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 second is in terms of a proof system for open term where an infinitary rule is not necessary but judgements of the proof system are relative to properties of the data domain.

1 Introduction

The classical approach to communication in process algebra is to consider handshaking communication as primitive [8, 1]. Broadcast communication is then an implementable feature. However recent work on multiway rendezvous [10] shows that any implementation of an n-way synchronisation using m-way synchronisations as primitives will not yield true n-way rendezvous whenever $m < n$. Such considerations aside, if we treat broadcast communication as an implemented operator then this has consequences for proving congruence between processes which use broadcast. Each time one has a proof obligation concerning broadcast to fulfil, one has to use this implementation to translate the broadcast into handshake communications. Thus the complexity of this implementation directly affects the complexity of congruence proofs. It is evident then that this approach is not an efficient way of reasoning about broadcast comunication.

The alternative to the classical approach is to introduce a multiway synchronisation operator. This has been investigated in [12] in the context of pure process algebras, *i.e.* actions are unanalysed. However pure process algebras exceed their limitations when value-passing is to be considered [4]. Therefore a natural generalisation of the multiway synchronisation operator of [12] in the setting of a value-passing calculus would be desirable. In fact this is to be found in CBS [13].

CBS is presented in the style of value-passing CCS (VCCS). The main departure from VCCS lies in the use of channel names. VCCS explicitly names the medium or channel on which each communication takes place. This gives rise to the notions of local channels and communication topology. Unfortunately channel names do not sit well in a broadcast setting because communication takes place between *all* agents in the network not just ones with communication capabilities on a particular channel, i.e. the communication topology is discrete. For this reason CBS uses but a single channel often referred to as the *ether*. This does not preclude local message passing; local communication is achieved by tagging messages with an identifier. A tagged message is not a distinguished piece of data and is treated simply as another value or message. This approach to local messages necessitates the use of *pattern matching* of values.

*This work has been supported by the ESPRIT/BRA CONCUR2 project and the EPSRC grant GR/H16537

Pattern matching in value-passing calculi is usually effected by a post-reception boolean test; the recipient deciding, having received the value, what action to take. We propose that the pattern matching in CBS is to be done prior to reception. A process not intended to receive a particular class of values will simply not receive them. In some sense the matching is done by the ether. As we shall see this choice has a notable effect on the proof system for semantic equivalence.

So far it would appear that there will be little difference between congruence proofs in VCCS and CBS. So long as we restrict our attention to the finite sublanguages of each of these the difference in the communication mechanism is accounted for by using different expansion rules for parallelism. So why is the theory significantly different? The answer lies in the notion of observational equivalence we use. In a handshaking calculus it seems quite reasonable to treat reception as an observable action. This is due to the fact that in order to receive some data the recipient has to make its presence known to the sender. However when we move to a broadcasting calculus observability of reception is not so obvious. In fact a process which has just transmitted a value onto the ether has no real way of telling which other processes, if any, received that value. The difference in the notion of observable actions between a handshaking and a broadcasting calculus provide the differences in the theories of the calculi. In this paper we look at the effect this has on the notion of strong bisimulation for a broadcast calculus. This leads to a new type of bisimulation which we call *noisy bisimulation* and we justify our interest in these by appealing to the idea of *barbed bisimulations* of [16].

We then furnish the calculus with sound and complete proof systems for resulting semantic equivalence, which we call *noisy bisimulation equivalence*. Following the approach of [4] we do not give a detailed description of what constitutes a value domain, or even the language of boolean expressions; we merely state assumed properties of such. Consequently any proof system presented for our language must rely upon auxiliary proof systems for reasoning about data. Using proof systems identical to those in [4] is advantageous as it facilitates a straightforward implementation in VPAM [6], a verification system designed with this parametric approach to data in mind.

We now give a brief outline of the content of the paper. The next section introduces the language and its operational semantics. This is very similar to the language considered in [14] except that the input prefix construct $x?T$ is replced by the family of input prefix constructs $x \in S?T$ where S ranges over arbitrary subsets of values; thus informally we continue to refer to our language as CBS. We then derive our notion of bisimulation by defining barbed bisimulation [16] in CBS. Barbed bisimulation uses simple uncontentious observability predicates which circumvent the question of observability of reception. The resulting equivalence, *barbd bisimulation equivalence* $\sim_{barb}$, is preserved by very few of the operators of the language but we give a simple characterisation of the largest congruence contained in $\sim_{barb}$. This is *noisy bisimulation equivalence* $\sim_n$, defined in terms of the new kinds of bisimulations, *noisy bisimulations*, referred to above.

The remainder of the paper is devoted to syntactic characterisations of $\sim_n$ in terms of axioms and proof systems. In Section 3 we present a proof system for closed terms of a core sublanguage of finite CBS which only contains the prefix and choice operators. Moreover for simplicity the only input prefix construct alowed is $x?T$ which can be taken to be a notation for $x \in Val?T$ where Val is the universe of allowed values. Although this may seem like a trivial language its treatment will identify the essential properties of noisy bisimulation.

Because the proof system presented in Section 3 deals with closed terms only it is necessarily contains an infinitary rule in order to handle the input prefix construct $x?T$, [4]. We improve on this in Section 4 by presenting an open term proof system for the same sublanguage. Here

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 The 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 τ a special value not in Val; τ 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 ρ from Var to Val, behave in a reasonable manner when extended to $ValExp$ and

$BoolExp$; when e (or b) is closed, i.e. contains no occurrences of variables, then the value of the expression e is independent of ρ and we denote it by $[\![e]\!]$. Substitutions in data and boolean expressions are written as $e[e'/x], b[e'/x]$ respectively, meaning the substitution in e, b of all occurrences of x by e'. This substitution is extended to process terms T homomorphically in the obvious way, denoted by $T[e/x]$, except that only *free occurrences* of x are substituted; as is usual in the term $y \in S?U$ y acts as a binder for all occurrences of y in U and this gives rise to the set of free variables of a term T, $fv(T)$, and α-equivalence $\equiv_\alpha$ between terms. Finally we use $T, U, \ldots$ to range over arbitrary process terms whereas $P, Q, \ldots$ denote closed process terms or *agents*, i.e. terms with no free variables.

We now consider an operational semantics for this language, CBS; again this more or less coincides with that presented in [14]. Throughout we assume that with each constant name A we have an associated definition:

$$A(\tilde{x}) \stackrel{def}{=} T$$

where $\tilde{x}$ contains all of the free variables that appear in T, and A occurs guarded in T. The most notable difference between the operational semantics of CCS, [8], and CBS is the introduction of a new kind of transition called *discard*, written $T \xrightarrow{w:} T$. CBS is neither a wholly synchronous nor asynchronous calculus; the transmission of data is an autonomous action and agents may asynchronously do so. However, reception is reactive and agents that are ready to receive must synchronise with transmitting agents. In order to present this operationally the semantics is given in the manner of a synchronous calculus and the apparent asynchrony is codified by using these discard transitions.

The operational semantics is presented in Figure 1. It consists of three different kinds of binary relations over agents, $P \xrightarrow{v?} Q$ representing the effect of inputing a value v, $P \xrightarrow{w!} Q$, $w \in Val \cup \tau$, representing the output of the value w and the novel discard relation $P \xrightarrow{w:} Q$. The reader is refered to [14] for more explanation and discussion of these rules.

Some simple properties of these relations are given in the following lemma:

Lemma 2.1 *For every agent P*

- *if $P \xrightarrow{w:} Q$ then Q is P*
- *$P \xrightarrow{v:} P$ if and only if there does not exist a Q such that $P \xrightarrow{v?} Q$*
- *$P \xrightarrow{\tau:} P$*

Proof. By induction on the rules of inference in Figure 1. □

Intuitively a process discards a value when it is in a state in which values can not be received. So the first property of this Lemma is very natural: ignoring or discarding a broadcasted value does not change the state of a process. The second property states that discarding a value is exactly the same as not being able to receive it. The final property states that all processes ignore noise. This relies on the fact that in all guarded input terms $x \in S?T$, S is a subset of Val, i.e. does not contain τ.

At the level of labelled transition systems CBS appears to be very similar to the value-passing process algebras of [4] and the operational semantics given above corresponds very much to the *early* operational semantics of that paper. However it is worth pointing out that at least one expected property is not true: $P \xrightarrow{v?} Q$ does NOT imply that for every value v' there is a process $Q_{v'}$ such that $P \xrightarrow{v'?} Q_{v'}$. One reason is the use of guarded inputs, $x \in S?T$; here a value can be input only if it is in S. However even if the only input construct allowed is $x \in Val?T$ the property still does not hold. For example the process $(x \in Val?P)_{(f,g)}$ can only receive the values from Val which g doesn't map to τ.

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*, $\sim_{barb}$. 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 $\sim_{barb}$.

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 $\sim_{barb}$ 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 $P \sim^c_{barb} Q$ if $C[P] \sim_{barb} C[Q]$ for every* CBS *context $C[\_]$.*

The remainder of this section is devoted to giving a bisimulation type characterisation of $\sim^c_{barb}$.

The characterisation is easiest to explain in terms of a new relation. In a broadcasting calculus an observer can not see whether a given process can actually input a particular broadcasted value or simply discard it; it reacts internally to the broadcast either by accepting the value and adjusting its internal state or ignoring the broadcast entirely. These different reactions can not be observed externally. This is captured by the following definition.

let $P \xrightarrow{v??} Q$ if $P \xrightarrow{v?} Q$ or $P \xrightarrow{v:} Q$.

With this new arrow we define a new kind of bisimulation relation. A symmetric relation $\mathcal{R}$ between agents is called a *noisy bisimulation* if whenever $(P,Q) \in \mathcal{R}$ then:

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

We let $P \sim_n Q$ if there exists some noisy bisimulation R such that $(P,Q) \in R$, i.e. $\sim_n$ is the largest noisy bisimulation.

Because of Lemma 2.1 noisy bisimulations can be simplified considerably:

Proposition 2.3 *Let R be a symmetric relation over agents. Then R is a noisy bisimulation if and only if when $(P,Q) \in R$ then*

1. *$P \xrightarrow{w!} P'$ implies there is some Q' such that $P'\mathcal{R}Q'$ and $Q \xrightarrow{w!} Q'$ and*
2. *$P \xrightarrow{v?} P'$ implies there is some Q' such that $P'\mathcal{R}Q'$ and either $Q \xrightarrow{v?} Q'$ or $Q \xrightarrow{v:} Q'$*

Proof. Suppose R satisfies the conditions of the Proposition. We need only check that a discard move $P \xrightarrow{v:} P'$, where $(P,Q) \in R$, can be matched by a move from Q. We know from the first part of Lemma 2.1 that P' must be P. In fact if $Q \xrightarrow{v:} Q'$ then Q' must also be Q and we are done. Therefore we assume that $Q \not\xrightarrow{v:} Q$. The second part of Lemma 2.1 tells us that $Q \xrightarrow{v?} Q'$ for some Q'. This implies, using the second property of R, that $P \xrightarrow{v??} P'$ with $(P',Q') \in R$. It follows, again from Lemma 2.1, that $P \not\xrightarrow{v?}$ and so we know that P' is P and $(P,Q') \in R$. □

Proposition 2.4 *The relation $\sim_n$ is preserved by all of the CBS operators except choice.*

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

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

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 $\sim_{barb}$ using static contexts, i.e. contexts in which the 'hole' does not appear as a summand in a choice.

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

Proof. Given P, Q defined over a value set Val, we suppose that $C[P] \sim_{barb} C[Q]$ for every static context C and we assume the existence of a larger value set $Val^+ \stackrel{def}{=} Val \cup Val' \cup \{a, b\}$, where Val' is a set of values such that for each $v \in Val$ there exists exactly one v' in Val' with $v' \notin Val$ and $a, b \notin Val \cup Val'$. Let $f : Val^+ \cup \tau \to Val^+ \cup \tau$ be defined thus

$$f(w) = \begin{cases} \tau & \text{if } w \in Val \\ w & \text{otherwise.} \end{cases}$$

Let D be the constant with associated definition

$$D = x \in Val?(a!\mathbf{O} + x'!\mathbf{O} + \tau!D) + \sum_{v \in Val} v!(b!\mathbf{O} + v'!\mathbf{O} + \tau!D)$$

and let $C[\ \_\ ]$ be the context $(\ \_\ |D)_{[f,Id]}$.

Let $S = \{(R, S) \mid C[R] \sim_{barb} C[S], R, S : Val\}$, where $R : Val$ means that R is a closed term defined over the value set Val. We know that $(P, Q) \in S$ by hypothesis, so we aim to show that S is a noisy bisimulation.

Suppose that $P \stackrel{v_0!}{\longrightarrow} P'$. Then $C[P] \stackrel{\tau!}{\longrightarrow} C'_{v_0}[P']$, where

$$C'_{v_0}[\ \_\ ] = (\ \_\ |(a!\mathbf{O} + v'_0!\mathbf{O} + \tau!D))_{[f,Id]}.$$

We know that $C[P] \sim_{barb} C[Q]$ so $C[Q] \stackrel{\tau!}{\longrightarrow} R$ for some $R \sim_{barb} C'_{v_0}[P']$. Now $C'_{v_0}[P'] \downarrow a$ so $R \downarrow a$ necessarily, thus $Q \stackrel{v_1!}{\longrightarrow} Q'$ and $R \equiv C'_{v_1}[Q']$ for some v_1, Q'. We also know that $C'_{v_0}[P'] \downarrow v'_0$. So it must be that $R \downarrow v'_0$, which forces $v_1 = v_0$. We have that $Q \stackrel{v_0!}{\longrightarrow} Q'$ and must now show that $(P', Q') \in S$.

Observe that $C'_{v_0}[P'] \stackrel{\tau!}{\longrightarrow} C[P']$. It must be the case that $R \stackrel{\tau!}{\longrightarrow} R'$ with $R' \sim_{barb} C[P']$ because $R \sim_{barb} C'_{v_0}[P']$. But $C[P'] \not\downarrow v$ for all v and so $R' \downarrow v$ cannot hold for any v. Thus $R' \equiv C[Q']$, that is $(P', Q') \in S$.

Suppose $P \stackrel{v_0?}{\longrightarrow} P'$. Then $C[P] \stackrel{\tau!}{\longrightarrow} C''_{v_0}[P']$, where

$$C''_{v_0}[\ \_\ ] = (\ \_\ |(b!\mathbf{O} + v'_0!\mathbf{O} + \tau!D))_{[f,Id]}.$$

We have $C[Q] \stackrel{\tau!}{\longrightarrow} R$ for some $R \sim_{barb} C''_{v_0}[P']$. Now $C''_{v_0}[P'] \downarrow b$, so $R \downarrow b$. This means $R \equiv C''_{v_1}[Q']$ for some v_1, Q', such that $Q \stackrel{v_1??}{\longrightarrow} Q'$. We know $C''_{v_0}[P'] \downarrow v'_0$. So it must be that $R \downarrow v'_0$, which forces $v_1 = v_0$. We have $Q \stackrel{v_0??}{\longrightarrow} Q'$ and must show that $(P', Q') \in S$.

It is clear that $C''_{v_0}[P'] \stackrel{\tau!}{\longrightarrow} C[P']$. So there exists an R' such that $R \stackrel{\tau!}{\longrightarrow} R'$ with $R' \sim_{barb} C[P']$. Now $C[P] \not\downarrow v$ for all v, so it is also the case that $R' \not\downarrow v$ for all v. Thus $R' \equiv C[Q']$, which means that $(P', Q') \in S$.

Suppose $P \stackrel{\tau!}{\longrightarrow} P'$. Then $C[P] \stackrel{\tau!}{\longrightarrow} C[P']$, so $C[Q] \stackrel{\tau!}{\longrightarrow} R$ for some $R \sim_{barb} C[P']$. But $C[P'] \not\downarrow a, b$, so it is also the case that $R \not\downarrow a, b$. This means that no communication can have taken place between the context and Q. Thus $Q \stackrel{\tau!}{\longrightarrow} Q'$ with $R \equiv C[Q']$, so $(P', Q') \in S$. □

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

$$x \in Val?\mathbf{O} \sim_n \mathbf{O}$$

but

$$v!\mathbf{O} + x \in Val?\mathbf{O} \not\sim_n v!\mathbf{O} + \mathbf{O}$$

because the agent on the right hand side can perform the sequence of actions $w??v!$ which is not possible for that on the left hand side. However it can be easily modified to take choice contexts into account.

Definition 2.6 *Let $P \simeq_n Q$ be given by*

$$\begin{array}{lll} \textit{if } P \xrightarrow{w!} P' & \textit{then} & \exists Q'.Q \xrightarrow{w!} Q' \textit{ and } P' \sim_n Q' \\ \textit{if } P \xrightarrow{v?} P' & \textit{then} & \exists Q'.Q \xrightarrow{v?} Q' \textit{ and } P' \sim_n Q'. \end{array}$$

We say that P and Q are strong noisy congruent.

Theorem 2.7 *$P \sim^c_{barb} Q$ if and only if $P \simeq_n Q$.*

Proof. It is straightforward to adapt Proposition 2.4 to show that $\simeq_n$ is preserved by all *CBS* operators. Since $P \simeq_n Q$ trivially implies that $P \sim_{barb} Q$ it follows immediately that $P \simeq_n Q$ implies $P \sim^c_{barb} Q$.

Conversely, suppose $P \sim^c_{barb} Q$. Let v be a new value not occurring in P, Q. Then $P + x \in Val?v!\mathbf{O} \sim_n Q + x \in Val?v!\mathbf{O}$ implies $P \simeq_n Q$. But $P \sim^c_{barb} Q$ means that for every static context $C[\,]$, $C[P + x \in Val?v!\mathbf{O}] \sim_{barb} C[Q + x \in Val?v!\mathbf{O}]$ and therefore by Proposition 2.5 it follows that $P + x \in Val?v!\mathbf{O} \sim_n Q + x \in Val?v!\mathbf{O}$. □

This theorem justifies our choice of $\simeq_n$ as the appropriate version of strong bisimulation equivalence for *CBS* and will be studied in the next two sections.

3 Characterising Strong Noisy Congruence over Simple Agents

In this section we give an algebraic characterisation of Strong Noisy Congruence over a simple class of finite agents. In fact we restrict our attention to closed terms of the simple language given by:

$$T ::= \mathbf{O} \mid e!T \mid x?T \mid b \gg T \mid T + T.$$

In order to obtain a finite language we have replaced the summation operator $\sum_I$ with the binary choice $+$. We have also removed the guards from the input prefixes, the syntax $x?T$ being shorthand for $x \in Val?T$. The extra *CBS* operators, parallel and restriction or translation, will be accommodated later. In order to accommodate these though the patterned guards on inputs will have to be reintroduced; we attend to this in Section 5. Let us use $\mathcal{SA}$ to denote the set of agents definable in this sub-language.

Unlike *CBS*, processes in $\mathcal{SA}$ have the very simple property that if they can discard one value then they can discard every value, or equivalently if they can input one value they can input every value:

Lemma 3.1 *For all processes P in $\mathcal{SA}$ if there exists some value v such that $P \xrightarrow{v:}$ then for every value v, $P \xrightarrow{v:}$.*

Proof. By structural induction on P. □

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

$$v!P + x?v!P \sim_n v!P.$$

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

$$Q + x?Q \sim_n Q$$

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 Σ-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:

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

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:

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

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

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

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

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

cl-INPUT $\quad \dfrac{\tau!T[v/x] + \sum_{j\in J} \tau!U_j[v/x] = \sum_{j\in J} \tau!U_j[v/x] \quad \text{for every } v \in Val}{x?T + \sum_{j\in J} x?U_j = \sum_{j\in J} x?U_j}$

OUTPUT $\quad \dfrac{T = U, [\![e]\!] = [\![e']\!]}{[\![e]\!]!T = [\![e']\!]!U}$

BOOL $\quad \dfrac{[\![b]\!] = \mathbf{tt}}{b \gg T = T} \quad \dfrac{[\![b]\!] = \mathbf{ff}}{b \gg T = \mathbf{O}}$

Figure 2: Inference Rules

In short, for agents in $\mathcal{SA}$, instead of considering the congruence generated by a set of axioms $\mathcal{AX}$ we consider the identities derivable in the proof system given in Figure 2.

For any agents P, Q let $\mathcal{A}_N \vdash_{cl} P = Q$ mean that $P = Q$ can be derived in this proof system from the axioms $\mathcal{A}$ together with the schema *Noisy*.

Theorem 3.2 *(Soundness and Completeness)* $\mathcal{A}_N \vdash_{cl} P = Q$ *if and only if* $P \simeq_n Q$. □

We omit the proof of this theorem as it can be reconstructed from that of Theorem 4.2 and Theorem 4.12.

4 Characterising Strong Noisy Congruence over Open Terms

The proof system of the previous section is only of theoretical interest since it contains an infinitary rule of inference. In this section we show that this can be avoided by developing a proof system for open terms. In such a system the identity $x?T = x?U$ can be inferred from $T = U$ but because of open terms the proof system needs to be more complicated.

Judgements are of the form

$$b \rhd T = U$$

where b is a boolean expression and T and U are arbitrary terms. We will use the notation, $\rho \models b$ to mean $[\![b\rho]\!] = \mathbf{tt}$ and $b \models b'$ to mean that for every ρ such that $\rho \models b$ then $\rho \models b'$. The inference rules for the system are given in Figure 3. For a detailed explanation of the proof rules, and discussion of the general approach we refer the reader to [4], which uses an identical system save for small notational differences. The axioms for the proof system appear

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

AXIOM $\quad \dfrac{T = U \in \mathcal{AX}}{\mathbf{tt} \rhd T\rho = U\rho}$

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

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

INPUT $\quad \dfrac{b \rhd \tau!T + \sum_{j \in J} \tau!U_j = \sum_{j \in J} \tau!U_j}{b \rhd x?T + \sum_{j \in J} x?U_j = \sum_{j \in J} x?U_j} \quad x \notin fv(b)$

OUTPUT $\quad \dfrac{b \models e = e' \quad b \rhd T = U}{b \rhd e!T = e'!U}$

TAU $\quad \dfrac{b \rhd T = U}{b \rhd \tau!T = \tau!U}$

GUARD $\quad \dfrac{b \wedge b' \rhd T = U \quad b \wedge \neg b' \rhd \mathbf{O} = U}{b \rhd b' \gg T = U}$

CUT $\quad \dfrac{b \models b_1 \vee b_2 \quad b_1 \rhd T = U \quad b_2 \rhd T = U}{b \rhd T = U}$

ABSURD $\quad \dfrac{}{\mathbf{ff} \rhd T = U}$

Figure 3: Inference Rules

in Figure 4; these consists of $\mathcal{A}$ used in the previous section together with a version of the *Noisy* schema appropriate for open terms. Here $T_!$ stands for any term of the form

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

Note that any closed instantiation of such a term discards every transmitted value since it can not receive an input. Allowing a slight abuse of notation let us again use $\mathcal{A}_N$ to refer to this collection of axioms and for a given set of axioms $\mathcal{B}$ let $\mathcal{B} \vdash b \rhd T = U$ mean that $b \rhd T = U$ can be derived in the proof system of Figure 3 from the axioms in $\mathcal{B}$.

Lemma 4.1 *(Axiom Noisy is sound) For all ρ, if $x \notin fv(T_!)$ then $(e!(T_! + x?T_!))\rho \simeq_n (e!T_!)\rho$.*

Proof. Consider an arbitrary closed instantiation of *Noisy*: $v!(P + x?P) \simeq_n v!P$ where $x \notin fv(P)$ and P has the form $P_!$. It is sufficient then to show that $P + x?P \sim_n P$. Let $\mathcal{I}$ be the identity relation over agents. We show that $\mathcal{I}' = \mathcal{I} \cup \{(P + x?P, P), (P, P + x?P)\}$ is a noisy bisimulation. The only non-trivial move to match is $P + x?P \xrightarrow{v?} P[v/x]$. Since

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

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

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 $\stackrel{b,\alpha}{\longrightarrow}$ α has the form :, $x?$ or $e!$. Intuitively the move α 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 $(T', U') \in S^{b'}$

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

Figure 5: Abstract operational semantics

- if $\alpha \equiv x?$ then ($\alpha' \equiv x?$ or $\alpha' \equiv :$) and $(T', U') \in S^{b'}$.

We call S a noisy symbolic bisimulation if $S \subseteq NSB(S)$ (point-wise inclusion) and denote the largest such relation by $\{\sim_n^b\}$. It is evident that this relation is not a congruence for the language as it is not preserved by summation. As before we modify it so that we obtain the largest *CBS* congruence contained within it:

let $T \simeq_n^b U$ if whenever $T \xrightarrow{b_1,\alpha} T'$ ($\alpha \equiv x?$ 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 $U \xrightarrow{b_2,\alpha'} U'$ such that $b' \models b_2$ and

- If $\alpha \equiv e!$ then $\alpha' \equiv e'!$ with $b' \models e = e'$ and $T' \sim_n^{b'} U'$
- If $\alpha \equiv x?$ then $\alpha' \equiv x?$ and $T' \sim_n^{b'} U'$

The procedure now is to show completeness of the proof system with respect to $\{\simeq_n^b\}$, that is $T \simeq_n^b U$ if and only if $\mathcal{A}_N \vdash b \rhd T = U$. Given this we then need only relate the symbolic congruence to the concrete congruence. This is done in a straightforward way in the following theorem.

Theorem 4.4 *$T \simeq_n^b U$ if and only if $\forall \rho \cdot \rho \models b$ implies $T\rho \simeq_n U\rho$. In particular for agents we have that $P \simeq_n^{\mathbf{tt}} Q$ if and only if $P \simeq_n Q$*

Proof. As in [4], we use Proposition 4.3 to prove that whenever S is a noisy symbolic bisimulation then

$$R_S \stackrel{def}{=} \{(T\rho, U\rho) \mid \exists b \cdot \rho \models b \text{ and } (T,U) \in S^b\}$$

is a noisy bisimulation. Similarly, whenever R is a noisy bisimulation then

$$S_R^b \stackrel{def}{=} \{(T,U) \mid \rho \models b \text{ implies } (T\rho, U\rho) \in R\}$$

forms a noisy symbolic bisimulation. The result follows easily from this. □

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 have $\bigvee c_K \equiv \mathbf{tt}, c_K \wedge c_{K'} \equiv \mathbf{ff}$ whenever $K \neq K'$. Using the previous proposition we can show

$$\mathcal{A} \vdash \mathbf{tt} \rhd T = \sum_K c_K \gg (\sum_{k\in K} b_k \gg \alpha_k.T_k).$$

Using CUT and the proposition we can obtain, for each K,

$$\mathcal{A} \vdash c_K \rhd T = \sum_K c_K \gg (\sum_{k\in K} \alpha_k.T_k).$$

Thus, given that $\bigvee c_K \equiv \mathbf{tt}$, CUT gives

$$\mathcal{A} \vdash \mathbf{tt} \rhd T = \sum_K c_K \gg (\sum_{k\in K} \alpha_k.T_k).$$

□

The first use of these normal forms is in generalising the rule INPUT to deal with guarded terms; this generalisation will be used in the completeness theorem.

Proposition 4.8 *Suppose $x \notin fv(b, c_i, d_j)$. Then*

$$INPUT^{\gg} \quad \frac{b \rhd c \gg \tau!T + \sum_{j\in J} d_j \gg \tau!U_j = \sum_{j\in J} d_j \gg \tau!U_j}{b \rhd c \gg x?T + \sum_{j\in J} d_j \gg x?U_j = \sum_{j\in J} d_j \gg x?U_j}$$

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.*, $T \xrightarrow{DC(T),c} T$ and whenever $T \xrightarrow{b,c} T$ 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 $T \equiv \sum_I b_i \gg \alpha_i.T_i$ be a standard form, $T' \equiv \sum_{K\in\mathcal{P}I} c_K \gg (\sum_{i\in K} \alpha_{K_i}.T_{K_i})$ be the normal form constructed from T as described in Lemma 4.7. Then $\bigwedge_{i\in I_?} \neg b_i$ is logically equivalent to $DC(T')$.*

Proof. Suppose $\rho \models \bigwedge_{i\in I_?} \neg b_i$. Then $\rho \not\models (\bigwedge_{j\in K} b_j) \wedge (\bigwedge_{j\in I-K} \neg b_j)$ whenever $I_? \cap K \neq \emptyset$. This amounts to saying that $\rho \not\models c_K$ whenever $?_{T'}(K)$. That is, $\rho \models c_K$ for all K such that $?_{T'}(K)$ and so $\rho \models DC(T')$.

Conversely suppose that $\rho \models DC(T')$ and suppose for contradiction that there is an $i_0 \in I_?$ such that $\rho \models b_{i_0}$. We let $K_0 = \{i_0\}$ and define a strictly increasing sequence of subsets of I

$$K_0 \subset K_1 \subset \cdots \subset K_n \subset \cdots$$

with the property that $\rho \models b_j$ for all $j \in K_n$ for all n. Given K_n we know that $?_{T'}(K_n)$ because $i_0 \in K_n$. We know then that $\rho \models \neg c_{K_n}$. Recall that $\neg c_{K_n} = (\bigvee_{j\in K_n} \neg b_j) \vee (\bigvee_{j\in I-K_n} b_j)$. We know that there must exist a $j_0 \in I - K_n$ such that $\rho \models b_{j_0}$ because of the property of K_n that $\rho \models b_j$ for each $j \in K_n$. Let $K_{n+1} = K_n \cup \{j_0\}$. This defines a strictly increasing sequence which is bounded by the finite set I, which is a contradiction. □

We come now to the theorem which lies at the heart of the completeness theorem. It relates the symbolic noisy bisimulation relation to the symbolic noisy congruence relation. The completeness theorem for finite *CCS* terms with respect to weak bisimulation congruence $\approx_c$, [8], page 156, relies on a similar relationship between weak bisimulation, $\approx$, and bisimulation congruence, $\approx_c$: if $P \approx Q$ then either $P \approx_c Q$, $P \approx_c \tau.Q$ or $\tau.P \approx_c Q$. For *CBS* the corresponding relation is if $P \sim_n Q$ then either $P \simeq_n Q$, $P \simeq_n x?Q + Q$ or $x?P + P \simeq_n Q$ where x is a new variable. However, at the symbolic level the relationship is a little more complicated.

Theorem 4.10 *If T, U are normal forms then $T \sim_n^b U$ 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* $b' \models DC(U)$
3. $(T + x?T \simeq_n^{b'} U)$ *and* $b' \models DC(T)$.

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 $U \equiv \sum_{j\in J} d_j \gg (\sum_{l\in J_j} \beta_{jl}.U_{jl})$.

Let $B' \stackrel{def}{=} \{b \wedge c_i \wedge d_j \mid i \in I, j \in J\}$. Then we know that $\bigvee B' = b$. Consider $b' \equiv b\wedge c_i\wedge d_j \in B'$. We know that $T \sim_n^{b'} U$ because $b' \models b$. So whenever $T \stackrel{c_i,x?}{\longrightarrow} T_k$, there exists a b'-partition, B_k such that for each $b_{k_i} \in B_k$ there is a matching move from U. Similarly, there is a b'-partition, B_l for each move $U \stackrel{d_j,x?}{\longrightarrow} U_l$. We have a set of n partitions $\{B_{k_1}, B_{k_2}, \ldots, B_{k_n}\}$ and m partitions $\{B_{l_1}, B_{l_2}, \ldots, B_{l_m}\}$, say. If $n = m = 0$ then we define $B_{b'}$ to be $\{b'\}$. Otherwise we consider all conjunctions of length $n + m$ whose conjucts are drawn one from each partition. Define $B_{b'} = \left\{ (\bigwedge_{i=1}^{n} b_i) \wedge (\bigwedge_{j=1}^{m} b_j) \mid b_i \in B_{k_i}, b_j \in B_{l_j} \right\}$. Then $\bigvee B_{b'} = b'$ and furthermore $B_{b'}$ enjoys the following property:

For each $b'' \in B_{b'}$ we have that $T \sim_n^{b''} U$ and whenever $T \stackrel{c_i,x?}{\longrightarrow} T'$ then there is a U' such that $U \stackrel{d_j,x?}{\longrightarrow} U'$ with $T' \sim_n^{b''} U'$ or $U \stackrel{DC(U),\alpha}{\longrightarrow} U$ with $T' \sim_n^{b''} U$. Similarly for U.

So we let $B = \bigcup_{b'\in B'} B_{b'}$ and consider the three cases which arise. Take $b'' \in B_{b'}$.

Case 1. There exists a $T \stackrel{c_i,x?}{\longrightarrow} T'$ such that for all $U \stackrel{d_j,x?}{\longrightarrow} U_{jl}$, $T' \not\sim_n^{b''} U_{jl}$.

Therefore $U \stackrel{DC(U),\alpha}{\longrightarrow} U$ and $T' \sim_n^{b''} U$ (and $b'' \models DC(U)$) since $T \sim_n^{b''} U$. We now show that $T \simeq_n^{b''} U + x?U$.

Recall that b'' corresponds to just one of the c_i and d_j, in the sense that $b''\wedge c_{i'} = b''\wedge d_{j'} = \mathbf{ff}$ whenever $i' \neq i, j' \neq j$. Therefore we need only consider moves of the form $T \stackrel{c_i,\alpha}{\longrightarrow}$ and $U \stackrel{d_j,\alpha}{\longrightarrow}$.

Suppose then that $T \stackrel{c_i,e!}{\longrightarrow} T_{ik}$. Since $T \sim_n^{b''} U$ we know there exists a b''-partition, B'', such that for each $b_1 \in B''$ there exists a $U \stackrel{d_j,e'!}{\longrightarrow} U_{jl}$ with $b_1 \models d_j$, $b_1 \models (e = e')$ and $T_{ik} \sim_n^{b_1} U_{jl}$. As $U + x?U \stackrel{d_j,e'!}{\longrightarrow} U_{jl}$ also, we have a match for $T \stackrel{c_i,e!}{\longrightarrow} T_{ik}$.

Suppose $T \stackrel{c_i,x?}{\longrightarrow} T_{ik}$. As $b'' \models d_j$ and $b'' \models DC(U)$ then $b'' \models \neg d_{j'}$ for each j' such that $?_U(j')$. Clearly then it cannot be the case that $?_U(j)$ holds because $b'' \models d_j$. Therefore no $l \in J_j$ and no variable x are such that β_{jl} is $x?$. Thus $U \stackrel{d_j,x?}{\not\longrightarrow}$ and therefore $U \stackrel{DC(U),\alpha}{\longrightarrow} U$ with $T_{ik} \sim_n^{b''} U$. Given this we use b'' to partition itself, $U + x?U \stackrel{\mathbf{tt},x?}{\longrightarrow} U$ being the matching move.

Suppose that $U + x?U \stackrel{d_j,e!}{\longrightarrow} U_{jl}$. Then, as before, we use the fact that $T \sim_n^{b''} U$ to get a matching partition and move. Suppose that $U + x?U \stackrel{d,x?}{\longrightarrow} U'$. By assumption, d is d_j or $\mathbf{tt}$. Clearly d cannot be d_j because, as we have already established, $U \stackrel{d_j,x?}{\not\longrightarrow} U'$. Thus d must be $\mathbf{tt}$ and U' must be U. Again, b'' partitions itself to get the matching move $T \stackrel{c_i,x?}{\longrightarrow} T_{ik}$.

Case 2. Symmetrical argument of case one which yields $T + x?T \simeq_n^{b''} U$ and $b' \models DC(T)$.

Case 3. Neither of the above.

That is, for every $T \stackrel{c_i,x?}{\longrightarrow} T_{ik}$ there exists a $U \stackrel{d_j,x?}{\longrightarrow} U_{jl}$ such that $T_{ik} \sim_n^{b''} U_{jl}$. Also, for every $U \stackrel{d_j,x?}{\longrightarrow} U_{jl}$ there exists a $T \stackrel{c_i,x?}{\longrightarrow} T_{ik}$ such that $T_{ik} \sim_n^{b''} U_{jl}$. It is easy to show then that $T \simeq_n^{b''} U$. □

We now see that Theorem 4.10 can be lifted to deal with the simpler notion of standard form. We also reinterpret the statement $b' \models DC(T)$ in terms of provability in the proof system.

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 $T \sim_n^b U$ 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 $T_i[z/x_i] \sim_n^{b'} U_j[z/y_j]$. 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

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

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 α-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

longer true. For example the agent $x? \in \{0,1\}.\mathbf{O}$ discards the value 3 but it can not discard either of the values 0 or 1.

We first consider the characterisation of closed agents by seeing what adjustments need to be made to the proof system of Section 3. The main difficulty is that in the presence of pattern matching agents the axiom schema *Noisy* is no longer sufficiently powerful. In $\mathcal{SA}$ this schema is based on the fact that for any process P which can discard values

$$P \simeq_n P + x?P.$$

In $\mathcal{SPA}$ the corresponding fact is that

$$P \simeq_n P + x \in S?P$$

provided that S does not contain any of the values which P can input. This set, $I(P)$, can be defined by structural induction on terms.

- $I(\mathbf{O}) = \emptyset$
- $I(e!P) = \emptyset$
- $I(x \in S?T) = S$
- $I(P + Q) = I(P) \cup I(Q)$
- $I(b \gg P) = \begin{cases} I(P) & \text{if } [\![b]\!] = \mathbf{tt} \\ \emptyset & \text{otherwise} \end{cases}$

Proposition 5.1 *For every agent P, $v \in I(P)$ if and only if $P \xrightarrow{v?}$.*

Proof. By structural induction on P. □

In addition to this we see that the two processes $x \in S?T + x \in S'?T$ and $x \in S \cup S'?T$ are strong noisy congruent, yet there is no way of showing this in the closed term proof system as there is no rules to manipulate pattern sets.

The actual changes we make to the proof system then are these: The axiom schema *Noisy* is replaced by the schema *P-Noisy*:

$$e!(P + x \in S?P) = e!P \qquad \text{if } S \cap I(P) = \emptyset.$$

The axiom *Pattern* is added,

$$x \in S?X + x \in S'?X = x \in S \cup S'?X$$

and the cl-INPUT rule is adapted to

$$\text{cl-P-INPUT} \quad \frac{\tau!T[v/x] + \tau!U[v/x] = \tau!U[v/x] \quad \text{for every } v \in S}{x \in S?T + x \in S'?U = x \in S'?U} \quad S \subseteq S'.$$

Notice that the condition $S \subseteq S'$ is essential. Without it we could prove, for instance, that for any terms T, U,

$$x \in \{0\}?T + x \in \{1\}?(x = 0 \gg T + U) = x \in \{1\}?(x = 0 \gg T + U)$$

which is in general not true. Also notice that, unlike the proof systems for $\mathcal{SA}$, summation is not required in this rule. With an early semantics it is often the case that without this

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,

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

and

$$x \in N?P + x \in N?\mathbf{O}$$

can not be proven congruent without the summation on the input rule. However we now have the use of the axiom *Pattern*, wherein lies the difference. In order to prove the above agents congruent we would first use *Pattern* to transform the latter agent into

$$x \in \{0,1\}?P + x \in \{2,3,\ldots\}?P + x \in \{0,1\}?\mathbf{O} + x \in \{2,3,\ldots\}?\mathbf{O},$$

use the modified Input rule, cl-P-INPUT, and then use *Pattern* again. This is similar to the technique used in [11] to give proof systems for early semantics for the π-calculus.

Finally, we also need the single axiom

$$x \in \emptyset?T = \mathbf{O},$$

which we call *Empty*.

Let $\mathcal{A}_P$ denote the set of axioms $\mathcal{A}$ augmented with the three axioms: *P-Noisy, Pattern* and *Empty*. Use $\mathcal{A}_P \vdash_{cl} P = Q$ to denote that $P = Q$ can be derived from these axioms using the proof system in Figure 2 but with the rule cl-P-INPUT instead of the rule cl-INPUT. We intend to prove soundness and completness of this modified proof system. However it is first necessary to present a result about closed terms analogous to Theorem 4.10 (Note that the depth of a patterned input prefix is, as expected $d(x \in S?T) = 1 + d(T)$).

Theorem 5.2 *Let $P, Q \in \mathcal{SPA}$ then*

$$P \sim_n Q \text{ iff } P + x \in (I(Q) - I(P))?P \simeq_n Q + x \in (I(P) - I(Q))?Q.$$

Moreover, when $I(Q) - I(P)$ and $I(P) - I(Q)$ are both non-empty there exist P', Q' such that $d(P') < d(P), d(Q') < d(Q)$ and $P' \sim_n P \sim_n Q \sim_n Q'$.

Proof. We outline the '$\Rightarrow$' direction. If $P \xrightarrow{v?} P'$ then we know there exists a Q' such that $Q \xrightarrow{v??} Q'$ with $P' \sim_n Q'$ because $P \sim_n Q$. If $v \in I(Q)$ then we know that $Q \xrightarrow{v?} Q'$. Otherwise $v \notin I(Q)$ and $Q \xrightarrow{v:} Q' (\equiv Q)$. In this case though $v \in I(P) - I(Q)$ which means that $x \in I(P) - I(Q)?Q \xrightarrow{v?} Q$ matches the move from P.

We know that $x \in I(Q) - I(P)?P \xrightarrow{v?} P$ whenever $v \in I(Q) - I(P)$. So we require a match from Q. $Q \xrightarrow{v?} Q'$ as $v \in I(Q)$ and $v \notin I(P)$ so $P \xrightarrow{v:} P$. We know then that $P \sim_n Q'$ necessarily because $P \sim_n Q$.

When the two sets are both non-empty we let $v_1 \in I(Q) - I(P)$, $v_2 \in I(P) - I(Q)$, then $P \xrightarrow{v_2?} P'$ for some P'. We know $v_2 \notin I(Q)$ so $Q \xrightarrow{v_2:} Q$ must match this move, that is, $P' \sim_n Q$. Similarly, we get $Q' \sim_n P$ using v_1. Transitivity of $\sim_n$ gives the result. □

Theorem 5.3 *(Soundness and Completeness) For all agents P, Q*

$$\mathcal{A}_P \vdash_{cl} P = Q \text{ if and only if } P \simeq_n Q$$

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 $T_j[v/x] \sim_n U_l[v/x]$ because $P \simeq_n Q$. Let $S_l^j = \{v \in S_j \cap S_l \mid U_l[v/x] \sim_n T_j[v/x]\}$. 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 $T_j[v/x] \sim_n U_l[v/x]$. We will show that

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

For convenience let P, Q denote $T_j[v/x], U_l[v/x]$ respectively. We now apply Theorem 5.2 to get

$$P + x \in U?P \simeq_n Q + x \in V?Q$$

Where $U = I(Q) - I(P)$ and $V = I(P) - I(Q)$. We have four cases to consider.

1. $U = V = \emptyset$
   Since $x \in \emptyset?T \simeq_n \mathbf{O}$ we can immediately conclude that $P \simeq_n Q$ and apply induction to obtain $\mathcal{A}_P \vdash_{cl} P = Q$ and therefore the required $\mathcal{A}_P \vdash_{cl} \tau!P = \tau!Q$.

2. $U = \emptyset, V \neq \emptyset$
   Here we have $P \simeq_n Q + x \in V?Q$. and again we can use induction to obtain $\mathcal{A}_P \vdash_{cl} \tau!P = \tau!(Q + x \in V?Q)$. Now we can apply the *P-Noisy* schema to obtain $\mathcal{A}_P \vdash_{cl} \tau!Q = \tau!(Q + x \in V?Q)$ from which the required result follows.

3. $U \neq \emptyset, V = \emptyset$
   Similar.

4. $U \neq \emptyset, V \neq \emptyset$
   Here we have $P + x \in U?P \simeq_n Q + x \in V?Q$ and in this case we can not apply induction immediately as the combined size of the terms has not decreased. But Thereom 5.2 tells us that there exists P', Q' such that $d(P') < d(P)$ and $d(Q') < d(Q)$ such that $P' \sim_n P$ and $Q' \sim_n Q$. Suppose without loss of generality that $d(P) \leq d(Q)$. Then, since $\tau!P \simeq_n \tau!P'$ we can use induction to obtain $\mathcal{A}_P \vdash_{cl} \tau!P = \tau!P'$. Then a simple application of the cl-P-INPUT rule gives $\mathcal{A}_P \vdash_{cl} x \in U?P = x \in U?P'$. This in turn implies that $P + x \in U?P' \simeq_n Q + x \in V?Q$ and here we can apply induction since the combined size has decreased. So we obtain, as before, $\mathcal{A}_P \vdash_{cl} \tau!(P + x \in U?P') = \tau!(Q + x \in V?Q)$. Using the fact that $\mathcal{A}_P \vdash_{cl} x \in U?P = x \in U?P'$ we obtain $\mathcal{A}_P \vdash_{cl} \tau!(P + x \in U?P) = \tau!(Q + x \in V?Q)$ from which the required $\mathcal{A}_{cl'} \vdash \tau!P = \tau!Q$ follows by two applications of the *P-Noisy* rule.

□

6 Pattern Matching with Open Terms

We look at adapting the open term proof system of Section 4 to deal with pattern matching on inputs. Reasoning as for the closed case we see that the modifications necessary are similar to those made in the previous section. Explicitly, we augment $\mathcal{A}$ with the axioms *Pattern* and *Empty*, modify the INPUT rule and axiom *Noisy*.
The INPUT rule becomes

$$\text{P-INPUT} \quad \frac{b \wedge x \in S \rhd \tau!T + \tau!U = \tau!U}{b \rhd x \in S?T + x \in S'?U = x \in S'?U} \quad \text{if } x \notin fv(b),\ S \subseteq S'$$

Note that, as before, we can derive a guarded version of this rule called P-INPUT$^{\gg}$ using Proposition 4.6.

What is to be done about axiom *Noisy*? In the previous section we presented *P-Noisy* using the construction $I(P)$, the set of values which a process can receive. Unfortunately there is not such a clean notion as this for open terms. We need to talk about the set of values a term can receive relative to a boolean world. We will denote this by $I(b, T)$. An obvious property which we will require $I(b, T)$ to satisfy is

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

This $I(b, T)$ is not a trivial notion to characterise. For example consider the agent

$$T \equiv b_0 \gg x \in S_1?T' + \neg b_0 \gg x \in S_2?T''.$$

What would be the set of values which T can receive in the boolean world $\mathbf{tt}$, say? In any evaluation we know that either b_0 or $\neg b_0$ will be satisfied. So for some evaluations we may have $I(T\rho) = S_1$ and for some we may have $I(T\rho) = S_2$. If $I(b, T)$ is to satisfy the property stated above then it clearly makes no sense to ask what $I(\mathbf{tt}, T)$ should be. The approach we take is to characterise the boolean expressions b for which $I(b, T)$ can have a meaningful definition which satisfies the required property.

Given a standard form

$$T \equiv \sum_{i \in I_!} b_i \gg e_i!T_i + \sum_{i \in I_?} b_i \gg x \in S_i?T_i$$

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 ρ 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 $T + x \in S?T \sim^b_{pn} T$. 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 $T\rho + x \in S?T\rho \sim_n T\rho$ 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 $T\rho[v/x] \sim_n U\rho[v/x]$. 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 $\left\{\sim^b_{pn}\right\}$. Once again we now use the definition of $\sim^b_{pn}$ to define $\simeq^b_{pn}$ the largest congruence contained in $\sim^b_{pn}$:

$T \simeq^b_{pn} U$ 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 $T, U \in \mathcal{SPA}$*

$$T \simeq^b_{pn} U \text{ iff } (\forall \rho \cdot \rho \models b \text{ implies } T\rho \simeq_n U\rho).$$

Proof. The 'only if' part is straightforward: given a symbolic noisy bisimulation $\{S^b\}$, we define $R = \{(T\rho, U\rho) \mid \exists b \cdot \rho \models b \text{ and } (T, U) \in S^b\}$ and show that this is a noisy bisimulation. For the 'if' part we suppose that we have a noisy bisimulation R and define

$$S^b = \{(T, U) \mid \rho \models b \text{ implies } (T\rho, U\rho) \in R\}.$$

We aim to show that $S^b \subseteq PNSB(S)^b$. Suppose that $T \xrightarrow{b', x \in S?} T'$. Let

$$\mathcal{U} = \left\{ U' \mid U \xrightarrow{b(U'), \alpha(U')} U', \alpha(U') \in \{x \in S'?, S' :\} \right\}$$

and let $U' \in \mathcal{U}$. Define $b^{aux}_{U'}$ such that $\rho \models b^{aux}_{U'}$ iff $(T'\rho, U'\rho) \in R$ and define

$$b_{U'} = b(U') \wedge x \in \mathbf{set}(\alpha(U')) \wedge b^{aux}_{U'}.$$

We let B be the set $\{b \wedge b' \wedge x \in S \wedge b_{U'} \mid U' \in \mathcal{U}\}$. It is clear that $\bigvee B \models b \wedge b' \wedge x \in S$ and we show the converse. Suppose $\rho \models b \wedge b' \wedge x \in S$. Then $\rho = \rho[v_0/x]$ for some $v_0 \in S$ and $(T\rho, U\rho) \in R$. By a simple generalisation of Proposition 4.3 we know that $T\rho \xrightarrow{v?} T'\rho[v/x]$, for any $v \in S$. So we know that there exists a matching move $U\rho \xrightarrow{v?} U_v\rho[v/x]$ or $U\rho \xrightarrow{v:} U_v\rho[v/x]$ such that $(T'\rho[v/x], U_v\rho[v/x]) \in R$.

By the same result we see there exists a $U_v \in \mathcal{U}$ such that $U \xrightarrow{b(U_v), \alpha(U_v)} U_v$ with $\rho \models b(U_v)$ and $v \in \mathbf{set}(\alpha(U_v))$. In particular we have this for v_0 and so $(T'\rho, U_{v_0}) \in R$. Which implies that $\rho \models b_{U_{v_0}}$. Therefore $\rho \models \bigvee B$.

This gives us the partition required. For each $b'' = b \wedge b' \wedge x \in S \wedge b_{U'} \in B$ we have $U \xrightarrow{b(U'), \alpha(U')} U'$ with $b'' \models b(U')$, $b'' \models x \in \mathbf{set}(\alpha(U'))$ and $b'' \models b^{aux}_{u'}$. This implies that $(T'\rho, U'\rho) \in R$, hence $(T', U') \in S^{b''}$.

The symbolic transmissions from T can be dealt with in a similar, slightly simpler, manner. □

Having developed a suitable notion of patterned symbolic bisimulation we develop a version of Theorem 5.2 for open terms.

Theorem 6.4 *If T and U are standard forms then $T \sim^b_{pn} U$ if and only if there exists a b-partition, B, such that each $b' \in B$ is both T and U-uniform and*

$$T + z \in S?T \simeq^{b'}_{pn} U + z \in S'?U$$

where $z \notin fv(b, T, U)$, $S = (I(b', U) - I(b', T))$ and $S' = (I(b', T) - I(b', U))$. Moreover, when both S and S' are non-empty, there exist T', U' such that $d(T') < d(T), d(U') < d(U)$ and $T' \sim^{b'}_{pn} T, U' \sim^{b'}_{pn} U$.

Proof. We use Lemma 6.1 We exhibit two b-partitions, B_1, B_2 such that each $b_1 \in B_1$ is T-uniform and each $b_2 \in B_2$ is U-uniform. These are simply $B_1 = \{b \wedge b_K \mid K \subseteq I_?\}$ and $B_2 = \{b \wedge b_L \mid L \subseteq J_?\}$ where $I_?$, $J_?$ are the indexing sets of $T_?$ and $U_?$ respectively. We let $B = \{b_1 \wedge b_2 \mid b_1 \in B_1, b_2 \in B_2\}$. By Lemma 6.1 B has the property that for each $b' \in B$, $\rho \models b'$ implies $I(T\rho) = I(b', T)$ and $I(U\rho) = I(b', U)$. Suppose $\rho \models b'$. Then $T\rho \sim_n U\rho$. We apply Theorem 5.2 to get

$$T\rho + z \in S?T\rho \sim_n U\rho + z \in S'?U\rho$$

where $S = I(U\rho) - I(T\rho)$ and $S' = I(T\rho) - I(U\rho)$. By properties of the partition we know that $S = I(b', U) - I(b', T)$ and $S' = I(b', T) - I(b', U)$. This is true for each $\rho \models b'$ so we have that $T + z \in S?T \sim^{b'}_{pn} U + z \in S'?U$.

What remains to be proved is the existence of T', U', in the case where S, S', are non-empty. This also follows from Theorem 5.2, save for a mild complication. For each $\rho \models b'$ there is a P', Q' such that $P' \sim_n T\rho$ and $Q' \sim_n U'\rho$. These P', Q' are obtained by considering two values $v_1 \in S, v_2 \in S'$ and using the fact that $T\rho \xrightarrow{v_2?} P'$ and $U\rho \xrightarrow{v_1?} Q'$. This means that P' is of the form $(T_i[v_2/x])\rho$ and similarly Q' is of the form $(U_j[v_1/x])\rho$. Let T'_ρ denote this $T_i[v_2/x]$ and let U'_ρ denote $U_j[v_1/x]$. For different ρ these T'_ρ, U'_ρ may be different although there are only finitely many such agents.

To resolve this we partition b' further. Define

$$\Sigma_{ij} = \left\{\rho \mid \rho \models b' \text{ and } T'_\rho = T_i[v_2/x] \text{ and } U'_\rho = U_j[v_1/x]\right\}$$

and define $\rho \models b_{ij}$ if and only if $\rho \in \Sigma_{ij}$. These $\{b_{ij}\}$ partition b' and for each $\rho \models b_{ij}$ we have terms T', U' such that $T'\rho \sim_n T\rho$ and $U'\rho \sim_n U\rho$. □

Theorem 6.5 *(Completeness)* $T \simeq^b_{pn} U$ *implies* $\mathcal{A}_P \vdash b \rhd T = U$.

Proof. As before we adopt standard forms for T, U and proceed by induction on the sum of the depths, $d(T) + d(U)$. Again it is sufficient to show

$$\mathcal{A}_P \vdash b \rhd T_! = U_! \text{ and } \mathcal{A}_P \vdash b \rhd T_? = U_?$$

independently. We concentrate on the latter though we must first modify the standard forms slightly. Suppose

$$T_? \equiv \sum_{I_?} c_i \gg x_i \in S_i?T_i$$

where $z \notin fv(b, T, U)$. Then we have (for each i) that $T \xrightarrow{c_i, z \in S_i?} T_i[z/x_i]$. Since $T \simeq^b_{pn} U$ we know that there exists a matching $b \wedge c_i \wedge z \in S_i$-partition, B. Because $z \notin fv(b, c_i)$ we know that each element of B is logically equivalent to something of the form $b' \wedge z \in S_{i_k}$ (for some indexing set K) where $\bigvee b' \equiv b \wedge c_i$ and $\bigcup S_{i_k} = S_i$. We use the axiom *Pattern* to decompose the summand $x_i \in S_i?T_i$ of T into the sum $\sum_{k \in K} x_i \in S_{i_k}?T_i$ and Proposition 4.6 to distribute c_i across this sum. We repeat this for each $i \in I_?$ and also for U.

Having done this we show $\mathcal{A}_P \vdash b \rhd T_? = U_?$ where $T_? \equiv \sum_I c_i \gg x_i \in S_i?T_i$ and $U_? \equiv \sum_J d_j \gg x_j \in S_j?U_j$. Let $U^z_?$ denote the term

$$\sum_{j \in J_?} d_j \gg z \in S_j?U_j[z/y_j].$$

It is sufficient to show

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

for each $i \in I?$. For once we have obtained this we can apply GUARD and add to get

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

Repeating this argument we obtain a symmetric version of this which results in $\mathcal{A}_P \vdash b \rhd \sum_{i \in I} c_i \gg z \in S_i?T_i[z/x_i] = U^z_?$. The final result follows by α-conversion.

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 $T_i[z/x_i] \sim^{b'}_{pn} U_j[z/y_j]$. The fact that $b' \wedge z \in S_i \models z \in S_j$ gives us that $S_i \subseteq S_j$. This will witness the side condition of the P-INPUT rule.

By Theorem 6.4 we get a $b' \wedge z \in S_i$-partition, B', such that for each $b'' \in B'$ we have that b'' is both $T_i[z/x_i]$ and $U_j[z/y_j]$-uniform and that

$$T_i[z/x_i] + x \in S?T_i[z/x_i] \simeq^{b''}_{pn} U_j[z/y_j] + x \in S'?U_j[z/y_j]$$

where S is $I(b'', U_j[z/y_j]) - I(b'', T_i[z/x_i])$ and S' is $I(b'', T_i[z/x_i]) - I(b'', U_j[z/y_j])$. Once again we have four cases to consider. In each of which we demonstrate

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

1. $S = S' = \emptyset$.
   Since $x \in \emptyset?T \simeq^{\mathbf{tt}}_{pn} \mathbf{O}$ we conclude that $T_i[z/x_i] \simeq^{b''}_{pn} U_j[z/y_j]$ and apply induction to obtain $\mathcal{A}_P \vdash b'' \rhd T_i[z/x_i] = U_j[z/y_j]$. Whence $\mathcal{A}_P \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$.

2. $S = \emptyset, S' \neq \emptyset$.
   We have $T_i[z/x_i] \simeq^{b''}_{pn} U_j[z/y_j] + x \in S'?U_j[z/y_j]$. Again we apply induction and use axiom *Noisy* to obtain $\mathcal{A}_P \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$.

3. $S \neq \emptyset, S' = \emptyset$.
   Similar.

4. $S \neq \emptyset, S' \neq \emptyset$.
   By Theorem 6.4 there exists T', U' such that $T_i[z/x_i] \sim^{b''}_{pn} T'$ and $U_j[z/y_j] \sim^{b''}_{pn} U'$. Without loss of generality, suppose that $d(T) \leq d(U)$. By induction we get $\mathcal{A}_P \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!T'$. Whence $\mathcal{A}_P \vdash b'' \rhd x \in S?T_i[z/x_i] = x \in S?T'$ by P-INPUT. We can deduce that
   $$T_i[z/x_i] + x \in S?T' \simeq^{b''}_{np} U_j[z/y_j] + x \in S'?U_j[z/y_j]$$
   and see that induction is applicable here also. Therefore we get
   $$\mathcal{A}_P \vdash b'' \rhd T_i[z/x_i] + x \in S?T' = U_j[z/y_j] + x \in S'?U_j[z/y_j].$$
   Using the previous result we can substitute T' for $T_i[z/x_i]$ then apply TAU and axiom *Noisy* twice to get $\mathcal{A}_P \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$ as required.

So we have seen that for each $b'' \in B'$ we can prove $\mathcal{A}_P \vdash b'' \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$ so we apply CUT to give $\mathcal{A}_P \vdash b' \wedge z \in S_i \rhd \tau!T_i[z/x_i] = \tau!U_j[z/y_j]$. We know that $b' \models c_i \wedge d_j$ and we can use Proposition 4.6 to give

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

This facilitates application of P-INPUT$^{\gg}$ to give

$$\mathcal{A}_P \vdash b' \rhd d_j \gg z \in S_j?U_j[z/y_j] = d_j \gg z \in S_j?U_j[z/y_j] + c_i \gg z \in S_i?T_i[z/x_i].$$

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

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

This is true for each b' so we can apply CUT to give

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

□

7 Finite *CBS*

Having now dealt with the pattern matching construct it remains only to to consider the parallel and translation constucts in order to have a proof system for all of finite *CBS*. The standard approach is to introduce axioms or axiom schemas which are sufficient to translate agents of finite CBS into agents of $\mathcal{SPA}$. The parallel operator is usually treated using an expansion theorem while translations, being generalisations of the restriction and renaming operators of *CCS*, can be handled by a set of axiom schemas which when used as rewrite rules can reduce a term of the form $P_{(f,g)}$, where $P \in \mathcal{SPA}$, to a term in $\mathcal{SPA}$. We give a brief outline of the necesary axiom schemas but leave much of the details to the reader.

The expansion theorem presented in [14] is not sufficient here as we need to deal with the pattern sets. However a suitable adaption of this is obtained:

Proposition 7.1 *Suppose* $fv(T) \cap bv(U) = fv(U) \cap bv(T) = \emptyset$ *and*

$$T \equiv \sum_{i \in I} w_i!T_i + \sum_{j \in J} x \in S_j?T_j$$

and

$$U \equiv \sum_{k \in K} w_k!U_k + \sum_{l \in L} x \in S_l?U_l.$$

Then, writing $\bar{S_L} \stackrel{def}{=} \bigcap_{l \in L}(Val - S_l)$, *writing* $L_{w_i} \stackrel{def}{=} \{l \in L \mid w_i \in S_l\}$ *and writing* $\bar{I_L} \stackrel{def}{=} \{i \in I \mid w_i \in \bar{S_L}$ *or* $w_i = \tau\}$ *we have*

$$\begin{aligned} T \mid U \simeq_n & \sum_{i \in I, l \in L_{w_i}} w_i!(T_i \mid U_l[w_i/x]) + \\ & \sum_{k \in K, j \in J_{w_k}} w_k!(T_j[w_k/x] \mid U_k) + \\ & \sum_{i \in \bar{I_L}} w_i!(T_i \mid U) + \\ & \sum_{k \in \bar{K_J}} w_k!(T \mid U_k) + \\ & \sum_{j \in J, l \in L} x \in S_j \cap S_l?(T_j \mid U_l) + \\ & \sum_{j \in J} x \in (S_j \cap \bar{S_L})?(T_j \mid U) + \\ & \sum_{l \in L} x \in (S_l \cap \bar{S_J})?(T \mid U_l). \end{aligned}$$

Proof. Sufficient to prove that $P|Q \stackrel{\alpha}{\longrightarrow} P'$ if and only if the $RHS \stackrel{\alpha}{\longrightarrow} P'$ for any closed instantions P, Q of T, U respectively. This can be proved directly from the operational semantics; the details are left to the reader. □

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 an axiom schema, which we call EXP, while TRANS is used to denote the obvious axiom schemas underlying the above encoding; each line gives rise to a separate axiom schema.

Theorem 7.3 *For any two terms of finite CBS, T and U,*

$$T \simeq_n U \text{ iff } \mathcal{A}_P, EXP, TRANS \vdash T = U.$$

Proof. Use soundness of translations along with soundness and completeness results of the previous section. □

A similar result can be obtained for the proof system for closed terms, but we leave the details to the reader.

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 Express Workshop, CWI*, 1994.

[11] J. Parrow and D. Sangiorgi. Algebraic theories for name-passing calculi. Technical Report ECS-LFCS-93-262, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1993.

[12] A. Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In Wilfried Brauer, editor, *Proceedings 12^{th} ICALP,* Nafplion, volume 194 of *Lecture Notes in Computer Science*, pages 15–32. Springer-Verlag, July 1985.

[13] K.V.S. Prasad. A calculus of broadcast systems. In *TAPSOFT 91 Volume 1: CAAP.* Springer Verlag, 1991.

[14] K.V.S. Prasad. A calculus of value broadcasts. Technical report, Dept. of Computer Science, Chalmers, 1992.

[15] K.V.S. Prasad. Programming with broadcasts. In Best [2].

[16] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms.* PhD thesis, University of Edinburgh, 1993.

[17] Martin Weichert. Algebra of communicating systems. In *Wintermötet Tanum Strand, draft proceedings*, 1993.