



**HAL**  
open science

## A static view of localities

Luca Aceto

► **To cite this version:**

| Luca Aceto. A static view of localities. [Research Report] RR-1483, INRIA. 1991. inria-00075079

**HAL Id: inria-00075079**

**<https://inria.hal.science/inria-00075079>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE  
INRIA-SOPHIA ANTIPOLIS

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P.105  
78153 Le Chesnay Cedex  
France  
Tél.:(1) 39 63 55 11

# Rapports de Recherche

N° 1483

*Programme 2*  
*Calcul Symbolique, Programmation*  
*et Génie logiciel*

## A STATIC VIEW OF LOCALITIES

**Luca ACETO**

**Juillet 1991**



\* R R . 1 4 8 3 \*

## A Static View of Localities

### Sur les processus statiquement répartis

*Luca Aceto*

INRIA Sophia-Antipolis

06560- VALBONNE FRANCE

#### **Abstract.**

This paper proposes alternative, effective characterizations for nets of automata of the location equivalence and preorder recently introduced by Boudol, Castellani, Hennessy and Kiehn. Contrary to the technical development they propose, where locations are dynamically associated to the sub-parts of a process in the operational semantics, the equivalence and preorder we introduce are based on a static association of locations to the parallel components of a net. Following this static approach, it is possible to give these “distributed nets” a standard operational semantics which associates with each net a finite labelled transition system. Using this operational semantics for distributed nets, we introduce effective notions of equivalence and preorder which are shown to coincide with those proposed by Boudol et al.

#### **Résumé.**

Cet article présente une caractérisation, pour les réseaux d'automates, de l'équivalence de répartition récemment proposée par Boudol, Castellani, Hennessy et Kiehn. Contrairement à leur approche, où la répartition d'un système est observée dynamiquement au cours de son comportement, nous adoptons une vision statique, des noms de site étant associés à chaque composant parallèle d'un réseau. Dans cette approche statique de la notion de répartition, le comportement d'un réseau réparti est défini très naturellement. Nous introduisons des notions d'équivalence et de préordre sur les réseaux d'automates répartis, et nous montrons qu'elles coïncident avec celles proposées par Boudol et al. Ce résultat fournit une présentation effective de l'équivalence de répartition.

# A Static View of Localities

Luca Aceto

INRIA-Sophia Antipolis

F-06560 Valbonne Cedex, France

## Abstract

This paper proposes alternative, effective characterizations for nets of automata of the location equivalence and preorder recently introduced by Boudol et al. in [B\*91b]. Contrary to the technical development in the above given reference, where locations are dynamically associated to the sub-parts of a process in the operational semantics, the equivalence and preorder we propose are based on a static association of locations to the *parallel* components of a net. Following this static approach, it is possible to give these “distributed nets” a standard operational semantics which associates with each net a finite labelled transition system. Using this operational semantics for distributed nets, we introduce effective notions of equivalence and preorder which are shown to coincide with those proposed in [B\*91b].

# 1 Introduction

Recently, there has been an increasing interest in semantic theories for process description languages, like CCS [Mil89], CSP [Hoare85], ACP [BK85] and MEIJE [AB84], which take the distributed nature of processes into account. The search for such “distributed” semantics has been spurred by the realization that the standard semantic theories for the above given process algebras may not be well-suited for the description of properties of systems which depend on their distribution in space. An example of such a property is that of *local deadlock*, i.e. deadlock in one of the parallel components of a system, as opposed to *global deadlock*, i.e. deadlock of the system as a whole. Consider for example the solution to a simple mutual exclusion problem, taken from [B\*91a], in which a semaphore is used to serialize the access of two readers to a resource. This system may be described in CCS by

$$\begin{aligned} \text{Reader} &\leftarrow \bar{p}.\text{enter}.\text{exit}.v.\text{Reader} \\ \text{Sem} &\leftarrow p.\bar{v}.\text{Sem} \\ \text{Sys} &\leftarrow (\text{Reader}|\text{Sem}|\text{Reader}) \setminus p \setminus v. \end{aligned}$$

Consider now a version of the above system *Sys* in which one of the readers is faulty and may deadlock after exiting the critical region. By abstracting on the actual form of the deadlocked behaviour, this system may be described in CCS by

$$\begin{aligned} \text{Faulty} &\leftarrow \bar{p}.\text{enter}.\text{exit}.v.\text{Faulty} + \text{exit}.v.NIL \\ \text{FSys} &\leftarrow (\text{Reader}|\text{Sem}|\text{Faulty}) \setminus p \setminus v. \end{aligned}$$

In the standard theory of weak bisimulation equivalence for CCS, usually denoted by  $\approx$ , the systems *Sys* and *FSys* are equivalent, in the sense that  $\text{Sys} \approx \text{FSys}$ . This essentially depends on the fact that both systems implement the specification *Spec* given by

$$\text{Spec} \leftarrow \text{enter}.\text{exit}.\text{Spec}$$

and are thus *globally* deadlock-free. However, if one views *Sys* and *FSys* as distributed systems then it makes sense to distinguish them as *FSys* has the potential for a local deadlock in its faulty component, whilst both readers in *Sys* are deadlock-free.

To the author’s knowledge, the first attempt at providing a semantic theory for CCS which takes the distribution of processes into account and which can be used to analyze properties of systems like local deadlock has been made by Castellani and Hennessy in [CH89], where a notion of *distributed bisimulation* was introduced and studied for a subset of CCS. The resulting semantic theory for processes turned out to enjoy many of the useful properties of the standard one for weak bisimulation. Unfortunately, however, suitable extensions of their theory to full CCS have turned out to be rather difficult, especially with respect to the treatment of the CCS restriction operator. (See [Kiehn89] for a generalization of distributed bisimulation to the whole of finite CCS and a discussion of its technical difficulties.)

Recently, Boudol, Castellani, Hennessy and Kiehn [B\*91a,B\*91b] have reexamined the approach taken in [CH89] and have proposed a semantic theory for CCS which takes the distribution of processes into account and supports many of the proof techniques familiar from the theory of standard weak bisimulation equivalence. The basic idea underlying their approach is that an observer can not only test the ability of a process of performing a specific task, but it can also identify the sub-part of a system which is responsible for it. An observer of this kind would, for instance, be able to distinguish the systems *Sys* and *FSys* because the latter can reach a state in which no further action will be observed at the faulty component, while both components in the former system will remain active throughout its evolution.

The intuitive idea of observation described above is formalized in [B\*91a,B\*91b] by means of an association of *location names* to the parallel components of a distributed system. For reasons explained at length in [B\*91a], such an association of locations to subsystems is conceived of as being part of the observation itself and is dynamic in nature. Technically, the possible effects of these composite observations which can be made of processes, consisting of the request of execution of a task and of an association of a location to the component responsible for it, are formalized by means of transition relations of the form  $\frac{a}{u.l}$ , where  $a$  is an action,  $u$  is a word over the set of locations, standing for the access path to the parallel component of a process which executes action  $a$ , and  $l$  is the new location which is generated by the act of observation. This new operational view of processes is obtained by means of a rather minor variation on the standard transition system semantics for CCS. The new rules which embody the essence of the new discipline of observation are

$$a.p \xrightarrow{c.l} l :: p \quad \text{for any location } l, \quad (1)$$

where  $l :: \_$  is a new location prefixing operator, and

$$p \xrightarrow{u.l} p' \text{ implies } k :: p \xrightarrow{ku.l} k :: p'. \quad (2)$$

In [B\*91a,B\*91b], a standard weak bisimulation equivalence over the resulting labelled transition system for CCS, called *location equivalence* and denoted by  $\approx_l$ , is presented and studied in detail. This equivalence enjoys many of the properties of  $\approx$ . For example, it can be equationally characterized and has a logical counterpart in a slight variation on Hennessy-Milner logic [Mil89].

Location equivalence is a non-interleaving equivalence. In fact, parallelism is a primitive concept of its theory and is not semantically reduced to nondeterminism, as it is the case for the standard semantic theories for the above mentioned process algebras. For example, the processes *Spec* and *Sys* given above are distinguished by it because, although they have the same “interactive behaviour”, the latter is distributed whilst the former is purely sequential. As argued in [Ab89,Ace89], it might be helpful to have a semantic theory for processes based on a notion of “concurrency refinement” preorder formalizing the intuitive idea that a process  $p$  has the same behaviour of a process  $q$  and  $q$  is at least as parallel as  $p$ . Such a theory would, for instance, allow one to state that process *Sys* is a

concurrent refinement of *Spec*, but the converse does not hold. In general, a semantic theory of this kind allows one to relate nondeterministic specifications, like *Spec*, and concurrent implementations, like *Sys*, and can be used to enforce that an implementation exhibits at least all the parallelism present in the specification. Based on the location transition system semantics, Boudol et al. also propose a notion of “concurrency refinement” preorder, called *location preorder* and denoted by  $\lesssim_l$ , over full CCS and study some of its properties in [B\*91b].

The dynamic formalization of the association of locations to subsystems by means of rule (1) above is technically very useful. It paves the way to a theory of distributed processes which is based on standard techniques from bisimulation semantics and allows a smooth semantic treatment of all the CCS combinators. However, these advantages are obtained, to a certain extent, at the cost of reducing the intuitive appeal of the theory. For example, intuitively one would expect location names to be associated with the operands of the parallel composition operator, rather than being encoded as “access paths” by means of the action-prefixing operators, as it is done in rule (1). Moreover, essentially because of the same rule, the location transition system semantics for CCS given in [B\*91a,B\*91b] associates an infinite state and infinitely branching labelled transition system with each non-trivial CCS process. For example, the regular CCS process *Spec* can perform the transition

$$Spec \xrightarrow[e.l]{enter} l :: exit.Spec$$

for each location  $l$ . The versions of location equivalence and preorder for CCS based on such a transition system semantics are thus not effective over finite state processes. This means that the original versions of these relations do not readily lend themselves to an automated analysis. Moreover, the standard algorithms for model-checking cannot be used to verify whether a CCS process satisfies a property specified by a formula in the logic which characterizes location equivalence.

The aim of this paper is to provide alternative characterizations of the location equivalence and preorder over nets of automata, which are more in line with the intuitive idea that locations should be associated with subparts of processes in relation with their parallel structure rather than by means of an encoding via access paths. These “static” versions of  $\approx_l$  and  $\lesssim_l$  will turn out to be effective and open the possibility of an automated analysis of these relations.

*Nets of automata* are built from regular processes by means of the static combinators of CCS, namely parallel composition, restriction and relabelling. They form a rather rich class of finite state processes which, on the one hand, can be used to specify many interesting concurrent systems and, on the other, is amenable to automated verification techniques. The basic idea upon which the static versions of the location equivalence and preorder are based is that a net of automata can be viewed as a distributed system by *statically* assigning different locations to its parallel components. This may be thought of as an assignment “at compile time” of different virtual processors to the parallel components of a net. The static annotation of the parallel components of nets with locations is made possible by the fact that the distributed structure of these processes is itself static and is

reflected in the syntactic form of the process expressions. These “distributed nets” can then be given a standard operational semantics which associates a *finite* labelled transition system with each annotated net. The transition system semantics is based on relations of the form  $\xrightarrow[l]{a}$ . Intuitively, for distributed nets  $d$  and  $d'$ ,  $d \xrightarrow[l]{a} d'$  will mean that the parallel component of  $d$  at location  $l$  is capable of performing action  $a$  according to the standard CCS operational semantics.

The operational semantics for distributed nets will then be used to define an equivalence and a preorder, which will induce effective characterizations of  $\approx_l$  and  $\lesssim_l$  over nets. These versions of the relations defined in [B\*91b] will, we hope, also reinforce the location equivalence and preorder as very natural bases for a semantic theory of processes which takes their distribution into account and will shed new light on, and provide intuitive justification for, some technical choices made in [B\*91b]. The static versions of  $\approx_l$  and  $\lesssim_l$  over nets will be based on refinements of bisimulation equivalence for distributed nets. Two annotated nets will be considered to be equivalent if they are weakly bisimilar modulo an association between their locations. Intuitively, in order for two distributed nets  $d$  and  $d'$  to be related modulo an association  $\phi$ , it will be required that, for each pair of locations  $(l, l')$  in  $\phi$ , the parallel component of  $d$  at location  $l$  has the same behaviour of the parallel component of  $d'$  at location  $l'$ . It will be shown how, by varying the kind of permitted association between locations, it is possible to obtain an equivalence and a preorder over distributed nets which induce static versions of  $\approx_l$  and  $\lesssim_l$  over nets.

The paper is organized as follows. Section 2 is devoted to a brief summary of material from [B\*91a, B\*91b]. In particular, this section introduces the notions of location equivalence and preorder and recalls some of their basic properties. The language of distributed processes is introduced in §3, together with its operational semantics. Section 4 is devoted to the discussion of an equivalence relation over distributed nets and of the equivalence this induces over nets of automata. The coincidence of the new equivalence with location equivalence over the language of nets of automata is proven in §5. Finally sections 6 and 7 are devoted to the discussion of an effective version of the location preorder over nets of automata.

## 2 Location Equivalence for Nets of Automata

The basic building blocks of nets of automata are the so-called regular processes. These are syntactic descriptions of finite, nondeterministic automata familiar from the theory of regular languages. Formally, let  $\Lambda = \{\alpha, \beta, \dots\}$  be a countable set of *names* and  $V$  be a countable set of *process variables*. The set of *co-names* is given by  $\bar{\Lambda} = \{\bar{\alpha}, \bar{\beta}, \dots\}$ ; we shall use  $Act$  to stand for  $\Lambda \cup \bar{\Lambda}$  and  $a, b$  to range over  $Act$ . The complementation notation is extended to the whole of  $Act$  by assuming that  $\bar{\bar{\alpha}} = \alpha$ , for all  $\alpha \in \Lambda$ . Finally,  $Act_\tau$  will be used to stand for  $Act \cup \{\tau\}$ , where  $\tau$  is a symbol not occurring in  $Act$  which, following Milner, will be used to denote internal, unobservable activity. We shall use  $\mu$  to range over  $Act_\tau$ .



The set of *regular processes* over  $Act_\tau$  and  $V$ ,  $\mathcal{RP}$  ( $p, q \in \mathcal{RP}$ ), is the set of closed terms generated by the following grammar:

$$p ::= NIL \mid \mu.p \mid p + p \mid x \mid rec\ x. p,$$

where  $\mu \in Act_\tau$ ,  $x \in V$  and  $rec\ x. \_$  is the binding construct. Nets of automata are essentially obtained by combining regular processes by means of the static process combinators of CCS [Mil89]. Formally, the set of *nets of automata*  $\mathcal{N}$  ( $n, m, \dots \in \mathcal{N}$ ) is given by the grammar:

$$n ::= p \mid n|n \mid n \setminus \alpha \mid n[f],$$

where  $p \in \mathcal{RP}$ ,  $\alpha \in A$  and  $f : Act_\tau \rightarrow Act_\tau$  is a relabelling, i.e. a function which preserves complementation of actions and leaves  $\tau$  fixed. Intuitively,  $n \setminus \alpha$  will denote a process which behaves like  $n$ , but with all actions  $\alpha, \bar{\alpha}$  forbidden;  $n[f]$  will stand for a process which can perform action  $f(a)$  whenever  $n$  can perform  $a$ . The process  $n|m$ , the parallel composition of  $n$  and  $m$ , can perform any interleaving of the actions of  $n$  and  $m$ ; in addition it can perform a synchronization move whenever  $n$  and  $m$  can execute complementary actions.

In order to define a location operational semantics for  $\mathcal{N}$  based on the one presented in [B\*91b], we assume a given countable set of location names  $Loc$  and introduce a *location prefixing* operator  $u :: \_$  for each  $u \in Loc^*$ . We shall use  $u, v$  to range over  $Loc^*$  and  $\lambda$  to range over  $Loc \cup \{\varepsilon\}$ . The set of *states*  $S$  ( $s, s', \dots \in S$ ) is given by the grammar:

$$s ::= p \mid u :: s \mid s \setminus \alpha \mid s[f] \mid s|s.$$

Note that  $\mathcal{N}$  is a sublanguage of  $S$ . The location operational semantics for  $S$  is given in terms of next-state relations of the form  $\xrightarrow{\tau}$  and  $\xrightarrow[u, \lambda]{a}$ , for  $a \in Act$ ,  $\lambda \in Loc \cup \{\varepsilon\}$  and  $u \in Loc^*$ . Formally,  $\xrightarrow{\tau}$  is the least binary relation over  $S$  which satisfies the axiom and rules in Figure 1. This is a straightforward extension to the language  $S$  of the standard  $\tau$ -transition relation over CCS terms. The defining rules of  $\xrightarrow[u, \lambda]{a}$  are given in Figure 2 and are a slight modification of those presented in [B\*91b]. (The only difference is that in rule (L1) we allow transitions which do not generate any new location, i.e. transitions with  $\lambda = \varepsilon$ .) The interested reader is referred to that reference and the companion paper [B\*91a] for detailed comments on the defining rules of the location transitions. Intuitively,  $s \xrightarrow[u, \lambda]{a} s'$  for some state  $s'$  if the parallel component of  $s$  with access path  $u$  can perform action  $a$  and generate location  $\lambda$  in doing so. The weak transition relations  $\xRightarrow{\varepsilon}$  and  $\xRightarrow[u, \lambda]{a}$ , which abstract from the internal evolution of processes, are defined in the standard way as  $\xrightarrow{\tau}^*$  and  $\xRightarrow{\varepsilon} \circ \xrightarrow[u, \lambda]{a} \circ \xRightarrow{\varepsilon}$ , respectively.

The following definition, taken from [B\*91b], introduces the notion of *generalized bisimulation*. This is just a mild generalization of standard weak bisimulation [Mil89] and of the bisimulation induced by a preorder on actions studied in [Thom87].

**Definition 2.1 (Generalized Bisimulation)** *Let  $R$  be a preorder over  $Loc^*$  which is preserved by concatenation of words. Then  $G_R$  is the largest relation over  $S$  which satisfies, for all  $s_1, s_2 \in S$ ,*

$$s_1 G_R s_2 \text{ if for all } a \in Act, u \in Loc^* \text{ and } l \in Loc,$$

---

(T1) $\mu.p \xrightarrow{\mu} p$	
(T2) $p \xrightarrow{\mu} p'$	implies $p + q \xrightarrow{\mu} p'$ $q + p \xrightarrow{\mu} p'$
(T3) $p[\text{rec } x. p/x] \xrightarrow{\mu} p'$	implies $\text{rec } x. p \xrightarrow{\mu} p'$
(T4) $s \xrightarrow{\mu} s'$	implies $u :: s \xrightarrow{\mu} u :: s'$
(T5) $s \xrightarrow{\mu} s'$ and $\mu \notin \{\alpha, \bar{\alpha}\}$	implies $s \setminus \alpha \xrightarrow{\mu} s' \setminus \alpha$
(T6) $s \xrightarrow{\mu} s'$	implies $s[f] \xrightarrow{f(\mu)} s'[f]$
(T7) $s_1 \xrightarrow{\mu} s'_1$	implies $s_1   s_2 \xrightarrow{\mu} s'_1   s_2$ $s_2   s_1 \xrightarrow{\mu} s_2   s'_1$
(T8) $s_1 \xrightarrow{a} s'_1$ and $s_2 \xrightarrow{a} s'_2$	imply $s_1   s_2 \xrightarrow{\tau} s'_1   s'_2$

---

Figure 1: Axiom and rules for  $\xrightarrow{\tau}$

- (1)  $s_1 \xrightarrow[u.l]{a} s'_1$  implies  $s_2 \xrightarrow[v.l]{a} s'_2$  for some  $s'_2$  and  $v$  such that  $uRv$  and  $s'_1 G_R s'_2$ ;
- (2)  $s_2 \xrightarrow[u.l]{a} s'_2$  implies  $s_1 \xrightarrow[v.l]{a} s'_1$  for some  $s'_1$  and  $v$  such that  $vRu$  and  $s'_1 G_R s'_2$ ;
- (3)  $s_1 \xrightarrow{e} s'_1$  implies  $s_2 \xrightarrow{e} s'_2$  for some  $s'_2$  such that  $s'_1 G_R s'_2$ ;
- (4)  $s_2 \xrightarrow{e} s'_2$  implies  $s_1 \xrightarrow{e} s'_1$  for some  $s'_1$  such that  $s'_1 G_R s'_2$ .

The properties of the relation  $G_R$  over CCS are studied in detail in [B\*91b]<sup>1</sup>. For the sake of completeness, we just recall that  $G_R$  is in general a preorder and that it is an equivalence relation if  $R$  is. Note that  $G_R$  coincides with standard weak bisimulation equivalence,  $\approx$ , if we take  $R$  to be the universal relation over  $Loc^*$ . In what follows, we shall provide effective characterizations of the relations obtained by considering  $R = Id$ , the identity relation over  $Loc^*$ , and  $R = \prec$ , the subword relation over  $Loc^*$ . In [B\*91b] these two relations are called *location equivalence*, denoted by  $\approx_l$ , and *location preorder*, denoted by  $\lesssim_l$ , respectively. The interested reader is referred to [B\*91a, B\*91b] for several, well-argued examples of nets which are equated or differentiated by  $\approx_l$  and  $\lesssim_l$ . Intuitively, two nets  $n_1$  and  $n_2$  are location equivalent if they are weakly bisimilar and they have the same distributed structure in every computation. A characteristic of  $\approx_l$  is that it equates two nets only if they can exhibit the same degree of parallelism in every computation. On the other hand, the

---

<sup>1</sup>The definition given above differs slightly from the one given in [B\*91b]. There the authors take  $R$  to be a reflexive relation over  $Loc^*$  which is preserved by concatenation of words. Such a degree of generality will not be needed in this paper.

---

(L1) $p \xrightarrow{a} p'$	implies $p \xrightarrow[\varepsilon \cdot \lambda]{a} \lambda :: p' \quad (\lambda \in Loc \cup \{\varepsilon\})$
(L2) $s \xrightarrow[u \cdot \lambda]{a} s'$	implies $v :: s \xrightarrow[vu \cdot \lambda]{a} v :: s'$
(L3) $s \xrightarrow[u \cdot \lambda]{a} s'$ and $a \notin \{\alpha, \bar{\alpha}\}$	implies $s \setminus \alpha \xrightarrow[u \cdot \lambda]{a} s' \setminus \alpha$
(L4) $s \xrightarrow[u \cdot \lambda]{a} s'$	implies $s[f] \xrightarrow[u \cdot \lambda]{f(a)} s'[f]$
(L5) $s_1 \xrightarrow[u \cdot \lambda]{a} s'_1$	implies $s_1   s_2 \xrightarrow[u \cdot \lambda]{a} s'_1   s_2$ $s_2   s_1 \xrightarrow[u \cdot \lambda]{a} s_2   s'_1$

---

Figure 2: Defining rules for location transitions

preorder  $\lesssim_l$  is an attempt at formalizing the idea that, for nets  $n_1$  and  $n_2$ ,  $n_1 \lesssim_l n_2$  if the two nets are weakly bisimilar and  $n_2$  is at least as distributed as  $n_1$ .

From [B\*91a,B\*91b] we know that  $\approx_l$  and  $\lesssim_l$  are both preserved by the static CCS combinators. Moreover, they satisfy the laws

$$s = \varepsilon :: s$$

$$l_1 :: \dots :: l_n :: s = l_1 \dots l_n :: s,$$

where  $s \in S$ .

**Definition 2.2 (Location Relabellings)** A location relabelling is a map  $\rho : Loc \rightarrow Loc^*$ . For each  $s \in S$ ,  $s\rho$  will denote the state obtained by syntactically replacing each  $l$  occurring in  $s$  with  $\rho(l)$ .

The standard notations for substitutions will be used throughout the paper. For example, for each  $X \subseteq Loc$ ,  $\lambda \in Loc \cup \{\varepsilon\}$  and location relabelling  $\rho$ ,  $\rho[X \rightarrow \lambda]$  will denote the location relabelling such that

$$\rho[X \rightarrow \lambda](l) = \begin{cases} \lambda & \text{if } l \in X \\ \rho(l) & \text{otherwise.} \end{cases}$$

We shall use  $[l_1 \rightarrow \lambda_1, \dots, l_n \rightarrow \lambda_n]$ , where the  $l_i$ 's are all distinct locations, to denote a relabelling which maps  $l_i$  to  $\lambda_i$ ,  $i \in \{1, \dots, n\}$ , and acts like the identity on  $Loc \setminus \{l_1, \dots, l_n\}$ .

The following property of  $\approx_l$  and  $\lesssim_l$  will be useful in the remainder of the paper. (See [B\*91a], Proposition 3.6.)

**Fact 2.1**  $\approx_l$  and  $\lesssim_l$  are preserved by location relabellings.

To end this section, we shall introduce an alternative version of location equivalence which will be useful in § 5. Let  $\approx_l^f$  be the largest symmetric relation over  $S$  which satisfies

$s_1 \approx_l^f s_2$  if for all  $a \in Act$ ,  $u \in Loc^*$  and  $\lambda \in Loc \cup \{\varepsilon\}$ ,

(1)  $s_1 \xrightarrow[u, \lambda]{a} s'_1$  implies  $s_2 \xrightarrow[u, \lambda]{a} s'_2$  for some  $s'_2$  such that  $s'_1 \approx_l^f s'_2$ ;

(2)  $s_1 \xrightarrow{e} s'_1$  implies  $s_2 \xrightarrow{e} s'_2$  for some  $s'_2$  such that  $s'_1 \approx_l^f s'_2$ .

The only difference between the definitions of  $\approx_l$  and  $\approx_l^f$  is that in the latter we also require matching of transitions which do not generate new location names, i.e. transitions of the form  $\xrightarrow[u, e]{a}$ . We shall now show that this does not increase the discriminating power of the resulting equivalence.

**Notation 2.1** For all  $s \in S$ ,  $Loc(s)$  will denote the set of location names occurring in  $s$ .

The following lemma relates the transitions of a state  $s$  of the form  $s \xrightarrow[u, e]{a} s'$  for some  $s'$  with those which generate new locations, i.e. those of the form  $s \xrightarrow[u, l]{a} s'$ .

**Lemma 2.1** Let  $s \in S$  and  $l \notin Loc(s)$ .

(1) Assume that  $s \xrightarrow[u, e]{a} s'$ . Then there exists  $\hat{s} \in S$  such that  $s \xrightarrow[u, l]{a} \hat{s}$  and  $s' = \hat{s}[l \rightarrow \varepsilon]$ .

(2) Assume that  $s \xrightarrow[u, l]{a} s'$ . Then  $s \xrightarrow[u, e]{a} s'[l \rightarrow \varepsilon]$ .

Using the above lemma we can now show that  $\approx_l$  and  $\approx_l^f$  coincide over  $S$ .

**Proposition 2.1** For all  $s_1, s_2 \in S$ ,  $s_1 \approx_l s_2$  iff  $s_1 \approx_l^f s_2$ .

**Proof:** The only interesting thing to check is that  $\approx_l \subseteq \approx_l^f$ . To this end, we shall show that  $\approx_l$  satisfies the defining clauses of  $\approx_l^f$ . In order to prove this statement, by symmetry, it is sufficient to show that, for all  $s_1, s_2 \in S$  such that  $s_1 \approx_l s_2$

$$s_1 \xrightarrow[u, e]{a} s'_1 \text{ implies } s_2 \xrightarrow[u, e]{a} s'_2 \text{ for some } s'_2 \text{ such that } s'_1 \approx_l s'_2. \quad (3)$$

We now prove (3). Assume that  $s_1 \approx_l s_2$  and that  $s_1 \xrightarrow[u, e]{a} s'_1$ . Let  $l \notin Loc(s_1) \cup Loc(s_2)$ . Then, by Lemma 2.1(1), there exists  $\hat{s}_1$  such that  $s_1 \xrightarrow[u, l]{a} \hat{s}_1$  and  $s'_1 = \hat{s}_1[l \rightarrow \varepsilon]$ . As  $s_1 \approx_l s_2$ , there exists  $\hat{s}_2$  such that  $s_2 \xrightarrow[u, l]{a} \hat{s}_2$  and  $\hat{s}_1 \approx_l \hat{s}_2$ . By Lemma 2.1(2),  $s_2 \xrightarrow[u, l]{a} \hat{s}_2$  implies  $s_2 \xrightarrow[u, e]{a} \hat{s}_2[l \rightarrow \varepsilon]$ . As  $\hat{s}_1 \approx_l \hat{s}_2$ , we have that  $\hat{s}_1[l \rightarrow \varepsilon] \approx_l \hat{s}_2[l \rightarrow \varepsilon]$  because  $\approx_l$  is preserved by location relabellings.  $\square$

### 3 Distributed Processes and Their Operational Semantics

The operational semantics for  $\mathcal{N}$  given in § 2 is based on the dynamic creation of location names by means of rule (L1) in Figure 2. This gives rise to the association of an infinite state and infinitely branching labelled transition systems (lts) to all non-trivial nets of automata. For example, the finite, regular process  $a.b.NIL$  is capable of performing the transition  $a.b.NIL \xrightarrow[e, l]{a} l :: b.NIL$  for all  $l \in Loc$ . A consequence of this fact is that one cannot readily apply the decision algorithms for bisimulation-like relations over finite lts's to the relations  $\approx_l$  and  $\lesssim_l$  over  $\mathcal{N}$ . In order to overcome

this problem, we shall present an alternative, static version of the location operational semantics. This operational semantics will be used in the remainder of the paper to define effective versions over  $\mathcal{N}$  of the location equivalence and preorder considered in [B\*91b].

The basic idea is that a net of automata can be viewed as a “distributed process” by statically assigning a different location name to each of its parallel components. We shall then be able to define a standard labelled transition semantics for these annotated nets which associates with each such net a *finite* lts.

**Definition 3.1 (Distributed Processes)** *The set  $\mathcal{D}$  of distributed processes is the least one satisfying the following rules:*

- $p \in \mathcal{RP}$  and  $l \in \text{Loc}$  imply  $l :: p \in \mathcal{D}$  and  $\text{Loc}(l :: p) = \{l\}$ ,
- $d \in \mathcal{D}$  implies  $d \setminus \alpha, d[f] \in \mathcal{D}$  and  $\text{Loc}(d \setminus \alpha) = \text{Loc}(d[f]) = \text{Loc}(d)$ ,
- $d_1, d_2 \in \mathcal{D}$  and  $\text{Loc}(d_1) \cap \text{Loc}(d_2) = \emptyset$  imply  $d_1|d_2 \in \mathcal{D}$  and  $\text{Loc}(d_1|d_2) = \text{Loc}(d_1) \cup \text{Loc}(d_2)$ .

We shall use  $d, d', d_1 \dots$  to range over  $\mathcal{D}$ .

The operational semantics for distributed processes is given in terms of transition relations of the form  $\xrightarrow[l]{a} \mathcal{D}$  and  $\xrightarrow{\tau} \mathcal{D}$ . The defining rules for these relations may be found in Figure 3. Intuitively, for distributed processes  $d$  and  $d'$ ,  $d \xrightarrow[l]{a} d'$  if the parallel component of  $d$  at location  $l$  can perform an  $a$ -action according to the standard CCS operational semantics for regular processes. The weak transition relations over  $\mathcal{D}$ ,  $\xRightarrow[l]{a} \mathcal{D}$  and  $\xRightarrow{\epsilon} \mathcal{D}$ , are defined in the standard way. The following facts are easily seen to hold:

**Fact 3.1** *Let  $d \in \mathcal{D}$ . Then:*

- (1)  $d \xRightarrow[l]{a} d'$  implies  $l \in \text{Loc}(d)$  and  $\text{Loc}(d) = \text{Loc}(d')$ , and
- (2)  $d \xRightarrow{\epsilon} d'$  implies  $\text{Loc}(d) = \text{Loc}(d')$ .

To end this section, we introduce a notation which will find considerable application in the remainder of the paper.

**Notation 3.1** *For each  $d \in \mathcal{D}$ , we shall use  $\pi(d)$  to denote the underlying net of  $d$ , i.e. the net obtained by abstracting from the location names in  $d$ .*

## 4 An Equivalence over $\mathcal{D}$

We shall now define a notion of equivalence over  $\mathcal{D}$  which takes the information about the distribution of processes into account, as expressed by the labelling of their parallel components with location names. The equivalence over  $\mathcal{D}$  will then be used to induce an equivalence over  $\mathcal{N}$  via the map  $\pi$ . This equivalence will turn out to coincide with the location equivalence  $\approx_l$  of [B\*91b].

---

(D1) $p \xrightarrow[a]{i} p'$	implies $l :: p \xrightarrow[a]{i} p' \quad l :: p'$
(D2) $d \xrightarrow[a]{i} d'$	implies $d[f] \xrightarrow[f(a)]{i} d'[f]$
(D3) $d \xrightarrow[a]{i} d'$ and $a \notin \{\alpha, \bar{\alpha}\}$	imply $d \setminus \alpha \xrightarrow[a]{i} d' \setminus \alpha$
(D4) $d_1 \xrightarrow[a]{i} d'_1$	implies $d_1 d_2 \xrightarrow[a]{i} d'_1 d_2$ $d_2 d_1 \xrightarrow[a]{i} d_2 d'_1$
(DI1) $p \xrightarrow{\tau} p'$	implies $l :: p \xrightarrow{\tau} p' \quad l :: p'$
(DI2) $d \xrightarrow{\tau} d'$	implies $d[f] \xrightarrow{\tau} d'[f]$
(DI3) $d \xrightarrow{\tau} d'$	implies $d \setminus \alpha \xrightarrow{\tau} d' \setminus \alpha$
(DI4) $d_1 \xrightarrow{\tau} d'_1$	implies $d_1 d_2 \xrightarrow{\tau} d'_1 d_2$ $d_2 d_1 \xrightarrow{\tau} d_2 d'_1$
(DI5) $d_1 \xrightarrow[a]{i} d'_1$ and $d_2 \xrightarrow[a']{i'} d'_2$	imply $d_1 d_2 \xrightarrow{\tau} d'_1 d'_2$

---

Figure 3: Defining rules for distributed transitions

Intuitively, two distributed processes  $d_1$  and  $d_2$  will be equivalent or inequivalent with respect to an association  $\phi$  between their locations. Roughly, one may think of  $\phi$  as a set of constraints on the observations which are possible of  $d_1$  and  $d_2$ , in the sense that for  $d_1$  and  $d_2$  to be equivalent modulo  $\phi$  it must be the case that, for each  $(l, l') \in \phi$ , the observable behaviour of the parallel component of  $d_1$  at location  $l$  must be the same of that of the parallel component of  $d_2$  at location  $l'$ . In order to require that  $d_1$  and  $d_2$  be equivalent modulo  $\phi$  only if they can exhibit the same degree of parallelism, the association  $\phi$  will take the form of (the graph of) a partial bijection  $\phi : Loc \rightarrow Loc$ .

**Definition 4.1** Let  $\mathcal{F}$  denote the set of partial bijections from  $Loc$  onto  $Loc$ .

(1) A pair of locations  $(l, l')$  is consistent with  $\phi \in \mathcal{F}$  iff

- $l \notin \text{dom}(\phi)$  and  $l' \notin \text{range}(\phi)$ , or
- $(l, l') \in \phi$ .

(2) An  $\mathcal{F}$ -indexed family of relations over  $\mathcal{D}$   $\{R_\phi \mid \phi \in \mathcal{F}\}$  is symmetric iff for all  $\phi \in \mathcal{F}$ ,

$$R_{\phi^{-1}} = (R_\phi)^{-1}.$$

Note that if  $\phi \in \mathcal{F}$  and  $(l, l')$  is compatible with it then  $\phi \cup \{(l, l')\} \in \mathcal{F}$ . Intuitively, for two distributed processes  $d_1$  and  $d_2$  to be equivalent with respect to an association  $\phi$  we require that:

(1) each observable transition  $d_1 \xrightarrow[l]{a} d'_1$  of  $d_1$  must be matched by an observable transition of  $d_2$ ,  $d_2 \xrightarrow[l']{a} d'_2$ , in such a way that the pair of locations  $(l, l')$  is consistent with  $\phi$  and the states  $d'_1$  and  $d'_2$  are related with respect to the augmented association  $\phi \cup \{(l, l')\}$ . The notion of consistency given above ensures that if  $l$  is a *new* location not occurring in  $\text{dom}(\phi)$ , which we take to mean that  $l$  is a location where no action has yet been observed, then the matching transition of  $d_2$  must also occur at a new location. Conversely, if  $l$  is a location in  $\text{dom}(\phi)$ , which we take to mean that some action has previously been observed at  $l$ , then the matching transition of  $d_2$  must occur at the unique location  $l'$  which had previously been associated with it in  $\phi$ ;

(2) silent transitions must be matched as usual.

This is the intuitive content of the following formal definition:

**Definition 4.2**  $\simeq_{\mathcal{F}} = \{\simeq_{\phi} \mid \phi \in \mathcal{F}\}$  is the largest symmetric  $\mathcal{F}$ -indexed family of relations over  $\mathcal{D}$  satisfying:

$d_1 \simeq_{\phi} d_2$  if for all  $a \in \text{Act}$ ,  $l \in \text{Loc}$ ,

(1)  $d_1 \xrightarrow[l]{a} d'_1$  implies that there exist  $l' \in \text{Loc}$  and  $d'_2 \in \mathcal{D}$  such that  $(l, l')$  is consistent with  $\phi$ ,  $d_2 \xrightarrow[l']{a} d'_2$  and  $d'_1 \simeq_{\phi \cup \{(l, l')\}} d'_2$ ;

(2)  $d_1 \xrightarrow{\epsilon} d'_1$  implies that there exists  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow{\epsilon} d'_2$  and  $d'_1 \simeq_{\phi} d'_2$ .

The family of relations  $\simeq_{\mathcal{F}}$  can be used to induce an equivalence relation over nets of automata. Intuitively, two nets  $n_1$  and  $n_2$  will be equivalent if there exist two annotations for them such that the resulting distributed processes are related modulo the empty association.

**Definition 4.3** For all  $n_1, n_2 \in \mathcal{N}$ , we write  $n_1 \simeq n_2$  iff there exist  $d_1, d_2 \in \mathcal{D}$  such that  $\pi(d_i) = n_i$  ( $i=1,2$ ) and  $d_1 \simeq_{\emptyset} d_2$ .

We shall now prove that  $\simeq$  is indeed an equivalence relation over  $\mathcal{N}$ . This will follow from the following lemma, which states some basic properties of the family of relations  $\simeq_{\mathcal{F}}$ .

**Lemma 4.1 (Properties of  $\simeq_{\mathcal{F}}$ )**

(1) For all  $d \in \mathcal{D}$ ,  $d \simeq_{1(d)} d$ , where  $1(d) = \{(l, l) \mid l \in \text{Loc}(d)\}$ .

(2) For all  $d_1, d_2, d_3 \in \mathcal{D}$ ,  $\phi, \varphi \in \mathcal{F}$ ,  $d_1 \simeq_{\phi} d_2$  and  $d_2 \simeq_{\varphi} d_3$  imply  $d_1 \simeq_{\varphi \circ \phi} d_3$ .

(3) For all  $d_1, d_2 \in \mathcal{D}$ ,  $\phi, \varphi \in \mathcal{F}$ ,  $d_1 \simeq_{\phi} d_2$  and  $\varphi \subseteq \phi$  imply  $d_1 \simeq_{\varphi} d_2$ .

As an immediate consequence of the above lemma, we have that  $\simeq$  is an equivalence relation over  $\mathcal{N}$ .

**Proposition 4.1**  $\simeq$  is an equivalence relation over  $\mathcal{N}$ .

**Proof:**  $\simeq$  is reflexive by Lemma 4.1(1) and (3). It is symmetric by the definition of  $\simeq_{\mathcal{F}}$  and transitive by Lemma 4.1(2).  $\square$

A few examples of nets which are equivalent or inequivalent with respect to  $\simeq$  are now in order. These examples will hopefully clarify the role played by the consistency requirement in the definition of  $\simeq_{\mathcal{F}}$  and, consequently, of  $\simeq$ .

#### Example 4.1

(1)  $p = a.b.NIL + b.a.NIL \neq a.NIL|b.NIL = n$ . In fact, let  $d_1 = l :: p$  and  $d_2 = l_1 :: a.NIL|l_2 :: b.NIL$ , with  $l_1 \neq l_2$ . Then, in order for  $d_1 \simeq_{\emptyset} d_2$  to hold, we should have that

$$l :: b.NIL \simeq_{\{(l,l_1)\}} l_1 :: NIL|l_2 :: b.NIL. \quad (4)$$

However, (4) does not hold because  $(l, l_2)$  is not compatible with  $\{(l, l_1)\}$ . Intuitively, in order for  $d_1$  to be equivalent to  $d_2$ , location  $l$  in  $d_1$  should play the role of two locations in  $d_2$ . This is not in agreement with the idea that two equivalent distributed processes should have the same degree of distribution and is formally disallowed by the notion of consistency.

(2)  $n = (\alpha.a.NIL|(\bar{\alpha} + a)) \setminus \alpha \simeq a.NIL = p$ . In fact, let  $d_1 = (l_1 :: \alpha.a.NIL|l_2 :: (\bar{\alpha} + a)) \setminus \alpha$  and  $d_2 = l :: p$ . Then the family of relations given by:

$$\begin{aligned} R_{\emptyset} &= \{ \langle d_1, d_2 \rangle, \langle d_2, d_1 \rangle, \langle (l_1 :: a.NIL|l_2 :: NIL) \setminus \alpha, l :: p \rangle, \langle l :: p, (l_1 :: a.NIL|l_2 :: NIL) \setminus \alpha \rangle \} \\ R_{\{(l_1, l)\}} &= \{ \langle (l_1 :: NIL|l_2 :: NIL) \setminus \alpha, l :: NIL \rangle \} \\ R_{\{(l_1, l_1)\}} &= \{ \langle l :: NIL, (l_1 :: NIL|l_2 :: NIL) \setminus \alpha \rangle \} \\ R_{\{(l_2, l)\}} &= \{ \langle (l_1 :: \alpha.a.NIL|l_2 :: NIL) \setminus \alpha, l :: NIL \rangle \} \\ R_{\{(l, l_2)\}} &= \{ \langle l :: NIL, (l_1 :: \alpha.a.NIL|l_2 :: NIL) \setminus \alpha \rangle \} \\ R_{\emptyset} &= \emptyset \text{ otherwise} \end{aligned}$$

satisfies the defining clauses of  $\simeq_{\mathcal{F}}$ .

(3)  $n = (a.(\alpha + b)|\bar{\alpha}.b) \setminus \alpha \neq (a.\alpha|\bar{\alpha}.b) \setminus \alpha = m$ . In fact, let  $d_1 = (l_1 :: a.(\alpha + b)|l_2 :: \bar{\alpha}.b) \setminus \alpha$  and  $d_2 = (l :: a.\alpha|l' :: \bar{\alpha}.b) \setminus \alpha$ , with  $l_1 \neq l_2$  and  $l \neq l'$ . Then, in order to prove that  $d_1 \simeq_{\emptyset} d_2$ , we should have that

$$d'_1 = (l_1 :: (\alpha + b)|l_2 :: \bar{\alpha}.b) \setminus \alpha \simeq_{\{(l_1, l)\}} (l :: \alpha|l' :: \bar{\alpha}.b) \setminus \alpha = d'_2. \quad (5)$$

However, (5) does not hold because the transition  $d'_1 \xrightarrow[l_1]{b} (l_1 :: NIL|l_2 :: \bar{\alpha}.b) \setminus \alpha$  cannot be matched by  $d'_2 \xrightarrow[l']{b} (l :: NIL|l' :: NIL) \setminus \alpha$  because  $(l_1, l')$  is not compatible with  $\{(l_1, l)\}$ .

We shall now show that the definition of  $\simeq$  does not depend on the actual association of locations to sub-processes.



**Definition 4.4** For all  $d_1, d_2 \in \mathcal{D}$  such that  $\pi(d_1) = \pi(d_2)$ , the association of locations  $\phi(d_1, d_2)$  is defined by:

- $\phi(l :: p, l' :: p) = \{(l, l')\}$ ,
- $\phi(d_1 \setminus \alpha, d_2 \setminus \alpha) = \phi(d_1[f], d_2[f]) = \phi(d_1, d_2)$ , and
- $\phi(d_1 | d_2, d'_1 | d'_2) = \phi(d_1, d'_1) \cup \phi(d_2, d'_2)$ .

It is easy to see that  $\phi(d_1, d_2) \in \mathcal{F}$  for all  $d_1, d_2 \in \mathcal{D}$  such that  $\pi(d_1) = \pi(d_2)$ . Moreover, we have that  $\phi(d_2, d_1) = \phi(d_1, d_2)^{-1}$ . The following lemma expresses the relationship between the association  $\phi(d_1, d_2)$  and the labelled transition system semantics for  $d_1$  and  $d_2$ .

**Lemma 4.2** Let  $d_1, d_2 \in \mathcal{D}$  be such that  $\pi(d_1) = \pi(d_2)$ .

- (1) Assume that  $d_1 \xrightarrow{a}_l d'_1$ . Then there exist  $l' \in \text{Loc}$  and  $d'_2 \in \mathcal{D}$  such that  $(l, l') \in \phi(d_1, d_2)$ ,  $d_2 \xrightarrow{a}_{l'} d'_2$ ,  $\pi(d'_1) = \pi(d'_2)$  and  $\phi(d'_1, d'_2) = \phi(d_1, d_2)$ .
- (2) Assume that  $d_1 \xrightarrow{e}_D d'_1$ . Then there exists  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow{e}_D d'_2$ ,  $\pi(d'_1) = \pi(d'_2)$  and  $\phi(d'_1, d'_2) = \phi(d_1, d_2)$ .

**Proof:** Both statements are shown by induction on the length of the relevant derivation. The details are omitted.  $\square$

**Proposition 4.2** For all  $d_1, d_2 \in \mathcal{D}$ ,  $\pi(d_1) = \pi(d_2)$  implies  $d_1 \simeq_\emptyset d_2$ .

**Proof:** By Lemma 4.1(3), it is sufficient to show that  $\pi(d_1) = \pi(d_2)$  implies  $d_1 \simeq_{\phi(d_1, d_2)} d_2$ . To this end, we show that the family of relations  $\{R_\phi \mid \phi \in \mathcal{F}\}$  given by

$$R_\phi = \{\langle d_1, d_2 \rangle \mid \pi(d_1) = \pi(d_2) \text{ and } \phi(d_1, d_2) = \phi\}$$

satisfies the defining clauses of  $\simeq_{\mathcal{F}}$ . This is straightforward using Lemma 4.2.  $\square$

As a corollary of the above result we can now show that the definition of  $\simeq$  is independent of the actual choice of locations for the parallel components of nets.

**Corollary 4.1** For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \simeq n_2$  implies that for all  $d_1, d_2 \in \mathcal{D}$  such that  $\pi(d_i) = n_i$  ( $i=1,2$ ),  $d_1 \simeq_\emptyset d_2$ .

**Proof:** Assume that  $n_1 \simeq n_2$  and  $\pi(d_i) = n_i$  ( $i=1,2$ ). As  $n_1 \simeq n_2$ , there exist  $d'_1, d'_2 \in \mathcal{D}$  such that  $\pi(d'_i) = n_i$  ( $i=1,2$ ) and  $d'_1 \simeq_\emptyset d'_2$ . By the above proposition,  $d_1 \simeq_\emptyset d'_1 \simeq_\emptyset d'_2 \simeq_\emptyset d_2$ . By Lemma 4.1(2), we then have that  $d_1 \simeq_\emptyset d_2$ .  $\square$

This completes our analysis of the definition of  $\simeq$ . In the following section we shall show that  $\simeq$  provides an alternative, effective characterization of location equivalence over the language  $\mathcal{N}$ .

## 5 Relationship with Location Equivalence

We shall now investigate the relationships between location equivalence,  $\approx_l$ , as introduced in [B\*91b], and  $\simeq$  over the language  $\mathcal{N}$ . Our aim in this section is to prove that the two equivalences coincide over  $\mathcal{N}$ . This result will, on the one hand, reinforce  $\approx_l$  as a natural notion of equivalence over nets of automata which takes their distribution into account and, on the other, will provide us with a means of effectively checking location bisimilarity over  $\mathcal{N}$ .

**Note:** From now on, we shall be working modulo the congruence over  $\mathcal{N}$  with respect to the static CCS operators generated by the equalities

$$\varepsilon :: x = x \quad (6)$$

$$l_1 :: \dots :: l_n :: x = l_1 \dots l_n :: x \quad (n \geq 0) \quad (7)$$

The equality symbol will always stand for “equality modulo (6)-(7)”. Note that, with this convention,  $\pi(d) = d[Loc(d) \rightarrow \varepsilon] = d[Loc \rightarrow \varepsilon]$  for all  $d \in \mathcal{D}$ .  $\square$

First of all, we shall show that  $\approx_l$  is contained in  $\simeq$  over  $\mathcal{N}$ . In view of Proposition 2.1, it is sufficient to show that this holds for  $\approx_f^l$ . Our aim is to construct an  $\mathcal{F}$ -indexed family of relations over  $\mathcal{D}$ ,  $\{\tilde{R}_\phi \mid \phi \in \mathcal{F}\}$ , which satisfies the defining clauses of  $\simeq_{\mathcal{F}}$  and such that  $\langle d_1, d_2 \rangle \in \tilde{R}_\phi$  for all  $d_1, d_2 \in \mathcal{D}$  with  $\pi(d_1) \approx_f^l \pi(d_2)$ . Intuitively,  $\langle d_1, d_2 \rangle \in \tilde{R}_\phi$ , for some  $\phi \in \mathcal{F}$ , if it is possible to relabel the location names occurring in  $d_1$  and  $d_2$  in a manner which is consistent with  $\phi$  to obtain two states which are equivalent with respect  $\approx_f^l$ . The relabelling we choose is based on the idea that no observable action has yet been observed at the location names occurring in  $d_1$ , but not in  $dom(\phi)$ . These location names will then be erased in the corresponding state, i.e. relabelled by  $\varepsilon$ . Similarly for the location names occurring in  $d_2$ , but not in  $range(\phi)$ . On the other hand, each pair of locations  $(l, l') \in \phi$  stands for an association between two locations where some observable event has been observed. In order to capture the association between locations  $l$  and  $l'$  in the location operational semantics, we shall relabel  $l'$  by  $l$  in  $d_2$ .

Formally, the family of relations  $\{\tilde{R}_\phi \mid \phi \in \mathcal{F}\}$  is given by

$$\tilde{R}_\phi =_{def} \{ \langle d_1, d_2 \rangle \mid d_1[Loc(d_1) \setminus dom(\phi) \rightarrow \varepsilon] \approx_f^l d_2[Loc(d_2) \setminus range(\phi) \rightarrow \varepsilon, \phi^{-1}] \}.$$

We then have the following theorem:

**Theorem 5.1** *For all  $\phi \in \mathcal{F}$ ,  $\tilde{R}_\phi \subseteq \simeq_\phi$ .*

As a corollary of the above theorem we can then prove the first half of the characterization result.

**Corollary 5.1 (Characterization of  $\approx_l$ : Part 1)** *For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \approx_l n_2$  implies  $n_1 \simeq n_2$ .*

**Proof:** By Proposition 2.1, it is sufficient to prove the result for  $\approx_f^l$ . Assume then that  $n_1 \approx_f^l n_2$  and let  $d_1, d_2 \in \mathcal{D}$  be such that  $\pi(d_i) = n_i$ ,  $i=1,2$ . As  $\pi(d_i) = d_i[Loc(d_i) \rightarrow \varepsilon]$ ,  $i=1,2$ , it is easy to see

that  $\langle d_1, d_2 \rangle \in \tilde{R}_\emptyset$ . By the above theorem, this implies that  $d_1 \simeq_\emptyset d_2$ . Hence,  $n_1 \simeq n_2$ .  $\square$

We are thus left to show Theorem 5.1. This will require a detailed analysis of the location operational behaviour of the states  $d_1[Loc(d_1) \setminus dom(\phi) \rightarrow \varepsilon]$  and  $d_2[Loc(d_2) \setminus range(\phi) \rightarrow \varepsilon, \phi^{-1}]$  and of its relationships with that of  $d_1$  and  $d_2$ , respectively.

**Lemma 5.1** *Let  $d \in \mathcal{D}$  and  $\rho : Loc \rightarrow Loc^*$  be a location relabelling. Then:*

(1)  $d \xrightarrow[l]{a} d'$  implies that  $d\rho \xrightarrow{a} d'\rho$ ;

(2)  $d \xrightarrow[\rho]{\varepsilon} d'$  implies that  $d\rho \xrightarrow{\varepsilon} d'\rho$ .

**Proof:** Statement (1) is shown by induction on the length of the proof of the derivation  $d \xrightarrow[l]{a} d'$ . Statement (2) is proven by induction on the length of the derivation  $d \xrightarrow[\rho]{\varepsilon} d'$ , using (1) for the case  $d \xrightarrow[\rho]{\tau} d'$ . The details are omitted.  $\square$

We can now prove a result which will allow us to derive moves of a state of the form  $d_1[Loc(d_1) \setminus dom(\phi) \rightarrow \varepsilon] \in \mathcal{S}$  from those of  $d_1 \in \mathcal{D}$ .

**Lemma 5.2** *Let  $d \in \mathcal{D}$  and  $X \subseteq Loc$ . Assume that  $d \xrightarrow[l]{a} d'$ .*

(1) *Suppose that  $l \notin X$ . Then  $d[Loc(d) \setminus X \rightarrow \varepsilon] \xrightarrow[\varepsilon, l]{a} d'[Loc(d') \setminus (X \cup \{l\}) \rightarrow \varepsilon]$ .*

(2) *Suppose that  $l \in X$ . Then  $d[Loc(d) \setminus X \rightarrow \varepsilon] \xrightarrow[l, \varepsilon]{a} d'[Loc(d') \setminus X \rightarrow \varepsilon]$ .*

**Proof:** It is sufficient to prove both statements when  $d \xrightarrow[l]{a} d'$  as the more general results will then follow immediately by Lemma 5.1(2). We prove that both statements hold for single step derivations simultaneously by induction on the length of the proof of the derivation  $d \xrightarrow[l]{a} d'$  and only give two representative cases of the proof.

- $d = l :: p \xrightarrow[l]{a} l :: p' = d'$  because  $p \xrightarrow{a} p'$ . We examine the two statements separately.

(1) Assume that  $l \notin X$ . Then, as  $p \xrightarrow{a} p'$  implies that  $p \xrightarrow[\varepsilon, l]{a} l :: p'$ , we have that

$$(l :: p)[\{l\} \rightarrow \varepsilon] = \varepsilon :: p = p \xrightarrow[\varepsilon, l]{a} l :: p'.$$

The claim now follows because  $[\{l\} \setminus (X \cup \{l\}) \rightarrow \varepsilon]$  is the identity location relabelling.

(2) Assume that  $l \in X$ . Then we have that  $(l :: p)[\{l\} \setminus X \rightarrow \varepsilon] = l :: p$ . By the location operational semantics,  $p \xrightarrow{a} p'$  implies  $l :: p \xrightarrow[l, \varepsilon]{a} l :: p'$ . Moreover, we have that  $l :: p' = (l :: p')[\{l\} \setminus X \rightarrow \varepsilon]$ .

- $d = d_1|d_2 \xrightarrow[l]{a} d'_1|d_2 = d'$  because  $d_1 \xrightarrow[l]{a} d'_1$ . Again, we examine the two statements separately.

(1) Assume that  $l \notin X$ . Then, by the inductive hypothesis for (1),

$$d_1[Loc(d_1) \setminus X \rightarrow \varepsilon] \xrightarrow[\varepsilon.l]{a} d'_1[Loc(d'_1) \setminus (X \cup \{l\}) \rightarrow \varepsilon].$$

By the location operational semantics and the fact that  $Loc(d_1) \cap Loc(d_2) = \emptyset$ , we then have that

$$\begin{aligned} (d_1|d_2)[Loc(d) \setminus X \rightarrow \varepsilon] &= (d_1[Loc(d_1) \setminus X \rightarrow \varepsilon])|(d_2[Loc(d) \setminus X \rightarrow \varepsilon]) \\ &\xrightarrow[\varepsilon.l]{a} (d'_1[Loc(d'_1) \setminus (X \cup \{l\}) \rightarrow \varepsilon])|(d_2[Loc(d) \setminus X \rightarrow \varepsilon]) \\ &= (d'_1|d_2)[Loc(d'_1|d_2) \setminus (X \cup \{l\}) \rightarrow \varepsilon] \end{aligned}$$

as  $Loc(d_1) = Loc(d'_1)$  and  $l \notin Loc(d_2)$ . Hence statement (1) is met.

(2) Assume that  $l \in X$ . The proof follows that for the previous case.

The remaining cases are dealt with in similar fashion.  $\square$

In the proof of Theorem 5.1, we shall also make use of decomposition results which will allow us to derive moves of  $d \in \mathcal{D}$  from those of  $d\rho \in \mathcal{S}$ , where  $\rho$  is a location relabelling.

**Lemma 5.3** *Let  $d \in \mathcal{D}$  and  $\rho : Loc \rightarrow Loc^*$  be a location relabelling. Then:*

(1)  $d\rho \xrightarrow{a} s$  implies that there exist  $l \in Loc(d)$  and  $d' \in \mathcal{D}$  such that  $d \xrightarrow[l]{a} d'$  and  $s = d'\rho$ ;

(2)  $d\rho \xrightarrow{\varepsilon} s$  implies that  $d \xrightarrow[\rho]{\varepsilon} d'$  for some  $d' \in \mathcal{D}$  such that  $s = d'\rho$ .

We can now prove our first decomposition result for transitions of the form  $\xrightarrow[\varepsilon.l]{a}$ .

**Lemma 5.4** *Let  $d \in \mathcal{D}$  and  $\phi \in \mathcal{F}$ . Assume that  $d[Loc(d) \setminus range(\phi) \rightarrow \varepsilon, \phi^{-1}] \xrightarrow[\varepsilon.l]{a} s$ . Then there exist  $l' \notin range(\phi)$  and  $d' \in \mathcal{D}$  such that  $d \xrightarrow[l']{a} d'$  and  $s = d'[Loc(d') \setminus (range(\phi) \cup \{l'\}) \rightarrow \varepsilon, \phi^{-1} \cup \{(l', l)\}]$ .*

**Proof:** It is sufficient to prove the statement when  $d[Loc(d) \setminus range(\phi) \rightarrow \varepsilon, \phi^{-1}] \xrightarrow[\varepsilon.l]{a} s$  as the more general result will follow immediately by using Lemma 5.3(2). The proof of this result is by induction on the structure of  $d$ . Here we just give the most interesting case of the proof.

- Assume that  $d = l' :: p$  for some  $p \in \mathcal{R}\mathcal{P}$ . Suppose that

$$(l' :: p)[\{l'\} \setminus range(\phi) \rightarrow \varepsilon, \phi^{-1}] \xrightarrow[\varepsilon.l]{a} s.$$

Then, by the location operational semantics, it is easy to see that it must be the case that  $l' \notin range(\phi)$ . Hence, we have that

$$(l' :: p)[\{l'\} \setminus range(\phi) \rightarrow \varepsilon, \phi^{-1}] = \varepsilon :: p \xrightarrow[\varepsilon.l]{a} s.$$

By the location operational semantics, this implies that  $p \xrightarrow{a} p'$  and  $s = l' :: p'$  for some  $p' \in \mathcal{R}\mathcal{P}$ . Hence, by rule (D1) in Figure 3,  $l' :: p \xrightarrow[l']{a} l' :: p' = d'$ . Moreover,

$$(l' :: p')[\{l'\} \setminus (range(\phi) \cup \{l'\}) \rightarrow \varepsilon, \phi^{-1} \cup \{(l', l)\}] = l' :: p' = s. \quad \square$$

In the proof of Theorem 5.1 we shall also need a decomposition result for transitions of the form  $\xrightarrow[l.\varepsilon]{a}$ .

**Lemma 5.5** *Let  $d \in \mathcal{D}$  and  $\phi \in \mathcal{F}$ . Assume that  $d[\text{Loc}(d) \setminus \text{range}(\phi) \rightarrow \varepsilon, \phi^{-1}] \xrightarrow[l.\varepsilon]{a} s$ . Then there exist  $l' \in \text{range}(\phi)$  and  $d' \in \mathcal{D}$  such that  $d \xrightarrow[l']{a} d'$ ,  $\phi(l) = l'$  and  $s = d'[\text{Loc}(d') \setminus \text{range}(\phi) \rightarrow \varepsilon, \phi^{-1}]$ .*

**Proof:** Similar to the one of Lemma 5.4. Omitted.  $\square$

We now have enough technical material to prove Theorem 5.1. For the sake of clarity, we reiterate the statement of the theorem here.

**Theorem 5.1** *For all  $\phi \in \mathcal{F}$ ,  $\tilde{R}_\phi \subseteq \simeq_\phi$ .*

**Proof:** It is sufficient to prove that the family of relations  $\{\tilde{R}_\phi \mid \phi \in \mathcal{F}\}$  at page 15 satisfies the defining clauses of  $\simeq_{\mathcal{F}}$ . Assume that  $\langle d_1, d_2 \rangle \in \tilde{R}_\phi$ . By the definition of  $\tilde{R}_\phi$ , this is because

$$s_1 = d_1[\text{Loc}(d_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon] \approx_i^f d_2[\text{Loc}(d_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, \phi^{-1}] = s_2.$$

We shall check that the defining clauses of  $\simeq_{\mathcal{F}}$  are met by  $\langle d_1, d_2 \rangle$ .

- Assume that  $d_1 \xrightarrow[l']{a} d'_1$ . Then, by Lemma 5.2, we have that either

(a)  $l \notin \text{dom}(\phi)$  and  $s_1 \xrightarrow[l']{a} d'_1[\text{Loc}(d'_1) \setminus (\text{dom}(\phi) \cup \{l\}) \rightarrow \varepsilon] = s'_1$ , or

(b)  $l \in \text{dom}(\phi)$  and  $s_1 \xrightarrow[l.\varepsilon]{a} d'_1[\text{Loc}(d'_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon] = s'_1$ .

We examine the two possibilities in turn.

- (a) Assume that  $l \notin \text{dom}(\phi)$  and  $s_1 \xrightarrow[l']{a} d'_1[\text{Loc}(d'_1) \setminus (\text{dom}(\phi) \cup \{l\}) \rightarrow \varepsilon] = s'_1$ . Then, as  $s_1 \approx_i^f s_2$ , there exists  $s'_2$  such that  $s_2 \xrightarrow[l']{a} s'_2$  and  $s'_1 \approx_i^f s'_2$ . By Lemma 5.4, this implies that there exist  $l' \notin \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l']{a} d'_2$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus (\text{range}(\phi) \cup \{l'\}) \rightarrow \varepsilon, \phi^{-1} \cup \{(l', l)\}].$$

Obviously,  $(l, l')$  is compatible with  $\phi$  and  $\langle d'_1, d'_2 \rangle \in \tilde{R}_{\phi \cup \{(l, l')\}}$ .

- (b) Assume that  $l \in \text{dom}(\phi)$  and  $s_1 \xrightarrow[l.\varepsilon]{a} d'_1[\text{Loc}(d'_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon] = s'_1$ . Then, as  $s_1 \approx_i^f s_2$ , there exists  $s'_2$  such that  $s_2 \xrightarrow[l.\varepsilon]{a} s'_2$  and  $s'_1 \approx_i^f s'_2$ . By Lemma 5.5, this implies that there exist  $l' \in \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l']{a} d'_2$ ,  $\phi(l) = l'$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, \phi^{-1}].$$

The compatibility of  $(l, l')$  with respect to  $\phi$  is obvious. By the definition of  $\tilde{R}_\phi$ , we have that  $\langle d'_1, d'_2 \rangle \in \tilde{R}_\phi$ . Moreover,  $\phi = \phi \cup \{(l, l')\}$ , as  $(l, l') \in \phi$ .

This completes the verification of this clause of the definition of  $\simeq_{\mathcal{F}}$ .

- Assume that  $d_1 \xrightarrow{\epsilon} d_1'$ . Then a matching move from  $d_2$  can be found as in the above case using Lemmas 5.1 and 5.3. The details are omitted.
- We are thus left to check that  $\{\tilde{R}_\phi \mid \phi \in \mathcal{F}\}$  is symmetric. By symmetry, it is sufficient to prove that  $\langle d_2, d_1 \rangle \in \tilde{R}_{\phi^{-1}}$ . As  $\approx_i^f$  is symmetric, we have that  $s_2 \approx_i^f s_1$ . Moreover, we know that  $\approx_i^f$  is preserved by location relabellings as  $\approx_i = \approx_i^f$  and this holds of  $\approx_i$ . Hence we have that  $s_2\phi \approx_i^f s_1\phi$ . It is easy to see that  $s_2\phi = d_2[Loc(d_2) \setminus dom(\phi^{-1}) \rightarrow \epsilon]$  and  $s_1\phi = d_1[Loc(d_1) \setminus range(\phi^{-1}) \rightarrow \epsilon, \phi]$ . This implies that  $\langle d_2, d_1 \rangle \in \tilde{R}_{\phi^{-1}}$ .  $\square$

We shall now aim at showing the second half of the characterization result for  $\approx_i$ , namely that, for all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \simeq n_2$  implies that  $n_1 \approx_i n_2$ . In order to prove this result, we shall construct a location bisimulation  $R$  over  $S$  which contains the pair  $\langle \pi(d_1), \pi(d_2) \rangle$  if  $d_1 \simeq_\emptyset d_2$ . The intuition upon which the construction of  $R$  is based is very similar to that for  $\{\tilde{R}_\phi \mid \phi \in \mathcal{F}\}$ . Intuitively, two states  $s_1$  and  $s_2$  will be related by the relation  $R$  if they can be obtained by suitably renaming the locations occurring in two distributed processes  $d_1$  and  $d_2$  which are equivalent with respect to an association  $\phi$ . The renaming we choose is similar to the one used in the definition of  $\{\tilde{R}_\phi \mid \phi \in \mathcal{F}\}$  at page 15. The main difference is in the relabelling of the locations  $l \in Loc(d_1)$  and  $l' \in Loc(d_2)$  such that  $(l, l') \in \phi$ . These locations play the same role in the behaviour of  $d_1$  and  $d_2$ , modulo  $\phi$ , and we shall make the association between  $l$  and  $l'$  precise in the location operational semantics by relabelling them with the same word over  $Loc^*$ .

Formally, the relation  $R \subseteq S \times S$  is given by:

$$R = \{ \langle s_1, s_2 \rangle \mid \text{there exist } d_1, d_2 \in \mathcal{D}, \phi \in \mathcal{F}, f : dom(\phi) \rightarrow Loc^+ \text{ and } g : range(\phi) \rightarrow Loc^+ \text{ such that} \}$$

- (a)  $d_1 \simeq_\phi d_2$ ,
- (b)  $f = g \circ \phi$ ,
- (c)  $s_1 = d_1[Loc(d_1) \setminus dom(\phi) \rightarrow \epsilon, f]$ ,
- (d)  $s_2 = d_2[Loc(d_2) \setminus range(\phi) \rightarrow \epsilon, g]$ .

We then have the following theorem:

**Theorem 5.2**  *$R$  is contained in  $\approx_i$  over  $S$ .*

Using the above theorem, we can then show the second-half of the characterization result for  $\approx_i$ .

**Corollary 5.2 (Characterization of  $\approx_i$ : Part 2)** *For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \simeq n_2$  implies  $n_1 \approx_i n_2$ .*

**Proof:** Assume that  $n_1 \simeq n_2$ . Then there exist  $d_1, d_2 \in \mathcal{D}$  such that  $d_1 \simeq_\emptyset d_2$  and  $\pi(d_i) = n_i$ ,  $i=1,2$ . Take  $f = g = \emptyset$ . Then, as  $n_i = d_i[Loc(d_i) \rightarrow \epsilon]$ ,  $i=1,2$ , we have that  $\langle n_1, n_2 \rangle \in R$ . By the above theorem, this implies that  $n_1 \approx_i n_2$ .  $\square$

By Corollaries 5.1 and 5.2, we have in fact shown that:

**Theorem 5.3 (Characterization of  $\approx_i$ )**  $\approx_i$  and  $\simeq$  coincide over  $\mathcal{N}$ .

We are thus left to prove Theorem 5.2. The proof of this result also relies on a series of operational results relating the behaviour of distributed processes and states.

**Lemma 5.6** *Let  $d \in \mathcal{D}$ ,  $X \subseteq \text{Loc}$  and  $f : X \rightarrow \text{Loc}^+$ . Assume that  $s = d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon, l]{a} s'$ . Then there exist  $l' \in \text{Loc}(d) \setminus X$  and  $d' \in \mathcal{D}$  such that  $d \xrightarrow[l']{a} d'$  and  $s' = d'[\text{Loc}(d') \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}]$ .*

**Proof:** By induction on the length of the derivation  $s = d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon, l]{a} s'$ .

- Base case,  $s \xrightarrow[\varepsilon, l]{a} s'$ . The proof proceeds by induction on the structure of  $d$ . We only give the two most interesting cases of the proof.

- $d = l' :: p$  for some regular process  $p$ . Assume that

$$s = (l' :: p)[\{l'\} \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon, l]{a} s'.$$

Then, by the location operational semantics, it is easy to see that it must be the case that  $l' \notin X$  and  $s = \varepsilon :: p = p \xrightarrow[\varepsilon, l]{a} s'$ . By the location operational semantics, this implies that  $p \xrightarrow{a} p'$  for some  $p'$  such that  $l :: p' = s'$ . Then  $l' :: p \xrightarrow[l']{a} l' :: p'$  and

$$(l' :: p')[\{l'\} \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}] = l :: p' = s'.$$

- $d = d_1 | d_2$ . Assume that

$$(d_1 | d_2)[\text{Loc}(d_1 | d_2) \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon, l]{a} s'.$$

Without loss of generality, we may assume that this is because  $s_1 = d_1[\text{Loc}(d_1 | d_2) \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon, l]{a} s'_1$  and  $s' = s'_1 | (d_2[\text{Loc}(d_1 | d_2) \setminus X \rightarrow \varepsilon, f])$ . Note that, as  $\text{Loc}(d_1 | d_2) = \text{Loc}(d_1) \cup \text{Loc}(d_2)$  and  $\text{Loc}(d_1) \cap \text{Loc}(d_2) = \emptyset$ , we have that  $s_1 = [d_1[\text{Loc}(d_1) \setminus X \rightarrow \varepsilon, f]]$ . By the inductive hypothesis, there exist  $l' \in \text{Loc}(d_1) \setminus X$  and  $d'_1 \in \mathcal{D}$  such that  $d_1 \xrightarrow[l']{a} d'_1$  and  $s'_1 = d'_1[\text{Loc}(d'_1) \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}]$ . Obviously,  $l' \in \text{Loc}(d_1 | d_2) \setminus X$ . Moreover, by the operational semantics,  $d_1 | d_2 \xrightarrow[l']{a} d'_1 | d_2 = d'$ . Let  $\hat{s} = d_2[\text{Loc}(d') \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}]$ . We can now complete the verification by calculating as follows:

$$\begin{aligned} d'[\text{Loc}(d') \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}] &= \\ (d'_1[\text{Loc}(d'_1) \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}]) | \hat{s} &= \text{as } \text{Loc}(d'_1) \cap \text{Loc}(d_2) = \emptyset \\ s'_1 | (d_2[\text{Loc}(d) \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}]) &= \text{as } \text{Loc}(d) = \text{Loc}(d') \\ s' &= \text{as } l' \notin \text{Loc}(d_2). \end{aligned}$$

The remaining cases are checked in similar fashion.

- Inductive step. Assume that  $s = d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon \cdot l]{\alpha} s'$ . Then there exist  $s_1, s_2 \in S$  such that  $s \xrightarrow{\varepsilon} s_1 \xrightarrow[\varepsilon \cdot l]{\alpha} s_2 \xrightarrow{\varepsilon} s'$ . The claim now follows by the base case and Lemma 5.3(2).  $\square$

We shall now prove a composition result which is a converse of the above lemma.

**Lemma 5.7** *Let  $d \in \mathcal{D}$ ,  $X \subseteq \text{Loc}$  and  $f : X \rightarrow \text{Loc}^+$ . Assume that  $l' \notin X$  and  $d \xrightarrow[l']{\alpha} d'$ . Then  $s = d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[\varepsilon \cdot l']{\alpha} d'[\text{Loc}(d') \setminus (X \cup \{l'\}) \rightarrow \varepsilon, f \cup \{(l', l)\}]$ , for any  $l \in \text{Loc}$ .*

**Proof:** By induction on the length of the derivation  $d \xrightarrow[l']{\alpha} d'$ . The details are straightforward.  $\square$

The following lemma states a decomposition result for moves of the form  $d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[u \cdot l]{\alpha} s'$  with  $u \neq \varepsilon$  and its converse composition result.

**Lemma 5.8** *Let  $d \in \mathcal{D}$ ,  $X \subseteq \text{Loc}$  and  $f : X \rightarrow \text{Loc}^+$ .*

- (1) *Assume that  $s = d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[u \cdot l]{\alpha} s'$  and  $u \neq \varepsilon$ . Then there exist  $l' \in \text{Loc}(d) \cap X$  and  $d' \in \mathcal{D}$  such that*
  - (a)  $f(l') = u$ ,
  - (b)  $d \xrightarrow[l']{\alpha} d'$  and
  - (c)  $s' = d'[\text{Loc}(d') \setminus X \rightarrow \varepsilon, f[l' \rightarrow u \cdot l]]$ .
- (2) *Assume that  $d \xrightarrow[l']{\alpha} d'$ ,  $l' \in \text{Loc}(d) \cap X$  and  $f(l') = u$ . Then  $s = d[\text{Loc}(d) \setminus X \rightarrow \varepsilon, f] \xrightarrow[u \cdot l]{\alpha} d'[\text{Loc}(d') \setminus X \rightarrow \varepsilon, f[l' \rightarrow u \cdot l]]$ , for any  $l \in \text{Loc}$ .*

**Proof:** Both statements are shown by induction on the length of the relevant derivation. The base cases of the inductive arguments are proven by induction on the structure of  $d \in \mathcal{D}$  and follow the lines of the proof of Lemma 5.6.  $\square$

We now have enough technical material to prove Theorem 5.2. For the sake of clarity, we reiterate here the statement of the theorem.

**Theorem 5.2** *The relation  $R$  given at page 19 is contained in  $\approx_l$  over  $S$ .*

**Proof:** First of all, note that  $R$  is symmetric. Assume now that  $\langle s_1, s_2 \rangle \in R$ . Then, by the definition of  $R$ , there exist  $d_1, d_2 \in \mathcal{D}$ ,  $\phi \in \mathcal{F}$ ,  $f : \text{dom}(\phi) \rightarrow \text{Loc}^+$  and  $g : \text{range}(\phi) \rightarrow \text{Loc}^+$  such that

- (a)  $d_1 \simeq_{\phi} d_2$ ,
- (b)  $f = g \circ \phi$ ,
- (c)  $s_1 = d_1[\text{Loc}(d_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon, f]$  and
- (d)  $s_2 = d_2[\text{Loc}(d_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, g]$ .



We shall prove that each move of  $s_1$  can be matched by a move of  $s_2$  modulo  $R$ . The claim will then follow by the symmetry of  $R$ .

- Assume that  $s_1 \xrightarrow[u]{a} s'_1$ . We proceed by distinguishing two possibilities depending on whether  $u = \varepsilon$  or not.
  - (1) Assume that  $u = \varepsilon$ . Then, by Lemma 5.6, there exist  $l_1 \in Loc(d_1) \setminus dom(\phi)$  and  $d'_1 \in \mathcal{D}$  such that  $d_1 \xrightarrow[l_1]{a} d'_1$  and  $s'_1 = d'_1[Loc(d'_1) \setminus (dom(\phi) \cup \{l_1\}) \rightarrow \varepsilon, f \cup \{(l_1, l)\}]$ . As  $d_1 \simeq_\phi d_2$ , there exist  $l_2 \in Loc(d_2)$  and  $d'_2 \in \mathcal{D}$  such that  $(l_1, l_2)$  is compatible with  $\phi$ ,  $d_2 \xrightarrow[l_2]{a} d'_2$  and  $d'_1 \simeq_{\phi \cup \{(l_1, l_2)\}} d'_2$ . As  $l_1 \notin dom(\phi)$ , we have that  $l_2 \notin range(\phi)$ . Hence, by Lemma 5.7,  $s_2 \xrightarrow[\varepsilon \cdot l]{a} s'_2$ , where  $s'_2 = d'_2[Loc(d'_2) \setminus (range(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l)\}]$ . It is now easy to see that  $\langle s'_1, s'_2 \rangle \in R$ .
  - (2) Assume that  $u \neq \varepsilon$ . Then, by Lemma 5.8(1), there exist  $l_1 \in dom(\phi) \cap Loc(d_1)$  and  $d'_1 \in \mathcal{D}$  such that  $f(l_1) = u$ ,  $d_1 \xrightarrow[l_1]{a} d'_1$  and  $s'_1 = d'_1[Loc(d'_1) \setminus dom(\phi) \rightarrow \varepsilon, f[l_1 \rightarrow u \cdot l]]$ . As  $d_1 \simeq_\phi d_2$ , there exists  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l_2]{a} d'_2$ ,  $\phi(l_1) = l_2$  and  $d'_1 \simeq_\phi d'_2$ . (Note that, in this case, we have that  $\phi = \phi \cup \{(l_1, l_2)\}$ .) As  $f = g \circ \phi$ , we have that  $g(l_2) = f(l_1) = u$ . Hence, by Lemma 5.8(2),  $s_2 \xrightarrow[u \cdot l]{a} s'_2$ , where  $s'_2 = d'_2[Loc(d'_2) \setminus range(\phi) \rightarrow \varepsilon, g[l_2 \rightarrow u \cdot l]]$ . Again, it is easy to see that  $\langle s'_1, s'_2 \rangle \in R$ .
- Assume that  $s_1 \xrightarrow{\varepsilon} s'_1$ . Then, by lemma 5.3(2), there exists  $d'_1 \in \mathcal{D}$  such that  $d_1 \xrightarrow{\varepsilon} d'_1$  and  $s'_1 = d'_1[Loc(d_1) \setminus dom(\phi) \rightarrow \varepsilon, f]$ . As  $Loc(d_1) = Loc(d'_1)$ , we have that  $s'_1 = d'_1[Loc(d'_1) \setminus dom(\phi) \rightarrow \varepsilon, f]$ . As  $d_1 \simeq_\phi d_2$ , there exists  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow{\varepsilon} d'_2$  and  $d'_1 \simeq_\phi d'_2$ . By Lemma 5.1(2),  $s_2 \xrightarrow{\varepsilon} s'_2$ , where  $s'_2 = d'_2[Loc(d_2) \setminus range(\phi) \rightarrow \varepsilon, g]$ . Again, as  $Loc(d_2) = Loc(d'_2)$ , we have that  $s'_2 = d'_2[Loc(d'_2) \setminus range(\phi) \rightarrow \varepsilon, g]$ . It is now easy to see that  $\langle s'_1, s'_2 \rangle \in R$ .
 

□

## 6 A Preorder over $\mathcal{D}$

We shall now address the issue of obtaining an effective version of the location preorder  $\lesssim_l$  over the language  $\mathcal{N}$ . Our aim will be to modify the definition of the equivalence  $\simeq$  over  $\mathcal{N}$  in such a way as to obtain a preorder which coincides with  $\lesssim_l$  over nets of automata.

As in § 4, we shall use a notion of preorder over  $\mathcal{D}$  to induce one over  $\mathcal{N}$ . Two distributed processes will be related or not modulo an association  $\phi$  between their locations. In the characterization of  $\approx_l$  given in § 4, these associations took the form of partial bijections from  $Loc$  onto  $Loc$  in order to capture the intuitive idea that two nets are equivalent only if they are weakly bisimilar and can exhibit the same degree of parallelism in every computation. In choosing the notion of association suitable for inducing a “concurrency refinement” preorder over  $\mathcal{D}$ , we shall again consider the intuition underlying the definition of the location preorder. Intuitively, two nets  $n_1$  and  $n_2$  are related by  $\lesssim_l$ ,

$n_1 \lesssim_1 n_2$ , if  $n_1$  and  $n_2$  are weakly bisimilar and  $n_2$  is “at least as parallel” as  $n_1$ . A way of capturing this idea is that one location in  $n_1$  plays the role of (possibly) many locations in  $n_2$ . This informal discussion prompts us to consider associations which are *one-to-many* relations over *Loc*. Formally:

**Definition 6.1**

(1) Let  $\phi \subseteq \text{Loc} \times \text{Loc}$ . We say that  $\phi$  is one-to-many iff for all  $(l_1, l), (l_2, l) \in \phi$ ,  $l_1 = l_2$ .

We shall use  $\mathcal{R}$  to denote the set of one-to-many relations over *Loc*.

(2) Let  $\phi \in \mathcal{R}$  and  $(l_1, l_2) \in \text{Loc} \times \text{Loc}$ . Then  $(l_1, l_2)$  is *p-consistent* with  $\phi$  iff  $l_2 \in \text{range}(\phi)$  implies  $(l_1, l_2) \in \phi$ .

Intuitively,  $(l_1, l_2)$  is *p-consistent* with a one-to-many relation  $\phi$  iff either  $l_2$  is a new location not occurring in the range of  $\phi$  or  $l_2$  is a location which had been previously associated with  $l_1$  (and  $l_1$  only). Note that if  $\phi \in \mathcal{R}$  and  $(l_1, l_2)$  is *p-consistent* with  $\phi$ , then  $\phi \cup \{(l_1, l_2)\} \in \mathcal{R}$ . Moreover, if  $l_2 \in \text{range}(\phi)$  then  $\phi \cup \{(l_1, l_2)\} = \phi$ .

We can now define an  $\mathcal{R}$ -indexed family of relations over  $\mathcal{D}$  which will be used to induce a preorder over  $\mathcal{N}$  via the map  $\pi$ . The definition is very similar to that of  $\simeq_{\mathcal{F}}$ , modulo the replacement of bijective with one-to-many associations and the weaker notion of consistency.

**Definition 6.2**  $\sqsubseteq_{\mathcal{R}} = \{\sqsubseteq_{\phi} \mid \phi \in \mathcal{R}\}$  is the largest  $\mathcal{R}$ -indexed family of relations over  $\mathcal{D}$  satisfying  $d_1 \sqsubseteq_{\phi} d_2$  if for all  $a \in \text{Act}$ ,

- (1)  $d_1 \xrightarrow{a}_l d'_1$  implies that there exist  $l' \in \text{Loc}$  and  $d'_2 \in \mathcal{D}$  such that  $(l, l')$  is *p-consistent* with  $\phi$ ,  $d_2 \xrightarrow{a}_{l'} d'_2$  and  $d'_1 \sqsubseteq_{\phi \cup \{(l, l')\}} d'_2$ ;
- (2)  $d_1 \xrightarrow{e}_D d'_1$  implies that there exists  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow{e}_D d'_2$  and  $d'_1 \sqsubseteq_{\phi} d'_2$ ;
- (3)  $d_2 \xrightarrow{a}_{l'} d'_2$  implies that there exist  $l \in \text{Loc}$  and  $d'_1 \in \mathcal{D}$  such that  $(l, l')$  is *p-consistent* with  $\phi$ ,  $d_1 \xrightarrow{a}_l d'_1$  and  $d'_1 \sqsubseteq_{\phi \cup \{(l, l')\}} d'_2$ ;
- (4)  $d_2 \xrightarrow{e}_D d'_2$  implies that there exists  $d'_1 \in \mathcal{D}$  such that  $d_1 \xrightarrow{e}_D d'_1$  and  $d'_1 \sqsubseteq_{\phi} d'_2$ .

The family of relations  $\sqsubseteq_{\mathcal{R}}$  can be used to induce a preorder over  $\mathcal{N}$  via the map  $\pi$  mimicking the definition of  $\simeq$  given at page 12.

**Definition 6.3** For all  $n_1, n_2 \in \mathcal{N}$ , we write  $n_1 \sqsubseteq n_2$  iff there exist  $d_1, d_2 \in \mathcal{D}$  such that  $\pi(d_i) = n_i$ ,  $i=1,2$ , and  $d_1 \sqsubseteq_{\emptyset} d_2$ .

We shall now prove that  $\sqsubseteq$  is indeed a preorder over  $\mathcal{N}$ . This will follow from some basic properties of the family of relations  $\sqsubseteq_{\mathcal{R}}$  listed in the following lemma.

**Lemma 6.1**

(1) For all  $d_1, d_2 \in \mathcal{D}$ ,  $\phi \in \mathcal{F}$ ,  $d_1 \simeq_{\phi} d_2$  implies  $d_1 \sqsubseteq_{\phi} d_2$ .

---


$$\begin{aligned}
\langle \hat{d}, \vec{d} \rangle \in R_{\{(l, l_1), (l, l_2)\}} & \text{ iff } \langle \hat{d}, \vec{d} \rangle = \langle d_1, d_2 \rangle \\
& \text{ or } \langle \hat{d}, \vec{d} \rangle = \langle l :: b, l_1 :: \text{NIL} | l_2 :: b \rangle \\
& \text{ or } \langle \hat{d}, \vec{d} \rangle = \langle l :: \text{NIL}, l_1 :: \text{NIL} | l_2 :: \text{NIL} \rangle \\
& \text{ or } \langle \hat{d}, \vec{d} \rangle = \langle l :: a, l_1 :: a | l_2 :: \text{NIL} \rangle
\end{aligned}$$

$$R_\phi = \emptyset \quad \text{if } \phi \neq \{(l, l_1), (l, l_2)\}$$


---

Figure 4: Family of Relations for Example 5.1(1)

- (2) For all  $d_1, d_2, d_3 \in \mathcal{D}$  and  $\phi, \varphi \in \mathcal{R}$  such that  $\text{range}(\phi) = \text{dom}(\varphi)$ ,  $d_1 \sqsubseteq_\phi d_2 \sqsubseteq_\varphi d_3$  imply  $d_1 \sqsubseteq_{\varphi \circ \phi} d_3$ .
- (3) For all  $d_1, d_2 \in \mathcal{D}$ ,  $\phi, \varphi \in \mathcal{R}$ ,  $d_1 \sqsubseteq_\phi d_2$  and  $\varphi \subseteq \phi$  imply  $d_1 \sqsubseteq_\varphi d_2$ .

As an immediate corollary of the above lemma, we have that  $\sqsubseteq$  is reflexive, by (1), and transitive, by (2). Thus we shown that:

**Corollary 6.1**  $\sqsubseteq$  is a preorder over  $\mathcal{N}$  which extends  $\simeq$ .

A few examples of processes which are related or not with respect to  $\sqsubseteq$  are now in order. These examples will hopefully give the reader an idea of the role played by the notion of p-consistency in the definition of the family of relations  $\sqsubseteq_{\mathcal{R}}$ .

**Example 6.1**

- (1)  $p = a.b + b.a \sqsubseteq a|b = n$ . In fact, let  $d_1 = l :: p$  and  $d_2 = l_1 :: a|l_2 :: b$ . Then the family of relations  $\{R_\phi \mid \phi \in \mathcal{R}\}$  given in Figure 4 satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ . The claim then follows by Lemma 6.1(3).
- (2)  $n = a|b \not\sqsubseteq a.b + b.a = p$ . In fact, let  $d_1$  and  $d_2$  be defined as in (1). Then, in order for  $d_2 \sqsubseteq_\emptyset d_1$  to hold, it must be the case that

$$l_1 :: \text{NIL} | l_2 :: b \sqsubseteq_{\{(l_1, l)\}} l :: b.$$

However, this is not possible as  $(l_2, l)$  is not p-consistent with  $\{(l_1, l)\}$ . Intuitively, what goes wrong here is that, in order for  $d_2$  to be related to  $d_1$  modulo the empty association, one location

---


$$\begin{aligned}
S_\emptyset &= \{(d_1, d_2)\} \\
S_{\{(l, l_1)\}} &= \{(l :: (b + \alpha)|l' :: \bar{\alpha}.b) \setminus \alpha, (l_1 :: \alpha|l_2 :: \bar{\alpha}.b) \setminus \alpha, \\
&\quad \{(l :: NIL|l' :: b) \setminus \alpha, (l_1 :: NIL|l_2 :: b) \setminus \alpha\} \\
S_{\{(l, l_1), (l, l_2)\}} &= \{(l :: NIL|l' :: \bar{\alpha}.b) \setminus \alpha, (l_1 :: NIL|l_2 :: NIL) \setminus \alpha\} \\
S_{\{(l, l_1), (l', l_2)\}} &= \{(l :: NIL|l' :: NIL) \setminus \alpha, (l_1 :: NIL|l_2 :: NIL) \setminus \alpha\} \\
S_\phi &= \emptyset \text{ otherwise}
\end{aligned}$$


---

Figure 5: Family of Relations for Example 5.1(3)

in  $d_1$  should play the role of two locations in  $d_2$ . This fact does not square with the idea that  $d_1$  should be at least as parallel as  $d_2$ . Formally, this is captured by the notion of  $p$ -consistency.

(3)  $n_1 = (a.(b + \alpha)|\bar{\alpha}.b) \setminus \alpha \sqsubseteq (a.\alpha|\bar{\alpha}.b) \setminus \alpha = n_2$ . In fact, let  $d_1 = (l :: a.(b + \alpha)|l' :: \bar{\alpha}.b) \setminus \alpha$  and  $d_2 = (l_1 :: a.\alpha|l_2 :: \bar{\alpha}.b) \setminus \alpha$ . Then it is easy to check that the family of relations  $\{S_\phi \mid \phi \in \mathcal{R}\}$  given in Figure 5 satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ .

(4)  $p = \text{rec } x. a.x \sqsubseteq (\text{rec } x. a.x) \mid (\text{rec } x. a.x) = n$ . In fact, let  $d_1 = l :: p$  and  $d_2 = l_1 :: (\text{rec } x. a.x) \mid l_2 :: (\text{rec } x. a.x)$ . Then, it is easy to see that the family of relations  $\{S_\phi \mid \phi \in \mathcal{R}\}$  given by

$$\begin{aligned}
S_{\{(l, l_1), (l, l_2)\}} &= \{(d_1, d_2)\} \\
S_\phi &= \emptyset \text{ otherwise}
\end{aligned}$$

satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ . The claim then follows by Lemma 6.1(9).

(5) Let  $p$  and  $n$  be as in (4). Then  $n \not\sqsubseteq p$ . In fact, in order for  $d_2 \sqsubseteq_\emptyset d_1$  to hold, we should have that  $d_2 \sqsubseteq_{\{(l_1, l)\}} d_1$ . However, this is not possible because  $(l_2, l)$  is not  $p$ -consistent with  $\{(l_1, l)\}$ .

We shall now show that also the definition of  $\sqsubseteq$  is independent of the actual association of locations to sub-processes.

**Proposition 6.1** *Let  $n_1, n_2 \in \mathcal{N}$  be such that  $n_1 \sqsubseteq n_2$ . Then, for all  $d_1, d_2 \in \mathcal{D}$ ,  $\pi(d_i) = n_i$ ,  $i=1,2$ , implies  $d_1 \sqsubseteq_\emptyset d_2$ .*

**Proof:** Assume that  $n_1 \sqsubseteq n_2$  and  $\pi(d_i) = n_i$ ,  $i=1,2$ . By the definition of  $\sqsubseteq$ ,  $n_1 \sqsubseteq n_2$  implies that there exist  $d, d' \in \mathcal{D}$  such that  $\pi(d) = n_1$ ,  $\pi(d') = n_2$  and  $d \sqsubseteq_\emptyset d'$ . By Lemma 4.2,  $\pi(d) = \pi(d_1)$  and  $\pi(d_2) = \pi(d')$  imply  $d_1 \simeq_\emptyset d$  and  $d' \simeq_\emptyset d_2$ , respectively. By Lemma 6.1(1) and (3), this implies that

$d_1 \sqsubseteq_{\emptyset} d \sqsubseteq_{\emptyset} d' \sqsubseteq_{\emptyset} d_2$ . By Lemma 6.1(2), we then have that  $d_1 \sqsubseteq_{\emptyset} d_2$ .  $\square$

We end this section with a result on the relationships between the preorder  $\sqsubseteq$  and standard weak bisimulation equivalence over  $\mathcal{N}$ . As expected,  $\sqsubseteq$  is contained in weak bisimulation equivalence. Moreover, the two relations coincide on  $\mathcal{R}\mathcal{P} \times \mathcal{N}$ . The import of this result is that when trying to establish that  $p \sqsubseteq n$ , for some regular process  $p$  and net  $n$ , it is sufficient to give a standard weak bisimulation containing the pair  $\langle p, n \rangle$ .

**Proposition 6.2**

- (1) For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \sqsubseteq n_2$  implies  $n_1 \approx n_2$ .
- (2) For all  $p \in \mathcal{R}\mathcal{P}$ ,  $n \in \mathcal{N}$ ,  $p \approx n$  iff  $p \sqsubseteq n$ .

**Proof:** We prove the two statements separately.

- (1) Consider the relation  $R \subseteq \mathcal{N} \times \mathcal{N}$  given by

$$R = \{ \langle n_1, n_2 \rangle \mid \text{there exist } d_1, d_2 \in \mathcal{D}, \phi \in \mathcal{R} \text{ such that} \\ \begin{aligned} & (1) \ \pi(d_i) = n_i, \ i=1,2, \text{ and} \\ & (2) \ d_1 \sqsubseteq_{\phi} d_2 \}. \end{aligned}$$

We shall show that  $R$  is a weak bisimulation over  $\mathcal{N}$ . First of all, recall that  $\pi(d) = d[Loc \rightarrow \varepsilon]$  because we work modulo equality (6) at page 15. Assume now that  $\langle n_1, n_2 \rangle \in R$  and  $n_1 \xrightarrow{a} n'_1$ . Then, by Lemma 5.3, we have that  $d_1 \xrightarrow{a}_l d'_1$  for some  $l \in Loc(d_1)$  and  $d'_1 \in \mathcal{D}$  such that  $n'_1 = d'_1[Loc \rightarrow \varepsilon]$ . As  $d_1 \sqsubseteq_{\phi} d_2$ , there exist  $l' \in Loc(d_2)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow{a}_{l'} d'_2$  and  $d'_1 \sqsubseteq_{\phi \cup \{(l, l')\}} d'_2$ . By Lemma 5.1,  $n_2 = d_2[Loc \rightarrow \varepsilon] \xrightarrow{a} n'_2$ , where  $n'_2 = d'_2[Loc \rightarrow \varepsilon]$ . It is now easy to see that  $\langle n'_1, n'_2 \rangle \in R$ . The remaining cases of the definition of  $\approx$  can be checked similarly. Hence  $R \subseteq \approx$  and, as  $\sqsubseteq \subseteq R$ , the claim follows.

- (2) In view of statement (1), it is sufficient to prove that  $p \approx n$  implies  $p \sqsubseteq n$ . To this end, consider the family of relations  $S = \{S_{\phi} \mid \phi \in \mathcal{R}\}$  given by:

$$S_{\phi} = \{ \langle l :: p, d \rangle \mid p \in \mathcal{R}\mathcal{P}, d \in \mathcal{D}, p \approx \pi(d) \text{ and } \phi = \{l\} \times Loc(d) \}.$$

We now show that  $S$  satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ . Assume that  $\langle l :: p, d \rangle \in S_{\phi}$ . Then, by the definition of  $S_{\phi}$ , we have that  $p \approx \pi(d)$  and  $\phi = \{l\} \times Loc(d)$ . We just examine one of the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ .

- Assume that  $l :: p \xrightarrow{a}_l \hat{d}$ . Then, by the operational semantics for distributed processes,  $p \xrightarrow{a} p'$  and  $\hat{d} = l :: p'$  for some  $p' \in \mathcal{R}\mathcal{P}$ . As  $p \approx \pi(d)$ , there exists  $n'$  such that  $\pi(d) \xrightarrow{a} n'$  and  $p' \approx n'$ . By Lemma 5.3,  $\pi(d) = d[Loc \rightarrow \varepsilon] \xrightarrow{a} n'$  implies that  $d \xrightarrow{a}_{l'} d'$  for some  $l' \in Loc(d)$  and  $d' \in \mathcal{D}$  such that  $n' = d'[Loc \rightarrow \varepsilon] = \pi(d')$ . Obviously,  $(l, l')$  is  $p$ -consistent with  $\phi$  as  $(l, l') \in \phi$ . Moreover, as  $Loc(d) = Loc(d')$ ,  $\langle l :: p', d' \rangle \in S_{\phi \cup \{(l, l')\}} = S_{\phi}$ .

The remaining clauses of the definition of  $\sqsubseteq_{\mathcal{R}}$  may be checked in similar fashion. Hence  $S$  is contained in  $\sqsubseteq_{\mathcal{R}}$ .

Assume now that  $p \approx n$  and let  $d \in \mathcal{D}$  be such that  $\pi(d) = n$ . Then  $\langle l :: p, d \rangle \in S_{\phi}$ , where  $\phi = \{l\} \times \text{Loc}(d)$ . This implies that  $l :: p \sqsubseteq_{\phi} d$ . By Lemma 6.1(3), we then have that  $l :: p \sqsubseteq_{\emptyset} d$ , i.e.  $p \sqsubseteq n$ .  $\square$

## 7 Relationship with the Location Preorder

We shall now study the relationship between the location preorder  $\lesssim_l$  and  $\sqsubseteq$  over the language  $\mathcal{N}$ . We shall prove in this section that the two preorders coincide over  $\mathcal{N}$ . This result will shed new light on the intuitive significance of the location preorder, especially concerning the role played by the subword relation over locations in its definition, and will provide us with an effective version of the location preorder over nets of automata.

We shall prove, first of all, that  $\sqsubseteq$  is contained in  $\lesssim_l$  over  $\mathcal{N}$ . The proof of this result will follow the lines of that of Theorem 5.2 at page 19.

**Theorem 7.1 (Characterization of  $\lesssim_l$ : Part 1)** *For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \sqsubseteq n_2$  implies  $n_1 \lesssim_l n_2$ .*

**Proof:** Consider the relation  $S \subseteq \mathcal{N} \times \mathcal{N}$  given by

$$S = \{ \langle s_1, s_2 \rangle \mid \text{there exist } d_1, d_2 \in \mathcal{D}, \phi \in \mathcal{R}, f : \text{dom}(\phi) \rightarrow \text{Loc}^+ \text{ and } g : \text{range}(\phi) \rightarrow \text{Loc}^+ \text{ such that} \begin{aligned} (1) & \quad d_1 \sqsubseteq_{\phi} d_2, \\ (2) & \quad f(l) \prec g(l'), \text{ for all } (l, l') \in \phi, \\ (3) & \quad s_1 = d_1[\text{Loc}(d_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon, f] \text{ and} \\ (4) & \quad s_2 = d_2[\text{Loc}(d_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, g] \}. \end{aligned} \}$$

Note the similarity between the definition of  $S$  and that of the relation  $R$  at page 19. Apart from the use of one-to-many associations in place of partial bijections, the only difference is that locations in  $d_1$  and  $d_2$  which are related by  $\phi$  are relabelled by location words which are in the subword relation rather than equal.

It is now sufficient to prove that  $S$  satisfies the defining clauses of  $\lesssim_l$ . Assume, in fact, that this is the case and let  $n_1, n_2 \in \mathcal{N}$  be such that  $n_1 \sqsubseteq n_2$ . Then, by the definition of  $\sqsubseteq$ , there exist  $d_1, d_2 \in \mathcal{D}$  such that  $\pi(d_i) = n_i$ ,  $i=1,2$ , and  $d_1 \sqsubseteq_{\emptyset} d_2$ . Taking  $f = g = \emptyset$ , it is now easy to see that  $\langle n_1, n_2 \rangle \in S$ . As  $S$  satisfies the defining clauses of  $\lesssim_l$ , we then have that  $n_1 \lesssim_l n_2$ .

We are thus left to prove that  $S$  satisfies the definition of  $\lesssim_l$ . To this end, assume that  $\langle s_1, s_2 \rangle \in S$ . Then, by the definition of  $S$ , there exist  $d_1, d_2 \in \mathcal{D}$ ,  $\phi \in \mathcal{R}$ ,  $f : \text{dom}(\phi) \rightarrow \text{Loc}^+$  and  $g : \text{range}(\phi) \rightarrow$

$Loc^+$  such that

- (1)  $d_1 \sqsubset_\phi d_2$ ,
- (2)  $f(l) < g(l')$ , for all  $(l, l') \in \phi$ ,
- (3)  $s_1 = d_1[Loc(d_1) \setminus dom(\phi) \rightarrow \varepsilon, f]$  and
- (4)  $s_2 = d_2[Loc(d_2) \setminus range(\phi) \rightarrow \varepsilon, g]$ .

We shall now show that each observable move of  $s_1$  (respectively  $s_2$ ) can be matched by  $s_2$  (respectively  $s_1$ ) modulo  $S$ . Similar, but simpler arguments can be used to show that also each unobservable move of  $s_1$  can be matched by  $s_2$  and viceversa.

- Assume that  $s_1 \xrightarrow[u \cdot l]{a} s'_1$ . We proceed by distinguishing two cases depending on whether  $u = \varepsilon$  or not.

- (A) Suppose that  $u = \varepsilon$ . Then, by Lemma 5.6,  $s_1 \xrightarrow[\varepsilon \cdot l]{a} s'_1$  implies that there exist  $l_1 \in Loc(d_1) \setminus dom(\phi)$  and  $d'_1 \in \mathcal{D}$  such that  $d_1 \xrightarrow[l_1]{a} d'_1$  and

$$s'_1 = d'_1[Loc(d'_1) \setminus (dom(\phi) \cup \{l_1\}) \rightarrow \varepsilon, f \cup \{(l_1, l)\}].$$

As  $d_1 \sqsubset_\phi d_2$  and  $l_1 \notin dom(\phi)$ , there exist  $l_2 \notin range(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l_2]{a} d'_2$  and  $d'_1 \sqsubset_{\phi \cup \{(l_1, l_2)\}} d'_2$ . By Lemma 5.7,  $d_2 \xrightarrow[l_2]{a} d'_2$  and  $l_2 \notin range(\phi)$  imply that  $s_2 \xrightarrow[\varepsilon \cdot l]{a} s'_2$ , where

$$s'_2 = d'_2[Loc(d'_2) \setminus (range(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l)\}].$$

Obviously  $\varepsilon < \varepsilon$ . Moreover, it is easy to see that  $\langle s'_1, s'_2 \rangle \in S$ .

- (B) Suppose that  $u \neq \varepsilon$ . Then, by Lemma 5.8(1),  $s_1 \xrightarrow[u \cdot l]{a} s'_1$  implies that there exist  $l_1 \in Loc(d_1) \cap dom(\phi)$  and  $d'_1 \in \mathcal{D}$  such that  $f(l_1) = u$ ,  $d_1 \xrightarrow[l_1]{a} d'_1$  and

$$s'_1 = d'_1[Loc(d'_1) \setminus dom(\phi) \rightarrow \varepsilon, f[l_1 \rightarrow u \cdot l]].$$

As  $d_1 \sqsubset_\phi d_2$  and  $l_1 \in dom(\phi)$ , we have that  $d_2 \xrightarrow[l_2]{a} d'_2$  for some  $l_2 \in Loc(d_2)$  and  $d'_2 \in \mathcal{D}$  such that  $d'_1 \sqsubset_{\phi \cup \{(l_1, l_2)\}} d'_2$  and

(B.1) either  $l_2 \notin range(\phi)$

(B.2) or  $(l_1, l_2) \in \phi$ .

We proceed by examining the above possibilities in turn.

- (B.1) Assume that  $l_2 \notin range(\phi)$ . Then, by Lemma 5.7,  $d_2 \xrightarrow[l_2]{a} d'_2$  and  $l_2 \notin range(\phi)$  imply that  $s_2 \xrightarrow[\varepsilon \cdot l]{a} s'_2$ , where

$$s'_2 = d'_2[Loc(d'_2) \setminus (range(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l)\}].$$

Note that  $u < \varepsilon$  as required. We claim that  $\langle s'_1, s'_2 \rangle \in S$ . Consider, in fact,  $d'_1, d'_2 \in \mathcal{D}$ ,  $\phi' = \phi \cup \{(l_1, l_2)\}$ ,  $f' = f[l_1 \rightarrow u \cdot l]$  and  $g' = g \cup \{(l_2, l)\}$ . Note that  $dom(\phi') = dom(\phi)$ , as  $l_1 \in dom(\phi)$ , and  $range(\phi') = range(\phi) \cup \{l_2\}$ , as  $l_2 \notin range(\phi)$ . It is now easy to see that  $\langle s'_1, s'_2 \rangle$  meets the definition of  $S$ .

(B.2) Assume that  $(l_1, l_2) \in \phi$ . In this case, by Lemma 5.8(2),  $d_2 \xrightarrow[l_2]{\alpha} d'_2$  and  $l_2 \in \text{range}(\phi)$  imply that  $s_2 \xrightarrow[v \cdot l_2]{\alpha} s'_2$ , where  $g(l_2) = v$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, g[l_2 \rightarrow v \cdot l_2]].$$

By the definition of  $S$ , we have that  $u < v$ . We are thus left to show that  $\langle s'_1, s'_2 \rangle$  meets the definition of  $S$ . This is easily checked noting that, in this case,  $\phi \cup \{(l_1, l_2)\} = \phi$ .

We have thus shown that each observable move of  $s_1$  can be suitably matched by  $s_2$ .

• Assume that  $s_2 \xrightarrow[u \cdot l_2]{\alpha} s'_2$ . As above, we distinguish two cases depending on whether  $u = \varepsilon$  or not.

(A) Assume that  $u = \varepsilon$ . Then, by Lemma 5.6,  $s_2 \xrightarrow[\varepsilon \cdot l_2]{\alpha} s'_2$  implies that there exist  $l_2 \in \text{Loc}(d_2) \setminus \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l_2]{\alpha} d'_2$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus (\text{range}(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l_2)\}].$$

As  $d_1 \sqsupset_{\phi} d_2$ , there exist  $l_1 \in \text{Loc}$  and  $d'_1 \in \mathcal{D}$  such that  $(l_1, l_2)$  is p-compatible with  $\phi$ ,  $d_1 \xrightarrow[l_1]{\alpha} d'_1$  and  $d_1 \sqsupset_{\phi \cup \{(l_1, l_2)\}} d'_2$ . As  $l_2 \notin \text{range}(\phi)$ , we have that  $(l_1, l_2) \notin \phi$ . Moreover, either  $l_1 \in \text{dom}(\phi)$  or not. We proceed by examining these two possibilities in turn.

(A.1) Assume that  $l_1 \in \text{dom}(\phi)$ . Then, by Lemma 5.8(2),  $d_1 \xrightarrow[l_1]{\alpha} d'_1$ ,  $l_1 \in \text{dom}(\phi)$  and  $f(l_1) = v$  imply that  $s_1 \xrightarrow[v \cdot l_1]{\alpha} s'_1$ , where

$$s'_1 = d'_1[\text{Loc}(d'_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon, f[l_1 \rightarrow v \cdot l_1]].$$

Obviously,  $v < \varepsilon$ . Let  $\phi' = \phi \cup \{(l_1, l_2)\}$ . Note that  $\text{dom}(\phi') = \text{dom}(\phi)$ , as  $l_1 \in \text{dom}(\phi)$ , and  $\text{range}(\phi') = \text{range}(\phi) \cup \{l_2\}$ , as  $l_2 \notin \text{range}(\phi)$ . It is now easy to see that  $\langle s'_1, s'_2 \rangle \in S$ .

(A.2) Assume that  $l_1 \notin \text{dom}(\phi)$ . Then, by Lemma 5.7,  $l_1 \notin \text{dom}(\phi)$  and  $d_1 \xrightarrow[l_1]{\alpha} d'_1$  imply that  $s_1 \xrightarrow[\varepsilon \cdot l_1]{\alpha} s'_1$ , where

$$s'_1 = d'_1[\text{Loc}(d'_1) \setminus (\text{dom}(\phi) \cup \{l_1\}) \rightarrow \varepsilon, f \cup \{(l_1, l_1)\}].$$

Obviously  $\varepsilon < \varepsilon$ . Moreover, it is easy to see that  $\langle s'_1, s'_2 \rangle \in S$ .

This completes the verification for the case  $u = \varepsilon$ .

(B) Assume that  $u \neq \varepsilon$ . In this case, by Lemma 5.8(1),  $s_2 \xrightarrow[u \cdot l_2]{\alpha} s'_2$  implies that there exist  $l_2 \in \text{Loc}(d_2) \cap \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l_2]{\alpha} d'_2$ ,  $g(l_2) = u$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, g[l_2 \rightarrow u \cdot l_2]].$$

As  $d_1 \sqsupset_{\phi} d_2$ , there exist  $l_1 \in \text{Loc}$  and  $d'_1 \in \mathcal{D}$  such that  $(l_1, l_2)$  is p-compatible with  $\phi$ ,  $d_1 \xrightarrow[l_1]{\alpha} d'_1$  and  $d_1 \sqsupset_{\phi \cup \{(l_1, l_2)\}} d'_2$ . As  $l_2 \in \text{range}(\phi)$  and  $(l_1, l_2)$  is p-compatible with  $\phi$ , we



have that  $(l_1, l_2) \in \phi$ . By the definition of  $S$ , this implies that  $v = f(l_1) \prec g(l_2) = u$ . By Lemma 5.8(2),  $d_1 \xrightarrow[\iota_1]{\alpha} d'_1$ ,  $l_1 \in \text{dom}(\phi)$  and  $v = f(l_1)$  imply that  $s_1 \xrightarrow[v \cdot \iota]{\alpha} s'_1$ , where

$$s'_1 = d'_1[\text{Loc}(d'_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon, f[l_1 \rightarrow v \cdot \iota]].$$

We have already seen that  $v \prec u$ . It is then easy to check that  $\langle s'_1, s'_2 \rangle \in S$ .

The verification of the fact that  $S$  meets the remaining clauses of the definition of  $\lesssim_l$  follows similar lines.  $\square$

We are thus left to prove that  $\lesssim_l$  is contained in  $\sqsubseteq$  over the language  $\mathcal{N}$ .

**Theorem 7.2 (Characterization of  $\lesssim_l$ : Part 2)** *For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \lesssim_l n_2$  implies  $n_1 \sqsubseteq n_2$ .*

**Proof:** We shall construct an  $\mathcal{R}$ -indexed family of relations  $\tilde{S} = \{\tilde{S}_\phi \mid \phi \in \mathcal{R}\}$  such that  $\langle d_1, d_2 \rangle \in \tilde{S}_\emptyset$  if  $\pi(d_1) \lesssim_l \pi(d_2)$  and show that it satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ . The key to the definition of  $\tilde{S}$  is in the encoding of a one-to-many relation  $\phi$  in terms of the subword relation over sequences of locations. See clause (2) of the definition of  $\tilde{S}_\phi$  given below.

Let  $\tilde{S} = \{\tilde{S}_\phi \mid \phi \in \mathcal{R}\}$  be given by

$$\begin{aligned} \tilde{S}_\phi = \{ \langle d_1, d_2 \rangle \mid & \text{there exist } f : \text{dom}(\phi) \rightarrow \text{Loc}^+ \text{ and } g : \text{range}(\phi) \rightarrow \text{Loc}^+ \text{ such that} \\ & (1) \ d_1[\text{Loc}(d_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon, f] \lesssim_l d_2[\text{Loc}(d_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, g] \text{ and} \\ & (2) \text{ for all } (l_1, l_2) \in \phi \text{ there exist } n \geq m > 0 \text{ such that } f(l_1) = l_1^n \text{ and } g(l_2) = l_2^m \}. \end{aligned}$$

Note that, for all  $(l_1, l_2) \in \phi$ ,  $f(l_1) \prec g(l_2)$  and that  $g(l_2) = l_2^m$  for some  $m > 0$  implies that  $(l_1, l_2) \in \phi$ . In order to prove the theorem, it is sufficient to show that  $\tilde{S}$  satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ . Assume, in fact, that this is the case. Then let  $n_1 \lesssim_l n_2$  and  $d_1, d_2 \in \mathcal{D}$  be such that  $\pi(d_i) = n_i$ ,  $i=1,2$ . Taking  $f = g = \emptyset$ , we have that  $\langle d_1, d_2 \rangle \in \tilde{S}_\emptyset$ . As  $\tilde{S}$  satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ , it follows that  $n_1 \sqsubseteq n_2$ .

We are thus left to show that the family of relations  $\tilde{S}$  given above satisfies the defining clauses of  $\sqsubseteq_{\mathcal{R}}$ . To this end, assume that  $\langle d_1, d_2 \rangle \in \tilde{S}_\phi$ . By the definition of  $\tilde{S}_\phi$ , this is because there exist  $f : \text{dom}(\phi) \rightarrow \text{Loc}^+$  and  $g : \text{range}(\phi) \rightarrow \text{Loc}^+$  such that

- (1)  $s_1 = d_1[\text{Loc}(d_1) \setminus \text{dom}(\phi) \rightarrow \varepsilon, f] \lesssim_l d_2[\text{Loc}(d_2) \setminus \text{range}(\phi) \rightarrow \varepsilon, g] = s_2$  and
- (2) for all  $(l_1, l_2) \in \phi$  there exist  $n \geq m > 0$  such that  $f(l_1) = l_1^n$  and  $g(l_2) = l_2^m$ .

We shall now check that the defining clauses of  $\sqsubseteq_{\mathcal{R}}$  are met by  $\langle d_1, d_2 \rangle$ .

- Assume that  $d_1 \xrightarrow[\iota_1]{\alpha} d'_1$ . We proceed by distinguishing two cases in the proof depending on whether  $l_1 \in \text{dom}(\phi)$  or not.

(A) Assume that  $l_1 \in \text{dom}(\phi)$ . Then, by Lemma 5.8(2), we have that  $s_1 \xrightarrow[\epsilon, l_1]{a} s'_1$ , where  $f(l_1) = l_1^n$  and

$$s'_1 = d'_1[\text{Loc}(d'_1) \setminus \text{dom}(\phi) \rightarrow \epsilon, f[l_1 \rightarrow l_1^{n+1}]].$$

As  $s_1 \lesssim_l s_2$ , there exist  $v \in \text{Loc}^*$  and  $s'_2 \in \mathcal{S}$  such that  $s_2 \xrightarrow[v, l_1]{a} s'_2$ ,  $l_1^n < v$  and  $s'_1 \lesssim_l s'_2$ . Note now that there are only two possible forms that  $v$  may take. In fact, either  $v = \epsilon$  or  $v = l_1^m$  for some  $n \geq m > 0$ . We proceed by considering these two cases separately.

(A.1) Assume that  $v = \epsilon$ . Then, by Lemma 5.6,  $s_2 \xrightarrow[\epsilon, l_1]{a} s'_2$  implies that there exist  $l_2 \notin \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l_2]{a} d'_2$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus (\text{range}(\phi) \cup \{l_2\}) \rightarrow \epsilon, g \cup \{(l_2, l_1)\}].$$

As  $l_2 \notin \text{range}(\phi)$ , we have that  $(l_1, l_2)$  is p-compatible with  $\phi$ . Moreover, as  $l_1 \in \text{dom}(\phi)$  and  $l_2 \notin \text{range}(\phi)$ , it follows that  $\phi \subset \phi \cup \{(l_1, l_2)\} = \phi'$ ,  $\text{dom}(\phi') = \text{dom}(\phi)$  and  $\text{range}(\phi') = \text{range}(\phi) \cup \{l_2\}$ . It is now easy to check that the maps  $f[l_1 \rightarrow l_1^{n+1}]$  and  $g \cup \{(l_2, l_1)\}$  satisfy clause (2) of the definition of  $\tilde{S}_{\phi'}$ . Hence,  $\langle d'_1, d'_2 \rangle \in \tilde{S}_{\phi'}$ .

(A.2) Assume that  $v = l_1^m$  for some  $n \geq m > 0$ . Then, by Lemma 5.8,  $s_2 \xrightarrow[l_1^m, l_1]{a} s'_2$  implies that there exist  $l_2 \in \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $g(l_2) = l_1^m$ ,  $d_2 \xrightarrow[l_2]{a} d'_2$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus \text{range}(\phi) \rightarrow \epsilon, g[l_2 \rightarrow l_1^{m+1}]].$$

By clause (2) of the definition of  $\tilde{S}_{\phi}$ ,  $f(l_1) = l_1^n$  and  $g(l_2) = l_1^m$  imply  $(l_1, l_2) \in \phi$ . Hence  $(l_1, l_2)$  is p-compatible with  $\phi$  and  $\phi \cup \{(l_1, l_2)\} = \phi$ . It is now immediate to check that  $\langle d'_1, d'_2 \rangle \in \tilde{S}_{\phi}$ .

This completes the verification of case (A).

(B) Assume that  $l_1 \notin \text{dom}(\phi)$ . Then, by Lemma 5.7,  $d_1 \xrightarrow[\epsilon, l_1]{a} d'_1$  implies that  $s_1 \xrightarrow[\epsilon, l_1]{a} s'_1$ , where

$$s'_1 = d'_1[\text{Loc}(d'_1) \setminus (\text{dom}(\phi) \cup \{l_1\}) \rightarrow \epsilon, f \cup \{(l_1, l_1)\}].$$

As  $s_1 \lesssim_l s_2$ , there exists  $s'_2 \in \mathcal{S}$  such that  $s_2 \xrightarrow[\epsilon, l_1]{a} s'_2$  and  $s'_1 \lesssim_l s'_2$ . By Lemma 5.6,  $s_2 \xrightarrow[\epsilon, l_1]{a} s'_2$  implies that there exist  $l_2 \notin \text{range}(\phi)$  and  $d'_2 \in \mathcal{D}$  such that  $d_2 \xrightarrow[l_2]{a} d'_2$  and

$$s'_2 = d'_2[\text{Loc}(d'_2) \setminus (\text{range}(\phi) \cup \{l_2\}) \rightarrow \epsilon, g \cup \{(l_2, l_1)\}].$$

Obviously,  $(l_1, l_2)$  is p-compatible with  $\phi$ . It is now easy to see that  $\langle d'_1, d'_2 \rangle \in \tilde{S}_{\phi \cup \{(l_1, l_2)\}}$ .

- Assume that  $d_2 \xrightarrow[l_2]{a} d'_2$ . As above we proceed by distinguishing two cases depending on whether  $l_2 \in \text{range}(\phi)$  or not. We only give the proof for the case in which  $l_2$  is a location not occurring in  $\text{range}(\phi)$ . The proof for the other case follows the lines of that for (A.2) above.

- Assume that  $d_2 \xrightarrow[l_2]{\alpha} d_2'$  and  $l_2 \notin \text{range}(\phi)$ . Let  $l \notin \text{Loc}(s_1) \cup \text{Loc}(s_2)$ . Then, by Lemma 5.7, we have that  $s_2 \xrightarrow[\varepsilon, l]{\alpha} s_2'$ , where

$$s_2' = d_2'[Loc(d_2') \setminus (\text{range}(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l)\}].$$

As  $s_1 \lesssim_l s_2$ , there exist  $u \in Loc^*$  and  $s_1' \in \mathcal{S}$  such that  $u < \varepsilon$ ,  $s_1 \xrightarrow[u, l]{\alpha} s_1'$  and  $s_1' \lesssim_l s_2'$ . We proceed by distinguishing two sub-cases depending on whether  $u = \varepsilon$  or not.

- (a) Assume that  $u = \varepsilon$ . Then, by Lemma 5.6,  $s_1 \xrightarrow[\varepsilon, l]{\alpha} s_1'$  implies that there exist  $l_1 \notin \text{dom}(\phi)$  and  $d_1' \in \mathcal{D}$  such that  $d_1 \xrightarrow[l_1]{\alpha} d_1'$  and

$$s_1' = d_1'[Loc(d_1') \setminus (\text{dom}(\phi) \cup \{l_1\}) \rightarrow \varepsilon, f \cup \{(l_1, l)\}].$$

As  $\lesssim_l$  is preserved by location relabellings, we have that  $s_1'[l \rightarrow l_1] \lesssim_l s_2'[l \rightarrow l_1]$ . It is easy to see that, as  $l \notin \text{Loc}(s_1) \cup \text{Loc}(s_2)$ ,

$$s_1'[l \rightarrow l_1] = d_1'[Loc(d_1') \setminus (\text{dom}(\phi) \cup \{l_1\}) \rightarrow \varepsilon, f \cup \{(l_1, l_1)\}] \text{ and}$$

$$s_2'[l \rightarrow l_1] = d_2'[Loc(d_2') \setminus (\text{range}(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l_1)\}].$$

Obviously,  $(l_1, l_2)$  is p-compatible with  $\phi$ . It remains to be checked that  $\langle d_1', d_2' \rangle \in \tilde{S}_{\phi \cup \{(l_1, l_2)\}}$ , where  $l_1 \notin \text{dom}(\phi)$  and  $l_2 \notin \text{range}(\phi)$ . This follows immediately from the above observations.

- (b) Assume that  $u \neq \varepsilon$ . Then, by Lemma 5.8(1),  $s_1 \xrightarrow[u, l]{\alpha} s_1'$  implies that there exist  $l_1 \in \text{dom}(\phi)$  and  $d_1' \in \mathcal{D}$  such that  $f(l_1) = u$ ,  $d_1 \xrightarrow[l_1]{\alpha} d_1'$  and

$$s_1' = d_1'[Loc(d_1') \setminus \text{dom}(\phi), f[l_1 \rightarrow u \cdot l]].$$

By the definition of  $\tilde{S}_{\phi}$ ,  $u = l_1^n$  for some  $n > 0$ . As  $\lesssim_l$  is preserved by location relabellings, we have that  $s_1'[l \rightarrow l_1] \lesssim_l s_2'[l \rightarrow l_1]$ . Again, as  $l \notin \text{Loc}(s_1) \cup \text{Loc}(s_2)$ , we have that

$$s_1'[l \rightarrow l_1] = d_1'[Loc(d_1') \setminus \text{dom}(\phi) \rightarrow \varepsilon, f[l_1 \rightarrow l_1^{n+1}]] \text{ and}$$

$$s_2'[l \rightarrow l_1] = d_2'[Loc(d_2') \setminus (\text{range}(\phi) \cup \{l_2\}) \rightarrow \varepsilon, g \cup \{(l_2, l_1)\}].$$

As  $l_2 \notin \text{range}(\phi)$ ,  $(l_1, l_2)$  is p-compatible with  $\phi$ . Note that  $\text{dom}(\phi \cup \{(l_1, l_2)\}) = \text{dom}(\phi)$  and  $\text{range}(\phi \cup \{(l_1, l_2)\}) = \text{range}(\phi) \cup \{l_2\}$ . It is now easy to see that the maps  $f[l_1 \rightarrow l_1^{n+1}]$  and  $g \cup \{(l_2, l_1)\}$  satisfy clause (2) of the definition of  $\tilde{S}_{\phi \cup \{(l_1, l_2)\}}$ . Hence  $\langle d_1', d_2' \rangle \in \tilde{S}_{\phi \cup \{(l_1, l_2)\}}$ .

The clauses having to do with the matching of  $\varepsilon$ -transitions can be checked in similar fashion.

□

We can now put together Theorem 7.1 and Theorem 7.2 to prove the main result of this section, i.e. that  $\sqsubseteq$  and  $\lesssim_l$  coincide over the language  $\mathcal{N}$ .

**Theorem 7.3 (Static Characterization of  $\lesssim_l$ )** For all  $n_1, n_2 \in \mathcal{N}$ ,  $n_1 \sqsubseteq n_2$  iff  $n_1 \lesssim_l n_2$ .

The preorder  $\sqsubseteq$  is thus an effective version of the location preorder over nets of automata. We believe that the coincidence of the two preorders gives an intuitive justification to the location preorder given in [B\*91b]. In fact, it shows that the use of the subword relation in the definition of  $\lesssim_l$  essentially corresponds to requiring that, for nets  $n_1$  and  $n_2$  such that  $n_1 \lesssim_l n_2$ , each location in the possibly more sequential net  $n_1$  plays the role of (possibly) many locations in  $n_2$ . We think that this is a rather intuitive formalization of the idea that  $n_1 \lesssim_l n_2$  if the two nets exhibit the same behaviour and  $n_2$  is at least as parallel as  $n_1$ .

**Acknowledgements:** The idea of a static approach to location equivalence is due to Gérard Boudol, who presented it at an informal CEDISYS meeting held in Brighton, England, in September 1990. I should like to thank Ilaria Castellani, Matthew Hennessy and him for introducing me to the issues addressed in this paper, their support and many inspiring discussions.

## References

- [Ab89] S. Abramsky. Causal Semantics in Process Algebra. Unpublished manuscript, 1989.
- [AB84] D. Austry and G. Boudol. Algebre de Processus et Synchronisations. *Theoretical Computer Science*, 30(1):91-131, 1984.
- [Ace89] L. Aceto. On Relating Concurrency and Nondeterminism. Report 6/89, University of Sussex, October 1989. To appear in the *Proceedings MFPS '91*, Lecture Notes in Computer Science, Springer-Verlag, 1991.
- [B\*91a] G. Boudol, I. Castellani, M. Hennessy and A. Kiehn. Observing Localities. Report 4/91, University of Sussex, March 1991.
- [B\*91b] G. Boudol, I. Castellani, M. Hennessy and A. Kiehn. A Theory of Processes with Localities. In preparation, 1991.
- [BK85] J. A. Bergstra and J. W. Klop. Algebra of Communicating Processes with Abstraction. *Theoretical Computer Science*, 37(1):77-121, 1985.
- [CH89] I. Castellani and M. Hennessy. Distributed Bisimulations. *Journal of the ACM*, 36(4):887-911, 1989.
- [Hoare85] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
- [Kiehn89] A. Kiehn. Distributed Bisimulations for Finite CCS. Report 7/89, University of Sussex, 1989.
- [Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[Thom87] B. Thomsen. *An Extended Bisimulation Induced by a Preorder on Actions*. M.Sc. Thesis in Computer Science, Aalborg University Centre, 1987.

**ISSN 0249 - 6399**