Tau Laws for Pi Calculus

Yuxi Fu * Zhenrong Yang

Department of Computer Science
Shanghai Jiaotong University, Shanghai 200030, China

Abstract

The paper investigates the non-symbolic algebraic semantics of the weak bisimulation congruences on finite pi processes. The weak bisimulation congruences are studied both in the absence and in the presence of the mismatch operator. Some interesting phenomena about the open congruences are revealed. Several new tau laws are discovered and their relationship is discussed. The contributions of the paper are mainly as follows:

1. It is proved that Milner's three tau laws fail to lift a complete system for the strong open congruence to a complete system for the weak open congruence in the absence of both the mismatch operator and the restriction operator. A fourth tau law is proposed to deal with the match operator under the prefix operation. It is shown that for this calculus a complete system for the strong open congruence extended with all the four tau laws is complete for the weak open congruence.
2. It is verified that the four tau laws are also enough for the weak open congruence of the pi calculus without the mismatch operator. Two complete systems are given, one using distinctions and the other using a schematic law for the restriction operator.
3. It is pointed out that the standard definition of the weak open congruence gives rise to a bad equivalence relation in the presence of the mismatch operator. Two alternatives are proposed. These are the late open congruence and the early open congruence. Their difference is similar to that between the weak late congruence and the weak early congruence. Complete axiomatic systems for the two weak open congruences are given.

Key Words: Process Algebra, Mobile Process, Bisimulation, Axiomatization

*The author is supported by the NNSFC (69873032), the 863 Hi-Tech Project (863-306-ZT06-02-2), the Young Scientist Research Fund and the University Scholar Funding Scheme. He is also supported by BASICS, Center of Basic Studies in Computing Science, sponsored by Shanghai Education Commission. BASICS is affiliated to the Department of Computer Science of Shanghai Jiaotong University.

Contents

1 State of Art

For more than ten years various calculi of mobile processes, notably the π-calculus ([32]), have been the focus of research in concurrency theory. These calculi are distinguished from the process calculi like CCS ([21, 31]) in that they are capable of dealing with processes whose communication structures can change during their evolution. This ability of dynamic creation of communication link lies at the heart of mobility. In the calculi of mobile processes mobility is achieved by passing a channel name from one process to another. The process that receives the channel name may communicate to a third process through that received channel name. As it turns out the name-passing communication mechanism is a potent one. This can be seen from the following facts: Firstly name-passing calculi are able to simulate process-passing calculi in a non-distributed framework ([44, 40, 45, 46]). The idea is simply that, instead of sending a process to another, the sender may pass to the other a name through which one can access to the process. Secondly many variants of the π-calculus, the asynchronous π-calculus ([6, 22, 23, 24, 2]), the πI-calculus ([41, 5]), the local π-calculus ([34]) have all been shown to be able to simulate each other to a more or less degree. Even the χ-calculus ([9, 10, 13]) and Fusion Calculus ([39]), which departs from the π-calculus more than any of the above variants, does not seem to add any additional expressive power to the name-passing communication mechanism. One gets the feeling that all name-manipulating mechanisms are more or less equivalent in terms of expressive power.

Apart from the variants of the π-calculus many other process calculi for mobility have been proposed in literature. One recent example is the Calculus of Mobile Ambients ([7]). While it is important to give formal frameworks to capture the rich concept of modern computation, it is equally important, if not more important, to achieve a deep understanding of these frameworks from a mathematical perspective. It is in the latter aspect that sustained efforts are called for. Most people would agree that the theory of π-calculus should be far richer than that of λ-calculus. Yet we know a lot more about the λ-calculus than about the π-calculus.

The name-passing mechanism introduces significant difference between the algebraic theory of calculi of mobile processes and that of CCS ([31]). The difference is mainly caused by three operators:

1. The first is the restriction operator. In CCS this operator is simple. In π-calculus however it comes with complications. Operationally there must be labeled transition rules to take care of name migration. Algebraically distinctions are introduced to define open bisimulations. The restriction operator allows one to abstract away from internal communications. It localizes, so to speak, possible actions so that they do not have global impact. This is a fundamental operator since the theory of process calculus is largely observational.

2. The second is the match operator. This operator is useful at least in two aspects: First it has a very natural motivation from a programming point of view and is necessary in modelling programming phenomena. Second it makes possible an expansion law for π-calculus. One aspect of the match operator is to keep track of the mobility in a nondeterministic fashion. Subsequent research has shown that the match operator plays a crucial role in the axiomatic theory of the π-calculus.

3. The third is the mismatch operator. Compared to the restriction and the match operators, this operator is more debatable. The argument is not to do with its practical motivation for in reality one really needs this operator in a variety of situations where one uses calculi of mobile processes ([38, 39, 1]). It is about if one should introduce such an operator into a basic model for concurrent computation at all. The mismatch operator complicates the algebraic theory because it renders the definitions of observational equivalences more involved. On the other hand it also simplifies the algebraic theory in the sense that it allows to accomplish what otherwise can only be achieved by introducing more gadgets, like distinction, in the meta theory. The main reason to introduce the operator is to make easy the axiomatization of the early and late equivalences. On the other hand, as we shall see later, the open semantics is definitely made more difficult by it.

The match and the mismatch operators help to internalize a lot of meta theory. For instance a condition, which consists of a set of equality and inequality assumptions, and a distinction, which declares a set of names to be pairwise distinct, are two meta theoretic gadgets often used in the investigation of the algebraic theory of mobile processes. Neither a condition nor a distinction is part of the syntax of the π-calculus. They are extralogical entities. In a lot of cases one can do without them by working with the match and the mismatch operators. Now these operators are entities of the π-calculus. It is in this

$\neq$	**early congruence**	**late congruence**	**open congruence**
strong	Boreale, De Nicola	Boreale, De Nicola	Li
weak	Lin	Lin	?

Figure 1: Symbolic Axiomatization with Mismatch

$=$	**early congruence**	**late congruence**	**open congruence**
strong	Lin	Lin	Li
weak	Lin	Lin	Li

Figure 2: Symbolic Axiomatization without Mismatch

way these two operators raise the expressive power of the π-calculus. Our personal opinion is that none of the three operators has been studied in depth. This is sad because, to our understanding, the theory of the π-calculus is largely a theory about these three operators.

The name-passing mechanism renders bisimulation equivalences more interesting. The idea of bisimulation is the most influential one in process algebra. It has been widely used in mathematical logics and game theory. The introduction of bisimulation to concurrency theory is due to the work of Park ([35]) and Milner ([31]). Equality relations based on bisimulation are usually finer than any observational equivalence one finds useful in practice. The virtue of them is that they are subject to algebraic investigation in a more or less standard manner and that it is usually easy to establish the equality of two processes by presenting a bisimulation relation. For π-calculus people have introduced early and late bisimulation equivalences ([32]), barbed equivalences ([33]) and open congruences ([42]), to name a few. It has to be said however that although a lot of attention has been paid to these equivalences our knowledge about them are still preliminary.

The difficulty of bisimulation equivalences is often to do with the weak versions of these relations. A weak bisimulation equivalence ignores internal (tau) actions, which are deemed unobservable. Two processes are equivalent if they can simulate each other's observable actions. A tau law is an equality that manipulates unobservable internal actions. The three well known tau laws people usually refer to were proposed by Milner ([31]). They are the following axioms:

$$\begin{aligned} \alpha.\tau.P &= \alpha.P \\ P+\tau.P &= \tau.P \\ \alpha.(P+\tau.Q) &= \alpha.(P+\tau.Q)+\alpha.Q \end{aligned}$$

where α is a prefix capable of inducing an α-action whereas τ is a prefix that codes up an internal action. The tau laws have been widely used in the theory of process algebra.

There are two main approaches to the semantics of mobile processes:

- One is the symbolic approach ([20, 4]). This approach has been used to study the observational equivalences of π-calculus. In [26] Lin has constructed complete systems for the strong early equivalence, the strong late equivalence, the weak early congruence and the weak late congruence. The symbolic approach applies to π-calculus both with and without the mismatch operator. Prior to Lin's work, Boreale and De Nicola has studied the strong early equivalence and the strong late equivalence in the symbolic framework. In [25] Li has worked out the complete systems for the strong open bisimilarity and the weak open congruence of the π-calculus without the mismatch operator and for the strong open bisimilarity of the π-calculus with the mismatch operator. The situation is summarized in Figure 1 and Figure 2. The picture is almost complete, except for the case of weak open congruence in the presence of the mismatch operator.

- The other is the non-symbolic approach that has been the major choice in semantic investigation. When it comes to axiomatic investigation of π-calculus, the non-symbolic approach presents a less clear picture. Here are the status quos of some related issues:
  - In the first published paper ([32]) on π-calculus Milner, Parrow and Walker defined four bisimulation congruences. They are early equivalence, late equivalence, weak early congruence and

$\neq$	**early congruence**	**late congruence**	**open congruence**
strong	Parrow, Sangiorgi	Parrow, Sangiorgi	?
weak	Parrow	Parrow	?

Figure 3: Non-Symbolic Axiomatization with Mismatch

$=$	**early congruence**	**late congruence**	**open congruence**
strong	?	?	Sangiorgi
weak	?	?	?

Figure 4: Non-Symbolic Axiomatization without Mismatch

weak late congruence. No complete system was given for any of the four congruences. In [37] Parrow and Sangiorgi improved the situation by giving complete systems for both the early equivalence and the late equivalence. The π-calculus used in [37] has the mismatch operator. Only recently Parrow has provided a proof that adding Milner's three tau laws to the complete system of the strong early, respectively late, equivalence is enough to get a complete system for the weak early, respectively late, congruence ([36]). The approach Parrow has used is to translate Lin's symbolic system to a non-symbolic system in a uniform manner.

– We know even less about axiomatization for the π-calculus without the mismatch operator. In this case the definitions of the early equivalence, the late equivalence, the weak early congruence and the weak late congruence remain the same. But no axiomatic system for any of the four has been discovered. The same problem is also open for testing congruence ([8, 3]).

– From a slightly different motivation Sangiorgi proposed open bisimilarities ([42]). His investigation focused on the strong open bisimilarity for the π-calculus without the mismatch operator. A complete system for the congruence was presented using distinction indexed laws. The theory of the weak open congruence was left undiscussed. In particular one does not have any clue about the tau laws for the weak open congruence except the fact that Milner's three tau laws are valid for it.

– The open semantics for the π-calculus with the mismatch operator is virtually unknown. For one thing nobody even knows how to define weak open bisimilarity in this case. If one uses the mismatch operator in the study of the late and the early congruences, s/he has little reason to reject the operator in the open semantics.

We summarize the status quo of non-symbolic axiomatization in Figure 3 and Figure 4.

There are many more open problems, in the theory of π-calculus, than we have listed above. We give below two of them that have been known for several years:

- One fundamental question is about the very definition of bisimulation. If we restrict our attention to the fragment of the π-calculus with none of the choice, match, mismatch and replication operators, we could define a strong bisimulation equivalence for this fragment completely the same as we have defined the strong bisimilarity for CCS processes. Our intuition tells us that this is a congruence relation! But a proof of this conjecture has been so far beyond our reach. A proof of similar result for the asynchronous π-calculus is available though ([2]).

- Another well-known unsettled problem is about the relationship between the weak barbed equivalence and the weak early equivalence. Sangiorgi has proved in his PhD thesis that the two relations coincide for the π-calculus with the mismatch operator *and* the big sigma operator $\sum$, where the big sigma operator refers to the choice operator that can be applied to an arbitrary number of operands. For the π-calculus with only the binary choice operator, the problem is still open. Although most people believe that the coincidence does hold for π-calculus with binary choice, and some even stated the conjecture as a theorem, the fact remains that nobody has come up with a proof! All we know so far is that the coincidence holds for the image-finite processes and for the asynchronous π-processes ([2]). By the way, the relationship of the barbed equivalence to other equivalences in some calculi of mobile processes is much more subtle than most people would think. In χ-calculus for example the barbed congruence is very different from the 'standard bisimulation congruence'. See [12, 14] for details.

In present the biggest problem with the theory of process calculus is that it lacks of proof techniques. We know how to prove a relation to be some bisimulation using inductions on the complexity of the structures of processes or on the height of derivations. But that seems to be all we know. Experiments with mobile processes have told us that a finer-tuned analysis of the operational semantics, as well as the algebraic semantics, of process calculus is called for. Attacking the open problems in process calculi may well help us in this direction. Results and proof methodologies obtained in the study of one process calculus are very likely to be useful to the study of another process calculus.

This paper does not claim to provide any novel techniques for solving problems in process calculus. Its contribution is to solve some of the problems raised in the above, and therefore to provide some insight into the weak observational equivalences and the tau laws for π-calculus. Some of the results are surprising, others are interesting. Our contributions can be summarized as follows:

- We formally prove that Milner's three tau laws are sufficient to lift the complete systems for the strong early equivalence and the strong late equivalence to complete systems for the corresponding weak early congruence and weak late congruence. Our approach is different from Parrow's and shows that the mismatch operator plays a crucial role in the resulting systems. This part of work was carried out independently from Parrow's.

- We show that Milner's three tau laws are definitely not enough for the weak open congruence. We propose a simple and powerful tau law

$$\tau.P = \tau.(P+[x=y]\tau.P)$$

  that reveals more of the dynamic aspect of mobile processes than any of Milner's tau laws. With the help of this fourth tau law, we are able to construct a complete system for the weak open congruence. This part of work should not be understood within the framework of open semantics. The message it conveys has broader implication: For a calculus of *mobile* processes, one needs the fourth tau law in general, although in some cases the law is derivable from other laws.

- We point out that the definition of the weak open bisimilarity for the π-calculus without the mismatch operator can not be applied to the full π-calculus with the mismatch operator. We propose two solutions. They give rise to early open bisimilarity and late open bisimilarity. So even in the framework of open semantics there is an early/late dichotomy. The early open congruence and the late open congruence are completely axiomatized using some new tau laws.

In this paper a lot of attention is paid to the tau laws. Some interesting relationship among the tau laws used in this paper is revealed. Each of the tau laws is used in one of the complete systems studied in this paper. There are altogether five complete systems for respective five weak congruences of the full π-calculus. Three complete systems for strong congruences are also covered. Two of which are presented for completeness. The other, the complete system for the strong open bisimilarity in the presence of the mismatch operator, is the first of such system. The proofs of all the completeness results are structured to bring out a general picture of the completeness proofs.

The main body of the paper is structured into three parts:

1. Section 2 studies the calculus of nondeterministic mobile processes ([42]). It is essentially the π-calculus without the parallel composition, the restriction and the mismatch operators. Some general preliminaries, like the open bisimilarity, equational system, Hennessy Lemma, saturation lemma, promotion property, completeness theorem, are introduced in terms of this calculus. It is pointed out that Milner's three tau laws fall short of characterizing the weak open congruence as far as the calculus of nondeterministic mobile processes is concerned. It is also pointed out that the well-known Hennessy Lemma fails for the calculus of nondeterministic mobile processes. This fact is related to the failure of Milner's tau laws. A fourth tau law is introduced. It is proved that the four tau axioms are enough to support a promotion lemma for the weak open congruence, from which the completeness theorem follows easily.

2. Section 3 takes a look at the π-calculus without the mismatch operator. The impact of the restriction operator on both the definitions of the open bisimilarities and the equational systems is discussed. Two complete systems for the weak open congruence are established: One uses distinction indexed equality laws. This is essentially Sangiorgi's system ([42]) extended with the four tau laws. The other uses a schematic law in favour of the indexed equality laws. In both systems the fourth tau law plays a crucial role.

3. Section 4 focuses on the full π-calculus with the mismatch operator. This part consists of three subparts:

   (a) Section 4.1 introduces equivalence relations for the full π-calculus. A construction central to later definitions of open bisimilarities is defined. The definition of strong open bisimilarity is given by making use of the mismatch operator. It is shown that this equivalence coincides with Sangiorgi's open bisimilarity on the set of π-processes without the mismatch operator. Three weak open bisimilarities for the full π-calculus are introduced. They are the same as Sangiorgi's weak open bisimilarity when restricted to the π-processes without the mismatch operator. For completeness the definitions of strong/weak early and late equivalences are also given.

   (b) Section 4.2 investigates equalities for the full π-calculus and their inter-dependent relationship. These equalities include those for the mismatch operator and various tau laws. Using these equalities, eight axiomatic systems are defined for the eight congruence relations. Some of these systems are included for completeness.

   (c) Section 4.3 establishes the completeness result for each of the eight systems, introduced in the previous section, with respect to the corresponding congruence relation. The proofs of these results are given in a systematic manner.

The rest of the paper complements the main body in two aspects:

1. Section 5 makes a few comments and discusses some related issues.

2. Appendix A collects some proofs omitted in Section 4.

Section 2 is taken from an unpublished paper ([15]) by the first author. The technical results in Section 3 are also due to him. The rest deal with bisimulation equivalences for the full π-calculus with the mismatch operator, which is the joint work of both authors. A related piece of work is reported in two extended abstracts ([18, 19]), also by the present authors, which discusses the χ-calculus with the mismatch operator.

2 The Calculus of Nondeterministic Mobile Processes

The π-calculus of Milner, Parrow and Walker ([32]) is the most widely studied calculus of mobile processes. Communications in π-calculus incur changes in topological structures of the participating processes, causing the configuration of channel connections to evolve dynamically. This kind of mobility is supported by the simple mechanism of confining the contents of communications to the set of channel names.

For the moment the full π-calculus is unnecessarily large. In this section we will be using a calculus of nondeterministic mobile processes ([42]), obtained from π-calculus by omitting the composition operator and the restriction operator.

Let $\mathcal{N}$ be a set of names, ranged over by lower case letters; and let $\overline{\mathcal{N}}$ denote the set of conames $\{\overline{x} \mid x \in \mathcal{N}\}$. The union $\mathcal{N} \cup \overline{\mathcal{N}}$ will be ranged over by α. The processes are defined by BNF as follows:

$$P := \mathbf{0} \mid \pi.P \mid P{+}P \mid [x{=}y]P$$

where $\pi \in \{a(x), \overline{a}x \mid a, x \in \mathcal{N}\} \cup \{\tau\}$. Here $\mathbf{0}$ is the inactive process. A trailing $\mathbf{0}$ in prefix forms is often omitted. The process $a(x).P$ is in input prefix form. It must receive a name at a to instantiate x throughout P before P can be activated. On the other hand $\overline{a}x.P$ is a process in output prefix form. It emits x at a first and then evolves as P. The process $\tau.P$ can become P after performing an internal communication. The process $[x{=}y]P$ behaves like either P or $\mathbf{0}$, depending on whether $x = y$ or not. The constructor $[x{=}y]$ is often referred to as a match operator. The choice operator '+' is well-known.

The process $P+Q$ acts either as P or as Q exclusively, with the choice being made nondeterministically. The name x in $a(x).P$ is bound. A name appearing in P is free if it is not bound. The notations $bn(P)$, $fn(P)$ and $n(P)$ denote respectively the set of bound names, the set of free names and the set of names appeared in P.

A context is a process with a hole. Formally a context $C[]$ is either $[]$, or $\pi.C'[]$ for some context $C'[]$ and some prefix π, or $C'[]+P$ $(P+C'[])$ for some context $C'[]$ and some process P, or $[x{=}y]C'[]$ for some context $C'[]$ and some names x and y.

A substitution σ is a map from $\mathcal{N}$ to $\mathcal{N}$ such that $\{x \mid \sigma(x) \neq x \wedge x \in \mathcal{N}\}$ is finite. The notation $P\sigma$ denotes the process obtained by replacing the free names in P according to σ. A substitution is often written as $[y_1/x_1, \ldots, y_n/x_n]$, indicating that it maps x_i onto y_i, $1 \leq i \leq n$, and is constant elsewhere. If $\sigma = [y_1/x_1, \ldots, y_n/x_n]$ then $n(\sigma) \stackrel{\text{def}}{=} \{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ and $rng(\sigma) \stackrel{\text{def}}{=} \{y_1, \ldots, y_n\}$.

The operational semantics is defined by the following labeled transition system:

Prefix

$$\frac{}{\pi.P \stackrel{\pi}{\longrightarrow} P}$$

Match

$$\frac{P \stackrel{\pi}{\longrightarrow} P'}{[x{=}x]P \stackrel{\pi}{\longrightarrow} P'}$$

Choice

$$\frac{P \stackrel{\pi}{\longrightarrow} P'}{P+Q \stackrel{\pi}{\longrightarrow} P'} \qquad \frac{Q \stackrel{\pi}{\longrightarrow} Q'}{P+Q \stackrel{\pi}{\longrightarrow} Q'}$$

In the above rules, π ranges over the set $\{a(x), \overline{a}x \mid a, x \in \mathcal{N}\} \cup \{\tau\}$. So we have confused the meta notation for action labels and that for prefixes.

Let $\Longrightarrow$ be the reflexive and transitive closure of $\stackrel{\tau}{\longrightarrow}$. We will write $\stackrel{\pi}{\Longrightarrow}$ for $\Longrightarrow\stackrel{\pi}{\longrightarrow}\Longrightarrow$. We will also write $\stackrel{\widehat{\pi}}{\Longrightarrow}$ for $\stackrel{\pi}{\Longrightarrow}$ if $\pi \neq \tau$ and for $\Longrightarrow$ otherwise.

We will need to deal with a sequence of match constructs concatenated one after another. It is convenient to have a meta symbol for such a sequence. In the rest of this paper $\mu, \mu', \ldots$ denote finite lists of match equalities. Suppose μ is $x_1{=}y_1, \ldots, x_n{=}y_n$. Then μP denotes $[x_1{=}y_1] \ldots [x_n{=}y_n]P$. If μ logically implies μ', we write $\mu \Rightarrow \mu'$; and if both $\mu \Rightarrow \mu'$ and $\mu' \Rightarrow \mu$ we write $\mu \Leftrightarrow \mu'$. If μ is an empty list, it plays the role of logical truth, in which case μP is just P. Clearly a list μ of match equalities defines an equivalence relation on the set $\mathcal{N}$ of names. We write σ_μ to denote an arbitrary substitution that sends all members of an equivalence class to a representative of that class.

We will use the widely adopted α-convention saying that a bound name in a process can be replaced by a fresh name, a name that does not appear in the process, without changing the syntax of the process. We will assume throughout this paper that different bound names appeared in a process expression are distinct. When we write $P\sigma \stackrel{a(x)}{\longrightarrow} P'$ for instance, the reader should understand that we have applied the necessary α-conversion so that x appears neither in σ nor in P as a free name. The same assumption will be applied to the names introduced by the restriction operator studied later on.

2.1 Open Congruence

The main focus of the algebraic theory of process calculi is on equivalences of processes. These equivalences are extensional, or observational, in the sense that two processes are deemed to be equal if no difference between them can be detected by other processes. Many observational equivalence relations have been proposed in literature. They differ in the way observations are carried out and the extent observations are made.

In an interleaving framework there are basically two approaches to detect difference between processes. In the static approach two processes are placed in a same environment and the behaviours of the two resulting systems are observed to the extent required by a particular application. Here the environment is assumed static in the sense that it is not affected by external factors. The environment evolves only as the result of its interactions with the subject processes. This approach is very much influenced by the traditional view of equivalence of sequential programmes. The most well known equivalence relation using the static approach is probably the testing equivalence ([8, 3]). The other approach is based on a dynamic viewpoint. In this latter approach environments are subject to the influence of the outside world and therefore can change all the time. This open world view is a reasonable one in the light of modern development of the concept of computation. For mobile computing, distributed computing, Internet computing or global computing, it is safer to regard the environment as dynamic. Now for two processes to be indistinguishable in the dynamic viewpoint, not only that the operational behaviours of the two processes should simulate each other in every environment but also that the descendants of the two processes after the simulation should be able to simulate each other in every possible environment. Equivalence relations enjoying the above conditions are various bisimulation equivalence relations on CCS processes. The idea of Park ([35]) and Milner ([31]) is that two (strongly) bisimilar processes P and Q should meet the zigzag property:

> P and Q are bisimilar if and only if the following two conditions are satisfied:
> (i) If $P \xrightarrow{\lambda} P'$ then $Q \xrightarrow{\lambda} Q'$ for some Q' such that P' and Q' are bisimilar.
> (ii) If $Q \xrightarrow{\lambda} Q'$ then $P \xrightarrow{\lambda} P'$ for some P' such that P' and Q' are bisimilar.

Here λ stands for an arbitrary action. The zigzag argument guarantees that whenever P performs an action and evolves to P' in an environment then Q can perform the same action and become Q' such that P' and Q' are subject to the same zigzag argument. This bisimulation property is strong enough to ensure that bisimilar processes are observational equivalent in a dynamic computational environment. It is clear that the above clauses are the description of a property rather than the prescription of a definition. There are many relations that enjoy the zigzag property. Any one of them gives rise to an equivalence relation. So what one is looking for is the largest such relation, which is often called bisimilarity. The strong bisimilarity of CCS is defined as follows:

> Let $\mathcal{R}$ be a binary relation on the set of CCS processes. $\mathcal{R}$ is said to be a strong bisimulation if $P\mathcal{R}Q$ implies the following properties:
> (i) If $P \xrightarrow{\lambda} P'$ then $Q \xrightarrow{\lambda} Q'$ for some Q' such that $P'\mathcal{R}Q'$.
> (ii) If $Q \xrightarrow{\lambda} Q'$ then $P \xrightarrow{\lambda} P'$ for some P' such that $P'\mathcal{R}Q'$.
> The strong bisimilarity is the largest strong bisimulation.

Studies on π-calculus for example have inherited a lot of techniques and ideas from previous investigations on CCS. There is however many phenomena in π-calculus that call for completely new treatment. The very definition of bisimulation of π-processes for instance poses problems not known in the studies of CCS. For example, if one simply transplants the definition of strong bisimilarity for CCS to π-calculus, one would obtain a relation too weak to be practical.

Definition 1. Let $\mathcal{R}$ be a binary relation on the set of nondeterministic mobile processes. $\mathcal{R}$ is said to be a strong ground bisimulation if $P\mathcal{R}Q$ implies the following properties:
(i) If $P \xrightarrow{\pi} P'$ then $Q \xrightarrow{\pi} Q'$ for some Q' such that $P'\mathcal{R}Q'$.
(ii) If $Q \xrightarrow{\pi} Q'$ then $P \xrightarrow{\pi} P'$ for some P' such that $P'\mathcal{R}Q'$.
The strong ground bisimilarity $\sim_g$ is the largest strong ground bisimulation.

For one thing the strong ground bisimilarity is not closed under the prefix operation. For a binary relation on the set of nondeterministic mobile processes to be closed under input operator, it is necessary that the relation be closed under substitutions of names. One solution is to take the largest bisimulation closed under substitution to be the equivalence relation. This is precisely the approach adopted by Sangiorgi ([42]).

Definition 2. The strong open bisimilarity $\sim_o$ is the largest strong ground bisimulation that is closed under substitution.

In a standard manner we can define weak open bisimilarity.

Definition 3. Let $\mathcal{R}$ be a binary relation on the set of nondeterministic mobile processes. $\mathcal{R}$ is said to be a weak open bisimulation if it is closed under substitution and if whenever $P\mathcal{R}Q$ then the following properties hold:
(i) If $P \xrightarrow{\pi} P'$ then $Q \stackrel{\widehat{\pi}}{\Longrightarrow} Q'$ for some Q' such that $P'\mathcal{R}Q'$.
(ii) If $Q \xrightarrow{\pi} Q'$ then $P \stackrel{\widehat{\pi}}{\Longrightarrow} P'$ for some P' such that $P'\mathcal{R}Q'$.
The weak open bisimilarity $\approx_o$ is the largest weak open bisimulation.

M1	μP	$=$	$\mu' P$	if $\mu \Leftrightarrow \mu'$
M2	$[x{=}y]P$	$=$	$[x{=}y]P[y/x]$	
M3	$[x{=}y](P{+}Q)$	$=$	$[x{=}y]P{+}[x{=}y]Q$	
S1	$P{+}\mathbf{0}$	$=$	P	
S2	$P{+}Q$	$=$	$Q{+}P$	
S3	$P{+}(Q{+}R)$	$=$	$(P{+}Q){+}R$	
S4	$[x{=}y]P{+}P$	$=$	P	

Figure 5: The Core System AS_c

The relation $\approx_o$ is not a congruence relation; it is not closed under the choice operator. From both a programming point of view and an algebraic point of view one is interested in congruence relations. The canonical way to obtain a congruence from a bisimulation equivalence is to take the largest congruence relation contained in the bisimulation equivalence. This approach was adopted by Milner in the study of CCS ([31]) and is now widely used in process algebra. Alternatively one can define the congruence in terms of the bisimulation equivalence. For π-calculus this alternative definition can be given in a uniform manner. In the following definition we assume that $\approx$ is an arbitrary bisimulation equivalence on π-processes.

Definition 4. Two processes P and Q are weakly congruent, notation $P \simeq Q$, if $P \approx Q$ and, for each substitution σ, the following conditions are satisfied:
(i) If $P\sigma \xrightarrow{\tau} P'$ then Q' exists such that $Q\sigma \stackrel{\tau}{\Longrightarrow} Q'$ and $P' \approx Q'$.
(ii) If $Q\sigma \xrightarrow{\tau} Q'$ then P' exists such that $P\sigma \stackrel{\tau}{\Longrightarrow} P'$ and $P' \approx Q'$.

We will write $\simeq_o$ for the open congruence defined from $\approx_o$ in the manner of Definition 4.

2.2 Equational System

For each equivalence, one searches for an inference system consisting of a finite set of equalities and a finite set of rules. A useful inference system should be both sound and complete for the observational equivalence. Soundness means that all derivable equalities are equivalent whereas completeness says that all equivalent processes can be proved equal in the system. A completeness theorem for an equivalence is a landmark in our understanding of the relation.

In [32] Milner, Parrow and Walker propose four bisimulation congruence relations, which are respectively strong early equivalence, strong late equivalence, weak early congruence and weak late congruence. To study the axiomatization of the π-calculus, they introduce the match operator and formulate the expansion law in terms of this new operator. The importance of the match operator is that it enables one to code up concurrency of mobile processes in terms of nondeterminism in an interleaving framework, using the well-known expansion law. In the same paper Milner, Parrow and Walker have initiated the study of complete systems for mobile processes. Technically speaking the systems they propose are for the strong late bisimilarity and the strong early bisimilarity. These two relations are not congruence relations. In the systems for the two bisimilarities there is a noteworthy inference rule:

If $P[y/x] = Q[y/x]$ for every name y free in either P or Q then $a(x).P = a(x).Q$.

For systems of congruence relations the above inference rule can be simplified to the following rule:

If $P = Q$ then $a(x).P = a(x).Q$.

In this paper all axiomatic systems are for congruence relations.

A complete system for a weak observational equivalence consists of two parts. One is the subsystem complete for the corresponding strong observational equivalence. The other contains the so-called tau laws. In this section we give an equational system that is the core of all the systems studied in this paper. This system is given in Figure 5, which was first used by Sangiorgi in [42]. It is actually the complete system for the strong open bisimilarity on the finite nondeterministic mobile processes.

Suppose $R_1, \ldots, R_n$ are equational laws. Provability in system $AS_c \cup \{R_1, \ldots, R_n\}$ is defined as follows: $AS_c \cup \{R_1, \ldots, R_n\} \vdash P = Q$ if one of the followings holds:

- $P \equiv Q$, meaning that P and Q are the same syntactical object;

MD1	$[x{=}y]\mathbf{0}$	$=$	$\mathbf{0}$
MD2	$[x{=}x]P$	$=$	P
MD3	μP	$=$	$\mu(P\sigma_\mu)$
SD1	$P{+}P$	$=$	P
SD2	$\mu P{+}P$	$=$	P

Figure 6: Derived Laws of AS_c

- Some context $C[]$ and processes A, B exist such that

$$\begin{array}{rcl} P & \equiv & C[A] \\ C[B] & \equiv & Q \end{array}$$

  and either $A = B$ or $B = A$ is an instance of one of the axioms in $AS_c \cup \{R_1, \ldots, R_n\}$;

- Some contexts $C_1[], \ldots, C_n[]$ and processes $A_1, \ldots, A_n, B_1, \ldots, B_n$, for $n \geq 2$, exist such that

$$\begin{array}{rcl} P & \equiv & C_1[A_1] \\ C_1[B_1] & \equiv & C_2[A_2] \\ & \vdots & \\ C_{n-1}[B_{n-1}] & \equiv & C_n[A_n] \\ C_n[B_n] & \equiv & Q \end{array}$$

  and, for all $i \in \{1, \ldots, n\}$, either $A_i = B_i$ or $B_i = A_i$ is an instance of one of the axioms in $AS_c \cup \{R_1, \ldots, R_n\}$.

A more popular way to define the provability in AS_c is to introduce equivalence and congruence rules. The equivalence rules say that the provability relation is reflexive, symmetric and transitive. The congruence rules declare that the provability relation is closed under all the operators of the calculus. For instance from $P = Q$ one can derive that $\pi.P = \pi.Q$, $P{+}R = Q{+}R$ and $[x{=}y]P = [x{=}y]Q$. Write $AS_c \vdash' P = Q$ to mean that the equality $P = Q$ is derivable from the axioms of AS_c together with the equivalence and congruence rules. The following lemma can be proved by simple induction.

Lemma 5. *$AS_c \vdash' P = Q$ if and only if $AS_c \vdash P = Q$.*

From now on we will confuse the two notations and use only $\vdash$.

When $AS_c \cup \{R_1, \ldots, R_n\} \vdash P = Q$ we say that $P = Q$ is provable in $AS_c \cup \{R_1, \ldots, R_n\}$ or that $P = Q$ is a derived law of $AS_c \cup \{R_1, \ldots, R_n\}$. We will write $P \overset{R_1,\ldots,R_n}{=} Q$ to indicate that $R_1, \ldots, R_n$ are the major laws and/or rules in the derivation of $P = Q$. Some derived laws of AS_c are given in Figure 6, which have all been used by Sangiorgi in [42].

For a finite collection $\{P_i\}_{i\in\{1,\ldots,n\}}$ of processes, we write

$$\sum_{i\in\{1,\ldots,n\}} P_i$$

for the process

$$(\ldots((P_1{+}P_2){+}P_3){+}\ldots){+}P_n$$

In view of S2 and S3 the order of the occurrence of P_i in the above expression does not matter. As a matter of fact we will often write

$$P_1{+}P_2{+}P_3{+}\ldots{+}P_n$$

without any parenthesis since it does not cause any confusion in the algebraic investigation.

A process P is in normal form if it is of the shape

$$\sum_{i\in I_1} \mu_i a_i(x).P_i + \sum_{i\in I_2} \mu_i \overline{a_i} x_i.P_i + \sum_{i\in I_3} \mu_i \tau.P_i$$

T1	$\pi.\tau.P$	$=$	$\pi.P$
T2	$P+\tau.P$	$=$	$\tau.P$
T3	$\pi.(P+\tau.Q)$	$=$	$\pi.(P+\tau.Q)+\pi.Q$

Figure 7: Milner's Tau Laws

in which x does not appear free in P and I_1, I_2, I_3 are pairwise distinct. The depth of a normal form process, notation $d(P)$, is defined as follows: (i) $d(\mathbf{0}) \stackrel{\text{def}}{=} 0$; (ii) $d(\pi.P) \stackrel{\text{def}}{=} 1 + d(P)$; (iii) $d(P{+}Q) \stackrel{\text{def}}{=} max\{d(P), d(Q)\}$; (iv) $d([x{=}y]P) \stackrel{\text{def}}{=} d(P)$.

In AS_c a process can be converted to a normal form process without increasing its depth. Its proof is standard.

Lemma 6. *A process P is provably equal to a normal form process P' in the system AS_c such that $d(P') \leq d(P)$.*

The proof of the following completeness theorem can be found in [42].

Theorem 7. *AS_c is sound and complete for $\sim_o$.*

2.3 The Insufficiency of Milner's Tau Laws for Open Semantics

In view of the fact that in CCS Milner's three tau laws are capable of promoting a complete system for the strong congruence to a complete system for the weak congruence, one is tempted to think that the same is true for the calculus of nondeterministic mobile processes. Figure 7 gives the tau laws in π-calculus. The soundness is obvious.

Theorem 8. $\mathrm{AS}_c \cup \{T1, T2, T3\}$ *is sound for $\simeq_o$.*

The popular view would have us believe that the corresponding completeness theorem also holds. In the rest of this section we prove that this is not the case.

A process P is a t-process if it contains neither input prefix nor output prefix. For instance $[x{=}y](\tau + [a{=}b]\tau.\tau)$ is a t-process, but $[x{=}y]a(x)$ is not. In the following proof, we need to work with t-processes with holes. For that purpose we introduce the notion of t-context defined as follows: (i) $[]$ is a t-context; (ii) if $C[]$ is a t-context then $C[] + P$, $P + C[]$, $\tau.C[]$ and $[x{=}y]C[]$ are t-contexts, where P is a t-process. In other words t-contexts are contexts with neither input prefix nor output prefix. With the help of t-contexts, it is easy to indicate an occurrence of τ in a t-process. Suppose P is a t-process and $P \equiv C[\tau.Q]$ for some t-context $C[]$ and some t-process Q. Then the explicit τ in $C[\tau.Q]$ is an occurrence of τ in P. In what follows we will simply say that $P \equiv C[\tau.Q]$ is a tau occurrence in P. A terminating tau occurrence is a tau occurrence $P \equiv C[\tau.Q]$ such that Q contains no occurrence of τ.

Definition 9. The match guard of a t-context $C[]$, written $M(C[])$, is a list of equalities defined as follows: (i) $M([])$ is the empty list; (ii) $M(C[]) \stackrel{\text{def}}{=} M(C'[])$ if $C[]$ is $C'[]{+}P$, or $P{+}C'[]$, or $\tau.C'[]$; (iii) $M([x{=}y]C[]) \stackrel{\text{def}}{=} x{=}y, M(C[])$.

We will say that the match guard of a (terminating) tau occurrence $P \equiv C[\tau.Q]$ in P is the match list $M(C[])$.

Definition 10. A process P is in t4-form if it is a t-process and for some distinct x and y the following properties hold:

1. There exists some tau occurrence $P \equiv C[\tau.Q]$.
2. For every terminating tau occurrence $P \equiv C'[\tau.Q']$ it holds that $M(C'[]) \Rightarrow x{=}y$.

If a process has a tau occurrence it has a terminating tau occurrence. It follows that a process in t4-form contains at least one match operator.

Lemma 11. *Suppose P is a process in t4-form and $AS_c \cup \{T1, T2, T3\} \vdash P = Q$. Then Q is also in t4-form.*

Proof. Suppose P is a process in t4-form. Suppose further that $AS_c \cup \{T1, T2, T3\} \vdash P = Q$ is $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ for some t-context $C[]$, obtained by applying one of the axioms.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is $AS_c \cup \{T1, T2, T3\} \vdash C[\mu R] = C[\mu' R]$, obtained by applying M1. Suppose that for a particular terminating tau occurrence in R its match guard in $C[\mu R]$ and $C[\mu' R]$ are μ_1 and μ_1' respectively. Then $\mu_1 \Leftrightarrow \mu_1'$ because $\mu \Leftrightarrow \mu'$. It follows from $\mu_1 \Rightarrow x{=}y$ that $\mu_1' \Rightarrow x{=}y$. If R has no occurrence of τ then $C[\mu R]$ has a terminating tau occurrence in $C[]$. So $C[\mu R]$ has a terminating tau occurrence. Thus $C[\mu' R]$ satisfies the two conditions of Definition 10 and therefore is in t4-form.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is $AS_c \cup \{T1, T2, T3\} \vdash C[[a{=}b]R] = C[[a{=}b]R[b/a]]$, obtained by applying M2. If $a = b$ then $C[[a{=}b]R[b/a]] \equiv C[[a{=}b]R]$. Otherwise let $\mu, a = b, a_1 = b_1, \ldots, a_n = b_n$ be the match guard in $C[[a{=}b]R]$ of a terminating tau occurrence in R. Then the match guard μ' of that particular terminating tau occurrence in $C[[a{=}b]R[b/a]]$ is the list $\mu, a = b, a_1[b/a] = b_1[b/a], \ldots, a_n[b/a] = b_n[b/a]$. It should be clear that

$$\begin{aligned} \mu' &\Leftrightarrow \mu, a = b, a_1[b/a] = b_1[b/a], \ldots, a_n[b/a] = b_n[b/a] \\ &\Leftrightarrow \mu, a = b, a_1 = b_1, \ldots, a_n = b_n \\ &\Rightarrow x = y. \end{aligned}$$

  Therefore $C[[a{=}b]R[b/a]]$ is in t4-form.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is $AS_c \cup \{T1, T2, T3\} \vdash C[[a{=}b](R + S)] = C[[a{=}b]R + [a{=}b]S]$, obtained by applying M3. Then the match guard in $C[[a{=}b](R{+}S)]$ of a terminating tau occurrence in R is the same as the match guard in $C[[a{=}b]R + [a{=}b]S]$ of the terminating tau occurrence in R. It follows that $C[[a{=}b](R + S)]$ is in t4-form if and only if $C[[a{=}b]R + [a{=}b]S]$ is in t4-form.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is obtained by applying S1, or S2, or S3. In these cases $C[B]$ is obviously in t4-form.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is $AS_c \cup \{T1, T2, T3\} \vdash C[[x{=}y]R + R] = C[R]$, obtained by applying S4. If R contains a tau occurrence, then all terminating tau occurrences in $[x{=}y]R + R$ are also terminating tau occurrences in $C[[x{=}y]R + R]$. As $C[[x{=}y]R + R]$ is in t4-form, all these terminating tau occurrences satisfy the second condition in Definition 10. It follows that $C[R]$ also satisfies the second condition in Definition 10. Thus $C[R]$ is in t4-form.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is obtained by applying T1. If P has no occurrence of tau, then both $\tau.\tau.P$ and $\tau.P$ have only one terminating occurrence. Otherwise a tau occurrence in P is terminating in $\tau.\tau.P$ if and only if it is terminating in $\tau.P$. This is enough to show that $C[B]$ is in t4-form.

- $AS_c \cup \{T1, T2, T3\} \vdash C[A] = C[B]$ is obtained by applying T2 or T3. The situation is similar to the previous one.

This completes the proof. □

Theorem 12. *$\tau.[x{=}y]\tau = \tau$ is not provable in $AS_c \cup \{T1, T2, T3\}$.*

Proof. It is clear that $\tau.[x{=}y]\tau$ is in t4-form. If $AS_c \cup \{T1, T2, T3\} \vdash \tau.[x{=}y]\tau = Q$ then, by Lemma 11, Q is in t4-form. So Q contains at least one match operator, which means that it can not be τ. □

2.4 The Failure of Hennessy Lemma for Mobile Processes

In the proof of completeness theorem for the weak congruence in CCS, the following result, which is attributed to Hennessy in [31], plays a crucial role:

Lemma 13 (Hennessy Lemma). *If $P \approx Q$ then either $\tau.P \simeq Q$ or $P \simeq Q$ or $P \simeq \tau.Q$.*

Here $\approx$ is the weak bisimilarity and $\simeq$ is the largest congruence relation contained in $\approx$. In the proof of the completeness theorem by induction, Hennessy Lemma helps to lift $P \approx Q$ to either $\tau.P \simeq Q$ or $P \simeq Q$ or $P \simeq \tau.Q$, thus allowing the induction hypothesis to apply.

T4	$\tau.P$	$=$	$\tau.(P+[x{=}y]\tau.P)$

Figure 8: The Fourth Tau Law

In π-calculus however Hennessy Lemma does not hold in general! For a counter example, consider the following three propositions

$$\tau.[x{=}y]\tau =_o \mathbf{0} \tag{1}$$
$$[x{=}y]\tau =_o \mathbf{0} \tag{2}$$
$$[x{=}y]\tau =_o \tau \tag{3}$$

None of them holds although $[x{=}y]\tau \approx_o \mathbf{0}$ is true. In (1) a tau action from $\tau.[x{=}y]\tau$ can not be matched up by any tau action from $\mathbf{0}$. In (2) a tau action from $([x{=}y]\tau)[x/y]$ can not be matched up by any tau action from $\mathbf{0}[x/y]$. And in (3) a tau action from τ can not be matched up by any tau action from $[x{=}y]\tau$.

In the calculus of mobile processes, local version of Hennessy Lemma does hold. Let's explain what we mean by that using the above example. Assuming $x = y$ then (3) is valid. If $x \neq y$ then (2) is true. In the symbolic approach ([20, 4]), complete systems for weak congruence relations can be achieved by exploiting the local Hennessy Lemma.

A provability judgement in the symbolic approach is of the form $C \vdash_{AS} P = Q$, where AS is an axiomatic system and C is a set of equalities and/or inequalities on names. Among the non-symbolic approach, Sangiorgi's open semantics is the simplest ([42]). Provability in Sangiorgi's system is of the form $\vdash_{AS} P = Q$ for at least the part of π-calculus without the restriction operator. Axiomatization of open congruences has been discussed in the strong case, but not in the weak case. The failure of Hennessy Lemma is another way to say that the three tau laws fail to lift a complete system for the strong open bisimilarity to a complete system for the weak open congruence as the resulting system is not capable of proving the equality $\tau.[x{=}y]\tau = \tau$.

2.5 The Fourth Tau Law

We need to extend $AS_c \cup \{T1, T2, T3\}$ to achieve a complete system for the weak open congruence. We will show that it is sufficient to add a single law, the fourth tau law. The new axiom, T4 in Figure 8, involves a match operator. It is the simplest among all the alternatives we have come up with. Some of the consequences of T4 are discussed below.

Lemma 14. *$AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.(P + \mu\tau.P)$.*

Proof. When μ is empty, the result follows from T1 and T2. So the base case holds. Now observe that

$$\begin{aligned}
\tau.P &\overset{T4}{=} \tau.(P+[x{=}y]\tau.P)\\
&\overset{SD2}{=} \tau.(P+\mu[x{=}y]\tau.P+[x{=}y]\tau.P)\\
&\overset{I.H.}{=} \tau.(P+\mu[x{=}y]\tau.P+[x{=}y]\tau.(P+\mu\tau.P))\\
&\overset{M2}{=} \tau.(P+\mu[x{=}y]\tau.P+[x{=}y]\tau.(P+\mu\tau.P)[y/x])\\
&\overset{M1}{=} \tau.(P+\mu[x{=}y]\tau.P+[x{=}y]\tau.(P+\mu[x{=}y]\tau.P)[y/x])\\
&\overset{M2}{=} \tau.(P+\mu[x{=}y]\tau.P+[x{=}y]\tau.(P+\mu[x{=}y]\tau.P))\\
&\overset{T4}{=} \tau.(P+\mu[x{=}y]\tau.P)
\end{aligned}$$

So the proof can be completed by induction on the number of match operators in μ. □

Lemma 15. *$AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.(P + \sum_{i\in I} \mu_i\tau.P)$ for a finite nonempty indexing set I.*

Proof. Lemma 14 provides the base step, and the following derivation provides the induction step:

$$\begin{aligned}
\tau.(P + \sum_{i\in I} \mu_i\tau.P + \mu\tau.P) &\stackrel{I.H.}{=} \tau.(P + \sum_{i\in I} \mu_i\tau.P + \mu\tau.(P + \sum_{i\in I} \mu_i\tau.P)) \\
&= \tau.(P + \sum_{i\in I} \mu_i\tau.P) \\
&\stackrel{I.H.}{=} \tau.P
\end{aligned}$$

where the second equality holds by Lemma 14. □

2.6 Saturation

The proof of completeness is always much harder than that of soundness. In the standard proof of completeness theorem for the weak congruence on finite CCS processes ([21, 31]), one verifies first that every normal form process is provably equivalent to a saturated normal form process using the three tau laws. Recall that a process P is saturated if, for every α, $P \stackrel{\alpha}{\longrightarrow} P'$ whenever $P \stackrel{\alpha}{\Longrightarrow} P'$. It follows that P is in saturated normal form if and only if whenever $P \stackrel{\alpha}{\Longrightarrow} P'$ then $\alpha.P'$ is a summand of P. Now if P and Q are weakly congruent saturated normal form processes and $P \stackrel{\alpha}{\longrightarrow} P'$ then $Q \stackrel{\alpha}{\Longrightarrow} Q'$ for some Q' such that $Q' \approx P'$, where $\approx$ denotes the weak equivalence. By saturation, $Q \stackrel{\alpha}{\longrightarrow} Q'$ and therefore $\alpha.Q'$ is a summand of Q. If, and this is a nontrivial if, we can deduce by induction hypothesis that $\alpha.P'$ is provably equal to $\alpha.Q'$, which is much weaker than saying that P' is provably equal to Q', then we can conclude that every summand of P is provably equal to a summand of Q, and vice versa. This gives us the required completeness.

If one is only interested in completeness proof, then the notion of saturated process is not needed. What is really necessary is the following saturation property:

If $P \stackrel{\alpha}{\Longrightarrow} P'$ and P is in normal form then P and $P + \alpha.P'$ are provably equal.

From the point of view of axiomatization, the role of saturation property is to relate operational semantics to equational rewriting.

For π-calculus the situation is a little bit more complex.

Lemma 16 (saturation). *Suppose Q is in normal form. The following saturation properties hold:*
(i) If $Q\sigma_\mu \stackrel{\overline{a}x}{\Longrightarrow} Q'$ then $AS_c \cup \{T1, T2, T3\} \vdash Q = Q + \mu\overline{a}x.Q'$.
(ii) If $Q\sigma_\mu \stackrel{a(x)}{\Longrightarrow} Q'$ then $AS_c \cup \{T1, T2, T3\} \vdash Q = Q + \mu a(x).Q'$.
(iii) If $Q\sigma_\mu \stackrel{\tau}{\Longrightarrow} Q'$ then $AS_c \cup \{T1, T2, T3\} \vdash Q = Q + \mu\tau.Q'$.

Proof. A standard exercise using Milner's tau laws. □

In the proof of the saturation lemma, the role of Milner's tau laws is to systematically remove tau prefixes which only induce operationally unobservable actions.

2.7 Promotion

A careful examination of the role of Hennessy Lemma in the completeness proof for CCS shows that what it really comes down to is the following property:

If $P \approx Q$ then either $\tau.P = Q$, or $P = Q$, or $P = \tau.Q$ is provable.

So Hennessy Lemma helps to transfer a semantic statement to a proof theoretical one. As a matter of fact, as far as completeness is concerned, the following weaker property is all one needs:

If $P \approx Q$ then $\tau.P = \tau.Q$ is provable.

We will call it promotion property. Although the Hennessy Lemma does not hold in the calculus of nondeterministic mobile processes, the promotion property does hold.

Lemma 17 (promotion). *If $P \approx_o Q$ then $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.Q$.*

Proof. Suppose $P \approx_o Q$. The proof is carried out by induction on the sum of the depths of P and Q. By Lemma 6 we may concentrate on normal form processes. Suppose P is of the form

$$\sum_{i\in I_1} \mu_i a_i(x).P_i + \sum_{i\in I_2} \mu_i \overline{a_i} x_i.P_i + \sum_{i\in I_3} \mu_i \tau.P_i$$

and Q is of the form

$$\sum_{j\in J_1} \mu_j a_j(x).Q_j + \sum_{j\in J_2} \mu_j \overline{a_j} x_j.Q_j + \sum_{j\in J_3} \mu_j \tau.Q_j$$

If $\mu_i a_i(x).P_i$ is a summand of P then $(\mu_i a_i(x).P_i)\sigma_{\mu_i} \xrightarrow{a_i\sigma_{\mu_i}(x)} P_i\sigma_{\mu_i}$ must be matched up by $Q\sigma_{\mu_i} \stackrel{a_i\sigma_{\mu_i}(x)}{\Longrightarrow} Q'$ such that $P_i\sigma_{\mu_i} \approx_o Q'$. Both $P_i\sigma_{\mu_i}$ and Q' are in normal form and $d(P_i\sigma_{\mu_i}) + d(Q') < d(P) + d(Q)$. So by induction hypothesis $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P_i\sigma_{\mu_i} = \tau.Q'$. It follows that

$$\begin{aligned} AS_c \cup \{T1, T2, T3, T4\} \vdash \mu_i a_i(x).P_i &\stackrel{MD3}{=} \mu_i a_i\sigma_{\mu_i}(x).P_i\sigma_{\mu_i} \\ &\stackrel{T1}{=} \mu_i a_i\sigma_{\mu_i}(x).\tau.P_i\sigma_{\mu_i} \\ &\stackrel{I.H.}{=} \mu_i a_i\sigma_{\mu_i}(x).\tau.Q' \\ &\stackrel{T1}{=} \mu_i a_i\sigma_{\mu_i}(x).Q' \end{aligned}$$

Therefore $AS_c \cup \{T1, T2, T3, T4\} \vdash \mu_i a_i(x).P_i + Q = \mu_i a_i\sigma_{\mu_i}(x).Q' + Q = Q$ by Lemma 16. Similarly one proves that $AS_c \cup \{T1, T2, T3, T4\} \vdash \mu_i \overline{a_i} x_i.P_i + Q = Q$ whenever $\mu_i \overline{a_i} x_i.P_i$ is a summand of P.

Now suppose $\mu_i \tau.P_i$ is a summand of P. Then $(\mu_i \tau.P_i)\sigma_{\mu_i} \xrightarrow{\tau} P_i\sigma_{\mu_i}$. Some Q' must exist such that $P_i\sigma_{\mu_i} \approx_o Q'$ and either $Q\sigma_{\mu_i} \stackrel{\tau}{\Longrightarrow} Q'$ or $Q\sigma_{\mu_i} \equiv Q'$. By induction hypothesis $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P_i\sigma_{\mu_i} = \tau.Q'$ is provable. In the first case it can be easily shown that $AS_c \cup \{T1, T2, T3, T4\} \vdash \mu_i\tau.P_i + Q = Q$, using Lemma 16. In the second case one has

$$\begin{aligned} AS_c \cup \{T1, T2, T3, T4\} \vdash \mu_i \tau.P_i &\stackrel{MD3}{=} \mu_i \tau.P_i\sigma_{\mu_i} \\ &= \mu_i \tau.Q\sigma_{\mu_i} \\ &\stackrel{MD3}{=} \mu_i \tau.Q \end{aligned}$$

In summary $AS_c \cup \{T1, T2, T3, T4\} \vdash \sum_{i\in I_3} \mu_i\tau.P_i + Q = \sum_{i\in I} \mu_i\tau.Q + Q$ for some subset I of I_3. So $AS_c \cup \{T1, T2, T3, T4\} \vdash P+Q = \sum_{i\in I} \mu_i\tau.Q + Q$. It follows from Lemma 15 that $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.(P+Q) = \tau.(\sum_{i\in I} \mu_i\tau.Q + Q) = \tau.Q$. Symmetrically $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.(P+Q) = \tau.P$. Therefore $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.Q$. □

The promotion lemma relates the algebraic semantics to equational rewriting. It promotes a pair of semantically equivalent processes to a pair of proof theoretically equal processes.

2.8 Completeness

The saturation and promotion properties suffice to establish the following absorption property:

> If P and Q are congruent then $P + Q$ is provably equal to P.

Of course, under the same assumption, $P + Q$ is also provably equal to Q. Hence the completeness.

Theorem 18 (completeness). *$AS_c \cup \{T1, T2, T3, T4\}$ is sound and complete for $=_o$.*

Proof. The soundness part is clear. The proof of completeness is almost the same as that of Lemma 17. Suppose $P =_o Q$ for normal form processes P and Q and that $\mu_i\tau.P_i$ is a summand of P. Then $(\mu_i\tau.P_i)\sigma_{\mu_i} \xrightarrow{\tau} P_i\sigma_{\mu_i}$ must be matched up by $Q\sigma_{\mu_i} \stackrel{\tau}{\Longrightarrow} Q'$ for some Q' such that $P_i\sigma_{\mu_i} \approx_o Q'$. By Lemma 17, $AS_c \cup \{T1, T2, T3, T4\} \vdash \tau.P_i\sigma_{\mu_i} = \tau.Q'$. Therefore $AS_c \cup \{T1, T2, T3, T4\} \vdash \mu_i\tau.P_i + Q = Q$. Thus $AS_c \cup \{T1, T2, T3, T4\} \vdash P+Q = Q$. Symmetrically $AS_c \cup \{T1, T2, T3, T4\} \vdash P+Q = P$. Therefore $AS_c \cup \{T1, T2, T3, T4\} \vdash P = Q$. □

The proof of the completeness theorem is very similar to that of promotion lemma. The difference is to do with the induction step. In the latter proof one uses induction hypothesis whereas in the former proof one refers to the promotion lemma. This is a general phenomenon. As a matter of fact the difference between the proof of a promotion property and that of the associated completeness theorem is so minor and routine that in most cases the latter proof can be safely omitted.

2.9 Remark

The previous sections convey three pieces of information. The first is that Hennessy Lemma does not hold in calculi of mobile processes. The second is that Milner's three tau laws are insufficient for weak open congruence on mobile processes. And the third is that a new tau law is necessary to deal with match operator under prefix operation. The first two are closely related. All the three observations have wide implications to non-symbolic approach to axiomatization.

It has come a long way to settle down on the axiom T4. The first solution, proposed by Fu in an early version of [14], is a conditional equation rule formulated as follows:

$$\frac{P+\sum_{i\in I}\mu_i\tau.P = Q+\sum_{j\in J}\mu_i\tau.Q}{\tau.P=\tau.Q}$$

The premises of the rule is an equational formalization of $P \approx_o Q$. The role of the rule is to promote $P \approx_o Q$ to $\tau.P = \tau.Q$, which is necessary to allow the proof of Lemma 17 to go through. In the final version of [14] Fu made an observation that the rule is equivalent to the following law:

$$\tau.P = \tau.(P+\sum_{i\in I}\mu_i\tau.P)$$

It was after the submission of the final version of [14] that he realized that, in the presence of $AS_c \cup \{T1, T2, T3\}$, the above equality can be simplified to

$$\tau.P = \tau.(P+\mu\tau.P)$$

which can be further simplified to T4. It is difficult to imagine an equivalent axiom that is simpler than T4. This fourth tau law pinpoints the places where Milner's three tau laws fail. In the presence of the match operator, Milner's tau laws fail to cover all the aspects of mobility incurred by internal communication.

In [14] the following equality is given as an equality that holds semantically but is possibly not provable by Milner's tau laws:

$$\tau.(\overline{a}x + [x{=}y]\overline{a}y) = \tau.\overline{a}x \tag{4}$$

Notice that (4) is an instance of T4. The discovery of T4 led us immediately to

$$\tau.[x{=}y]\tau = \tau \tag{5}$$

This is probably the simplest counter example. Thanks to its simplicity, we are able to formally justify our intuition that (5) is not provable using only Milner's tau laws.

3 The Pi Calculus without Mismatch

The π-calculus without the mismatch operator is the calculus of nondeterministic mobile processes extended with the parallel composition operator and the restriction operator. In this section we investigate some aspects of the algebraic theory of this language. The novelty of this section is as follows: (i) We give a complete system for the weak open congruence of the π-calculus without the mismatch operator using distinction. (ii) We give an alternative complete system without using distinction.

3.1 Parallel Composition

Syntactically if P and Q are processes then $P|Q$ is a process of the parallel composition form. Semantically the two components of the processes $P|Q$ can either evolve independently or communicate through common names. In π-calculus the operational semantics of the operator is defined as follows:

Composition

$$\frac{P \xrightarrow{\lambda} P' \qquad bn(\lambda)\cap fn(Q)=\emptyset}{P|Q \xrightarrow{\lambda} P'|Q} \qquad \frac{P \xrightarrow{a(x)} P' \qquad Q \xrightarrow{\overline{a}y} Q'}{P|Q \xrightarrow{\tau} P'[y/x]|Q'} \qquad \frac{P \xrightarrow{a(x)} P' \qquad Q \xrightarrow{\overline{a}(x)} Q'}{P|Q \xrightarrow{\tau} (x)(P'|Q')}$$

We have left out the symmetric rules. This operator does not have any effect on the definition of open bisimilarity. Definition 22 remains valid. From the point of view of axiomatization, the parallel composition operator can be easily dealt with using the well-known expansion law. In π-calculus the law takes the following form ([32]):

$$P|Q \quad = \quad \sum_{i\in I}\mu_i\lambda_i.(P_i|Q) + \sum_{j\in J}\mu_j\lambda_j.(P|Q_j) + \sum_{\substack{\lambda_i=a(x)\\ \lambda_j=\overline{a}y}}\mu_i\mu_j\tau.(P_i[y/x]|Q_j) + \sum_{\substack{\lambda_i=\overline{a}y\\ \lambda_j=a(x)}}\mu_i\mu_j\tau.(P_i|Q_j[y/x])$$

where P is $\sum_{i\in I}\mu_i\lambda_i.P_i$ and Q is $\sum_{j\in J}\mu_j\lambda_j.Q_j$. This law makes most evident the philosophy of interleaving semantics. The expansion law for the full π-calculus with the mismatch operator can be obtained by replacing the μ_i and μ_j in the above law by φ_i and φ_j respectively, where φ_i and φ_j denote arbitrary finite sequences of mixed match and/or mismatch operators.

In the rest of this paper we ignore the parallel composition operator most of the time. This operator is significant only in Section 4.1.5.

3.2 Restriction

A restriction process usually takes the form $(x)P$, where (x) is the restriction operator. The name x in $(x)P$ is also bound. The operational semantics of the subcalculus of π with the prefix, restriction, match and choice operators is defined below.

Prefix

$$\frac{}{\pi.P \xrightarrow{\pi} P}$$

Restriction

$$\frac{P \xrightarrow{\lambda} P' \qquad x \notin n(\lambda)}{(x)P \xrightarrow{\lambda} (x)P'} \qquad\qquad \frac{P \xrightarrow{\overline{a}x} P'}{(x)P \xrightarrow{\overline{a}(x)} P'}$$

Match

$$\frac{P \xrightarrow{\lambda} P'}{[x{=}x]P \xrightarrow{\lambda} P'}$$

Choice

$$\frac{P \xrightarrow{\lambda} P'}{P{+}Q \xrightarrow{\lambda} P'} \qquad\qquad \frac{Q \xrightarrow{\lambda} Q'}{P{+}Q \xrightarrow{\lambda} Q'}$$

In the above rules λ ranges over $\{a(x), \overline{a}x, \overline{a}(x) \mid a, x \in \mathcal{N}\} \cup \{\tau\}$. The label $\overline{a}(x)$ denotes a restricted output, where x is bound. We will also use λ to range over the set of extended prefixes, which contains the tau, the input prefixes, the output prefixes and the restricted output prefixes. A restricted output prefix is defined as follows:

$$\overline{a}(x).P \stackrel{\text{def}}{=} (x)\overline{a}x.P$$

The set of the subject names of λ, notation $subj(\lambda)$, is defined as follows: (i) $subj(\tau) \stackrel{\text{def}}{=} \emptyset$; (ii) $subj(a(x)) \stackrel{\text{def}}{=} \{a\}$; (iii) $subj(\overline{a}x) \stackrel{\text{def}}{=} \{a\}$; (iv) $subj(\overline{a}(x)) \stackrel{\text{def}}{=} \{a\}$. The definition of context has to take into consideration of restriction. Formally we add to the previous definition of context the following clause: $(x)C[]$ is a context whenever $C[]$ is a context.

The restriction operator adds a lot of expressive power to the calculus of mobile processes. It also adds complications to semantics. For instance, the definition of the open bisimulation must be modified since a free name introduced by a restricted output action should be kept distinct from any other name. In order to formalize the idea, Milner, Parrow and Walker introduced distinctions ([32]). A distinction imposes permanent inequalities on names.

Definition 19. A distinction is a finite symmetric and irreflexive relation on names. Distinctions will be denoted by D, D' etc.. The notation $D\backslash x$ denotes $D \setminus \{(x,y),(y,x) \mid y \text{ is a name}\}$. The set of distinctions will be denoted by $\mathcal{D}$.

Definition 20. A substitution σ respects a distinction D if $(a,b) \in D$ implies $\sigma(a) \neq \sigma(b)$. Similarly, a match sequence μ respects a distinction D if $(a,b) \in D$ implies $\mu \not\Rightarrow a{=}b$.

L1	$(x)\mathbf{0}$	$=$	$\mathbf{0}$	
L2	$(x)(y)P$	$=$	$(y)(x)P$	
L3	$(x)(P+Q)$	$=$	$(x)P+(x)Q$	
L4	$(x)\pi.P$	$=$	$\pi.(x)P$	if $x \notin n(\pi)$
L5	$(x)\pi.P$	$=$	$\mathbf{0}$	if $x \in subj(\pi)$
L6	$(x)[y{=}z]P$	$=$	$[y{=}z](x)P$	if $x \notin \{y,z\}$
L7	$(x)[x{=}y]P$	$=$	$\mathbf{0}$	if $x \neq y$

Figure 9: General Axioms for Restriction

Using the above definition, Sangiorgi defined a class of distinction indexed bisimulations in [42].

Definition 21. The set $\mathcal{R} = \{\mathcal{R}^D\}_{D\in\mathcal{D}}$ of binary symmetric relations on processes is a strong open bisimulation if, for each distinction $D \in \mathcal{D}$ and for each substitution σ respecting D, $P\mathcal{R}^D Q$ implies the following properties:
(i) If $P\sigma \xrightarrow{\lambda} P'$ and λ is not a restricted output action then Q' exists such that $Q\sigma \xrightarrow{\lambda} Q'$ and $P'\mathcal{R}^{D\sigma}Q'$.
(ii) If $P\sigma \xrightarrow{\overline{a}(x)} P'$ then Q' exists such that $Q\sigma \xrightarrow{\overline{a}(x)} Q'$ and $P'\mathcal{R}^{D'}Q'$, where D' is $D\sigma\cup\{x\}\times fn(P\sigma{+}Q\sigma)\cup fn(P\sigma{+}Q\sigma)\times\{x\}$.
P is strong open bisimilar to Q with respect to D, notation $P \sim_o^D Q$, if there exists a strong open bisimulation $\mathcal{R} = \{\mathcal{R}^D\}_{D\in\mathcal{D}}$ such that $(P,Q) \in \mathcal{R}^D$. We say that P is strong open bisimilar to Q if $P \sim_o^\emptyset Q$.

The equivalence $\sim_o^\emptyset$ will be compared to an alternative definition of strong open bisimilarity for the full π-calculus with the mismatch operator. Since we will take a close look at the weak version of $\sim_o^\emptyset$, we will not say anything more about it except stating a completeness result at the end of this section.

Definition 22. The set $\mathcal{R} = \{\mathcal{R}^D\}_{D\in\mathcal{D}}$ of binary symmetric relations on processes is a weak open bisimulation if for each distinction $D \in \mathcal{D}$ and for each substitution σ respecting D, $P\mathcal{R}^D Q$ implies the following properties:
(i) If $P\sigma \xrightarrow{\lambda} P'$ and λ is not a restricted output action then Q' exists such that $Q\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'$ and $P'\mathcal{R}^{D\sigma}Q'$.
(ii) If $P\sigma \xrightarrow{\overline{a}(x)} P'$ then Q' exists such that $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $P'\mathcal{R}^{D'}Q'$, where D' is $D\sigma\cup\{x\}\times fn(P\sigma{+}Q\sigma)\cup fn(P\sigma{+}Q\sigma)\times\{x\}$.
P is weak open bisimilar to Q with respect to D, notation $P \approx_o^D Q$, if there exists a weak open bisimulation $\mathcal{R} = \{\mathcal{R}^D\}_{D\in\mathcal{D}}$ such that $(P,Q) \in \mathcal{R}^D$. We say that P is weak open bisimilar to Q if $P \approx_o^\emptyset Q$.

In the standard manner one can define the largest congruence $\simeq_o^\emptyset$ contained in $\approx_o^\emptyset$.

The proofs of the two lemmas below use the standard argument ([42]).

Lemma 23. *If $P \approx_o^D Q$ and $D \subseteq D'$ then $P \approx_o^{D'} Q$.*

Lemma 24. *If μ respects D then $P\sigma_\mu \approx_o^{D\sigma_\mu} Q\sigma_\mu$ implies $\mu P \approx_o^D \mu Q$.*

3.3 System Using Distinction

We now begin to investigate the weak open congruence. This subsection follows [42] closely. Our contribution is to give a complete system for the weak open congruence. To get a complete system for $\simeq_o^\emptyset$, we need laws that deal with the restriction operator. Now the system actually allows one to derive equalities indexed by distinctions. So there are also some laws that manipulate indexed equalities. Figure 9 summarizes some general laws for restriction, which appeared first in [32]; and Figure 10 provides three distinction indexed laws due to Sangiorgi ([42]).

Let AS_d be $AS_c \cup \{L1, L2, L3, L4, L5, L6, L7, LI1, LI2, LI3\}$. Note that the law L7 is derivable from $AS_d \setminus \{L7\}$ using LI1 and LI3. In some systems without the indexed equalities LI1, LI2 and LI3, the law L7 is indispensable.

In AS_d one can show that every process can be rewritten to a normal form. But since equalities are now indexed by distinctions, we need to define normal forms with respect to distinctions.

Definition 25. Suppose D is a distinction. A process P is a D-normal form, or D-nf, if $P \equiv \sum_{i\in I} \mu_i\lambda_i.P_i$ such that, for each i, μ_i respects D and $P_i\sigma_{\mu_i}$ is a $D\sigma_{\mu_i}$-normal form.

LI1	$[a{=}b]P$	$=_D$	$\mathbf{0}$	if $(a,b)\in D$
LI2	P	$=_D$	Q	if $P=_{D'}Q$ and $D'\subseteq D$
LI3	$(x)P$	$=_{D\backslash x}$	$(x)Q$	if $P=_D Q$

Figure 10: Distinction Indexed Axioms

The definition of the depth of a process need be modified to take into account of the restriction operator. Let $d((x)P)$ be the same as $d(P)$.

Lemma 26. *Suppose D is a distinction and P is a process. Then there is a D-nf H such that $AS_d \vdash P =_D H$ and $d(H)\le d(P)$.*

The proof of the above lemma can be found in [42]. The proof of the saturation properties, stated in the next lemma, is simple.

Lemma 27. *Suppose Q is a normal form. The following saturation properties hold:*
(i) If $Q\sigma_\mu \stackrel{\overline{a}x}{\Longrightarrow} Q'$ then $AS_d\cup\{T1,T2,T3\}\vdash Q = Q+\mu\overline{a}x.Q'$.
(ii) If $Q\sigma_\mu \stackrel{a(x)}{\Longrightarrow} Q'$ and x does not appear in μ then $AS_d\cup\{T1,T2,T3\}\vdash Q = Q+\mu a(x).Q'$.
(iii) If $Q\sigma_\mu \stackrel{\tau}{\Longrightarrow} Q'$ then $AS_d\cup\{T1,T2,T3\}\vdash Q = Q+\mu\tau.Q'$.
(iv) If $Q\sigma_\mu \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and x does not appear in μ then $AS_d\cup\{T1,T2,T3\}\vdash Q = Q+\mu\overline{a}(x).Q'$.

The next lemma states the promotion properties.

Lemma 28. *If $P\approx_o^D Q$ then $AS_d\cup\{T1,T2,T3,T4\}\vdash \tau.P =_D \tau.Q$.*

Proof. The proof is similar to that of Lemma 17. By Lemma 26, we can assume that both P and Q are D-nf's. We only consider restricted output actions. Suppose $\mu_i\overline{a}(x).P_i$ is a summand of P and $P\sigma_{\mu_i} \stackrel{\overline{a}(x)}{\longrightarrow} P_i\sigma_{\mu_i}$. Then Q' exists such that $Q\sigma_{\mu_i} \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $P_i\sigma_{\mu_i} \approx_o^{D'} Q' \equiv Q'\sigma_{\mu_i}$, where $D' = D\sigma_{\mu_i}\cup\{x\}\times fn(P\sigma_{\mu_i}{+}Q\sigma_{\mu_i})\cup fn(P\sigma_{\mu_i}{+}Q\sigma_{\mu_i})\times\{x\}$. Let D'' be $D\cup\{x\}\times fn(P{+}Q)\cup fn(P{+}Q)\times\{x\}$. Then $\mu_iP_i \approx_o^{D''} \mu_iQ'$ by Lemma 24. So $AS_d\cup\{T1,T2,T3,T4\}\vdash \tau.\mu_iP_i =_{D''} \tau.\mu_iQ'$ by induction hypothesis. Hence $AS_d\cup\{T1,T2,T3,T4\}\vdash \overline{a}x.\mu_iP_i =_{D''} \overline{a}x.\mu_iQ'$, from which it follows from LI3 that $AS_d\cup\{T1,T2,T3,T4\}\vdash \overline{a}(x).\mu_iP_i =_D \overline{a}(x).\mu_iQ'$. Thus

$$\begin{array}{rcl} AS_d\cup\{T1,T2,T3,T4\}\vdash Q & =_D & Q{+}\mu_i\overline{a}(x).Q' \\ & \stackrel{M2}{=}_D & Q{+}\mu_i\overline{a}(x).\mu_iQ' \\ & =_D & Q{+}\mu_i\overline{a}(x).\mu_iP_i \\ & \stackrel{M2}{=}_D & Q{+}\mu_i\overline{a}(x).P_i \end{array}$$

The rest of the proof is similar to that of Lemma 17. □

The proof of the completeness theorem is similar to that of Lemma 28.

Theorem 29. *$AS_d\cup\{T1,T2,T3,T4\}$ is sound and complete for $\simeq_o^\emptyset$.*

For the purpose of completeness we state a relevant result due to Sangiorgi ([42]).

Theorem 30. *AS_d is sound and complete for $\sim_o^\emptyset$.*

We end this section by mentioning that the strong and weak open bisimilarities are conservative extensions over the ones on nondeterministic mobile processes.

Lemma 31. *On the set of nondeterministic mobile processes $\sim_o^\emptyset$ and $\approx_o^\emptyset$ coincide respectively with $\sim_o$ and $\approx_o$.*

The proof of the above lemma amounts to showing that $\sim_o$, respectively $\approx_o$, is a weak, respectively strong, $\emptyset$-open bisimulation and conversely $\sim_o^\emptyset$, respectively $\approx_o^\emptyset$, is a strong, respectively weak, open bisimulation. These can be done in a routine manner.

L8	$(x)C[[x{=}y]P]$	$=$	$(x)C[\mathbf{0}]$	x, y are not bound in $C[]$ and $x{\neq}y$

Figure 11: A Schematic Law for Restriction

3.4 A Schematic Law for Restriction

There is an alternative to the complete system studied in the previous subsection. In this new system one uses a schematic law in favour of the indexed laws LI1, LI2 and LI3. The schematic law, labelled L8, is given in Figure 11. The reason to call it schematic is that an arbitrary context $C[]$ is used in the description of the law. When applying L8 one must make sure that neither x nor y is bound in the context $C[]$ and that they are distinct names. It is clear that the law L7 is a special case of L8 in the presence of L1.

The reason to have L8, a law stronger than L7, is to deal with the match operator under restricted output prefix. For example

$$\overline{a}(x).[x{=}y]P = \overline{a}(x)$$

is an instance of L8. It is very unlikely that it can be derived from L7.

Let AS_r be $AS_c \cup \{L1, L2, L3, L4, L5, L6, L7, L8\}$. Then we have the following important result.

Theorem 32. *AS_r is sound and complete for $\sim_o^{\emptyset}$.*

To prove the theorem we need to introduce some additional machinery and establish quite a few intermediate results. We are not going into these proofs in the present paper. The interested reader is referred to [17].

The generalization of these proofs to the weak case is a formality.

Theorem 33. *$AS_r \cup \{T1, T2, T3, T4\}$ is sound and complete for $\simeq_o^{\emptyset}$.*

4 The Full Pi Calculus

Complete systems for the strong late congruence and the strong early congruence are given in [37]. The version of the π-calculus they use has the mismatch operator. Although the operator is debatable from a computational point of view, its role in axiomatization seems indispensable in many places. The full π-calculus with the mismatch operator has the following abstract syntax:

$$P := \mathbf{0} \mid \pi.P \mid (x)P \mid P{+}P \mid [x{=}y]P \mid [x{\neq}y]P$$

where $\pi := a(x) \mid \overline{a}x \mid \tau$. The operational semantics of the mismatch operator is as follows:

Mismatch

$$\frac{P \xrightarrow{\lambda} P' \qquad x \neq y}{[x{\neq}y]P \xrightarrow{\lambda} P'}$$

In the presence of the mismatch operator the operational semantics is no longer preserved by substitution. In other words the *stability property* for operational semantics, as stated below, fails:

If $P \xrightarrow{\lambda} P'$ then $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$ for any substitution σ.

However the next lemma is valid.

Lemma 34. *Suppose $P\sigma \xrightarrow{\lambda'} P''$ then P' and λ exist such that $P'' \equiv P'\sigma$ and $\lambda' = \lambda\sigma$.*

So we may write $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$ without losing any generality. This fact will be used implicitly later on.

Suppose Y is a finite set $\{y_1, \ldots, y_n\}$ of names. The notation $[y{\notin}Y]P$ stands for $[y{\neq}y_1]\ldots[y{\neq}y_n]P$. When using this notation, the order of the mismatch operators in $[y{\neq}y_1]\ldots[y{\neq}y_n]P$ is not relevant. From now on we will write ϕ and ψ to stand for sequences of match and mismatch operators concatenated one after another, δ for a sequence of mismatch operators, and as before μ for a sequence of match operators. Consequently we write ψP. When the length of ψ is zero, ψP is just P. The notation $\phi \Rightarrow \psi$ says that ϕ logically implies ψ and $\phi \Leftrightarrow \psi$ that ϕ and ψ are logically equivalent. The closure of ψ, notation $c(\psi)$, is defined as follows:

For x, y such that $x \neq y$, $c(\psi)$ contains $x = y$, respectively $x \neq y$, whenever $\psi \Rightarrow x{=}y$, respectively $\psi \Rightarrow x{\neq}y$.

We will write $\psi_{\backslash x}$ for the sequence of match and mismatch operators in $c(\psi)$ that do not involve x. We will write $\neg\psi$ to mean the disjunction, so to speak, of the negations of the components of ψ. For instance if ψ is $[x{=}y][a{\neq}b][c{=}d]$ then $\neg\psi P$ stands for $[x{\neq}y]P{+}[a{=}b]P{+}[c{\neq}d]P$.

Definition 35. Let V be a finite set of names. We say that ψ is complete on V if $n(\psi) \subseteq V$ and for each pair x, y of names in V it holds that either $\psi \Rightarrow x{=}y$ or $\psi \Rightarrow x{\neq}y$.

Suppose ψ is complete on V and $n(\phi) \subseteq V$. Then it should be clear that either $\psi\phi \Leftrightarrow \psi$ or $\psi\phi \Leftrightarrow \bot$. In sequel this fact will be used implicitly.

Definition 36. Suppose σ is a substitution and ψ is a sequence of match/mismatch operators. Then σ respects ψ if $\psi \Rightarrow x{=}y$ implies $\sigma(x) = \sigma(y)$ and $\psi \Rightarrow x{\neq}y$ implies $\sigma(x) \neq \sigma(y)$. Dually ψ respects σ if $\sigma(x) = \sigma(y)$ implies $\psi \Rightarrow x{=}y$ and $\sigma(x) \neq \sigma(y)$ implies $\psi \Rightarrow x{\neq}y$. Moreover σ agrees with ψ, and ψ agrees with σ, if they respect each other. The substitution σ is induced by ψ if it agrees with ψ and $rng(\sigma) \subseteq n(\psi)$.

Definition 37. Suppose δ is a sequence of mismatch operators and D is a distinction. Then D respects δ if $\delta \Rightarrow x{\neq}y$ implies $(x, y) \in D$. Dually δ respects D if $(x, y) \in D$ implies $\delta \Rightarrow x{\neq}y$. Moreover δ agrees with D if they respect each other.

The proof of the next lemma can be found in [37].

Lemma 38. *If ϕ and ψ are complete on V and both agree with σ then $\phi \Leftrightarrow \psi$.*

When defining open bisimilarities we need, for every pair x, y of names, an auxiliary operation denoted by $(\_)^{[x\neq y]}$. This operation inserts the mismatch operator $[x{\neq}y]$ in front of every prefix operator. The structural definition goes as follows:

$$\begin{aligned}
(\mathbf{0})^{[x\neq y]} &\stackrel{\text{def}}{=} \mathbf{0} \\
(\pi.P)^{[x\neq y]} &\stackrel{\text{def}}{=} [x{\neq}y]\pi.P^{[x\neq y]} \text{ where } \{x, y\} \cap bn(\pi) = \emptyset \\
((z)P)^{[x\neq y]} &\stackrel{\text{def}}{=} (z)P^{[x\neq y]} \text{ where } z \notin \{x, y\} \\
(P{+}Q)^{[x\neq y]} &\stackrel{\text{def}}{=} P^{[x\neq y]}{+}Q^{[x\neq y]} \\
([u{=}v]P)^{[x\neq y]} &\stackrel{\text{def}}{=} [u{=}v]P^{[x\neq y]} \\
([u{\neq}v]P)^{[x\neq y]} &\stackrel{\text{def}}{=} [u{\neq}v]P^{[x\neq y]}
\end{aligned}$$

When constructing $P^{[x\neq y]}$ one must rename the bound names of P to prevent x, y from being captured. The following lemma records a simple fact about this operation. It says that the presence of a mismatch operator is irrelevant from an operational point of view if the names involved are free and distinct.

Lemma 39. *Suppose $bn(\lambda) \cap \{x, y\} = \emptyset$ and x, y are distinct names. Then the following properties hold:*
(i) If $P \xrightarrow{\lambda} P'$ and $x \neq y$ then $P^{[x\neq y]} \xrightarrow{\lambda} (P')^{[x\neq y]}$.
(ii) If $P^{[x\neq y]} \xrightarrow{\lambda} P''$ then P' exists such that $P \xrightarrow{\lambda} P'$ and $P'' \equiv (P')^{[x\neq y]}$.

For $Y = \{y_1, \ldots, y_n\}$, let $P^{[x\notin Y]}$ denote $(\ldots(P^{[x\neq y_1]})^{[x\neq y_2]}\ldots)^{[x\neq y_n]}$. When this notation is used the order of applying the operations $(\_)^{[x\neq y_1]}, \ldots, (\_)^{[x\neq y_n]}$ is definitely not important.

It should be easy to see that the operation $(\_)^{[x\neq y]}$ is to make distinctions syntactically explicit.

4.1 Bisimulation Theory of the Full Calculus

The stability of the operational semantics simplifies the theory of the bisimulation equivalence because it allows one to define bisimilarities that enjoy the following property:

If $P \xrightarrow{\lambda} P'$ is simulated by $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'$ then $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$ is simulated by $Q\sigma \stackrel{\widehat{\lambda\sigma}}{\Longrightarrow} Q'\sigma$.

This property holds for the open bisimilarities of the calculi studied in the previous two sections. If stability fails, the above property is too strong to be of practical interest. In this case the best one could expect, even though we know that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'$ simulates $P \stackrel{\lambda}{\longrightarrow} P'$, is that for each substitution σ some Q_1 exists such that $Q\sigma \stackrel{\widehat{\lambda\sigma}}{\Longrightarrow} Q_1$ simulates $P\sigma \stackrel{\lambda\sigma}{\longrightarrow} P'\sigma$ provided that $P\sigma$ can perform the action $\lambda\sigma$. It is therefore natural to anticipate that the bisimulation theory without the stability property is quite different from what we have seen in the previous two sections.

The full π-calculus fails to hold of the stability property. The algebraic investigation we have carried out for the π-calculus without the mismatch operator needs to be redone, starting from the very definitions of bisimulations.

4.1.1 Late and Early Equivalences

It is well-known ([32]) that there are two versions of strong bisimilarity, late and early bisimilarities. They differ in the treatment of input actions. The early approach regards input as an action that receives a name at the same time the action happens whereas the late one thinks of input as a commitment of an action that is to pick up a name before the next observable action occurs. The latter reminds one of asynchronous communication. It is a rather intentional interpretation of the operational semantics. The late bisimilarity is strictly stronger than the early one whether the model has mismatch operator or not. Let's review the definitions of the strong early equivalence and the strong late equivalence.

Definition 40. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a strong late bisimulation if whenever $P\mathcal{R}Q$ then the following property holds:
(i) If λ is not an input action and $P \stackrel{\lambda}{\longrightarrow} P'$ then some Q' exists such that $Q \stackrel{\lambda}{\longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P \stackrel{a(x)}{\longrightarrow} P'$ then some Q' exists such that $Q \stackrel{a(x)}{\longrightarrow} Q'$ and, for every y, $Q'[y/x]\mathcal{R}P'[y/x]$.
The strong late bisimilarity $\dot{\sim}_l$ is the largest strong late bisimulation. Two processes P and Q are strongly late equivalent, notation $P \sim_l Q$, if $P\sigma\dot{\sim}_l Q\sigma$ for every substitution σ.

Definition 41. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a strong early bisimulation if whenever $P\mathcal{R}Q$ then the following property holds:
(i) If λ is not an input action and $P \stackrel{\lambda}{\longrightarrow} P'$ then some Q' exists such that $Q \stackrel{\lambda}{\longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P \stackrel{a(x)}{\longrightarrow} P'$ then, for every y, some Q' exists such that $Q \stackrel{a(x)}{\longrightarrow} Q'$ and $Q'[y/x]\mathcal{R}P'[y/x]$.
The strong early bisimilarity $\dot{\sim}_e$ is the largest strong early bisimulation. Two processes P and Q are strongly early equivalent, notation $P \sim_e Q$, if $P\sigma\dot{\sim}_e Q\sigma$ for every substitution σ.

It is easily seen that $\sim_l$ is strictly contained in $\sim_e$. The following is an example exhibiting the difference: One has

$$a(x).R{+}a(x){+}a(x).[x{=}y]R \ \sim_e \ a(x).R{+}a(x)$$

but not

$$a(x).R{+}a(x){+}a(x).[x{=}y]R \ \sim_l \ a(x).R{+}a(x)$$

Using the mismatch operator a more general counter example can be given: It is clear that

$$a(x).P{+}a(x).Q{+}a(x).([x{=}y]P{+}[x{\neq}y]Q) \ \sim_e \ a(x).P{+}a(x).Q$$

but not

$$a(x).P{+}a(x).Q{+}a(x).([x{=}y]P{+}[x{\neq}y]Q) \ \sim_l \ a(x).P{+}a(x).Q$$

As a matter of fact this counter example is so general that it characterizes the difference between early and late equivalences.

An interesting equality that holds for both the strong early equivalence and the strong late equivalence is the following

$$C[[x{=}y]P]{+}C[\mathbf{0}] \sim_l [x{=}y]C[P]{+}C[\mathbf{0}]$$

where both x and y are free in $C[[x{=}y]P]$. If a substitution σ identifies x and y then $C[[x{=}y]P]\sigma$ can be simulated by $([x{=}y]C[P])\sigma$, otherwise it can be simulated by $C[\mathbf{0}]\sigma$.

Next we define the corresponding weak equivalences.

Definition 42. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a weak late bisimulation if whenever $P\mathcal{R}Q$ then the following property holds:
(i) If λ is not an input prefix and $P \xrightarrow{\lambda} P'$ then some Q' exists such that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P \xrightarrow{a(x)} P'$ then some Q' exists such that $Q \Longrightarrow\xrightarrow{a(x)} Q'$ and for every y some Q'' exists such that $Q'[y/x]\Longrightarrow Q''\mathcal{R}P'[y/x]$.
The weak late bisimilarity $\dot{\approx}^l_w$ is the largest weak late bisimulation. Two processes P and Q are weakly late equivalent, notation $P \approx^l_w Q$, if $P\sigma\dot{\approx}^l_w Q\sigma$ for every substitution σ.

Let $\dot{\approx}^{l'}_w$ be the alternative of $\dot{\approx}^l_w$ defined by replacing the clause (ii) by the following requirement

(ii') If $P \xrightarrow{a(x)} P'$ then some Q' exists such that $Q \stackrel{a(x)}{\Longrightarrow} Q'$ and $Q'[y/x]\mathcal{R}P'[y/x]$ for every y.

It is well-known that $\dot{\approx}^{l'}_w$ is not even an equivalence relation. The following counter example is taken from [42]. Let α be different from τ and define:

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} c(a).[a{=}b]\alpha{+}c(a).(\tau.[a{=}b]\alpha{+}\tau.\alpha{+}\tau) \\ P_2 &\stackrel{\text{def}}{=} c(a).(\tau.[a{=}b]\alpha{+}\tau.\alpha{+}\tau) \\ P_3 &\stackrel{\text{def}}{=} c(a).(\tau.\alpha{+}\tau) \end{aligned}$$

Then $P_1\dot{\approx}^{l'}_w P_2\dot{\approx}^{l'}_w P_3$ but $P_1\dot{\not\approx}^{l'}_w P_3$. On the other hand, $P_1\dot{\approx}^{l}_w P_2\dot{\approx}^{l}_w P_3$ and $P_1\dot{\approx}^{l}_w P_3$.

Definition 43. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a weak early bisimulation if whenever $P\mathcal{R}Q$ then the following property holds:
(i) If λ is not an input action and $P \xrightarrow{\lambda} P'$ then some Q' exists such that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P \xrightarrow{a(x)} P'$ then for every y some Q' and Q'' exist such that $Q \Longrightarrow\xrightarrow{a(x)} Q''$ and $Q''[y/x]\Longrightarrow Q'\mathcal{R}P'[y/x]$.
The weak early bisimilarity $\dot{\approx}^e_w$ is the largest weak early bisimulation. Two processes P and Q are weakly early equivalent, notation $P \approx^e_w Q$, if $P\sigma\dot{\approx}^e_w Q\sigma$ for every substitution σ.

The next two lemmas appeared in [37], in which $\approx$ stands for either $\approx^e_w$ or $\approx^l_w$, and similarly $\dot{\approx}$ stands for either $\dot{\approx}^e_w$ or $\dot{\approx}^l_w$.

Lemma 44. *Suppose that σ is injective on $fn(P{+}Q)$. Then $P\dot{\approx}Q$ if and only if $P\sigma\dot{\approx}Q\sigma$.*

Lemma 45. *Suppose ψ is complete on $V \supseteq fn(P{+}Q)$ and σ agrees with ψ. Then $\psi P\sigma\dot{\approx}\psi Q\sigma$ if and only if $\psi P\dot{\approx}\psi Q$ if and only if $(\psi P)\sigma\dot{\approx}(\psi Q)\sigma$ if and only if $\psi P \approx \psi Q$.*

Proof. By Lemma 44 we may assume that σ is induced by ψ. It follows easily from $[x{=}y]P\dot{\approx}[x{=}y]P[y/x]$ that $\psi P\sigma\dot{\approx}\psi Q\sigma$ if and only if $\psi P\dot{\approx}\psi Q$. Now σ is injective on $fn(P\sigma{+}Q\sigma)$. So $(\psi P\sigma)\sigma\dot{\approx}(\psi Q\sigma)\sigma$ by Lemma 44. Therefore $(\psi P)\sigma\dot{\approx}(\psi Q)\sigma$. Consequently $\psi P \approx \psi Q$. □

4.1.2 Strong Open Bisimilarity

The strong open bisimilarity introduced by Sangiorgi makes use of distinctions, whose role is to record names that should be kept distinct from each other in two relevant simulating processes. Distinctions formalize the fact that, in π-calculus, a local name in a process is never identified with any other name during the evolution of the process. In this paper we show that the distinctions are not necessary in the presence of the mismatch operator. To start with we show how this can be achieved for strong open bisimilarity.

Definition 46. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a strong open bisimulation if whenever $P\mathcal{R}Q$ then the following properties hold for every substitution σ:
(i) If $P\sigma \xrightarrow{\lambda} P'$, where λ is not a restricted output action, then some Q' exists such that $Q\sigma \xrightarrow{\lambda} Q'\mathcal{R}P'$.
(ii) If $P\sigma \xrightarrow{\overline{a}(x)} P'$ and $x \notin fn(Q)$ then some Q' exists such that $Q\sigma \xrightarrow{\overline{a}(x)} Q'$ and $(P')^{[x\notin Y]}\mathcal{R}(Q')^{[x\notin Y]}$, where Y is $fn(P'{+}Q') \setminus \{x\}$.
The strong open bisimilarity $\sim_o$ is the largest strong open bisimulation.

Intuitively a restricted name is kept distinct from any other name by inserting enough mismatch operators in front of all occurrences of prefix. This is necessary because names can be instantiated at every bisimulation step in the open semantics. It should be pointed out that in the above definition

$$(P')^{[x\notin(fn(P'+Q')\setminus\{x\})]}\mathcal{R}(Q')^{[x\notin(fn(P'+Q')\setminus\{x\})]}$$

should not be replaced by $(P')^{[x\notin(fn(P')\setminus\{x\})]}\mathcal{R}(Q')^{[x\notin(fn(Q')\setminus\{x\})]}$. Otherwise it would result in a bad equivalence relation. But for the above definition it is not obvious that $\sim_o$ is an equivalence relation. We are not going to prove this fact here since we will establish the equivalence property for the weak open bisimilarities.

We have also overloaded the notation $\sim_o$. Some clarification is called for. In what follows we show that the two definitions of $\sim_o$ are consistent. We need two auxiliary lemmas whose proofs we omit, the first of which is due to Sangiorgi ([42]).

Lemma 47. *Suppose P and Q are processes in the π-calculus without the mismatch operator, D is a distinction and σ is a substitution. If $P \sim_o^D Q$ and σ respects D then $P\sigma \sim_o^{D\sigma} Q\sigma$.*

Lemma 48. *Suppose D is a distinction. If $P \sim_o^D Q$ and $x \notin fn(P+Q)$ then $P \sim_o^{D\setminus x} Q$.*

In order to get our result, we prove a more general result.

Theorem 49. *Suppose P and Q are processes in the π-calculus without the mismatch and the distinction D agrees with the substitution δ. Then $P \sim_o^D Q$ if and only if $P^\delta \sim_o Q^\delta$.*

Proof. (i) Let $\mathcal{R}$ be $\{(P^\delta, Q^\delta) \mid P \sim_o^D Q$ and D agrees with δ for distinction D and substitution $\sigma\}\cup \sim_o$. We prove that $\mathcal{R}$ is a strong open bisimulation in the sense of Definition 21. First of all we need to show that $\mathcal{R}$ is closed under substitution. Now suppose $P^\delta\mathcal{R}Q^\delta$, that is $P \sim_o^D Q$ and D agrees with δ. If σ respects δ then σ respects D, therefore $P\sigma \sim_o^{D\sigma} Q\sigma$ by Lemma 47, and consequently $(P\sigma)^{\delta\sigma}\mathcal{R}(Q\sigma)^{\delta\sigma}$. If σ does not respect δ then $(P^\delta)\sigma \sim_o \mathbf{0} \sim_o (Q^\delta)\sigma$.

Now suppose that $P \sim_o^D Q$ and that D agrees with δ. If $P^\delta \xrightarrow{\lambda} P''$ then, by Lemma 39, one has $P \xrightarrow{\lambda} P'$ and $P'' \equiv P'^\delta$. There are two cases:

- λ is not a restricted output action. It follows from $P \sim_o^D Q$ that $Q \xrightarrow{\lambda} Q' \sim_o^D P'$ for some Q'. By Lemma 39, $Q^\delta \xrightarrow{\lambda} Q'^\delta\mathcal{R}P'^\delta$.
- λ is a restricted output action $\overline{a}(x)$. It follows from $P \sim_o^D Q$ that $Q \xrightarrow{\overline{a}(x)} Q' \sim_o^{D'} P'$, where D' is $D \cup \{x\}\times fn(P+Q)$. Clearly $Q^\delta \xrightarrow{\overline{a}(x)} Q'^\delta$ and D' agrees with $\delta[x\notin fn(P+Q)]$. It is also clear that $fn(P'+Q') \setminus \{x\} \subseteq fn(P+Q)$. By Lemma 48, $P' \sim_o^{D'} Q'$ implies $P' \sim_o^{D''} Q'$ where D'' is a distinction that agrees with $\delta[x\notin(fn(P'+Q') \setminus \{x\})]$. So $P'^{\delta[x\notin(fn(P'+Q')\setminus\{x\})]}\mathcal{R}Q'^{\delta[x\notin(fn(P'+Q')\setminus\{x\})]}$.

So $\mathcal{R}$ is a strong open bisimulation in the sense of Definition 21. Thus $P \sim_o^D Q$ implies $P^\delta \sim_o Q^\delta$.

(ii) The proof that $P^\delta \sim_o Q^\delta$ implies $P \sim_o^D Q$, where D agrees with δ, is similar to (i). For a distinction D, let $\mathcal{R}^D$ be $\{(P,Q) \mid P^\delta \sim_o Q^\delta,\ \delta$ agrees with $D\}$ and let $\mathcal{R}$ be $\{\mathcal{R}^D\}_{D\in\mathcal{D}}$. It can be proved that $\mathcal{R}$ is a strong open bisimulation, in the sense of Definition 46, for the π-calculus without the mismatch operator. □

Corollary 50. *$P \sim_o^\emptyset Q$ if and only if $P \sim_o Q$ for P, Q that do not contain any mismatch operator.*

Therefore $\sim_o$ of the full π-calculus is a conservative extension of $\sim_o^\emptyset$, which by Lemma 31 is a conservative extension of $\sim_o$ of the calculus of nondeterministic mobile processes. So the confusions are harmless.

4.1.3 Weak Open Bisimilarities

In the presence of the mismatch operator there are at least three weak versions of open bisimilarity. In what follows we examine each of the weak open bisimilarity in turn. We first take a look at the most loose weak open congruence. It is an obvious generalization of the strong open bisimilarity. This is akin to Sangiorgi's version of weak open congruence he introduced for mobile processes without the mismatch operator.

Definition 51. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a weak open bisimulation if whenever $P\mathcal{R}Q$ then the following properties hold for every substitution σ:
(i) If $P\sigma \stackrel{\lambda}{\longrightarrow} P'$, where λ is not a restricted output action, then some Q' exists such that $Q\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P\sigma \stackrel{\overline{a}(x)}{\longrightarrow} P'$ and $x \notin fn(Q)$ then some Q' exists such that $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(P')^{[x\notin Y]}\mathcal{R}(Q')^{[x\notin Y]}$, where Y is $fn(P'+Q') \setminus \{x\}$.
The weak open bisimilarity $\approx_o^w$ is the largest weak open bisimulation.

Weak open bisimilarity is a very loose equivalence relation. This can be seen from the following example:

$$a(x).[x{\neq}y]\tau.P{+}a(x).P \approx_o^w a(x).[x{\neq}y]\tau.P$$

The action $a(x).[x{\neq}y]\tau.P + a(x).P \stackrel{a(x)}{\longrightarrow} P$ is simulated by $a(x).[x{\neq}y]\tau.P \stackrel{a(x)}{\longrightarrow}\stackrel{\tau}{\longrightarrow} P$. The equivalence is reasonable in the absence of the parallel operator. But the combination of the mismatch and the composition would render anomalous the weak open bisimilarity. For instance the above equivalence does not imply the following

$$(a(x).[x{\neq}y]\tau.P{+}a(x).P)|\overline{a}y \approx_o^w (a(x).[x{\neq}y]\tau.P)|\overline{a}y$$

The action $(a(x).[x{\neq}y]\tau.P{+}a(x).P)|\overline{a}y \stackrel{\tau}{\longrightarrow} P[y/x]|\mathbf{0}$ can not be matched up by any action from the process $(a(x).[x{\neq}y]\tau.P)|\overline{a}y$. In our view $(a(x).[x{\neq}y]\tau.P{+}a(x).P)|\overline{a}y$ and $(a(x).[x{\neq}y]\tau.P)|\overline{a}y$ should not be identified by any reasonable bisimulation equivalence.

The anomaly is caused by the delay of instantiation of input actions, which would not have been a great problem in the absence of the mismatch operator. With the mismatch operator around it seems better to let the instantiation happen immediately. For that purpose we introduce below two versions of refinement of the weak open bisimilarity. The first one is the late open bisimilarity.

Definition 52. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is a late open bisimulation if whenever $P\mathcal{R}Q$ then the following properties hold for every substitution σ:
(i) If $P\sigma \stackrel{\lambda}{\longrightarrow} P'$, where λ is neither a restricted output action nor an input action, then some Q' exists such that $Q\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P\sigma \stackrel{a(x)}{\longrightarrow} P'$ then some Q' exists such that $Q\sigma \Longrightarrow\stackrel{a(x)}{\longrightarrow} Q'$, and for each y, some Q'' exists such that $Q'[y/x]{\Longrightarrow}Q''\mathcal{R}P'[y/x]$.
(iii) If $P\sigma \stackrel{\overline{a}(x)}{\longrightarrow} P'$ and $x \notin fn(Q)$ then some Q' exists such that $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(P')^{[x\notin Y]}\mathcal{R}(Q')^{[x\notin Y]}$, where Y is $fn(P'+Q') \setminus \{x\}$.
The late open bisimilarity $\approx_o^l$ is the largest late open bisimulation.

The third open bisimilarity we consider is the early open bisimilarity. The difference between the late open bisimilarity and the early open bisimilarity is in analogy to that between the weak late bisimilarity and the weak early bisimilarity.

Definition 53. Suppose $\mathcal{R}$ is a symmetric binary relation on the set of processes. The relation $\mathcal{R}$ is an early open bisimulation if whenever $P\mathcal{R}Q$ then the following properties hold for every substitution σ:
(i) If $P\sigma \stackrel{\lambda}{\longrightarrow} P'$, where λ is neither a restricted output action nor an input action, then some Q' exists such that $Q\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P\sigma \stackrel{a(x)}{\longrightarrow} P'$ then for every y some Q' and Q'' exists such that $Q\sigma \Longrightarrow\stackrel{a(x)}{\longrightarrow} Q''$ and $Q''[y/x] \Longrightarrow Q'\mathcal{R}P'[y/x]$.
(iii) If $P\sigma \stackrel{\overline{a}(x)}{\longrightarrow} P'$ and $x \notin fn(Q)$ then some Q' exists such that $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(P')^{[x\notin Y]}\mathcal{R}(Q')^{[x\notin Y]}$, where Y is $(fn(P'+Q') \setminus \{x\})$.
The early open bisimilarity $\approx_o^e$ is the largest early open bisimulation.

As one would expect the late open bisimilarity is stronger than the early open bisimilarity. The next lemma summarizes the relationship among the three open bisimilarities.

Lemma 54. $\approx_o^l \subset \approx_o^e \subset \approx_o^w$ *and the inclusions are strict.*

Proof. The first inclusion is by definition. Here is a counter example that justifies the strictness of the inclusion:

$$a(x).[x{\neq}y]\tau.P{+}a(x).[x{=}y]\tau.P \approx_o^e a(x).[x{\neq}y]\tau.P{+}a(x).[x{=}y]\tau.P{+}a(x).P$$

but

$$a(x).[x{\neq}y]\tau.P{+}a(x).[x{=}y]\tau.P \not\approx_o^l a(x).[x{\neq}y]\tau.P{+}a(x).[x{=}y]\tau.P{+}a(x).P$$

Suppose $P \approx_o^e Q$ and $P \xrightarrow{a(x)} P'$. Then some Q'' and Q' exist such that $Q \Longrightarrow\xrightarrow{a(x)} Q''$ and $Q'' \Longrightarrow Q' \approx_o^e P'$. That is $Q \stackrel{a(x)}{\Longrightarrow} Q' \approx_o^e P'$. Hence $P \approx_o^w Q$. The strictness of the second inclusion is verified by the following counter example:

$$a(x).[x{\neq}y]\tau.P \approx_o^w a(x).[x{\neq}y]\tau.P + a(x).P$$

but

$$a(x).[x{\neq}y]\tau.P \not\approx_o^e a(x).[x{\neq}y]\tau.P + a(x).P$$

This completes the proof. □

4.1.4 Weak Open Bisimilarities are Equivalence Relations

The open bisimilarities are by definition closed under substitution. It is however not obvious that they are equivalence relations. The next lemma, whose proof can be found in Appendix A.1, is the crux of the equivalence proof. It says that the presence or absence of a mismatch operator does not affect the algebraic equality of two processes if the relevant names are distinct and not both of them appear in the processes.

Lemma 55. *The following properties hold for $\approx_o$:*
(i) If $P \approx_o^l Q$ then $P^{[x\neq y]} \approx_o^l Q^{[x\neq y]}$.
(ii) If $P^{[x\neq y]} \approx_o^l Q^{[x\neq y]}$ and $\{x,y\} \not\subseteq fn(P{+}Q)$ then $P \approx_o^l Q$.
(i) and (ii) also hold for $\approx_o^e$ and $\approx_o^w$.

Now we are ready to show that the weak open bisimilarities are equivalence relations.

Lemma 56. *$\approx_o^l$, $\approx_o^e$ and $\approx_o^w$ are all equivalence relations.*

Proof. We prove transitivity and consider only input and restricted output actions. Suppose $P \approx_o^l Q$ and $Q \approx_o^l R$. If $P \xrightarrow{a(x)} P'$ then some Q' and Q'' exist such that $Q \Longrightarrow\xrightarrow{a(x)} Q''$ and, for every y, $Q''[y/x] \Longrightarrow Q' \approx_o^l P'[y/x]$. As $Q \approx_o^l R$, we have R'' and R''' such that $R \Longrightarrow\xrightarrow{a(x)} R''$ and, for every y, $R''[y/x] \Longrightarrow R''' \approx_o^l Q''[y/x]$. Consequently $R''' \Longrightarrow R' \approx_o^l Q' \approx_o^l P'[y/x]$ for some R'. Now suppose $P \xrightarrow{\overline{a}(x)} P'$. Then by definition $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $R \stackrel{\overline{a}(x)}{\Longrightarrow} R'$ for some Q' and R' such that $(P')^{[x\notin(fn(P'+Q')\setminus\{x\})]} \approx_o^l (Q')^{[x\notin(fn(P'+Q')\setminus\{x\})]}$ and $(Q')^{[x\notin(fn(R'+Q')\setminus\{x\})]} \approx_o^l (R')^{[x\notin(fn(R'+Q')\setminus\{x\})]}$. By (i) of Lemma 55 one gets

$$(P')^{[x\notin(fn(P'+Q'+R')\setminus\{x\})]} \approx_o^l (Q')^{[x\notin(fn(P'+Q'+R')\setminus\{x\})]} \approx_o^l (R')^{[x\notin(fn(P'+Q'+R')\setminus\{x\})]}$$

It follows from (ii) of Lemma 55 that $(P')^{[x\notin(fn(P'+R')\setminus\{x\})]} \approx_o^l (R')^{[x\notin(fn(P'+R')\setminus\{x\})]}$.

The proofs of the other two equivalence relations are the same. □

It is easily checked that Lemma 47 and Lemma 48 also hold for $\approx_o^D$. This fact is used in the proof of the following theorem.

Theorem 57. *The three relations $\approx_o^l$, $\approx_o^e$ and $\approx_o^w$ coincide with $\approx_o^\emptyset$ on the set of π-processes without the mismatch operator.*

Proof. The π-calculus without the mismatch operator enjoys the property that if $P \xrightarrow{\lambda} P'$ then $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$ for each substitution σ. By definition $\approx_o^e$, $\approx_o^l$ and $\approx_o^w$ are closed under substitution. It is then easy to see that the three relations coincide with each other on the set of π-processes without the mismatch operator since it does not matter which instantiation scheme we use. We only have to show that $\approx_o^w$ also coincides with $\approx_o$ when restricted to these processes. Notice that $\approx_o^w$ is closed under parallel composition for processes without the mismatch operator.

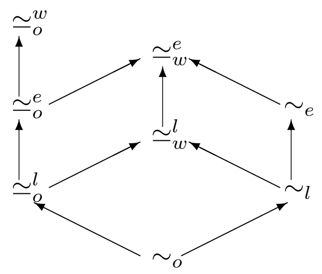

Figure 12: The Order Structure of the Eight Congruence Relations

(i) Let $\mathcal{R}$ be $\{(P^\delta, Q^\delta) \mid P \approx_o^D Q$ for some D such that D agrees with $\delta\} \cup \approx_o^w$. We prove that $\mathcal{R}$ is a weak open bisimulation up to $\approx_o^w$ on the set of π-processes without the mismatch. First we show that $\mathcal{R}$ is closed under substitution. Suppose $P^\delta \mathcal{R} Q^\delta$. If σ agrees with δ, it agrees with D as well. By Lemma 47, we have that $P\sigma \approx_o^{D\sigma} Q\sigma$. So $(P\sigma)^{\delta\sigma} \mathcal{R} (Q\sigma)^{\delta\sigma}$. Otherwise σ does not agree with δ, in which case $(P^\delta)\sigma \approx_o^w \mathbf{0} \approx_o^w (Q^\delta)\sigma$. Now consider the actions of P^δ. If $P^\delta \xrightarrow{\lambda} P''$ then, by Lemma 39, we have $P \xrightarrow{\lambda} P'$ and $P'' \equiv P'^\delta$. There are two cases:

- If λ is not a restricted output action then it follows from $P \approx_o^D Q$ that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q' \approx_o^D P'$ for some Q' and, by Lemma 39, $Q^\delta \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'^\delta \mathcal{R} P'^\delta$.
- If λ is a restricted output action $\overline{a}(x)$ then it follows from $P \approx_o^D Q$ that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q' \approx_{D'} P'$, where D' is $D \cup \{x\} \times fn(P+Q)$. Therefore $Q^\delta \stackrel{\overline{a}(x)}{\Longrightarrow} Q'^\delta$. Using the facts that $P' \approx_o^{D'} Q'$ and that $(fn(P'+Q') \setminus \{x\}) \subseteq fn(P+Q)$, one obtains that $P' \approx_o^{D''} Q'$ where D'' is $D \cup \{x\} \times (fn(P'+Q') \setminus \{x\}) \cup (fn(P'+Q') \setminus \{x\}) \times \{x\}$. Since D'' agrees with $\delta[x \notin (fn(P'+Q') \setminus \{x\})]$, one gets that $P'^{\delta[x \notin (fn(P'+Q') \setminus \{x\})]} \mathcal{R} Q'^{\delta[x \notin (fn(P'+Q') \setminus \{x\})]}$.

If we let D be the empty set then $P \approx_o Q$ implies $P \approx_o^w Q$ for P and Q without the mismatch operator.

(ii) Now we show that $\approx_o^w \subseteq \approx_o^\emptyset$. The proof is similar to (i). For a distinction D, define $\mathcal{R}^D$ to be

$$\{(P, Q) \mid P^\delta \approx_o^w Q^\delta,\ D \text{ agrees with } \delta\}$$

and let $\mathcal{R}$ be $\{\mathcal{R}^D\}_{D \in \mathcal{D}}$. Then $\mathcal{R}$ is a weak open bisimulation in the π-calculus without the mismatch operator. By letting D be the empty set one gets that $P \approx_o^w Q$ implies $P \approx_o^\emptyset Q$.

This finishes the proof. □

4.1.5 Weak Congruence Revisited

It is clear that $\approx_o^e$ is not a congruence relation. It is preserved neither by the summation operator nor by the mismatch operator. For instance

$$[x \neq y]\overline{a}a \not\approx_o^e [x \neq y]\tau.\overline{a}a$$

This is because the right hand can perform a tau action and become $\overline{a}a$. But $\overline{a}a$ is not early open bisimilar to $[x \neq y]\overline{a}a$. The same can be said about the other two weak open bisimilarities. We will write $\simeq_w^l$ and $\simeq_w^e$ for the congruence relations defined respectively from $\dot{\approx}_w^l$ and $\dot{\approx}_w^e$ in the manner of Definition 4. Similarly we will write $\simeq_o^l$, $\simeq_o^e$ and $\simeq_o^w$ for the congruence relations defined respectively from $\approx_o^l$, $\approx_o^e$ and $\approx_o^w$ in the manner of Definition 4.

The inclusion structure of the eight congruence relations is described in Figure 12, where an arrow indicates a strict inclusion.

For the full π-calculus with the parallel operator one can apply the well-known expansion law to transplant its algebraic theory to that of nondeterministic mobile processes with restricted output. The only exception is the weak open bisimilarity. However we can redefine the weak open congruence $\simeq_o^w$ as the largest bisimulation congruence relation contained in $\approx_o^w$. In light of the following proposition this does not give rise to a new congruence.

Proposition 58. *In the full π-calculus with the parallel composition operator, $\simeq_o^w$ coincides with $\simeq_o^e$.*

Proof. The proof of $\simeq_o^e \subseteq \simeq_o^w$ is trivial.

Suppose $P \simeq_o^w Q$. We now prove that it is an open early bisimulation. We only consider the case of input action as the rest is easier. Suppose $P\sigma \xrightarrow{a(x)} P'$. Because $\simeq_o^w$ is a congruence relation, $P\sigma|\overline{a}y.\overline{b}b \simeq_o^w Q\sigma|\overline{a}y.\overline{b}b$ for fresh names y and b. Now $P\sigma|\overline{a}y.b \xrightarrow{\tau} P'[y/x]|\overline{b}b$. This action must be matched up by $Q\sigma|\overline{a}y.\overline{b}b \Longrightarrow Q_1|\overline{a}y.\overline{b}b \xrightarrow{\tau} Q_2|\overline{b}b \Longrightarrow Q_3|\overline{b}b \simeq_o^w P[y/x]|\overline{b}b$, from which it is easy to see that $Q\sigma \Longrightarrow Q_1 \xrightarrow{a(x)} Q'$, $Q'[y/x] \equiv Q_2 \Longrightarrow Q_3$ and $Q_3 \simeq_o^w P[y/x]$. Consequently $\simeq_o^w$ is an open early bisimulation. Conclude that $\simeq_o^w \subseteq \simeq_o^e$. $\square$

In what follows we will refer to $\sim_s^l$, $\sim_s^e$, $\approx_w^l$, $\approx_w^e$ as the ground congruences and $\sim_o^s$, $\approx_o^l$, $\sim_o^e$, $\approx_o^w$ as the open congruences.

4.2 Laws of the Full Calculus

The laws of the full π-calculus can be divided into three groups: The first group consists of basic laws that hold for all the eight congruences. In addition to the laws given in Section 3 and Section 4, this group has laws about the mismatch operator. The second group contains those laws that are valid for some strong congruence but not all the strong congruences. These laws are particularly interesting. The third group collects all the tau laws used in this paper. Some of the tau laws are good for all the congruences. Others only hold for some of the congruences. The relationship among the laws is complex. This section tries to unveil part of the picture.

4.2.1 Basic Laws

In Figure 13 some basic laws for the full π-calculus are listed, together with all the equivalence and congruence rules, all of them can be found in [32] or [37]. The laws M4 and M5 deal with the mismatch operator. The law S5 also involves the mismatch operator. The rest of the laws have already been seen in Section 3.2. This set of rules and axioms will be referred to as AS. They are sound for all the observational equalities, for the full π-calculus, that we are aware of. Their intended meanings are obvious.

The axioms about the mismatch operator in Figure 13 enables us to derive some interesting equalities for π-processes. In Figure 14 some important derived laws are given. The proofs that these laws follow from AS can be found in [37]. To establish D5 for example, one shows that both $(x)[y{\neq}z]P$ and $[y{\neq}z](x)P$ are provably equal to $[y{\neq}z](x)[y{\neq}z]P$, using L6 and S5. Notice that D6 is derived from L7.

In [42] completeness theorem is proved for strong open bisimilarity on π-processes without the mismatch operator. A more streamlined approach is used in previous sections to establish complete result for weak open congruence of the same language. In the presence of mismatch operator, the method of [42] and previous sections should be modified. First of all the normal forms of the processes in the full π-calculus need be redefined.

Definition 59. Let V be a finite set of names. A process P is a complete normal form on V if it is of the form

$$\sum_{i\in I} \psi_i \lambda_i.P_i$$

such that $bn(\lambda_i) \cap V = \emptyset$, ψ_i is complete on V and P_i is a complete normal form on $V \cup bn(\lambda_i)$ for each $i \in I$.

Let $d(P)$ be the maximal number of nested prefix operators in P. The definition of $d(P)$ extends the one given previously by requiring that $d([x{\neq}y]P) \stackrel{\text{def}}{=} d(P)$.

Lemma 60. *For a process P and a finite set V of names such that $fn(P) \subseteq V$ there is a complete normal form Q on V such that $d(Q) \leq d(P)$ and $AS \vdash Q = P$.*

The proof of the above lemma is by simple induction. Use S5 if necessary to transfer outermost sequences of match and mismatch operators to complete ones on V.

We now derive some more equalities in AS.

Lemma 61. *Suppose that $y_1, \ldots, y_n$ are pairwise distinct names and Y is the set $\{y_1, \ldots, y_n\}$. Then $AS \vdash P = [x{=}y_1]P + \ldots + [x{=}y_n]P + [x{\notin}Y]P$.*

E1	P	$=$	P	
E2	Q	$=$	P	if $P = Q$
E3	P	$=$	R	if $P = Q = R$
C1	$\pi.P$	$=$	$\pi.Q$	if $P = Q$
C2	$[x{=}y]P$	$=$	$[x{=}y]Q$	if $P = Q$
C3	$[x{\neq}y]P$	$=$	$[x{\neq}y]Q$	if $P = Q$
C4	$P{+}R$	$=$	$Q{+}R$	if $P = Q$
C5	$(x)P$	$=$	$(x)Q$	if $P = Q$
L1	$(x)\mathbf{0}$	$=$	$\mathbf{0}$	
L2	$(x)(y)P$	$=$	$(y)(x)P$	
L3	$(x)(P{+}Q)$	$=$	$(x)P{+}(x)Q$	
L4	$(x)\pi.P$	$=$	$\pi.(x)P$	if $x \notin n(\pi)$
L5	$(x)\pi.P$	$=$	$\mathbf{0}$	if $x \in subj(\pi)$
L6	$(x)[y{=}z]P$	$=$	$[y{=}z](x)P$	if $x \notin \{y, z\}$
L7	$(x)[x{=}y]P$	$=$	$\mathbf{0}$	if $x \neq y$
M1	ϕP	$=$	ψP	if $\phi \Leftrightarrow \psi$
M2	$[x{=}y]P$	$=$	$[x{=}y]P[y/x]$	
M3	$[x{=}y](P{+}Q)$	$=$	$[x{=}y]P{+}[x{=}y]Q$	
M4	$[x{\neq}y](P{+}Q)$	$=$	$[x{\neq}y]P{+}[x{\neq}y]Q$	
M5	$[x{\neq}x]P$	$=$	$\mathbf{0}$	
S1	$P{+}\mathbf{0}$	$=$	P	
S2	$P{+}Q$	$=$	$Q{+}P$	
S3	$P{+}(Q{+}R)$	$=$	$(P{+}Q){+}R$	
S4	$[x{=}y]P{+}P$	$=$	P	
S5	$[x{=}y]P{+}[x{\neq}y]P$	$=$	P	

Figure 13: *AS*: Basic Rules and Axioms for the Full Pi Calculus

D1	$\psi P{+}P$	$=$	P	
D2	$[x{=}x]P$	$=$	P	
D3	$[x{=}y]\mathbf{0}$	$=$	$\mathbf{0}$	
D4	$[x{\neq}y]\mathbf{0}$	$=$	$\mathbf{0}$	
D5	$(x)[y{\neq}z]P$	$=$	$[y{\neq}z](x)P$	if $x \notin \{y, z\}$
D6	$(x)[x{\neq}y]P$	$=$	$(x)P$	if $x \neq y$
D7	$(x)P$	$=$	P	if $x \notin fn(P)$

Figure 14: Some Laws Derivable from *AS*

Proof. First of all notice that the following equality is valid by D1 and S5:

$$AS \vdash [x{=}y]P{+}[x{\neq}y][x{=}z]P = [x{=}y]P{+}[x{=}z]P \tag{6}$$

which can be derived as follows:

$$\begin{aligned} [x{=}y]P{+}[x{\neq}y][x{=}z]P &\overset{D1}{=} [x{=}y]P{+}[x{=}y][x{=}z]P{+}[x{\neq}y][x{=}z]P \\ &\overset{S5}{=} [x{=}y]P{+}[x{=}z]P \end{aligned}$$

L8	$(x)C[[x{=}y]P]$	$=$	$(x)C[\mathbf{0}]$	$x, y \notin bn(C[])$ and $x{\neq}y$
M6	$[x{\neq}y]\pi.P$	$=$	$[x{\neq}y]\pi.[x{\neq}y]P$	$\{x, y\} \cap bn(\pi) = \emptyset$
S6	$a(x).P{+}a(x).Q$	$=$	$a(x).P{+}a(x).Q{+}a(x).([x{=}y]P{+}[x{\neq}y]Q)$	

Figure 15: Additional Laws

L8a	$(x)C[[x{\neq}y]P]$	$=$	$(x)C[P]$	$x, y \notin bn(C[])$ and $x{\neq}y$
L8b	$(x)P^{[x\neq y]}$	$=$	$(x)P$	$x \notin bn(P)$ and $x{\neq}y$
M6a	$P^{[x\neq y]}$	$=$	$[x{\neq}y]P$	

Figure 16: Laws Equivalent to L8 or M6

When $n = 1$ the lemma is just S5. Suppose the lemma holds for $n > 1$. Then

$$\begin{array}{rcl} AS \vdash P & \stackrel{S5}{=} & [x{=}y_{n+1}]P{+}[x{\neq}y_{n+1}]P \\ & \stackrel{I.H.}{=} & [x{=}y_{n+1}]P{+}[x{\neq}y_{n+1}]([x{=}y_1]P{+}\ldots[x{=}y_n]P{+}[x{\notin}Y]P) \\ & \stackrel{M4}{=} & [x{=}y_{n+1}]P{+}[x{\neq}y_{n+1}][x{=}y_1]P{+}\ldots{+}[x{\neq}y_{n+1}][x{=}y_n]P{+}[x{\notin}Y']P \\ & \stackrel{(6)}{=} & [x{=}y_{n+1}]P{+}[x{=}y_1]P{+}\ldots{+}[x{\neq}y_{n+1}][x{=}y_n]P{+}[x{\notin}Y']P \\ & \vdots & \\ & \stackrel{(6)}{=} & [x{=}y_{n+1}]P{+}[x{=}y_1]P{+}\ldots{+}[x{=}y_n]P{+}[x{\notin}Y']P \\ & = & [x{=}y_1]P{+}\ldots{+}[x{=}y_n]P{+}[x{=}y_{n+1}]P{+}[x{\notin}Y']P \end{array}$$

where $Y' = \{y_1, \ldots, y_n, y_{n+1}\}$. □

The following generalization of Lemma 61 is also useful. For a proof of it, see Appendix A.2.

Lemma 62. *If $\bigvee_{i\in I} \psi_i$ is a tautology then $AS \vdash \sum_{i\in I} \psi_i P = P$.*

The next lemma explains why M5 is necessary.

Lemma 63. *If $\psi \Leftrightarrow \bot$ then $AS \vdash \psi P = \mathbf{0}$.*

Proof. Obviously $\psi \Leftrightarrow \bot$ if and only if $\psi \Leftrightarrow x{\neq}x$. We are done by using M5. □

4.2.2 Additional Laws

All complete systems studied in the rest of the paper extend AS. Each of these systems contains some of the axioms listed in Figure 15. The law S6 was first proposed in [37] and the law M6 was introduced in [32]. Notice that L8 and L1 imply L7.

In Figure 16 some equivalent formulations of L8 and M6 are given. The next lemma establishes the equivalence of L8, L8a and L8b on top of the system AS.

Lemma 64. *$AS \cup \{L8\} \vdash L8a$, $AS \cup \{L8a\} \vdash L8b$ and $AS \cup \{L8b\} \vdash L8$.*

Proof. The following inference shows that $AS \cup \{L8\} \vdash L8a$:

$$\begin{array}{rcl} (x)C[P] & = & (x)C[[x{=}y]P{+}[x{\neq}y]P] \\ & \stackrel{L8}{=} & (x)C[\mathbf{0}{+}[x{\neq}y]P] \\ & = & (x)C[[x{\neq}y]P] \end{array}$$

The fact $AS \cup \{L8a\} \vdash L8b$ is clear. And $AS \cup \{L8b\} \vdash L8$ is established easily as follows:

$$\begin{array}{rcl} (x)C[[x{=}y]P] & \stackrel{L8b}{=} & (x)C[[x{=}y]P]^{[x\neq y]} \\ & = & (x)C[\mathbf{0}]^{[x\neq y]} \\ & \stackrel{L8b}{=} & (x)C[\mathbf{0}] \end{array}$$

where the second equality holds by Lemma 63. □

L8 and L8a are formulated in terms of contexts while L8b has the virtue that it does not explicitly refer to the notion of contexts. On the other hand L8 has the advantage of not using mismatch operator and therefore is applicable to a wider range of calculi.

The next lemma relates M6 and M6a.

Lemma 65. *$AS \cup \{M6\} \vdash M6a$ and $AS \cup \{M6a\} \vdash M6$.*

Proof. Assume that $\{x,y\} \cap bn(\pi) = \emptyset$. Then M6a implies M6 because

$$\begin{aligned}
[x{\neq}y]\pi.P &\stackrel{M6a}{=} (\pi.P)^{[x\neq y]} \\
&\stackrel{M1}{=} (\pi.[x{\neq}y]P)^{[x\neq y]} \\
&\stackrel{M6a}{=} [x{\neq}y]\pi.[x{\neq}y]P
\end{aligned}$$

The converse is proved by structural induction. □

In view of the above two lemmas, we will use L8 to stand for any of the three laws L8, L8a, L8b and M6 for either of the two laws M6, M6a.

L8 is valid for the strong open bisimilarity whereas M6 holds for the strong early/late bisimilarity. It should come as no surprise that the latter implies the former in AS.

Lemma 66. *$AS \cup \{M6\} \vdash L8$.*

Proof. $AS \vdash (x)P^{[x\neq y]} \stackrel{M6}{=} (x)[x{\neq}y]P \stackrel{D6}{=} (x)P$. □

The law S6 can not be deduced from the law S5. The latter is valid for all the congruences whereas the former is invalid for the open congruences.

4.2.3 Tau Laws

In order to extend AS to complete systems for the weak observational congruences of the full π-calculus, we need the tau laws in Figure 17. There are altogether six tau laws. The first two are familiar. The rest four need some explanation:

- The T3 in Figure 17 is a generalization of the T3 in Figure 7, taking into account of the mismatch operator. The general T3 is equivalent to the restricted T3 in the presence of M6. But in general the former strictly subsumes the latter because the equality $\pi.(P{+}[x{\neq}y]\tau.Q) = \pi.(P{+}[x{\neq}y]\tau.Q) + \pi.Q$, where $bn(\pi) \cap \{x,y\} = \emptyset$, is not implied by the restricted T3.
- The T4 in Figure 17 is a generalization of the T4 in Figure 8, again taking into account of the mismatch operator. This law is subsumed by M6.
- T5 is a law valid for the early open congruence. The following is a simple instance of the law:

  $$a(x).(P_0{+}[x{=}y]\tau.Q) + a(x).(P_1{+}[x{\neq}y]\tau.Q) = a(x).(P_0{+}[x{=}y]\tau.Q) + a(x).(P_1{+}[x{\neq}y]\tau.Q) + a(x).Q$$

  More generally we have the following binary version, as it were, of T5:

  $$a(x).(P_0 + \psi_0\tau.Q) + a(x).(P_1 + \psi_1\tau.Q) = a(x).(P_0 + \psi_0\tau.Q) + a(x).(P_1 + \psi_1\tau.Q) + \psi a(x).Q$$

  where $\psi_0 \vee \psi_1 \Leftrightarrow \psi$ and $x \notin n(\psi)$. This binary law does not seem to imply T5. All our attempts to simplify the law have failed.
- T6 holds for the weak open congruence. It implies T5. Notice that this law makes sense only in a late operational semantics. It has far less significance than the other five tau laws.

The overloading of the name T3, respectively T4, is not a problem since a same law takes different forms in different subcalculi.

In Figure 18 some derived laws of T3, T4 and T5 are presented. The joint effect of T3a and T3b is the same as that of T3. Similarly T4a and T4b together have the same power as T4. T5a is a more manageable version of T5, which is more useful in practice. It is simpler yet equivalent to T5.

We now formalize some of the above claims in the rest of this section.

T1	$\pi.\tau.P$	$=$	$\pi.P$	
T2	$P+\tau.P$	$=$	$\tau.P$	
T3	$\pi.(P+\psi\tau.Q)$	$=$	$\pi.(P+\psi\tau.Q)+\psi\pi.Q$	$bn(\pi)\cap n(\psi)=\emptyset$
T4	$\tau.P$	$=$	$\tau.(P+\psi\tau.P)$	
T5	$\sum_{i\in I} a(x).(P_i+\psi_i\tau.Q)$	$=$	$\sum_{i\in I} a(x).(P_i+\psi_i\tau.Q)+\psi a(x).Q$	$\bigvee_{i\in I}\psi_i \Leftrightarrow \psi,\ x\notin n(\psi)$
T6	$a(x).(P+\delta\tau.Q)$	$=$	$a(x).(P+\delta\tau.Q)+\delta_{\backslash x} a(x).Q$	

Figure 17: Tau Laws for the Full Pi Calculus

T3a	$\overline{a}x.(P+\delta\tau.Q)$	$=$	$\overline{a}x.(P+\delta\tau.Q)+\delta\overline{a}x.Q$	
T3b	$a(x).(P+\delta\tau.Q)$	$=$	$a(x).(P+\delta\tau.Q)+\delta a(x).Q$	$x\notin n(\delta)$
T4a	$\tau.P$	$=$	$\tau.(P+[x{=}y]\tau.P)$	
T4b	$\tau.P$	$=$	$\tau.(P+\delta\tau.P)$	
T5a	$\sum(a,P,Q,\delta)$	$=$	$\sum(a,P,Q,\delta)+\delta a(x).Q$	$x\notin n(\delta)$
In T5a, $\sum(a,P,Q,\delta)$ is $\sum_{y\in Y} a(x).(P_y{+}\delta[x{=}y]\tau.Q){+}a(x).(P{+}\delta[x{\notin}Y]\tau.Q)$				

Figure 18: Derived Tau Laws

Lemma 67. *The following properties hold:*
(i) $AS\cup\{\pi.(P{+}\tau.Q)=\pi.(P{+}\tau.Q)+\pi.Q\}\cup\{M6\}\vdash T3a$.
(ii) $AS\cup\{\pi.(P{+}\tau.Q)=\pi.(P{+}\tau.Q)+\pi.Q\}\cup\{M6\}\vdash T3b$.
(iii) $AS\cup\{T1,T2\}\cup\{T3a,T3b\}\vdash T3$.

Proof. The proof of (i) can be given as follows:

$$\begin{array}{rcl}\pi.(P{+}\delta\tau.Q) & = & \pi.(P{+}\delta\tau.Q)+\delta\pi.(P{+}\delta\tau.Q)\\ & \stackrel{M6}{=} & \pi.(P{+}\delta\tau.Q)+\delta\pi.(P{+}\tau.Q)\\ & = & \pi.(P{+}\delta\tau.Q)+\delta\pi.(P{+}\tau.Q)+\delta\pi.Q\\ & = & \pi.(P{+}\delta\tau.Q)+\delta\pi.Q\end{array}$$

The proof of (ii) is similar. For (iii) notice that $\tau.(P{+}\tau.Q)=\tau.(P{+}\tau.Q)+\tau.Q$ follows from T1 and T2. Hence $\pi.(P{+}\delta\tau.Q)=\pi.(P{+}\delta\tau.Q)+\delta\pi.Q$ when $bn(\pi)\cap n(\delta)=\emptyset$. Suppose $n(\psi)\cap bn(\pi)=\emptyset$. By M1 one may assume that $\psi=\mu\delta$, where μ is the match part of ψ and δ is the mismatch part of ψ. When $\mu=[]$ the lemma is just T3a, T3b and T3c. When $\mu=[x{=}y]\mu'$, then

$$\begin{array}{rcl}\pi.(P{+}[x{=}y]\mu'\delta\tau.Q) & \stackrel{D1}{=} & [x{=}y]\pi.(P{+}[x{=}y]\mu'\delta\tau.Q){+}\pi.(P{+}[x{=}y]\mu'\delta\tau.Q)\\ & \stackrel{M1,M2}{=} & [x{=}y]\pi.(P{+}\mu'\delta\tau.Q){+}\pi.(P{+}[x{=}y]\mu'\delta\tau.Q)\\ & \stackrel{I.H.}{=} & [x{=}y](\pi.(P{+}\mu'\delta\tau.Q){+}\mu'\delta\pi.Q){+}\pi.(P{+}[x{=}y]\mu'\delta\tau.Q)\\ & \stackrel{M3}{=} & [x{=}y]\pi.(P{+}\mu'\delta\tau.Q){+}[x{=}y]\mu'\delta\pi.Q{+}\pi.(P{+}[x{=}y]\mu'\delta\tau.Q)\\ & \stackrel{D1}{=} & \pi.(P{+}[x{=}y]\mu'\delta\tau.Q){+}[x{=}y]\mu'\delta\pi.Q\end{array}$$

This completes the proof. □

Lemma 68. *The following properties hold:*
(i) $AS\cup\{T1,T2\}\cup\{M6\}\vdash T4$.
(ii) $AS\cup\{T4a,T4b\}\vdash T4$.

Proof. (i) The proof is by induction. The base case is taken care of by T1 and T2. For the inductive step observe that

$$\begin{array}{rcl}\tau.(P+[x{=}y]\psi\tau.P) & \stackrel{S5}{=} & [x{=}y]\tau.(P+[x{=}y]\psi\tau.P)+[x{\neq}y]\tau.(P+[x{=}y]\psi\tau.P)\\ & \stackrel{M1,M2}{=} & [x{=}y]\tau.(P+\psi\tau.P)+[x{\neq}y]\tau.(P+[x{=}y]\psi\tau.P)\\ & \stackrel{M6,M4}{=} & [x{=}y]\tau.(P+\psi\tau.P)+[x{\neq}y]\tau.([x{\neq}y]P+[x{\neq}y][x{=}y]\psi\tau.P)\\ & = & [x{=}y]\tau.(P+\psi\tau.P)+[x{\neq}y]\tau.[x{\neq}y]P\\ & \stackrel{M6}{=} & [x{=}y]\tau.(P+\psi\tau.P)+[x{\neq}y]\tau.P\\ & \stackrel{I.H.}{=} & [x{=}y]\tau.P+[x{\neq}y]\tau.P\\ & \stackrel{S5}{=} & \tau.P\end{array}$$

where the fourth equality is by Lemma 63. Similarly one can prove that $AS_w^l \vdash \tau.P = \tau.(P+[x{\neq}y]\psi\tau.P)$ whenever $AS_w^l \vdash \tau.P = \tau.(P+\psi\tau.P)$.

(ii) The proof is by induction. By M1 one can suppose that $\psi = \mu\delta$, where μ is the match part of ψ and δ is the mismatch part of ψ. When $\mu = []$ the lemma is simply T4b. When $\mu = [x{=}y]\mu'$, one has

$$\begin{array}{rcl}\tau.P & \stackrel{T4a}{=} & \tau.(P+[x{=}y]\tau.P)\\ & \stackrel{D1}{=} & \tau.(P+[x{=}y]\mu'\delta\tau.P+[x{=}y]\tau.P)\\ & \stackrel{I.H.}{=} & \tau.(P+[x{=}y]\mu'\delta\tau.P+[x{=}y]\tau.(P+\mu'\delta\tau.P))\\ & \stackrel{M1}{=} & \tau.(P+[x{=}y]\mu'\delta\tau.P+[x{=}y]\tau.(P+[x{=}y]\mu'\delta\tau.P))\\ & \stackrel{T4a}{=} & \tau.(P+[x{=}y]\mu'\delta\tau.P)\end{array}$$

This completes the proof. □

The Lemma 66, Lemma 67 and Lemma 68 explain why one hasn't seen L8, the general T3 and the general T4 in literature: They are all subsumed by M6. It is when one investigate bisimulation congruences which fail M6 that one discovers these laws. The next lemma is another example of using the power of M6, whose proof is placed in Appendix A.3.

Lemma 69. *$AS \cup \{M6\} \cup \{T1, T2, T3\} \vdash \tau.(P+\phi\tau.Q) = \tau.(P+\tau.(\phi\tau.Q+\neg\phi\tau.P))$.*

Next we discuss properties concerned with T5 and T6. The proof of the next lemma can be found in Appendix A.4.

Lemma 70. *In $AS \cup \{T1, T2, T3\}$ the law T5*

$$\sum_{i\in I} a(x).(P_i+\psi_i\tau.Q) = \sum_{i\in I} a(x).(P_i+\psi_i\tau.Q) + \psi a(x).Q$$

is equivalent to the law T5a

$$\begin{array}{rl} & \sum_{i=1}^{k} a(x).(P_i+\delta[x{=}y_i]\tau.Q)+a(x).(P+\delta[x{\notin}Y]\tau.Q)\\ = & \sum_{i=1}^{k} a(x).(P_i+\delta[x{=}y_i]\tau.Q)+a(x).(P+\delta[x{\notin}Y]\tau.Q)+\delta a(x).Q\end{array}$$

In the above two equalities, $\bigvee_{i\in I}\psi_i \Leftrightarrow \psi$, $x \notin n(\psi)$, $k \geq 1$, $Y = \{y_1, y_2, \ldots, y_k\}$ and $x \notin n(\delta)$.

The next lemma says that T6 is stronger than T5.

Lemma 71. *$AS \cup \{T1, T2, T3\} \cup \{T6\} \vdash T5$.*

Proof. It is easily seen that T6 implies T5a. We are done by using Lemma 70. □

Lemma 72. *Let ψ be a set of mismatch and match operators none of which contains x, δ is a set of mismatch operators each of which contains x. Then $AS \cup \{T1, T2, T3\} \cup \{T6\} \vdash a(x).(P+\psi\delta\tau.Q) = a(x).(P+\psi\delta\tau.Q)+\psi a(x).Q$.*

System	Axioms in Addition to AS		Congruence
AS_s^l	M6		strong late equivalence
AS_s^e	M6,S6		strong early equivalence
AS_w^l	M6	T1,T2,T3	weak late congruence
AS_w^e	M6,S6	T1,T2,T3	weak early congruence
AS_o^s	L8		strong open bisimilarity
AS_o^l	L8	T1,T2,T3,T4	late open congruence
AS_o^e	L8	T1,T2,T3,T4,T5	early open congruence
AS_o^w	L8	T1,T2,T3,T4,T6	weak open congruence

Figure 19: Systems for the Full Pi Calculus

Proof. Using induction on the numbers of match/mismatch operators in ψ, the proof is similar to that of Lemma 67. □

We have seen that S6 is the only law that separates the early congruence and the late congruence. We will see that T5 is the only law that distinguishes the early open congruence and the late open congruence. One would be interested to know the relationship between the two if there is any. The next lemma seems to be the best one can expect in view of the fact that S6 is not valid for the early open congruence.

Lemma 73. $AS \cup \{T1, T2, T3\} \cup \{S6\} \vdash T5$.

The proof of the above lemma is in Appendix A.5.

4.3 Completeness Results for the Full Calculus

Finally we come to the point where we can prove the completeness results for the full calculus. In Figure 19 eight systems are defined, where each system on the left column is obtained from the system AS by adding the axioms in the corresponding middle columns, with the corresponding congruence relation indicated on the right column. For instance AS_w^e, the system for the weak early congruence, is defined to be the system $AS \cup \{M6, S6\} \cup \{T1, T2, T3\}$.

By simply looking at Figure 19, one gets two obvious yet important messages:

- When moving from the systems for the ground congruences to the systems for the open congruences, M6 must be weakened to L8.
- When moving from the systems for the weak ground congruences to the systems for the weak open congruences, T4 must be added.

The thing we do not understand very well at the moment is what happens when moving from the systems for the late congruences to the systems for the early congruences. We will come back to it later.

In the next three subsections we will follow the standard approach to establish the completeness results of the eight systems.

4.3.1 Saturation Property

Saturation properties are about the relationship between operational semantics and axiomatic systems. A general statement goes like this: If $Q\sigma \overset{\lambda}{\Longrightarrow} Q'$ then $AS' \vdash Q = Q+\psi\lambda.Q'$ for the system AS' of interest. In the π-calculus without the mismatch operator, ψ must satisfy the condition that $\sigma(x) = \sigma(y)$ whenever $\psi \Rightarrow x{=}y$. In the presence of the mismatch operator a stronger relationship between σ and ψ is called for.

The first lemma deals with the saturation properties enjoyed by all the systems of the full π-calculus.

Lemma 74. *Suppose Q is a complete normal form on some $V \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution that is induced by ψ. Then the following properties hold:*
(i) If $Q\sigma \overset{\tau}{\Longrightarrow} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + \psi\tau.Q'$.
(ii) If $Q\sigma \overset{\overline{a}x}{\Longrightarrow} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + \psi\overline{a}x.Q'$.

(iii) If $Q\sigma \Longrightarrow\stackrel{\overline{a}(x)}{\longrightarrow} Q'$ *then* $AS \cup \{T1, T2, T3\} \vdash Q = Q + \psi\overline{a}(x).Q'$.
(iv) If $Q\sigma \Longrightarrow\stackrel{a(x)}{\longrightarrow} Q'$ *then* $AS \cup \{T1, T2, T3\} \vdash Q = Q + \psi a(x).Q'$.

Proof. To give a flavor of the proof, we take a look at a special instance of (iii). Suppose $Q\sigma \stackrel{\tau}{\longrightarrow}\stackrel{\tau}{\longrightarrow}\stackrel{\overline{a}(x)}{\longrightarrow} Q'$. By Lemma 34, we may assume that $Q\sigma \stackrel{\tau}{\longrightarrow} Q_1\sigma \stackrel{\tau}{\longrightarrow} Q_2\sigma \stackrel{\overline{a}(x)}{\longrightarrow} Q' \equiv Q_3\sigma$. As Q is a complete normal form, there must be some summand $\phi\tau.Q_1$ of Q such that $\psi \Rightarrow \phi$. By definition ϕ is complete on V. So $\phi \Leftrightarrow \psi$ by Lemma 38. It follows that

$$AS \cup \{T1, T2, T3\} \vdash Q \stackrel{M1}{=} Q + \psi\tau.Q_1 \stackrel{M2}{=} Q + \psi\tau.Q_1\sigma$$

Similarly it follows from $Q_1\sigma \stackrel{\tau}{\longrightarrow} Q_2\sigma$ that

$$AS \cup \{T1, T2, T3\} \vdash Q_1 = Q_1 + \psi\tau.Q_2 = Q_1 + \psi\tau.Q_2\sigma$$

For the same reason, $Q_2\sigma \stackrel{\overline{a}(x)}{\longrightarrow} Q'$ implies that

$$AS \cup \{T1, T2, T3\} \vdash Q_2 = Q_2 + \psi\overline{a}(x).Q'$$

Putting all these together one has

$$\begin{array}{rcl} AS \cup \{T1, T2, T3\} \vdash Q & = & Q + \psi\tau.Q_1 \\ & = & Q + \psi\tau.(Q_1 + \psi\tau.Q_2) \\ & \stackrel{T2}{=} & Q + \psi(\tau.(Q_1 + \psi\tau.Q_2) + (Q_1 + \psi\tau.Q_2)) \\ & \stackrel{M3,M4}{=} & Q + \psi\psi\tau.Q_2 \\ & \stackrel{M1}{=} & Q + \psi\tau.Q_2 \\ & = & Q + \psi\tau.(Q_2 + \psi\overline{a}(x).Q') \\ & = & Q + \psi\overline{a}(x).Q' \end{array}$$

This treatment of a special case should be enough to convey the general idea of the proof. □

The ψ appeared in the above lemma must be complete on V. Otherwise the lemma would be false. Here is a counter example: One has

$$([x{=}y][x{\neq}z]\tau.P)[y/x] \stackrel{\tau}{\longrightarrow} P$$

but not

$$AS \cup \{T1, T2, T3\} \vdash [x{=}y][x{\neq}z]\tau.P = [x{=}y][x{\neq}z]\tau.P{+}[x{=}y]\tau.P$$

The action $([x{=}y][x{\neq}z]\tau.P{+}[x{=}y]\tau.P)[y/x, y/z] \stackrel{\tau}{\longrightarrow} P[y/x, y/z]$ can not be matched up by any tau action from $([x{=}y][x{\neq}z]\tau.P)[y/x, y/z]$.

Lemma 75. *Suppose* Q *is a complete normal form on some* $V \supseteq fn(Q)$, ψ *is complete on* V, *and* σ *is a substitution induced by* ψ. *If* $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ *then the equality* $Q = Q + \psi\overline{a}(x).Q'$ *is provable in* AS^l_w *and* AS^l_o.

Proof. Suppose $Q\sigma \Longrightarrow\stackrel{\overline{a}(x)}{\longrightarrow} Q''\sigma \stackrel{\tau}{\Longrightarrow} Q'$. By Lemma 74, $AS \cup \{T1, T2, T3\} \vdash Q = Q + \psi\overline{a}(x).Q''$. It is clear that $fn(Q'') \subseteq fn(Q) \cup \{x\}$, $[x{\notin}V]\psi$ is complete on $V \cup \{x\} \supseteq fn(Q'')$ and σ is induced by $[x{\notin}V]\psi$. If $Q''\sigma \equiv Q'$ we are done. Otherwise

$$AS \cup \{T1, T2, T3a\} \vdash Q'' = Q'' + [x{\notin}V]\psi\tau.Q'$$

Therefore

$$\begin{array}{rcl}
Q & = & Q + \psi\overline{a}(x).Q'' \\
& = & Q + \psi\overline{a}(x).(Q'' + [x\notin V]\psi\tau.Q') \\
& \stackrel{L8}{=} & Q + \psi\overline{a}(x).(Q'' + [x\notin V]\psi\tau.Q')^{[x\notin V]} \\
& \stackrel{M1}{=} & Q + \psi\overline{a}(x).(Q'' + \psi\tau.Q')^{[x\notin V]} \\
& \stackrel{L8}{=} & Q + \psi\overline{a}(x).(Q'' + \psi\tau.Q') \\
& \stackrel{T3}{=} & Q + \psi(\overline{a}(x).(Q'' + \psi\tau.Q') + \psi\overline{a}(x).Q') \\
& \stackrel{M1}{=} & Q + \psi\overline{a}(x).(Q'' + \psi\tau.Q') + \psi\overline{a}(x).Q' \\
& = & Q + \psi\overline{a}(x).Q'
\end{array}$$

The use L8 is justified by Lemma 66. □

Corollary 76. *Suppose Q is a complete normal form on some $V \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution induced by ψ. If $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ then the equality $Q = Q + \psi\overline{a}(x).Q'$ is provable in AS^e_w, AS^e_o and AS^w_o.*

It remains to show the saturation property for input actions. This will be stated in the next four lemmas, three of which are proved in Appendix A.6, Appendix A.7 and Appendix A.8 respectively.

Lemma 77. *Suppose Q is a complete normal form on some $V = \{y_1, \ldots, y_k\} \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution that is induced by ψ. Then the following saturation properties hold: If*

$$\begin{array}{l}
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'\sigma, \\
Q'\sigma[y_1/x] \Longrightarrow Q_1, \\
Q'\sigma[y_2/x] \Longrightarrow Q_2, \\
\vdots \\
Q'\sigma[y_k/x] \Longrightarrow Q_k, \\
Q'\sigma \Longrightarrow Q_{k+1}
\end{array}$$

then $Q = Q + \psi a(x).(\tau.Q' + \psi \sum_{j=1}^{k}[x{=}y_j]\tau.Q_j + \psi[x\notin V]\tau.Q_{k+1}))$ is provable in $AS \cup \{T1, T2, T3\}$.

Lemma 78. *Suppose Q is a complete normal form on some $V = \{y_1, \ldots, y_k\} \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution that is induced by ψ. If*

$$\begin{array}{l}
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_1\sigma,\ Q'_1\sigma[y_1/x] \Longrightarrow Q_1, \\
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_2\sigma,\ Q'_2\sigma[y_2/x] \Longrightarrow Q_2, \\
\vdots \\
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_k\sigma,\ Q'_k\sigma[y_k/x] \Longrightarrow Q_k, \\
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_{k+1}\sigma \Longrightarrow Q_{k+1}
\end{array}$$

then $Q + \psi \sum_{j=1}^{k} a(x).(\tau.Q'_j + \psi[x{=}y_j]\tau.Q_j) + \psi a(x).(\tau.Q'_{k+1} + \psi[x\notin V]\tau.Q_{k+1}))$ is provably equal to Q in $AS \cup \{T1, T2, T3\}$.

Proof. The proof is almost the same as that of Lemma 77. □

Lemma 79. *Suppose Q is a complete normal form on some $V = \{y_1, \ldots, y_k\} \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution that is induced by ψ. If*

$$\begin{array}{l}
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_1\sigma,\ Q'_1\sigma[y_1/x] \Longrightarrow Q_1, \\
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_2\sigma,\ Q'_2\sigma[y_2/x] \Longrightarrow Q_2, \\
\vdots \\
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_k\sigma,\ Q'_k\sigma[y_k/x] \Longrightarrow Q_k, \\
Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_{k+1}\sigma \Longrightarrow Q_{k+1}
\end{array}$$

then $AS^e_w \vdash Q = Q + \psi a(x).([x{=}y_1]\tau.Q_1 + \ldots + [x{\neq}y_1]\ldots[x{\neq}y_{k-1}][x{=}y_k]\tau.Q_k + [x{\notin}V]\tau.Q_{k+1})$.

Lemma 80. *Suppose Q is a complete normal form on some $V = \{y_1, \ldots, y_k\} \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution that is induced by ψ. If $Q\sigma \stackrel{a(x)}{\Longrightarrow} Q'$ then $AS^w_o \vdash Q = Q + \psi a(x).Q'$.*

4.3.2 Promotion Lemma

As has been said before, the basic idea of promotion lemmas is to lift two bisimulation congruent processes P and Q to the equality $\tau.P = \tau.Q$ in sufficiently rich systems. In this section we establish promotion properties for the five weak congruences in the full π-calculus. These properties are stated in a single lemma since their proofs have a great deal of overlap.

Lemma 81 (promotion). *The following properties hold:*
(i) If $P \approx^l_w Q$ then $AS^l_w \vdash \tau.P = \tau.Q$.
(ii) If $P \approx^e_w Q$ then $AS^e_w \vdash \tau.P = \tau.Q$.
(iii) If $P \approx^l_o Q$ then $AS^l_o \vdash \tau.P = \tau.Q$.
(iv) If $P \approx^e_o Q$ then $AS^e_o \vdash \tau.P = \tau.Q$.
(v) If $P \approx^w_o Q$ then $AS^w_o \vdash \tau.P = \tau.Q$.

The proof of the above lemma is placed in Appendix A.9.

4.3.3 Completeness Theorem

We now arrive at the most important theorem of this paper. For the benefit of the readers, we also state below the completeness results for the strong early congruence and the strong late congruence.

Theorem 82 (completeness). *The following properties hold:*
(i) AS^l_s is sound and complete for $\sim_l$.
(ii) AS^e_s is sound and complete for $\sim_e$.
(iii) AS^l_w is sound and complete for $\simeq^l_w$.
(iv) AS^e_w is sound and complete for $\simeq^e_w$.
(v) AS^s_o is sound and complete for $\sim_o$.
(vi) AS^l_o is sound and complete for $\simeq^l_o$.
(vii) AS^e_o is sound and complete for $\simeq^e_o$.
(viii) AS^w_o is sound and complete for $\simeq^w_o$.

Proof. The soundness can be easily established. For the completeness part the proofs for all the weak cases are similar to each other. Here we only prove (vii). By Lemma 60 we may assume that both P and Q are complete normal forms on $V \stackrel{\text{def}}{=} fn(P{+}Q){=}\{y_1, \ldots, y_k\}$. Let P be

$$\sum_{i \in I} \phi_i \lambda_i.P_i$$

and Q be

$$\sum_{j \in J} \psi_j \lambda_j.Q_j$$

Suppose σ is induced by ϕ_i and $P\sigma \xrightarrow{\lambda_i\sigma} P_i\sigma$. There are several cases:

- λ_i is an output prefix or a tau prefix. From $P \simeq^e_o Q$ one obtains some Q' such that $Q\sigma \stackrel{\lambda_i\sigma}{\Longrightarrow} Q'$ and $P_i\sigma \approx^e_o Q'$. By Lemma 81, $AS^e_o \vdash \tau.P_i\sigma = \tau.Q'$. By Lemma 74,

$$\begin{aligned} AS^e_o \vdash Q &= Q + \phi_i\lambda_i\sigma.Q' \\ &= Q + \phi_i\lambda_i\sigma.\tau.Q' \\ &= Q + \phi_i\lambda_i\sigma.\tau.P_i\sigma \\ &= Q + \phi_i\lambda_i.\tau.P_i \\ &= Q + \phi_i\lambda_i.P_i \end{aligned}$$

- λ_i is a restricted output action $\overline{a}(x)$. Now $P \simeq_o^e Q$ implies that some Q' exists such that $Q\sigma \stackrel{\overline{a}\sigma(x)}{\Longrightarrow} Q'$ and $P_i\sigma^{[x\notin D]} \approx_o^e Q'^{[x\notin D]}$, where D is $(fn(P_i\sigma + Q') \setminus \{x\})$. By Lemma 81

$$AS_o^e \vdash \tau.P_i\sigma^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} = \tau.Q'^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]}$$

  By Lemma 75

$$\begin{array}{rcl} AS_o^e \vdash Q & = & Q + \phi_i\overline{a}\sigma(x).Q' \\ & \stackrel{L8}{=} & Q + \phi_i\overline{a}\sigma(x).\tau.Q'^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \\ & = & Q + \phi_i\overline{a}\sigma(x).\tau.P_i\sigma^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \\ & \stackrel{L8}{=} & Q + \phi_i\overline{a}\sigma(x).\tau.P_i\sigma \\ & = & Q + \phi_i\overline{a}(x).P_i \end{array}$$

- λ_i is an input action $a(x)$. It follows from $P \simeq_o^e Q$ that some y, Q'' and Q' exist such that $Q\sigma \Longrightarrow\stackrel{a\sigma(x)}{\longrightarrow} Q''$ and $Q''[y/x] \Longrightarrow Q' \approx_o^e P_i\sigma[y/x]$. Using a similar proof to that of Lemma 81 one can show that $AS_o^e \vdash Q = Q + \phi_i a(x).P_i$.

In summary $AS_o^e \vdash P{+}Q = Q$. Symmetrically $AS_o^e \vdash P{+}Q = P$. It follows that $AS_o^e \vdash P = Q$. □

4.4 An Alternative to the Law S6

We have indicated in Section 4.3 that the law S6 is a bit mysterious. It is used in the ground scenario to move from the late congruences to the early congruences. Yet it does not appear in any of the open congruences. In the studies of symbolic systems Lin has used the following alternative to S6:

$$\frac{C \rhd \sum_{i\in I} \tau.P_i = \sum_{j\in J} \tau.Q_j}{C \rhd \sum_{i\in I} a(x).P_i = \sum_{j\in J} a(x).Q_j}\text{E-Input}$$

He proved that the E-Input rule is enough to transfer the late systems to the early systems. Parrow has discussed the complete issue for the weak early congruence and the weak late congruence ([36]). His system for the weak late congruence is virtually the same as ours. On the other hand his system for the weak early congruence uses the non-symbolic version of the above rule:

$$\frac{\sum_{i\in I} \tau.P_i = \sum_{j\in J} \tau.Q_j}{\sum_{i\in I} a(x).P_i = \sum_{j\in J} a(x).Q_j}\text{E-Input'}$$

In this paper we use a slightly stronger version of E-Input', the rule E-Input":

$$\frac{\sum_{i\in I} \psi_i\tau.P_i = \sum_{j\in J} \psi_j\tau.Q_j}{\sum_{i\in I} \psi_i a(x).P_i = \sum_{j\in J} \psi_j a(x).Q_j}\text{E-Input''}$$

in which $x \notin n(\psi_i)$ for each $i \in I$ and $x \notin n(\psi_j)$ for each $j \in J$.

E-Input" clearly subsumes E-Input'. It is not clear under what condition the reverse implication holds. For the relationship between the E-Input rules and S6, Parrow has pointed out the following fact:

Lemma 83. *$AS \cup \{M6\} \cup \{E\text{-}Input'\} \vdash S6$.*

Consequently $AS \cup \{M6\} \cup \{E\text{-}Input''\} \vdash S6$. The following lemma reveals another interesting fact.

Lemma 84. *$AS \cup \{E\text{-}Input''\} \vdash T5$.*

Proof. We prove that $AS \cup \{E\text{-}Input''\} \vdash T5a$. Suppose Y is $\{y_1, \ldots, y_n\}$ and $x \notin n(\delta)$. Then

$$\begin{array}{rl} & \sum_{i=1}^{n} \tau.(P_i{+}\delta[x{=}y_i]\tau.Q) + \tau.(P{+}\delta[x{\notin}Y]\tau.Q) \\ \stackrel{T2}{=} & \sum_{i=1}^{n} \tau.(P_i{+}\delta[x{=}y_i]\tau.Q) + \tau.(P{+}\delta[x{\notin}Y]\tau.Q) + \sum_{i=1}^{n}(P_i{+}\delta[x{=}y_i]\tau.Q) + (P{+}\delta[x{\notin}Y]\tau.Q) \\ = & \sum_{i=1}^{n} \tau.(P_i{+}\delta[x{=}y_i]\tau.Q) + \tau.(P{+}\delta[x{\notin}Y]\tau.Q) + \sum_{i=1}^{n} \delta[x{=}y_i]\tau.Q + \delta[x{\notin}Y]\tau.Q \\ = & \sum_{i=1}^{n} \tau.(P_i{+}\delta[x{=}y_i]\tau.Q) + \tau.(P{+}\delta[x{\notin}Y]\tau.Q) + \delta\tau.Q \end{array}$$

where the third equality holds by Lemma 61. That is

$$\sum_{i=1}^{n} \tau.(P_i{+}\delta[x{=}y_i]\tau.Q) + \tau.(P{+}\delta[x{\notin}Y]\tau.Q) = \sum_{i=1}^{n} \tau.(P_i{+}\delta[x{=}y_i]\tau.Q) + \tau.(P{+}\delta[x{\notin}Y]\tau.Q) + \delta\tau.Q \quad (7)$$

By applying E-Input" to (7) we obtain T5a. □

The relationship between a rule and an axiomatic system is less simple than that between a law and an axiomatic system. To explain that we introduce two related notions.

Definition 85. Suppose $\mathcal{R}$ is a binary relation on the set of finite π-processes and R is the following rule

$$\frac{A_1 = B_1 \quad \ldots \quad A_n = B_n}{A = B}$$

R is *derivable* in $\mathcal{R}$ if $A\mathcal{R}B$ holds under the assumption that $A_1\mathcal{R}B_1, \ldots, A_n\mathcal{R}B_n$ hold. R is *admissible* in $\mathcal{R}$ if it is true that $A_1\mathcal{R}B_1, \ldots, A_n\mathcal{R}B_n$ hold then $A\mathcal{R}B$ holds.

Derivability is clearly stronger than admissibility. In terms of axiomatic systems for π-processes, the two properties can be defined as follows:

- R is *derivable* in AS if $AS \vdash A = B$ can be derived under the *assumption* that $AS \vdash A_1 = B_1, \ldots, AS \vdash A_n = B_n$ are derivable.
- R is *admissible* in AS if whenever $AS \vdash A_1 = B_1, \ldots, AS \vdash A_n = B_n$ *are* derivable then $AS \vdash A = B$ *is* derivable.

According to the definition, if R is derivable in AS then $AS \vdash A = B$ is provable as long as we assume that $AS \vdash A_1 = B_1, \ldots, AS \vdash A_n = B_n$ are all provable. It doesn't matter whether any of $AS \vdash A_1 = B_1, \ldots, AS \vdash A_n = B_n$ is actually provable or not. On the other hand, to show that R is admissible, we have to verify the cases in which all of $AS \vdash A_1 = B_1, \ldots, AS \vdash A_n = B_n$ are all provable. A good way to distinguish the two properties is that derivability is about proof systems while admissibility is about models.

Lemma 86. *The following properties hold:*
(i) E-Input" is not admissible in $\sim^l_s$.
(ii) E-Input" is derivable in $\sim^e_s$.
(iii) E-Input" is not admissible in $\simeq^l_w$.
(iv) E-Input" is derivable in $\simeq^e_w$.
(v) E-Input" is derivable in $\sim^s_o$.
(vi) E-Input" is not admissible in $\simeq^l_o$.
(vii) E-Input" is derivable in $\simeq^e_o$.
(viii) E-Input" is derivable in $\simeq^w_o$.

Proof. (i) through (iv) are well known. The proof of (v) is simpler than that of (vii) given below.

(vi) It is clear that $\tau.[x{=}y]\tau.P{+}\tau.[x{\neq}y]\tau.P \approx^l_o \tau.[x{=}y]\tau.P{+}\tau.[x{\neq}y]\tau.P{+}\tau.P$. It is also clear that $a(x).[x{=}y]\tau.P{+}a(x).[x{\neq}y]\tau.P \not\approx^l_o a(x).[x{=}y]\tau.P{+}a(x).[x{\neq}y]\tau.P{+}a(x).P$.

(vii) Suppose $\sum_{i\in I}\psi_i\tau.P_i \simeq^e_o \sum_{j\in J}\psi_j\tau.Q_j$ and $x \notin n(\psi_i\psi_j)$ for each $i \in I$ and each $j \in J$. Let σ be a substitution that validates ψ_i. Then

$$(\sum_{i\in I}\psi_i a(x).P_i)\sigma \stackrel{a\sigma(x)}{\longrightarrow} P_i\sigma$$

For each name y one has

$$(\sum_{i\in I}\psi_i\tau.P_i)\sigma[y/x] \stackrel{\tau}{\longrightarrow} P_i\sigma[y/x]$$

It follows that some Q' and $j \in J$ exist such that $\sigma[y/x]$ validates ψ_j, which is the same as saying that σ validates ψ_j since $x \notin n(\psi_j)$, and that

$$(\sum_{j\in J}\psi_j\tau.Q_j)\sigma[y/x] \stackrel{\tau}{\longrightarrow} Q_j\sigma[y/x] \Longrightarrow Q' \approx^e_o P_i\sigma[y/x]$$

System	Axioms in Addition to AS		Congruence
As_s^e	M6,E-Input”		strong early equivalence
As_w^e	M6,E-Input”	T1,T2,T3	weak early congruence
As_o^e	L8,E-Input”	T1,T2,T3,T4	early open congruence

Figure 20: Alternative Systems Using E-Input”

Therefore

$$(\sum_{j\in J}\psi_j\tau.Q_j)\sigma \xrightarrow{\tau} Q_j\sigma$$

and

$$Q_j\sigma[y/x] \Longrightarrow Q' \approx_o^e P_i\sigma[y/x]$$

What we have verified is that $\sum_{i\in I}\psi_i a(x).P_i \simeq_o^e \sum_{j\in J}\psi_j a(x).Q_j$.

(viii) The proof is simpler than that of (vii). □

Using Lemma 83 and Lemma 7, one can define systems slightly different from those in Figure 19. In Figure 20 three systems for the early congruences are defined using E-Input”.

Theorem 87. *The following completeness results hold:*
(i) As_s^e is sound and complete for $\sim_e$.
(ii) As_w^e is sound and complete for $\simeq_w^e$.
(iii) As_o^e is sound and complete for $\simeq_o^e$.

Proof. The completeness is given by Lemma 83, Lemma 84 and Theorem 82. The soundness is supported by Lemma 86. □

So adding E-Input” transfers systems for late congruences to systems for early congruences.

Corollary 88. *The following properties hold:*
(i) E-Input” is not admissible in AS_s^l.
(ii) E-Input” is derivable in AS_s^e.
(iii) E-Input” is not admissible in AS_w^l.
(iv) E-Input” is derivable in AS_w^e.
(v) E-Input” is not derivable in AS_o^s but is admissible in AS_o^s.
(vi) E-Input” is not derivable in AS_o^l but is admissible in AS_o^l.
(vii) E-Input” is derivable in AS_o^e.
(viii) E-Input” is derivable in AS_o^w.

The fundamental difference between S6 and E-Input” is that the former is an *equational law* whereas the latter is an *equational rule*. The two enjoy different properties: Suppose AS_0 and AS_1 are two systems on finite π-processes. Suppose further that $AS_1 \vdash P = Q$ whenever $AS_0 \vdash P = Q$.

- Then $A = B$ is a derived law of AS_1 if it is a (derived) law of AS_0. So an equational law persists through system extension.
- If R is derivable in AS_0 then R is *not* necessarily admissible in AS_1. So an equational rule is not persistent through system extension.

We have seen the strict inclusions $AS_o^s \subseteq AS_o^l \subseteq AS_o^e$. The rule E-Input” is derivable in AS_o^s. It is not admissible in AS_s^l. It is however derivable in AS_s^e.

Persistence property is a good thing to have. This is one of the reasons that a law is preferred to a rule. But there is something that E-Input” can offer: Its restriction to the π-calculus without the mismatch operator is straightforward.

$\neq$	**early congruence**	**late congruence**	**open congruence**
strong	Parrow, Sangiorgi	Parrow, Sangiorgi	Fu, Yang
weak	Parrow, Fu, Yang	Parrow, Fu, Yang	Fu, Yang

Figure 21: Axiomatization with Mismatch, Revisited

$=$	**early congruence**	**late congruence**	**open congruence**
strong	?	?	Sangiorgi
weak	?	?	Fu

Figure 22: Axiomatization without Mismatch, Revisited

5 Comment and Related Issue

We have not been able to say anything about the axiomatizations of the four congruences of the π-calculus without the mismatch operator. But we have come up with some results on open congruences. Figure 21 and Figure 22 summarize the achievements of this paper. We have proposed several new tau laws and constructed complete systems for the three open congruence relations using these tau laws. We believe that we have found out the correct definitions of weak open congruence for mobile processes. The overlook of this issue in previous work is probably due to the fact that the three open bisimilarities coincide for the π-calculus without the mismatch operator. In our view the results of this paper are important because we believe that the mismatch operator has a very important role in the algebraic theory of mobile processes. This does not necessary mean that the operator is indispensable. But the tradeoff against it is probably not worthwhile in most cases. Take the case of testing equivalence for instance. In [3] the authors argued that, in their view, testing equivalence would be rather unfamiliar in the absence of the mismatch operator.

It came as a little surprise that the open semantics gives rise to two different bisimulation congruences. The discrepancy between the early semantics and the late semantics, in both the open case and the ground case, is due to the timing of instantiation. An early instantiation replaces the abstract name by another name at the time the input action happens while a late instantiation does the same after the input action has committed itself. In the absence of the mismatch operator, an early instantiation by a fresh name achieve the same effect as a late instantiation. Therefore the mismatch operator is the reason of there being more than one open bisimilarities. Having realized this one is tempted to think that the early and late equivalences might be the same had one removed the match operator. This is actually false. The following is a counter example: Let A be $a(x).(x(z).\overline{y}y+\overline{y}y.x(z)) + a(x).(x(z).\overline{y}y+\overline{y}y.x(z) + \tau)$. Then

$$A + a(x).(x(z)|\overline{y}y) \sim_e A$$

but

$$A + a(x).(x(z)|\overline{y}y) \not\sim_l A$$

since the action $A + a(x).(x(z)|\overline{y}y) \xrightarrow{a(x)} x(z)|\overline{y}y$ is matched up by neither $A \xrightarrow{a(x)} x(z).\overline{y}y+\overline{y}y.x(z)$ nor $A \xrightarrow{a(x)} x(z).\overline{y}y+\overline{y}y.x(z) + \tau$. More generally

$$a(x).P + \sum_{y\in fn(P)} a(x).P_y \sim_e \sum_{y\in fn(P)} a(x).P_y$$

but

$$a(x).P + \sum_{y\in fn(P)} a(x).P_y \not\sim_l \sum_{y\in fn(P)} a(x).P_y$$

where, for each $y \in fn(P)$, $P_y \dot{\sim}_e P[y/x]$ but not $P_y \dot{\sim}_l P[y/x]$. So for $\sim_l$ and $\sim_e$ to coincide at all one also has to get rid of the choice operator.

One might ask the question of why should we need two more bisimulation congruences, the early open congruence and the late open congruence, given that we already have several bisimulation congruences. This is of course not a good question. The result of this paper brings out a fuller picture of the open semantics than the previous work has done. Sangiorgi's definition of open bisimilarity turns out to be a special case of our definitions. Two better questions are: Why should we need the open semantics? Or is the mismatch operator good? We are not going to reinforce our argument for the mismatch operator apart from pointing out that the theoretical investigation of the weak early and late congruences use the mismatch operator in an essential way. We will answer the question about the open semantics by reaffirming our view about it: The early (late) *open* congruence has a better claim to be *the* bisimulation congruence than the early (late) congruence. As we have mentioned in the paper, two processes are bisimilar if they can not be told apart when putting in dynamic contexts. A dynamic context may well replace a free name in the processes by another name. For calculi of mobile processes, closure under substitution is a defining property rather than a derived property for *bisimulation* based equivalences. This view is supported by some theoretical results. In [2] it is shown that the CCS-like bisimilarity, as well as the barbed bisimilarity, for the asynchronous π-calculus are closed under substitution! One of the open problems raised in the introduction is also about this phenomenon. These can be interpreted as saying that the defining property of closure under substitution can be dropped in some special cases. In [9, 10, 12, 13, 14, 18, 19] a simplified yet equally powerful calculus of mobile processes, the χ-calculus, is proposed and investigated. The language has two motivations. One is to give a proof theoretical interpretation of process ([11]) and the other is to simplify the π-calculus. Many bisimilarities for the χ-processes have been studied. These include the barbed bisimilarity, the open bisimilarity, and more generally the L-bisimilarities. These equivalences are all required to be closed under substitution. Otherwise the resulting bisimilarities would not be even closed under composition, which definitely rule out any of these relations as observational equivalence. In other words, the observationality of the bisimilarities on χ-processes automatically imposes the condition of closure of substitution. The χ-calculus also makes it clear how dynamic environments can change the free names of the processes involved.

The symbolic approach to the open bisimilarities for the π-calculus without the mismatch operator has been studied by Li ([25]). But a correct treatment of the open bisimilarities on π-processes with the mismatch operator is not yet available. However this should be a formality in view of the results of this paper.

The paper would have looked a lot better had we removed the two major regrets of this paper:

- One regret is about the question marks in Figure 22. All our attempts to remove any of them have so far been unsuccessful. The failure makes one wonder if there exists a complete system for any of the four equivalences. However the message one gets from Lin's work using symbolic approach has to be that such systems do exist. The priority at the moment is to get one complete system for any of the four congruences, whatever the system may look like. Improvement of the system may come afterwards. A result like this might help to reveal some interesting relationship between the symbolic approach and the non-symbolic approach.

- The other is about the generalization of Lemma 11 to the full π-calculus. The main obstacle is posed by the restriction operator. We have not be able to find a solution to overcome the problem.

From the point of view of axiomatization, some important issues have not been discussed in this paper, which include the following:

- Axiomatization of infinite π-processes is theoretically possible for those processes with finite states. Milner initiated research in this direction for both the strong and the weak observational equivalences of CCS ([29, 30]). Lin has constructed complete systems for the strong and the weak congruences on finite control π-processes using a symbolic approach ([27, 28]). One might discover something new by investigating the axiomatization of finite control mobile processes in a non-symbolic setting.

- In practice equivalence relations are used just as often as congruence relations. Since weak early and late equivalences are not necessarily closed under substitution, their axiomatic systems trades off the rule

$$\frac{P = Q}{a(x).P = a(x).Q}$$

for the following rule

$$\frac{P[y/x] = Q[y/x] \text{ for every } y \in fn(P|Q)}{a(x).P = a(x).Q}$$

The treatment of the match and the mismatch operator is a lot more easier. There are only two laws:

$$[x{=}y]P \quad = \quad P \tag{8}$$
$$[x{\neq}y]P \quad = \quad \mathbf{0} \tag{9}$$

The details of the systems are not difficult to work out. See [32] for more on the subject.

- The testing equivalence for the π-calculus is quite different from the bisimulation equivalences. Boreale and De Nicola have defined in [3] the testing equivalence for the π-calculus. This equivalence is not a congruence. The testing congruence can be defined by the following: P and Q are testing congruent if and only if $P\sigma$ and $Q\sigma$ are testing equivalent for every substitution σ. In [16] a complete system for the testing congruence is given based upon Boreale and De Nicola's system. The π-calculus used in [3, 16] comes with the mismatch operator. In our opinion this is largely technical since we know nothing about how to axiomatize the testing equivalence/congruence without the mismatch. Although removing the mismatch operator introduces an equivalence relation that looks strange, testing without this operator is definitely interesting. But it appears that one should first solve the problems in Figure 22.
- Axiomatization of the barbed equivalence has never been an issue. Since in the π-calculus the strong barbed equivalence coincides with the strong early equivalence and their weak versions are most likely to be the same, there seems little point in studying barbed equivalence for the purpose of axiomatization. Recently Sangiorgi and Walker proposed what they call open barbed bisimilarity([43]). They showed that it is strictly between the barbed equivalence and the open bisimilarity. We haven't looket at the issue of axiomatization for the open barbed congruence.

There are many problems about the other aspects of π-calculus. Most of these problems are subtle. For the purpose of illustration let's take a closer look at one of the problems mentioned in the introduction. The strong ground bisimilarity is obviously too weak to be useful for the whole of π-calculus. The question is for which subcalculus of the π-calculus the strong ground bisimilarity $\sim_g$ is good. The first observation is that the subcalculus should contain neither the match operator nor the mismatch operator because in general $[x{=}y]P \sim_g \mathbf{0}$ does not imply $([x{=}y]P)[y/x] \sim_g \mathbf{0}[y/x]$. The subcalculus can not contain the replication operator either. An example that supports this claim can be given in CCS. Let P and Q be the processes defined below:

$$P \stackrel{\mathrm{def}}{=} (a)(!(u)(\overline{b}.\overline{c}.u|c.b.\overline{u}.a)|!\overline{a}.R)$$
$$Q \stackrel{\mathrm{def}}{=} (a)(!(u)\overline{b}.(\overline{c}.u|c.b.\overline{u}.a)|!(u)c.(b.\overline{u}.a|\overline{b}.\overline{c}.u)|!\overline{a}.R)$$

where R is $\overline{w}.\mathbf{0}$. Then one can easily verifies that $P|Q \sim_g Q$. But not $(P|Q)[c/b] \sim_g Q[c/b]$. This is because the action sequence

$$(P|Q)[c/b] \xrightarrow{\tau}\xrightarrow{\tau}\xrightarrow{\tau}\xrightarrow{\tau}\xrightarrow{\overline{w}} A,$$

for suitable A, can not be matched up by any action sequence from $Q[c/b]$. For $Q[c/b]$ to perform a $\overline{w}$-action it must perform at least five consecutive τ-actions first. So to be precise there are two questions to be asked:

- When confined to the subcalculus with none of the choice, match, mismatch and replication operators, is the strong ground bisimilarity closed under substitution?
- When confined to the subcalculus with none of the choice, match and mismatch operators, is the weak ground bisimilarity closed under substitution?

These questions are related to the question of when the early bisimilarity and the late bisimilarity coincide. But we are not going into that issue in this paper. For more about the usefulness of the ground bisimilarity we refer to [41].

Acknowledgement The authors are grateful to the two anonymous referees for their criticism and suggestions. The care they have given to the previous drafts of this paper and the extent they have gone into these drafts are truly remarkable. They have helped tremendously in improving both the content and the shape of the paper.

References

[1] R. Amadio and S. Prasad, Modelling IP Mobility, *CONCUR '98*, Lecture Notes in Computer Science 1466 (Springer, 1998) 301–316.

[2] R. Amadio, I. Castellani and D. Sangiorgi, On Bisimulations for the Asynchronous π-Calculus, *CONCUR '96*, Lecture Notes in Computer Science 1119 (Springer, 1996) 146–162.

[3] M. Boreale and R. De Nicola, Testing Equivalence for Mobile Processes. *Information and Computation*, **120** (1995) 279–303.

[4] M. Boreale and R. De Nicola. A Symbolic Semantics for the π-Calculus. *Information and Computation*, **126** (1996) 34–52.

[5] M. Boreale, On the Expressiveness of Internal Mobility in Name-Passing Calculi, *CONCUR '96*, Lecture Notes in Computer Science 1119 (Springer, 1996) 161–178.

[6] G. Boudol, Asynchrony and π-Calculus, Research Report 1702 (INRIA, Sophia-Antipolis, 1991).

[7] L. Cardelli and A. Gordon, Mobile Ambients, To Appear in *Theoretical Computer Science*.

[8] R. De Nicola and M. Hennessy, Testing Equivalence for Processes, *Theoretical Computer Science* **34** (1984) 83-133.

[9] Y. Fu, The χ-Calculus, *Proceedings of the International Conference on Advances in Parallel and Distributed Computing* (IEEE Computer Society Press, 1997) 74–81.

[10] Y. Fu, A Proof Theoretical Approach to Communications, *ICALP'97*, Lecture Notes in Computer Science 1256 (Springer, 1997) 325–335.

[11] Y. Fu, Reaction Graphs, *Journal of Computer Science and Technology*, **13** (1998) 510–530.

[12] Y. Fu, Bisimulation Lattice of Chi Processes, *ASIAN'98*, Lecture Notes in Computer Science 1538 (Springer, 1998) 245–262.

[13] Y. Fu, Variations on Mobile Processes, *Theoretical Computer Science*, **221** (1999) 327–368.

[14] Y. Fu, Open Bisimulations of Chi Processes, *CONCUR'99*, Lecture Notes in Computer Science 1664 (Springer, 1999) 304–319.

[15] Y. Fu, An Open Problem of Mobile Processes, Unpublished Paper, 1999.

[16] Y. Fu, Testing Congruence for Mobile Processes, *Journal of Computer Science and Technology*, **17** (2002).

[17] Y. Fu, A Schematic Axiom for Open Congruence, submitted.

[18] Y. Fu and Z. Yang, Chi Calculus with mismatch, *CONCUR 2000*, Lecture Notes in Computer Science 1877 (Springer, 2000) 596–610.

[19] Y. Fu and Z. Yang, The Ground Congruence for Chi Calculus, *FST&TCS 2000*, Lecture Notes in Computer Science 1974 (Springer, 2000).

[20] M. Hennessy and H. Lin, Symbolic Bisimulations, *Theoretical Computer Science*, **138** (1995) 353–369.

[21] M. Hennessy and R. Milner, Algebraic Laws for Nondeterminism and Concurrency, Journal of ACM, **67** (1985) 137–161.

[22] K. Honda and M. Tokoro, An Object Calculus for Asynchronous Communication, *ECOOP '91*, Lecture Notes in Computer Science 512 (Springer, 1991) 133–147.

[23] K. Honda and M. Tokoro, On Asynchronous Communication Semantics, *Object-Based Concurrent Computing*, Lecture Notes in Computer Science 612 (Springer, 1991) 21–51.

[24] K. Honda and N. Yoshida, On Reduction Based Process Semantics, *Theoretical Computer Science*, **152** (1995) 437–486.

[25] Z. Li, Theories and Algorithms for the Verification of Bisimulation Equivalences in Value-Passing CCS and the π-Calculus, PhD Thesis (Changsha Institute of Technology, Changsha, 1999).

[26] H. Lin, Complete Inference Systems for Weak Bisimulation Equivalences in the π-Calculus, *Proceedings of Sixth International Joint Conference on the Theory and Practice of Software Development*, Lecture Notes in Computer Science 915 (Springer, 1995) 187–201.

[27] H. Lin, Unique Fixpoint Induction for Mobile Processes, *CONCUR '95*, Lecture Notes in Computer Science 962 (Springer, 1995) 88–102.

[28] H. Lin, Complete Proof Systems for Observation Congruences in Finite-Control π-Calculus, *ICALP '98*, Lecture Notes in Computer Science 1443 (Springer, 1998) 443–454.

[29] R. Milner, A Complete Inference System for a Class of Regular Behaviours, *Journal of Computer and System Science*, **28** (1984) 439–466.

[30] R. Milner, A Complete Axiomatization System for Observational Congruence of Finite State Behaviours, *Information and Computation*, **81** (1989) 227-247.

[31] R. Milner, *Communication and Concurrency* (Prentice Hall, 1989).

[32] R. Milner, J. Parrow and D. Walker, A Calculus of Mobile Processes, *Information and Computation*, **100** (1992) 1–40 (Part I), 41–77 (Part II).

[33] R. Milner and D. Sangiorgi, Barbed Bisimulation, *ICALP '92*, Lecture Notes in Computer Science 623 (Springer, 1992) 685–695.

[34] M. Merro and D. Sangiorgi, On Asynchrony in Name-Passing Calculi, *ICALP'98*, Lecture Notes in Computer Science 1443 (Springer, 1998) 856–867.

[35] D. Park, Concurrency and Automata on Infinite Sequences, Lecture Notes in Computer Science 154 (Springer, 1981) 561–572.

[36] J. Parrow, On the Relationship Between Two Proof Systems for the Pi-Calculus, Note, 1999.

[37] J. Parrow and D. Sangiorgi, Algebraic Theories for Name-Passing Calculi, *Information and Computation*, **120** (1995) 174–197.

[38] J. Parrow and B. Victor, The Update Calculus, *AMAST '97*, Lecture Notes in Computer Science 1119 (Springer, 1997) 389–405.

[39] J. Parrow and B. Victor, The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes, *LICS '98* (IEEE Computer Society, 1998) 176–185.

[40] D. Sangiorgi, *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*, PhD thesis (Department of Computer Science, University of Edinburgh, 1993).

[41] D. Sangiorgi, π-Calculus, Internal Mobility and Agent-Passing Calculi, *Theoretical Computer Science*, **167** (1996) 235–274.

[42] D. Sangiorgi, A Theory of Bisimulation for π-Calculus, *Acta Informatica*, **3** (1996) 69–97.

[43] D. Sangiorgi and D. Walker, On Barbed Equivalence in π-Calculus, *CONCUR '01*, 2001.

[44] B. Thomsen, A Calculus of Higher Order Communicating Systems, *POPL '89* (ACM, 1989) 143–154.

[45] B. Thomsen, Plain CHOCS—A Second Generation Calculus for Higher Order Processes, *Acta Informatica*, **30** (1993) 1–59.

[46] B. Thomsen, A Theory of Higher Order Communicating Systems, *Information and Computation*, **116** (1995) 38–57.

A Proofs

A.1 The Proof of Lemma 55

Proof. We prove the lemma for the late open bisimilarity. The proofs for the other two are almost the same.

(i) Let $\mathcal{R}$ be $\{(P^\delta, Q^\delta) | P \approx_o^l Q\} \cup \approx_o^l$. We prove that $\mathcal{R}$ is a late open bisimulation. Suppose $P^\delta \mathcal{R} Q^\delta$. If σ does not respect δ then $P^\delta\sigma \approx_o^l \mathbf{0} \approx_o^l Q^\delta\sigma$, otherwise consider actions performed by $P^\delta\sigma$. There are several cases:

- If $P^\delta\sigma \xrightarrow{\lambda} P'$ and λ is an output action or a tau action then, by (ii) of Lemma 39, $P\sigma \xrightarrow{\lambda} P''$ for some P'' such that $P' \equiv (P'')^{(\delta\sigma)}$, bearing in mind that $P^\delta\sigma \equiv (P\sigma)^{\delta\sigma}$. It follows from $P \approx_o^l Q$ that $Q\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} Q' \approx_o^l P''$ for some Q'. Therefore $Q^\delta\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} (Q')^{(\delta\sigma)} \mathcal{R} P'$.
- If $P^\delta\sigma \xrightarrow{a(x)} P'$ then the proof is similar to the one in previous case.
- If $P^\delta\sigma \xrightarrow{\overline{a}(x)} P'$ then, by (ii) of Lemma 39, $P\sigma \xrightarrow{\overline{a}(x)} P''$ for some P'' such that $(P'')^{\delta\sigma} \equiv P'$. It follows from $P \approx_o^l Q$ that $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ for some Q' such that $(P'')^{[x\notin(fn(P''+Q')\setminus\{x\})]} \approx_o^l (Q')^{[x\notin(fn(P''+Q')\setminus\{x\})]}$. Thus $Q^\delta\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} (Q')^{\delta\sigma}$ and
$$P''^{(\delta\sigma)[x\notin(fn(P''+Q')\setminus\{x\})][x\notin n(\delta\sigma)]} \mathcal{R} Q'^{(\delta\sigma)[x\notin(fn(P''+Q')\setminus\{x\})][x\notin n(\delta\sigma)]}$$

This completes the proof of (i).

(ii) Let $\mathcal{R}$ be $\{(P, Q) \mid P^\delta \approx_o^l Q^\delta \text{ and } \forall x, y.(\delta \Rightarrow x \neq y) \Rightarrow (\{x, y\} \not\subseteq fn(P+Q))\}$. We show that $\mathcal{R}$ is a late open bisimulation. Consider the process $P\sigma$. If σ substitutes a name z in $fn(P+Q)$ for some name not in $fn(P+Q)$ Then $P\sigma \approx_o^l Q\sigma$ if and only if $P\sigma' \approx_o^l Q\sigma'$, where σ' differs from σ only in that the former maps z onto itself. So without loss of generality we may assume that σ maps names in $fn(P+Q)$ onto names in $fn(P+Q)$.

- $P\sigma \xrightarrow{\lambda} P'$ and λ is an output action or a tau action. Then $P^\delta\sigma \xrightarrow{\lambda} (P')^{\delta\sigma}$ by Lemma 34 and Lemma 39. So $Q^\delta\sigma \stackrel{\widehat{\lambda'}}{\Longrightarrow} Q'' \approx_o^l (P')^{\delta\sigma}$ for some Q''. By Lemma 39 some Q' exists such that $Q\sigma \stackrel{\widehat{\lambda'}}{\Longrightarrow} Q'$ and $(Q')^{\delta\sigma} \equiv Q''$. So $P'\mathcal{R}Q'$.
- $P\sigma \xrightarrow{a(x)} P'$. The proof is similar to the one in previous case.
- If $P\sigma \xrightarrow{\overline{a}(x)} P'$ then $P^\delta\sigma \equiv (P\sigma)^{\delta\sigma} \xrightarrow{\overline{a}(x)} P'' \equiv (P')^{\delta\sigma}$ for some P' and P''. Therefore $Q^\delta\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'' \approx_o^l (P')^{\delta\sigma}$ for some Q''. Then by Lemma 39, $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ for some Q' such that $(Q')^{\delta\sigma} \equiv Q''$. Now
$$(Q'')^{[x\notin(fn(P''+Q'')\setminus\{x\})]} \approx_o^l (P'')^{[x\notin(fn(P''+Q'')\setminus\{x\})]}$$
implies by (i) that
$$(Q')^{(\delta\sigma)[x\notin(fn(P'+Q')\setminus\{x\})][x\notin n(\delta\sigma)]} \approx_o^l (P')^{(\delta\sigma)[x\notin(fn(P'+Q')\setminus\{x\})][x\notin n(\delta\sigma)]}$$
It is easy to see that $(\delta\sigma)[x \notin n(\delta\sigma) \setminus fn(P' + Q')]$ satisfies the condition of $\mathcal{R}$. Hence
$$(Q')^{[x\notin(fn(P'+Q')\setminus\{x\})]} \mathcal{R} (P')^{[x\notin(fn(P'+Q')\setminus\{x\})]}$$

We have proved that $\mathcal{R}$ is a late open bisimulation. $\square$

A.2 The Proof of Lemma 62

Proof. Suppose $\bigvee_{i\in I} \psi_i$ is a tautology. We may assume that it contains at least two distinct names. Otherwise the result can be easily proved. Let $fn(\bigvee_{i\in I} \psi_i)$ be $\{x, y_1, \ldots, y_n\}$. We may assume that ψ_i is complete on $\{x, y_1, \ldots, y_n\}$, for each $i \in I$, because otherwise we could use the equivalence
$$x{=}y \vee x{\neq}y \Leftrightarrow \top$$

to expand it. Now consider

$$x{=}y_j \wedge (\bigvee_{i\in I} \psi_i) \Leftrightarrow x{=}y_j \tag{10}$$

where $j \in \{1, \ldots, n\}$. For each $i \in I$, either $\psi_i \Rightarrow x{\neq}y_j$ or $\psi_i \Rightarrow x{=}y_j$ by completeness. Because $\bigvee_{i\in I} \psi_i$ is a tautology, there must exist some $i \in I$ such that $\psi_i \Rightarrow x{\neq}y_j$. Consequently (10) is equivalent to

$$x{=}y_j \wedge (\bigvee_{i\in I'} \psi_i) \Leftrightarrow x{=}y_j \tag{11}$$

for some strict subset I' of I such that for each $i \in I'$ one has $\psi_i \Rightarrow x{=}y_j$. By substituting y_j for x in (11), one obtains

$$\bigvee_{i\in I'} \psi_i[y_j/x] \Leftrightarrow \top \tag{12}$$

Since $\bigvee_{i\in I'} \psi_i[y_j/x]$ contains strictly less free names than $\bigvee_{i\in I} \psi_i$, we could use the induction hypothesis to conclude that

$$\sum_{i\in I'} \psi_i[y_j/x]P = P$$

and therefore

$$[x{=}y_j] \sum_{i\in I} \psi_i P = [x{=}y_j]P \tag{13}$$

Let Y be $\{y_1, \ldots, y_n\}$. Next we consider

$$x{\notin}Y \wedge (\bigvee_{i\in I} \psi_i) \Leftrightarrow x{\notin}Y \tag{14}$$

It follows easily from (14) that

$$x{\notin}Y \wedge (\bigvee_{i\in I''} \psi'_i) \Leftrightarrow x{\notin}Y$$

for some I'' and ψ'_i, where $i \in I''$, such that the following conditions are satisfied:

- I'' is a strict subset of I;
- $x{\notin}Y \wedge \psi'_i \Leftrightarrow \psi_i$ for each $i \in I''$;
- $x \notin n(\bigvee_{i\in I''} \psi'_i)$.

It follows that

$$x{\notin}Y \Rightarrow \bigvee_{i\in I''} \psi''_i$$

from which it follows that $\bigvee_{i\in I''} \psi'_i$ is a tautology. So by induction hypothesis

$$\sum_{i\in I''} \psi'_i[x{\notin}Y]P = [x{\notin}Y]P \tag{15}$$

It follows easily from (15) that

$$[x{\notin}Y] \sum_{i\in I} \psi_i P = [x{\notin}Y]P \tag{16}$$

Uisng Lemma 61, (13) and (16), one gets

$$\begin{aligned} P &= [x{=}y_1]P + \ldots + [x{=}y_n]P + [x{\notin}Y]P \\ &= [x{=}y_1](\sum_{i\in I} \psi_i P) + \ldots + [x{=}y_n](\sum_{i\in I} \psi_i P) + [x{\notin}Y](\sum_{i\in I} \psi_i P) \\ &= \sum_{i\in I} \psi_i P \end{aligned}$$

This completes the proof. □

A.3 The Proof of Lemma 69

Proof. By Lemma 62 one has that $P = \phi P + \neg\phi P$. Intuitively let $\neg\phi$ be $\bigvee_{i\in I}\psi_i$, where each ψ_i, for $i \in I$, is either a match operator or a mismatch operator. Formally $\neg\phi P$ is $\sum_{i\in I}\psi_i P$. It is clear that $\psi_i\phi \Leftrightarrow \perp$ for each $i \in I$. It is also clear that $\neg\phi\pi.(P+\phi Q) \stackrel{M6}{=} \neg\phi\pi.P$ and $\phi\pi.(P+\neg\phi Q) \stackrel{M6}{=} \phi\pi.P$. Another important equality is

$$\neg\phi\pi.\neg\phi P = \neg\phi\pi.P \tag{17}$$

which can be justified as follows:

$$\begin{aligned}
\neg\phi\pi.\neg\phi P &= \sum_{i\in I}\psi_i\pi.\sum_{j\in I}\psi_j P \\
&= \sum_{i\in I}\psi_i\pi.(\psi_i P + \sum_{j\in I}^{j\neq i}\psi_j P) \\
&\stackrel{M6}{=} \sum_{i\in I}\psi_i\pi.(P + \sum_{j\in I}^{j\neq i}\psi_j P) \\
&\stackrel{D1}{=} \sum_{i\in I}\psi_i\pi.P \\
&= \neg\phi\pi.P
\end{aligned}$$

Therefore

$$\begin{aligned}
\tau.(P+\phi\tau.Q) &= \phi\tau.(P+\phi\tau.Q)+\neg\phi\tau.(P+\phi\tau.Q) \\
&= \phi\tau.(P+\tau.Q)+\neg\phi\tau.P
\end{aligned}$$

On the other hand

$$\begin{aligned}
\tau.(P+\tau.(\phi\tau.Q+\neg\phi\tau.P)) &= \phi\tau.(P+\tau.(\phi\tau.Q+\neg\phi\tau.P))+\neg\phi\tau.(P+\tau.(\phi\tau.Q+\neg\phi\tau.P)) \\
&\stackrel{(17)}{=} \phi\tau.(\phi P+\phi\tau.(\phi\tau.Q+\neg\phi\tau.P))+\neg\phi\tau.(\neg\phi P+\neg\phi\tau.(\phi\tau.Q+\neg\phi\tau.P)) \\
&= \phi\tau.(\phi P+\phi\tau.\phi\tau.Q)+\neg\phi\tau.(\neg\phi P+\neg\phi\tau.\neg\phi\tau.P) \\
&\stackrel{(17)}{=} \phi\tau.(P+\tau.Q)+\neg\phi\tau.(\neg\phi P+\neg\phi\tau.\tau.P) \\
&\stackrel{(17)}{=} \phi\tau.(P+\tau.Q)+\neg\phi\tau.(P+\tau.\tau.P) \\
&= \phi\tau.(P+\tau.Q)+\neg\phi\tau.P
\end{aligned}$$

This completes the proof. □

A.4 The Proof of Lemma 70

Proof. Using $T5a$ one can prove by induction on the number of the match operators in ϕ that

$$\begin{aligned}
&\sum_{i=1}^{k} a(x).(P_i+\phi[x{=}y_i]\tau.Q)+a(x).(P+\phi[x\notin Y]\tau.Q) \\
= &\sum_{i=1}^{k} a(x).(P_i+\phi[x{=}y_i]\tau.Q)+a(x).(P+\phi[x\notin Y]\tau.Q)+\phi a(x).Q
\end{aligned} \tag{18}$$

is provable in $AS\cup\{T1,T2,T3\}$, where $x \notin n(\phi)$.

Suppose $V = n(\cup_{i\in V}\psi_i)\backslash\{x\} = \{y_1,\ldots,y_n\}$. Let $\{\mathcal{R}_l\}$ be the set of all equivalence relations on V and $\{\phi_l\}$ the set of corresponding conditions that are complete on V. For each ϕ_l, either $\phi_l\psi \Leftrightarrow \phi_l$ or $\phi_l\psi \Leftrightarrow \perp$. Suppose $k \in \{1,\ldots,n\}$. If $\phi_l\psi \Leftrightarrow \perp$ then $\phi_l\psi[x{=}y_k]\tau.Q = \mathbf{0}$. If $\phi_l\psi \Leftrightarrow \phi_l$ then $\phi_l\psi[x{=}y_k]$ is complete on $V\cup\{x\}$. It follows from $\phi_l[x{=}y_k] \Leftrightarrow \phi_l\psi[x{=}y_k] \Leftrightarrow \bigvee_{i\in I}\phi_l\psi_i[x{=}y_k]$ that $\phi_l[x{=}y_k] \Leftrightarrow \phi_l\psi_{j_k}[x{=}y_k]$ for some $j_k \in I$. By using $D1$ and $M1$ one gets that

$$\begin{aligned}
AS\cup\{T1,T2,T3\} \vdash a(x).(P_{j_k}+\psi_{j_k}\tau.Q) &\stackrel{D1}{=} a(x).(P_{j_k}+\psi_{j_k}\tau.Q+\phi_l\psi_{j_k}[x{=}y_k]\tau.Q) \\
&\stackrel{M1}{=} a(x).(P_{j_k}+\psi_{j_k}\tau.Q+\phi_l\psi[x{=}y_k]\tau.Q) \\
&= a(x).(P'_{j_k}+\phi_l\psi[x{=}y_k]\tau.Q)
\end{aligned}$$

where P'_{j_k} is $P_{j_k} + \psi_{j_k}\tau.Q$. Similarly some $j \in I$ exists such that

$$AS \cup \{T1, T2, T3\} \cup \{T5a\} \vdash a(x).(P_j + \psi_j \tau.Q) = a(x).(P'_j + \phi_l \psi [x \notin V]\tau.Q)$$

where P'_j is $P_j + \psi_j \tau.Q$. So

$$\begin{aligned} & AS \cup \{T1, T2, T3\} \cup \{T5a\} \vdash \sum_{i\in I} a(x).(P_i + \psi_i \tau.Q) \\ = \;& \sum_{i\in I} a(x).(P_i + \psi_i \tau.Q) + \sum_{k=1}^{n} a(x).(P'_{j_k} + \phi_l \psi[x{=}y_k]\tau.Q) + a(x).(P'_j + \phi_l \psi [x{\notin}V]\tau.Q) \\ \overset{(18)}{=} \;& \sum_{i\in I} a(x).(P_i + \psi_i \tau.Q) + \phi_l \psi a(x).Q \end{aligned}$$

from which one gets

$$\begin{aligned} AS \cup \{T1, T2, T3\} \cup \{T5a\} \vdash \sum_{i\in I} a(x).(P_i + \psi_i \tau.Q) &= \sum_{i\in I} a(x).(P_i + \psi_i \tau.Q) + \sum_{l} \phi_l \psi a(x).Q \\ &= \sum_{i\in I} a(x).(P_i + \psi_i \tau.Q) + \psi a(x).Q \end{aligned}$$

By Lemma 62. □

A.5 The Proof of Lemma 73

Proof. We show that $AS \cup \{T1, T2, T3\} \cup \{S6\} \vdash T5a$. Now suppose $k \geq 1$, $Y = \{y_1, y_2, \ldots, y_k\}$ and $x \notin n(\delta)$. Then

$$\begin{aligned} & \sum_{i=1}^{k} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).(P{+}\delta[x{\notin}Y]\tau.Q) \\ \overset{S6}{=} \;& \sum_{i=1}^{k-1} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).([x{=}y_k](P_k{+}\delta[x{=}y_k]\tau.Q){+}[x{\neq}y_k](P{+}\delta[x{\notin}Y]\tau.Q)) \\ = \;& \sum_{i=1}^{k-1} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).(\ldots{+}\delta[x{=}y_k]\tau.Q{+}\delta[x{\notin}Y]\tau.Q) \\ \overset{S6}{=} \;& \sum_{i=1}^{k-2} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) \\ & {+}a(x).([x{=}y_{k-1}](P_{k-1}{+}\delta[x{=}y_{k-1}]\tau.Q){+}[x{\neq}y_{k-1}](\ldots{+}\delta[x{=}y_k]\tau.Q{+}\delta[x{\notin}Y]\tau.Q)) \\ = \;& \sum_{i=1}^{k-2} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).(\ldots{+}\delta[x{=}y_{k-1}]\tau.Q{+}\delta[x{\neq}y_{k-1}][x{=}y_k]\tau.Q{+}\delta[x{\notin}Y]\tau.Q) \\ \overset{(6)}{=} \;& \sum_{i=1}^{k-2} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).(\ldots{+}\delta[x{=}y_{k-1}]\tau.Q{+}\delta[x{=}y_k]\tau.Q{+}\delta[x{\notin}Y]\tau.Q) \end{aligned}$$

where (6) is the equality used in the proof of Lemma 61. Continuing in this way one gets

$$\begin{aligned} & \sum_{i=1}^{k} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).(P{+}\delta[x{\notin}Y]\tau.Q) \\ = \;& a(x).(\ldots{+}\delta[x{=}y_1]\tau.Q{+}\ldots{+}\delta[x{=}y_{k-1}]\tau.Q{+}\delta[x{=}y_k]\tau.Q{+}\delta[x{\notin}Y]\tau.Q) \\ = \;& a(x).(\ldots{+}\delta\tau.Q) \\ \overset{T3}{=} \;& a(x).(\ldots{+}\delta\tau.Q) + \delta a(x).Q \\ = \;& \sum_{i=1}^{k} a(x).(P_i{+}\delta[x{=}y_i]\tau.Q) + a(x).(P{+}\delta[x{\notin}Y]\tau.Q) + \delta a(x).Q \end{aligned}$$

This completes the proof. □

A.6 The Proof of Lemma 77

Proof. By the assumption of the lemma and Lemma 74

$$Q = Q + \psi a(x).Q'\sigma = Q + \psi a(x).\tau.Q'\sigma = Q + \psi a(x).\tau.Q'$$

For each $j \in \{1, \ldots, k\}$, if $Q'\sigma[y_j/x] \stackrel{\tau}{\Longrightarrow} Q_j$ then

$$\tau.Q' = \tau.(Q' + \psi[x{=}y_j]\tau.Q_j) = \tau.Q' + \psi[x{=}y_j]\tau.Q_j$$

otherwise

$$\tau.Q'\sigma[y_j/x] = \tau.Q_j = \tau.Q_j + \psi[x{=}y_j]\tau.Q_j$$

In the former case

$$Q = Q + \psi a(x).\tau.Q' = Q + \psi a(x).(\tau.Q' + \psi[x{=}y_j]\tau.Q_j)$$

In the latter case

$$\begin{array}{rcl} Q & = & Q + \psi a(x).\tau.Q'\sigma \\ & \stackrel{D1}{=} & Q + \psi a(x).(\tau.Q'\sigma + [x{=}y_j]\tau.Q'\sigma) \\ & \stackrel{M2}{=} & Q + \psi a(x).(\tau.Q'\sigma + [x{=}y_j]\tau.Q'\sigma[y_j/x]) \\ & = & Q + \psi a(x).(\tau.Q'\sigma + [x{=}y_j](\tau.Q_j + \psi[x{=}y_j]\tau.Q_j)) \\ & = & Q + \psi a(x).(\tau.Q'\sigma + \psi[x{=}y_j]\tau.Q_j) \\ & \stackrel{M2}{=} & Q + \psi a(x).(\tau.Q' + \psi[x{=}y_j]\tau.Q_j) \end{array}$$

So in either case one has

$$Q = Q + \psi a(x).(\tau.Q' + \psi[x{=}y_j]\tau.Q_j)$$

It follows by induction that

$$Q = Q + \psi a(x).(\tau.Q' + \sum_{j=1}^{k} \psi[x{=}y_j]\tau.Q_j)$$

If $Q'\sigma \stackrel{\tau}{\Longrightarrow} Q_{k+1}$ then

$$Q' = Q' + \psi[x{\notin}V]\tau.Q_{k+1}$$

otherwise

$$\tau.Q'\sigma = \tau.Q_{k+1} = \tau.Q_{k+1} + \psi[x{\notin}V]\tau.Q_{k+1} = \tau.Q'\sigma + \psi[x{\notin}V]\tau.Q_{k+1}$$

Putting everything together, one gets the required result. □

A.7 The Proof of Lemma 79

Proof. By assumption, for each $j \in \{1, \ldots, k\}$, the following conditions hold:

- $Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_j\sigma$ and either $Q'_j\sigma[y_j/x] \stackrel{\tau}{\Longrightarrow} Q_j$ or $Q'_j\sigma[y_j/x] \equiv Q_j$.
- $Q\sigma \Longrightarrow \stackrel{a(x)}{\longrightarrow} Q'_{k+1}\sigma$ and either $Q'_{k+1}\sigma \stackrel{\tau}{\Longrightarrow} Q_{k+1}$ or $Q'_{k+1}\sigma \equiv Q_{k+1}$.

By Lemma 74 one gets that

$$AS^e_w \vdash Q = Q + \psi a(x).Q'_1 + \ldots + \psi a(x).Q'_k + \psi a(x).Q'_{k+1}$$

Now $fn(Q'_j) \subseteq \{x\} \cup fn(Q) \subseteq \{x\} \cup V$; $\sigma[y_j/x]$ is complete on $\{x\} \cup V$ and is induced by $\psi[x{=}y_j]$. There are two cases:

- If $Q'_j\sigma[y_j/x] \stackrel{\tau}{\Longrightarrow} Q_j$ then $Q'_j = Q'_j + \psi[x{=}y_j]\tau.Q_j$ follows by Lemma 74. Now by Lemma 69, one has

$$\tau.(P{+}[x{=}y]\tau.Q) = \tau.(P{+}\tau.([x{=}y]\tau..Q{+}\tau.[x{\neq}y]\tau.P))$$

Using the above equality, one has

$$\begin{array}{rcl} \psi a(x).Q'_j &=& \psi a(x).(Q'_j{+}\psi[x{=}y_j]\tau.Q_j) \\ &=& \psi a(x).(Q'_j{+}[x{=}y_j]\tau.Q_j) \\ &=& \psi a(x).(Q'_j{+}\tau.([x{=}y_j]\tau.Q_j{+}[x{\neq}y_j]\tau.Q'_j)) \\ &=& \psi a(x).Q'_j + \psi a(x).([x{=}y_j]\tau.Q_j{+}[x{\neq}y_j]\tau.Q'_j) \end{array}$$

- If $Q'_j\sigma[y_j/x] \equiv Q_j$ then

$$\begin{array}{rcl} \psi a(x).Q'_j &=& \psi a(x).Q'_j + \psi a(x).([x{=}y_j]\tau.Q'_j{+}[x{\neq}y_j]\tau.Q'_j) \\ &=& \psi a(x).Q'_j + \psi a(x).([x{=}y_j]\tau.Q'_j\sigma[y_j/x]{+}[x{\neq}y_j]\tau.Q'_j) \\ &=& \psi a(x).Q'_j{+}\psi a(x).([x{=}y_j]\tau.Q_j{+}[x{\neq}y_j]\tau.Q'_j) \end{array}$$

Now if $Q'_{k+1}\sigma \stackrel{\tau}{\Longrightarrow} Q_{k+1}$ then by Lemma 69 one gets

$$\tau.(P{+}[x{\notin}V]\tau.Q) = \tau.(P{+}\tau.([x{\notin}V]\tau.Q{+}[x{\in}V]\tau.P))$$

By Lemma 74 and the above equality one has

$$\begin{array}{rcl} \psi a(x).Q'_{k+1} &=& \psi a(x).(Q'_{k+1}{+}\psi[x{\notin}V]\tau.Q_{k+1}) \\ &=& \psi a(x).(Q'_{k+1}{+}[x{\notin}V]\tau.Q_{k+1}) \\ &=& \psi a(x).(Q'_{k+1}{+}\tau.([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1})) \\ &=& \psi a(x).Q'_{k+1}{+}\psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1}) \end{array}$$

If $Q'_{k+1}\sigma \equiv Q_{k+1}$ then

$$\begin{array}{rcl} \psi a(x).Q'_{k+1} &\stackrel{M2}{=}& \psi a(x).Q'_{k+1}\sigma \\ &=& \psi a(x).Q_{k+1} \\ &=& \psi a(x).\tau.Q_{k+1} \\ &\stackrel{S5}{=}& \psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q_{k+1}) \end{array}$$

In either case we may assume that

$$\psi a(x).Q'_{k+1} = \psi a(x).Q'_{k+1}{+}\psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1})$$

It follows that

$$\begin{array}{rcl} Q &=& Q + \psi a(x).([x{=}y_1]\tau.Q_1{+}[x{\neq}y_1]\tau.Q'_1) \\ && \vdots \\ && {+}\psi a(x).([x{=}y_k]\tau.Q_k{+}[x{\neq}y_k]\tau.Q'_k) \\ && {+}\psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1}) \end{array}$$

Using S6 one has

$$\begin{array}{rl} & \psi a(x).([x{=}y_k]\tau.Q_k{+}[x{\neq}y_k]\tau.Q'_k) + \psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1}) \\ = & \psi a(x).([x{=}y_k]\tau.Q_k{+}[x{\neq}y_k]\tau.Q'_k) + \psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1}) \\ & {+}\psi a(x).([x{=}y_k]([x{=}y_k]\tau.Q_k{+}[x{\neq}y_k]\tau.Q'_k) + [x{\neq}y_k]([x \notin V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1})) \\ = & \psi a(x).([x{=}y_k]\tau.Q_k{+}[x{\neq}y_k]\tau.Q'_k) + \psi a(x).([x{\notin}V]\tau.Q_{k+1}{+}[x{\in}V]\tau.Q'_{k+1}) \\ & {+}\psi a(x).([x{=}y_k]\tau.Q_k{+}[x{\notin}V]\tau.Q_{k+1}) \end{array}$$

To continue consider the following process

$$\psi a(x).([x{=}y_{k-1}]\tau.Q_{k-1}{+}[x{\neq}y_{k-1}]\tau.Q'_{k-1}) + \psi a(x).([x{=}y_k]\tau.Q_k{+}[x{\notin}V]\tau.Q_{k+1})$$

Using similar argument we get

$$\psi a(x).([x{=}y_{k-1}]\tau.Q_{k-1}{+}[x{\neq}y_{k-1}][x{=}y_k]\tau.Q_k{+}[x{\notin}V]\tau.Q_{k+1})$$

By repeating this construction we finally get

$$\psi a(x).([x{=}y_1]\tau.Q_1{+}\ldots{+}[x{\neq}y_1]\ldots[x{\neq}y_{k-1}][x{=}y_k]\tau.Q_k{+}[x{\notin}V]\tau.Q_{k+1})$$

Therefore

$$Q = Q + \psi a(x).([x{=}y_1]\tau.Q_1{+}\ldots{+}[x{\neq}y_1]\ldots[x{\neq}y_{k-1}][x{=}y_k]\tau.Q_k{+}[x{\notin}V]\tau.Q_{k+1})$$

This complete the proof. □

A.8 The Proof of Lemma 80

Proof. If $Q\sigma{\Longrightarrow}Q_1 \xrightarrow{a(x)} Q'$ then $AS_o^g \vdash Q = Q{+}\psi a(x).Q'$ follows by Lemma 74. If $Q\sigma{\Longrightarrow} \xrightarrow{a(x)} Q_1 \overset{\tau}{\Longrightarrow} Q'$ then $[x{\notin}V]\psi$ is complete on $fn(Q_1)$ since $fn(Q_1) \subseteq fn(Q) \cup \{x\}$. Thus σ is induced by $[x{\notin}V]\psi$. Consequently by Lemma 74

$$AS_o^g \vdash Q_1 = Q_1{+}[x{\notin}V]\psi\tau.Q'$$

It follows that

$$\begin{array}{rcl} AS_o^g \vdash Q & = & Q{+}\psi a(x).Q_1 \\ & = & Q{+}\psi a(x).(Q_1{+}[x{\notin}V]\psi\tau.Q') \\ & = & Q{+}\psi(a(x).(Q_1{+}[x{\notin}V]\psi\tau.Q'){+}\psi a(x).Q') \\ & = & Q{+}\psi a(x).Q' \end{array}$$

where the third equation is valid by Lemma 72. □

A.9 The Proof of Lemma 81

Proof. The proofs of all the five clauses are similar to each other. The only difference is in the treatment of input actions and, to a lesser degree, restricted output actions. Now suppose $P \approx Q$, where $\approx$ is any of the five bisimilarities. By Lemma 60 we may assume that both P and Q are complete normal forms on $V \stackrel{\text{def}}{=} fn(P{+}Q){=}\{y_1, \ldots, y_k\}$. Let P be

$$\sum_{i\in I} \phi_i\lambda_i.P_i$$

and Q be

$$\sum_{j\in J} \psi_j\lambda_j.Q_j$$

Suppose σ is induced by ϕ_i and $P\sigma \xrightarrow{\lambda_i\sigma} P_i\sigma$.

The proof of this lemma is carried out by induction on the sum of depths of P and Q.

If λ_i is a τ prefix and the τ action is simulated by $Q\sigma \overset{\tau}{\Longrightarrow} Q_j\sigma$ for some $j \in J$, then by induction hypothesis $AS \cup \{T1, T2, T3\} \vdash \tau.P_i\sigma = \tau.Q_j\sigma$. The simulation can happen only if $\phi_i \Rightarrow \psi_j$. But then $\phi_i \Leftrightarrow \psi_j$ since both ϕ_i and ψ_j are complete on V and agree with σ. By (i) of Lemma 74, $AS \cup \{T1, T2, T3\} \vdash Q{+}\phi_i\tau.P_i = Q{+}\phi_i\tau.P_i\sigma = Q{+}\psi_j\tau.Q_j\sigma = Q{+}\psi_j\tau.Q_j = Q$. If $P\sigma \xrightarrow{\tau} P_i\sigma$ is simulated by vacuous action then $AS \cup \{T1, T2, T3\} \vdash Q{+}\phi_i\tau.P_i = Q{+}\phi_i\tau.P_i\sigma = Q{+}\phi_i\tau.Q\sigma = Q{+}\phi_i\tau.Q$.

If λ_i is an output prefix $\overline{a}x$ then, by (ii) of Lemma 74 and similar approach, it is easy to see that $AS \cup \{T1, T2, T3\} \vdash Q{+}\phi_i\overline{a}x.P_i = Q$.

Suppose λ_i is a restricted output action $\overline{a}(x)$. If $P \approx_w^l Q$ then some Q' exists such that $Q\sigma \stackrel{\overline{a\sigma}(x)}{\Longrightarrow} Q'\sigma \dot{\approx}_w^l P_i\sigma$. Therefore $([x\notin V]\phi_i Q')\sigma \approx_w^l ([x\notin V]\phi_i P_i)\sigma$. Since $[x\notin V]\phi_i$ is complete on $V \cup \{x\} \supseteq fn(Q'+P_i\sigma)$ we may deduce from Lemma 45 that $[x\notin V]\phi_i Q' \approx_w^l [x\notin V]\phi_i P_i$. By induction hypothesis $AS_w^l \vdash \tau.[x\notin V]\phi_i Q' = \tau.[x\notin V]\phi_i P_i$. Consequently

$$
\begin{array}{rcl}
AS_w^l \vdash Q+\phi_i\overline{a}(x).P_i & = & Q+\phi_i\overline{a\sigma}(x).[x\notin V]\phi_i P_i \\
& = & Q+\phi_i\overline{a\sigma}(x).\tau.[x\notin V]\phi_i P_i \\
& = & Q+\phi_i\overline{a}(x).\tau.[x\notin V]\phi_i Q' \\
& = & Q+\phi_i\overline{a}(x).[x\notin V]\phi_i Q' \\
& \stackrel{L8}{=} & Q+\phi_i\overline{a}(x).\phi_i Q' \\
& = & Q+\phi_i\overline{a}(x).Q' \\
& = & Q
\end{array}
$$

The proof for weak early bisimilarity is similar. If $P \approx_o^l Q$ then some Q' exists such that $Q\sigma \stackrel{\overline{a\sigma}(x)}{\Longrightarrow} Q'$ and $(P_i\sigma)^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \approx_o^l (Q')^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]}$. By induction hypothesis one has

$$AS_o^l \vdash \tau.(P_i\sigma)^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} = \tau.(Q')^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]}$$

Using L8 one obtains that $\overline{a}(x).P = \overline{a}(x).P^{[x\notin Y]}$ for any set Y whose elements are names free in P. Therefore by Lemma 75 one gets

$$
\begin{array}{rcl}
AS_o^l \vdash Q+\phi_i\overline{a}(x).P_i & = & Q+\phi_i\overline{a\sigma}(x).P_i\sigma \\
& = & Q+\phi_i\overline{a\sigma}(x).(P_i\sigma)^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \\
& = & Q+\phi_i\overline{a\sigma}(x).\tau.(P_i\sigma)^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \\
& = & Q+\phi_i\overline{a\sigma}(x).\tau.(Q')^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \\
& = & Q+\phi_i\overline{a\sigma}(x).(Q')^{[x\notin(fn(P_i\sigma+Q')\setminus\{x\})]} \\
& = & Q+\phi_i\overline{a\sigma}(x).Q' \\
& = & Q
\end{array}
$$

The proofs for the cases of the weak early bisimilarity and the weak open bisimilarity are similar.

Next we examine input actions in each of the five cases. Suppose λ_i is $a\sigma(x)$.

(i) $P \approx_w^l Q$. λ_i is an input prefix. Let σ be a substitution that is induced by ϕ_i. Now $P\sigma \stackrel{a\sigma(x)}{\longrightarrow} P_i\sigma$. It follows from $P \approx_w^l Q$ that Q' exists such that the following properties hold:

1. $Q\sigma \Longrightarrow\stackrel{a\sigma(x)}{\longrightarrow} Q'\sigma$.
2. Q_{i_l} exists such that $Q'\sigma[y_l/x] \Longrightarrow Q_{i_l}\sigma[y_l/x] \dot{\approx}_w^l P_i\sigma[y_l/x]$ for $l \in \{1,\ldots,k\}$.
3. $Q_{i_{k+1}}$ exists such that $Q'\sigma \Longrightarrow Q_{i_{k+1}}\sigma \dot{\approx}_w^l P_i\sigma$.

It follows that, for $l \in \{1,\ldots,k\}$,

$$
\begin{array}{rcl}
(\phi_i[x{=}y_l]Q_{i_l})\sigma[y_l/x] & \dot{\approx}_w^l & (\phi_i[x{=}y_l]P_i)\sigma[y_l/x] \\
(\phi_i[x\notin V]Q_{i_{k+1}})\sigma & \dot{\approx}_w^l & (\phi_i[x\notin V]P_i)\sigma
\end{array}
$$

Consequently, for $l \in \{1,\ldots,k\}$,

$$
\begin{array}{rcl}
\phi_i[x{=}y_l]Q_{i_l} & \approx_w^l & \phi_i[x{=}y_l]P_i \\
\phi_i[x\notin V]Q_{i_{k+1}} & \approx_w^l & \phi_i[x\notin V]P_i
\end{array}
$$

by Lemma 45. Then by induction one has, for $l \in \{1,\ldots,k\}$,

$$
\begin{array}{rcl}
AS_w^l \vdash \tau.\phi_i[x{=}y_l]Q_{i_l} & = & \tau.\phi_i[x{=}y_l]P_i \\
AS_w^l \vdash \tau.\phi_i[x\notin V]Q_{i_{k+1}} & = & \tau.\phi_i[x\notin V]P_i
\end{array}
$$

By Lemma 77, one can get

$$
\begin{array}{rcl}
Q & = & Q+\phi_i a\sigma(x).(\tau.Q'+\phi_i\sum_{l=1}^{k}[x=y_l]\tau.Q_{i_l}+\phi_i[x\notin V]\tau.Q_{i_{k+1}}) \\
& \overset{M6}{=} & Q+\phi_i a(x).(\tau.Q'+\phi_i\sum_{l=1}^{k}[x=y_l]\tau.\phi_i[x=y_l]Q_{i_l}+\phi_i[x\notin V]\tau.\phi_i[x\notin V]Q_{i_{k+1}}) \\
& = & Q+\phi_i a(x).(\tau.Q'+\phi_i\sum_{l=1}^{k}[x=y_l]\tau.\phi_i[x=y_l]P_i+\phi_i[x\notin V]\tau.\phi_i[x\notin V]P_i) \\
& = & Q+\phi_i a(x).(\tau.Q'+\phi_i\sum_{l=1}^{k}[x=y_l]\tau.P_i+\phi_i[x\notin V]\tau.P_i) \\
& = & Q+\phi_i a(x).(\phi_i\tau.Q'+\phi_i\tau.P_i) \\
& = & Q+\phi_i a(x).(\phi_i\tau.Q'+\tau.P_i) \\
& = & Q+\phi_i a(x).P_i
\end{array}
$$

This completes the proof of (i).

(ii) $P\approx_w^e Q$ and $P\sigma \overset{a\sigma(x)}{\longrightarrow} P_i\sigma$. It follows from $P\approx_w^e Q$ that

$Q\sigma \overset{a\sigma(x)}{\Longrightarrow} Q'_{i_1}\sigma$, $Q'_{i_1}\sigma[y_1/x]\Longrightarrow Q_{i_1}\sigma[y_1/x]\dot{\approx}_w^e P_i\sigma[y_1/x]$,

$\vdots$

$Q\sigma \overset{a\sigma(x)}{\Longrightarrow} Q'_{i_k}\sigma$, $Q'_{i_k}\sigma[y_k/x]\Longrightarrow Q_{i_k}\sigma[y_k/x]\dot{\approx}_w^e P_i\sigma[y_k/x]$,

$Q\sigma \overset{a\sigma(x)}{\Longrightarrow} Q_{i_{k+1}}\sigma\dot{\approx}_w^e P_i\sigma$

for $Q'_{i_1},\ldots,Q'_{i_k},Q_{i_1},\ldots,Q_{i_k},Q_{i_{k+1}}$. So for each $l\in\{1,\ldots,k\}$,

$$
\begin{array}{rcl}
(\phi_i[x=y_l]Q_{i_l})\sigma[y_l/x] & \dot{\approx}_w^e & (\phi_i[x=y_l]P_i)\sigma[y_l/x] \\
(\phi_i[x\notin V]Q_{i_{k+1}})\sigma & \dot{\approx}_w^e & (\phi_i[x\notin V]P_i)\sigma
\end{array}
$$

By Lemma 45 one gets, for $l\in\{1,\ldots,k\}$, that

$$
\begin{array}{rcl}
\phi_i[x=y_l]Q_{i_l} & \approx_w^e & \phi_i[x=y_l]P_i \\
\phi_i[x\notin V]Q_{i_{k+1}} & \approx_w^e & \phi_i[x\notin V]P_i
\end{array}
$$

Using induction hypothesis one infers, for $l\in\{1,\ldots,k\}$, that

$$
\begin{array}{rcl}
AS_w^e & \vdash & \tau.(\phi_i[x=y_l]Q_{i_l})=\tau.(\phi_i[x=y_l]P_i) \\
AS_w^e & \vdash & \tau.(\phi_i[x\notin V]Q_{i_{k+1}})=\tau.(\phi_i[x\notin V]P_i)
\end{array}
$$

By Lemma 79, one derives that

$$
\begin{array}{rcl}
Q & = & Q+\phi_i a\sigma(x).([x=y_1]\tau.Q_{i_1}+[x\neq y_1][x=y_2]\tau.Q_{i_2}+\ldots+[x\notin V]\tau.Q_{i_{k+1}}) \\
& = & Q+\phi_i a(x).(\phi_i[x=y_1]\tau.\phi_i[x=y_1]Q_{i_1}+\phi_i[x\neq y_1][x=y_2]\tau.\phi_i[x=y_2]Q_{i_2} \\
& & +\ldots+\phi_i[x\notin V]\tau.\phi_i[x\notin V]Q_{i_{k+1}}) \\
& = & Q+\phi_i a(x).(\phi_i[x=y_1]\tau.\phi_i[x=y_1]P_i+\phi_i[x\neq y_1][x=y_2]\tau.\phi_i[x=y_2]P_i \\
& & +\ldots+\phi_i[x\notin V]\tau.\phi_i[x\notin V]P_i) \\
& = & Q+\phi_i a(x).([x=y_1]\tau.P_i+[x\neq y_1][x=y_2]\tau.P_i+\ldots+[x\notin V]\tau.P_i) \\
& = & Q+\phi_i a(x).P_i
\end{array}
$$

which completes the proof of (ii).

(iii) $P\approx_o^l Q$ and $P\sigma \overset{a\sigma(x)}{\longrightarrow} P_i\sigma$. It follows from $P\approx_o^l Q$ that Q' exists such that $Q\sigma \Longrightarrow\overset{a\sigma(x)}{\longrightarrow} Q'\sigma$ and the following conditions hold:

- For each $l\in\{1,\ldots,k\}$, Q_{i_l} exists such that $Q'\sigma[y_l/x]\Longrightarrow Q_{i_l}\approx_o^l P_i\sigma[y_l/x]$.

- $Q_{i_{k+1}}$ exists such that $Q'\sigma \Longrightarrow Q_{i_{k+1}} \approx_o^l P_i\sigma$.

Now by Lemma 77 one can get

$$\begin{array}{rcl} AS_o^l \vdash Q & = & Q+\phi_i a(x).(\tau.Q'+\sum_{l=1}^{k}\phi_i[x=y_l]\tau.Q_{i_l}\sigma[y_l/x]+\phi_i[x\notin V]\tau.Q_{i_{k+1}}\sigma) \\ & = & Q+\phi_i a(x).(\tau.Q'+\sum_{l=1}^{k}\phi_i[x=y_l]\tau.P_i\sigma[y_l/x]+\phi_i[x\notin V]\tau.P_i\sigma) \\ & = & Q+\phi_i a(x).(\tau.Q'+\sum_{l=1}^{k}\phi_i[x=y_l]\tau.P_i+\phi_i[x\notin V]\tau.P_i) \\ & = & Q+\phi_i a(x).(\tau.Q'+\phi_i\tau.P_i) \\ & = & Q+\phi_i a(x).(\tau.Q'+\tau.P_i) \\ & = & Q+\phi_i a(x).P_i \end{array}$$

where the fourth equality is justified by Lemma 61.

(iv) $P \approx_o^e Q$ and$P\sigma \stackrel{a\sigma(x)}{\longrightarrow} P_i\sigma$. Then both of the following conditions hold:

- For each $l \in \{1,\ldots,k\}$, Q'_{i_l} and Q_{i_l} exist such that $Q\sigma\Longrightarrow \stackrel{a\sigma(x)}{\longrightarrow} Q'_{i_l}\sigma$ and $Q'_{i_l}\sigma[y_l/x] \Longrightarrow Q_{i_l} \approx_o^e P_i\sigma[y_l/x]$.
- Some $Q'_{i_{k+1}}$ and $Q_{i_{k+1}}$ exist such that $Q\sigma\Longrightarrow \stackrel{a\sigma(x)}{\longrightarrow} Q'_{i_{k+1}}\sigma \Longrightarrow Q_{i_{k+1}} \approx_o^e P_i\sigma$.

By Lemma 78 one can get

$$\begin{array}{rcl} AS_o^e \vdash Q & = & Q+\sum_{l=1}^{k}\phi_i a(x).(\tau.Q'_{i_l}+\phi_i[x=y_l]\tau.Q_{i_l})+\phi_i a(x).(\tau.Q'_{i_{k+1}}+\phi_i[x\notin V]\tau.Q_{i_{k+1}}) \\ & = & Q+\sum_{l=1}^{k}\phi_i a(x).(\tau.Q'_{i_l}+\phi_i[x=y_l]\tau.P_i\sigma[y_l/x])+\phi_i a(x).(\tau.Q'_{i_{k+1}}+\phi_i[x\notin V]\tau.P_i\sigma) \\ & = & Q+\sum_{l=1}^{k}\phi_i a(x).(\tau.Q'_{i_l}+\phi_i[x=y_l]\tau.P_i)+\phi_i a(x).(\tau.Q'_{i_{k+1}}+\phi_i[x\notin V]\tau.P_i) \\ & \stackrel{T5}{=} & Q+\sum_{l=1}^{k}\phi_i a(x).(\tau.Q'_{i_l}+\phi_i[x=y_l]\tau.P_i)+\phi_i a(x).(\tau.Q'_{i_{k+1}}+\phi_i[x\notin V]\tau.P_i)+\phi_i a(x).P_i \\ & = & Q+\phi_i a(x).P_i \end{array}$$

This completes the proof of (iv).

(v) $P \approx_o^w Q$ and $P\sigma \stackrel{a\sigma(x)}{\longrightarrow} P_i\sigma$. It follows that $Q\sigma \stackrel{a\sigma(x)}{\Longrightarrow} Q' \approx_o^w P_i\sigma$. By induction one has that

$$AS_o^w \vdash \tau.Q' = \tau.P_i\sigma$$

By Lemma 80

$$\begin{array}{rcl} AS_o^w \vdash Q & = & Q+\phi_i a\sigma(x).Q' \\ & = & Q+\phi_i a\sigma(x).\tau.Q' \\ & = & Q+\phi_i a\sigma(x).\tau.P_i\sigma \\ & = & Q+\phi_i a(x).\tau.P_i \\ & = & Q+\phi_i a(x).P_i \end{array}$$

This completes the proofs of all the input cases.

In summary in each of the five axiomatic systems AS', one has that

$$AS'\vdash \sum_{i\in I'}\phi_i\tau.Q+Q=P+Q$$

for some subset I' of I. It follows from $T4$ that

$$AS'\vdash\tau.(P+Q)=\tau.(\sum_{i\in I'}\phi_i\tau.Q+Q)=\tau.Q$$

Symmetrically $AS'\vdash\tau.(P{+}Q){=}\tau.P$. Hence $AS'\vdash\tau.P{=}\tau.Q$. $\square$