Semantic Barbs and Biorthogonality

Julian Rathke, Vladimiro Sassone, and Paweł Sobociński*

ECS, University of Southampton

Abstract. We use the framework of biorthogonality to introduce a novel semantic definition of the concept of barb (basic observable) for process calculi. We develop a uniform basic theory of barbs and demonstrate its robustness by showing that it gives rise to the correct observables in specific process calculi which model synchronous, asynchronous and broadcast communication regimes.

1 Introduction

Labelled transition systems and structural operational semantics [17] have been the idiomatic approach to the semantics of communicating concurrent systems for many years [13,9]. Such semantics naturally yield many notions of equivalence based on traces or bisimulations. As communication patterns in these calculi grew more complex, there was a need to justify the ad-hoc labelled transition semantics being provided with respect to simpler, more canonical equivalences.

Following the seminal contribution of Berry and Boudol [3], calculi began to be equipped with a reduction relation meaning that widely accepted techniques from the theory of lambda-calculi, such as the definition of a contextually defined reduction congruence, were able to be studied in the setting of calculi for concurrency and mobility. However, even in the setting of CCS [13], reduction congruence is coarser than standard bisimilarity; this led Milner and Sangiorgi to introduce the concept of a basic observable of a process, which came to be dubbed a *barb* [14]. Together with a reduction semantics, barbs yield a canonical notion of process equivalence for most modern calculi.

Barbs are notable in that they are perhaps the most well-known concept of formal concurrent semantics which, despite being frequently used, do *not* actually have a general formal definition. In [14], a barb is understood simply to be a predicate on processes which captures the intuitive notion of basic observable. In many other settings specific barbs are precisely defined but no account is taken of whether these definitions are appropriate. For example, in the calculus, CCS, the choice of barb as being the ability to synchronise on a given name is uncontentious. However, even moving to such a simple setting as an asynchronous version of CCS leads to questions about the suitability of certain synchronisations as barbs. In this setting, it is accepted [2] for instance that the ability to synchronise as a receive action is *not* suitable as a barb. To date though, there is no formal definition which justifies this.

* Research partially supported by EPSRC grant EP/D066565/1.

H. Seidl (Ed.): FOSSACS 2007, LNCS 4423, pp. 302–316, 2007.

The goal of this paper is to provide an abstract, semantic definition of what it means to be a barb. We aim to make the definition as general as possible, whilst ensuring that we encompass the intuitive definitions of barbs in various well-known examples. The benefits of this approach will be that in complex languages or formal systems of communication, we will have a canonical definition with which to identify the basic patterns of interaction. The need for more thorough foundational insights is becoming increasingly important with the advent of complex communication patterns in calculi for mobility [5, 19] and biological modelling [4].

A defining feature of our approach is that we obtain a notion of *observable* for our calculi based solely on their reduction semantics. Moreover, the observables are required to be suitably minimal so as to allow them to be considered basic. In this sense, our work is related to Leifer and Milner's attempts to obtain a labelled semantics from a reduction semantics [11], however we focus on static capability for interaction between agents and imbue the framework with a notion of 'successful' interaction which is key to understanding the nature of observability.

Our definition of barb is based around the notion of a closure operator, which, given a process (or a set of processes), will construct the set of all processes which offer the immediate interactions of that process (or set of processes). For example, if we take the process: $a!p_0 \parallel b?q_0$, (where $a!$ and $b?$ denote, respectively, the ability to output on a channel a and the ability to receive on a channel b) then its closure is the set of all processes of the form

$$a!p \parallel b?q \parallel r \quad \text{for any} \quad p, q, r$$

as these are the processes which offer *both* an $a!$ and $b?$ interaction. This closed set is considered to represent the abstract concept of a $a!$ and $b?$ interaction. Similarly, the closure of $\{a!p_0, b?q_0\}$ is the set of all processes of the form

$$a!p \parallel r \quad \text{or} \quad b?q \parallel r$$

that is, all processes which offer an $a!$ *or* a $b?$ interaction. Similar ideas have been used both in logic and computer science, for example in Girard's phase semantics of linear logic [8], Krivine's realisability [6], Pitts' $\top\top$-closure [16] and formal concept theory [20]. In particular, we shall show how all of the aforementioned examples fit within a basic general framework.

To achieve a decent account of minimality in the barbs we need to consider the closed sets that represent the least possible interactions. In the examples above, we should not consider either of these minimal as they are made up of combinations of distinct interaction capabilities in CCS. To capture this we make use of the notion of *irreducibility* from algebraic geometry, see [18] for an introductory account. Indeed, we only consider those closed sets of processes which cannot be decomposed into a combination of two separate smaller closed sets. This certainly rules out the closure of $\{a!p_0, b?q_0\}$ from being a barb, but we also wish to rule out the closure of $a!p_0 \parallel b?q_0$. This is done in a similar, and dual, way of considering the closed sets of contexts which this process can successfully interact with and demanding that this closed set of contexts cannot be decomposed either. In this example, the contexts which successfully interact are of the form

$$a?p \parallel r \quad \text{or} \quad b!q \parallel r$$

and we see that these are not minimal in the sense suggested.

This use of irreducibility in both the closed sets of processes and the contexts for them is, to our knowledge, novel and is a main contribution of the paper. The remainder of the paper is arranged so that we present our general notion of interaction frameworks and biorthogonality along with familiar examples of such frameworks. We then tailor the setting to specifically allow us to study interaction in processes and we introduce three paradigmatic example process calculi. The definition of irreducibility and the main definition of barb is then given. After this we study properties of barbs in example languages, including full process languages such as π-calculus.

2 Biorthogonality

The notion of *biorthogonality* is the device underpinning our entire approach. It is at the same time a conceptual and a technical tool; its strength resides in the simplicity and elegance with which it captures mathematically –amongst other things– the notion of test as relationship between processes and contexts. We will use it to understand the concept and the role of 'observation' at a foundational level. We shall omit the proofs in this section because all of the results are basic and well-known, even if we are unaware of previous work which collected all of the examples listed at the conclusion of this section as instances of a single framework.

We assume the following basic ingredients:

– sets $\mathbf{T}$, $\mathbf{\Gamma}$ and $\mathbf{\Pi}$ which we shall self-evidently refer to as *terms*, *contexts* and *processes*, respectively;
– a function $@\colon\ \mathbf{T} \times \mathbf{\Gamma} \to \mathbf{\Pi}$, representing the insertion of a term in a context to yield a process;
– a subset $\perp\ \subseteq \mathbf{\Pi}$ of *successful* processes, which we think of as a unary predicate.

Informally, it is we think of $t \mathbin{@} \gamma \in \perp$ as stating that t passes the test (represented by context) γ, and vice versa; we shall say that t is successful for γ and that γ is successful for t. We shall usually write the shorthand form $t \perp \gamma$ to mean $t \mathbin{@} \gamma \in\perp$. From these basic notions we derive the maps:

$$\begin{array}{l} (-)^{\perp} : \mathcal{P}(\mathbf{T}) \to \mathcal{P}(\mathbf{\Gamma}) \\ T \mapsto \{\, \gamma \in \mathbf{\Gamma} \ :\ \forall t \in T.\ t \perp \gamma \,\} \end{array} \quad \text{and} \quad \begin{array}{l} (-)^{\perp} : \mathcal{P}(\mathbf{\Gamma}) \to \mathcal{P}(\mathbf{T}) \\ \Gamma \mapsto \{\, t \in \mathbf{T} \ :\ \forall \gamma \in \Gamma.\ t \perp \gamma \,\}. \end{array}$$

Thus, given a set T of terms, $T^{\perp}$ is the set of contexts which are successful for every $t \in T$, and similarly, given a set Γ of contexts, $\Gamma^{\perp}$ is the set of terms which are successful for every $\gamma \in \Gamma$. The reader is asked to tolerate the abuse of notation, justified by the symmetry of the definitions, as the type of the argument to $(-)^{\perp}$ will always be made clear.

The following lemma proves some basic properties of the functions, and strengthens the intuition of their combination $(-)^{\perp\perp}$ as a closure operation.

Lemma 1

(i) $T \subseteq U$ *implies* $U^{\perp} \subseteq T^{\perp}$;
(ii) $T \subseteq T^{\perp\perp}$;
(iii) $T^{\perp} = T^{\perp\perp\perp}$ *(and therefore* $(T^{\perp\perp})^{\perp\perp} = T^{\perp\perp}$*).*

This allows us to define the central notion of a biorthogonal set of terms. Due to the symmetry in the definitions, one can also define a biorthogonal set of contexts, we leave this fact implicit. The lemma which follows the definition illustrates some intuitively equivalent formulations of biorthogonality.

Definition 2 (Biorthogonals). We shall call a subset T' of $\mathbf{T}$ a *biorthogonal* if there exists $T \subseteq \mathbf{T}$ such that $T' = T^{\perp\perp}$.

Lemma 3. *The following are equivalent:*

(i) T *is a biorthogonal;*
(ii) $T = T^{\perp\perp}$;
(iii) there exists $\Gamma \subseteq \mathbf{\Gamma}$ *such that* $T = \Gamma^{\perp}$.

The basic algebraic properties of biorthogonals are expressed by the following two lemmas.

Lemma 4. $T^{\perp\perp}$ *is the smallest biorthogonal containing* T*, for all* $T \subseteq \mathbf{T}$.

Lemma 5. *Biorthogonals are closed under arbitrary intersections.* [1]

Two sets of terms T and U are said to be *logically congruent* when $T^{\perp} = U^{\perp}$. Logical congruence is an equivalence relation. The remainder of this section is devoted to illustrating several examples of the basic framework.

Example 6 (Girard's phase semantics for linear logic [8]). Let $\langle P, \cdot, 1\rangle$ be a commutative monoid with identity; let $\mathbf{T} = \mathbf{\Gamma} = \mathbf{\Pi}$ be P and @ be the action '·'. Let $\perp \subseteq P$. Then P is a phase space and the biorthogonals are its *facts*.

Example 7 (Krivine's realizability). For simplicity, let $\mathbf{T}$ be the set of terms of the simply-typed lambda calculus, and $\mathbf{\Gamma}$ a set of stacks, see [6] for details. Then let $\mathbf{\Pi}$ be the set of syntactic expressions $\langle t \mid \gamma\rangle$, for $t \in \mathbf{T}$ and $\gamma \in \mathbf{\Gamma}$. The expected reduction semantics reduction semantics on is defined on $\mathbf{\Pi}$ and $\perp \subseteq \mathbf{\Pi}$ is taken be left-closed with respect to it:

$$\langle t \mid \gamma\rangle \in \perp \quad \text{and} \quad \langle t' \mid \gamma'\rangle \to^{*} \langle t \mid \gamma\rangle \quad \text{implies} \quad \langle t' \mid \gamma'\rangle \in \perp.$$

Under the obvious interpretation of | as @, the biorthogonals are called *truth values*. See also [12] for a recent application of this technique.

[1] Biorthogonals are *not* in general closed under even finite unions, as we shall illustrate in Example 19.

Example 8 ($\top\top$-closed relations). We'll use Abadi's 'semantic' interpretation [1] of Pitts' $\top\top$-closed relations [16]. Let A, B be CPOs with bottom $\bot$. Let 2 be the poset $\{\{\bot, \top\}, \{\bot \sqsubseteq \top\}\}$, and let 2^A (resp. 2^B) denote the set of monotonic, strict and continuous functions $A \to 2$ (resp. $B \to 2$). Let $\mathbf{T} = A \times B$, $\mathbf{\Gamma} = 2^A \times 2^B$, and $\mathbf{\Pi} = 2 \times 2$, and define $@ : (A \times B) \times (2^A \times 2^B) \to 2 \times 2$ to be the obvious evaluation function $@((a, b), (f, g)) = (fa, gb)$. Let $\bot$ be the equality relation on 2. The biorthogonals are the $\top\top$*-closed relations.*

Example 9 (Formal concept theory). Introduced by Wille [20], this subject shares our notion of frameworks for biorthogonality and was proposed as a way of restructuring lattice theory to account for the interaction between what was called *objects* and *attributes*. The notion of 'concept' corresponds to our notion of biorthogonal. The goal of this research effort seems to be one of useful representations of lattices and efficient computations on these.

Example 10 (Classical algebraic geometry). Let $\mathbf{T}$ be an n-dimensional affine space over an algebraically closed field k (say k^n). Let $\mathbf{\Gamma}$ be the ring $k[x_1, \ldots, x_n]$ of polynomials with n variables. Let $\mathbf{\Pi}$ be the field k, and $@ : k^n \times k[x_1, \ldots, x_n] \to k$ be the evaluation map. Let $\bot = \{0\}$. Then the biorthogonals are the *affine varieties*. Varieties which are *irreducible*, that is, cannot be written as a nontrivial union of two other varieties, are of particular interest. We shall make use of irreducibility in §5.

3 A Refined Model: $\bot$ as an Ideal

In preparation for applications of biorthogonality to sets of terms with richer structure, we now refine our framework with a specialised notion of composition. Recall that for $\langle M, \cdot, 1\rangle$ a commutative monoid, an ideal I is a subset of M closed under the action of M, i.e., $\forall i \in I\, \forall m \in M.\ i \cdot m \in I$. We shall write $[M']$ to mean the ideal generated by a set $M' \subseteq M$. We shall now extend the basic framework as follows:

- $\mathbf{T} \subseteq \mathbf{\Pi}$ and $\mathbf{\Gamma} \subseteq \mathbf{\Pi}$;
- there exists a binary operator $\parallel$ and $\epsilon \in \mathbf{\Pi}$ such that $\langle \mathbf{\Pi}, \parallel, \epsilon\rangle$ is a commutative monoid, and for all $t \in \mathbf{T}$ and $\gamma \in \mathbf{\Gamma}$ we define $t \mathbin{@} \gamma = t \parallel \gamma$; moreover, $\mathbf{T}$ and $\mathbf{\Gamma}$ are submonoids, ie $\epsilon \in \mathbf{T}$, $\epsilon \in \mathbf{\Gamma}$ and for all $t, t' \in \mathbf{T}$, $t \parallel t' \in \mathbf{T}$ and for all $\gamma, \gamma' \in \mathbf{\Gamma}$, $\gamma \parallel \gamma' \in \mathbf{\Gamma}$;
- $\bot$ is an ideal of $\langle \mathbf{\Pi}, \parallel, \epsilon\rangle$.

Example 11 (Phase semantics for affine linear logic). With reference to Example 6, observe that collapsing $\mathbf{T} = \mathbf{\Gamma} = \mathbf{\Pi}$ *and* additionally requiring $\bot$ to be an ideal, we obtain phase semantics for affine linear logic; that is, the logic obtained from linear logic by introducing weakening.

The following lemma lists some basic properties of the refined framework. Let $\bot_{\mathbf{\Gamma}} = \bot \cap \mathbf{\Gamma}$ and $\bot_{\mathbf{T}} = \bot \cap \mathbf{T}$.

Lemma 12

(i) For every $T \subseteq \mathbf{T}$*, we have* $\bot_{\mathbf{\Gamma}} \subseteq T^{\bot}$*;*
(ii) for every $\Gamma \subseteq \mathbf{\Gamma}$*, we have* $\bot_{\mathbf{T}} \subseteq \Gamma^{\bot}$*;*
(iii) every biorthogonal $V \subseteq \mathbf{T}$ *is an ideal of* $\mathbf{T}$*;*
(iv) every biorthogonal $\Gamma \subseteq \mathbf{\Gamma}$ *is an ideal of* $\mathbf{\Gamma}$*.*

Proof. (i) Obvious, since $\bot$ is an ideal of $\mathbf{\Pi}$, for any $\pi \in \bot_{\mathbf{\Gamma}}$ and $t \in T$ we have $\pi \parallel t \in \bot$, so $\pi \in T^{\bot}$. (ii) Is immediate by duality. (iii) For arbitrary $t \in \mathbf{T}$, $v \in V$, $\gamma \in V^{\bot}$, we have that $t \parallel v \parallel \gamma \in \bot$ since $v \parallel \gamma \in \bot$. (iv) Is immediate by duality. □

4 Idealised Process Calculi

Idealised process calculi, introduced here, will be the main examples of the extended framework defined in the previous section. Although the calculi we consider do not cover the entire realm of process models systematically, they are carefully chosen to span a significant spectrum of cases. In particular, our idealised process calculi have only two constructs, action prefix and parallel composition. They are equipped with different reduction semantics which specifies the communication regime used.

The set of 'ordinary' processes P can be generated freely by a simple grammar, given in Fig 1 for some fixed countably infinite set of channel names A. Basic actions $a?$ and $a!$ represent action/co-action synchronisation pairs à la CCS. We consider the set of processes to be quotiented by structural congruence $\equiv$ which makes $\langle P, \parallel, \epsilon \rangle$ a commutative monoid. More concretely, $\equiv$ is the smallest congruence which includes the equations:

$$P \parallel Q \equiv Q \parallel P \quad \text{and} \quad P \parallel \epsilon \equiv P.$$

Ordinary processes will form the set $\mathbf{T}$ of terms.

The set of contexts, denoted C, is obtained in a similar way. A typical context is a finite parallel composition of $\checkmark$s prefixed by a single name. Here, the syntactic entity $\checkmark$ represents success. Our contexts have a simple structure because we shall always be interested in the top level structure of a term; indeed, we shall test only for a process's immediate capabilities for interaction. Finally, $P_{\checkmark}$ is the set of extended processes. An extended process is a finite parallel composition of ordinary processes, contexts and $\checkmark$s. As done for P, we also quotient C and $P_{\checkmark}$ by the structural congruence $\equiv$.

In order to simplify notation, we shall often denote action prefixing with mere juxtaposition and also write simply $a!$ or $a?$ to mean $a!\epsilon$ or $a?\epsilon$, respectively.

Different communication regimes are specified with individual sets of reduction schemas over extended processes. The *reduction semantics* is obtained by closing the reduction rules with respect to $\parallel$ in the sense that if $t \to t'$ then $t \parallel \sigma \to t' \parallel \sigma$ for any $\sigma \in P_{\checkmark}$.

In the asynchronous case, instead of following the tradition of allowing an output to prefix only the null process, we simply only include the reductions

$$
\begin{array}{rcl}
P & ::= & \epsilon \quad | \quad P \parallel P \quad | \quad M.P \\
M & ::= & a? \quad | \quad a! \qquad (a \in A) \\
C & ::= & \epsilon \quad | \quad C \parallel C \quad | \quad M_{\checkmark} \\
M_{\checkmark} & ::= & M\checkmark \\
P_{\checkmark} & ::= & P_{\checkmark} \parallel P_{\checkmark} \quad | \quad P \quad | \quad C \quad | \quad \checkmark
\end{array}
$$

Fig. 1. Idealised process calculi: processes P and contexts C

where this is the case. Thus in the asynchronous calculus a term of the form $a!.P$ where $P \neq \epsilon$ is operationally indistinguishable from ϵ as it cannot take active part in any reduction.

Example 13 (Synchrony). The reduction rules are given by the schema below where P and Q range over arbitrary extended processes.

$$a!P \parallel a?Q \;\rightarrow\; P \parallel Q \quad (a \in A)$$

Example 14 (Asynchrony). The reduction rules are given by the schema below where P ranges over arbitrary extended processes.

$$a! \parallel a?P \quad \rightarrow \quad P \qquad (a \in A)$$

Example 15 (Broadcast). The reduction rules are given by the schema below; I is any finite (possibly empty) set while P and Q_i range over extended processes.

$$a!P \parallel \prod_{i \in I} a?Q_i \quad \rightarrow \quad P \parallel \prod_{i \in I} Q_i$$

In each of the cases, the three simple calculi described above fit within our refined framework: we let $\mathbf{T} = P$, $\mathbf{\Gamma} = C$ and $\mathbf{\Pi}$ be parallel compositions of these. We define application to be parallel composition:

$$
\begin{array}{rcl}
@ : P \times C \rightarrow \mathbf{\Pi} & & \\
(t, \gamma) & \mapsto & t \parallel \gamma.
\end{array}
$$

In order to define the success predicate we make use of the extended processes $P_{\checkmark}$. We call an extended process (a member of $P_{\checkmark}$) '*spent*' if has precisely one $\checkmark$ at top level - ie $\checkmark$ is a parallel component. An extended process is deemed to be *successful* if it reduces in one step to a spent extended process. In formulae:

$$\pi \in \bot \quad \text{iff} \quad \exists \pi' \in P_{\checkmark}. \quad \pi' \text{ spent } \wedge \; \pi \rightarrow \pi'.$$

As usual, we write $p \perp \pi$ when $p @ \pi = p \parallel \pi$ is a successful extended process. Essentially, the definition of success ensures that a context has to either engage the term (since a context which would reduce by itself via a non-trivial interaction, using any of our three reduction schemas, would result in two instances of $\checkmark$) or have an atomic parallel component which reduces by itself to $\checkmark$ (for instance, the context $a!\checkmark$ in the broadcast paradigm). It is clear that in all three examples $\perp_{\mathbf{T}} = \varnothing$. In Examples 13 and 14, also $\perp_{\mathbf{\Gamma}} = \varnothing$. In Example 15 $\perp_{\mathbf{\Gamma}} = [\{\, a!\checkmark \;:\; a \in A \,\}]$, since any $a!\checkmark$ can reduce to $\checkmark$ in one step.

We conclude this section by illustrating typical biorthogonals for the simple calculi illustrated above. Recall that we use the square bracket notation to mean the smallest ideal generated by the indicated set of terms.

Example 16 (Synchrony – cf Example 13). Biorthogonals for basic terms with single communication capability are the set of all terms which have that immediate capability. So we have $\{a!\}^{\perp\perp} = ([a?\checkmark] \cup \perp)^{\perp} = [a!P]$, the set of all terms ready to output on a. Symmetrically, $\{a?\}^{\perp\perp} = [a?P]$. In general, starting with a *single* term, the biorthogonal yields all the terms that have the same selection of immediate capabilities for communication. For instance $\{a? \parallel b!c?\}^{\perp\perp} = [a!\checkmark, b?\checkmark]^{\perp} = [a?P \parallel b!Q]$. For *sets* of terms, the biorthogonal is the smallest biorthogonal which contains all of the terms. In the case of Example 13 this is simply the union of the biorthogonals of the individual terms. However as hinted at previously and illustrated by the calculus of Example 19, this need not be so, as the union of biorthogonals is in general not a biorthogonal.

Example 17 (Asynchrony – cf Example 14). We have again $\{a!\}^{\perp\perp} = [a?\checkmark]^{\perp} = [a!P]$. However, since output actions $a!$ cannot guard $\checkmark$ in a reduction, we have $\{a?\}^{\perp} = \perp_{\mathbf{\Gamma}}$ and therefore $\{a?\}^{\perp\perp} = \mathbf{T}$.

Example 18 (Broadcast – cf Example 15). Since $\perp_{\mathbf{\Gamma}} = [\{\, a!\checkmark \;:\; a \in A \,\}]$, we have $\{a?\}^{\perp\perp} = \perp_{\mathbf{\Gamma}}^{\perp} = \mathbf{T}$. Once again, $\{a!\}^{\perp\perp} = [a!P]$.

5 Irreducibility and Barbs

As indicated previously, irreducibility is an important concept of algebraic geometry. Here we shall apply the concept to biorthogonals, which in general are not closed under finite unions:

Example 19. Consider the following reduction rules which capture an interaction pattern reminiscent of the features of the join calculus [7]:

$$\begin{array}{rcll} a?P \parallel a!P' & \rightarrow & P \parallel P' & (a \in A) \\ ab?P \parallel a!P' \parallel b!P'' & \rightarrow & P \parallel P' \parallel P'' & (a, b \in A) \end{array}$$

Here the $ab?$ prefix needs the presence of both $a!$ and $b!$ to reduce. Then $\{a?\checkmark\}^{\perp\perp} = [a!P]^{\perp} = [a?\checkmark]$ and similarly $\{b?\checkmark\}^{\perp\perp} = [b!P]^{\perp} = [b?\checkmark]$. However, $\{a?\checkmark, b?\checkmark\}^{\perp\perp} = [a!P \parallel b!Q]^{\perp} = [a?\checkmark, b?\checkmark, ab?\checkmark]$ is strictly larger than $[a?\checkmark] \cup [b?\checkmark]$. Thus we see that the union of the two biorthogonals $[a?\checkmark]$ and $[b?\checkmark]$ is not itself a biorthogonal.

Definition 20 (Sum). Given biorthogonals V_1 and V_2, their *sum* $V_1 + V_2$ is defined to be the smallest biorthogonal which contains both V_1 and V_2.

It is easy to verify that $V_1 + V_2 = (V_1 \cup V_2)^{\perp\perp} = (V_1^{\perp} \cap V_2^{\perp})^{\perp}$. Thus, intuitively, $V_1 + V_2$ consists of all the terms which pass the test suites of each of V_1 and V_2. Sum $+$ is clearly commutative and easily checked to be associative. It has $\varnothing^{\perp\perp}$ as the identity. Furthermore, the binary operations $+$ and $\cap$ on biorthogonals are related by De Morgan equations: $V \cap W = (V^{\perp} + W^{\perp})^{\perp}$ and $V + W = (V^{\perp} \cap W^{\perp})^{\perp}$.

A sum $V = V_1 + V_2$ is said to be *nontrivial* when $V \neq V_1$ and $V \neq V_2$.

Definition 21 (Irreducibility). A biorthogonal is said to be *irreducible* if it cannot be written as a nontrivial sum of two biorthogonals.

The following definition is one of the central contributions of this paper.

Definition 22 (Barb). A *barb* is defined to be an proper irreducible biorthogonal B, whose orthogonal $B^{\perp}$ is also proper irreducible. For $T \subseteq \mathbf{T}$ any set of terms, T is said to *barb* on B, written $T{\downarrow_B}$, if $T^{\perp\perp} \subseteq B$. In particular, a single term t barbs on B when $\{t\}^{\perp\perp} \subseteq B$. A term $t \in \mathbf{T}$ is said to weakly barb on B, written $t{\Downarrow_B}$, if there exists t' such that $t \rightarrow^* t'$ and $t'{\downarrow_B}$.

The definition identifies barbs abstractly as 'replete' or 'maximal' sets of terms that exhibit a given basic behaviour in tests (biorthogonality). Such behaviour must be nontrivial (properness), and 'atomic' (irreducibility). The irreducibility condition on $B^{\perp}$ means that barbs are testable by suitably atomic set of contexts.

Definition 22 allows the immediate possibility of defining the standard notions of (strong and weak) barbed bisimilarity and (strong and weak) barb congruence, which provide canonical notions of equivalence. Note that there is a choice in how one defines the congruence, one can either follow Milner and Sangiorgi's original definition [14] of the largest congruence contained in barbed bisimilarity, or to take the largest congruence which is also a bisimulation. The latter equivalences are sometimes described as *dynamic*, see [15, 10].

The definition of barb which we have formulated is widely applicable; however, in any given framework, it may take some effort to identify the irreducible biorthogonals. For this reason we now seek to find a straightforward characterisation of the barbs in our range of example calculi. For now, in order to do this we have tailored the success predicate towards handshaking synchronisation and we shall also make further restrictions on the type of calculi considered. These restrictions, while intuitive and natural, disallow some more complex examples of interaction (eg Example 19), which will be the subject of future work.

6 Simple Calculi

In this section we shall require additional structure on the algebra of biorthogonals. A calculus is said to be *simple* when its algebra of biorthogonals enjoys the extra structure.

Definition 23. A biorthogonal V is said to be *finitely generated* (fg) when there exist $v_1, \ldots, v_n \in V$ such that $V = \{v_1, \ldots, v_n\}^{\perp\perp}$.

In any framework, the sum of two fg biorthogonals is fg: if $V = \{v_1, \ldots, v_k\}^{\perp\perp}$ and $W = \{w_1, \ldots, w_l\}^{\perp\perp}$ then $V + W = \{v_1, \ldots, v_k, w_1, \ldots, w_l\}^{\perp\perp}$.

Lemma 24. *Any irreducible fg biorthogonal is generated by one of its elements. Thus if V is fg and irreducible then there exists $v \in V$ such that $V = \{v\}^{\perp\perp}$.*

Proof. Suppose that V is irreducible and fg. Then there exist $v_1, \ldots, v_n \in V$ such that $V = \{v_1, \ldots, v_n\}^{\perp\perp}$. Indeed, clearly $V = \{v_1\}^{\perp\perp} + \{v_2, \ldots, v_n\}^{\perp\perp}$. By irreducibility, either $V = \{v_1\}^{\perp\perp}$ and we are finished or $V = \{v_2, \ldots, v_n\}$, where we repeat the procedure. □

In particular, if all biorthogonals are fg then all irreducible biorthogonals can be generated by a single element. If biorthogonals are closed under binary union, then the converse, that all single element generated biorthogonals are irreducible, is also true.

Lemma 25. *Suppose that biorthogonals are closed under finite union, in other words $V + W = V \cup W$. Then any biorthogonal generated by a single element is irreducible.*

Proof. Suppose that $V = \{v\}^{\perp\perp} = V_1 + V_2 = V_1 \cup V_2$. Then either $v \in V_1$ or $v \in V_2$. In the first case $V = \{v\}^{\perp\perp} \subseteq V_1$, meaning that $V = V_1$; similarly, in the second case $V = V_2$. □

We shall now show that the calculi of Examples 13, 14 and 15 have biorthogonals which are closed under unions. We shall need two technical lemmas.

Lemma 26. *The examples satisfy the following dual properties:*

(i) for all contexts $\gamma \in \mathbf{\Gamma}$ and terms t_1, $t_2 \in \mathbf{T}$:

$$t_1 \parallel t_2 \perp \gamma \quad \textit{iff} \quad t_1 \perp \gamma \textit{ or } t_2 \perp \gamma;$$

(ii) for all terms $t \in \mathbf{T}$ and contexts γ_1, $\gamma_2 \in \mathbf{\Gamma}$:

$$t \perp \gamma_1 \parallel \gamma_2 \quad \textit{iff} \quad t \perp \gamma_1 \textit{ or } t \perp \gamma_2.$$

Proof. The ‘if’ direction is obvious for both cases.

Suppose that $t_1 \parallel t_2 \perp \gamma$. If γ reduces to a spent process with no need for interaction then clearly both $t_1 \perp \gamma$ and $t_2 \perp \gamma$. Otherwise, since all reduction rules of Examples 13 and 14 have at most two parallel components in the redex, and γ has to provide at least one (since both t_1 or t_2 are ordinary processes and thus have no occurrences of ✓), it follows that $t_1 \perp \gamma$ or $t_2 \perp \gamma$. In Example 15, it is enough to consider γ of the form $a?$✓. Clearly then either t_1 or t_2 must have output capability on a, and thus $t_1 \perp \gamma$ or $t_2 \perp \gamma$.

Now suppose that $t \perp \gamma_1 \parallel \gamma_2$. Notice that $\gamma_1 \parallel \gamma_2$ reduces to a spent process without interaction if and only if either γ_1 or γ_2 (or both) can do so independently. Indeed, any interaction between γ_1 and γ_2 results in two instances of ✓ result in the reactum. Hence t must provide a part of the redex. □

Proposition 27. *The examples satisfy* $\{t_1 \parallel t_2\}^\perp = \{t_1\}^\perp \cup \{t_2\}^\perp$*, for any terms* t_1 *and* t_2*. More generally, for* T_1 *and* T_2 *any sets of terms,* $(T_1 \parallel T_2)^\perp = T_1^\perp \cup T_2^\perp$*, where* $\parallel$ *is extended to sets in the obvious pointwise manner.*

Proof. Clearly the more general second statement implies the first. Also, it is obvious that $A^\perp \cup B^\perp \subseteq (A \parallel B)^\perp$. Now suppose that $\pi \in (A \parallel B)^\perp$. If, for all $a \in A$, $\pi \perp a$, then $\pi \in A^\perp$ and we are finished. Suppose then that there exists $a \in A$ such that not $\pi \perp a$. Then, by the assumption on π, for all $b \in B$, $\pi \perp a \parallel b$. Lemma 26 implies that $\pi \perp b$. Thus $\pi \in B^\perp$. □

The examples satisfy $\{\gamma_1 \parallel \gamma_2\}^\perp = \{\gamma_1\}^\perp \cup \{\gamma_2\}^\perp$, for all contexts γ_1 and γ_2. More generally, $(\Gamma_1 \parallel \Gamma_2)^\perp = \Gamma_1^\perp \cup \Gamma_2^\perp$ for all sets of contexts Γ_1 and Γ_2.

Corollary 28. *In the examples, biorthogonals are closed under finite unions.*

Definition 29 (Simple calculi). We shall say that an idealised process calculus is simple when:

(i) $V + W = V \cup W$, for all biorthogonals V and W;
(ii) every irreducible biorthogonal has a single generating element.

Proposition 30. *The calculi introduced in Examples 13, 14 and 15 are simple.*

Proof. Corollary 28 shows that the calculi satisfy (i). If V is a finitely generated irreducible biorthogonal, then it is generated by a single element as shown in Lemma 24. It remains to show that any irreducible biorthogonal is finitely generated, the proof of this fact is more involved and will appear in a fuller version of this paper. □

Proposition 31. *In a simple idealised process calculus, the barbs coincide with proper biorthogonals generated by a single term whose orthogonal is also proper and generated by a single term.*

Proof. By definition, barbs are proper irreducible biorthogonals B whose orthogonal $B^\perp$ is also proper irreducible. We know that, by simplicity, any irreducible biorthogonal has a single generating element. On the other hand, any biorthogonal which is generated by a single term is irreducible by Lemma 25, thus if both the biorthogonal and its orthogonal are proper and generated by a single term then they are both irreducible, and the biorthogonal is a barb.

Remark 32. Several interesting and reasonable features of calculi fall outside the framework of simple calculi. For instance, the ability of synchronising with two separate processes as illustrated in Example 19, can mean that biorthogonals are not closed under binary unions. The requirement of irreducibility in the definition of 'simple' above is necessary even for the simplest calculi. An example below, written in the synchronous idealised language of Example 13, demonstrates a non finitely generated reducible biorthogonal. This example is useful because it shows that, in a finitary calculus, it is irreducibility that forces finite generation for barbs. We also provide an example of a calculus which contains non finitely generated irreducible biorthogonals. In this case, the calculus contains non-finitary reduction rules.

Example 33 (Non finitely generated reducible biorthogonal). We use the idealised synchronous language of Example 13 and let $T = \{ t_i : i \geq 0 \}$; it holds that $T^\perp$ is $\{ [\pi_i] : i \geq 0 \}$, where we define each t_i and π_i as below:

$$t_i = \prod_{j<i} b_j! \parallel a_i! \qquad \pi_i = b_i?\checkmark \parallel \prod_{j\leq i} a_j?\checkmark .$$

Consequently $T^{\perp\perp} = \{ [t_i] : i \geq 0 \}$. We claim that $T^{\perp\perp}$ is not finitely generated but is reducible as we can express $T^{\perp\perp}$ as $T^{\perp\perp} = \{t_0\}^{\perp\perp} + \{ t_i : i \geq 1 \}^{\perp\perp}$ with $\{t_0\}^{\perp\perp} \neq \{ t_i : i \geq 1 \}^{\perp\perp}$.

Example 34 (Non finitely generated irreducible biorthogonals). Suppose that the set of prefixes is

$$M \quad ::= \quad a_i? \quad | \quad b_i! \quad | \quad b_\omega \qquad (i \in \mathbb{N})$$

and that there is an infinite reaction rule schema of the form:

$$a_i?P \parallel b_j!Q \to P \parallel Q \quad (i \leq j \vee j = \omega).$$

For each i, $\{a_i?\}^{\perp\perp} = \{ b_j!\checkmark : i \leq j \vee j = \omega \}^\perp = [\{ a_k P : P \in \mathbf{T}, k \leq i \}]$. In particular, there is an infinite ascending chain

$$\{a_0\}^{\perp\perp} \subset \{a_1\}^{\perp\perp} \subset \dots$$

Now the biorthogonal $\{b_\omega\}^\perp = \{ a_i : i \in \mathbb{N} \}^{\perp\perp}$ is irreducible and not finitely generated.

We are now in a position to show the barbs for each of our three main examples. Before we do this, it is helpful to consider why certain biorthogonal are *not* barbs. Firstly, any non irreducible biorthogonal is not a barb, for instance, in the synchronous case $\{a?\}^{\perp\perp} \cup \{b?\}^{\perp\perp}$. Intuitively, this is so because the biorthogonal does not capture a single set of capabilities for interaction, its terms have either one type ($a?$) *or* the other ($b?$). An irreducible biorthogonal may also fail to be a barb because its orthogonal is reducible, meaning that no suitable single test exists for it – for instance, $\{a! \parallel b!\}^\perp = \{a?\}^{\perp\perp} \cup \{b?\}^{\perp\perp}$. Intuitively, this is because each of the terms in the biorthogonal has two distinct possibilities for interaction, here both $a?$ *and* $b?$. Finally, when a biorthogonal is not proper (it contains all the terms), it is not a barb. In this case, the term does not have non-trivial observations (cf $\{a?\}^{\perp\perp}$ in the asynchronous case).

We know because of Propositions 30 and 31 that in identifying barbs in our simple calculi it is sufficient to examine biorthogonals of singletons:

Proposition 35 (Synchronous barbs). *In Example 13, the barbs are:*

1. *for any $a \in A$, the processes which can output on a: $\{a!\}^{\perp\perp}$;*
2. *for any $a \in A$, the processes which can input on a: $\{a?\}^{\perp\perp}$.*

Proof. Both $\{a!\}^{\perp\perp}$ and $\{a?\}^{\perp\perp}$ are proper biorthogonals with orthogonals $\{a?\checkmark\}^{\perp\perp}$ and $\{a!\checkmark\}^{\perp\perp}$, respectively. Both are thus barbs by Proposition 31.

If the term has (immediate) capability to communicate on two separate channels then its orthogonal is reducible, hence it cannot be a barb. It thus suffices to notice that $\{a! \parallel a!\}^{\perp\perp} = \{a!\}^{\perp\perp}$ and $\{a? \parallel a?\}^{\perp\perp} = \{a?\}^{\perp\perp}$. □

Proposition 36 (Asynchronous barbs). *For the calculus of Example 14, the barbs are, for any $a \in A$, the processes which can output on a, $\{a!\}^{\perp\perp}$.*

Proof. As before, it suffices to check terms which have a communication capability on a single name. But $\{a!\}^{\perp\perp} = \{[a?\checkmark]\}^{\perp} = [a!P]$, a proper biorthogonal, while $\{a?\}^{\perp\perp} = \perp^{\perp} = \mathbf{T}$. □

Proposition 37 (Broadcast barbs). *For the calculus of Example 15, the barbs are, for any $a \in A$, the processes which can output on a, $\{a!\}^{\perp\perp}$.*

Proof. Again, $\{a!\}^{\perp\perp} = \{[a?\checkmark]\}^{\perp} = [a!P]$ while $\{a?\}^{\perp\perp} = \mathbf{T}$. □

7 Extension to Full Process Calculi

We have presented a general notion of barb which behaves well in our idealised calculi; this does not, however, allow us to characterise barbs in full process calculi. In this section we remedy this by observing that since the nature of barbs in their various settings is closely tied to the basic pattern of interaction between processes, and the idealised calculi are sufficient to model these interactions, then barbs in full process calculi can be obtained via translation in to the idealised languages. We shall show that, given well-behaved translations, the biorthogonals of an extended framework derived from a full process calculus are preserved through translation.

Let us assume a process calculus $\mathcal{C}$ with an inert process ϵ and parallel composition $\parallel$. This calculus can be construed as a testing framework by augmenting it with $\checkmark$ in an identical way to the idealised calculi. We will use P and C to range over terms and contexts of this framework.

Definition 38 (Translations). An interaction-preserving translation into the idealised calculus $\mathcal{I}$ consists of a pair of maps

$$[\![\ ]\!] : \mathcal{C} \to \mathcal{I} \qquad\qquad [\![\ ]\!]^{-1} : \mathcal{I} \to \mathcal{C}$$

which preserve ϵ, $\parallel$ and $\checkmark$ and moreover satisfy:

- $[\![\ ]\!]$ is surjective
- $[\![[\![P]\!]]\!]^{-1}$ is logically congruent to P.
- $[\![P]\!]@\pi \in \perp$ iff $P@[\![\pi]\!]^{-1} \in \perp$ and, dually, $[\![p]\!]^{-1}@C \in \perp$ iff $p@[\![C]\!] \in \perp$.

Lemma 39. *For any interaction-preserving translation, $[\![\ ]\!]^{-1}$ is surjective up to logical congruence.*

Proposition 40 (Translation correctness). *For any interaction-preserving translation and any set of terms/contexts A of $\mathcal{C}$, $[\![A^\perp]\!] = [\![A]\!]^\perp$.*

This tells us that interaction-preserving translations preserve (irreducible) biorthogonals, thus barbs are preserved. To find the barbs of a full process calculus then, it suffices to provide an interaction preserving translation and identify the barbs in the idealised language. We give an example of such a translation for the π-calculus below.

Example 41. We shall translate the π-calculus in to the synchronous idealised calculus. We define the mapping $[\![\ ]\!]^{-1}$ as a simple embedding which preserves, $\epsilon, \checkmark$ and $\|$ and

$$[\![a?p]\!]^{-1} = a(n).\epsilon \qquad [\![a!p]\!]^{-1} = \bar{a}n.\epsilon$$

where n is a reserved fixed name and we include special cases of the translation to preserve prefixed ticks. For the forward mapping, we also preserve $\epsilon, \checkmark$ and $\|$:

$$\begin{array}{ll} [\![\bar{a}nP]\!] = a!\epsilon & [\![!P]\!] = [\![P]\!] \\ [\![a(n)P]\!] = a?\epsilon & [\![\nu nP]\!] = \prod_{m_i \neq n?,n!} m_i\epsilon \ \text{ if } \ [\![P]\!] = \prod_I m_i\epsilon \end{array}$$

and also allow for special cases to preserve prefixed ticks. Note that, as we are interested solely in initial reductions, the role of replication, dynamic scoping and dynamically received names do not impact upon the translation. We leave it to the reader to check that this does indeed form an interaction preserving translation.

8 Conclusions and Future Work

We have introduced a formal definition of the well-known notion of barb, a basic observable of a process calculus. The definition relies only on the presence of a suitable underlying reduction semantics and relies on *biorthogonality*, a simple framework with deep roots in logic and computer science, and *irreducibility*, a concept from algebraic geometry.

We have shown that our definition yields the commonly accepted observables in idealised calculi for synchronous, asynchronous and broadcast communication. The latter fact was made possible by a characterisation of barbs in a particular class of *simple* calculi whose algebra of biorthogonals satisfies additional axioms. We have also shown how to use our idealised calculi to compute the barbs for standard calculi.

Finally, we have identified some synchronisation mechanisms which do not fit within the framework of simple calculi, but which still fit into the general framework. Our future research will concern understanding barbs in such calculi.

References

1. M. Abadi. ⊤⊤-closed relations and admissibility. *Mathematical Structures in Computer Science*, 10:313–320, 2000.
2. R. M. Amadio, I. Castellani, and D. Sangiorgi. On bisimulations for the asynchronous pi-calculus. *Theoretical Computer Science*, 195(2):291–324, 1998.

3. G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.
4. L. Cardelli. Brane calculi: Interactions of biological membranes. In *Computational Methods in Systems Biology CMSB '04*, volume 3082 of *LNCS*, pages 257–280, 2005.
5. L. Cardelli and A. D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures, FoSSaCS '98*. Springer Verlag, 1998.
6. V. Danos and J.-L. Krivine. Disjunctive tautologies as synchronisation schemes. In *Computer Science Logic CSL '00*, volume 1862 of *LNCS*, pages 292–301, 2000.
7. C. Fournet and G. Gonthier. The reflexive chemical abstract machine and the join-calculus. In *Proceedings of Symposium on Principles of Programming Languages, POPL'96*, pages 372–385. ACM Press, 1996.
8. J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–101, 1987.
9. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
10. K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 151(2):437–486, 1995.
11. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *International Conference on Concurrency Theory Concur '00*, volume 1877 of *LNCS*, pages 243–258. Springer, 2000.
12. P.-A. Melliès and J. Vouillon. Recursive polymorphic types and parametricity in an operational framework. In *Logic in Computer Science LICS '05*, pages 82–91. IEEE, 2005.
13. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
14. R. Milner and D. Sangiorgi. Barbed bisimulation. In *9th Colloquium on Automata, Languages and Programming, ICALP '92*, volume 623 of *LNCS*, pages 685–695. Springer, 1992.
15. U. Montanari and V. Sassone. Dynamic congruence vs. progressing bisimulation for CCS. *Fundamenta Informaticae*, XVI:171–199, 1992.
16. A. M. Pitts. Parametric polymorphism and operational equivalence. *Mathematical Structures in Computer Science*, 10:321–359, 2000.
17. G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University, 1981. Reprinted in Journal of Logic and Algebraic Programming 60–61 (2004) 17-139.
18. M. Reid. *Undergraduate Algebraic Geometry*, volume 12 of *London Mathematical Society Student Texts*. Cambridge University Press, 1988.
19. J. Vitek and G. Castagna. A calculus of secure mobile computations. In *IEEE Workshop on Internet Programming Languages*, 1998.
20. R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.