequentiality relations. Inexpressibility of functions with different p-levels can then be determined via a very simple criterion. We will conclude with characterizations in terms of p-levels of the important classes of functions we will be considering.

4.1 Definitions and Fundamental Lemmas

The attractiveness of presequentiality relations stems from the fact that they are much simpler to use (albeit less discriminating) than the general sequentiality relations. In fact, the following two lemmas show that it is not necessary to consider every presequentiality relations when determining which presequentiality relation a function is invariant under. The Reduction Lemma (4.1) tells us that is it sufficient to look at presequentiality relations of the form $S_m^{\{1,\ldots,m\},\{1,\ldots,m\}}$ and $S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$ (as opposed to the general $S_n^{A,B}$ with $A \subseteq B \subseteq \{1,\ldots,n\}$ form). The Closure Lemma (4.2) says that if a function is invariant under a presequentiality relation $S_n^{A,B}$, invariance holds under any presequentiality relation with "smaller" A and B.

Lemma 4.1 (Reduction Lemma) *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function and $A \subseteq B \subseteq \{1,\ldots,n\}$, one of the following holds:*

1. *$(A = B)$ f is invariant under $S_n^{A,A}$ $\Leftrightarrow$ f is invariant under $S_{|A|}^{\{1,\ldots,|A|\},\{1,\ldots,|A|\}}$*
2. *$(A \subset B)$ f is invariant under $S_n^{A,B}$ $\Leftrightarrow$ f is invariant under $S_{|A|+1}^{\{1,\ldots,|A|\},\{1,\ldots,|A|+1\}}$.*

The proof of this lemma can be found in Appendix A.

Lemma 4.2 (Closure Lemma) *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function and m any integer, the following holds:*

1. *f invariant under $S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$ $\Rightarrow$ f invariant under $S_m^{\{1,\ldots,m\},\{1,\ldots,m\}}$.*
2. *f invariant under $S_{m+1}^{\{1,\ldots,m+1\},\{1,\ldots,m+1\}}$ $\Rightarrow$ f invariant under $S_m^{\{1,\ldots,m\},\{1,\ldots,m\}}$*
3. *f invariant under $S_{m+2}^{\{1,\ldots,m+1\},\{1,\ldots,m+2\}}$ $\Rightarrow$ f invariant under $S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$*

Proof: (1) Immediate via lemma A.6.

(2) Given the tuples

$$\begin{array}{rcl} (x_1^1,\ldots,x_m^1) & \in & S_m^{\{1,\ldots,m\},\{1,\ldots,m\}} \\ & \vdots & \\ \left(x_1^k,\ldots,x_m^k\right) & \in & S_m^{\{1,\ldots,m\},\{1,\ldots,m\}} \end{array}$$

we show $(y_1, \dots, y_m) \in S_m^{\{1,\dots,m\},\{1,\dots,m\}}$ with $y_i = f(x_i^1, \dots, x_i^k)$.

Consider the tuples

$$\begin{aligned} \left(x_1^1, \dots, x_m^1, x_1^1\right) &\in S_{m+1}^{\{1,\dots,m+1\},\{1,\dots,m+1\}} \\ &\vdots \\ \left(x_1^k, \dots, x_m^k, x_1^k\right) &\in S_{m+1}^{\{1,\dots,m+1\},\{1,\dots,m+1\}} \end{aligned}$$

By invariance of f under $S_{m+1}^{\{1,\dots,m+1\},\{1,\dots,m+1\}}$, we have

$$(y_1, \dots, y_m, y_1) \in S_{m+1}^{\{1,\dots,m+1\},\{1,\dots,m+1\}}$$

which means that either $\exists i \leq m$ s.t. $y_i = \perp$ or $\forall i, j \leq m$, $y_i = y_j$. Hence

$$(y_1, \dots, y_m) \in S_m^{\{1,\dots,m\},\{1,\dots,m\}}.$$

(3) Given the tuples

$$\begin{aligned} \left(x_1^1, \dots, x_{m+1}^1\right) &\in S_{m+1}^{\{1,\dots,m\},\{1,\dots,m+1\}} \\ &\vdots \\ \left(x_1^k, \dots, x_{m+1}^k\right) &\in S_{m+1}^{\{1,\dots,m\},\{1,\dots,m+1\}} \end{aligned}$$

we show $(y_1, \dots, y_{m+1}) \in S_{m+1}^{\{1,\dots,m\},\{1,\dots,m+1\}}$, with $y_i = f(x_i^1, \dots, x_i^k)$.

Consider the tuples

$$\begin{aligned} \left(x_1^1, \dots, x_m^1, x_1^1, x_{m+1}^1\right) &\in S_{m+2}^{\{1,\dots,m+1\},\{1,\dots,m+2\}} \\ &\vdots \\ \left(x_1^k, \dots, x_m^k, x_1^k, x_{m+1}^k\right) &\in S_{m+2}^{\{1,\dots,m+1\},\{1,\dots,m+2\}} \end{aligned}$$

By invariance of f under $S_{m+2}^{\{1,\dots,m+1\},\{1,\dots,m+2\}}$, we have

$$(y_1, \dots, y_m, y_1, y_{m+1}) \in S_{m+2}^{\{1,\dots,m+1\},\{1,\dots,m+2\}}$$

which means either that $\exists i \leq m$ s.t. $y_i = \perp$ or $\forall i, j \leq m+1$, $y_i = y_j$. Hence,

$$(y_1, \dots, y_{m+1}) \in S_{m+1}^{\{1,\dots,m\},\{1,\dots,m+1\}}.$$

$\square$

These two fundamental lemmas allow us to make the following definition:

Definition 4.3 *A continuous function $f : \mathcal{B}^k \to \mathcal{B}$ is said to have a* p-level *of (i, j) if f is invariant under $S_i^{\{1,\ldots,i\},\{1,\ldots,i\}}$ and $S_{j+1}^{\{1,\ldots,j\},\{1,\ldots,j+1\}}$ but not under $S_{i+1}^{\{1,\ldots,i+1\},\{1,\ldots,i+1\}}$ and $S_{j+2}^{\{1,\ldots,j+1\},\{1,\ldots,j+2\}}$.*

Not every pair of natural numbers (i, j) can meaningfully be said to be the p-level of a function. By the Closure Lemma, the p-level (i, j) of a function f must be such that $i \geq j$. We will use the notation (∞, j) (resp. (∞, ∞)) if f is invariant under every presequentiality relation $S_i^{\{1,\ldots,i\},\{1,\ldots,i\}}$ (resp. $S_{i+1}^{\{1,\ldots,i\},\{1,\ldots,i+1\}}$).

By the Reduction and the Closure Lemma, a function f with a p-level of (i, j) is invariant under a presequentiality relation $S_n^{A,B}$ iff either

1. $|A| = |B| \leq i$
2. $|A| < |B|$, and $|A| \leq j$.

The p-level of the least upperbound of two functions has a direct relationship to the p-level of the given functions:

Proposition 4.4 *Given $f : \mathcal{B}^k \to \mathcal{B}$, $g : \mathcal{B}^{k'} \to \mathcal{B}$ two continuous functions with p-levels of (i_f, j_f) and (i_g, j_g) respectively. Then the p-level of $f + g$ is*

$$(\min(i_f, i_g), \min(j_f, j_g))$$

.

Proof: Let $i_m = \min(i_f, i_g)$ and $j_m = \min(j_f, j_g)$. As before, we assume without loss of generality that there exists $l \leq 0$ with $k' = k - l$.

By the Closure Lemma, it is sufficient to show that $f + g$ is invariant under $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$ and $S_{j_m+1}^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}$ but not $S_{i_m+1}^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}$ and $S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}}$.

We first show $f+g$ is invariant under $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$ and $S_{j_{m+1}}^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}$. We know f and g are invariant under $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$ and $S_{j_{m+1}}^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}$ (by assumption on the p-levels of f and g). Given tuples

$$\begin{array}{rcl}\left(x_1^1,\ldots,x_{i_m}^1\right) & \in & S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}} \\ & \vdots & \\ \left(x_1^{k+1},\ldots,x_{i_m}^{k+1}\right) & \in & S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}\end{array}$$

we want to show that $(y_1,\ldots,y_{i_m}) \in S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$ for $y_i = (f+g)(x_i^1,\ldots,x_i^{k+1})$. Three cases occur:

1. $(x_1^1,\ldots,x_{i_m}^1)$ is all *tt*, then

   $$y_i = (f+g)(tt, x_i^2,\ldots,x_i^{k+1})$$

   which is just $f(x_i^2,\ldots,x_i^{k+1})$ and since f is invariant under $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$, we get that $(y_1,\ldots,y_{i_m})$ is in $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$.
2. $(x_1^1,\ldots,x_{i_m}^1),\ldots,(x_1^{l+1},\ldots,x_{i_m}^{l+1})$ are all *ff*. then

   $$y_i = (f+g)(ff,\ldots,ff, x_i^{l+2},\ldots,x^{k+1})$$

   which is just $g(x_i^{l+2}\ \ldots,x_i^{k+1})$ and since g is invariant under $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$, we get that $(y_1,\ldots,y_{i_m})$ is in $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$.
3. $(x_1^1,\ldots,x_{i_m}^1)$ has a $\perp$, or it is all *ff* and $(x_1^2,\ldots,x_{i_m}^2),\ldots,(x_1^{l+1},\ldots,x_{i_m}^{l+1})$ are not all *ff*, then one of the y_i must be $\perp$, and $(y_1,\ldots,y_{i_m})$ is in $S_{i_m}^{\{1,\ldots,i_m\},\{1,\ldots,i_m\}}$.

Given tuples

$$\begin{array}{rcl}\left(x_1^1,\ldots,x_{j_m+1}^1\right) & \in & S_{j_m+1}^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}} \\ & \vdots & \\ \left(x_1^{k+1},\ldots,x_{j_m+1}^{k+1}\right) & \in & S_{j_m+1}^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}\end{array}$$

we want to show that $(y_1,\ldots,y_{j_m+1}) \in S_{j_m+1}^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}$ for $y_i = (f+g)(x_i^1,\ldots,x_i^{k+1})$. Three cases occur:

1. $(x^1_1, \ldots, x^1_{j_m+1})$ is all tt, then

$$y_i = (f+g)(tt, x^2_i, \ldots, x^{k+1}_i)$$

which is just $f(x^2_i, \ldots, x^{k+1}_i)$ and since f is invariant under $S^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}_{j_m+1}$, we get that $(y_1, \ldots, y_{j_m+1})$ is in $S^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}_{j_m+1}$.

2. $(x^1_1, \ldots, x^1_{j_m+1}), \ldots, (x^{l+1}_1, \ldots, x^{l+1}_{j_m+1})$ are all *ff*. then

$$y_i = (f+g)(ff, \ldots, ff, x^{l+2}_i, \ldots, x^{k+1})$$

which is just $g(x^{l+2}_i \ \ldots, x^{k+1}_i)$ and since g is invariant under $S^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}_{j_m+1}$, we get that $(y_1, \ldots, y_{j_m+1})$ is in $S^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}_{j_m+1}$.

3. $(x^1_1, \ldots, x^1_{j_m+1})$ has a $\perp$, or it is all *ff* and $(x^2_1, \ldots, x^2_{j_m+1}), \ldots, (x^{l+1}_1, \ldots, x^{l+1}_{j_m+1})$ are not all *ff*, then one of the y_i must be $\perp$, and $(y_1, \ldots, y_{j_m+1})$ is in $S^{\{1,\ldots,j_m\},\{1,\ldots,j_m+1\}}_{j_m+1}$.

We now show that $f+g$ cannot be invariant under $S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1}$. Without loss of generality, let us assume that $i_m = i_f$ (the case $i_m = i_g$ is similar). Then f is not invariant under $S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1}$ (by assumption on the p-level of f). In other words, there exist tuples

$$\begin{array}{rcl}\left(x^1_1, \ldots, x^1_{i_m+1}\right) & \in & S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1} \\ & \vdots & \\ \left(x^k_1, \ldots, x^k_{i_m+1}\right) & \in & S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1}\end{array}$$

with $(y_1, \ldots, y_{i_m+1}) \notin S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1}$, $y_i = f(x^1_i, \ldots, x^k_i)$. Consider the following tuples:

$$\begin{array}{rcl}(tt, \ldots, tt) & \in & S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1} \\ \left(x^1_1, \ldots, x^1_{i_m+1}\right) & \in & S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1} \\ & \vdots & \\ \left(x^k_1, \ldots, x^k_{i_m+1}\right) & \in & S^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}_{i_m+1}\end{array}$$

and $\Big((f+g)(tt, x_1^1, \ldots, x_1^k), \ldots, (f+g)(tt, x_{i_m+1}^1, \ldots, x_{i_m+1}^k)\Big)$ is simply $(y_1, \ldots, y_{i_m+1})$, which is not in $S_{i_m+1}^{\{1,\ldots,i_m+1\},\{1,\ldots,i_m+1\}}$.

We now show that $f+g$ is not invariant under $S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}}$. Without loss of generality, let us again assume that $j_m = j_f$ (the case $j_m = j_g$ is similar). Then f is not invariant under $S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}}$ (by assumption on the p-level of f). In other words, there exist tuples

$$\begin{array}{rcl} \Big(x_1^1, \ldots, x_{j_m+2}^1\Big) & \in & S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}} \\ & \vdots & \\ \Big(x_1^k, \ldots, x_{j_m+2}^k\Big) & \in & S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}} \end{array}$$

with $(y_1, \ldots, y_{j_m+2}) \notin S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}}$, $y_i = f(x_i^1, \ldots, x_i^k)$. Consider the following tuples:

$$\begin{array}{rcl} (tt, \ldots, tt) & \in & S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}} \\ \Big(x_1^1, \ldots, x_{j_m+2}^1\Big) & \in & S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}} \\ & \vdots & \\ \Big(x_1^k, \ldots, x_{j_m+2}^k\Big) & \in & S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}} \end{array}$$

and $\Big((f+g)(tt, x_1^1, \ldots, x_1^k), \ldots, (f+g)(tt, x_{j_m+2}^1, \ldots, x_{j_m+2}^k)\Big)$ is simply $(y_1, \ldots, y_{j_m+2})$, which is not in $S_{j_m+2}^{\{1,\ldots,j_m+1\},\{1,\ldots,j_m+2\}}$. □

Since every function in a degree of parallelism must be invariant under the same presequentiality relations, it makes sense to talk about the p-level of a degree of parallelism.

Theorem 3.7 allows us to derive the following non-expressiblity result

Proposition 4.5 *Given $f : \mathcal{B}^k \to \mathcal{B}$ and $g : \mathcal{B}^{k'} \to \mathcal{B}$, continuous functions with p-levels of (i_f, j_f) and (i_g, j_g) respectively. If $i_f > i_g$ or $j_f > j_g$, then $g \not\preceq f$.*

Proof: Immediate by definition of p-levels and theorem 3.7. □

4.2 Coefficients of Linear Coherence

Given a continuous function $f : \mathcal{B}^k \to \mathcal{B}$, we define two numerical properties related to the linearly coherent subsets of $\pi_1(tr(f))$.

Definition 4.6 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. We define the* coefficient of (linear) coherence *of f by*

$$cc(f) = \min\{|A| : A \subseteq \pi_1(tr(f)), |A| \geq 2, A \textit{ coherent}\}$$

and $cc(f)$ is defined to be ∞ when $\pi_1(tr(f))$ has no non-singleton linearly coherent subset.

Definition 4.7 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. We define the* bivalued coefficient of (linear) coherence *of f by*

$$bcc(f) = \min\{|A| : A \subseteq \pi_1(tr(f))\, |A| \geq 3, A \textit{ coherent and bivalued}\}$$

and $bcc(f)$ is defined to be ∞ when $\pi_1(tr(f))$ has no non-singleton bivalued linearly coherent subset.

Recall that a coherent bivalued subset of the first projection of a trace must have a size of at least 3. It is easy to see that $cc(f) \leq bcc(f)$, where $\leq$ is extended to account for the ∞ symbol in the usual way.

The following proposition relates coefficients of coherence and p-levels, and in conjunction with proposition 4.5 gives a quick way to determine inexpressibility of various functions.

Proposition 4.8 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous functions. Then f has a p-level of $(bcc(f) - 1, cc(f) - 1)$, where $bcc(f) - 1$ is ∞ when $bcc(f) = \infty$ and $cc(f) - 1$ is ∞ when $cc(f) = \infty$.*

Proof: We first start by examining $cc(f)$. There are three cases we consider

1. ($cc(f)=2$) We show that f is invariant under $S_2^{\{1\},\{1,2\}}$ but not $S_3^{\{1,2\},\{1,2,3\}}$. Assume f is not invariant under $S_2^{\{1\},\{1,2\}}$. Then there exist tuples

$$\begin{array}{ccc} \left(x_1^1, x_2^1\right) & \in & S_2^{\{1\},\{1,2\}} \\ & \vdots & \\ \left(x_1^k, x_2^k\right) & \in & S_2^{\{1\},\{1,2\}} \end{array}$$

such that $\{y_1, y_2\} \notin S_2^{\{1\},\{1,2\}}$, with $y_i = f(x_i^1, \ldots, x_i^k)$. This means that $y_1 \neq \perp$ and $y_1 \neq y_2$. It is easy to see that $(x_1^1, \ldots, x_1^k) \leq (x_2^1, \ldots, x_2^k)$, since for each $i \leq k$, either $x_1^i = \perp$ or $x_1^i = x_2^i$. So by monotonicity of f, $y_1 \leq y_2$, contradicting $y_1 \neq \perp$, and $y_1 \neq y_2$. So f must be invariant under $S_2^{\{1\},\{1,2\}}$. On the other hand, applying f to the tuples

$$\begin{array}{ccc} \left(x_1^1, x_2^1, \perp\right) & \in & S_3^{\{1,2\},\{1,2,3\}} \\ & \vdots & \\ \left(x_1^k, x_2^k, \perp\right) & \in & S_3^{\{1,2\},\{1,2,3\}} \end{array}$$

where the first two columns are the elements of the first projection of the trace forming a linearly coherent subset of size 2 (since $cc(f)=2$) yields the tuple $(tt, tt, \perp)$ or $(ff, ff, \perp)$, neither of which is in $S_3^{\{1,2\},\{1,2,3\}}$.

2. ($3 \leq cc(f) < \infty$) We show f is invariant under $S_{cc(f)}^{\{1,\ldots,cc(f)-1\},\{1,\ldots,cc(f)\}}$ but not under $S_{cc(f)+1}^{\{1,\ldots,cc(f)\},\{1,\ldots,cc(f)+1\}}$. Assume f is not invariant under $S_{cc(f)}^{\{1,\ldots,cc(f)-1\},\{1,\ldots,cc(f)\}}$. Then there exist tuples

$$\begin{array}{ccc} \left(x_1^1, \ldots, x_{cc(f)}^1\right) & \in & S_{cc(f)}^{\{1,\ldots,cc(f)-1\},\{1,\ldots,cc(f)\}} \\ & \vdots & \\ \left(x_1^k, \ldots, x_{cc(f)}^k\right) & \in & S_{cc(f)}^{\{1,\ldots,cc(f)-1\},\{1,\ldots,cc(f)\}} \end{array}$$

such that $\left(y_1, \ldots, y_{cc(f)}\right) \notin S_{cc(f)}^{\{1,\ldots,cc(f)-1\},\{1,\ldots,cc(f)\}}$ with $y_i = f(x_i^1, \ldots, x_i^k)$. This means $\forall i \leq cc(f)-1$, $y_i \neq \perp$ and $\exists I, J$ with $y_I \neq y_J$. Let $C \subseteq \pi_1(tr(f))$ be an Egli-Milner lowerbound of the first $cc(f)-1$ columns of the given tuples, $|C| \leq cc(f)-1$. We cannot have $|C| = \{\underline{v}\}$, since that would imply that $\underline{v} \leq (x^1_{cc(f)}, \ldots, x^k_{cc(f)})$ (for each $i \leq k$, either one of $x^i_j = \perp$ for $j \leq cc(f) - 1$ — hence $v_j = \perp$ — or $x^i_j = x^i_{j'}$ for all $j, j' \leq cc(f) - 1$ — hence $v_j \leq x^i_j = x^i_{cc(f)}$). But monotonicity of f would imply that for all i, j, $y_i = y_j$, a contradiction. Hence, $|C| \geq 2$. But since the first $cc(f) - 1$ columns of the given tuples form a coherent subset, C being an Egli-Milner lowerbound must also be coherent. But this contradicts the fact that the minimal size for a coherent subset of $\pi_1(tr(f))$ is $cc(f)$. So, f is invariant under $S^{\{1,\ldots,cc(f)-1\},\{1,\ldots,cc(f)\}}_{cc(f)}$. On the other hand, consider the tuples

$$\begin{array}{ccc} \left(x^1_1, \ldots, x^1_{cc(f)}, \perp\right) & \in & S^{\{1,\ldots,cc(f)\},\{1,\ldots,cc(f)+1\}}_{cc(f)+1} \\ & \vdots & \\ \left(x^k_1, \ldots, x^k_{cc(f)}, \perp\right) & \in & S^{\{1,\ldots,cc(f)\},\{1,\ldots,cc(f)+1\}}_{cc(f)+1} \end{array}$$

where the first $cc(f)$ columns are the elements of a coherent subset of size $cc(f)$ of $\pi_1(tr(f))$ (which exists by assumption). Appplying f to these tuples yields a tuple $(y_1, \ldots, y_{cc(f)}, \perp)$ with $y_i \neq \perp$ for $i \leq cc(f)$, which cannot be in $S^{\{1,\ldots,cc(f)\},\{1,\ldots,cc(f)+1\}}_{cc(f)+1}$.

3. ($cc(f) = \infty$) We show that f is invariant under all presequentiality relations of the form $S^{\{1,\ldots,i\},\{1,\ldots,i+1\}}_{i+1}$. Assume that there exists an i such that f is not invariant under $S^{\{1,\ldots,i\},\{1,\ldots,i+1\}}_{i+1}$. The same reasoning as in the previous case leads to a contradiction (instead of contradicting the minimal size of a coherent subset of $\pi_1(tr(f))$ being $cc(f)$, we contradict the fact that there is no coherent subset of $\pi_1(tr(f))$).

We now examine $bcc(f)$. The cases are fundamentally similar.

1. ($bcc(f) = 3$) We show f is invariant under $S^{\{1,2\},\{1,2\}}_2$ but not $S^{\{1,2,3\},\{1,2,3\}}_3$. Assume f is not invariant under $S^{\{1,2\},\{1,2\}}_2$. Then there exist tuples

$$\begin{array}{ccc} \left(x^1_1, x^1_2\right) & \in & S^{\{1,2\},\{1,2\}}_2 \\ & \vdots & \\ \left(x^2_1, x^2_2\right) & \in & S^{\{1,2\},\{1,2\}}_2 \end{array}$$

such that $(y_1, y_2) \notin S_2^{\{1,2\},\{1,2\}}$ with $y_i = f(x_i^1, \ldots, x_i^k)$. This means that $y_1, y_2 \neq \perp$ and $y_1 \neq y_2$. But $(x_1^1, \ldots, x_1^k)$ and $(x_2^1, \ldots, x_2^k)$ are linearly coherent by assumption, so by monotonicity, $y_1 = y_2$, contradicting the above statement. Hence, f must be invariant under $S_2^{\{1,2\},\{1,2\}}$. On the other hand, consider the tuples

$$\begin{array}{rcl} (x_1^1, \ldots, x_3^1) & \in & S_3^{\{1,2,3\},\{1,2,3\}} \\ & \vdots & \\ \left(x_1^k, \ldots, x_3^k\right) & \in & S_3^{\{1,2,3\},\{1,2,3\}} \end{array}$$

where each column is an element of the coherent subset of size 3 of $\pi_1(tr(f))$ (which exists by assumption). Since the subset is bivalued, applying f to these tuples yields a tuple (y_1, y_2, y_3) which has no $\perp$ and which has $y_I \neq y_J$ for some I, J. So this tuple is not in $S_3^{\{1,2,3\},\{1,2,3\}}$.

2. $(4 \leq bcc(f) < \infty)$ We show f is invariant under $S_{bcc(f)-1}^{\{1,\ldots,bcc(f)-1\},\{1,\ldots,bcc(f)-1\}}$ but not under $S_{bcc(f)}^{\{1,\ldots,bcc(f)\},\{1,\ldots,bcc(f)\}}$. Assume f is not invariant under $S_{bcc(f)-1}^{\{1,\ldots,bcc(f)-1\},\{1,\ldots,bcc(f)-1\}}$. Then there exist tuples

$$\begin{array}{rcl} \left(x_1^1, \ldots, x_{bcc(f)-1}^1\right) & \in & S_{bcc(f)-1}^{\{1,\ldots,bcc(f)-1\},\{1,\ldots,bcc(f)-1\}} \\ & \vdots & \\ \left(x_1^k, \ldots, x_{bcc(f)-1}^k\right) & \in & S_{bcc(f)-1}^{\{1,\ldots,bcc(f)-1\},\{1,\ldots,bcc(f)-1\}} \end{array}$$

such that applying f yields $\left(y_1, \ldots, y_{bcc(f)-1}\right) \notin S_{bcc(f)-1}^{\{1,\ldots,bcc(f)-1\},\{1,\ldots,bcc(f)-1\}}$ with $y_i = f(x_i^1, \ldots, x_i^k)$. This means $\forall i \leq bcc(f) - 1$, $y_i \neq \perp$ and $\exists I, J$ with $y_I \neq y_J$. Let $C \subseteq \pi_1(tr(f))$ be an Egli-Milner lowerbound of the columns of the given tuples, $|C| \leq bcc(f) - 1$. We cannot have $|C| = 1$, since that would imply that all y_i have the same value. Hence, $|C| \geq 2$. But since the columns of the given tuples form a coherent subset, C being an Egli-Milner lowerbound must also be coherent, and bivalued since not all y_i have the same value. But this contradicts the fact that the minimal size for a bivalued coherent subset of $\pi_1(tr(f))$ is $bcc(f)$. So, f is invariant under $S_{bcc(f)-1}^{\{1,\ldots,bcc(f)-1\},\{1,\ldots,bcc(f)-1\}}$. On the other hand, consider the tuples

$$\begin{array}{rcl}\left(x_1^1, \ldots, x_{bcc(f)}^1\right) & \in & S_{bcc(f)}^{\{1,\ldots,bcc(f)\},\{1,\ldots,bcc(f)\}} \\ & \vdots & \\ \left(x_1^k, \ldots, x_{bcc(f)}^k\right) & \in & S_{bcc(f)}^{\{1,\ldots,bcc(f)\},\{1,\ldots,bcc(f)\}}\end{array}$$

where the columns are the elements of a bivalued coherent subset of size $bcc(f)$ of $\pi_1(tr(f))$ (which exists by assumption). Applying f to these tuples yields a tuple $(y_1, \ldots, y_{bcc(f)})$ with $y_i \neq \perp$ for all i, and with $y_I \neq y_J$ for some I, J. This tuple cannot be in $S_{bcc(f)}^{\{1,\ldots,bcc(f)\},\{1,\ldots,bcc(f)\}}$.

3. $(bcc(f) = \infty)$ We show that f is invariant under all presequentiality relations of the form $S_i^{\{1,\ldots,i\},\{1,\ldots,i\}}$. Assume that there exists an i such that f is not invariant under $S_i^{\{1,\ldots,i\},\{1,\ldots,i\}}$. The same reasoning as in the previous case leads to a contradiction (instead of contradicting the minimal size of a bivalued coherent subset of $\pi_1(tr(f))$ being $bcc(f)$, we contradict the fact that there is no bivalued coherent subset of $\pi_1(tr(f))$).

□

Moreover, restricting a function by fixing one of its inputs affects the coefficients of coherence in a somewhat well-behaved way:

Lemma 4.9 *Given $f : \mathcal{B}^{k+1} \to \mathcal{B}$ a continuous function and $f' : \mathcal{B}^k \to \mathcal{B}$ defined by*

$$f'(x_1, \ldots, x_k) = f(x_1, \ldots, y, \ldots, x_k)$$

for some fixed y as the i^{th} argument of f. Then $cc(f') \geq cc(f)$ and $bcc(f') \geq bcc(f)$.

Proof: We will prove the result concerning coefficients of coherence. We consider two cases:

1. $(cc(f) = \infty)$ In this case, there is no linearly coherent subset of $\pi_1(tr(f))$, and hence there can be no linearly coherent subset of $\pi_1(tr(f'))$ (otherwise, it would yield a linearly coherent subset of $\pi_1(tr(f))$. Hence, $cc(f') = \infty \geq cc(f)$ by definition.

2. ($cc(f) < \infty$) Given $A \subseteq \pi_1(tr(f'))$ a coherent subset of size $cc(f')$. Let B be the following set:

$$\{(x_1, \ldots, x_{k+1}) \in \pi_1(tr(f)) : (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1}) \in A, x_i \leq y\}.$$

We check that $B \subseteq \pi_1(tr(f))$ is linearly coherent. First, notice that $|B| = |A|$. Moreover, we see that for all tuples in B, the i$^{\text{th}}$ position is either a $\perp$ or a value y. Added to the fact that A is linearly coherent, we see that B must be linearly coherent, and hence $cc(f) \leq cc(f')$.

The proof of the result for bivalued coefficients of coherence is identical. □

4.3 Characterizations of Function Classes

It is possible to classify boolean functions in various known classes by simply looking at which presequentiality relations they are invariant under — or equivalently, by looking at their p-level. We give in this section the p-level characterizations of the most important function classes, namely the continuous, stable and sequential function classes.

4.3.1 Continuous Functions

There exist some very weak presequentiality relations. In fact, some presequentiality relations are such that every continuous function is invariant under them.

Proposition 4.10 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous functions. Then f must have a p-level of the form (i, j) with $i \geq 2$, $j \geq 1$.*

Proof: We consider 3 cases:

1. ($cc(f) = \infty$) Then $bcc(f) = \infty$, and the p-level of f is just (∞, ∞), by 4.8.

2. ($cc(f) < \infty, bcc(f) = \infty$) Then there exists an $A \subseteq \pi_1(tr(f))$ with $|A| \geq 2$ (which is the minimal size for a non-singleton linearly coherent subset). Hence, $cc(f) \geq 2$, and the p-level of f must be of the form (∞, j) with $j \geq cc(f) - 1 \geq 1$.

3. ($bcc(f) < \infty$) Then $cc(f) < \infty$ as well. There exists an $A \subseteq \pi_1(tr(f))$ with $|A| \geq 2$ and a $B \subseteq \pi_1(tr(f))$ bivalued, with $|B| \geq 3$ (the minimal size for a non-singleton bivalued linearly coherent subset of the trace of a function). Hence, we must have $cc(f) \geq 2$ and $bcc(f) \geq 3$, so the p-level of f must be of the form (i, j) with $i \geq bcc(f) - 1 \geq 2$ and $j \geq cc(f) - 1 \geq 1$.

□

Hence, every continuous function must be invariant under the following presequentiality relations:

- $S_n^{\emptyset,B}$, $\forall n, B \subseteq \{1, \ldots, n\}$.
- $S_n^{\{i\},B}$, $\forall n, 1 \leq i \leq n, B \subseteq \{1, \ldots, n\}$.
- $S_n^{A,A}$, $\forall n, 1 \leq i \leq n, A \subseteq \{1, \ldots, n\}, |A| = 2$.

If it is known that a function f is monovalued, we can further restrict the p-level of f via the following proposition.

Proposition 4.11 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a monovalued continuous function. Then f has a p-level of the form (∞, j) with $j \geq 1$.*

Proof: If f is monovalued then $bcc(f) = \infty$, since there can be no bivalued coherent subset of $\pi_1(tr(f))$. Moreover, since f is continuous, it must have a p-level of the form (i, j) with $i \geq 2$ and $j \geq 1$. We know $i = \infty$ (since $bcc(f) = \infty$), so f must have a p-level of the form (∞, j) with $j \geq 1$. □

Hence every monovalued function must be invariant with respect to the following presequentiality relations:

- $S_n^{A,A}$, $\forall n$,$\forall A$ with $A \subseteq \{1, \ldots, n\}$.

along with those presequentiality relations under which all continuous functions are invariant.

4.3.2 Sequential Functions

By theorems 2.3 and 2.5, any sequential (PCF-definable) function must be invariant under all sequentiality relations — including presequentiality relations — implying that any sequential function must have a p-level of (∞, ∞). Moreover, it is sufficient for f to have a p-level of (∞, ∞) in order to be sequential. To show this, we use the following lemma that relates sequentiality to the coefficient of coherence of the function.

Lemma 4.12 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function, f is sequential iff $cc(f) = \infty$.*

Proof: ($\Rightarrow$) We shall prove the contrapositive, namely that if $cc(f) < \infty$, then f cannot be sequential.

We prove this by induction on the arity of f. First, note that if f has arity 1, it cannot have a linearly coherent subset of the first projection of the trace. If f has arity 2, it is easy to see that $\pi_1(tr(f))$ having a linearly coherent subset implies that f is not stable, and hence not sequential.

(induction step) Let f be a function of arity $k+1$. Let $A \subseteq \pi_1(tr(f))$, A coherent, $|A| \geq 3$ (since f is stable). We consider two cases:

1. A is $\perp$-covering. Then it is easy to see that f cannot be sequential (no index of sequentiality).

2. There exists an index of sequentiality j. Let y be the value of the tuples of A at position j. Consider the function $f' = f(x_1, \ldots, y, \ldots, x_{k+1})$ of arity k. The trace of this function must contain the tuples of A (minus column j), and these form a coherent subset of $\pi_1(tr(f'))$. So by induction hypothesis, f' is not sequential, and hence neither is f.

($\Leftarrow$) We again prove this by induction on the arity of f.

(base case) $f : \mathcal{B} \to \mathcal{B}$. Consider $f(\perp)$. If $f(\perp) \neq \perp$, then by monotonicity f is constant, and hence sequential. if $f(\perp) = \perp$, then consider $f(y)$ for a fixed y. This must be a constant, so f is sequential (by the definition of sequentiality). All this to show that any function of arity 1 is sequential — as should be obvious.

(induction step) Assume result holds for all functions of arity k. Consider $f : \mathcal{B}^{k+1} \to \mathcal{B}$, with $cc(f) = \infty$.

1. We first need to show that there exist an index of sequentiality. Assume not: $\forall i$, for any fixed $x_j, \forall j \neq i$, $f(x_1, \ldots, \perp, \ldots, x_{k+1}) \neq \perp$. Then $\pi_1(tr(f))$ must be $\perp$-covering, which contradicts $cc(f) = \infty$.
2. Given i the index of sequentiality of f, look at the function $f'(z_1, \ldots, z_k) = f(z_1, \ldots, y, \ldots, z_k)$ for a fixed y in position i. By lemma 4.9, $cc(f') = \infty$, and the induction hypothesis applies to show that f' and thereby f must be sequential.

$\square$

The main result now follows easily:

Proposition 4.13 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function. Then f is sequential iff f has a p-level of (∞, ∞).*

Proof: ($\Rightarrow$) Immediate, since f sequential implies that f is PCF-definable, and hence f must be invariant under all sequentiality relations — including presequentiality relations.

(⇐) f with a p-level of (∞, ∞) means that $cc(f) = \infty$, and this implies by lemma 4.12 that f is sequential. □

4.3.3 Stable Functions

Stability also has a nice characterization in terms of p-levels. In fact, a single presequentiality relation is sufficient to determine stability.

Proposition 4.14 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. Then f is stable iff f has a p-level of the form (i, j) with $i \geq 2$ and $j \geq 2$.*

Proof: (⇒) Given f a stable function. Then $cc(f) \geq 3$,and by proposition 4.8, f must have a p-level of the form (i, j) with $j \geq cc(f) - 1 \geq 2$. Since f is continuous, by proposition 4.10, $i \geq 2$.

(⇐) Given f with a p-level (i, j) with $j \geq 2$. By proposition 4.8, $cc(f) - 1 \geq 2$, so that $cc(f) \geq 3$. Hence, f must be stable. □

In other words, f is stable iff f is invariant under $S_n^{A,B}$, $\forall n$, $\forall A, B$ with $|A| = 2$ and $A \subset B \subseteq \{1, \ldots, n\}$ — along with the usual presequentiality relations under which every continuous function must be invariant. An unstable continuous function therefore must have a p-level of the form $(i, 1)$ for some $i \geq 2$.

Chapter 5

Stable Boolean Functions

In this chapter, we study stable first-order boolean functions. We are mainly interested in describing the structure of those functions as a substructure of the larger CONT semilattice. Among other things, we distinguish a natural partition of stable functions into 2 classes, and describe two hierarchies of functions, one in each of the classes. These two hierarchies (the Gustave hierarchy and the Bivalued-Gustave hierarchy) can be combined to form a "two-dimensional" infinite hierarchy, the Composite hierarchy, spanning the different p-levels. Moreover, the Gustave hierachy can be shown to be a minimal hierarchy, in the sense that any stable function must dominate one of the functions in the hierarchy.

5.1 Fundamental Lemmas

In this section, we make some remarks concerning stable functions of low arity. These will be useful as base cases for induction proofs.

We observe that stable functions become extremely simple as their arity decreases. Among other things, we note that:

Lemma 5.1 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a stable continuous function of arity k less than 3. Then f must be sequential.*

Proof: Consider $f : \mathcal{B} \to \mathcal{B}$, it is clear that f must be sequential.

Consider $f : \mathcal{B}^2 \to \mathcal{B}$. If $|tr(f)| = 1$, then f must be sequential. If $|tr(f)| > 1$, then at least one column of $\pi_1(tr(f))$ must have a $\perp$ (else f would be sequential). And if one column has a $\perp$, the other column cannot have $\perp$, or f would be unstable. So there is at most one column of $\pi_1(tr(f))$ with a $\perp$. Without loss of generality, assume the $\perp$ appears in the first column.

Two cases may arise:

1. There exists tuples of the form $(\perp, v)$ and $(\perp, \neg v)$ in $\pi_1(tr(f))$. Then there cannot be any other tuple in the trace or f would be unstable. But such a trace is sequential.

2. A tuple of the form $(\perp, v)$ only appears in $\pi_1(tr(f))$ (for some value v). Then there cannot be any other tuple of the form $(\_, v)$ in $\pi_1(tr(f))$ or f would be unstable. So every other tuple must be of the form $(\_, \neg v)$, and f is easily seen to be sequential.

$\square$

Lemma 5.2 *Given $f : \mathcal{B}^3 \to \mathcal{B}$ a stable non-sequential function. Then f is completely $\perp$-covering.*

Proof: f being stable and non-sequential, there must be at least one linearly coherent subset of $\pi_1(tr(f))$ by lemma 4.12.

Assume that some linearly coherent subset of $\pi_1(tr(f))$ is not $\perp$-covering. By stability of f, that subset $A \subseteq \pi_1(tr(f))$ must be of size at least 3. The set of tuples A must have column with only *tt* or *ff* (since it is not $\perp$-covering). The remaining two columns of A must be coherent as well, and hence there must be two tuples in A that are linearly coherent with respect to those two columns.

Since the remaining column contains (by assumption) the same value for both tuples, these tuples form a linearly coherent subset $B \subseteq A$ with $|B| = 2$. This contradicts the stability of f. $\square$

The following lemma will also prove useful

Lemma 5.3 *Given $f : \mathcal{B}^{k+1} \to \mathcal{B}$ a stable continuous function and $f' : \mathcal{B}^k \to \mathcal{B}$ defined by*

$$f'(x_1, \ldots, x_k) = f(x_1, \ldots, y, \ldots, x_k)$$

for some fixed y as the i^{th} argument of f. Then f' is stable.

Proof: Since the function f is stable, $cc(f) \geq 3$. So by lemma 4.9, $cc(f') \geq cc(f) \geq 3$, and f' is stable. $\square$

5.2 The Structure of Stable Functions

A *stable degree of parallelism* is an equivalence class of the $\equiv$ relation, where every function in the equivalence class is stable. This occurs as soon as a single stable function is part of the equivalence class.

Proposition 5.4 *Given $f : \mathcal{B}^k \to \mathcal{B}$ and $g : \mathcal{B}^{k'} \to \mathcal{B}$ continuous functions. If $f \equiv g$ and f is stable, then g is stable.*

Proof: Assume f is stable, g is not stable and $f \equiv g$. Then, $g \preceq f$. However, f stable implies f invariant under $S_3^{\{1,2\},\{1,2,3\}}$, but g unstable means g is not invariant under $S_3^{\{1,2\},\{1,2,3\}}$. $\square$

It thus makes sense to study STABLE, the subposet of stable degrees of parallelism. It turns out that just like CONT, STABLE is a sup-semilattice with a top and a bottom element. The bottom element of STABLE is just the equivalence class of sequential functions. We need to verify that there is a top element, and that least upperbounds exist. The following function, as was noted by Plotkin and communicated to Curien in [Cur93], is maximal amongst all stable functions.

Definition 5.5 *Let BP (Berry-Plotkin) be the function defined by the following trace (in matrix form):*

$$\begin{array}{|ccc|c|} \hline \perp & tt & ff & tt \\ tt & ff & \perp & ff \\ ff & \perp & tt & ff \\ \hline \end{array}$$

Proposition 5.6 *Given $f : \mathcal{B}^k \to \mathcal{B}$ any stable continuous function. Then $f \preceq BP$.*

Proof: We show this by induction on the arity of f.

(base case, arity less than 3) By lemma 5.1, f must be sequential, and $f \preceq$BP holds trivially.

(induction step, arity k+1) Assume the result holds for any function of arity less than $k + 1$. Given $f : \mathcal{B}^{k+1} \to \mathcal{B}$, define the following functions:

$$\begin{aligned} g(x_1, \ldots, x_k) &= f(tt, x_1, \ldots, x_k) \\ h(x_1, \ldots, x_k) &= f(ff, x_1, \ldots, x_k) \end{aligned}$$

By lemma 5.3, g and h are stable, hence by induction hypothesis $g \preceq$BP and $h \preceq$BP. Let G, H be such that $g = [\![G]\!]$BP and $h = [\![H]\!]$BP.

Define the functions g', h' via the following traces:

$$\begin{aligned} tr(g') &= \{((x_1, \ldots, x_k), y) : \\ &\quad f(\perp, x_1, \ldots, x_k) \text{ is defined and } y = tt \text{ or} \\ &\quad f(tt, x_1, \ldots, x_k) \text{ is defined and } y = ff\} \\ tr(h') &= \{((x_1, \ldots, x_k), y) : \\ &\quad f(\perp, x_1, \ldots, x_k) \text{ is defined and } y = tt \text{ or} \\ &\quad f(ff, x_1, \ldots, x_k) \text{ is defined and } y = ff\} \end{aligned}$$

It is easy to see that both g' and h' are stable, since $\pi_1(tr(g')) = \pi_1(tr(g))$ and $\pi_1(tr(h')) = \pi_1(tr(h))$. So by induction hypothesis, $g' \preceq$BP and $h' \preceq$BP. Let G', H' be such that $g' = [\![G']\!]$BP and $h' = [\![H']\!]$BP.

Consider the following test T:

$$T = \lambda b \lambda x_1 \ldots x_{k+1}.b(x_1, G'(b, x_2, \ldots, x_{k+1}), \neg H'(b, x_2, \ldots, x_{k+1})).$$

It is clear that

$$\begin{array}{ll} [\![T]\!](\mathrm{BP}, x_1, \ldots, x_{k+1}) = tt & \text{if } (\bot, x_2, \ldots, x_{k+1}) \text{ is in the trace of } f \\ [\![T]\!](\mathrm{BP}, x_1, \ldots, x_{k+1}) = f\!f & \text{if } (tt, x_2, \ldots, x_{k+1}) \text{ or } (f\!f, x_2, \ldots, x_{k+1}) \\ & \text{is in the trace of } f \end{array}$$

And we can construct the following term

$$\begin{array}{rcl} M & = & \lambda b \lambda x_1 \ldots x_{k+1}.\text{if } T(b, x_1, \ldots, x_{k+1}) \text{ then } G(b, x_2, \ldots, x_{k+1}) \\ & & \text{else if } x_1 \text{ then } G(b, x_2, \ldots, x_{k+1}) \text{ else } H(b, x_2, \ldots, x_{k+1}) \text{ fi fi} \end{array}$$

such that $f = [\![M]\!]\mathrm{BP}$. □

We can easily characterize the degree of parallelism of BP — the top element of the STABLE semilattice — in terms of p-levels:

Proposition 5.7 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function. Then f has a p-level of $(2,2)$ iff $f \equiv BP$.*

Proof: ($\Leftarrow$) Given $f \equiv$BP. Then f must be invariant under the same sequentiality relations, hence the p-level of f is the same as the p-level of BP, namely $(2,2)$.

($\Rightarrow$) Given f with a p-level of $(2,2)$. This means that $bcc(f) = 3$, in other words, there exists $A \subseteq \pi_1(tr(f))$ bivalued and linearly coherent, with $|A| = 3$. We can assume without loss of generality that one element of A returns tt and the remaining two return $f\!f$ (otherwise, consider $neg(f)$ which is equiparallel to f and has the desired property). Define $g : tr(\mathrm{BP}) \to tr(f)$ by sending the first trace element of BP (the one returning tt) to the element of A returning tt, and the remaining elements of BP to the elements of A returning $f\!f$. Since A is linearly coherent, it is clear that g satisfies the condition of corollary 3.11, and BP$\preceq f$, and hence $f \equiv$BP. □

As for least upperbounds in STABLE, we verify easily that the following holds.

Proposition 5.8 *Given $f : \mathcal{B}^k \to \mathcal{B}$, $g : \mathcal{B}^{k'} \to \mathcal{B}$ stable continuous functions. Then $f + g$ is stable.*

Proof: By proposition 4.14, f and g stable imply that f and g have p-levels of the form (i_f, j_f) and (i_g, j_g) with $j_f, j_g \geq 2$. By proposition 4.4, $f + g$ must have a p-level $(\min(i_f, i_g), \min(j_f, j_g))$, and $\min(j_f, j_g) \geq 2$, hence $f + g$ is stable. □

There exist an interesting partition of STABLE that we will start studying here and finalize in the next chapter. Informally, we can classify functions on whether or not they return only one value. Define a *monovalued degree of parallelism* to be a degree of parallelism containing a monovalued function. A degree of parallelism where all functions are bivalued is called a bivalued degree of parallelism. If the degree of parallelism in question is stable, we shall talk about a stable monovalued degree of parallelism or a stable bivalued degree of parallelism.

As the next proposition shows, "monovalueness" is preserved by the least upperbound operation

Proposition 5.9 *Given $[f], [g]$ monovalued degrees of parallelism. Then $[f + g]$ is also a monovalued degree of parallelism.*

Proof: Given f and g functions in a monovalued degree of parallelism, then there exist monovalued functions f' and g' with $f \equiv f'$ and $g \equiv g'$. Without loss of generality, since a function and its negative are equiparallel, we can assume that f' and g' return the same value. By construction, it is easy to see that $f' + g'$ is a monovalued function, and we know this is equiparallel to $f + g$. Hence, $f + g$ is in a monovalued degree of parallelism. □

Before characterizing monovalued degrees of parallelism in terms of p-levels, let us first prove this technical lemma:

Lemma 5.10 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function, with $bcc(f) = \infty$. Then $[f]$ is a monovalued degree of parallelism.*

Proof: We proceed by induction on the arity of f. Clearly, a function of arity 1 is sequential, hence it is trivially equiparallel to a monovalued function (all sequential functions are equiparallel). For the induction step, assume that the result holds for any function of arity at most k. We shall prove it holds for $f : \mathcal{B}^{k+1} \to \mathcal{B}$.

If f is $\perp$-covering, then f must by assumption be monovalued, so we are done. Therefore, consider the case where f has an index of sequentiality i. Without loss of generality, let i be 1. Define the following functions:

$$\begin{aligned} g(x_1, \ldots, x_k) &= f(tt, x_1, \ldots, x_k) \\ h(x_1, \ldots, x_k) &= f(f\!f, x_1, \ldots, x_k) \end{aligned}$$

It is easy to see that every linearly coherent subset of $\pi_1(tr(g))$ and $\pi_1(tr(h))$ must be monovalued (by an argument similar to the one used to prove lemma 4.9). So by induction hypothesis, $[g]$ and $[h]$ are monovalued degrees of parallelism, and by proposition 5.9, $[g + h]$ is monovalued as well. We claim $[f] = [g + h]$.

1. $(g + h \preceq f)$ It is easy to see that $g \preceq f$ and $h \preceq f$ — for example,

$$g = [\![\lambda p \lambda x_1 \ldots x_k . p(tt, x_1, \ldots, x_k)]\!] f$$

and hence because $g + h$ is the least upperbound of g and h, $g + h \preceq f$.

2. $(f \preceq g + h)$ Note that by definition, $g \preceq g + h$ and $h \preceq g + h$. Let $g = [\![G]\!](g + h)$ and $h = [\![H]\!](g + h)$. Since f has an index of sequentiality of 1, it must have an actual argument in its first position to return a result. Hence, the following term:

$$F = \lambda p \lambda x_1 \ldots x_{k+1}.\text{if } x_1 \text{ then } G(p, x_2, \ldots, x_{k+1}) \text{ else } H(p, x_2, \ldots, x_{k+1})$$

and $f = [\![F]\!](g + h)$.

Hence, $f \equiv g+h$ and $[f] = [g+h]$, so $[f]$ is a monovalued degree of parallelism.

$\square$

Proposition 5.11 *A degree of parallelism is monovalued iff it has a p-level of the form* (∞, j).

Proof: (of proposition 5.11)

($\Rightarrow$) Given $[f]$ a monovalued degree of parallelism. Then $f \equiv g$ with g a monovalued function. But by proposition 4.11, g must have a p-level of the form (∞, j).

($\Leftarrow$) Given $[f]$ a degree of parallelism with a p-level of the form (∞, j). Then by proposition 4.8, $bcc(f) = \infty$, and by lemma 5.10, $[f]$ must be monovalued.

$\square$

As an example of a bivalued function in a monovalued degree of parallelism, consider the function given by the following trace in matrix form:

$$\begin{array}{|cccc|c|}
\hline
tt & \perp & tt & ff & tt \\
tt & ff & \perp & tt & tt \\
tt & tt & ff & \perp & tt \\
ff & \perp & tt & ff & ff \\
ff & ff & \perp & tt & ff \\
ff & tt & ff & \perp & ff \\
\hline
\end{array}$$

This function can easily be seen as the least upperbound of GUST + neg(GUST), which are both monovalued (see section 2.1.4).

We note that the Bucciarelli hierarchy presented in section 3.5 is a subposet of monovalued degrees of parallelism.

5.3 The Gustave Hierarchy

We now identify a hierarchy of monovalued stable degrees of parallelism. This hierarchy, derived from Gustave's function is in fact part of the Bucciarelli hierarchy of section 3.5, but the representation we choose here is minimal and the traces are slightly easier to work with.

Definition 5.12 *Let $GUST_i : \mathcal{B}^{2i+1} \to \mathcal{B}$ $(i \geq 1)$ be defined by the following trace (in matrix form):*

$$\begin{array}{|cccccc|c|}
\hline
\perp & tt & ff & \cdots & tt & ff & tt \\
ff & \perp & tt & \cdots & ff & tt & tt \\
tt & ff & \perp & \cdots & tt & ff & tt \\
 & \vdots & & & & & \vdots \\
ff & tt & ff & \cdots & \perp & tt & tt \\
tt & ff & tt & \cdots & ff & \perp & tt \\
\hline
\end{array}$$

We note that GUST_1 is just GUST. Moreover, these functions are actually equiparallel to a subhierarchy of Bucciarelli functions. First, let us characterize the Bucciarelli functions we are interested in:

Lemma 5.13 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a monovalued continuous function with $|tr(f)| = cc(f) = n$, then $f \equiv BUCC_{(n,n)}$.*

Proof: By construction, we know $\left|tr(\mathrm{BUCC}_{(n,n)})\right| = n$ and $cc(\mathrm{BUCC}_{(n,n)}) = n$.

Define functions

$$g : tr(\mathrm{BUCC}_{(n,n)}) \to tr(f)$$

$$h : tr(f) \to tr(\mathrm{BUCC}_{(n,n)})$$

with the only constraint of being surjective. It is clear that the conditions of corollary 3.11 apply to both functions and that $f \equiv \mathrm{BUCC}_{(n,n)}$. □

And the following result can now be derived:

Proposition 5.14 *$GUST_i \equiv BUCC_{(2i+1,2i+1)}$.*

Proof: It is clear by inspection that $cc(\mathrm{GUST}_i) = 2i + 1$, and since GUST_i is monovalued and $|tr(\mathrm{GUST}_i)| = 2i + 1$, applying lemma 5.13 yields the equivalence. □

We now show that the Gustave hierarchy spans the p-level spectrum of monovalued degrees of parallelism.

Proposition 5.15 *$GUST_i$ has a p-level of $(\infty, 2i)$.*

Proof: Since GUST_i is monovalued and $cc(\mathrm{GUST}_i) = 2i + 1$, propositions 4.11 and 4.8 give that GUST_i has a p-level of $(\infty, 2i)$. □

This hierarchy will have an important role in the remainder of this chapter. It is, in some sense, the minimal non-sequential hierarchy, and it leads — via suitable transformations — to an important bivalued hierarchy.

5.4 Minimality of the Gustave Hierarchy

We present in this section some results concerning minimality in the STABLE semilattice. We know that the degree of sequential functions is the minimal degree in this semilattice. A natural question to ask is whether there is a non-sequential minimal degree of parallelism. The existence of the Gustave hierarchy allows us to answer this question negatively.

Proposition 5.16 *There is no minimal stable non-sequential function.*

Proof: Assume g is a stable non-sequential function that is minimal, i.e. $\forall f$, f stable, non-sequential, $g \preceq f$.

Since g is not sequential, by lemma 4.13, there is A, B, n such that g is not invariant under $S_n^{A,B}$.

Consider $\mathrm{GUST}_{|A|}$. By lemma 5.15, since $|A| \leq 2\,|A|$, $\mathrm{GUST}_{|A|}$ is invariant under $S_n^{A,B}$.

Hence $g \not\preceq \mathrm{GUST}_{|A|}$, a contradiction. □

However, we can show that the Gustave hierarchy form a minimal hierarchy, in the sense that any non-sequential function must dominate one of the Gustave functions.

Proposition 5.17 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a stable non-sequential function. Then there exists an integer i such that $GUST_i \preceq f$.*

Proof: The function f being non-sequential implies that $cc(f) < \infty$ (lemma 4.12). Moreover, f being stable implies that $cc(f) \geq 3$ (propositions 4.8 and 4.14). Let A be a linearly coherent subset of $\pi_1(tr(f))$ of size $cc(f)$. Define a function $g : tr(\mathrm{GUST}_{cc(f)}) \to tr(f)$ by sending every element of $tr(\mathrm{GUST}_{cc(f)})$ surjectively to the elements of A. It is easy to see that the conditions of corollary 3.11 are satisfied, so that $\mathrm{GUST}_{cc(f)} \preceq f$. □

5.5 The Bivalued-Gustave Hierarchy

We begin in this section an investigation of the structure of bivalued stable degrees of parallelism. We shall take as our starting point the Gustave hierarchy, where the function will be suitably transformed. We modify the trace of these functions — the second projection of the traces to be precise — to make them bivalued. We then study the structure of these new functions.[1]

[1]We could also have extended the obvious full $\mathrm{BUCC}_{(i,i)}$ subhierarchy. However, it is not clear that proposition 5.21 would hold. For the sake of simplicity, we therefore restrict our attention to

Definition 5.18 *Let $BGUST_i^j : \mathcal{B}^{2i+1} \to \mathcal{B}$ $(j \leq i)$ be the function defined by the following trace (in matrix form):*

$$\begin{array}{|cccccc|c|}
\hline
\perp & tt & ff & \cdots & tt & ff & r_1 \\
ff & \perp & tt & \cdots & ff & tt & r_2 \\
tt & ff & \perp & \cdots & tt & ff & r_3 \\
 & \vdots & & & & & \vdots \\
ff & tt & ff & \cdots & \perp & tt & r_{2i} \\
tt & ff & tt & \cdots & ff & \perp & r_{2i+1} \\
\hline
\end{array}$$

with

$$r_l = \begin{cases} ff & \text{if } 1 \leq l \leq j \\ tt & \text{otherwise} \end{cases}$$

Note that $\pi_1(tr(\mathrm{BGUST}_i^j)) = \pi_1(tr(\mathrm{GUST}_i))$ for all j.

Most of the work in this section will revolve around showing that for any i, the functions BGUST_i^k for all k are in fact equiparallel, and that they form a hierarchy. In fact, all functions f with $\pi_1(tr(f)) = \pi_1(tr(\mathrm{GUST}_i))$ and $\pi_2(tr(f)) = \{tt, ff\}$ are equiparallel — independently of exactly which trace elements return tt and which return ff.

Our first lemma shows that if f is a function with the same first projection ofits trace as the trace of GUST_i, and if there are j trace elements of f returning tt or j trace elements of f returning ff $(j \leq i)$, then f is equiparallel to BGUST_i^j.

Lemma 5.19 *Let $f : \mathcal{B}^{2i+1} \to \mathcal{B}$ be a continuous function, such that $\pi_1(tr(f)) = \pi_1(tr(GUST_i))$. If $\pi_2(tr(f)) = \{tt, ff\}$ and we let j be*

$$\min(|\{(v, tt) \in tr(f)\}|, |\{(v, ff) \in tr(f)\}|)$$

then $f \equiv BGUST_i^j$.

---

the Gustave hierarchy. Most of the theory developed in the next sections (including the Composite hierarchy) can (and should!) be extended to the full $\mathrm{BUCC}_{(i,i)}$ subhierarchy. We shall not do so here.

Proof: This is a straightforward application of corollary 3.11. Without loss of generality, assume that $|\{(v, \mathit{ff}) \in tr(f)\}| = j$ (if not, consider $neg(f)$ which is equiparallel to f). Consider any surjective functions

$$g : tr(f) \to tr(\mathrm{BGUST}_i^j)$$

$$h : tr(\mathrm{BGUST}_i^j) \to tr(f)$$

sending a trace returning *tt* to a trace returning *tt* and a trace returning *ff* to a trace returning *ff*. Since $\pi_1(tr(\mathrm{BGUST}_i^j)) = \pi_1(tr(f))$ has only one coherent subset (namely itself), it is easy to see that all the conditions of corollary 3.11 are satisfied, and that $f \equiv \mathrm{BGUST}_i^j$. □

We need now only show that for all $j \leq i$, the functions BGUST_i^j are all equiparallel.

Lemma 5.20 *Given $j, j' \leq i$, $BGUST_i^j \equiv BGUST_i^{j'}$.*

Proof: We prove by induction on j that $\forall j, \mathrm{BGUST}_i^j \equiv \mathrm{BGUST}_i^1$. The case $j = 1$ is trivial. For the induction step ($j \geq 2$), assume that $\mathrm{BGUST}_i^{j-1} \equiv \mathrm{BGUST}_i^1$ and consider BGUST_i^j. We show $\mathrm{BGUST}_i^j \equiv \mathrm{BGUST}_i^{j-1}$. Define the following terms:

$$\begin{aligned} M_1 &= \lambda f \lambda x_1 \ldots x_{2i+1}.\text{if } f(x_1, \ldots, x_{2i+1}) \\ &\quad\ \text{then } f(x_2, \ldots, x_{2i+1}, x_1) \text{ else } \mathit{ff} \text{ fi} \\ M_2 &= \lambda f \lambda x_1 \ldots x_{2i+1}.\text{if } f(x_1, \ldots, x_{2i+1}) \\ &\quad\ \text{then } \mathit{tt} \text{ else } f(x_{2i+1}, x_1, \ldots, x_{2i}) \text{ fi} \end{aligned}$$

It is not so hard to see that $\mathrm{BGUST}_i^j = [\![M_1]\!] \mathrm{BGUST}_i^{j-1}$ and $\mathrm{BGUST}_i^{j-1} = [\![M_2]\!] \mathrm{BGUST}_i^j$, thereby showing $\mathrm{BGUST}_i^j \equiv \mathrm{BGUST}_i^{j-1} \equiv \mathrm{BGUST}_i^1$ by induction hypothesis. □

To unclutter the notation, we shall denote BGUST_i^1 simply by BGUST_i, dropping the superscript.

Proposition 5.21 *Let $f : \mathcal{B}^{2i+1} \to \mathcal{B}$ be a continuous function with*

$$\pi_1(tr(f)) = \pi_1(tr(GUST_i)).$$

Then one of the following holds:

1. $f \equiv GUST_i$
2. $f \equiv BGUST_i$

Proof: Given f with $\pi_1(tr(f)) = \pi_1(tr(\mathrm{GUST}_i))$. We consider two cases:

1. If f is monovalued, then either $f = \mathrm{GUST}_i$ or $f = neg(\mathrm{GUST}_i)$. Either way, $f \equiv \mathrm{GUST}_i$.
2. If f is not monovalued, then let $j = |\{(v, tt) \in tr(f)\}|$. If $j \leq i$, then by lemma 5.19, $f \equiv \mathrm{BGUST}_i^j$, and by lemma 5.20, $f \equiv \mathrm{BGUST}_i$. If $j \geq i+1$, then $|\{(v, ff) \in tr(f)\}| = 2i + 1 - j \leq 2i + 1 - (i+1) \leq i$, so by lemma 5.19, $f \equiv \mathrm{BGUST}_i^{2i+1-j}$ and by lemma 5.20, $f \equiv \mathrm{BGUST}_i$.

□

It remains to show that the functions BGUST_i actually form a hierarchy. Let us first compute the p-level of these functions

Proposition 5.22 *$BGUST_i$ has a p-level of $(2i, 2i)$.*

Proof: We already know that $cc(\mathrm{BGUST}_i) = 2i$ (same first projection of the trace as GUST_i, proposition 5.15). Moreover, the only coherent subset of $\pi_1(tr(\mathrm{BGUST}_i))$ is also bivalued. Hence, $bcc(f) = cc(f) = 2i$, and applying proposition 4.8 yields the result. □

The following two propositions settle the hierarchy issue:

Proposition 5.23 *$\forall i$, $BGUST_{i+1} \preceq BGUST_i$.*

Proof: This is again a straightforward application of corollary 3.11. Consider any surjective function $g : \text{BGUST}_{i+1} \rightarrow \text{BGUST}_i$ sending the unique trace element returning *tt* to the unique trace element returning *tt*, and any trace element returning *ff* to any trace element returning *ff*. It is easy to see that all conditions of corollary 3.11 are satisfied, and $\text{BGUST}_{i+1} \preceq \text{BGUST}_i$. □

Proposition 5.24 *$\forall i$, $BGUST_i \not\preceq BGUST_{i+1}$.*

Proof: The p-level of BGUST_i is $(2i, 2i)$ and the p-level of BGUST_{i+1} is $(2i + 1, 2i + 1)$. By proposition 4.5, $\text{BGUST}_i \not\preceq \text{BGUST}_{i+1}$. □

One notices that $\text{BGUST}_1 \equiv \text{BP}$ (the Berry-Plotkin function of section 5.2). An interesting point to note is that BGUST_1 is maximal amongst stable functions, whereas the corresponding monovalued function, $\text{GUST}_1 \equiv \text{BUCC}_{(3,3)}$, is not maximal amongst monovalued stable functions (consider $\text{BUCC}_{(3,m)}$ for $m \geq 4$).

Functions in the Bivalued-Gustave hierarchy also relate directly to functions in the Gustave hierarchy, as the next proposition shows:

Proposition 5.25 *$\forall i$, $GUST_i \preceq BGUST_i$.*

Proof: This is a trivial application of corollary 3.11, since the first projection of the trace of both functions is the same, and GUST_i is monovalued. □

5.6 The Composite Hierarchy

Combining functions in the Gustave hierarchy and the Bivalued-Gustave hierarchy (via the least upperbound operation) allows us to produce a "two-dimensional" hierarchy. We will be considering functions of the form $\text{BGUST}_i + \text{GUST}_j$, and call the resulting hierarchy the Composite hierarchy. Let us first characterize the p-levels of the functions under consideration:

Proposition 5.26 *$BGUST_i + GUST_j$ has a p-level of $(2i, 2\min(i,j))$.*

Proof: This is a simple application of proposition 5.8, knowing that the p-level of BGUST_i is $(2i, 2i)$ and the p-level of GUST_j is $(\infty, 2j)$. □

The functions form a hierarchy, and the governing equations describing the hierarchy are as follows:

Proposition 5.27 *$BGUST_i + GUST_j \preceq BGUST_{i'} + GUST_{j'}$ if $i' \leq i$ and $\min(i', j') \leq \min(i,j)$.*

Proof: Since $i' \leq i$, proposition 5.23 tells us that $\mathrm{BGUST}_i \leq \mathrm{BGUST}_{i'} \leq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$. We then consider three cases:

1. $(\min(i,j) = i)$ Proposition 5.25 implies that

   $$\mathrm{GUST}_j \preceq \mathrm{BGUST}_j \preceq \mathrm{BGUST}_i \preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$$

   Hence, $\mathrm{BGUST}_i + \mathrm{GUST}_j \preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$.

2. $(\min(i,j) = j,\ \min(i',j') = i')$ By assumption, $i' \leq j$, and hence by proposition 5.25 $\mathrm{GUST}_j \preceq \mathrm{BGUST}_j \preceq \mathrm{BGUST}_{i'} \preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$. Hence $\mathrm{BGUST}_i + \mathrm{GUST}_j \preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$.

3. $(\min(i,j) = j,\ \min(i',j') = j')$ By assumption, $j' \leq j$, and hence

   $$\mathrm{GUST}_j \preceq \mathrm{GUST}_{j'} \preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$$

   Hence $\mathrm{BGUST}_i + \mathrm{GUST}_j \preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$.

□

Proposition 5.28 *$BGUST_i + GUST_j \not\preceq BGUST_{i'} + GUST_{j'}$ if $i < i'$ or $\min(i,j) < \min(i',j')$.*

Proof: If $i < i'$ or $\min(i,j) < \min(i',j')$, then by propositions 5.26 and 4.5, $\mathrm{BGUST}_i + \mathrm{GUST}_j \not\preceq \mathrm{BGUST}_{i'} + \mathrm{GUST}_{j'}$. □

Figure 5.1 presents a picture of part of the Composite hierarchy, with the directed edges giving the minimal implementability results (the rest can be obtained by transitivity).

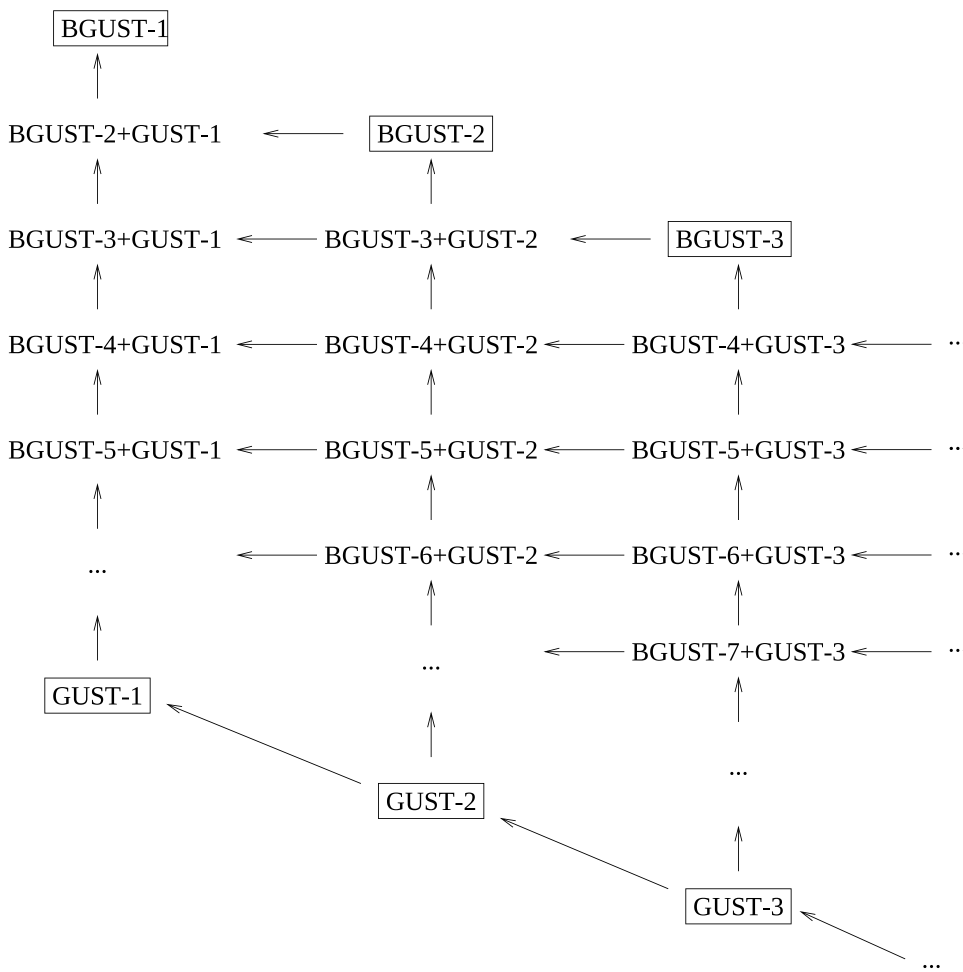

Figure 5.1: The Composite hierarchy

5.7 Summary

We review the structure of STABLE, the poset of stable degrees of parallelism. This poset forms a sup-subsemilattice of CONT with a top element (the degree of the Berry-Plotkin function) and a bottom element (the degree of sequential functions). The top degree of STABLE is chararacterized by containing all and only functions with p-level $(2, 2)$. A natural partition of functions in STABLE can be made by considering monovalued and bivalued degrees of parallelism and this partition has a nice expression in terms of p-levels, namely that a stable degree of parallelism is monovalued iff its p-level is of the form (∞, i), $i \geq 2$.

The Bucciarelli hierarchy introduced in section 3.5 lives inside the monovalued stable degrees of parallelism subposet, one of its subhierarchies (the Gustave hierarchy) admits a compact trace representation and spans the p-levels spectrum of monovalued degrees of parallelism. Modifying somewhat the returned value of functions in the Gustave hierarchy yields a new Bivalued-Gustave hierarchy living in the bivalued stable degrees of parallelism subposet. This new hierarchy also spans in some sense the p-level spectrum. By combining functions in both the Bivalued-Gustave hierarchy and the Gustave hierarchy, we obtain a "two-dimensional" infinite hierarchy. Figure 5.2 gives an illustration of the STABLE semilattice.

Finally, we showed that although there does not exist a minimal stable non-sequential degree of parallelism, every stable non-sequential function must be bounded below by a function in the Gustave hierarchy — proving it to be in a sense a minimal hierarchy.

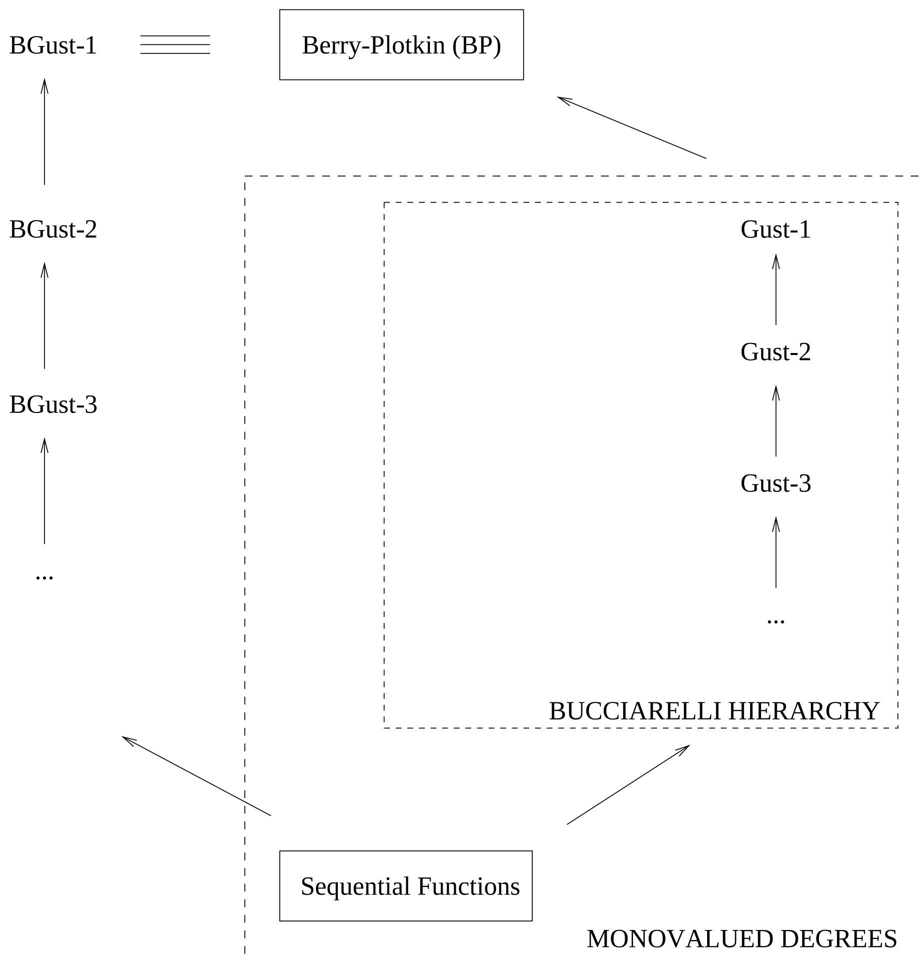

Figure 5.2: The STABLE semilattice

Chapter 6

Unstable Boolean Functions

In this chapter, we study unstable first-order boolean functions. We are mainly interested in describing the structure of those functions as a substructure of the larger CONT semilattice. We already know — via proposition 4.14 — that unstable functions are strictly more parallel than stable functions, in the sense that no stable function can dominate an unstable function in the definability order. We are interested in describing the class of stable functions a given unstable function dominates. We exhibit a hierarchy of unstable functions that dominate functions in the Bivalued-Gustave hierarchy. Along the way, we identify two new substructures of the CONT semilattice.

6.1 The Detector Function

To begin our discussion of unstable boolean functions, let us introduce the detector function (DET), which is in some sense the simplest unstable function one can find. This simplicity criterion will in fact be precised when we show that DET is in fact minimal amongst unstable functions.

DET first appeared in the context of asynchronous dataflow networks. It is defined to simply return *tt* if one of its two inputs has a value (*tt* or *ff* indifferently). In order to simplify the study of DET, we define the following equiparallel function:

Definition 6.1 *Let tt-DET be the function defined by the following trace (in matrix form):*

$$\begin{array}{|cc|c|}\hline tt & \perp & tt \\ \perp & tt & tt \\ \hline\end{array}$$

It is easy to show that DET and *tt*-DET are equiparallel.

Proposition 6.2 *tt-DET≡DET.*

Proof: Consider the following PCF-terms:

$$\begin{aligned} F_1 &= \lambda x.\text{if } x \text{ then } x \text{ else } \perp \text{ fi} \\ M_1 &= \lambda f \lambda x_1 x_2. f(F_1 x_1, F_1 x_2) \\ F_2 &= \lambda x.\text{if } (x \vee \neg x) \text{ then } tt \text{ else } \perp \text{ fi} \\ M_2 &= \lambda f \lambda x_1 x_2. f(F_2 x_1, F_2 x_2) \end{aligned}$$

It is easy to see that

$$\begin{aligned} tt\text{-DET} &= [\![M_1]\!] \text{DET} \\ \text{DET} &= [\![M_2]\!]\, tt\text{-DET}. \end{aligned}$$

□

We can further characterize *tt*-DET (and hence DET) in terms of its p-level. Recall that by proposition 4.14 an unstable function must have a p-level of the form $(i, 1)$, and by proposition 5.11 a monovalued function must have a p-level of the form (∞, j). Hence, *tt*-DET must have a p-level of $(\infty, 1)$.

6.2 The Structure of Monovalued Functions

DET was the last function we needed to complete our study of monovalued functions. Let MONO be the subposet of CONT composed of all monovalued degrees of parallelism. Just like STABLE, MONO is a sup-semilattice with a top and bottom element. The bottom element of MONO is just the degree of all sequential functions. By proposition 5.9, least upperbounds exist for finite subsets. We need only verify that there is a top element in the poset. The following proposition settles the question:

Proposition 6.3 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a monovalued continuous function. Then $f \preceq$DET.*

Proof: Without loss of generality, assume f always returns *tt* (if not, consider $neg(f)$ which is equiparallel to f). Let N-*tt*-DET be the function of arity N that returns *tt* if one of its arguments is *tt*. It is not hard to show that for all N, N-*tt*-DET$\preceq$*tt*-DET. Let $n = |tr(f)|$. Consider the following PCF-term:

$$M = \lambda p \lambda x_1 \dots x_k . p(t_1(x_1, \dots, x_k), \dots, t_n(x_1, \dots, x_k))$$

where t_j is a term checking if its arguments agree with the j$^{\text{th}}$ element of $\pi_1(tr(f))$ — and returning *tt* if they do and blocking if they don't. For example, for the Gustave function GUST, the terms look like:

$$\begin{aligned} t_1 &= \lambda x_1 x_2 x_3 . (x_2 \wedge \neg x_3) \\ t_2 &= \lambda x_1 x_2 x_3 . (x_1 \wedge \neg x_2) \\ t_3 &= \lambda x_1 x_2 x_3 . (x_3 \wedge \neg x_1) \end{aligned}$$

It is easy to see that $f = [\![M]\!] n$-*tt*-DET, and since n-*tt*-DET$\preceq$*tt*-DET, $f \preceq$*tt*-DET.

□

In the next section, we shall fully characterize the top degree of the MONO semilattice (proposition 6.7).

6.3 The Structure of Unstable Functions

An *unstable degree of parallelism* is an equivalence class of the $\equiv$ relation, where no function in the equivalence class is stable. By proposition 5.4, if a degree of parallelism is not stable, it must be unstable.

We look at the subposet of unstable degrees of parallelism, UNSTABLE. Like CONT and STABLE, UNSTABLE is a sup-semilattice with a top element and a bottom element. The top element of UNSTABLE is just the degree of POR (which is maximal amongst all continuous functions). We need only verify that there is a bottom element, and that least upperbounds exist.

Proposition 6.4 *Given $f : \mathcal{B}^k \rightarrow \mathcal{B}$ an unstable continuous function, tt-DET$\preceq f$.*[1]

Proof: This is an application of corollary 3.11. Since f is unstable, there must exists $A \subseteq \pi_1(tr(f))$ with A coherent and $|A| = 2$. Define a function

$$g : tr(tt\text{-DET}) \rightarrow tr(f)$$

with the only constraint that each element of the trace of *tt*-DET goes to a distinct element of the trace of f corresponding to the subset A. It is easy to see that all the conditions of corollary 3.11 are met, hence *tt*-DET$\preceq f$. □

As for least upperbounds, we need only verify that the following proposition holds:

Proposition 6.5 *Given $f : \mathcal{B}^k \rightarrow \mathcal{B}$ and $g : \mathcal{B}^{k'} \rightarrow \mathcal{B}$ unstable continuous functions. Then $f + g$ is unstable.*

In fact, unstability is a so-called *contagious* property of functions under the least upperbound operation, namely:

[1] *This result was first proved by Rabinovich [Rab96] in the context of asynchronous dataflow networks.*

Proposition 6.6 *Given $f : \mathcal{B}^k \to \mathcal{B}$ an unstable continuous function, $g : \mathcal{B}^{k'} \to \mathcal{B}$ a continuous function. Then $f + g$ is unstable.*

Proof: Given f an unstable continuous function. Then it must have a p-level of the form $(i_f, 1)$ (proposition 4.10 and 4.14). The function g is continuous, so it must have a p-level of the form (i_g, j_g) with $j_g \geq 1$. By proposition 4.4, $f + g$ must have a p-level of the form $(\min(i_f, i_g), \min(1, j_g))$. But $\min(1, j_g) = 1$, so that $f + g$ must be unstable (proposition 4.14). □

We are now ready to characterize the top degree of the MONO semilattice:

Proposition 6.7 *$[DET]$ is the only monovalued unstable degree of parallelism.*

Proof: Given f an unstable and monovalued boolean function. By minimality of DET DET$\preceq f$. By proposition 6.3 and f monovalued, $f \preceq$DET. Hence $f \equiv$DET. □

In other words, any unstable monovalued function must be in the top degree of MONO.

6.4 The Parallel OR Hierarchy

We are now ready to present a first hierarchy of unstable functions. This hierarchy, derived from POR, turns out to have an interesting property regarding its relationship to certain stable functions, as we shall see in section 6.6.

Definition 6.8 *Let $POR_i : \mathcal{B}^i \to \mathcal{B}$ $(i \geq 2)$ be defined by the following trace (in matrix form):*

tt	*tt*	*tt*	$\cdots$	*tt*	$\perp$	*tt*
tt	*tt*	*tt*	$\cdots$	$\perp$	*tt*	*tt*
	$\vdots$					$\vdots$
tt	*tt*	$\perp$	$\cdots$	*tt*	*tt*	*tt*
tt	$\perp$	*tt*	$\cdots$	*tt*	*tt*	*tt*
$\perp$	*tt*	*tt*	$\cdots$	*tt*	*tt*	*tt*
ff	*ff*	*ff*	$\cdots$	*ff*	*ff*	*ff*

Note that POR_2 is just POR. Intuitively, POR_i takes i inputs and returns *tt* if at least $i-1$ are *tt*, *ff* if all are *ff*.

These functions actually span the whole range of allowable p-levels for unstable functions, as the next proposition shows:

Proposition 6.9 *POR_i has a p-level of* $(i, 1)$.

Proof: Since POR_i is continuous and unstable, it must have a p-level of the form $(j, 1)$ for some $j \geq 2$ (propositions 4.10 and 4.14).

By inspection, we see that the only bivalued coherent subset of $\pi_1(tr(\mathrm{POR}_i))$ is $\pi_1(tr(\mathrm{POR}_i))$ itself. Hence, $bcc(f) = i + 1$ and by proposition 4.8, $j \geq bcc(f) - 1 \geq i$. $\square$

Let us now show that these functions in fact form a hierarchy:

Proposition 6.10 $POR_{i+1} \preceq POR_i$.

Proof: Consider the following PCF-term:

$$M = \lambda f.\lambda x_1 \ldots x_{i+1}.ALLEQ(t_1(x_1, \ldots, x_{i+1}), \ldots, t_{i+1}(x_1, \ldots, x_{i+1}))$$

where

$$ALLEQ = \lambda x_1 \ldots x_{i+1}.\text{if } (x_1 = \ldots = x_{i+1}) \text{ then } x_1 \text{ else } \perp \text{ fi}$$

which returns the value v if and only if all the arguments have the value v.

Each t_j is an application of POR_i to a subset of i inputs out of the $i+1$ possible inputs. Since $\begin{pmatrix} i+1 \\ i \end{pmatrix} = i+1$, there are $i+1$ such terms. We claim this term is such that $\mathrm{POR}_{i+1} = [\![M]\!] \mathrm{POR}_i$.

1. The t_j functions all return *tt* iff at least i *tt*'s appear in their arguments
   (a) (at least i *tt*'s) Each subset of size i has at least $i+1$ *tt*'s, so each t_j function returns *tt*.
   (b) (less then i *tt*'s) There exists one subset of size i with less than $i-1$ *tt*'s, so the corresponding t_j function returns $\perp$.
2. The t_j functions all return *ff* iff all inputs are *ff*.
   (a) (all *ff*'s) Every t_j returns *ff*.
   (b) (not all *ff*'s) There exists a subset of size i with not all inputs being *ff*. The corresponding t_j does not return *ff*.

$\square$

Proposition 6.11 *$POR_i \not\preceq POR_{i+1}$.*

Proof: By proposition 4.5, since POR_i has a p-level of $(i, 1)$ and POR_{i+1} has a p-level of $(i+1, 1)$, we get $\mathrm{POR}_i \not\preceq \mathrm{POR}_{i+1}$. $\square$

It is quite evident (by looking again at their respective p-levels) that none of the functions in the Parallel OR hierarchy is expressible from DET. Moreover, this hierarchy shows that there is no minimal bivalued unstable function — by an argument similar to the one used in proposition 5.16.

6.5 Stable-Dominating Functions

It is clear that unstability is strictly more powerful than stability, in the sense that no stable function can implement any unstable function (as one sees from the p-level characterization of stable and unstable functions), and one could think that an unstable function should be able to implement any stable function. However, this is clearly not true: simply consider the detector function, which can implement only functions in monovalued stable degrees of parallelism. In fact, functions in monovalued stable degrees of parallelism are the only stable functions that can be implemented by all unstable functions, for the very reason that they are dominated by the detector function. We can try to characterize the unstable functions that can implement all the stable functions. We first start with a definition:

Definition 6.12 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be an unstable continuous function. We say f is* stable-dominating *if for any stable continuous function $g : \mathcal{B}^{k'} \to \mathcal{B}$, we have $g \preceq f$.*

Since the STABLE semilattice has a top element (BP), a sufficient condition for an unstable function f to be stable-dominating is to have $\mathrm{BP} \preceq f$. Since any stable-dominating function f must dominate both BP (by definition) and DET (since f must be unstable), we have that $\mathrm{BP} + \mathrm{DET} \preceq f$ for any stable-dominating function f.

The following proposition completely characterizes stable-dominating functions:

Proposition 6.13 *Given $f : \mathcal{B}^k \to \mathcal{B}$ an unstable continuous function. Then f is stable-dominating iff f has a p-level of $(2,1)$.*

Proof: ($\Rightarrow$) Assume f is stable-dominating. Then by previous argument, $\mathrm{BP} + \mathrm{DET} \preceq f$. Since BP has p-level $(2,2)$ and DET has p-level $(\infty, 1)$, $\mathrm{BP} + \mathrm{DET}$ has p-level $(2,1)$ by proposition 4.4. Assume f does not have a p-level of $(2,1)$. By proposition 4.10, f must have a p-level of (i,j) with $i \geq 2$, $j \geq 1$ and $i \neq 2$ or $j \neq 1$. But by proposition 4.5, we get that $\mathrm{BP} + \mathrm{DET} \not\preceq f$, a contradiction.

($\Leftarrow$) Given f with p-level $(2,1)$. By proposition 4.14, f is unstable. We need only check that $\mathrm{BP} \preceq f$. By proposition 4.8, $bcc(f) = 3$. Let A be the subset of

$\pi_1(tr(f))$ of size 3. Assume without loss of generality that A has one element returning tt and two elements returning ff (if not, consider $neg(f)$ which is equiparallel to f). Define a function $g : tr(\mathrm{BP}) \to tr(f)$ sending the element of the trace of BP returning tt to the element of A returning tt and the elements of the trace of BP returning ff to the elements of A returning ff. It is easy to see that all the conditions of corollary 3.11 hold, and hence we have $\mathrm{BP} \preceq f$. So f is stable-dominating. $\square$

Define a *stable-dominating degree of parallelism* to be a degree of parallelism containing a stable-dominating function. As a consequence of the above characterization, we see that every function in the degree of parallelism must be stable-dominating. Moreover, we can show that stable-dominating degrees of parallelism themselves form a sup-subsemilattice of CONT called SDOM, with a top element (the degree of POR) and a bottom element (the degree of BP+DET). It is easy to see that least upperbounds exist, since if f and g are stable-dominating, then f and g have p-level $(2,1)$, so $f+g$ has p-level $(2,1)$ (proposition 4.4), and hence $f+g$ must be stable-dominating by proposition 6.13.

To show that this subsemilattice is non-trivial, we exhibit a hierarchy of functions in SDOM:

Proposition 6.14 *The functions BP+POR$_i$ are stable-dominating.*

Proof: We know BP has p-level $(2,2)$, and by proposition 6.9, POR_i has p-level $(i,1)$, $i \geq 2$. By proposition 4.4, BP+POR_i has p-level $(2,1)$, and by proposition 6.13, it is stable-dominating. $\square$

Note that $\mathrm{BP} + \mathrm{POR}_2 \equiv \mathrm{POR}_2$. We show these functions form a hierarchy:

Proposition 6.15 $\forall i \geq 2$, *$BP + POR_{i+1} \preceq BP + POR_i$.*

Proof: We know $\mathrm{BP} \preceq \mathrm{BP} + \mathrm{POR}_i$ for all $i \geq 2$. Similarly, by proposition 6.10, $\mathrm{POR}_{i+1} \preceq \mathrm{POR}_i \preceq \mathrm{BP} + \mathrm{POR}_i$. Hence, by the property of least upperbounds, we get that $\mathrm{BP} + \mathrm{POR}_{i+1} \preceq \mathrm{BP} + \mathrm{POR}_i$. $\square$

Proposition 6.16 $\forall i \geq 2$, $BP + POR_i \not\leq BP + POR_{i+1}$.

Proof: Define the following sequentiality relation of arity $i+1$

$$R = S_{i+1}^{\{1,2\},\{1,2\}} \bigcap \cdots \bigcap S_{i+1}^{\{1,\ldots,i+1\},\{1,\ldots,i+1\}}$$

We claim $\mathrm{BP} + \mathrm{POR}_{i+1}$ is invariant under R, but $\mathrm{BP} + \mathrm{POR}_i$ is not.

1. ($\mathrm{BP} + \mathrm{POR}_{i+1}$ invariant) Without loss of generality, assume

$$\mathrm{POR}_{i+1}(x_1, \ldots, x_{i+1}) = (\mathrm{BP} + \mathrm{POR}_{i+1})(tt, \ldots, tt, x_1, \ldots, x_i)$$

By contradiction, assume $\mathrm{BP} + \mathrm{POR}_{i+1}$ is not invariant under R. Then there exists tuples

$$\begin{array}{rcl} \left(x_1^1, \ldots, x_{i+1}^1\right) & \in & R \\ & \vdots & \\ \left(x_1^k, \ldots, x_{i+1}^k\right) & \in & R \end{array}$$

with $k = i+1$ if $i \geq 3$ and $k = 4$ for $i = 2$. Let $y = (y_1, \ldots, y_{i+1})$, with $y_j = f(x_j^1, \ldots, x_j^k)$, and $y \notin R$. We shall derive a contradiction.

By induction on j, we show $\mathrm{BP} + \mathrm{POR}_{i+1}$ must be invariant under $S_{i+1}^{\{1,\ldots,j\},\{1,\ldots,j\}}$, for $j \leq i$. For $j = 2$, assume $\mathrm{BP} + \mathrm{POR}_{i+1}$ is not invariant under $S_{i+1}^{\{1,2\},\{1,2\}}$. But the Closure Lemma and proposition 4.10 show that this is a contradiction. Hence, either y_1 or y_2 is $\perp$ or $y_1 = y_2$.

For the induction step, assume that $\mathrm{BP} + \mathrm{POR}_{i+1}$ is not invariant under $S_{i+1}^{\{1,\ldots,j+1\},\{1,\ldots,j+1\}}$. Then there is no $\perp$ in $y_1, \ldots, y_{j+1}$, and there exists I, J with $y_I \neq y_J$. By induction hypothesis, $\mathrm{BP} + \mathrm{POR}_{i+1}$ is invariant under $S_{i+1}^{\{1,\ldots,j\},\{1,\ldots,j\}}$, so we must have $y_1 = \cdots = y_j$, and hence the only possibility is that $y_{j+1} \neq y_1$. Since no $\perp$ appears in the resulting tuple, the first tuple above must all be tt or all be $f\!f$. If it is all $f\!f$, then the columns must come from the trace of BP, but since the first j columns are linearly coherent and return the same result, this would mean that the

Egli-Milner lowerbound of the first j column has only one element, and since it is also coherent with the last column (which returns a different result), this contradicts BP being stable. Hence, the first tuple must be all tt, and the columns must come from the trace of POR_{i+1}. But the $j+1$ columns form a linearly coherent set of size less than or equal to $i+1$, and hence they cannot contain the trace element of POR_{i+1} that returns false (easy observation). So we cannot have y_{j+1} different from y_1. Contradiction.

So, we must have $\mathrm{BP}+\mathrm{POR}_{i+1}$ invariant under $S_{i+1}^{\{1,\ldots,j\},\{1,\ldots,j\}}$ for $2 \leq j \leq i+1$, in other words, $\mathrm{BP}+\mathrm{POR}_{i+1}$ must be invariant under R.

2. ($\mathrm{BP}+\mathrm{POR}_i$ not invariant) Again without loss of generality, assume

$$\mathrm{POR}_{i+1}(x_1, \ldots, x_{i+1}) = (\mathrm{BP}+\mathrm{POR}_{i+1})(tt, \ldots, tt, x_1, \ldots, x_i)$$

and consider the tuples

$$\begin{array}{rcl} (tt, \ldots, tt) & \in & R \\ & \vdots & \\ (tt, \ldots, tt) & \in & R \\ \left(x_1^1, \ldots, x_{i+1}^1\right) & \in & R \\ & \vdots & \\ \left(x_1^i, \ldots, x_{i+1}^i\right) & \in & R \end{array}$$

where $\{(tt, \ldots, tt, x_j^1, \ldots, x_j^i)\}$ is the subset of the first projection of the trace corresponding to the POR_i function. Then applying $\mathrm{BP}+\mathrm{POR}_i$ to the tuples yields a new tuple $(y_1, \ldots, y_{i+1})$, with y_j not all equal (since POR_i is bivalued), and hence $(y_1, \ldots, y_{i+1})$ is not in $S_{i+1}^{\{1,\ldots,i+1\},\{1,\ldots,i+1\}}$, and hence $(y_1, \ldots, y_{i+1})$ is not in R.

$\square$

6.6 The Parallel OR Hierarchy and Stable Functions

Stable-dominating functions are functions dominating a certain class of stable functions (in that case, all of the stable functions). It is clear that except for POR itself, no function in the Parallel OR hierarchy is stable-dominating (looking at their p-levels). However, we can still identify stable functions that are dominated by functions in the Parallel OR hierarchy. It turns out that functions in the Parallel OR hierarchy dominate functions in the Bivalued-Gustave hierarchy:

Proposition 6.17 *$\forall i$, $BGUST_i \preceq POR_{2i}$.*

Proof: This is an application of corollary 3.11. The function BGUST_i has $2i+1$ trace elements (one of them returning *ff*, the rest *tt*), and so does POR_{2i} (one of them return *ff*, the rest *tt*). Moreover, $\pi_1(tr(\mathrm{POR}_{2i}))$ is linearly coherent. So any surjective function $g : tr(\mathrm{BGUST}_i) \to tr(\mathrm{POR}_{2i})$ mapping the trace element returning *tt* to the trace element returning *tt* and any trace element returning *ff* to any trace element returning *ff* must satisfy the conditions of corollary 3.11, and hence $\mathrm{BGUST}_i \preceq \mathrm{POR}_{2i}$. □

This is the best result one can reach comparing functions in both hierarchies, as the following demonstrates:

Proposition 6.18 *$BGUST_i \not\preceq POR_{2i+1}$.*

Proof: Looking at the respective p-levels and applying proposition 4.5. □

This has consequences regarding the implementability of functions in the Composite hierarchy as well. Since $\mathrm{DET} \preceq \mathrm{POR}_i$ $\forall i \geq 2$, we have $\mathrm{GUST}_j \preceq \mathrm{POR}_i$ $\forall i \geq 2, \forall j$. Hence, by propositions 6.17 and 6.18, we have $\mathrm{BGUST}_i + \mathrm{GUST}_j \preceq \mathrm{POR}_{2i}$ and $\mathrm{BGUST}_i + \mathrm{GUST}_j \not\preceq \mathrm{POR}_{2i+1}$.

6.7 Summary

We review the structure of UNSTABLE, the poset of unstable degrees of parallelism. This poset forms a sup-subsemilattice of CONT, with a top element (the degree of Parallel OR) and a bottom element (the degree of DET). The degree of DET is in fact the only monovalued unstable degree of parallelism, characterized by containing all and only functions with p-level $(\infty, 1)$. We showed that the monovalued degrees of parallelism form themselves a sup-subsemilattice of CONT — called MONO — with a top element (the degree of DET) and a bottom element (the degree of sequential functions). Monovalued degrees of parallelism are characterized by having a p-level of (∞, i), $i \geq 1$.

We introduced a hierarchy of unstable functon, the Parallel OR hierarchy, spanning the p-level spectrum of unstable function. We then started investigating the stable functions dominated by a given unstable function. It turns out that functions in the Parallel OR hierarchy dominate functions in the Bivalued-Gustave hierarchy. We characterized unstable functions that dominate every stable functions. These stable-dominating functions have a p-level of $(2, 1)$. Moreover, they form a sup-subsemilattice of UNSTABLE (and hence of CONT) called SDOM, with a top element (the degree of Parallel OR) and a bottom element (the degree of BP+DET). We exhibited a hierarchy in SDOM, formed by the function BP+POR$_i$. Figure 6.1 gives an illustration of the UNSTABLE semilattice.

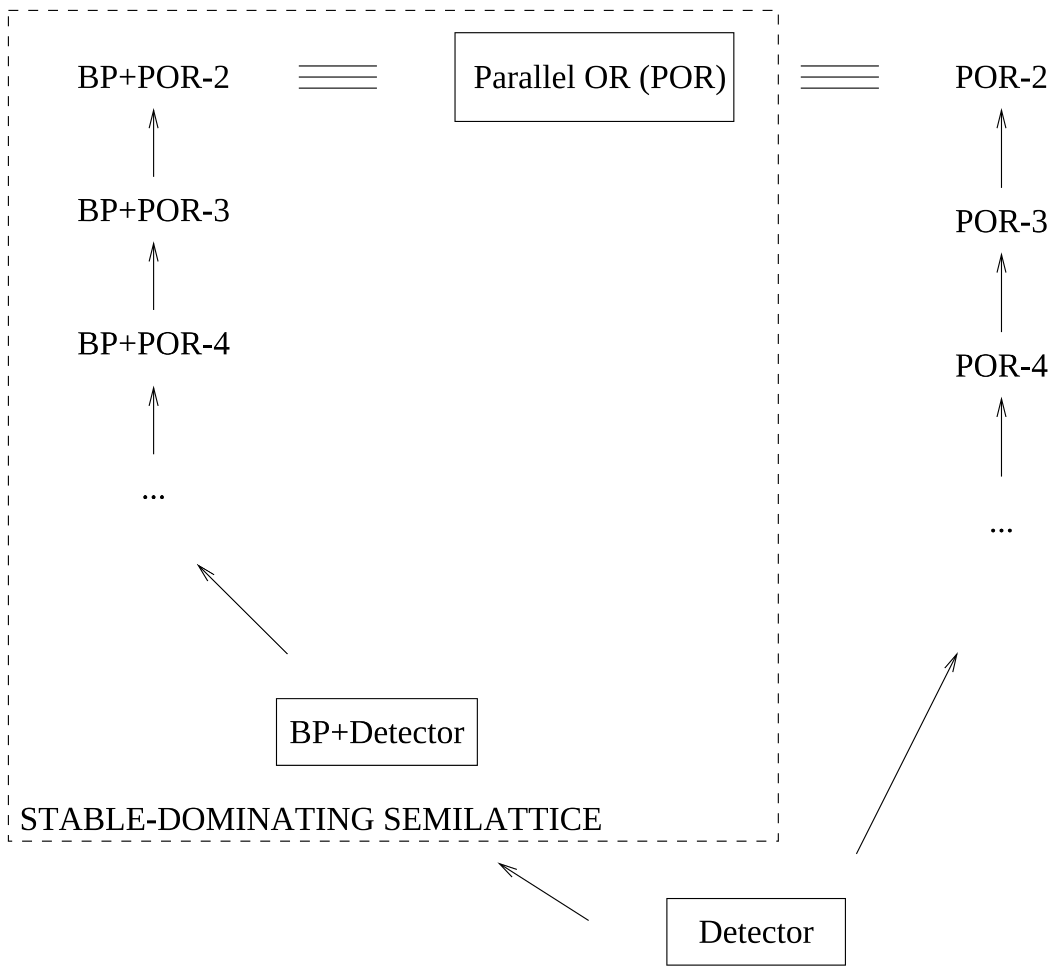

Figure 6.1: The UNSTABLE semilattice

Chapter 7

Further Work and Conclusion

To conclude this thesis, we mention possible directions of research that could yield interesting results, along with some questions that were raised during the course of the research herein presented.

7.1 The Extensional Ordering

There is some information on relative definability that one can extract from the extensional ordering (see section 2.1.2). Two important classes of functions are characterized by their extensional properties: subsequential functions and extensionally maximal functions.

7.1.1 Subsequential Functions

A boolean function f is said to be *subsequential* if there exist a sequential function g which extends f ($f \sqsubseteq g$). Subsequential functions are central to Bucciarelli and Malacaria's proof of theorem 3.10.

We will require the following lemma, proved in [BM95] (as proposition 4.0.14):

Lemma 7.1 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function. Then f is subsequential iff $bcc(f) = \infty$.*

The following proposition completely characterizes subsequential functions:

Proposition 7.2 *The function f is subsequential iff $[f]$ is monovalued.*

Proof: By lemma 7.1, f is subsequential iff $bcc(f) = \infty$. By proposition 4.8, f is subsequential iff f has p-level (∞, j) for some $j \geq 1$. By proposition 5.11, f is subsequential iff $[f]$ is monovalued. □

Hence, subsequential functions are equiparallel to monovalued functions, and are all expressible by DET (proposition 6.3).

7.1.2 Extensionally Maximal Functions

A function f is said to be *extensionally maximal* if it is maximal with respect to the extensional ordering — $\forall f'$ with $f \sqsubseteq f'$, we have $f = f'$.

The result we mention is a simple monotonicity property originally used to show (in the context of indeterminate dataflow networks expressibility) that DET could not implement POR [Pan]. We present it here in the light of PCF expressibility.

Proposition 7.3 (Maximality Principle) *Let f be an extensionally maximal continuous function, and g be a continuous function such that $f \not\preceq g$. If g' is a continuous function with $g' \sqsubseteq g$, then $f \not\preceq g'$.*

Proof: Let f be extensionally maximal and g be continuous, with $f \not\preceq g$. let g' be such that $g' \sqsubseteq g$. Assume that $f \preceq g'$. We shall derive a contradiction. By definition, $f \preceq g'$ implies that there exists a term M with $f = [\![M]\!]g'$. Since $g' \sqsubseteq g$ and $[\![M]\!]$ is continuous (and hence monotone), $f = [\![M]\!]g' \sqsubseteq [\![M]\!]g$. But f extensionally maximal implies that $f = [\![M]\!]g$. Hence $f \preceq g$, a contradiction. □

In its original form, the Maximality Principle was actually stated as a restricted version referring to subsequential functions exclusively: given f a non-sequential extensionally maximal function and g a subsequential function, then $f \not\preceq g$. This generalizes directly to the form stated above.

7.2 Hypergraphs and Algebraic Topology

One active area of investigation is directed at finding categorical counterparts of the relative definability semilattice (the CONT semilattice). Bucciarelli and Malacaria used hypergraphs, but the natural notion of morphisms between hypergraphs does not preserve relative definability, it merely reflects it (theorem 3.10 is a recasting of Bucciarelli and Malacaria's result in a non-hypergraph form). One possible venue would be to look for a more involved notion of morphism between hypergraphs. Another venue, which we shall explore at a later time, is to recast the theory of hypergraphs in an algrebraic topology context. We believe that by representing function traces via complexes and finding appropriate analogous of simplicial maps between them, we can preserve the relative definability ordering on functions. With this view, the result of Bucciarelli and Malacaria corresponds roughly to having relative definability of two functions when there exists some form of simplicial map between the complexes representing the functions. Simplicial maps are the simplest type of morphism one can design between complexes, and there is hope that by using more involved morphisms between complexes, an accurate reflection of the relative definability ordering can be reached.

7.3 Questions Raised

There are many points in the development of this thesis that require further investigation. Some of these points are mere technicalities whereas some are involved questions regarding the structure of the degrees of parallelism.

7.3.1 Purely Unstable Functions

Define a non-singleton subset $M \subseteq \mathcal{B}^n$ to be *stably coherent* if it is linearly coherent and there exists no subset $A \subseteq M$ with A linearly coherent and $|A| = 2$. Clearly a function f is stable if all linearly coherent subsets of $\pi_1(tr(f))$ are stably coherent. Define a function f to be *purely unstable* if it is unstable and $\pi_1(tr(f))$ has no stably coherent subset. Examples of purely unstable functions are provided by functions in the Parallel OR hierarchy (section 6.4). Define a degree of parallelism to be purely unstable if it contains a purely unstable function.

- One natural question to ask it whether or not all unstable degrees of parallelism are purely unstable. Clearly, if f is purely unstable, we can always find a function g which is not purely unstable yet equiparallel to f. For example, consider POR_i. In section 6.6, we show $\mathrm{BGUST}_i \preceq \mathrm{POR}_{2i}$. Hence, $\mathrm{POR}_{2i} \equiv \mathrm{POR}_{2i} + \mathrm{BGUST}_i$, and $\mathrm{POR}_{2i} + \mathrm{BGUST}_i$ is not purely unstable. The question is to ask whether the converse hold: for any unstable function f, does there exist another function g which is purely unstable and equiparallel to f?
- Related to this is to ask whether or not the function $\mathrm{BP} + \mathrm{POR}_i$ are in a purely unstable degree of parallelism (they are clearly not purely unstable).
- If not all unstable degrees of parallelism are purely unstable, are there any other purely unstable degrees of parallelism outside of [DET] and $[\mathrm{POR}_i]$ for all i?

7.3.2 Further Subsemilattices

The partition of functions into stable and unstable was motivated partly by historical reasons, and the fact that stability is an important property of functions. One way to look at that partitioning is to say that STABLE contains all functions with p-level $(\_, j)$ for $j \geq 2$, and UNSTABLE contains all functions of p-level $(\_, 1)$. An alternate partitioning of CONT would be to consider the subposets LEV-i ($i \geq 1$), where LEV-i contains all functions of p-level $(\_, i)$. Note that LEV-1=UNSTABLE.

- The first natural question is whether or not LEV-i forms a subsemilattice of CONT. It is easy to see that least upperbounds exist in LEV-i (by proposition 4.4), and it has a bottom element: DET for LEV-1 and $\mathrm{BUCC}_{(i,i)}$ for LEV-i, $i \geq 2$ — by a straightforward application of corollary 3.11. Hence the question becomes: is there a maximal function amongst functions with p-level $(\_, i)$? If we take as a starting point that BP is maximal in LEV-2, it seems natural to think that the functions BGUST_i are maximal in LEV-$2i$. Note that we know each poset LEV-i is non-trivial, since each contains the hierarchies $\mathrm{BUCC}_{(i,j)}$, $j \geq i$, and LEV-$2i$ further contains the hierarchy $\mathrm{BGUST}_j + \mathrm{GUST}_i$ (for all j).

- A related question concerns the structure of degrees of parallelism in any given p-level. We know some p-levels have a very rich structure, whereas some are trivial. For example, the p-level $(2, 1)$ contains the semilattice of stable-dominating functions, with its infinite hierarchy. In contrast, the p-level $(\infty, 1)$ of unstable monovalued degrees of parallelism contains only one degree, [DET]. One question we might want to ask is whether or not the p-levels (∞, i) for $i \geq 2$ are subsemilattices of CONT. We know each such degree has a minimal element ($\mathrm{BUCC}_{(i+1,i+1)}$) and contains an infinite hierarchy ($\mathrm{BUCC}_{(i+1,j)}$, $j \geq i$). The question becomes: is there a maximal element in (∞, i)?

7.3.3 The Hypergraph Ordering

Given two continuous boolean functions f and g. Define $f \preceq_H g$ iff there is an hypergraph morphism between the hypergraph of f and the hypergraph of g (see section 3.4). By definition of morphisms in a category, we know $\preceq_H$ is a partial order. Let $\equiv_H$ be the induced equivalence relation. Let CONT_H be the poset of $\equiv_H$-equivalence classes of continuous first-order boolean functions ordered by $\preceq_H$. We can study the embedding of CONT_H in CONT, to try and characterize the $\preceq_H$ ordering. This would give us a measure of the relative expressive power of hypergraph morphisms, which we know do not fully reflect the $\preceq$ ordering. Moreover, it could give us an idea on how to extend hypergraphs morphisms to reflect the $\preceq$ ordering faithfully.

7.3.4 Sequentiality Relations

In chapter 4, we proved two lemmas (the Reduction and Closure Lemmas) that allowed us to completely characterize the presequentiality relations under which a given function is invariant — characterization in terms of coefficients of coherence. We abstracted away from the notion of presequentiality relations by defining the p-level of a function. The next logical step in this process raises the following question: can we perform a similar abstraction for general sequentiality relations? In other words, can we define a numerical property completely characterizing sequentiality relations, and find a mechanical way to derive the numerical property from any given function? Some work has been done on the subject, but has not yet yielded a property that was in any way easier to study than sequentiality relations themselves.

Another view on sequentiality relations that could yield information on the $\preceq$ structure is the following: given a first-order continuous boolean function f, define S_f to be the set of sequentiality relations under which f is invariant. Theorem 3.7 gets recast in this context as: $f \preceq g$ iff $S_g \subseteq S_f$. We can study the structure of the poset of sets of sequentiality relations under the inclusion ordering. Two questions arise:

- Does $S_{f+g} = S_f \cap S_g$ hold?
- Given any set of sequentiality relations S, can we construct a function f such that $S_f = S$?

7.4 Conclusion

The underlying goal of the work presented in this thesis was one of exploration. We explored the poset of degrees of parallelism, in an attempt to clarify its structure, and gather clues as to its complexity. We developed some techniques to determine non-expressibility of functions, and used them to obtain most of our structural results.

These structures, by the nature of the techniques used, form in some way the skeleton of the poset of degrees of parallelism. We shall now review the results obtained: both reporting the structural information gathered throughout this research, and analyzing the techniques we focused on.

7.4.1 Structural Results

From a structural point of view, some information was already available concerning the poset of degrees of parallelism: degrees formed a sup-semilattice (that we called CONT) with [POR] as top element and the degree of sequential functions as bottom. Bucciarelli [Buc95], by exhibiting a "two-dimensional" hierarchy, further showed that the semilattice was highly non-trivial.

Our work started from that point. We first partitioned the semilattice into two disjoint posets, which themselves turned out to be sup-semilattices: the semilattice of stable functions (STABLE) with BP as a top element and the degree of stable functions as bottom, and the semilattice of unstable functions (UNSTABLE) with [POR] as top and [DET] as bottom. A third subsemilattice (MONO) was identifiedby considering degrees of monovalued functions (or equivalently, of subsequential functions), with [DET] as top element and the degree of sequential functions as bottom. This semilattice was orthogonal to the previous two and interested them both. The intersection of MONO with UNSTABLE yielded only one degree of parallelism, [DET]. The intersection of MONO and STABLE was larger, and contained the full hierarchy originally discovered by Bucciarelli.

The MONO semilattice had an interesting property: one of its subposets — the Gustave hierarchy — formed a minimal hierarchy, in the sense that any continuous function had to dominate a function in the hierarchy. The Gustave hierarchy also showed that there was no minimal non-sequential function. Modifying the functions in the Gustave hierarchy to make them bivalued produced a new hierarchy in STABLE (but not in MONO), the Bivalued-Gustave hierarchy. Taking the least upperbound of functions in both the Gustave and the Bivalued-Gustave hierarchy produced a "two-dimensional" hierarchy — the Composite hierarchy — vaguely similar to Bucciarelli's, but in STABLE (and not in MONO).

We next turned toward the study of the UNSTABLE semilattice. We identified a hierarchy of unstable functions derived from the POR function, the Parallel OR hierarchy. One consequence of the existence of this hierarchy was that there could be no minimal unstable function not equivalent to DET.

Most of the focus regarding unstable functions was directed at characterizing the stable functions dominated by a given unstable function. Among the functions studied, we determined that DET could implement all monovalued functions (and equivalents thereof), and functions in the Parallel OR hierarchy could implement corresponding functions in the Bivalued-Gustave hierarchy (and hence the Composite hierarchy). As for unstable functions that dominated all stable functions, called stable-dominating, they could be easily characterized and formed themselves a sub-semilattice of UNSTABLE (SDOM) with [POR] as top element and [BP + DET] as bottom. The existence of a hierarchy in SDOM showed that the semilattice was non-trivial.

7.4.2 Techniques Developed

The techniques we developed in this research were mostly concerned with finding a partitioning of continuous boolean functions that could yield immediate inexpressibility results for functions in different partitions. Most of the inexpressibility results in chapters 5 and 6 were proved using these techniques.

Our starting point was the theorem of Sieber relating the definability ordering and sequentiality relations (proposition 3.7). A consequence of this theorem was that one needed only exhibit a sequentiality relation under which a given function was invariant and another given function was not, to prove that the latter was not expressible from the former. We decided to examine a class of very simple sequentiality relations

— the presequentiality relations . Our motivations were two-fold: the simplicity of these relations made them easy to study and they provided a coarse separation between functions: if a presequentiality relation was sufficient (by Sieber's theorem) to determine that a function was not expressible from another, then the functions were fundamentally different. One of our goals was to determine what such a fundamental difference between functions would look like.

In order to achieve such a goal (and after deriving properties of presequentiality relations), we defined the p-level (i, j) of a function, a numerical characteristic embodying the presequentiality relations under which a function was invariant. An interesting property of p-levels was that they could be derived without ever refering to presequentiality relations: one needed only compute the coefficient of coherence and the bivalued coefficient of coherence of the function. Those two quantities could directly be computed from the trace of the function.

The practical use of p-levels was in partitioning the semilattice of degrees of parallelism into classes of functions with the same p-level. One could derive (by a direct translation of p-levels into the corresponding presequentiality relations) proposition 4.8, which provided a simple inexpressibility criterion that we used to prove most of inexpressibility results in this thesis. Moreover, the link between p-levels and coefficients of coherence answered our question regarding the fundamental difference between functions differentiated by presequentiality relations: the difference was in the minimum size of linearly coherence (bivalued or not) subsets of the first projection of the trace of the functions.

The inexpressibility result of proposition 4.5 is however the strongest result one can achieve from presequentiality relations alone. The p-level of a function tells us nothing when we are comparing it to a function with the same p-level. A lot of structural information gets lost — consider for example the functions with a p-level of $(2, 1)$, which as we saw in section 6.5 form a subsemilattice containing an infinite hierarchy.

As we hinted at in section 7.3, we are attempting a generalization of p-levels to encompass the full set of sequentiality relations . If successful, this would lead us to a function-independent counterpart to the semilattice of degrees of parallelism, which would most likely give us new insights on the fundamental structure induced by the relative definability ordering.

Appendix A

Proof of the Reduction Lemma

In this appendix, we prove the Reduction Lemma (lemma 4.1), which basically states that when considering the invariance of a function under presequentiality relations, we need only look at presequentiality relations of the form $S_m^{\{1,\ldots,m\},\{1,\ldots,m\}}$ and $S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$.

The proof uses many small technical lemmas, that we now state and prove.

The following three lemmas reflect the fact that when considering the invariance of a function under the presequentiality relation $S_n^{A,B}$, we need only consider the smallest n such that $B \subseteq \{1,\ldots,n\}$.

Lemma A.1 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. If f is invariant under $S_n^{A,B}$, then $\forall n' \leq n$ such that $B \subseteq \{1,\ldots,n'\}$, f is invariant under $S_{n'}^{A,B}$.*

Proof: By contradiction, by assume that there exist n, A, B, n' with $n' \leq n$ such that f is invariant under $S_n^{A,B}$ but not under $S_{n'}^{A,B}$.

This means there exist tuples

$$\begin{array}{rcl}(x_1^1,\ldots,x_{n'}^1) & \in & S_{n'}^{A,B} \\ & \vdots & \\ \left(x_1^k,\ldots,x_{n'}^k\right) & \in & S_{n'}^{A,B}\end{array}$$

and $(y_1, \ldots, y_{n'}) \notin S_{n'}^{A,B}$ with $y_i = f(x_i^1, \ldots, x_i^k)$.

Consider the tuples

$$\begin{array}{rcl} (x_1^1, \ldots, x_{n'}^1, \perp, \ldots, \perp) & \in & S_n^{A,B} \\ & \vdots & \\ (x_1^k, \ldots, x_{n'}^k, \perp, \ldots, \perp) & \in & S_n^{A,B} \end{array}$$

But $(y_1, \ldots, y_{n'}, \perp, \ldots, \perp) \notin S_n^{A,B}$, which contradicts the invariance of f under $S_n^{A,B}$. □

Lemma A.2 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. If f is invariant under $S_n^{A,B}$, then $\forall n' \geq n$, f is invariant under $S_{n'}^{A,B}$.*

Proof: By contradiction, assume there exist n, A, B and $n' \geq n$ such that f is invariant under $S_n^{A,B}$ but not under $S_{n'}^{A,B}$.

This means there exist tuples

$$\begin{array}{rcl} (x_1^1, \ldots, x_n^1) & \in & S_{n'}^{A,B} \\ & \vdots & \\ \left(x_1^k, \ldots, x_n^k\right) & \in & S_{n'}^{A,B} \end{array}$$

and $(y_1, \ldots, y_{n'}) \notin S_{n'}^{A,B}$ with $y_i = f(x_i^1, \ldots, x_i^k)$.

Observation shows that $(x_1, \ldots, x_{n'}) \in S_{n'}^{A,B} \Leftrightarrow (x_1, \ldots, x_n) \in S_n^{A,B}$.

Hence,

$$\begin{array}{rcl} (x_1^1, \ldots, x_n^1) & \in & S_n^{A,B} \\ & \vdots & \\ \left(x_1^k, \ldots, x_n^k\right) & \in & S_n^{A,B} \end{array}$$

but $(y_1, \ldots, y_n) \notin S_n^{A,B}$ contradicting the invariance of f under $S_n^{A,B}$. □

Lemma A.3 *Let $m(M)$ be the least n such that $M \subseteq \{1, \ldots, n\}$, and let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. The function f is invariant under $S_n^{A,B}$ iff f is invariant under $S_{m(B)}^{A,B}$.*

Proof: Immediate via previous two lemmas. □

The following lemmas clarifies the intuition that invariance under a presequentiality relation is not affected by the permutation of the columns of the tuples in the relation.

Lemma A.4 *Let $f : \mathcal{B}^k \to \mathcal{B}$ be a continuous function. Let A, B, C, D be sets with $A \subseteq B \subseteq \{1, \ldots, n\}, C \subseteq D \subseteq \{1, \ldots, n\}$. Let p be a permutation of $\{1, \ldots, n\}$ into $\{1, \ldots, n\}$ such that $p(A) = C$ and $p(B) = D$. Then f is invariant under $S_n^{A,B} \Leftrightarrow f$ is invariant under $S_n^{C,D}$.*

Proof: Let us first prove one small result that simplifies the remainder of the proof:

$$(x_1, \ldots, x_n) \in S_n^{A,B} \Leftrightarrow (x_{p^{-1}(1)}, \ldots, x_{p^{-1}(n)}) \in S_n^{C,D}.$$

Let $(x_1, \ldots, x_n) \in S_n^{A,B}$, and $y_i = x_{p^{-1}(i)}$. We show $(y_1, \ldots, y_n) \in S_n^{C,D}$. Two cases arise

1. $\exists i \in A, x_i = \perp$. In which case, let $c = p(i)$, with $c \in C$ since $i \in A$. Moreover, $y_c = x_{p^{-1}(c)} = x_{p^{-1}(p(i))} = x_i = \perp$, so $\exists j \in C, y_j = \perp$.
2. $\forall i, j \in B, x_i = x_j$. Assume $\exists i, j \in D, y_i \neq y_j$. Then $x_{p^{-1}(i)} \neq x_{p^{-1}(j)}$, hence $\exists i', j' \in B, x_{i'} \neq x_{j'}$, a contradiction. Hence $\forall i, j \in D, y_i = y_j$.

This shows $(y_1, \ldots, y_n) \in S_n^{C,D}$. The reverse direction follows by symmetry of the permutation p.

To prove the actual lemma, we first observe that we need only show one direction of the equivalence. The reverse direction follows by symmetry of the permutation p.

Given f invariant under $S_n^{A,B}$, consider any tuples

$$\begin{array}{rcl} (x_1^1, \ldots, x_n^1) & \in & S_n^{A,B} \\ & \vdots & \\ \left(x_1^k, \ldots, x_n^k\right) & \in & S_n^{A,B} \end{array}$$

Let $y_i = f(x_i^1, \ldots, x_i^k)$. We need to show $(y_1, \ldots, y_n) \in S_n^{A,B}$.

By the fact previously noticed, each tuple $\left(x_1^j, \ldots, x_n^j\right)$ is also in $S_n^{C,D}$. Since f is invariant under $S_n^{C,D}$, $(y_1, \ldots, y_n) \in S_n^{C,D}$. And again by the previous lemma, this implies that $(y_1, \ldots, y_n) \in S_n^{A,B}$. $\square$

Lemma A.5 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function, f is invariant under $S_n^{A,B}$ $\Leftrightarrow f$ is invariant under $S_n^{\{1,\ldots,|A|\},\{1,\ldots,|B|\}}$.*

Proof: By the previous lemma, we need only show that there exists a permutation p of $\{1, \ldots, n\}$ such that $p(A) = \{1, \ldots, |A|\}, p(B) = \{1, \ldots, |B|\}$.

Let

$$p_A : A \to \{1, \ldots, |A|\}$$
$$p_B : B \backslash A \to \{|A| + 1, \ldots, |B|\}$$
$$p_R : \{1, \ldots, n\} \backslash B \to \{|B| + 1, \ldots, n\}$$

be canonical bijections.

Define $p : \{1, \ldots, n\} \to \{1, \ldots, n\}$ as

$$p(i) = \begin{cases} p_A(i) & \text{if } i \in A \\ p_B(i) & \text{if } i \in B \backslash A \\ p_R(i) & \text{if } i \in \{1, \ldots, n\} \backslash B \end{cases}$$

This is a bijection , and obviously, $p(A) = \{1, \ldots, |A|\}, p(B) = \{1, \ldots, |B|\}$.

$\square$

The following three lemmas show that if f is invariant under some presequentiality relation $S_n^{A,B}$ with $A \subset B$, then f will be invariant under $S_n^{A,B'}$ for any B' such that $A \subset B'$.

Lemma A.6 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function. If f is invariant under $S_n^{A,B}$, then f is invariant under $S_n^{A,B'}$ for all $B' \subseteq B$.*

Proof: By contradiction, assume there exist n, A, B, B' with $B' \subseteq B$ such that f is invariant under $S_n^{A,B}$ but not under $S_n^{A,B'}$. Then there exist tuples

$$\begin{array}{rcl} (x_1^1, \ldots, x_n^1) & \in & S_n^{A,B'} \\ & \vdots & \\ \left(x_1^k, \ldots, x_n^k\right) & \in & S_n^{A,B'} \end{array}$$

such that $(y_1, \ldots, y_n) \notin S_n^{A,B'}$ with $y_i = f(x_i^1, \ldots, x_i^k)$.

Fix an arbitrary $k \in A$. Consider the following tuples: $\left(z_1^j, \ldots, z_n^j\right)$ for $1 \leq j \leq k$, with

$$z_i^j = \begin{cases} x_i^j & \text{if } i \in B' \\ x_k^j & \text{if } i \in B \backslash B' \\ \perp & \text{otherwise} \end{cases}$$

We first verify that these tuples are in $S_n^{A,B}$. For each $j, 1 \leq j \leq k$, consider the original tuple $\left(x_1^j, \ldots, x_n^j\right) \in S_n^{A,B'}$. In other words, either

1. $\exists i \in A,\ x_i^j = \perp$, and for that $i \in A$, we have $z_i^j = x_i^j = \perp$. Hence $\left(z_1^j, \ldots, z_n^j\right) \in S_n^{A,B}$.
2. $\forall i \in A,\ x_i^j \neq \perp$, and $\forall i, i', x_i^j = x_{i'}^j$. Hence, $\forall i, i', z_i^j = z_{i'}^j$. Moreover, $\forall i \in B \backslash B', z_i^j = x_k^j$ for $k \in A \subseteq B'$. Hence, $\forall i, i' \in B, z_i^j = z_{i'}^j$ and the tuple $\left(z_1^j, \ldots, z_n^j\right) \in S_n^{A,B}$.

By the above construction, we see that $\forall i \in B', f(z_i^1, \ldots, z_i^k) = y_i$.

Since $(y_1, \ldots, y_n) \notin S_n^{A,B'}$, we have $\forall i \in A, y_i \neq \perp$ and $\exists i, j \in B', y_i \ /y_j$. This implies that $\forall i \in A, f(z_i^1, \ldots, z_i^k) \neq \perp$ and $\exists i, j \in B' \subseteq B, f(z_i^1, \ldots, z_i^k) \neq f(z_j^1, \ldots, z_j^k)$. In other words, f is not invariant under $S_n^{A,B}$, an absurdity. $\square$

Lemma A.7 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function. If f is invariant under $S_n^{A,B}$, $|B \backslash A| = 1$, then for any B' such that $B \subseteq B'$, f is invariant under $S_n^{A,B'}$.*

Proof: By lemma A.3 and lemma A.5, we do this in two steps. For any m,

1. f invariant under $S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}} \Rightarrow f$ invariant under $S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}}$.
2. f invariant under $S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}} \Rightarrow f$ invariant under $S_n^{\{1,\ldots,m\},\{1,\ldots,n\}}$ for any $n \geq m+2$.

(1) By contradiction, assume that for some m, f is invariant under the pre-sequentiality relation $S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$ but not under $S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}}$. Then there are tuples

$$\begin{array}{rcl} \left(x_1^1, \ldots, x_{m+2}^1\right) & \in & S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}} \\ & \vdots & \\ \left(x_1^k, \ldots, x_{m+2}^k\right) & \in & S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}} \end{array}$$

but $(y_1, \ldots, y_{m+2}) \notin S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}}$, with $y_i = f(x_i^1, \ldots, x_i^k)$. Hence, $\forall i \leq m$, $y_i \neq \perp$ and $\exists I, J$ such that $y_I \neq y_J$.

We consider 3 cases:

1. $(I \leq m)$ Consider the following tuples

   $$\begin{array}{rcl} \left(x_1^1, \ldots, x_m^1, x_J^1\right) & \in & S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}} \\ & \vdots & \\ \left(x_1^k, \ldots, x_m^k, x_J^k\right) & \in & S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}} \end{array}$$

   and $(y_1, \ldots, y_m, y_J) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$ by assumption. Hence, either

   (a) $\exists i \leq m$ such that $y_i = \perp$ (contradiction)

   (b) $y_I = y_J$ (contradiction).

2. $(J \leq m)$ Same as argument as first case above, interchanging I and J.
3. $(I, J > m)$ We further consider 3 subcases.

(a) $(y_I = \bot)$. Consider the following tuples

$$\left(x_1^1, \ldots, x_m^1, x_I^1\right) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$$
$$\vdots$$
$$\left(x_1^k, \ldots, x_m^k, x_I^k\right) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$$

and $(y_1, \ldots, y_m, y_I) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$ by assumption on the invariance of f. So either

i. $\exists i \leq m$ such that $y_i = \bot$ (contradiction)

ii. $y_I = y_i$ for all $i \leq m$ (contradiction)

(b) $(y_J = \bot)$ Same argument as case above, with J used instead of I.

(c) $(y_I, y_J \neq \bot)$ Either all $y_i, \forall i \leq m$ are equal and let $z^j = x_I^j \text{or} x_J^j$, the one different from all y_i, or let $z^j = x_I^j$ (recall $y_I \neq y_J$.

Consider the tuples

$$\left(x_1^1, \ldots, x_m^1, z^1\right) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$$
$$\vdots$$
$$\left(x_1^k, \ldots, x_m^k, z^k\right) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$$

and $\left(y_1, \ldots, y_m, f(z^1, \ldots, z^k)\right) \in S_{m+1}^{\{1,\ldots,m\},\{1,\ldots,m+1\}}$, by assumption on invariance of f. So either

i. $\exists i \leq m$ such that $y_i = \bot$ (contradiction)

ii. $f(z^1, \ldots, z^k) = y_i$ for all $i \leq m$, but this is absurb by construction of z^j

(2) By contradiction, assume that there exists m such that f is invariant under $S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}}$ but there exists $n \geq m+2$ such that f is not invariant under $S_n^{\{1,\ldots,m\},\{1,\ldots,n\}}$. Then there exist tuples

$$(x_1^1, \ldots, x_n^1) \in S_n^{\{1,\ldots,m\},\{1,\ldots,n\}}$$
$$\vdots$$
$$\left(x_1^k, \ldots, x_n^k\right) \in S_n^{\{1,\ldots,m\},\{1,\ldots,n\}}$$

but $(y_1, \ldots, y_n) \notin S_n^{\{1,\ldots,m\},\{1,\ldots,n\}}$ with $y_i = f(x_i^1, \ldots, x_i^k)$. Hence, $\forall i \leq m$, $y_i \neq \perp$ and $\exists I, J$ such that $y_I \neq y_J$. Consider the tuples

$$\begin{aligned} \left(x_1^1, \ldots, x_m^1, x_I^1, x_J^1\right) &\in S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}} \\ &\vdots \\ \left(x_1^k, \ldots, x_m^k, x_I^k, x_J^k\right) &\in S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}} \end{aligned}$$

and $(y_1, \ldots, y_m, y_I, y_J) \in S_{m+2}^{\{1,\ldots,m\},\{1,\ldots,m+2\}}$ by assumption on the invariance of f. But this means that either

1. $\exists i \leq m$ such that $y_i = \perp$ (contradiction)
2. $y_I = y_J$ (contradiction)

$\square$

Lemma A.8 *Given $f : \mathcal{B}^k \to \mathcal{B}$ a continuous function. Then f is invariant under $S_n^{A,B}$, $|B \backslash A| = 1 \Leftrightarrow f$ is invariant under $S_n^{A,B'}$ for any B' such that $B \subseteq B'$.*

Proof: Immediate via the previous two lemmas. $\square$

With these lemmas in hand, the Reduction Lemma is immediate.

Proof: (Reduction Lemma 4.1)

1. $(A = B)$ By lemma A.5, we have that f is invariant under $S_n^{A,A} \Leftrightarrow f$ is invariant under $S_n^{\{1,\ldots,|A|\},\{1,\ldots,|A|\}}$ and by lemma A.3, f is invariant under $S_n^{\{1,\ldots,|A|\},\{1,\ldots,|A|\}} \Leftrightarrow f$ is invariant under $S_{|A|}^{\{1,\ldots,|A|\},\{1,\ldots,|A|\}}$.
2. $(A \subset B)$ By lemma A.5, f is invariant under $S_n^{A,B} \Leftrightarrow f$ is invariant under $S_n^{\{1,\ldots,|A|\},\{1,\ldots,|B|\}}$. By lemma A.8, f is invariant under $S_n^{\{1,\ldots,|A|\},\{1,\ldots,|B|\}}$ $\Leftrightarrow f$ is invariant under $S_n^{\{1,\ldots,|A|\},\{1,\ldots,|A|+1\}}$, and by lemma A.3, this happens $\Leftrightarrow f$ is invariant under $S_{|A|+1}^{\{1,\ldots,|A|\},\{1,\ldots,|A|+1\}}$.

$\square$

Bibliography

[BE91] Antonio Bucciarelli and Thomas Erhard. Sequentiality and Strong Stability. In *Proc. Sixth Annual IEEE Symposium on Logic in Computer Science*, 1991.

[BE94] Antonio Bucciarelli and Thomas Erhard. Sequentiality in an Extensional Framework. *Information and Computation*, 110(2), 1994.

[Ber76] Gérard Berry. Bottom-up Computations of Recursive Programs. *R.A.I.R.O. Informatique Théorique*, 10(3):47–82, 1976.

[BM95] Antonio Bucciarelli and Pasquale Malacaria. Relative Definability of Boolean Functions via Hypergraphs. To appear, 1995.

[Buc95] Antonio Bucciarelli. Degrees of Parallelism in the Continuous Type Hierarchy. Technical Report LIENS-DMI 95/16, Ecole Normale Supérieure de Paris, 1995.

[Cur93] P.-L. Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming*. Birkhäuser, revised edition edition, 1993.

[Gev95] Shai Geva. *A Study of Higher-order Sequential Computation*. PhD thesis, Carnegie-Mellon University, 1995.

[Gun92] C.A. Gunter. *Semanticsof Programming Languages: Structures and Techniques*. MIT Press, 1992.

[KP78] G. Kahn and G.D. Plotkin. Domaines Concrets. IRIA-LABORIA, Rapport 336, 1978.

[Mil77] Robin Milner. Fully Abstract Models of Typed Lambda-Calculi. *Theoretical Computer Science*, 4:1–22, 1977.

[Pan] Prakash Panangaden. Private communication.

[Plo77] Gordon Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–256, 1977.

[Plo80] Gordon Plotkin. Lambda-Definability in the Full Type Hierarchy. In J. Hindley J. Seldin, editor, *To H.B.Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 363–374. Academic Press, 1980.

[Rab96] Alexander Rabinovich. Modularity and Expressibility for Nets of Relations. To appear, 1996.

[Saz76] V. Yu Sazonov. Degrees of Parallelism in Computations. In *Proc. Conference on Mathematical Foundations of Computer Science*, number 45 in Lecture Notes in Computer Science, 1976.

[Sie92] Kurt Sieber. Reasoning about Sequential Functions via Logical Relations. In M. Fourman, P. Johnstone, A. Pitts eds, editor, *Proc. LMS Conference on the Applications of Categories to Computer Science*. Cambridge University Press, 1992.

[Sto94] Allen Stoughton. Mechanizing Logical Relations. In *Proc. Ninth International Conference on Mathematical Foundations of Programming Semantics*, number 802 in Lecture Notes in Computer Science, 1994.

[Tra73] Mark B. Trakhtenbrot. On Interpreted Functions in Program Schemes. In *Sistemnoe i teoretičeskoe programmirovanie*, pages 188–211. Novosibirsk, 1973. (in Russian).

[Tra75] Mark B. Trakhtenbrot. On Representation of Sequential and Parallel Functions. In *Proc. Fourth Symp. on Mathematical Foundations of Computer Science*, number 32 in Lecture Notes in Computer Science. Springer, 1975.

[Vui73] J. Vuillemin. *Proof Techniques for Recursive Programs.* PhD thesis, Stanford University, 1973.