A Modal Characterisation of Distributed Bisimulation

Søren Christensen*

Computer Science Department
Aarhus University, Ny Munkegade 116
DK-8000 Aarhus C.
Denmark

Marts 1991

Abstract

In this paper we consider the *distributed bisimulation equiualence* defined by Hennessy and Castellani in [HC88] and later developed by Castellani in [Cas88]. We present a logic in the style of Hennessy-Milner logic to characterise the equivalence, i.e. we seek a logic such that whenever two processes are distributed bisimulation equivalent, they satisfy the same set of formulae and vice versa.

Furthermore, for a small subset of CCS we provide a proof system which is shown to be sound and complete. The proof system is structural both in the structure of formulae and in the structure of processes. For the case of parallel composition of processes we present inference rules defined via a new combinator introduced. The combinator in question is *left merge*, a special kind of parallel composition in which the left operand has precedence over the other and must perform the first action observed.

*The author gratefully acknowledges financial support from the Danish Research Academy.

1 Introduction

In [HC88, Cas88] Hennessy and Castellani define an equivalence on CCS processes ba,sed on the well-known bisimulation technique [Mil89]. While ordinary bisimulation models independent actions via interleaving their equivalence, which is called *distributed bisimulation*, does not.

The distributed bisimulation is defined on a class of transition systems in which states are augmented with information about the distribution of processes in space. That is, in observing an event the resulting state will contain process components representing distribution of the system.

The transition systems to be considered are called *distributed transition systems*. A transition has the form $p \xrightarrow{a} \langle p', p'' \rangle$ where a is the atomic action observed. The state $\langle p', p'' \rangle$ is called the *compound residual* and it contains information about the distribution of processes on observing the action a. In [HC88], p' is called the *local residual* and p'' the *global residual*. Intuitively, p' is the local process at which the action a took place whereas p'' is the global result after the action a has been observed. For instance, in the framework of CCS we could have $a.p|q \xrightarrow{a} \langle p, p|q \rangle$. In [Cas88] a different viewpoint is taken: in the compound residual $\langle p', p'' \rangle$, p' is again the local residual but p'' is now called the *concurrent residual*. Intuitively, the concurrent residual is the part of the global residual which behave independently of the local residual. For instance, we could have $a.p|q \xrightarrow{a} \langle p, q \rangle$. In [Cas88] it is shown that those two interpretations of the compound residual leads to the same distributed bisimulation equivalence. In this paper we consider the interpretation given in [Cas88], i.e. the compound residual consists of a local and a concurrent residual.

In [HC88] the main effort has been to give an algebraic characterisation of the distributed bisimulation equivalence. In this paper we present a logic, in the style of Hennessy-Milner logic [HM85], to characterise distributed bisimulation equivalence. That is, whenever two processes are distributed bisimulation equivalent they satisfy the same set of formulae and vise versa.

The new logic is based on the modality of necessity, often described by a box ($[]$), and on the modality of possibility, often described by a diamond ($\langle\rangle$). But the modalities will now be *dyadic* thus obtaining formulae both for the local and the concurrent residual of a compound residual.

Furthermore, in this paper we consider a small subset of CCS; the combi-

nators are *prefix*, *sum* and *parallel composition*. Moreover, communication between processes is not allowed. The behaviour of processes contained in this language will be interpreted via distributed transition systems thus we can use our logic as a specification language for the processes.

For this subset of CCS we have defined a sound and complete proof system in the style of [Sti85]. The proof system is structural both in the structure of formulae and in the structure of processes. Concerning the parallel combinator we have obtained simple inference rules via a new combinator introduced. The combinator is *left merge*, denoted $\lfloor$. The combinator can be considered as a special kind of parallel composition in which the left process has precedence over the right process and must perform the first action observed.

In section 2 we define the distributed transition systems and the distributed bisimulation equivalence. The section is based on [Cas88]. In section 3 we define the logic and show the characterisation theorem. In section 4 we present the proof system which is shown to be complete in appendix A. Finally we finish the paper with a conclusion summing up the results and suggesting topics for future work.

2 Distributed Bisimulation

We begin by defining the transition systems to be considered in this paper and which form the basis for defining the semantics of processes. The transition systems we are interested in is based on the usual notion of transition systems. But, as explained in the introduction, each transition gives rise to a *compound residual* $\langle p, q\rangle$ consisting of the *local residual* p and the *concurrent residual* q. Intuitively, p and q are two independent subprocesses which are placed at different localities.[1] Based on these ideas we introduce the notion of *distributed transition systems*.

Definition 2.1 A *distributed transition system* is a triple $(\mathcal{P}, \mathcal{A}, \rightarrow)$ where

(i) $\mathcal{P}$ is a set of processes,

(ii) $\mathcal{A}$ is a set of *actions*, and

[1] We refer to [Cas88] for a more thorough explanation of the nature of the distribution of processes in space.

(iii) $\rightarrow$ is a relation contained $\mathcal{P} \times \mathcal{A} \times \mathcal{P} \times \mathcal{P}$ called the *transition relation.*

If $(p, a, p', p'') \in \rightarrow$ we will write $p \stackrel{a}{\rightarrow} \langle p', p'' \rangle$. When $p \stackrel{a}{\rightarrow} \langle p', p'' \rangle$ it is supposed to reflect that the process p can perform the action a and then become the compound residual $\langle p', p'' \rangle$. ■

The language to be considered in this paper is a small subset of CCS [Mil89]; the combinators are *prefix*, *sum* and *parallel composition*. We presuppose a set $\mathcal{A}$ of actions and let $a, b, c, \ldots$ with or without subscript range over $\mathcal{A}$. We assume no synchronisation between subprocesses, hence there is no structure on $\mathcal{A}$.

Definition 2.2 We let $\mathrm{CCS_S}$ denote the set of processes. Suppose $a \in \mathcal{A}$. Then $\mathrm{CCS_S}$ is the least set satisfying the following rules:

(i) $nil \in \mathrm{CCS_S}$, and

(ii) if $p, q \in \mathrm{CCS_S}$ then $a.p, p + q, p|q \in \mathrm{CCS_S}$.

The operator $(a.)$ is called prefix, the operator $(+)$ is called sum and finally the operator $(|)$ is called parallel composition. We let $p, q, r, \ldots$ with or without quotes and with or without subscript range over $\mathrm{CCS_S}$. ■

We assume some rules in order to improve readability of processes. We will often let ap be an abbreviation of the process $a.p$. Furthermore, nil is omitted and prefix has precedence over parallel composition which in turn has precedence over sum. Thus the process $(a.nil) + ((b.nil)|(c.nil))$ is similar to $a + b|c$.

Roughly, the interpretation of the combinators are as follows: $a.p$ is the process which can perform the action a and then behave as the process p; $p + q$ is the process which behaves either as p or as q; finally, $p|q$ is the process consisting of two independent processes, thus performing concurrently. But moreover, in observing the behaviour of processes we wish to obtain information about the distribution of processes in space. For instance, in observing the action a at the process $a.p|p$ we wish to know about the local process at which the action a took place, i.e. at the process p, and we wish to know about the possible processes independent of the process at which a took place, i.e. the process q in our example. Related to the distributed transition systems this information could be contained in the compound residual $\langle p, q \rangle$.

Based on these ideas we give the formal definition of the semantics of processes in $\mathrm{CCS_S}$.

Definition 2.3 The semantics of processes in $\mathrm{CCS_S}$ is captured through the distributed transition system $(\mathrm{CCS_S}, \mathcal{A}, \rightarrow)$ where $\rightarrow$ is the least relation obeying the following rules in which the relation below the line is to be inferred from that above the line.

(i)

$$\frac{}{a.p \xrightarrow{a} \langle p, nil \rangle}$$

(ii)

$$\frac{p \xrightarrow{a} \langle p', p'' \rangle}{p + q \xrightarrow{a} \langle p', p'' \rangle} \qquad \frac{q \xrightarrow{a} \langle q', q'' \rangle}{p + q \xrightarrow{a} \langle q', q'' \rangle}$$

(iii)

$$\frac{p \xrightarrow{a} \langle p', p'' \rangle}{p|q \xrightarrow{a} \langle p', p''|q \rangle} \qquad \frac{q \xrightarrow{a} \langle q', q'' \rangle}{p|q \xrightarrow{a} \langle q', p|q'' \rangle}$$

The behaviour of a particular process $p \in \mathrm{CCS_S}$ is captured in the transition system $(\mathrm{CCS_S}, \mathcal{A}, \rightarrow)$ by letting the start state be p itself. ■

Rules **(i)**, **(ii)** and **(iii)** define the behaviour of prefix, sum and parallel composition respectively. Rule **(i)** states that the process $a.p$ can perform the action a. The result is $\langle p, nil \rangle$, i.e. there is nothing which can happen in parallel with the action a. Rule **(ii)** is the usual interpretation of the sum operator. Rule **(iii)** states the behaviour of the parallel process $p|q$. Note the effect on the concurrent residual.

Although we have defined the behaviour of processes in $\mathrm{CCS_S}$ via a distributed transition system it is not at all clear how a process computes; it seems as if only the first step of processes can be observed. In order to be able to define computations of processes we extend the transition relation by adding the following rules for pairs $\langle p, q \rangle$ of processes.

(iv)

$$\frac{p \xrightarrow{a} \langle p', p'' \rangle}{\langle p, q \rangle \xrightarrow{a} \langle p', p''|q \rangle} \qquad \frac{q \xrightarrow{a} \langle q', q'' \rangle}{\langle p, q \rangle \xrightarrow{a} \langle q', p|q'' \rangle}$$

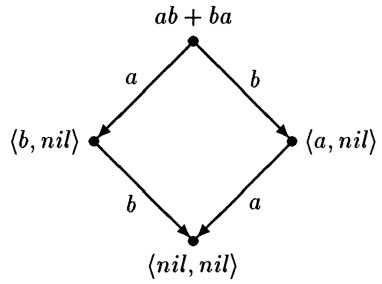
According to these rules, after each transition $p \xrightarrow{a} \langle p', p'' \rangle$ the execution resumes with the composition of the two residuals; the pair $\langle p', p'' \rangle$ has exactly the same behaviour as the process $p'|p''$.

Given the transition rules for the compound residuals we can extend the transition relation to sequences of actions. Let $t = a_1 a_2 \ldots a_n \in \mathcal{A}^*$ be a sequence of actions. Assume p, p' and p'' are $\mathrm{CCS_S}$ processes. Then $p \xrightarrow{t} \langle p', p'' \rangle$ iff there exists processes $p_1, q_1, \ldots, p_{n-1}, q_{n-1} \in \mathrm{CCS_S}$ such that $p \xrightarrow{a_1} \langle p_1, q_1 \rangle \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} \langle p_{n-1}, q_{n-1} \rangle \xrightarrow{a_n} \langle p', p'' \rangle$. If there is no transition from $\langle p', p'' \rangle$ then t is called a computation. Finally, by $p \rightarrow^* \langle p', p'' \rangle$ we denote that there exists a sequence t of actions such that $p \xrightarrow{t} \langle p', p'' \rangle$.

From the above rules it follows that the concurrent residual in $p \rightarrow^* \langle p', p'' \rangle$ is a process of the form $p_1| \cdots |p_n$ where some of the processes p_i equals nil and have been inserted by using rule **(i)** while all the others have been introduced by successive applications of rule **(iii)** and **(iv)**.

Example 2.4 Consider the $\mathrm{CCS_S}$ process $ab+ba$. The computation steps for this process, given via the extended transition relation, are shown in figure 1 below.

Figure 1

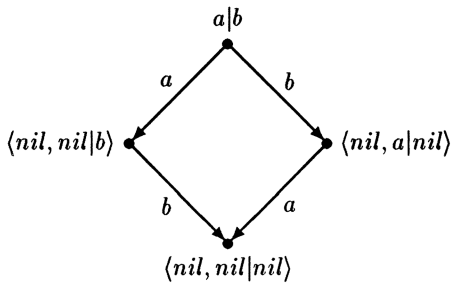


Likewise, the computation steps of the $\mathrm{CCS_S}$ process $a|b$ are given in figure 2 below.

Note that, although the process $ab + ba$ is based on non-determinism while $a|b$ is based on parallelism there is no observable difference between the two processes when the observation is based on the associated transition systems.

■

Figure 2



We now move on to define a relation between processes of $\mathcal{P}$. If $p, q \in \mathcal{P}$, the relation between p and q will be defined through the transition systems for p and q. The technique for defining the relation is closely related to the bisimulation technique [Mil89] but will be based on the information contained in the individual residuals. Thus the bisimulation relates the local and concurrent residuals separately.

Definition 2.5 The relation $R \in \mathcal{P} \times \mathcal{P}$ is a *distributed bisimulation* provided for all $(p, q) \in R$ the following is satisfied:

(i) $p \xrightarrow{a} \langle p', p'' \rangle$ implies $q \xrightarrow{a} \langle q, q'' \rangle$ such that $(p', q') \in R$ & $(p'', q'') \in R$, and

(ii) $q \xrightarrow{a} \langle q', q'' \rangle$ implies $p \xrightarrow{a} \langle p, p'' \rangle$ such that $(p', q') \in R$ & $(p'', q'') \in R$.

■

Definition 2.6 Let $p, q \in \mathcal{P}$. Then $p \sim_d q$ iff there exists a distributed bisimulation R such that $(p, q) \in R$. If $p \sim_d q$ we call p and q distributed bisimulation equivalent. ■

Example 2.7 We have $ab + ha \not\sim_d a|b$. If the two processes were distributed bisimulation equivalent then according to the definition of $\sim_d$ and the transition rules for $ab + ba$ and $a|b$ we must relate $nil|b$ and nil by a distributed bisimulation. This is impossible as the former can do a b action which the latter cannot. ■

We refer to [Cas88] and [HC88] for a thorough investigation of the relation $\sim_d$. In particular, the cited works contain a complete axiomatisation of $\sim_d$ and a comparison with other notions of equivalences between processes including

the bisimulation $\sim$.[2] The purpose of this paper is to define a logical language in the framework of Hennessy-Milner logic [HM85]. The logical language is required to characterise the distributed bisimulation just as Hennessy-Milner logic characterises the bisimulation equivalence $\sim$. In the next section we define the logic and subsequently show the characterisation theorem.

3 The Modal Characterisation

In [HM85] Hennessy and Milner define a modal logic providing a characterisation of bisimulation on finitely branching transition systems. They show that whenever two processes are bisimular they satisfy the same set of formulae and vice versa. Basically, the logic contains two modalities: the *necessity* modality, often described by a box ($[]$), and the *possibility* modality, often described by a diamond ($\langle\rangle$).

We aim to provide a logic in the style of Hennessy-Milner logic to characterise the distributed bisimulation equivalence. It turns out that the new logic also can be based on the necessity and possibility modalities. But whereas the modalities of the ordinary Hennessy-Milner logic are monadic we now define the modalities as dyadic operators thus obtaining formulae for the local and concurrent residual of a compound residual.

We first present the syntax of the logic.

Definition 3.1 We let $\mathcal{L}$ denote the set of formulae to be considered. Assume $a \in \mathcal{A}$. Then $\mathcal{L}$ is defined as the least set satisfying the following rules:

(i) $tt \in \mathcal{L}$,

(ii) $ff \in \mathcal{L}$, and

(iii) if $\alpha, \beta \in \mathcal{L}$ then $\alpha \wedge \beta, \alpha \vee \beta \langle a \rangle(\alpha, \beta), [a](\alpha, \beta) \in \mathcal{L}$.

We let $\alpha, \beta, \gamma, \ldots$ range over $\mathcal{L}$. ■

We proceed by defining the semantics of the logic.

Definition 3.2 Let $p \in \mathcal{P}$ and $\alpha \in \mathcal{L}$. Finally, let $p \models \alpha$ denote that p satisfies α. The relation $\models$ is defined by structural induction on the structure of α.

[2] As perhaps indicated by the previous example $\sim_d$ is strongly contained in $\sim$.

(i) $\forall p \in \mathcal{P} : p \models tt$,

(ii) $\forall p \in \mathcal{P} : p \not\models ff$,

(iii) $p \models \alpha \wedge \beta$ iff $p \models \alpha$ & $p \models \beta$,

(iv) $p \models \alpha \vee \beta$ iff $p \models \alpha$ or $p \models \beta$,

(v) $p \models \langle a \rangle(\alpha, \beta)$ iff $\exists p', p'' : p \xrightarrow{a} \langle p', p'' \rangle$ & $p' \models \alpha$ & $p'' \models \beta$, and

(vi) $p \models [a](\alpha, \beta)$ iff $\forall p', p'' : p \xrightarrow{a} \langle p', p'' \rangle$ implies $p' \models \alpha$ or $p'' \models \beta$.

■

Rule **(i)** and **(ii)** state the interpretation of the atomic formulae tt and ff. The formula tt stands for true and every process satisfies true, whereas the formula ff stands for false and no process satisfies false. Rule **(iii)** and **(iv)** state the usual interpretation of the logical connectives $\vee$ and $\wedge$. Rule **(v)** states the interpretation of the possibility modality. The relationship $p \models \langle a \rangle(\alpha, \beta)$ expresses intuitively that it is possible for the process p to do an a action whereupon the local residual will satisfy α and the concurrent residual will satisfy β. Finally, rule **(vi)** states the interpretation of the necessity modality. The relationship $p \models [a](\alpha, \beta)$ expresses intuitively that whenever the process p performs an a action the local residual will satisfy α or the concurrent residual will satisfy β.

Note that we have avoided negation as a logical connective in our logic. This is because the proof system, to be developed in the next section, can be defined without referring to any negative deduction if negation is avoided as a logical connective; there are no use of rules indicating when a process will *not* satisfy a particular formula. If negation was available as a logical connective we could have defined either tt by ff or vice versa, we could have defined either $\vee$ by $\wedge$ or vice versa, and finally we could have defined one of the modalities as the dual of the other.[3] Thus, in order to rule out negation as a connective we have to introduce three new symbols into our logic if expressiveness shall be preserved. But we are willing to pay this price in order to support the development of the proof system.

[3]If $\sim$ denotes negation with the obvious semant,ical definit,ion then by duality we mean that $\langle \alpha \rangle(\alpha, \beta)$ can be defined as $\sim [a](\sim \alpha, \sim \beta)$ or that $[a](\alpha, \beta)$ can be defined as $\sim \langle a \rangle(\sim \alpha, \sim \beta)$.

Observe that by the semantical definition of the possibility modality, i.e. definition 3.2 **(v)**, we require that the local residual satisfies α *and* that the concurrent residual satisfies β. On the other hand, according to the semantical definition of the necessity modality, i.e. definition 3.2 **(vi)**, we only require that the local residual satisfies a or that the concurrent residual satisfies β. Because of the duality we want to obtain between the possibility and necessity modality we are required to have 'and' at one of the modalities and 'or' at the other.[4] But what will happen if we exchange the two interpretations, i.e. what happens if we have 'or' in connection with the possibility modality and 'and' in connection with the necessity modality? The answer is that by this interpretation we get a logic which is weaker, viz. there exist processes which are not distributed bisimulation equivalent but cannot be distinguished by formulae of the logic. In appendix B we discuss this interpretation of the two modalities; in particular we show that this new logic is weaker that the logic $\mathcal{L}$ discussed here.

Example 3.3 By the relationship $a|b \models \langle a\rangle(tt, \langle b\rangle(tt, tt))$ we specify that the process $a|b$ is capable of performing an a action whereupon the concurrent residual can do a b action. Note that the relationship is satisfied as can be checked by the definition of $\models$ and the transition rules for $a|b$ shown in figure 2. Also note that $ab+ba \models \langle a\rangle(tt, \langle b\rangle(tt, tt))$ is not satisfied because whenever $ab + ba$ performs an a action there is no concurrent residual which can do a b action (see figure 1). ■

As observed by the last example we could distinguish between the two non distributed bisimulation equivalent processes $a|b$ and $ab + ba$ by a formula of our logic. An interesting question is whether non distributed bisimulation equivalent processes can be distinguished by formulae of $\mathcal{L}$ and vice versa. Before we answer this question we define a function on formulae which transforms a formula to its dual formula.

Definition 3.4 Let $D : \mathcal{L} \to \mathcal{L}$ be a transformation on formulae defined as follows:

(i) $D(tt) = ff$,

(ii) $D(ff) = tt$,

[4]The duality between the two modalities is needed in the proof of the characterisation theorem.

(iii) $D(\alpha \wedge \beta) = D(\alpha) \vee D(\beta)$,

(iv) $D(\alpha \vee \beta) = D(\alpha) \wedge D(\beta)$,

(v) $D(\langle a \rangle(\alpha, \beta)) = [a](D(\alpha), D(\beta))$ and

(vi) $D([a](\alpha, \beta)) = \langle a \rangle(D(\alpha), D(\beta))$.

Lemma 3.5 The transformation D satisfies the principle of duality, i.e.:

$$\forall \alpha \in \mathcal{L}, \forall p \in \mathcal{P} : p \models \alpha \Leftrightarrow p \not\models D(\alpha)$$

Proof The proof proceeds by structural induction on the structure of formulae. We omit the details. ■

We now prove that distributed bisimulation equivalent processes from $\mathcal{P}$ satisfy the same set of formulae from $\mathcal{L}$ and vice versa under the condition that we restrict our class of processes to so-called finitely branching processes, i.e. processes which have only finitely many a derivatives for each action a.

Theorem 3.6 Assume $p, q \in \mathcal{P}$ such that p and q are finitely branching processes. Then

$$p \sim_d q \Leftrightarrow (\forall \alpha \in \mathcal{L} : p \models \alpha \Leftrightarrow q \models \alpha)$$

Proof ($\Rightarrow$): Assume $p \sim_d q$. By induction on the structure of α we show $p \models \alpha \Leftrightarrow q \models \alpha$.

(i) $\alpha = tt$. By definition of $\models$ we have $p \models tt$ iff $q \models tt$.

(ii) $\alpha = ff$. Once again the required result follows from the definition of $\models$.

(iii) $\alpha = \beta \wedge \gamma$. We have $p \models \alpha$ iff $p \models \beta$ and $p \models \gamma$. By the induction hypothesis this is the case iff $q \models \beta$ and $q \models \gamma$, hence iff $q \models \alpha$.

(iv) $\alpha = \beta \vee \gamma$. Once again the required result follows from the induction hypothesis.

(v) $\alpha = \langle a \rangle(\beta, \gamma)$. By definition of $\models$, $p \models \langle a \rangle(\beta, \gamma)$ implies that there exists p', p'' such that $p \xrightarrow{a} \langle p', p'' \rangle$, $p' \models \beta$ and $p'' \models \gamma$. As $p \sim_d q$ we conclude that there exists q', q'' such that $q \xrightarrow{a} \langle q', q'' \rangle, q' \sim_d p'$ and $q'' \sim_d p''$. By structural induction we have $q' \models \beta$ and $q'' \models \gamma$, hence $q \models \langle a \rangle(\beta, \gamma)$. By similar arguments it can be shown that $q \models \langle a \rangle(\beta, \gamma)$ implies $p \models \langle a \rangle(\beta, \gamma)$.

(vi) $\alpha = [a](\beta, \gamma)$. By definition of $\models$, $p \models [a](\beta, \gamma)$ implies for all p and p'', if $p \xrightarrow{a} \langle p', p'' \rangle$ then $p' \models \beta$ or $p'' \models \gamma$. Now suppose $q \xrightarrow{a} \langle q', q'' \rangle$. As $p \sim_d q$ we conclude that there exists p', p'' such that $p \xrightarrow{a} \langle p', p'' \rangle$, and moreover $p' \sim_d q'$ and $p'' \sim_d q''$, thus by the induction hypothesis we conclude that $q' \models \beta$ or $q'' \models \gamma$. Since this argument holds for any q', q'' such that $q \xrightarrow{a} \langle q', q'' \rangle$ we have $q \models [a](\beta, \gamma)$. By similar arguments it can be shown that $q \models [a](\beta, \gamma)$ implies $p \models [a](\beta, \gamma)$.

($\Leftarrow$): Let $\sim_d^* = \{(p, q) |\ \forall \alpha \in \mathcal{L} : p \models \alpha \Leftrightarrow q \models \alpha\}$. We show that $\sim_d^*$ is a distributed bisimulation. Suppose $(p, q) \in \sim_d^*$ such that $p \xrightarrow{a} \langle p', p'' \rangle$ but for all q', q'' such that $q \xrightarrow{a} \langle q', q'' \rangle$ we have $p' \not\sim_d^* q'$ or $p'' \not\sim_d^* q''$. As q is assumed to be finitely branching let $\langle q'_1, q''_1 \rangle, \ldots, \langle q'_n, q''_n \rangle$ be the possible a derivatives of q. By definition of $\sim_d^*$ and lemma 3.5 we have for all $i = 1, \ldots, n$ that there exists α'_i such that $p' \models \alpha'_i$ and $q'_i \not\models \alpha'_i$ or that there exists α''_i such that $p'' \models \alpha''_i$ and $q''_i \not\models \alpha''_i$. Let $\alpha_{h(1)}, \ldots, \alpha_{h(j_1)}$ be those formulae coming from the property $p' \not\sim_d^* q'_i$, i.e. for all $k \in \{1, \ldots, j_1\}$ we have $p' \models \alpha_{h(k)}$, and if $p' \not\sim_d^* q'_i$ then there exists $k \in \{1, \ldots, j_1\}$ such that $q'_i \not\models \alpha_{h(k)}$. Let $\alpha_{m(1)}, \ldots, \alpha_{m(j_2)}$ be those formulae coming from the property $p'' \not\sim_d^* q''_i$, i.e. for all $k \in \{1, \ldots, j_2\}$ we have $p'' \models \alpha_{m(k)}$, and if $p'' \not\sim_d^* q''_i$ then there exists $k \in \{1, \ldots, j_2\}$ such that $q''_i \not\models \alpha_{m(k)}$. Let $\Gamma = \alpha_{h(1)} \wedge \ldots \wedge \alpha_{h(j_1)}$ and $\Phi = \alpha_{m(1)} \wedge \ldots \wedge \alpha_{m(j_2)}$. Then we have $p \models \langle a \rangle(\Gamma, \Phi)$ and $q \not\models \langle a \rangle(\Gamma, \Phi)$ contradicting the assumption $(p, q) \in \sim_d^*$. ■

According to the second half of the proof, if infinite conjunction was available the theorem would be true for infinite branching processes as well.

We end this section by defining some derived modalities which could be of use in specifying processes.

Definition 3.7

(i) $\langle a \rangle_l(\alpha) \stackrel{\text{def}}{=} \langle a \rangle(\alpha, tt)$

(ii) $\langle a \rangle_c(\alpha) \stackrel{\text{def}}{=} \langle a \rangle(tt, \alpha)$

(iii) $[a]_l(\alpha) \stackrel{\text{def}}{=} [a](\alpha, ff)$

(iv) $[a]_c(\alpha) \stackrel{\text{def}}{=} [a](ff, \alpha)$

■

Proposition 3.8 Suppose $p \in \mathcal{P}$ and $\alpha \in \mathcal{L}$. Then

(i) $p \models \langle a \rangle_l(\alpha)$ iff $\exists p', p'' : p \xrightarrow{a} \langle p', p \rangle$ & $p' \models \alpha$,

(ii) $p \models \langle a \rangle_c(\alpha)$ iff $\exists p', p'' : p \xrightarrow{a} \langle p', p \rangle$ & $p'' \models \alpha$,

(iii) $p \models [a]_l(\alpha)$ iff $\forall p', p'' : p \xrightarrow{a} \langle p', p'' \rangle$ implies $p' \models \alpha$, and

(iv) $p \models [a]_c(\alpha)$ iff $\forall p', p'' : p \xrightarrow{a} \langle p', p'' \rangle$ implies $p'' \models \alpha$.

Proof Is easily seen to be a consequence of definition 3.2 and 3.7. ■

The two modalities $\langle a \rangle_l$ and $\langle a \rangle_c$ are defined via the possibility modality $\langle a \rangle$ but they are monadic and referring to the local and concurrent residual respectively. The relationship $p \models \langle a \rangle_l(\alpha)$ indicates that p is capable of performing an a action whereupon the local residual will satisfy α. Similarly, $p \models \langle a \rangle_c(\alpha)$ indicates that p is capable of doing an a action whereupon the concurrent residual will satisfy α.

The last two derived modalities $[a]_l$ and $[a]_c$ are defined through the necessity modality but are monadic, referring to the local and concurrent residual respectively. The relationship $p \models [a]_l(\alpha)$ expresses intuitively that whenever p performs an a action the local residual will satisfy α. Finally, $p \models [a]_c(\alpha)$ expresses that whenever p performs an a action the concurrent residual satisfies α.

4 The Proof System

We aim to present a sound and complete proof system for our process language $\mathrm{CCS_S}$ defined in section 2. That is, we seek a proof system to decide whether, for arbitrary $\mathrm{CCS_S}$ processes p and formulae α, the relationship $p \models \alpha$ holds. The proof system will be structural both in the structure of formulae and in the structure of processes.

In constructing the proof system we are guided by the proof of completeness. If a specific rule is required in order to obtain the proof of completeness the rule is checked for soundness and then introduced as an inference rule.

It turns out that it is easy to develop axioms and inference rules for all the combinators except the parallel composition (|). For instance, in search of

inference rules for the case $p|q \vdash \langle a \rangle(\alpha, \beta)$ it would be necessary to examine the structure of both p and q. If p equals $a.p'$ then we could introduce an inference rule like

$$\frac{p' \vdash \alpha,\ q \vdash \beta}{a.p'|q \vdash \langle a \rangle(\alpha, \beta)}$$

But if p equals $p'|p''$ then we have to dig deeper into the structure of $p|q$ by examining p' and p'', thus leading to the same analysis once more. Taking this route it is not difficult to see that the inference rules for the parallel combinator will be very awkward and unpleasant to read.

Instead we introduce a new dyadic combinator of our language providing very intuitive inference rules for the parallel combinator. The price we pay is a set of inference rules for the new combinator but these inference rules are simpler and intuitively more attractive.

The combinator in question is *left merge*, denoted $\lfloor$, and its operational semantics is as follows:

$$\frac{p \xrightarrow{a} \langle p', p'' \rangle}{p \lfloor q \xrightarrow{a} \langle p', p''|q \rangle}$$

Note that $\lfloor$ has some similarity with the parallel combinator ($|$). Only with respect to the first step there is a difference between $|$ and $\lfloor$; in $p \lfloor q$ the process p has precedence over q and must perform the first action. This is not the case for $p|q$. We extend our set of processes to include the new combinator, left merge ($\lfloor$), and denote it by $\mathrm{CCS}_{\mathrm{S}}$.

As promised, by introducing the left merge combinator the inference rules for the parallel composition ($|$) becomes simple. In case the formula under consideration is $\langle a \rangle(\alpha, \beta)$ the only inference rules required are:

$$\frac{p \lfloor q \vdash \langle a \rangle(\alpha, \beta)}{p|q \vdash \langle a \rangle(\alpha, \beta)} \qquad \frac{q \lfloor p \vdash \langle a \rangle(\alpha, \beta)}{p|q \vdash \langle a \rangle(\alpha, \beta)}$$

The inference rules will be sound because for $p|q$ to satisfy the formula $\langle a \rangle(\alpha, \beta)$ either p or q has to perform the action a required.

We now present the proof system in full and then go into details about some of the axioms and inference rules afterwards.

Axioms

A1: $p \vdash \not\!\!t$

A2: $nil \vdash [a](\alpha, \beta)$

A3: $nil \lfloor p \vdash [a](\alpha, \beta)$

A4: $b.p \vdash [a](\alpha, \beta)$ whenever $b \neq a$

A5: $b.p \lfloor q \vdash [a](\alpha, \beta)$ whenever $b \neq a$

Inference Rules

R1:

$$\frac{p \vdash \alpha, p \vdash \beta}{p \vdash \alpha \wedge \beta}$$

R2:

$$\frac{p \vdash \alpha}{p \vdash \alpha \vee \beta} \qquad \frac{p \vdash \alpha}{p \vdash \beta \vee \alpha}$$

R3:

$$\frac{p \vdash a, nil \vdash \beta}{a.p \vdash \langle a \rangle(\alpha, \beta)}$$

R4:

$$\frac{p \vdash \langle a \rangle(\alpha, \beta)}{p + q \vdash \langle a \rangle(\alpha, \beta)} \qquad \frac{q \vdash \langle a \rangle(\alpha, \beta)}{p + q \vdash \langle a \rangle(\alpha, \beta)}$$

R5:

$$\frac{p \lfloor q \vdash \langle a \rangle(\alpha, \beta)}{p | q \vdash \langle a \rangle(\alpha, \beta)} \qquad \frac{q \lfloor p \vdash \langle a \rangle(\alpha, \beta)}{p | q \vdash \langle a \rangle(\alpha, \beta)}$$

R6:

$$\frac{p \vdash \alpha}{a.p \vdash [a](\alpha, \beta)} \qquad \frac{nil \vdash \beta}{a.p \vdash [a](\alpha, \beta)}$$

R7:

$$\frac{p \vdash [a](\alpha,\beta), q \vdash [a](\alpha,\beta)}{p+q \vdash [a](\alpha,\beta)}$$

R8:

$$\frac{p \lfloor q \vdash [a](\alpha,\beta), q \lfloor p \vdash [a](\alpha,\beta)}{p|q \vdash [a](\alpha,\beta)}$$

R9:

$$\frac{p \vdash \alpha, nil|q \vdash \beta}{a.p \lfloor q \vdash \langle a \rangle(\alpha,\beta)}$$

R10:

$$\frac{p \lfloor r \vdash \langle a \rangle(\alpha,\beta)}{(p+q) \lfloor r \vdash \langle a \rangle(\alpha,\beta)} \qquad \frac{q \lfloor r \vdash \langle a \rangle(\alpha,\beta)}{(p+q) \lfloor r \vdash \langle a \rangle(\alpha,\beta)}$$

R11:

$$\frac{p \lfloor (q|r) \vdash \langle a \rangle(\alpha,\beta)}{(p|q) \lfloor r \vdash \langle a \rangle(\alpha,\beta)} \qquad \frac{q \lfloor (p|r) \vdash \langle a \rangle(\alpha,\beta)}{(p|q) \lfloor r \vdash \langle a \rangle(\alpha,\beta)}$$

R12:

$$\frac{p \lfloor (q|r) \vdash \langle a \rangle(\alpha,\beta)}{(p \lfloor q) \lfloor r \vdash \langle a \rangle(\alpha,\beta)}$$

R13:

$$\frac{p \vdash \alpha}{a.p \lfloor q \vdash [a](\alpha,\beta)} \qquad \frac{nil|q \vdash \beta}{a.q \lfloor q \vdash [a](\alpha,\beta)}$$

R14:

$$\frac{p \lfloor r \vdash [a](\alpha,\beta),\ q \lfloor r \vdash [a](\alpha,\beta)}{(p+q) \lfloor r \vdash [a](\alpha,\beta)}$$

R15:

$$\frac{p \lfloor (q|r) \vdash [a](\alpha,\beta),\ q \lfloor (p|r) \vdash [a](\alpha,\beta)}{(p|q) \lfloor r \vdash [a](\alpha,\beta)}$$

R16:

$$\frac{p \lfloor (q|r) \vdash [a](\alpha,\beta)}{(p \lfloor q) \lfloor r \vdash [a](\alpha,\beta)}$$

Axioms **A1**, **A2** and **A4** are standard and and require no explanation. Axioms **A3** and **A5** are new and concern the left merge combinator. They are similar to **A2** and **A4** respectively and sound because the left component of the left merge operator has precedence over the right component in the first step.

Inference rules **R1** and **R2** are standard and require no explanation. Rules **R3** to **R5** deal with the formula $\langle a \rangle(\alpha,\beta)$. There is one rule for each of the combinators prefix, sum and parallel composition. The cases of sum and parallel composition require no explanation. For the case of prefix we have the extra requirement of $nil \vdash \beta$ because in observing the action a of $a.p$ the concurrent residual becomes the *nil* process.

Rules **R6** to **R8** deal with the formula $[a](\alpha,\beta)$. Again there is one rule for each of the combinators prefix, sum and parallel composition. Note the two cases for the prefix combinator; either we require that the local residual, i.e. p, satisfies α or that the concurrent residual, i.e. the *nil* process, satisfies β; compare with definition 3.2 **(vi)**. Observe that **R5** and **R8** are the only inference rules required for the parallel combinator.

The rest of the inference rules deal with the left merge combinator; rule **R9** to **R12** in case the formula is $\langle a \rangle(\alpha,\beta)$ and rule **R13** to **R16** in case the formula is $[a](\alpha,\beta)$. Observe that for each of the two formulae there is a rule for each of the possible structures the left process p in $p \lfloor q$ can have. We will not go into details about the rules; they more or less follow the same pattern as the rules in case we forget about tlie right process q in $p \lfloor q$. For instance, rule **R11** is similar to rule **R5**; we just have to remember that tlie right process of the left merge operator will become part of the concurrent residual.

Theorem 4.1 The proof system is sound and complet,e, i.e.

$$\forall p \in \mathrm{CCS}_S,\ \forall \alpha \in \mathcal{L} : p \models \alpha \Leftrightarrow p \vdash \alpha.$$

Proof It is rather easy to verify all the axioms and inference rules hence we will not be concerned with the soundness of the proof system. The completeness proof is long and tedious involving well-founded induction on pairs of formulae and processes equipped with a suitable well-founded order. We have postponed the proof until appendix A. ■

Example 4.2 We have already seen that $a|b \models \langle a\rangle(tt, \langle b\rangle(tt, tt))$. According to the completeness of our proof system we must have $a|b \vdash \langle a\rangle(tt, \langle b\rangle(tt, tt))$. This is the content of the following figure:

Figure 3

$$\dfrac{\dfrac{nil \vdash tt \qquad \dfrac{\dfrac{nil \vdash tt \quad nil|nil \vdash tt}{b\lfloor nil \vdash \langle b\rangle(tt,tt)}\,\mathrm{R9}}{nil|b \vdash \langle b\rangle(tt,tt)}\,\mathrm{R5}}{a\lfloor b \vdash \langle a\rangle(tt, \langle b\rangle(tt,tt))}\,\mathrm{R9}}{a|b \vdash \langle a\rangle(tt, \langle b\rangle(tt,tt))}\,\mathrm{R5}$$

5 Conclusion

In this paper we have succeeded in giving a logical characterisation of the distributed bisimulation equivalence. The logic has some similarity with Hennessy-Milner logic; it contains two modalities, viz. the necessity and the possibility modality. But whereas these modalities are monadic in Hennessy-Milner logic they are dyadic in our logic.

Furthermore, we have provided a structural proof system for a small subset of CCS containing prefix, sum and parallel composition as combinators. Via a new operator introduced, called left merge, we have obtained very simple inference rules for the parallel composition.

There is a number of interesting extensions to the presented work. It would be preferable to allow the processes to communicate either via visible or invisible actions. A solution to this extension would at least require modified inference rules for the parallel combinator. If τ stands for the communication action

then in search of inference rules for $p|q \vdash \langle a \rangle(\alpha, \beta)$ it is no longer enough to consider τ coming either from p or q; it could be the case that p and q communicated thus creating the τ action. Perhaps the problem can be solved by introducing yet another combinator which forces communication to occur between its operand.[5] In the future we will work on these problems.

It would also be interesting to extend the proof system to cover full CCS. At least with respect to the combinator *restriction* we see non-trivial problems. Hennessy and Castellani have not included restriction in [HC88] but mention that it would cause difficulties because their framework does not fit the restriction combinator. On solving the problems a good starting point would perhaps be [Kie89] where distributed bisimulation on a language including restriction has been considered.

Acknowledgement

I would like to thank Uffe Engberg, Mogens Nielsen and Henrik Reif Andersen at Aarhus University for careful reading of an earlier draft of this paper and for helpful discussions during the work.

[5] Such a combinator has already been considered in [HC88] in order to obtain an algebraic characterisation for $\sim_d$ on a language including communication.

References

[Kie89] A. Kiehn. *Distributed Bisimulations for Finite* CCS, University of Sussex, Dept. of Computer Science, Report no. 7/89, 1989.

[Cas88] I. Castellani. *Bisimulations for Concurrency*, Ph.D. Thesis, Edinburgh University, CST-51-88, April 1988.

[HC88] M. Hennessy and I. Castellani. *Distributed Bisimulations*, INRIA Reports de Recherche, No. 875, 1988.

[HM85] M. Hennessy and R. Milner. *Algebraic Law for Non-determinism and Concurrency*, Journal of ACM, Vol 32, No. 1, pp 137–161, 1985.

[Mil89] R. Milner. *Communication and Concurrency*, Prentice Hall, 1989.

[Sti85] C. Stirling. *A Complete Modal Proof System for a Subset of CCS*, Tapsoft Proceedings, Vol 1, LNCS 185, pp253–267, 1985.

A Appendix

The purpose of this appendix is to prove that the proof system is complete. The proof of completeness involves well-founded induction on pairs of formulae and processes. Before we define the well-founded order to be used in the proof we define an order on processes.

Let p, q and r be $\mathrm{CCS_S}$ processes. By $\sqsubset$ we denote the least relation on $\mathrm{CCS_S}$ satisfying the relationships shown in the following table:

$p \sqsubset p+q$	$q \sqsubset p+q$	$p \lfloor q \sqsubset p \mid q$
$q \lfloor p \sqsubset p \mid q$	$p \lfloor r \sqsubset (p+q) \lfloor r$	$q \lfloor r \sqsubset (p+q) \lfloor r$
$p \lfloor (q \mid r) \sqsubset (p \mid q) \lfloor r$	$q \lfloor (p \mid r) \sqsubset (p \mid q) \lfloor r$	$p \lfloor (q \mid r) \sqsubset (p \lfloor q) \lfloor r$

It is easily checked that the relation $\sqsubset$ is a well founded order. For instance, the following metric M:

(i) $M[nil) = 1$,

(ii) $M(a.p) = 1 + M(p)$,

(iii) $M(p+q) = M(p) + M(q)$,

(iv) $M(p|q) = M(p) + M(q) + 1$, and

(v) $M(p \lfloor q) = M(p) + 1$

will satisfy $M(p) < M(q)$ whenever $p \sqsubset q$ where $<$ is the *less than* ordering on natural numbers.

We now define the well-founded order used in the completeness proof.

Definition A.1 Suppose $\alpha, \beta \in \mathcal{L}$ and $p, q \in \mathrm{CCS_S}$. Let $\prec$ be a relation on $\mathcal{L} \times \mathrm{CCS_S}$ defined as follows:

$$(\alpha, p) \prec (\beta, q) \text{ iff } \alpha \propto \beta \text{ or } (\alpha \equiv \beta \text{ and } p \sqsubset q),$$

where $\propto$ denotes the structural order on formulae and $\equiv$ denotes syntactical equality on formulae. ■

As both $\sqsubset$ and $\propto$ are well-founded it follows that $\prec$ is a well-founded order.

We now present the completeness proof.

Theorem A.2 Let $p \in \mathrm{CCS_S}$ and $\alpha \in \mathcal{L}$. Then $p \models \alpha$ implies $p \vdash \alpha$.

Proof Suppose $p \models \alpha$. By well-founded induction on $(\mathcal{L} \times \mathrm{CCS_S}, \prec)$ we want to prove $p \vdash \alpha$. That is, based on the induction hypothesis

$$\forall (\beta, q) \in \mathcal{L} \times \mathrm{CCS_S} : \text{ if } (\beta, q) \prec (\alpha, p) \text{ then } q \models \beta \text{ implies } q \vdash \beta$$

we prove that $p \models \alpha$ implies $p \vdash \alpha$. We proceed by analysing the cases for α.

(i) $\alpha = tt$. Then we have $p \vdash \alpha$ for all $p \in \mathrm{CCS_S}$ by axiom **A1**.

(ii) $\alpha = ff$. Then we cannot have $p \models \alpha$ by definition of $\models$.

(iii) $\alpha = \beta \wedge \gamma$. By definition of $\models$, $p \models \beta \wedge \gamma$ implies $p \models \beta$ and $p \models \gamma$, hence $p \vdash \beta$ and $p \vdash \gamma$ according to the induction hypothesis. By inference rule **R1** we conclude that $p \vdash \beta \wedge \gamma$.

(iv) $\alpha = \beta \vee \gamma$. Once again the required result follows from the induction hypothesis but this time by using inference rule **R2** instead of **R1**.

(v) $\alpha = \langle a \rangle (\beta, \gamma)$. We proceed by analysing the cases for p.

- **(a)** $p = nil$. Then we cannot have $p \models \langle a \rangle (\beta, \gamma)$.
- **(b)** $p = b.p_1$. As $p \models \langle a \rangle (\beta, \gamma)$ we must have $a = b$, i.e. $a.p_1 \models \langle a \rangle (\beta, \gamma)$ which implies $p_1 \models \beta$ and $nil \models \gamma$. By the induction hypothesis we conclude $p_1 \vdash \beta$ and $nil \vdash \gamma$, hence by rule **R.3** we have $p \vdash \langle a \rangle (\beta, \gamma)$.
- **(c)** $p = q + r$. By definition of the transition rules for the sum operator $(+)$, $p \models \langle a \rangle (\beta, \gamma)$ implies $q \models \langle a \rangle (\beta, \gamma)$ or $r \models \langle a \rangle (\beta, \gamma)$. Thus by the induction hypothesis we conclude $q \vdash \langle a \rangle (\beta, \gamma)$ or $r \vdash \langle a \rangle (\beta, \gamma)$ which by rule **R4** implies $p \vdash \langle a \rangle (\beta, \gamma)$.
- **(d)** $p = q | s$. According to the transition rules for $|$ and $\lfloor$ we have that $q | s \models \langle a \rangle (\beta, \gamma)$ implies $q \lfloor s \models \langle a \rangle (\beta, \gamma)$ or $s \lfloor q \models \langle a \rangle (\beta, \gamma)$. By the induction hypothesis we conclude $q \lfloor s \vdash \langle a \rangle (\beta, \gamma)$ or $s \lfloor q \vdash \langle a \rangle (\beta, \gamma)$, hence by rule **R5** we have $q | s \vdash \langle a \rangle (\beta, \gamma)$.
- **(e)** $p = q \lfloor s$. We proceed by analysing the cases for q.
  - **i.** $q = nil$. Then we cannot have $q \lfloor a \models \langle a \rangle (\beta, \gamma)$.

ii. $q = b.q_1$. If $q \lfloor s \models \langle a \rangle(\beta, \gamma)$ then $a = b$, $q_1 \models \beta$ and $nil|s \models \gamma$. By the induction hypothesis conclude $q_1 \vdash \beta$ and $nil|s \vdash \gamma$, hence according to rule **R9** we conclude that $a.q \lfloor s \vdash \langle a \rangle(\beta, \gamma)$.

iii. $q = q_1 + q_2$. By definition of the transition rules for $\lfloor$ and sum $(+)$, $(q_1 + q_2) \lfloor s \models \langle a \rangle(\beta, \gamma)$ implies $q_1 \lfloor s \models \langle a \rangle(\beta, \gamma)$ or $q_2 \lfloor s \models \langle a \rangle(\beta, \gamma)$. By the induction hypothesis we conclude that $q_1 \lfloor s \vdash \langle a \rangle(\beta, \gamma)$ or $q_2 \lfloor s \vdash \langle a \rangle(\beta, \gamma)$, hence by rule **R10** we have $(q_1 + q_2) \lfloor s \vdash \langle a \rangle(\beta, \gamma)$.

iv. $q = q_1|q_2$. According to the transition rules for $|$ and $\lfloor$, $(q_1, q_2) \lfloor s \models \langle a \rangle(\beta, \gamma)$ implies $q_1 \lfloor (q_2|s) \models \langle a \rangle(\beta, \gamma)$ or $q_2 \lfloor (q_1|s) \models \langle a \rangle(\beta, \gamma)$. By the induction hypothesis we have $q_1 \lfloor (q_2|s) \vdash \langle a \rangle(\beta, \gamma)$ or $q_2 \lfloor (q_1|s) \vdash \langle a \rangle(\beta, \gamma)$, hence by rule **R11** we conclude $(q_1|q_2) \lfloor s \vdash \langle a \rangle(\beta, \gamma)$

v. $q = q_1 \lfloor q_2$. By definition of the transition rule for $\lfloor$, $(q_1 \lfloor q_2) \lfloor s \models \langle a \rangle(\beta, \gamma)$ implies $q_1 \lfloor (q_2|s) \vdash \langle a \rangle(\beta, \gamma)$. By the induction hypothesis it follows that $q_1 \lfloor (q_2|s) \vdash \langle a \rangle(\beta, \gamma)$, hence by rule **R12** $(q_1 \lfloor q_2) \lfloor s \vdash \langle a \rangle(\beta, \gamma)$.

(vi) We finally have to consider the case $\alpha = [a](\beta, \gamma)$. Again we proceed by analysing the cases for p.

(a) $p = nil$. By axiom **A2** we have $nil \vdash [a](\beta, \gamma)$.

(b) $p = b.p_1$. If $b \neq a$ then we have $b.p_1 \vdash [a](\beta, \gamma)$ axiom **A4**. If $b = a$ then by definition of $\models$, $a.p_1 \models [a](\beta, \gamma)$ implies $p_1 \models \beta$ or $nil \models \gamma$. By the induction hypothesis we conclude $p_1 \vdash \beta$ or $nil \vdash \gamma$, hence by rule **R6** we have $a.p_1 \vdash [a](\beta, \gamma)$.

(c) $p = q + s$. By definition of the transition rules for the sum operator $(+)$, $q + s \models [a](\beta, \gamma)$ implies $q \models [a](\beta, \gamma)$ and $s \models [a](\beta, \gamma)$. By the induction hypothesis we conclude $q \vdash [a](\beta, \gamma)$ and $s \vdash [a](\beta, \gamma)$, hence by rule **R7** we have $q + s \vdash [a](\beta, \gamma)$.

(d) $p = q|s$. By definition of the transition rule for $|$, $p|s \models [a](\beta, \gamma)$ implies $p \lfloor s \models [a](\beta, \gamma)$ and $s \lfloor p \models [a](\beta, \gamma)$. By the induction hypothesis we conclude $p \lfloor s \vdash [a](\beta, \gamma)$ and $s \lfloor p \vdash [a](\beta, \gamma)$, hence by rule **R8** we have $p|s \models [a](\beta, \gamma)$.

(e) We finally have to consider the case $p = q \lfloor s$. We proceed by analysing the cases for q.

i. $q = nil$. By axiom **A3** we have $nil \lfloor s \vdash [a](\beta, \gamma)$.

ii. $q = b.q_1$. If $b \neq a$ we have $b.q_1 \lfloor s \vdash [a](\beta, \gamma)$ by axiom **A5**. If $b = a$ then $a.q_1 \lfloor s \models [a](\beta, \gamma)$ implies $q_1 \models \beta$ or $nil|s \models \gamma$. By the induction hypothesis we have $q_1 \vdash \beta$ or $nil|s \vdash \gamma$, hence by rule **R13** we conclude $a.q_1 \lfloor s \vdash [a](\beta, \gamma)$.

iii. $q = q_1 + q_2$. By the transition rules for sum $(+)$ and $\lfloor$, $(q_1 + q_2) \lfloor s \models [a](\beta, \gamma)$ implies $q_1 \lfloor s \models [a](\beta, \gamma)$ and $q_2 \lfloor s \models [a](\beta, \gamma)$. By the induction hypothesis we conclude $q_1 \lfloor s \vdash [a](\beta, \gamma)$ and $q_2 \lfloor s \vdash [a](\beta, \gamma)$, hence by rule **R14** we have $(q_1 + q_2) \lfloor s \vdash [a](\beta, \gamma)$.

iv $q = q_1|q_2$. By definition of the transition rules for $|$ and $\lfloor$, we have that $(q_1|q_2) \lfloor s \models [a](\beta, \gamma)$ implies $q_1 \lfloor (q_2|s) \models [a](\beta, \gamma)$ and $q_2 \lfloor (q_1|s) \models [a](\beta, \gamma)$. By the induction hypothesis we conclude $q_1 \lfloor (q_2|s) \vdash [a](\beta, \gamma)$ and $q_2 \lfloor (q_1|s) \vdash [a](\beta, \gamma)$, hence by rule **R15** we have $(q_1|q_2) \lfloor s \vdash [a](\beta, \gamma)$.

v. $q = q_1|q_2$. According to the inference rules for $\lfloor$, $(q_1 \lfloor q_2) \lfloor s \models [a](\beta, \gamma)$ implies $q_1 \lfloor (q_2|s) \models [a](\beta, \gamma)$. By the induction hypothesis we conclude $q_1 \lfloor (q_2|s) \vdash [a](\beta, \gamma)$ hence by rule **R16** we have $(q_1 \lfloor q_2) \lfloor s \vdash [a](\beta, \gamma)$. ■

B Appendix

In this appendix we investigate the logic obtained by exchange 'and' for 'or' and 'or' for 'and' in the interpretations of the two modality operators diamond ($\langle\rangle$) and box ($[]$) respectively. For easiness of discussion let a new logic $\mathcal{L}'$ be based on $\mathcal{L}$ concerning the syntax. For a process p to satisfy $\langle a \rangle(\alpha, \beta)$ in the new logic we only require that there exists an a derivative of p such that the local residual satisfies α *or* such that the concurrent residual satisfies β. On the other hand, for p to satisfy $[a](\alpha, \beta)$ in the new logic we require for all a derivatives of p that the local residual satisfies α *and* that the concurrent residual satisfies β. To be formal we now define the semantics of the logic $\mathcal{L}'$.

Definition B.1 Let $p \in \mathcal{P}$ and $\alpha \in \mathcal{L}'$. Finally, let $p \models \alpha$ denote that p satisfies α. The relation $\models$ is defined by structural induction on the structure of α.

(i) $\forall p \in \mathcal{P} : p \models tt$,

(ii) $\forall p \in \mathcal{P} : p \not\models tt$,

(iii) $p \models \alpha \wedge \beta$ iff $p \models \alpha$ & $p \models \beta$,

(iv) $p \models \alpha \vee \beta$ iff $p \models \alpha$ or $p \models \beta$,

(v) $p \models \langle a \rangle(\alpha, \beta)$ iff $\exists p', p'' : p \xrightarrow{a} \langle p', p'' \rangle$ & $(p' \models \alpha$ or $p'' \models \beta)$, and

(vi) $p \models [a](\alpha, \beta)$ iff $\forall p', p'' : p \xrightarrow{a} \langle p', p'' \rangle$ implies $p' \models \alpha$ and $p'' \models \beta$.

■

It turns out that by this logic we get an equivalence on processes which is weaker than the distributed bisimulation equivalence, i.e. by the logic $\mathcal{L}'$ we obtain an equivalence on $\mathcal{P}$ which on the one hand contains $\sim_d$; but also equate non distributed bisimulation equivalent processes. In order to justify this postulate we first define an equivalence, denoted $\sim_{wd}$, on processes and subsequently show that $\sim_{wd}$ is characterised by the logic $\mathcal{L}'$.

Definition B.2 $R \in \mathcal{P} \times \mathcal{P}$ is a *weakly distributed bisimulation* provided for all $(p, q) \in \mathcal{R}$ the following is satisfied:

(i) $p \xrightarrow{a} \langle p', p'' \rangle$ implies $q \xrightarrow{a} \langle q', q'' \rangle$ such that $(p', q') \in \mathcal{R}$,

(ii) $p \xrightarrow{a} \langle p', p'' \rangle$ implies $q \xrightarrow{a} \langle q', q'' \rangle$ such that $(p'', q'') \in \mathcal{R}$,

(iii) $q \xrightarrow{a} \langle q', q'' \rangle$ implies $p \xrightarrow{a} \langle p', p'' \rangle$ such that $(p', q') \in \mathcal{R}$, and

(iv) $q \xrightarrow{a} \langle q', q'' \rangle$ implies $p \xrightarrow{a} \langle p', p'' \rangle$ such that $(p'', q'') \in \mathcal{R}$.

■

Definition B.3 Let $p, q \in \mathcal{P}$. Then $p \sim_{wd} q$ iff there exists a weakly distributed bisimulation R such that $(p, q) \in R$. If $p \sim_{wd} q$ we call p and q weakly distributed bisimulation equivalent. ■

Theorem B.4 Assume $p, q \in \mathcal{P}$ such that p and q are finitely branching processes. Then

$$p \sim_{wd} q \Leftrightarrow (\forall \alpha \in \mathcal{L}' : p \models \alpha \Leftrightarrow q \models \alpha)$$

Proof $(\Rightarrow)$: Assume $p \sim_{wd} q$. By induction on the structure of α we show $p \models \alpha \Leftrightarrow q \models \alpha$.

(i) $a = tt$. By definition of $\models$ we have $p \models tt$ iff $q \models tt$.

(ii) $a = ff$. Once again the required result follows from the definition of $\models$.

(iii) $\alpha = \beta \wedge \gamma$. We have $p \models \alpha$ iff $p \models \beta$ and $p \models \gamma$. By the induction hypothesis this is the case iff $q \models \beta$ and $q \models \gamma$, hence iff $q \models \alpha$.

(iv) $\alpha = \beta \vee \gamma$. Once again the required result follows from the induction hypothesis.

(v) $\alpha = \langle a \rangle(\beta, \gamma)$. By definition of $\models$, $p \models \langle a \rangle(\beta, \gamma)$ implies that there exists p', p'' such that $p \xrightarrow{a} \langle p', p'' \rangle$ with $p' \models \beta$ or $p'' \models \gamma$. Suppose first that $p' \models \beta$ is the case. Now, as $p \sim_{wd} q$ we have that there exists q', q'' with $q \xrightarrow{a} \langle q', q'' \rangle$ such that $q' \sim_{wd} p'$. By the induction hypothesis we conclude $q' \models \beta$ hence $q \models \langle a \rangle(\beta, \gamma)$. Secondly, suppose that $p'' \models \gamma$ is the case. Again, as $p \sim_{wd} q$, we have that there exists q', q'' with $q \xrightarrow{a} \langle q', q'' \rangle$ such that $q'' \sim_{wd} p''$. By the induction hypothesis we conclude $q'' \models \gamma$ hence $q \models \langle a \rangle(\beta, \gamma)$. By similar arguments it can be shown that $q \models \langle a \rangle(\beta, \gamma)$ implies $p \models \langle a \rangle(\beta, \gamma)$.

(vi) $\alpha = [a](\beta, \gamma)$. By definition of $\models$, $p \models \langle a \rangle(\beta, \gamma)$ implies for all p' and p'', if $p \xrightarrow{a} \langle p', p'' \rangle$ then $p' \models \beta$ and $p'' \models \gamma$. Now suppose $q \xrightarrow{a} \langle q', q'' \rangle$. As $p \sim_{wd} q$ we conclude that there exists p'_1, p''_1 such that $p \xrightarrow{a} \langle p'_1, p''_1 \rangle$ and $p'_1 \sim_{wd} q'$. Moreover, there exists p'_2, p''_2 such that $p \xrightarrow{a} \langle p'_2, p''_2 \rangle$ and $p'_2 \sim_{wd} q'$. By the induction hypothesis we get $q' \models \beta$ and $q'' \models \gamma$. Since this argument holds for any q', q'' such that $q \xrightarrow{a} \langle q', q'' \rangle$ we have $q \models [a](\beta, \gamma)$. By similar arguments it can be shown that $q \models [a](\beta, \gamma)$ implies $p \models [a](\beta, \gamma)$.

($\Leftarrow$) : Let $\sim^*_{wd} = \{(p, q) \mid \forall \alpha \in \mathcal{L}' : p \models \alpha \Leftrightarrow q \models \alpha\}$. We show that $\sim^*_{wd}$ is a weakly distributed bisimulation. The proof proceeds by contradiction, i.e. suppose there exists $(p, q) \in \sim^*_{wd}$ which does not satisfy definition B.2. Assume without loss of generality that $p \xrightarrow{a} \langle p', p'' \rangle$ but for all q', q'' such that $q \xrightarrow{a} \langle q', q'' \rangle$ we have $p' \not\sim^*_{wd} q'$ or for all q', q'' such that $q \xrightarrow{a} \langle q', q'' \rangle$ we have $p'' \not\sim^*_{wd} q''$. As q is assumed to be finitely branching let $\langle q'_1, q''_1 \rangle, \ldots, \langle q'_n, q''_n \rangle$ be the possible a derivatives of q. By definition of $\sim^*_{wd}$ we have for all $i = 1, \ldots, n$ that there exists α'_i such that $p' \models \alpha'_i$ and $q'_i \not\models \alpha'_i$ or we have for all $i = 1, \ldots, n$ that there exists α''_i such that $p'' \models \alpha''_i$ and $q''_i \not\models \alpha''_i$. Suppose first that we have the existence of the formulae α'_i. Then let $\Gamma = \alpha'_1 \wedge \ldots \wedge \alpha'_n$

and we conclude $p \models \langle a \rangle(\Gamma, ff)$ but $q \not\models \langle a \rangle(\Gamma, ff)$. Now suppose that we have the existence of the formulae α_i''. Then let $\Phi = \alpha_1'' \wedge \ldots \wedge \alpha_n''$ and we conclude $p \models \langle a \rangle(ff, \Phi)$ but $q \not\models \langle a \rangle(ff, \Phi)$. In both cases we arrive at a contradiction, hence $\sim_{wd}^*$ must be a weakly distributed bisimulation. ■

We end this appendix with a comparison of $\sim_{wd}$ with $\sim_d$ and $\sim$ where $\sim$ denotes strong bisimulation equivalence [Mil89]. First we show that $\sim_d$ is strongly contained in $\sim_{wd}$.

Proposition B.5 We have $\sim_d \subsetneq \sim_{wd}$.

Proof The proof that $\sim_d$ is contained in $\sim_{wd}$ follows directly from the definition of $\sim_d$ and $\sim_{wd}$. For the proof of strong inclusion let $p = ab|cd + a + ab + c + cd + a|c + ab|c + a|cd$ and $q = a + ab + c + cd + a|c + ab|c + a|cd$. Then $p \not\sim_d q$ because $p \xrightarrow{a} \langle b, nil|cd \rangle$ while for all q', q'' such that $q \xrightarrow{a} (q', q'')$ we have either $q' = nil$ or $q'' = nil$ except for the case where the a derivative comes from the summand $ab|c$. But in this case we have $q'' = nil|c$ which is not distributed bisimulation equivalent to $nil|cd$. On the other hand we have $p \sim_{wd} q$ because the set

$$\{(p, q), (nil|cd, nil|cd), (ab|nil, ab|nil), (nil|c, nil|c),$$
$$(a|nil, a|nil), (nil|nil, nil|nil), (b, b), (d, d), (nil, nil)\}$$

is a weakly distributed bisimulation. ■

Proposition B.6 The relations $\sim$ and $\sim_{wd}$ are incompatible where $\sim$ denotes strong bisimulation.

Proof We have $ab + ba \sim a|b$ but $ab + ba \not\sim_{wd} a|b$. On the other hand, $p \not\sim q$ and $p \sim_{wd} q$ where p and q are the processes defined in the previous proposition. ■