# Strong Bisimulations for a Calculus of Broadcasting Systems*

M. Hennessy, J. Rathke
University of Sussex

January 17, 1995

**Abstract**

We develop a version of barbed bisimulation equivalence for the broadcast calculus *CBS* and characterise the associated congruence using a new notion of *noisy bisimulation.* We then give two syntactic characterisations of noisy bisimulation equivalence over finite *CBS* terms. The first is an equational characterisation over closed terms but in this setting an infinitary inference rule is required to accommodate processes of the form $x \in S?t$. The

Pattern matching in value-passing calculi is usually

the judgements of the proof system take the form

$$b \rhd T = U$$

where $b$ is a boolean expression over the data domain. Intuitively this means that $T$ is semantically equivalent to $U$ in all instantiations or worlds which satisy $b$. The proof of completeness here relies on using symbolic bisimulations, [3] and therefore we have to present an abstract operational semantics and define a notion of symbolic bisimulation appropriate for this sublanguage.

Pattern-matching is reintroduced in Sections 5 and 6, each dealing with modifications of the proof systems of Sections 3 and 4 respectively. Having done this we now have enough expressive power to reason about finite CBS terms. This is outlined briefly in the final section, Section 7.

**Related work:**

Many programming examples of CBS in practice can be found in [14, 15]. These examples exploit the power of the broadcast operator and serve to illustrate how various algorithms can be formulated in broadcasting terms with relative ease. The only proof system we are aware of for CBS is that given in [17] which contains a sound and complete proof system for the conventional notion of strong bisimulation applied to an abstract version of CBS without value-passing. Motivations for and the development of symbolic bisimulations, which are central to our completess proofs, are presented in [3] while examples of their use are found in [4, 7].

## 2 he Broadcast Calculus

The calculus we consider is a minor variation on that of [14]. The syntax may be described by the following grammar:

$$T ::= \mathbf{O} \mid e!T \mid x \in S?T \mid b \gg T \mid \sum_{i \in I} T_i \mid T|T \mid T_{(f,g)} \mid A(\tilde{v}).$$

It has many of the usual operators of $CCS$, [8], including the nil process $\mathbf{O}$, parallel operator $|$, indexed sums $\sum_{i \in I}$ and process constants $A$, from some predefined set, which will used to define recursive processes. Input prefixes are guarded by sets of values, the process $x \in S?P$ may only receive values present in $S$. In this language communication is achieved by broadcasting values to all processes in the environment. The process $e!P$ broadcasts the value of the expression $e$ while $x \in S?P$ is a process which, on hearing the value $v$ proceeds to act like the process $P[v/x]$ providing $v \in S$; otherwise the value is ignored. The construct $b \gg T$ allows the testing of values while $T_{(f,g)}$ is a form of scoping or translation of data. Let $Val$ represent the set of values which can be broadcast and $\tau$ a special value not in $Val$; $\tau$ represents *noise* in the system, i.e. broadcasts of values which can not be deciphered by any process. Then in $T_{(f,g)}$ both $f$ and $g$ are strict functions from $Val \cup \tau$ to $Val \cup \tau$ in the sense that $f(\tau) = g(\tau) = \tau$. They are used to implement restriction and renaming and allow messages to be made local to particular processes. The strictness condition enforces the constraint that noise cannot be translated into an interpretable value.

This syntax presupposes an set of data expressions $Valexp$, ranged over by $e$ and a set of boolean expressions $BoolExp$, ranged over by $b$. We do not give a precise syntax for these languages but simply assume they have a minimal set of properties. Thus we assume $Valexp$ contains the set of values $Val \cup \{\tau\}$ and a set of variables $Var$, ranged over by $x$, and that for each pair $e, e'$ of value expressions, $e = e' \in BoolExp$. We also assume that *evaluations*, functions $\rho$ from $Var$ to $Val$, behave in a reasonable manner when extended to $ValExp$ and

$BoolExp$; when $e$ (or $b$) is closed, i.e.

| Discard | Input | Output |
|---|---|---|
| $\mathbf{O} \xrightarrow{w:} \mathbf{O}$ | | |
| $\dfrac{w \notin S}{x \in S?P \xrightarrow{w:} x \in S?P}$ | $\dfrac{v \in S}{x \in S?P \xrightarrow{v?} P[v/x]}$ | |
| $e!P \xrightarrow{w:} e!P$ | | $\dfrac{[\![e]\!] = w}{e!P \xrightarrow{w!} P}$ |
| $\dfrac{P \xrightarrow{w:} P \quad Q \xrightarrow{w:} Q}{P+Q \xrightarrow{w:} P+Q}$ | $\dfrac{P \xrightarrow{v?} P'}{P+Q \xrightarrow{v?} P'}$ | $\dfrac{P \xrightarrow{w!} P'}{P+Q \xrightarrow{w!} P'}$ |
| $\dfrac{[\![b]\!] = \mathbf{false}}{b \gg P \xrightarrow{w:} b \gg P}$ | | |
| $\dfrac{P \xrightarrow{w:} P}{b \gg P \xrightarrow{w:} b \gg P}$ | $\dfrac{P \xrightarrow{v?} P' \quad [\![b]\!] = \mathbf{true}}{b \gg P \xrightarrow{v?} P'}$ | $\dfrac{P \xrightarrow{w!} P' \quad [\![b]\!] = \mathbf{true}}{b \gg P \xrightarrow{w!} P'}$ |
| $\dfrac{P[\tilde{v}/\tilde{x}] \xrightarrow{w:}}{A(\tilde{v}) \xrightarrow{w:}}$ | $\dfrac{P[\tilde{v}/\tilde{x}] \xrightarrow{v?} P'}{A(\tilde{v}) \xrightarrow{v?} P'}$ | $\dfrac{P[\tilde{v}/\tilde{x}] \xrightarrow{w!} P'}{A(\tilde{v}) \xrightarrow{w!} P'}$ |
| $\dfrac{P \xrightarrow{gw:} P'}{P_{(f,g)} \xrightarrow{w:} P'_{(f,g)}}$ | $\dfrac{P \xrightarrow{gv?} P'}{P_{(f,g)} \xrightarrow{v?} P'_{(f,g)}}$ | $\dfrac{P \xrightarrow{w!} P'}{P_{(f,g)} \xrightarrow{fw!} P'_{(f,g)}}$ |
| $\dfrac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\beta} Q'}{P\|Q \xrightarrow{\alpha \bullet \beta} P'\|Q'} \quad \alpha \bullet \beta \neq \perp$ | | |

| $\bullet$ | $w!$ | $w?$ | $w:$ |
|---|---|---|---|
| $w!$ | $\perp$ | $w!$ | $w!$ |
| $w?$ | $w!$ | $w?$ | $w?$ |
| $w:$ | $w!$ | $w?$ | $w:$ |

Figure 1: Operational semantics for closed agents (upto symmetry of + operator).

Based on this operational semantics we wish to develop a version of strong bisimulation, [8], appropriate for *CBS*. However there is quite a range of possible definitions of when a relation over agents should be considered a strong bisimulation. Should only input and output moves be considered ? Should discards also be taken into account ? If so under what circumstances, if any, should input moves be allowed to be matched by discards ? Rather than develop a range of different theories we take the approach advocated in [16] by defining a version of *barbed bisimulation* for *CBS*, $\dot{}_{barb}$. This is straightforward and uncontroversial since it relies only on

1. a notion of reduction, which we have in $\xrightarrow{\tau!}$,
2. a notion of when agents have the ability to produce values, which we have in $\xrightarrow{v!}$.

The "correct" version of strong bisimulation for *CBS* will then be that version, if it exists, which coincides with the *CBS* congruence generated by $\ _{barb}$.

For any value $v$ let $P \downarrow v$ mean that $P \xrightarrow{v!} P'$ for some $P'$. Then a symmetric relation $\mathcal{R}$ between agents is called a *barbed bisimulation* if whenever $(P,Q) \in \mathcal{R}$ then:

$$\begin{array}{lll} \text{if } P \xrightarrow{\tau!} P' & \text{then} & \exists Q'.Q \xrightarrow{\tau!} Q' \text{ and } P'\mathcal{R}Q' \\ \text{if } P \downarrow v & \text{then} & Q \downarrow v. \end{array}$$

We use $\ _{barb}$ to denote the maximal such relation which is obviously an equivalence. However it is preserved by very few of the operators of *CBS* and is not very interesting as a semantic equivalence. Instead we concentrate on the associated congruence.

**Definition 2.2** *For agents $P$ and $Q$ let $P \ \overset{c}{}_{barb}\ Q$ if $C[P] \ _{barb}\ C[Q]$ for every* CBS *context $C[\ \_\ ]$.*

**Proof.** As in [8], to show that noisy bisimulation is preserved by composition, say, we simply let

$$\mathcal{R} = \{(P|R), (Q|R) \mid \text{for all } P, Q, R \text{ such that } P \quad_n Q\}$$

and show that $\mathcal{R}$ is a noisy bisimulation. The other operators are treated in a similar way. □

We can also capture noisy bisimulation equivalence from $\quad_{barb}$ using static contexts, i.e. contexts in which the 'hole' does not appear as a summand in a choice.

**Proposition 2.5** *If* $C[P] \quad_{barb} C[Q]$ *for every static context* $C[\_]$ *then* $P \quad_n Q$.

**Proof.** Given $P, Q$ defined over a value set $Val$, we suppose that $C[P] \quad_{barb} C[Q]$ for every static context $C$ and we assume the existence of a larger value set $Val^+ \stackrel{def}{=} Val \cup Val'$

Unlike strong bisimulation it turns out that noisy bisimulation is not preserved by the choice operator $+$. For example

$$x \in Val?\mathbf{O} \quad {}_n\, \mathbf{O}$$

but

$$v!\mathbf{O} + x \in Val?\mathbf{O} \;/_n\; v!\mathbf{O} + \mathbf{O}$$

because the agent on the right hand side can perform the sequence of actions $w??$

This property will prove invaluable in developing the axiomatisation of noisy congruence over $\mathcal{SA}$. For convenience let us introduce the notation $P \stackrel{:}{\longrightarrow}$ to denote the fact that $P$ can discard.

The axioms required to characterise strong bisimulation equivalence over *CCS* terms are simply the idempotency, symmetry and associativity of $+$ together with the fact that $\mathbf{O}$ is a zero for $+$, which we call $\mathcal{A}$:

$$\begin{aligned} X + \mathbf{O} &= X \\ X + X &= X \\ X + Y &= Y + X \\ (X + Y) + Z &= X + (Y + Z). \end{aligned}$$

In the setting of *CBS* this is insufficient. For example

$$\tau!(v!P + x?v!P) \simeq_n \tau!v!P$$

because

$$v!P + x?v!P \quad {}_n \; v!P.$$

Indeed if $Q$ is any process which can discard, *i.e.* $Q \stackrel{:}{\longrightarrow}$, then

$$Q + x?Q \quad {}_n \; Q$$

because $Q$ can discard any value. This in turn means that

$$v!(Q + x?Q) \simeq_n v!Q.$$

This phenomenon can be captured by a new axiom schema, *Noisy*:

$$\textit{Noisy:} \quad v!(P_! + x?P_!) = v!P_!$$

where $P_!$ is a meta-variable standing for any agent which can discard. For the present sublanguage $\mathcal{SA}$ this means any closed term of the form

$$\sum_{i \in I} v_i!P_i$$

for some finite index set $I$. We use $\mathcal{A}_N$ to denote the set of equations $\mathcal{A}$ together with the axiom schema *Noisy*.

There is an added complication for *CBS* which also exists for standard value-passing processing algebras, [4]. In a $\Sigma$-algebra the congruence generated by a set of equations is easily characterised in terms of substitution of equals for equals and the application of instances of the axioms. For agents in *CBS* more powerful rules are required. For although we can infer $v!P \simeq_n v!Q$ from $P \simeq_n Q$ it is not possible, in general, to infer $x?T \simeq_n x?U$ from any finite set of statements about agents; we can not require the establishment of $T \simeq_n U$ because these are open terms and the proof system only allows the manipulation of closed terms.

To overcome this problem, following [5], we introduce an infinitary proof rule:

$$\frac{T[v/x] = U[v/x] \quad \text{for every } v \in \mathit{Val}}{x?T = x?U}$$

In fact because the operational semantics we have given to *CBS* is an *early* operational semantics, [4, 9] we need a more complicated version of this rule:

$$\underline{\tau!T[v/x] + \textstyle\sum_{j \in J} \tau!U_j[v/x] = \sum_{j \in J} \tau!U_j[v/x]}$$

EQUIV $\dfrac{}{T = T} \quad \dfrac{T = U}{U = T} \quad \dfrac{T = U \quad U = V}{T = V}$

AXIOM $\dfrac{T = U \in \mathcal{AX}}{T\rho = U\rho}$

CONG $\dfrac{T_1 = U_1 \quad T_2 = U_2}{T_1 + T_2 = U_1 + U_2}$

$\alpha$-CONV $\dfrac{}{x?T = y?T[y/x]} \quad y \notin fv(T)$

cl-INPUT $\tau$

EQUIV $\dfrac{}{\mathbf{tt} \rhd T = T} \quad \dfrac{b \rhd T = U}{b \rhd U = T} \quad \dfrac{b \rhd T = U \qquad b \rhd U = V}{b \rhd T = V}$

AXIOM $T = U$

$$
\begin{array}{ll}
Ident: & X + \mathbf{O} = X \\
Idemp: & X + X = X \\
Symm: & X + Y = Y + X \\
Assoc: & (X + Y) + Z = X + (Y + Z)
\end{array}
$$

$$
Noisy: \quad e!(T_1 + x?T_2) = e!T_1 \quad \text{if } x \notin fv(T_1)
$$

Figure 4: Axioms $\mathcal{A}$ + *Noisy*

$x \notin fv(P)$ the agent $P[v/x]$ coincides with $P$. Also since $P$ is $P_1$ we know $P \stackrel{v?}{\not\longrightarrow}$ and therefore $P \stackrel{v:}{\longrightarrow} P$, which is the required match for $P + x?P \stackrel{v?}{\longrightarrow} P[v/x]$. □

**Proposition 4.2** *(Soundness) If* $\mathcal{A}_N \vdash b \rhd T = U$ *and* $\rho \models b$ *then* $T\rho \simeq_n U\rho$.

**Proof.** It is sufficient to check that all of the individual rules and axioms are sound which is straightforward; the only novelty is the axiom schema *Noisy* which is treated in the previous Lemma. □

In order to prove the completeness of our proof system we employ the techniques of [4] which unfortunately requires a notion of symbolic noisy bisimulations. These are defined using abstract transition relations which are presented in Figure 5. The abstract transitions $\stackrel{b,\alpha}{\longrightarrow}$ are labelled not only with actions but also with boolean expressions which are intended to act as guards for the move. Note that in the transition $\stackrel{b,\alpha}{\longrightarrow}$ $\alpha$ has the form :, $x?$ or $e!$. Intuitively the move $\alpha$ is enabled whenever the guard $b$ is true. This is made precise in the following

**Proposition 4.3**

**(i)** *if* $T\rho \stackrel{\tau!}{\longrightarrow} Q$ *then* $\exists b, T' \cdot T \stackrel{b,\tau!}{\longrightarrow} T'$ *where* $\rho \models b, Q \equiv_\alpha T'\rho$ *and conversely if* $T \stackrel{b,\tau!}{\longrightarrow} T'$ *and* $\rho \models b$ *then* $\exists Q \cdot T\rho \stackrel{\tau!}{\longrightarrow} Q$ *and* $Q \equiv_\alpha T'\rho$

**(ii)** *if* $T\rho \stackrel{v!}{\longrightarrow} Q$ *then* $\exists b, e, T' \cdot T \stackrel{b,e!}{\longrightarrow} T'$ *where* $\rho \models b, \rho(e) = v, Q \equiv_\alpha T'\rho$ *and conversely if* $T \stackrel{b,e!}{\longrightarrow} T'$ *and* $\rho \models b, \rho(e) = v$ *then* $\exists Q \cdot T\rho \stackrel{v!}{\longrightarrow} Q$ *and* $Q \equiv_\alpha T'\rho$

**(iii)** *if* $T\rho \stackrel{v?}{\longrightarrow} Q$ *then* $\exists b, x, T' \cdot T \stackrel{b,x?}{\longrightarrow} T'$ *where* $x \notin fv(T)$, $\rho \models b, Q \equiv_\alpha T'\rho[v/x]$ *and conversely if* $T \stackrel{b,x?}{\longrightarrow} T'$ *where* $x \notin fv(T)$ *and* $\rho \models b$ *then* $\exists Q \cdot T\rho \stackrel{v?}{\longrightarrow} Q$ *and* $Q \equiv_\alpha T'\rho[v/x]$

**(iv)** $T\rho \stackrel{v:}{\longrightarrow} T\rho$ *if and only if* $\exists b \cdot \rho \models b$ *and* $T \stackrel{b,:}{\longrightarrow} T$

**Proof.** A minor variation on Lemma 3.2 of [4]. □

We call a finite set, $B$, of boolean expressions a *b-partition* if $\bigvee B \equiv b$.

Let $S = \{S^b \mid b \in BoolExp\}$ be a family of symmetric relations on terms, indexed by boolean expressions. Define $NSB(S)$ by

> $(T, U) \in NSB(S)^b$ if whenever $T \stackrel{b_1,\alpha}{\longrightarrow} T'$ ($\alpha \equiv x?$ or $e!$) with $bv(\alpha) \cap fv(b, T, U) = \emptyset$, there is a $b \wedge b_1$-partition, $B$, such that for each $b' \in B$ there exists a $U \stackrel{b_2,\alpha'}{\longrightarrow} U'$ such that $b' \models b_2$ and
>
> - if $\alpha \equiv e!$ then $\alpha' \equiv e'!$ with $b' \models e = e'$ and $(T', U') \in S^{b'}$

The exposition of the proof of completeness of our system with respect to symbolic noisy bisimulation requires, as usual, the ability to rewrite arbitrary terms into special forms. First, a *standard form* $T$ is a term of the form

$$\sum_{i\in I_!} b_i \gg e_i!.T_i + \sum_{i\in I_?} b_i \gg x_i?T_i$$

for some finite indexing sets $I_?$ and $I_!$. We call the left hand sum $T_!$ and the right hand sum $T_?$. It is easy to see that every term can be transformed within the proof system into a standard form. However the following syntactic form will also be useful.

**Definition 4.5** *A process $T$ is said to be a* normal form *if it has the form*

$$\sum_{i\in I} c_i \gg (\sum_{k\in I_i} \alpha_{ik}.T_{ik})$$

*where $c_i \wedge c_j \equiv \mathbf{ff}$ whenever $i \neq j$ and $\bigvee_I c_i = \mathbf{tt}$.*

In order to prove that every term can be transformed into a normal form we first state a few simple facts about the proof system; the proofs are left to the reader.

**Proposition 4.6**

*(i)* $b \models b'$ *implies* $\mathcal{A} \vdash b \rhd T = b' \gg T$

*(ii)* $\mathcal{A} \vdash b \gg (T + U) = (b \gg T) + (b \gg U)$

*(iii)* $\mathcal{A} \vdash (b \gg T) + (b' \gg T) = b \vee b' \gg T$.

□

**Lemma 4.7** *For every term $T$, there exists a normal form $nf(T)$ such that $\mathcal{A} \vdash T = nf(T)$.*

**Proof.** Let the standard form of $T$ be $\sum_{j\in J} b_j \gg \alpha_j.T_J$. For each $K \subseteq J$ we define $c_K$ to be the boolean expression $\bigwedge_{k\in K} b_k \wedge \bigwedge_{k'\in J-K} \neg b_{k'}$. Thus we ha ?to

*is a derived rule of the proof system.*

**Proof.** See Proposition 3.7 of [4]. □

The following notion of a *discard condition*, $DC(T)$ for a normal form will be useful. This $DC(T)$ represents the weakest condition under which the normal form $T$ is triggered to discard. *i.e.*, $T \xrightarrow{DC(T),:} T$ and whenever $T \xrightarrow{b,:} T$ then $b \models DC(T)$. Given a normal form $T \equiv \sum_{i \in I} c_i \gg (\sum_{k \in I_i} \alpha_{ik}.T_{ik})$ then we define a predicate $?_T$ on $I$ by

$$?_T(i) \stackrel{def}{=} (\exists k \in I_i, x \in Var \cdot \alpha_{ik} \equiv x?)$$

and define $DC(T) \stackrel{def}{=} \bigwedge_{?_T(i)} \neg c_i$.

Although $DC$ is defined on normal forms the idea of a discard condition is not exclusive to them. We have a similar notion for standard forms and the construction of such is somewhat easier. In fact, we see that these constructions coincide for the two types of syntactic forms.

**Lemma 4.9** *Let* $T \equiv \sum_I$

**Proof.** The '$\Leftarrow$' direction is quite simple to prove using Theorem 4.4 so we concentrate on the '$\Rightarrow$' direction. One approach to proving this would be to prove the corresponding result about closed terms and then use Theorem 4.4 to translate to open terms. A more illuminating direct approach is given here.

We have normal forms for $T$ and $U$, that is, $T \equiv \sum_{i \in I} c_i \gg (\sum_{k \in I_i} \alpha_{ik}.T_{ik})$ and $U \equiv \sum_j$

**Corollary 4.11** *If $T, U$ are standard forms $\sum_I c_i \gg \alpha_i.T_i$, $\sum_J d_j \gg \beta_j.U_j$ respectively, then $T \ \overset{b}{n}\ U$ if and only if there exists a b-partition, $B$, such that for $x \notin fv(T, U, b')$ for each $b' \in B$ one of the following holds:*

1. $(T \simeq_n^{b'} U)$
2. $(T \simeq_n^{b'} U + x?U)$ *and* $\mathcal{A}_N \vdash b' \rhd U = U_!$
3. $(T + x?T \simeq_n^{b'} U)$ *and* $\mathcal{A}_N \vdash b' \rhd T = T_!$

**Proof.** We construct $nf(T), nf(U)$ as directed in Lemma 4.7. We know $T \simeq_n^b nf(T)$ and $U \simeq_n^b nf(U)$ by Soundness. Now apply Theorem 4.10 to get the three cases. The first case yields $nf(T) \simeq_n^b nf(U)$ and transitivity gives $T \simeq_n^b U$. In the second case we must show $\mathcal{A} \vdash b' \rhd U = U_!$. We have $b' \models DC(U)$ and Lemma 4.9 tells us that $b' \models \bigwedge_{j \in J_?} \neg d_j$.

It is simple to show that $\mathcal{A} \vdash b' \rhd U_! = U_!$. So we need only show $\mathcal{A} \vdash b' \rhd U_? = \mathbf{O}$. To do this we show that for each $j \in J_?$ we have $\mathcal{A} \vdash b' \rhd d_j \gg x_j?U_j = \mathbf{O}$. This is simply a matter of using ABSURD to get $\mathcal{A} \vdash b' \wedge d_j \rhd x_j?U_j = \mathbf{O}$ and then using GUARD to get $\mathcal{A} \vdash b' \rhd d_j \gg x_j?U_j = \mathbf{O}$.

The last case can be dealt with similarly. □

The proof of completeness is carried out by induction on a measure of the *depth* of a term:

- $d(\mathbf{O}) = 0$
- $d(x?T) = d(e!T) = 1 + d(T)$
- $d(b \gg T) = d(T)$
- $d(T_1 + T_2) = \max\{d(T_1), d(T_2)\}$

**Theorem 4.12** *(Completeness)* $T \simeq_n^b U$ *implies* $\mathcal{A}_N \vdash b \rhd T = U$

**Proof.** We assume that $T, U$ are the standard forms $\sum_{i \in I} c_i \gg \alpha_i.T_i$, $\sum_{j \in J} d_j \gg \beta_j.U_j$ respectively and proceed by induction on $d(T) + d(U)$.

We only show $\mathcal{A}_N \vdash b \rhd T_? = U_?$. The proof of $\mathcal{A}_N \vdash b \rhd T_! = U_!$ is similar and is omitted. Combining both of these we get the required $\mathcal{A}_N \vdash b \rhd T = U$.

Suppose we can prove

$$\mathcal{A}_N \vdash b \wedge c_i \rhd U_? + c_i \gg x_i?T_i = U_j$$

for each $i \in I_?$. Then an application of GUARD will yield

$$\mathcal{A}_N \vdash b \rhd U_? + c_i \gg x_i?T_i = U_?$$

Using CONG we can then combine these to get

$$\mathcal{A}_N \vdash b \rhd U_? + T_? = U_?$$

and an entirely symmetric argument will give us that

$$\mathcal{A}_N \vdash b \rhd T_? = U_? \ (= T_? + U_?).$$

Therefore we only have to fulfil the obligation of showing

$$\mathcal{A}_N \vdash b \wedge c_i \rhd U_? + c_i \gg x_i?T_i = U_?$$

for an arbitrary $i$.

Let $z$ be a variable not in $fv(b,T,U)$, let $T_?^z$ denote $\sum_{I_?} c_i \gg z?T_i[z/x_i]$ and $T_?^\tau$ denote $\sum_{I_?} c_i \gg \tau!T_i[z/x_i]$. Let $U_?^z, U_?^\tau$ denote the corresponding terms for $U$. Consider $T_? \xrightarrow{c_i, z?} T_i[z/x_i]$. Since $T \simeq_n^b U$ we know there exists a $b \wedge c_i$-partition, $B$, such that for each $b' \in B$ there exists a $U \xrightarrow{d_j, z?} U_j[z/y_j]$ such that $b' \models d_j$ and $T_i[z/x_i] \;\; {}_n^{b'} \; U_j[z/y_j]$. By Theorem 4.11 there exists a $b'$-partition, $B'$, such that for each $b'' \in B'$

1 $T_i[z/x_i] \simeq_n^{b''} U_j[z/y_j]$ or

2 $T_i[z/x_i] \simeq_n^{b''} U_j[z/y_j] + x?U_j[z/y_j]$ or

3 $T_i[z/x_i] + x?T_i[z/x_i] \simeq_n^{b''} U_j[z/y_j]$.

In each of these cases we will show how to deduce $\mathcal{A}_N \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$.

**Case 1.** We apply induction and then use the rule TAU.

**Case 2.** We apply induction again to get

$$\mathcal{A}_N \vdash b'' \rhd T_i[z/x_i] = U_j[z/y_j] + x?U_j[z/y_j]$$

In this case we also know that

$$\mathcal{A}_N \vdash b'' \rhd U_j[z/y_j] = (U_j[z/y_j]) .$$

Using axiom *Noisy* and TAU will then give

$$\mathcal{A}_N \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$$

**Case 3.** Symmetric to case 2.

For each $b'' \in B'$ we have proved $\mathcal{A}_N \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$ so we can use CUT to obtain $\mathcal{A}_N \vdash b' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$. Given that $b' \models c_i$, $b' \models d_j$ we can use Proposition 4.6 and axiom *Idemp* to produce

$$\mathcal{A}_N \vdash b' \rhd d_j \gg \tau!U_j[z/y_j] = (c_i \gg \tau!T_i[z/x_i]) + (d_j \gg \tau!U_j[z/y_j]) ..$$

Adding in the other summands of $U$ we get

$$\mathcal{A}_N \vdash b' \rhd U_?^\tau = U_?^\tau + c_i \gg \tau!T_i[z/x_i]$$

A further application of CUT gives

$$\mathcal{A}_N \vdash b \wedge c_i \rhd U_?^\tau = U_?^\tau + c_i \gg \tau!T_i[z/x_i]$$

Finally, we apply INPUT$^\gg$ to get $\mathcal{A}_N \vdash b \wedge c_i \rhd U_?^z = U_?^z + c_i \gg z?T_i[z/x_i]$. The result is obtained now by $\alpha$-conversion. □

# 5 Adding Pattern Matching

In this section we add to the finite language the pattern matching construct $x \in S?T$ and show how the two proof systems have to be adapted.

Let $\mathcal{SPA}$ denote the collection of all closed terms or agents generated by adding this construct to the language of Section 3. With the addition of this construct Lemma 3.1 is no

summation the proof system would fail to be complete. For example, in the proof systems for $\mathcal{SA}$ the agents defined over the naturals, $N$,

$$x \in N?(x > 1 \gg P) + x \in N?(x \leq 1 \gg$$

**Proof.** The soundness is simply a matter of checking the validity of the axioms and that the new rule preserves the semantic congruence. So we confine our outline to the proof of completeness. Again the proof is by induction on the combined depth of $P$ and $Q$.

Because of the newly introduced axiom *Empty* we can assume that any closed term can be transformed to a *patterned standard form*, i.e. a term of the form

$$\sum_I e_i!P_i + \sum_J x \in S_j?T_j,$$

where each set $S_j$ is non-empty. So let us assume that $P$ and $Q$ have the forms

$$\sum_I e_i!P_i + \sum_J x \in S_j?T_j, \quad \sum_K e_k!Q_k + \sum_L x \in S_l?U_l$$

respectively. It is sufficient to prove that that

$$\mathcal{A}_P \vdash_{cl} \sum_I e_i!P_i = \sum_K e_k!Q_k$$

and

$$\mathcal{A}_P \vdash_{cl} \sum_J x \in S_j?T_j = \sum_L x \in S_l?U_l$$

and as an example we consider the latter. To establish this it is sufficient, by symmetry, to prove for an arbitrary $j \in J$ that

$$\mathcal{A}_P \vdash_{cl} x \in S_j?T_j + \sum_L x \in S_l?U_l = \sum_L x \in S_l?U_l.$$

For each $v \in S_j$ we know that $P \xrightarrow{v?} T_j[v/x]$. We know that $Q \xrightarrow{v?} U_l[v/x]$ for some $l \in L$ such that $v \in S_l$ and $T_j[v/x] \;\; {}_n \; U_l[v/x]$ because $P \simeq_n Q$. Let $S_l^j = \{v \in S_j \cap S_l \mid U_l[v/x] \;\; {}_n \; T_j[v/x]\}$. This gives a *finite* partition $\{S_l^j\}_{l \in L}$ of $S_j$ such that $S_l^j \subseteq S_l$ for each $l \in L$. Then, by the idempotency of $+$ and the new axiom *Pattern* it is sufficient to show for each $l \in L$ that

$$\mathcal{A}_P \vdash_{cl} x \in S_l^j?T_j + x \in S_l?U_l = x \in S_l?U_l.$$

This can be inferred from the rule cl-P-INPUT if we can prove for each $v \in S_l^j$

$$\mathcal{A}_P \vdash_{cl} \tau!T_j[v/x] + \tau!U_l[v/x] = \tau!U_l[v/x].$$

So let us fix a particular $v \in S_l^j$ and see how this can be inferred. We know that $v \in S_l$ and $T_j$

and a boolean expression $b$, we say that $b$ is *$T$-uniform* if there exists a set $K \subseteq I_?$ such that $b \models b_K$, where $b_K$ is defined

$$\bigwedge_{i \in K} b_i \wedge \bigwedge_{i' \in I_? - K} \neg b_{i'}$$

The generalisation of $I(P)$ is defined

$$I(b, T) = \bigcup \{S_i \mid i \in I_?,\, b \models b_i\}\,.$$

We show that this is a reasonable definition by relating $I(b,T)$ to $I(T\rho)$ where $\rho$ is an evaluation such that $\rho \models b$.

**Lemma 6.1** *If $b$ is $T$-uniform then*

$$\rho \models b \textit{ implies } I(T\rho) = I(b, T)$$

**Proof.** If $b$ is $T$-uniform there exists a set $K \subseteq I_?$ such that $b \models b_K$. This gives us that $I(b,T) = \bigcup_{i \in K} S_i$, which is exactly $I(T\rho)$. □

Given this then we can present axiom *P-Noisy* for standard forms

$$b \rhd \tau!(T + x \in S?T) = \tau!T \quad \text{If } x \notin fv(T),\, b \text{ is } T\text{-uniform and } I(b,T) \cap S = \emptyset.$$

Again we simply write $\mathcal{A}_P \vdash b \rhd T = U$ to mean that $b \rhd T = U$ can be derived from the axioms in $\mathcal{A}_P$ ($\mathcal{A}$ plus *P-Noisy, Pattern*, and *Empty*) using the proof system in Figure 3 with the modified input rule, P-INPUT.

**Proposition 6.2** *(Soundness)*

$$\textit{If } \mathcal{A}_P \vdash b \rhd T = U \textit{ and } \rho \models b \textit{ then } T\rho \simeq_n U\rho.$$

**Proof.** We need only show that the modified rules/axioms are sound. The *Pattern* axiom and *Empty* axiom are evident.

For *P-Noisy* this amounts to showing that $T + x \in S?T \;\overset{b}{\underset{pn}{}}\; T$. Now we know that $b$ is $T$-uniform. Thus by Lemma 6.1 $\rho \models b$ implies $I(T\rho) = I(b,T)$. Given this we only need to show that $T\rho + x \in S?T\rho \;\;{}_n\; T\rho$ whenever $S \cap I(T\rho) = \emptyset$ for $\rho \models b'$. This follows easily.

For the rule P-INPUT, suppose $\rho \models b$. We need to show that $(x \in S?T)\rho + x \in S'?U\rho \simeq_n x \in S'?U)\rho$. The only non-trivial move to match is of the form $(x \in S?T)\rho \xrightarrow{v?} Q$. Here $Q$ must be of the form $T\rho[v/x]$ where $v \in S$. Since $x \notin fv(b)$ we have that $\rho[v/x] \models b \wedge x \in S$. So by assumption we have $\tau!T\rho[v/x] + \tau!U\rho[v/x]) \simeq_n \tau!U\rho[v/x]$ which means $T\rho[v/x] \;\;{}_n\; U\rho[v/x]$. We know that $S \subseteq S'$ and so $v \in S'$. Therefore $(x \in S'?U)\rho \xrightarrow{v?} U\rho[v/x]$ to match the move from $P$. □

Completeness is somewhat harder to prove and once more we have to appeal to to symbolic bisimulations. The general approach, and indeed the general outline of the proof, is very similar to that used in Section 4. However, we have added the pattern sets to the syntax of the language and therefore changes are necessary both to the definition of symbolic bisimulations and the associated proofs. Moreover the details are sufficiently subtle to warrant an exposition of the required modifications.

We extend our abstract operational semantics to incorporate the pattern sets in Figure 6. Recalling the abbreviation $x?T$ for the term $x \in Val?T$ we see that the extension is a conservative one. The transitions relations are, as before, labelled with boolean values acting as guards.

| Discard | Input | Output |
|---|---|---|
| $\mathbf{O} \xrightarrow{\mathbf{tt},Val:} \mathbf{O}$ | | |
| $x \in S?T \xrightarrow{\mathbf{tt},Val-S:} x \in S?T$ | $\dfrac{y \notin fv(x \in S?T)}{x \in S?T \xrightarrow{\mathbf{tt},y \in S?} T[y/x]}$ | |
| $e!T \xrightarrow{\mathbf{tt},Val:} e!T$ | | $e!T \xrightarrow{\mathbf{tt},e!} T$ |
| $\dfrac{T \xrightarrow{b,S:} T \quad U \xrightarrow{b',S':} U}{T+U \xrightarrow{b' \wedge b,S \cap S':} T+U}$ | $\dfrac{T \xrightarrow{b,x \in S?} T'}{T+U \xrightarrow{b,x \in S?} T'}$ | $\dfrac{T \xrightarrow{b,e!} T'}{T+U \xrightarrow{b,e!} T'}$ |
| $b' \gg T \xrightarrow{\neg b',Val:} b' \gg T$ | | |
| $\dfrac{T \xrightarrow{b,S:} T}{b' \gg T \xrightarrow{b,S:} b' \gg T}$ | $\dfrac{T \xrightarrow{b,x \in S?} T'}{b' \gg T \xrightarrow{b' \wedge b,x \in S?} T'}$ | $\dfrac{T \xrightarrow{b,e!} T'}{b' \gg T \xrightarrow{b' \wedge b,e!} T'}$ |

Figure 6: Patterned abstract operational semantics

The differences occur in transitions of the form $\xrightarrow{b,x \in S?}$ now decorated with the patterned input, and $\xrightarrow{b,S:}$ where $S$ records the set of values which may be discarded.

Having changed the abstract operational semantics we consider the changes in the definition of symbolic bisimulation. We give the definition of *patterned noisy symbolic bisimulations*. Suppose $\{R^b\}$ is a family of symmetric relations. Let $\mathbf{set}(x \in S?) = \mathbf{set}(S:) = S$. Define $PNSB(R)^b$ as follows:

$(T,U) \in PNSB(R)^b$ if whenever

- $T \xrightarrow{b',e!} T'$ there exists a $b \wedge b'$-partition, $B$, such that for each $b'' \in B$ there exists $U \xrightarrow{d,e'!} U'$ such that $b'' \models d$, $b'' \models e = e'$ and $(T',U') \in R^{b''}$

- $T \xrightarrow{b',x \in S?} T'$ such that $x \notin fv(b,T,U)$ there exists a $b \wedge b' \wedge x \in S$-partition, $B$, such that for each $b'' \in B$ there exists $U \xrightarrow{d,\alpha} U'$ with $\alpha \in \{x \in S'?, S':\}$ such that $b'' \models d$, $b'' \models x \in \mathbf{set}(\alpha)$ and $(T',U') \in R^{b''}$.

We call $\{R^b\}$ a patterned noisy symbolic bisimulation if $R^b \subseteq PNSB(R)^b$ for each $b$ and denote the largest such $R$ by $\left\{ \begin{smallmatrix} b \\ pn \end{smallmatrix} \right\}$. Once again we now use the definition of $\begin{smallmatrix} b \\ pn \end{smallmatrix}$ to define $\simeq^b_{pn}$ the largest congruence contained in $\begin{smallmatrix} b \\ pn \end{smallmatrix}$:

$T \simeq^b_{pn} U$ if whenever

- $T \xrightarrow{b',e!} T'$ there exists a $b \wedge b'$-partition, $B$, such that for each $b'' \in B$ there exists $U \xrightarrow{d,e'!} U'$ such that $b'' \models d$, $b'' \models e = e'$ and $(T',U') \in R^{b''}$

- $T \xrightarrow{b',x \in S?} T'$ such that $x \notin fv(b,T,U)$ there exists a $b \wedge b' \wedge x \in S$-partition, $B$, such that for each $b'' \in B$ there exists $U \xrightarrow{d,x \in S'} U'$ such that $b'' \models d$, $b'' \models x \in S'$ and $(T',U') \in R^{b''}$.

We see once more that this version of symbolic bisimulation characterises the corresponding concrete version.

**Proposition 6.3** *For any*

where $S = I(U\rho) - I(T\rho$

So for an arbitrary $i \in I_?$ we now show

$$\mathcal{A}_P \vdash b \wedge c_i \rhd U^z_? = U^z_? + c_i \gg z \in S_i?T_i[z/x_i].$$

Suppose that $T \xrightarrow{c_i, z \in S_i?} T_i[z/x_i]$. Since $T \simeq^b_{pn} U$ we know there exists a $b \wedge c_i \wedge z \in S_i$-partition, $B_i$, such that each element of $B_i$ is of the form $b' \wedge z \in S_i$ (where the $b'$ partition $b \wedge c_i$) such that there exists $U \xrightarrow{d_j, z \in S_j?} U_j[z/y_j]$ such that $b' \wedge z \in S_i \models d$, $b' \wedge z \in S_i \models z \in S_j$ and $T_i[z/x_i] \;\; {}^{b'}_{pn} \;\; U_j[z/y_j]$. The fact that $b' \wedge z \in S_i \models z \in S_j$ gives us that $S_i \subseteq S_j$. This will

Adding in the rest of the $U_j[z/y_j]$ for the $j \in J_?$ we get

$$\mathcal{A}_P \vdash b' \rhd U^z_? = U^z_? + c_i \gg z \in S_i?$$

To accommodate the translation functions we use the following coding defined inductively on terms. Let $\mathbf{dom}\,(g) = \{v \in Val \mid g(v) \neq \tau\}$.

- $\langle \mathbf{O} \rangle_{(f,g)} = \mathbf{O}$
- $\langle e!T \rangle_{(f,g)} = f(e[g(\tilde{x})/\tilde{x}])!\langle T \rangle_{(f,g)}$
- $\langle x \in S?T \rangle_{(f,g)} = x \in S \cap \mathbf{dom}\,(g)?\langle T \rangle_{(f,g)}$
- $\langle b \gg T \rangle_{(f,g)} = b[g(\tilde{x})/\tilde{x}] \gg \langle T \rangle_{(f,g)}$
- $\langle \sum_{i \in I} T_i \rangle_{(f,g)} = \sum_{i \in I} \langle T_i \rangle_{(f,g)}$
- $\langle T_{(f',g')} \rangle_{(f,g)} = \langle T \rangle_{(f \cdot f', g' \cdot g)}$

**Proposition 7.2** $\langle T \rangle_{(f,g)} \simeq_n T_{(f,g)}$.

**Proof.** Structural induction on $T$. □

The identity in Proposition 7.1 can be viewed as

## References


[1] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.

[2] E. Best, editor. *Proceedings CONCUR 93,* Hildesheim, volume 715 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[3] M. Hennessy and H. Lin. Symbolic bisimulations. Technical Report 1/92, University of Sussex, 1992.

[4] M. Hennessy and H. Lin. Proof systems for message-passing process algebras. In Best [2], pages 202–216.

[5] M. Hennessy and G.D. Plotkin. A term model for CCS. In P. Dembiński, editor, $9^{th}$ *Symposium on Mathematical Foundations of Computer Science*, volume 88 of *Lecture Notes in Computer Science*, pages 261–274. Springer-Verlag, 1980.

[6] Huimin Lin. A verification tool for value-passing processes. Computer Science Report 8/93, University of Sussex, 1993.

[7] Huimin Lin. Symbolic bisimulations and proof systems for the pi-calculus. Computer Science Report 7/94, University of Sussex, 1994.

[8] R. Milner. *Communication and Concurrency.* Prentice-Hall International, Englewood Cliffs, 1989.

[9] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, Part I + II. *Information and Computation*, 100(1):1–77, 1992.

[10] P. Panangaden and J. Reppy. The relative expressiveness of multiway rendezvous. In *Proceedings of the*