On Quasi Open Bisimulation*

Yuxi Fu†

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

Abstract

Quasi open bisimilarity is a variant of the open bisimilarity based on a closer examination of the observationality of local names. The paper investigates two alternative characterizations of the quasi open bisimilarity and provides a complete system for the weak quasi open congruence.

1 Introduction

In the theory of process calculus, mobility is formalized by the communication mechanism that allows names to be sent and received by the communicating processes. This idea was formally investigated by Milner, Parrow and Walker in [10] for the first time within the framework of the π-calculus. For an expository introduction to the π-calculus and its variants, see [15]. Through name updates, mobile processes may change their communication topologies during their evolutions. This dynamic feature adds to the difficulties of defining the equivalence relations for the mobile processes.

Equivalence relations on mobile processes are observational, meaning that two processes are equated if no environments can observe any difference by interacting with them. Formally the environments are represented by the contexts. When a process P is placed in an environment $C[\_]$, it is observed by the environment by letting the environment interact with it. Intuitively P and Q are observationally equivalent if an arbitrarily chosen environment $C[\_]$ can not tell which of the two is placed within it. Now the input prefixes in $C[\_]$ might bind free names in the process placed within it. Consequently observational equivalences on mobile processes are closed under substitution. One such equivalence is the open bisimilarity proposed by Sangiorgi [13]. This relation differs from other bisimulation equivalences in that it is closed under substitution in every bisimulation step. This stronger requirement is reasonable from the point of view of modern computing framework. In the scenario of internet computing, not only the programmes are mobile and therefore must be robust enough to survive in different environments, but also that the environments are changing all the time and programmes are under constant attacks from the environments. What it implies to the process equivalences is that the environments are dynamic in the sense that after an environment has made an observation it might become randomly different. Let's see a well-known example: We have two π-processes defined as follows:

$$A \stackrel{\text{def}}{=} \overline{a}a.\overline{b}b + \overline{a}a$$
$$B \stackrel{\text{def}}{=} A + \overline{a}a.[x{=}y]\overline{b}b$$

They are not open bisimilar. Suppose $C[\_]$ is a context and that it is ready to interact with A and B through channel a. Clearly the observation $C[B] \stackrel{\tau}{\longrightarrow} C'[[x{=}y]\overline{b}b]$ is admissible. Now A admits the same observation in two manners: Either $C[A] \stackrel{\tau}{\longrightarrow} C'[\overline{b}b]$ or $C[A] \stackrel{\tau}{\longrightarrow} C'[\mathbf{0}]$. In the π-calculus, $C'[\_]$ can *not* observe any difference between $[x{=}y]\overline{b}b$ and $\mathbf{0}$ if $x \neq y$ and between $[x{=}y]\overline{b}b$ and $\overline{b}b$ if $x = y$. But according to the above analysis, we should not take for granted that after the first observation the environment is $C'[\_]$. It might well have changed into $C''[\_]$ that contains a prefix '$c(x)$.' in front of the hole. It is possible that a sequence of observations of $C''[[x{=}y]\overline{b}b]$ is admissible neither by $C''[\overline{b}b]$ nor by $C''[\mathbf{0}]$. From the

*The paper has appeared in *Theoretical Computer Science*, 338: 96-126, 2005.

†The author is supported by The NNSFC for Distinguished Young Scholars (60225012) and The National 973 Project (2003CB316905).

bisimulation point of view, the inequivalence can be explained as follows: The action $B \xrightarrow{\overline{a}a} [x{=}y]\overline{b}b$ can not be simulated by any action from A. A substitution either identifies x with y or leaves them distinct. In the former case $\overline{b}b$ can be fired while in the latter case $\overline{b}b$ cannot be activated. So there is a choice for the environment. However the choice of the environment is void after A has performed an $\overline{a}a$ action.

In retrospect one of the selling points of bisimulations is that they have taken into account the malicious nature of the environments. The zigzag property of the bisimulation is what is necessary to uphold the soundness of the bisimulation equivalences against the dynamic environments. The weak bisimulation equivalence in CCS ([9]) can be understood in this interpretation. It is our personal belief that the open bisimilarity can claim to be *the* bisimulation equivalence on the mobile processes.

A more interesting counter example to the open bisimilarity is given by Sangiorgi and Walker [14]. Consider the pair of the π-processes:

$$C \stackrel{\text{def}}{=} (z)\overline{a}z.(a(w) + a(w).\overline{z}z + a(w).[w{=}z]\overline{z}z) \tag{1}$$

$$D \stackrel{\text{def}}{=} (z)\overline{a}z.(a(w) + a(w).\overline{z}z) \tag{2}$$

According to the definition of the open bisimulation $C \xrightarrow{\overline{a}(z)} \xrightarrow{a(w)} [w{=}z]\overline{z}z$ cannot be simulated by D. But from an observational viewpoint, C and D should be equivalent. In an interacting environment the component $a(w).[w{=}z]\overline{z}z$ could receive a name. This name could be one of the three kinds:

- It is z. In this case $[w{=}z]\overline{z}z$ is equivalent to $\overline{z}z$.
- It is a free name x. In this case $[w{=}z]\overline{z}z$ is equivalent to $\mathbf{0}$. This is because in the π-calculus the free name x is never identified with the local name z.
- It is a local name different from z. In this case $[w{=}z]\overline{z}z$ is also equivalent to $\mathbf{0}$. This is because in the π-calculus two distinct local names can never be identified.

So the component $a(w).[w{=}z]\overline{z}z$ could be simulated either by $a(w).\overline{z}z$ or by $a(w)$. From a context point of view the observations a context makes on $(z)\overline{a}z.(a(w) + a(w).\overline{z}z + a(w).[w{=}z]\overline{z}z)$ and $(z)\overline{a}z.(a(w) + a(w).\overline{z}z)$ are the same. After the observation these two processes have evolved into $a(w) + a(w).\overline{z}z + a(w).[w{=}z]\overline{z}z$ and $a(w) + a(w).\overline{z}z$ respectively. By the above analysis, the following observation

$$C[a(w) + a(w).\overline{z}z + a(w).[w{=}z]\overline{z}z] \xrightarrow{\tau} C'[[x{=}z]\overline{z}z]$$

by the context $C[\_]$ can be made to $a(w) + a(w).\overline{z}z$ as well. This counter example points out a deficiency of the open bisimilarity: The open semantics imposes too strong a requirement on names 'opened up' by the bound output actions, which has hardly any practical significance. In order to rectify this deficiency, Sangiorgi and Walker proposed quasi open bisimilarity in [14]. The quasi open bisimilarity is weaker than the open bisimilarity. For instance the processes defined in (1) and (2) are quasi open bisimilar although they are not open bisimilar. Sangiorgi and Walker have worked out the relationship of the quasi open bisimilarity to some of the well known bisimilarities. They have shown that the quasi open bisimilarity coincides with the open barbed bisimilarity, where the open barbed bisimilarity differs from the barbed equivalence in that the former is closed under substitution in every bisimulation step. In the light of the above discussions this result definitely adds weight to the importance of the quasi open bisimilarity.

In summary, the open bisimilarity has a more authentic role to play than either the early bisimilarity or the late bisimilarity. Our previous work has shown that the open bisimulations are more subtle than they appear to be ([2]). As the rectification of the open bisimilarity, the quasi open bisimilarity definitely calls for more attention.

In this paper we study the algebraic theory of the quasi open bisimilarity. The main contributions of the paper are as follows:

- We demonstrate the importance of the quasi open bisimilarity by showing that it is what one obtains if one formalizes the idea of observations of the dynamic environments.
- We introduce an operation that removes match/mismatch operators involving local names that have just been opened up. By using this operation we propose the Q-open bisimilarity that makes room for a proof of a completeness result for the quasi open congruence. The equational system contains a schematic rule that captures the difference between the open bisimilarity and the quasi open bisimilarity. Thus we provide a full picture on the equality defined by the quasi open bisimulations.

The paper is organized as follows. Section 2 introduces the preliminaries. Section 3 reviews the definitions and some basic properties of the quasi open bisimilarity. Section 4 does the same for the open barbed bisimilarity. Section 5 and Section 6 provide two alternative characterizations of the quasi open bisimilarity. Section 7 proposes a complete system for the quasi open congruence. Section 8 concludes.

2 Preliminary

The process calculus we focus in this paper is the full π-calculus equipped with the mismatch operator. The abstract syntax of the calculus is given below:

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

In the above definition π ranges over the set $\{a(x), \overline{a}x \mid a, x \in \mathcal{N}\} \cup \{\tau\}$ and the small letters range over the set $\mathcal{N}$ of names. In both $a(x).P$ and $(x)P$ the name x is bound. The bound output prefix is defined as follows:

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

The notations $n(\_)$, $fn(\_)$ and $bn(\_)$ represent the sets of names, respectively free names and bound names, in a syntactical object.

The operational semantics of the calculus is defined by the standard approach using a labelled transition system. In the following semantic rules, λ ranges over the set $\{ax, \overline{a}x, \overline{a}(x) \mid a, x \in \mathcal{N}\} \cup \{\tau\}$.

Prefix

$$\frac{}{a(x).P \xrightarrow{ay} P\{y/x\}} \qquad \frac{}{\overline{a}x.P \xrightarrow{\overline{a}x} P} \qquad \frac{}{\tau.P \xrightarrow{\tau} P}$$

Composition

$$\frac{P \xrightarrow{\lambda} P'}{P \,|\, Q \xrightarrow{\lambda} P' \,|\, Q} \qquad \frac{P \xrightarrow{ay} P' \quad Q \xrightarrow{\overline{a}y} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'} \qquad \frac{P \xrightarrow{ax} P' \quad Q \xrightarrow{\overline{a}(x)} Q'}{P \,|\, Q \xrightarrow{\tau} (x)(P' \,|\, Q')}$$

Restriction

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

Condition

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

Choice

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

Replication

$$\frac{P \,|\, !P \xrightarrow{\lambda} P'}{!P \xrightarrow{\lambda} P'}$$

In the above transition system, we have left out all the symmetric rules. In the third composition rule, one needs to use the α-conversion to rename a bound name so that the new name x does not occur in P. The second condition rule says that for distinct names x and y the process $[x{\neq}y]P$ behaves in the same way as the process P. In this paper we take the view that two names are different if they are syntactically distinct. To go along with this view, we assume that all bound names are pairwise distinct and are different from the global names. This is why we have dropped the side condition $bn(\lambda) \cap fn(Q) = \emptyset$ on the first composition rule. The notation $\{y/x\}$ stands for a substitution that replaces x by y throughout the term it applies. Formally 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, for $i \in \{1, \ldots, n\}$, and is constant elsewhere. If $\sigma = \{y_1/x_1, \ldots, y_n/x_n\}$ then $n(\sigma) \stackrel{\mathrm{def}}{=} \{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ and $rng(\sigma) \stackrel{\mathrm{def}}{=} \{y_1, \ldots, y_n\}$. The composition $\sigma_1\sigma_2$ of σ_1 and σ_2 is defined as follows: $P\sigma_1\sigma_2 \stackrel{\mathrm{def}}{=} (P\sigma_1)\sigma_2$.

The operational semantics we have defined is the early semantics, which is clear from the transition $a(x).P \xrightarrow{ay} P\{y/x\}$. In the early semantics one has that $a(x).P \xrightarrow{a(x)} P$. The difference is that in the former the instantiations of the input actions happen at the level of the observable actions while in the latter they occur in the unobservable internal communications.

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

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 $C'[] \,|\, P$, or symmetrically $P \,|\, C'[]$, for some strong context $C'[]$ and some process P;
- or $(x)C[]$ for some context $C'[]$ and some name x;
- or $[x{=}y]C'[]$ for some context $C'[]$ and some names x and y.

A strong context is either a context, or of the form $C'[] + P$, or symmetrically $P + C'[]$, for some strong context $C'[]$ and some process P, or of the form $[x{\neq}y]C'[]$ for some strong context $C'[]$ and names x, y.

We will abbreviate a sequence $x_1, \ldots, x_n$ of names to $\widetilde{x}$. Accordingly $(x_1)\ldots(x_n)P$ will be abbreviated to $(\widetilde{x})P$. When the length of $\widetilde{x}$ is zero, $(\widetilde{x})P$ is just P. By abuse of notation, we will also write $\widetilde{x}$ for the set $\{x_1, \ldots, x_n\}$. When the lengths of $\widetilde{x}$ and $\widetilde{y}$ are the same, we sometimes write $\{\widetilde{y}/\widetilde{x}\}$ for the substitution that replaces each $x \in \widetilde{x}$ by the corresponding $y \in \widetilde{y}$.

The following definition will play an important role in some major proofs of the paper. It is slightly different from what is introduced in [14].

Definition 1. A substitution σ respects $\widetilde{x}$ if $\forall x \in \widetilde{x}.x\sigma = x$ and $\forall y \notin \widetilde{x}.y\sigma \notin \widetilde{x}$.

It is clear that if both σ_1 and σ_2 respect $\widetilde{x}$ then $\sigma_1\sigma_2$ respects $\widetilde{x}$. This definition will be typically applied to a sequence $\widetilde{x}$ of local names that have been opened up, as it were, by bound output actions. A substitution respecting $\widetilde{x}$ pretends that $\widetilde{x}$ were still local names.

In the rest of this paper $\phi, \psi, \phi', \psi', \ldots$ denote finite lists of match and/or mismatch conditions. Consequently we will write ϕP and ψP etc.. If ϕ logically implies ψ, we write $\phi \Rightarrow \psi$; and if both $\phi \Rightarrow \psi$ and $\psi \Rightarrow \phi$ we write $\phi \Leftrightarrow \psi$. If ϕ is an empty list, it plays the role of the logical truth, denoted by $\top$, in which case ϕP is just P. Let $\perp$ denote the logical false operator. One then has for example $x{\neq}y \Leftrightarrow \perp$. When reasoning with process equality we sometimes write $\perp P$ for $\mathbf{0}$. Clearly ϕ defines an equivalence relation on the set $\mathcal{N}$ of names. We write s_ϕ to denote an arbitrarily chosen substitution that sends all the members of an equivalence class to a representative of that class.

Lemma 2. *Suppose σ is a substitution and φ is a sequence of matches and/or mismatches. For arbitrarily chosen $s_{\varphi\sigma}$ and s_φ there exists some substitution σ' such that $\sigma s_{\varphi\sigma} = s_\varphi \sigma' \sigma$.*

Proof. For any name a one has the following observations:

- a is mapped by s_φ onto a representative a' of the equivalence class $[a]$, where the equivalence relation is induced by φ;
- a is mapped by σ onto $a\sigma$ and then by $s_{\varphi\sigma}$ onto a representative $a''\sigma$ of the equivalence class $[a\sigma]$, where the equivalence relation is induced by $\varphi\sigma$.

Now define a substitution σ' as follows:

For each name a, let $a\sigma'$ be a''.

This substitution has the following property:

If a_1 and a_2 are in the same equivalence class $[a]$ of the equivalence relation induced by φ then $a_1\sigma' = a_2\sigma'$.

This is because $a_1\sigma, a_2\sigma$ must be in the same equivalence class of the equivalence relation induced by $\varphi\sigma$. It then follows easily that $\sigma s_{\varphi\sigma} = s_\varphi \sigma' \sigma$. □

In the full π-calculus, $P \xrightarrow{\lambda} P'$ does not necessarily imply that $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$. But the following property clearly holds, which we will be using without explicitly referring to.

Lemma 3. *If $P\sigma \xrightarrow{\lambda} P'$ then there exist P_1 and λ_1 such that $P\sigma \xrightarrow{\lambda_1\sigma} P_1\sigma \equiv P'$ and $\lambda = \lambda_1\sigma$.*

In some later proofs we will assume that the reader is familiar with the strong open bisimilarity $\sim$ ([13]) and the proof technique of bisimulation up to $\sim$ ([9]).

3 Quasi Open Bisimulation

Sangiorgi and Walker introduced the quasi open bisimulations for the π-calculus with the motivations explained in the introduction. The π-calculus they considered in [14] is the sub calculus without the mismatch operator. For the open bisimulations, the presence of the mismatch operator slightly complicates the observational theory. It is pointed out in [2] that there are two principal definitions of the open bisimulations which give rise to the early open bisimulations and the late open bisimulations. Here is an example that distinguishes the two: Let P be

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

Then $P{+}a(x).Q$ is early open bisimilar to P. But the two processes are not late open bisimilar. In this paper we focus on the early open bisimilarity. Hencefourth we will leave out the adjective "early".

Definition 4. A quasi open bisimulation is a family of symmetric relations $\{\mathcal{R}^{\widetilde{z}}\}_{\widetilde{z}\subseteq_f \mathcal{N}}$ on processes such that, for all P and Q, if $P\mathcal{R}^{\widetilde{z}}Q$ and σ respects $\widetilde{z}$ then the following properties hold:
(i) If $P\sigma \xrightarrow{\lambda} P'$, where λ is not a bound output action, then some Q' exists such that $Q\sigma \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}^{\widetilde{z}}P'$.
(ii) If $P\sigma \xrightarrow{\overline{a}(x)} P'$ then $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ for some Q' such that $P'\mathcal{R}^{\widetilde{z}x}Q'$.
We write $\{\approx_q^{\widetilde{z}}\}_{\widetilde{z}\subseteq_f \mathcal{N}}$ for the largest quasi open bisimulation and refer to $\approx_q^{\widetilde{z}}$ as the quasi open $\widetilde{z}$-bisimilarity. The quasi open bisimilarity $\approx_q$ is the quasi open $\emptyset$-bisimilarity $\approx_q^{\emptyset}$.

The above definition is taken from [15]. Had we used a late operational semantics as in the paper [15] of Sangiorgi and Walker, then the input actions would call for a special treatment since the calculus is equipped with the mismatch operator. The following simulating property, in the late operational semantics,

"if $P\sigma \xrightarrow{a(x)} P'$ then some Q' exists such that $Q\sigma \stackrel{a(x)}{\Longrightarrow} Q'\mathcal{R}^{\widetilde{z}}P'$"

is good for the calculus without the mismatch operator but wrong for the present calculus. We refer the reader to [2] for detailed treatment.

A quasi open bisimulation is a family of binary relations indexed by the finite sets of names. Intuitively $P \approx_q^{\widetilde{z}} Q$ if P and Q are open bisimilar under the assumption that the names $\tilde{z}$ are not subject to attacks from the environments. In other words, $P \approx_q^{\widetilde{z}} Q$ if P and Q are open bisimilar assuming that the environments can neither replace a name in $\tilde{z}$ by another name nor identify any two names in $\tilde{z}$. So when we assert that $P \approx_q^{\widetilde{z}} Q$, we regard the free names in $fn(P\,|\,Q) \cap \tilde{z}$ as previous local names that have been opened up, as it were. It is easy to see that $P \approx_q^{\widetilde{z}} Q$ implies that $(\widetilde{z})P \approx_q (\widetilde{z})Q$. But the converse implication does not hold.

For open bisimilarities, it is very important that they are closed under substitutions. Thus the next lemma should be stated first.

Lemma 5. *Suppose σ respects $\widetilde{z}$. If $P \approx_q^{\widetilde{z}} Q$ then $P\sigma \approx_q^{\widetilde{z}} Q\sigma$.*

Proof. It is easy to show that the relation $\mathcal{R}^{\widetilde{z}} \stackrel{\text{def}}{=} \{(P\sigma, Q\sigma) \mid P \approx_q^{\widetilde{z}} Q,\ \sigma \text{ respects } \widetilde{z}\}$ is a quasi open bisimulation. □

So the quasi open bisimilarity $\approx_q$ is indeed preserved by substitution. The other closure properties follow from Lemma 5, whose proof is routine and therefore is omitted[1].

Lemma 6. *The relation $\approx_q$ is an equivalence relation and is closed under all but the choice combinator.*

[1] Lemma 6 is misstated in [1].

Next we prove a couple of technical results to be used later on.

Lemma 7. *If* $P \approx_q^{\widetilde{z}x} Q$ *and* $x \notin fn(P \,|\, Q)$ *then* $P \approx_q^{\widetilde{z}} Q$.

Proof. Suppose $P \approx_q^{\widetilde{z}x} Q$ and $x \notin fn(P \,|\, Q)$. If for instance $P\sigma \xrightarrow{\tau} P'$ for some σ respecting $\widetilde{z}$ then $P\sigma' \xrightarrow{\tau} P'$ by assumption, where σ' may differ from σ only in that $\sigma'(x) = x$. Clearly σ' respects $\widetilde{z}x$. It follows from $P \approx_q^{\widetilde{z}x} Q$ that some Q' exists such that $Q\sigma' \Longrightarrow Q' \approx_q^{\widetilde{z}x} P'$. Thus $Q\sigma \Longrightarrow Q' \approx_q^{\widetilde{z}x} P'$ and $x \notin fn(P' \,|\, Q')$. This is enough to show how the bisimulation argument can be completed. □

Corollary 8. *Suppose* $P \approx_q^{\widetilde{z}x} Q$. *Then for each* A *it holds that* $(x)(P \,|\, A) \approx_q^{\widetilde{z}} (x)(Q \,|\, A)$.

4 Open Barbed Bisimulation

This section serves to introduce a result in [15]. The result is to be used in the next section to establish an alternative view of the quasi open bisimilarity.

The barbed bisimulations are proposed in [11] as a general method to define an observational equivalence. The idea is that two processes are equivalent if they can simulate each other's ability to communicate at particular channels. This ability is characterized as a relation between the processes and the channel names. For simplicity we will bypass this relation. The barbed relation defined below is non-standard. But it is equivalent to the standard one ([11]) in the sense that the largest barbed bisimulations they give rise to are the same.

Definition 9. A symmetric relation $\mathcal{R}$ is a barbed bisimulation if whenever $\langle P, Q\rangle \in \mathcal{R}$ then the following properties hold:
(i) If $P \xrightarrow{\tau} P'$ then $Q \Longrightarrow Q'\mathcal{R}P'$ for some Q'.
(ii) If $P \xrightarrow{\lambda} P'$ for $\lambda \neq \tau$ then $Q \overset{\lambda}{\Longrightarrow} Q'$ for some Q'.
The barbed bisimilarity $\approx_b$ is the largest barbed bisimulation.

It is clear that if two processes are barbed bisimilar then they can simulate each other's next observable actions. So the philosophy of the barbed bisimilarity can be stated as follows:

> Two processes are equivalent if no matter what a state one process has evolved into the other process can always reach to a state such that the two processes in the new states can simulate each other's *next* observable actions.

The barbed bisimilarity is too loose to be useful in practice. To refine the relation one must take into account the environments. There are two ways to make use of the environments. One is the static approach. Milner and Sangiorgi used this approach in [11] in their definition of the barbed equivalence.

Definition 10. Two processes P, Q are barbed equivalent, notation $P \approx_b^e Q$, if $C[P] \approx_b C[Q]$ for every context $C[\_]$.

The barbed equivalence is a tricky one. There are proofs, say the ones in [15], of the coincidence of $\approx_b^e$ with the early equivalence under some conditions, but the general coincidence result has not been established. It is easy to show that if $Q \approx_b^e P \xrightarrow{\lambda} P'$ for a non-bound action λ then there is some Q' such that $Q \overset{\widehat{\lambda}}{\Longrightarrow} Q' \approx_b^e P'$. But this bisimulation property has not been established for the bound output actions. What one has proved is that whenever $Q \approx_b^e P \xrightarrow{\overline{a}(x)} P'$ then $Q \overset{\overline{a}(x)}{\Longrightarrow} Q'$ such that $C[P'] \approx_b^e C[Q']$ for some Q' and for all contexts $C[\_]$ that localize x.

The other is the dynamic approach. This is the one adopted in the next definition due to Sangiorgi and Walker, see [15].

Definition 11. An open barbed bisimulation is a barbed bisimulation closed under context. The open barbed bisimilarity $\approx_b^o$ is the largest open barbed bisimulation.

The open barbed bisimilarity is closed under context at every bisimulation step. The property implies that it is closed under substitution as well. This leads to the following result due to Sangiorgi and Walker, see [15].

Proposition 12. *The equivalence relations* $\approx_b^o$ *and* $\approx_q$ *coincide.*

The inclusion $\approx_q \subseteq \approx_b^o$ is easy. For the converse inclusion the intuition is to show that the relations $\mathcal{R}^{\widetilde{x}}$, for finite sequence $\widetilde{x}$ of names, defined below constitute a quasi open bisimulation:

$$\{(P,Q) \mid \forall C[\_] \text{ bounds } \widetilde{x}.C[P] \approx_b^o C[Q]\}$$

As a matter of fact one only has to consider contexts of the special form

$$(x_1)\ldots(x_n)(!\overline{a_1}x_1 \mid \ldots \mid !\overline{a_n}x_n \mid \_)$$

Proposition 12 provides an interesting characterization of the quasi open bisimilarity. The easy proof of this coincidence result is a support to the quasi open bisimulations.

5 Q-Open Bisimulation

A quasi open bisimulation is a family of relations indexed by sets of names. Often a single relation is technically more convenient than a family of relations. This section sets out to provide an alternative characterization to the quasi open bisimulations. This alternative is a single relation rather than a family of indexed relations, satisfying the condition that the largest such relation coincides with the quasi open bisimilarity. The trade-off for a single relation is that we need to introduce a new construction, noted $(\_)^{\widetilde{z}}$, that operates on conditions and processes. The effect of this operation on conditions is defined as follows:

$$
\begin{aligned}
()^{\widetilde{z}} &\stackrel{\text{def}}{=} \top \\
([x{=}y]\psi)^{\widetilde{z}} &\stackrel{\text{def}}{=} \begin{cases} [x{=}y]\psi^{\widetilde{z}}, & \text{if } \widetilde{z}\cap\{x,y\}=\emptyset \\ \psi^{\widetilde{z}}, & \text{if } \exists z\in\widetilde{z}.x=z=y \\ \bot, & \text{otherwise} \end{cases} \\
([x{\neq}y]\psi)^{\widetilde{z}} &\stackrel{\text{def}}{=} \begin{cases} [x{\neq}y]\psi^{\widetilde{z}}, & \text{if } \widetilde{z}\cap\{x,y\}=\emptyset \\ \bot, & \text{if } \exists z\in\widetilde{z}.x=z=y \\ \psi^{\widetilde{z}}, & \text{otherwise} \end{cases}
\end{aligned}
$$

Intuitively the condition $\psi^{\tilde{z}}$ is obtained by removing the matches and the mismatches in ψ that involve a name in $\tilde{z}$. If we think of a name z in $\tilde{z}$ as a special name that can never be replaced by or identified to any other name, then a condition of the form $[x{=}z]\psi'$ or of the form $[z{\neq}z]\psi'$ is as good as $\mathbf{0}$ and a condition of the form $[x{\neq}z]\psi'$ or of the form $[z{=}z]\psi'$ behaves just like ψ'. This explains the clause concerning the match and the mismatch. When the operation $(\_)^{\widetilde{z}}$ applies to an empty list of match/mismatch operators it returns the logical truth. The proof of the next lemma is a simple structural induction.

Lemma 13. *If σ respects $\widetilde{z}$ then $\psi^{\widetilde{z}}\sigma = (\psi\sigma)^{\widetilde{z}}$.*

The structural definition of the operation on processes is given below:

$$
\begin{aligned}
(\mathbf{0})^{\widetilde{z}} &\stackrel{\text{def}}{=} \mathbf{0} \\
(a(x).P)^{\widetilde{z}} &\stackrel{\text{def}}{=} a(x).(\sum_{z\in\widetilde{z}}[x{=}z](P\{z/x\})^{\widetilde{z}} + [x{\notin}\widetilde{z}]P^{\widetilde{z}}), \quad \text{where } x\notin\widetilde{z} \\
(\tau.P)^{\widetilde{z}} &\stackrel{\text{def}}{=} \tau.P^{\widetilde{z}} \\
((x)P)^{\widetilde{z}} &\stackrel{\text{def}}{=} (x)P^{\widetilde{z}}, \quad \text{where } x\notin\widetilde{z} \\
(P{+}Q)^{\widetilde{z}} &\stackrel{\text{def}}{=} P^{\widetilde{z}}{+}Q^{\widetilde{z}} \\
([x{=}y]P)^{\widetilde{z}} &\stackrel{\text{def}}{=} ([x{=}y])^{\widetilde{z}}P^{\widetilde{z}} \\
([x{\neq}y]P)^{\widetilde{z}} &\stackrel{\text{def}}{=} ([x{\neq}y])^{\widetilde{z}}P^{\widetilde{z}} \\
(!P)^{\widetilde{z}} &\stackrel{\text{def}}{=} !P^{\widetilde{z}}
\end{aligned}
$$

To appreciate the clause about the input prefix, one only has to notice that $P^z\{z/x\} \not\equiv (P\{z/x\})^z$. For instance let P be $[z{=}x]\overline{a}a$. Then $P^z\{z/x\}$ is $\mathbf{0}$ whereas $(P\{z/x\})^z$ is $\overline{a}a$. Had we defined $(a(x).P)^{\widetilde{z}}$ as follows:

$$(a(x).P)^{\widetilde{z}} \stackrel{\text{def}}{=} a(x).P^{\widetilde{z}}$$

then $(a(x).[z{=}x]\overline{a}a)^{\widetilde{z}}$ would be $a(x).\mathbf{0}$. This is apparently wrong since the input might receive the name z and the action $\overline{a}a$ can then be fired.

In the above definition we have not considered the parallel composition operator. In the rest of the paper we shall assume that a process is free of composition operator when applying $(\_)^{\widetilde{z}}$ to it. It should be noticed that none of the names in $\widetilde{z}$ appears in any of the match/mismatch in $P^{\widetilde{z}}$.

As a warming-up exercise, we prove the following lemma.

Lemma 14. *Suppose a is fresh. Then $(x)(P\,|\,\overline{a}x) \approx_q (x)(P^x\,|\,\overline{a}x)$.*

Proof. We prove by structural induction that $(x)(C[P]\,|\,\overline{a}x) \approx_q (x)(C[P^x]\,|\,\overline{a}x)$, where $C[\_]$ is a context and x is not bound by $C[]$. We take a look at two cases:

- $P \equiv [u{=}v]P_1$. There are three cases:
  - $x \notin \{u, v\}$. Then

$$\begin{array}{rcl}(x)(C[P]\,|\,\overline{a}x) & \approx_q & (x)(C[[u{=}v]P_1]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[[u{=}v]P_1^x]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[([u{=}v]P_1)^x]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[P^x]\,|\,\overline{a}x)\end{array}$$

  - $u = x = v$. Then

$$\begin{array}{rcl}(x)(C[P]\,|\,\overline{a}x) & \approx_q & (x)(C[[x{=}x]P_1]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[P_1]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[P_1^x]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[([x{=}x]P_1)^x]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[P^x]\,|\,\overline{a}x)\end{array}$$

  - $x \in \{u, v\}$ and $u \neq v$. Then

$$\begin{array}{rcl}(x)(C[P]\,|\,\overline{a}x) & \approx_q & (x)(C[[u{=}v]P_1]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[\mathbf{0}]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[([u{=}v]P_1)^x]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[P^x]\,|\,\overline{a}x)\end{array}$$

- $P \equiv c(z).P_1$. Then

$$\begin{array}{rcl}(x)(C[P]\,|\,\overline{a}x) & \approx_q & (x)(C[c(z).P_1]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[c(z).([z{=}x]P_1 + [z{\neq}x]P_1)]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[c(z).([z{=}x]P_1\{x/z\} + [z{\neq}x]P_1)]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[c(z).([z{=}x](P_1\{x/z\})^x + [z{\neq}x]P_1)]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[c(z).([z{=}x](P_1\{x/z\})^x + [z{\neq}x]P_1^x)]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[(c(z).P_1)^x]\,|\,\overline{a}x)\\ & \approx_q & (x)(C[P^x]\,|\,\overline{a}x)\end{array}$$

This completes the proof. □

We now establish the key property for the operation $(\_)^{\widetilde{z}}$, which states that $P \approx_q^{\widetilde{z}} Q$ if and only if $P^{\widetilde{z}} \approx_q Q^{\widetilde{z}}$. In some sense the operation $(\_)^{\widetilde{z}}$ transfers an external judgement to an internal assertion. Putting in different words, $P \approx_q^{\widetilde{z}} Q$ is a judgement with the assumption on the names $\tilde{z}$ whereas $P^{\widetilde{z}} \approx_q Q^{\widetilde{z}}$ is an assertion made by working out the conditions in P, Q that involve names in $\tilde{z}$. The proof of this property makes use of the following four lemmas whose proofs are routine structural inductions.

Lemma 15. *If σ respects $\widetilde{z}$ then $P^{\widetilde{z}}\sigma \equiv (P\sigma)^{\widetilde{z}}$.*

Lemma 16. *Suppose σ respects $\widetilde{z}$ and λ is not an input prefix. Then the following properties hold:*
(i) If $P\sigma \xrightarrow{\lambda} P_1$ then $P^{\widetilde{z}}\sigma \xrightarrow{\lambda} P_1^{\widetilde{z}}$.
(ii) If $P^{\widetilde{z}}\sigma \xrightarrow{\lambda} P'$ then some P_1 exists such that $P\sigma \xrightarrow{\lambda} P_1$ and $P_1^{\widetilde{z}} \equiv P'$.

Lemma 17. *If $(\psi P)^{\widetilde{z}}\sigma \xrightarrow{\lambda} P'$ for some σ respecting $\widetilde{z}$ then $P^{\widetilde{z}}\sigma \xrightarrow{\lambda} P'$.*

All the three lemmas deal with properties about $P^{\widetilde{z}}\sigma$ if the substitution σ leaves the names $\widetilde{z}$ unchanged and untouched. Lemma 15 says that a $\widetilde{z}$-respecting substitution commutes with the operation $(\_)^{\widetilde{z}}$. Lemma 16 states that if a $\widetilde{z}$-respecting substitution enables an action in P it also enables the same action in $P^{\widetilde{z}}$, and vice versa. Lemma 17 states similar property. All are based on Lemma 13.

Proposition 18. *Suppose $\widetilde{y} \cap \widetilde{z} = \emptyset$. Then $P \approx_q^{\widetilde{y}\widetilde{z}} Q$ if and only if $P^{\widetilde{y}} \approx_q^{\widetilde{z}} Q^{\widetilde{y}}$.*

Proof. Let $\mathcal{R}^{\widetilde{z}}$ be

$$\{(P^{\widetilde{y}}\sigma, Q^{\widetilde{y}}\sigma) \mid P \approx_q^{\widetilde{y}\widetilde{z}} Q \wedge \widetilde{z} \cap \widetilde{y} = \emptyset \wedge (\sigma \text{ respects } \widetilde{z})\}$$

Suppose σ' respects $\widetilde{z}$ and $P^{\widetilde{y}}\sigma\sigma' \xrightarrow{\lambda} P'$. Then $\sigma\sigma'$ respects $\widetilde{z}$. For each $y \in \widetilde{y}$, let y' be a distinct fresh name not in $fn(P\sigma\sigma') \cup fn(Q\sigma\sigma')$. Let σ_1 be defined as follows:

$$\sigma_1(x) \stackrel{\text{def}}{=} \begin{cases} y, & \text{if } x = y \in \widetilde{y} \\ y', & \text{if } (x \notin \widetilde{y}) \wedge (\sigma'(\sigma(x)) = y \in \widetilde{y}) \wedge (x \in fn(P) \cup fn(Q)) \\ \sigma'(\sigma(x)), & \text{if } (x \notin \widetilde{y}) \wedge (\sigma'(\sigma(x)) \notin \widetilde{y}) \end{cases}$$

The substitution σ_1 is modified from $\sigma'\sigma$ in such a way that it respects $\widetilde{z}$ as σ does and that it respects $\widetilde{y}$ even if σ does not. It is clear that $\sigma\sigma'$ and $\sigma_1\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$ coincide on $fn(P) \cup fn(Q)$.

Now suppose $P^{\widetilde{y}}\sigma\sigma' \xrightarrow{\lambda} P'$ is induced by a summand $\psi_i\lambda_i.P_i$ of P. If λ is not an input action then clearly $\lambda = \lambda_i\sigma\sigma'$. By the fact that $\sigma\sigma'$ satisfies the head condition of $(\psi_i\lambda_i.P_i)^{\widetilde{y}}$ and that the head condition of $(\psi_i\lambda_i.P_i)^{\widetilde{y}}$ does not contain any name in $\widetilde{y}$, one infers that σ_1 satisfies ψ_i. Then $P\sigma_1 \xrightarrow{\lambda_i\sigma_1} P_i\sigma_1$. Now λ could be a free output action, a bound output action or a tau action. We consider them in turn:

- λ is a tau action or a free output action. By Lemma 15,

  $$P' \equiv P_i^{\widetilde{y}}\sigma\sigma' \equiv P_i^{\widetilde{y}}\sigma_1\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\} \equiv (P_i\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

  By definition some Q' exists such that $Q\sigma_1 \stackrel{\lambda_i\sigma_1}{\Longrightarrow} Q'\sigma_1 \approx_q^{\widetilde{y}\widetilde{z}} P_i\sigma_1$. This sequence of actions must be induced by a summand $\psi_j\lambda_j.Q_j$ of Q. Now the head condition in $(\psi_i\lambda_i.P_i)^{\widetilde{y}}$ is satisfied by $\sigma\sigma'$. Therefore $Q^{\widetilde{y}}\sigma\sigma' \stackrel{\lambda_i\sigma\sigma'}{\Longrightarrow} (Q'\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$. Hence

  $$(P_i\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}\mathcal{R}^{\widetilde{z}}(Q'\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

  since $\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$ respects $\widetilde{z}$.

- λ is a bound output action $\overline{a}(x)$. By Lemma 15,

  $$P' \equiv P_i^{\widetilde{y}}\sigma\sigma' \equiv P_i^{\widetilde{y}}\sigma_1\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\} \equiv (P_i\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

  By definition some Q' exists such that $Q\sigma_1 \stackrel{\lambda_i\sigma_1}{\Longrightarrow} Q'\sigma_1$ and $P_i\sigma_1 \approx_q^{\widetilde{y}\widetilde{z}x} Q'\sigma_1$. This sequence of actions must be induced by a summand $\psi_j\lambda_j.Q_j$ of Q. Now the head condition in $(\psi_i\lambda_i.P_i)^{\widetilde{y}}$ is satisfied by $\sigma\sigma'$. Therefore $Q^{\widetilde{y}}\sigma\sigma' \stackrel{\lambda_i\sigma\sigma'}{\Longrightarrow} (Q'\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$. Hence

  $$(P_i\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}\mathcal{R}^{\widetilde{z}x}(Q'\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

If λ is an input action aw then $\lambda_i = a_i(x)$, for some a_i and fresh x, and $\lambda = a_i\sigma\sigma'w$. It is clear that

$$P' \equiv P_i^{\widetilde{y}}\sigma\sigma'\{w/x\} \equiv P_i^{\widetilde{y}}\sigma_1\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}\{w/x\} \equiv (P_i\sigma_1)^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

By definition some Q_1, Q_2 exist such that

$$Q\sigma_1 \Longrightarrow\stackrel{(a\sigma_1)w}{\longrightarrow} Q_1\sigma_1\{w/x\} \Longrightarrow Q_2\sigma_1\{w/x\} \approx_q^{\widetilde{y}\widetilde{z}} P_i\sigma_1\{w/x\}$$

Without losing generality we may assume that

$$P^{\widetilde{y}}\sigma_1 \quad \stackrel{(a\sigma_1)w}{\longrightarrow} \quad (\sum_{y\in\widetilde{y}}[x{=}y](P_i\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]P_i^{\widetilde{y}})\sigma_1\{w/x\}$$

and

$$Q^{\widetilde{y}}\sigma_1 \quad \Longrightarrow\stackrel{(a\sigma_1)w}{\longrightarrow} \quad (\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\sigma_1\{w/x\}$$

To proceed notice that there are two subcases:

- $w = y \in \widetilde{y}$. Then

$$\begin{aligned}(\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\sigma_1\{w/x\} \quad &\equiv \quad \ldots + [x{=}y](Q_1\{y/x\})^{\widetilde{y}}\sigma_1\{w/x\} + \ldots \\ &\equiv \quad \ldots + [y{=}y]Q_1^{\widetilde{y}}\sigma_1\{w/x\} + \ldots \\ &\Longrightarrow \quad Q_2^{\widetilde{y}}\sigma_1\{w/x\}\end{aligned}$$

  because x is a bound name. It follows that

$$Q^{\widetilde{y}}\sigma\sigma' \quad \Longrightarrow\stackrel{(a\sigma_1)w}{\longrightarrow} \quad (\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\sigma\sigma'\{w/x\}$$

  and

$$(\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\sigma\sigma'\{w/x\} \Longrightarrow (Q_2\sigma_1\{w/x\})^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

- $w \notin \widetilde{y}$. Then

$$\begin{aligned}(\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\sigma_1\{w/x\} \quad &\equiv \quad \ldots + [x{=}y](Q_1\{y/x\})^{\widetilde{y}}\sigma_1\{w/x\} + \ldots \\ &\equiv \quad \ldots + [w{\notin}\widetilde{y}]Q_1^{\widetilde{y}}\sigma_1\{w/x\} \\ &\Longrightarrow \quad Q_2^{\widetilde{y}}\sigma_1\{w/x\}\end{aligned}$$

  So in this case we also have

$$Q^{\widetilde{y}}\sigma\sigma' \quad \stackrel{(a\sigma_1)w}{\Longrightarrow} \quad (Q_2\sigma_1\{w/x\})^{\widetilde{y}}\{\widetilde{y}/\widetilde{y'}, \widetilde{y}\sigma\sigma'/\widetilde{y}\}$$

We conclude that $\mathcal{R}^{\widetilde{z}}$ is a quasi open bisimulation. Consequently $P \approx_q^{\widetilde{y}\widetilde{z}} Q$ implies $P^{\widetilde{y}} \approx_q^{\widetilde{z}} Q^{\widetilde{y}}$.

To prove the reverse implication, let $\mathcal{S}^{\widetilde{y}\widetilde{z}}$ be $\{(P,Q) \mid P^{\widetilde{y}} \approx_q^{\widetilde{z}} Q^{\widetilde{y}} \wedge \widetilde{z} \cap \widetilde{y} = \emptyset\}$. Suppose $P\mathcal{S}^{\widetilde{y}\widetilde{z}}Q$ and σ respects $\widetilde{y}\widetilde{z}$. Suppose further that $P\sigma \stackrel{\lambda}{\longrightarrow} P_1$. Then $P^{\widetilde{y}}\sigma \stackrel{\lambda}{\longrightarrow} P_1^{\widetilde{y}}$.

- λ is a tau action or a free output action. Since $P^{\widetilde{y}} \approx_q^{\widetilde{z}} Q^{\widetilde{y}}$ one has some Q_1 such that $Q^{\widetilde{y}}\sigma \stackrel{\lambda}{\Longrightarrow} Q_1^{\widetilde{y}} \approx_q^{\widetilde{z}} P_1^{\widetilde{y}}$. Now $Q\sigma \stackrel{\lambda}{\Longrightarrow} Q_1$ because σ respects $\widetilde{y}$. Also $P_1\mathcal{S}^{\widetilde{y}\widetilde{z}}Q_1$ by definition.
- λ is an input action aw and $\lambda_i = a_i(x)$, for some a_i, x, and $\lambda = a_i\sigma\sigma'w$. Some Q_1 exists such that

$$Q^{\widetilde{y}}\sigma \Longrightarrow\stackrel{\lambda}{\longrightarrow} (\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\{w/x\}$$

  and

$$(\sum_{y\in\widetilde{y}}[x{=}y](Q_1\{y/x\})^{\widetilde{y}} + [x{\notin}\widetilde{y}]Q_1^{\widetilde{y}})\{w/x\} \Longrightarrow Q_2^{\widetilde{y}}\{w/x\} \approx_q^{\widetilde{z}} P_1^{\widetilde{y}}\{w/x\}$$

  It follows that $Q\sigma \Longrightarrow\stackrel{\lambda}{\longrightarrow} Q_1\{w/x\}$, $Q_1\{w/x\} \Longrightarrow Q_2\{w/x\}$ and $(P_1\{w/x\})^{\widetilde{y}} \approx_q^{\widetilde{z}} (Q_2\{w/x\})^{\widetilde{y}}$. Thus $P_1\{w/x\}\mathcal{S}^{\widetilde{y}\widetilde{z}}Q_2\{w/x\}$.

- λ is a bound output action $\overline{a}(x)$. Some Q_1 exists such that $Q^{\widetilde{y}}\sigma \stackrel{\lambda}{\Longrightarrow} Q_1^{\widetilde{y}}$ and $P_1^{\widetilde{y}} \approx_q^{\widetilde{z}x} Q_1^{\widetilde{y}}$. Now $Q\sigma \stackrel{\lambda}{\Longrightarrow} Q_1$ because σ respects $\widetilde{y}$ and $P_1 \mathcal{S}^{\widetilde{y}\widetilde{z}x} Q_1$ by definition.

This completes the proof. □

The lemma says that $()^{\widetilde{z}}$ is an internalization operator as it were. Using this operator one can give an alternative characterization of the quasi open bisimilarity. Now in Definition 4 only the clause on the restricted output actions introduces the indexes on the quasi open bisimulations. According to Proposition 18, the indexed equivalence $P \approx_q^{\widetilde{z}} Q$ can be transposed to a non-indexed equivalence $P^{\widetilde{z}} \approx_q Q^{\widetilde{z}}$. These observations suggest to use the operation $()^{\widetilde{z}}$ in place of the index $\widetilde{z}$ in the definition of the quasi open bisimulations. The selling point of the new definition is that it introduces a single relation rather than an indexed family of relations. This has advantage when it comes to axiomatization.

Definition 19. Suppose $\mathcal{R}$ is a symmetric binary relation closed under substitution. The relation $\mathcal{R}$ is a q-open bisimulation if the following properties hold whenever $P\mathcal{R}Q$:
(i) If $P \stackrel{\lambda}{\longrightarrow} P_1$, where λ is not a bound output, then some Q_1 exists such that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q_1 \mathcal{R} P_1$.
(ii) If $P \stackrel{\overline{a}(x)}{\longrightarrow} P_1$ then some Q_1 exists such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q_1$ and $P_1^x \mathcal{R} Q_1^x$.
The q-open bisimilarity $\approx_{qo}$ is the largest q-open bisimulation.

The following proposition enforces the above definition.

Proposition 20. *$P \approx_q^{\widetilde{z}} Q$ if and only if $P^{\widetilde{z}} \approx_{qo} Q^{\widetilde{z}}$.*

Proof. (i) By Lemma 5, $\approx_q^l$ is closed under substitution. Suppose $P \approx_q^l Q$. If $P \stackrel{\overline{a}(x)}{\longrightarrow} P_1$ then Q_1 exists such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q_1$ and $P_1 \approx_q^x Q_1$. Using Proposition 18, one has that $P_1^x \approx_q Q_1^x$. It follows that $\approx_q^l$ is a q-open bisimulation, which implies that $\approx_q^l \subseteq \approx_{qo}$. By Proposition 18, $P \approx_q^{\widetilde{z}} Q$ implies $P^{\widetilde{z}} \approx_q^l Q^{\widetilde{z}}$, which in turn implies $P^{\widetilde{z}} \approx_{qo} Q^{\widetilde{z}}$.

(ii) Conversely $\{\mathcal{R}_{\widetilde{z}}\}_{\widetilde{z}}$, where $\mathcal{R}_{\widetilde{z}} \stackrel{\text{def}}{=} \{(P,Q) \mid P^{\widetilde{z}} \approx_{qo} Q^{\widetilde{z}}\}$, is a quasi open bisimulation. It is clear that $\mathcal{R}_{\widetilde{z}}$ is closed under substitution respecting $\widetilde{z}$. So we do not have to consider substitutions in the following argument. Now suppose $P\mathcal{R}_{\widetilde{z}}Q$.

- If $P \stackrel{\overline{a}(x)}{\longrightarrow} P_1$ then $P^{\widetilde{z}} \stackrel{\overline{a}(x)}{\longrightarrow} P_1^{\widetilde{z}}$ by Lemma 16. Since $P^{\widetilde{z}} \approx_{qo} Q^{\widetilde{z}}$, there must exist some Q_1 such that $Q^{\widetilde{z}} \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $P_1^{\widetilde{z}x} \approx_{qo} (Q')^x$. By Lemma 16, $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q_1$ for some Q_1 such that $Q_1^{\widetilde{z}} \equiv Q'$. It follows that $P_1 \mathcal{R}_{\widetilde{z}x} Q_1$.
- If $P \stackrel{aw}{\longrightarrow} P_1'$ then, by the definition given on page 7, one has, without losing generality, that

$$P^{\widetilde{z}} \stackrel{aw}{\longrightarrow} P' \equiv (\sum_{z\in\widetilde{z}}[x{=}z](P_1\{z/x\})^{\widetilde{z}} + [x\notin\widetilde{z}]P_1^{\widetilde{z}})\{w/x\}$$

  for some P_1 such that $P_1' \equiv P_1\{w/x\}$. Since $P^{\widetilde{z}} \approx_{qo} Q^{\widetilde{z}}$, there must exist some Q' such that $Q^{\widetilde{z}} \Longrightarrow\stackrel{aw}{\longrightarrow} Q' \Longrightarrow Q''$ and $P' \approx_{qo} Q''$. Again by the definition given on page 7, Q' must be of the form

$$(\sum_{z\in\widetilde{z}}[x{=}z](A\{z/x\})^{\widetilde{z}} + [x\notin\widetilde{z}]A^{\widetilde{z}})\{w/x\}$$

  Then clearly $Q \Longrightarrow\stackrel{aw}{\longrightarrow} A\{w/x\}$. By Lemma 16, some A_1 exists such that $A\{w/x\} \Longrightarrow A_1$ and $A_1^{\widetilde{z}} \equiv Q''$. We need to show that $A_1 \mathcal{R}_{\widetilde{z}} P_1'$, which is the same as to showing that $A_1^{\widetilde{z}} \approx_{qo} (P_1')^{\widetilde{z}}$. Since $P' \approx_{qo} Q''$ and $A_1^{\widetilde{z}} \equiv Q''$, it amounts to showing that

$$\sum_{z\in\widetilde{z}}[w{=}z](P_1\{z/x\})^{\widetilde{z}} + [y\notin\widetilde{z}]P_1^{\widetilde{z}}\{w/x\} \approx_{qo} (P_1')^{\widetilde{z}} \tag{3}$$

  If we can prove that

$$[w\notin\widetilde{z}]P_1^{\widetilde{z}}\{w/x\} \approx_{qo} [w\notin\widetilde{z}](P_1\{w/x\})^{\widetilde{z}} \tag{4}$$

  and

$$[w{=}z](P_1\{z/x\})^{\widetilde{z}} \approx_{qo} [w{=}z](P_1\{w/x\})^{\widetilde{z}} \tag{5}$$

for each $z \in \widetilde{z}$, then (3) is derivable from

$$(P_1\{w/x\})^{\widetilde{z}} \approx_{qo} (P_1')^{\widetilde{z}}$$

To see (4) observe that if $w \in \widetilde{z}$ then both $[w \notin \widetilde{z}]P_1^{\widetilde{z}}\{w/x\}$ and $[w \notin \widetilde{z}](P_1\{w/x\})^{\widetilde{z}}$ are bisimilar to $\mathbf{0}$; if $w \notin \widetilde{z}$ then $[w \notin \widetilde{z}]P_1^{\widetilde{z}}\{w/x\} \equiv [w \notin \widetilde{z}](P_1\{w/x\})^{\widetilde{z}}$ by Lemma 15. For (5) we only have to consider the situation when $w = z$. In this case it is obvious that

$$[w{=}z](P_1\{z/x\})^{\widetilde{z}} \equiv [w{=}z](P_1\{w/x\})^{\widetilde{z}}$$

Remember that we only have to consider substitutions that respect $\widetilde{z}$.

- Other cases are simpler.

This completes the proof. □

Corollary 21. *The two relations $\approx_q$, $\approx_{qo}$ coincide.*

6 Local Open Bisimulation

One objection to the standard early and late bisimilarities, as well as the open bisimilarity, is that the restricted output actions are treated in completely the same manner as the other observable actions. From the viewpoint of bisimulation equivalence, a free output action of one process must be simulated by the same free output action of an equivalent process in order for the two processes to exert the same observable effect on an environment. For bound output actions, the situation is slightly different. Since in π-calculus a local name will remain local forever, it is too much to require that the action $P \stackrel{\overline{a}(x)}{\longrightarrow} P'$ be simulated by $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ for equivalent P and Q. In other words it is too much to say that P' and Q' should be equivalent. The most we could ask for is that P' and Q' to be equivalent in any context in which the name x is localized. This brings us to the following definition.

Definition 22. A symmetric relation $\mathcal{R}$ on π-processes is a local open bisimulation if it is closed under substitution and whenever $P\mathcal{R}Q$ then the following properties hold:
(i) If $P \stackrel{\lambda}{\longrightarrow} P'$, where λ is not a bound output, then some Q' exists such that $Q \stackrel{\lambda}{\Longrightarrow} Q'$ and $P'\mathcal{R}Q'$;
(ii) If $P \stackrel{\overline{a}(x)}{\longrightarrow} P'$ then, for each process A, some Q' exists such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(x)(P' \,|\, A)\mathcal{R}(x)(Q' \,|\, A)$.
The local open bisimilarity $\approx_{lo}$ is the largest local open bisimulation.

Definition 22 proposes yet another treatment of the restricted output actions: In the simulation $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ of $P \stackrel{\overline{a}(x)}{\longrightarrow} P'$ the equivalence between P' and Q' is tested in 'local contexts' of the form $(x)(\_ \,|\, A)$. This is more reasonable than requiring that P' and Q' be tested in all contexts.

Like the quasi open bisimilarity, the local open bisimilarity $\approx_{lo}$ enjoys the following closure property.

Lemma 23. *The relation $\approx_{lo}$ is closed under prefix, composition, match and restriction operations.*

Proof. It is easy to see that the relation is closed under the prefix and match operations. The proofs for the parallel composition and restriction operations are carried out at one go. Let $\mathcal{R}$ be

$$\{((\tilde{x})(P \,|\, R), (\tilde{x})(Q \,|\, R)) \mid P \approx_{lo} Q\}$$

We show that $\mathcal{R}$ is a local bisimulation up to $\sim$. We take a look at a few cases:

- If λ is a tau action, an input action or a free output action and $(\tilde{x})(P \,|\, R) \stackrel{\lambda}{\longrightarrow} (\tilde{x'})(P' \,|\, R')$ is caused either by $P \stackrel{\lambda}{\longrightarrow} P'$ or by $R \stackrel{\lambda}{\longrightarrow} R'$, then the simulation is simple.

- $(\tilde{x})(P \,|\, R) \stackrel{\overline{a}(y)}{\longrightarrow} (\tilde{x})(P \,|\, R')$ is caused by $R \stackrel{\overline{a}(y)}{\longrightarrow} R'$. Then $(\tilde{x})(Q \,|\, R) \stackrel{\overline{a}(y)}{\longrightarrow} (\tilde{x})(Q \,|\, R')$ and for each process A it holds that

$$(y)((\tilde{x})(P \,|\, R') \,|\, A) \sim (\tilde{x})(P \,|\, (y)(R' \,|\, A)) \;\mathcal{R}\; (\tilde{x})(Q \,|\, (y)(R' \,|\, A)) \sim (y)((\tilde{x})(Q \,|\, R') \,|\, A)$$

- $(\tilde{x})(P\,|\,R) \xrightarrow{\overline{a}(x)} (\tilde{x'})(P\,|\,R')$ is caused by $R \xrightarrow{\overline{a}x} R'$ such that $x \in \tilde{x}$. Then the simulation is matched up by $(\tilde{x})(Q\,|\,R) \xrightarrow{\overline{a}(x)} (\tilde{x'})(Q\,|\,R')$ and, for each process A,

$$(x)((\tilde{x'})(P\,|\,R')\,|\,A) \sim (\tilde{x})(P\,|\,(R'\,|\,A))\ \mathcal{R}\ (\tilde{x})(Q\,|\,(R'\,|\,A)) \sim (x)((\tilde{x'})(Q\,|\,R')\,|\,A)$$

- $(\tilde{x})(P\,|\,R) \xrightarrow{\overline{c}(y)} (\tilde{x})(P'\,|\,R)$ is caused by $P \xrightarrow{\overline{c}(y)} P'$. Then, for each process A, some Q' exists such that $Q \overset{\overline{c}(y)}{\Longrightarrow} Q'$ and $(y)(P'\,|\,A) \approx_{lo} (y)(Q'\,|\,A)$. Clearly $(\tilde{x})(Q\,|\,R) \overset{\overline{c}(y)}{\Longrightarrow} (\tilde{x})(Q'\,|\,R)$ and

$$(y)((\tilde{x})(P'\,|\,R)\,|\,A) \sim (\tilde{x})((y)(P'\,|\,A)\,|\,R)\ \mathcal{R}\ (\tilde{x})((y)(Q'\,|\,A)\,|\,R) \sim (y)((\tilde{x})(Q'\,|\,R)\,|\,A)$$

- $(\tilde{x})(P\,|\,R) \xrightarrow{\overline{c}(x)} (\tilde{x'})(P'\,|\,R)$ is caused by $P \xrightarrow{\overline{c}x} P'$ such that $x \in \tilde{x}$. Then some Q' exists such that $Q \overset{\overline{c}x}{\Longrightarrow} Q' \approx_{lo} P'$. Therefore $(\tilde{x})(Q\,|\,R) \overset{\overline{c}(x)}{\Longrightarrow} (\tilde{x'})(Q'\,|\,R)$ and, for each process A,

$$(x)((\tilde{x'})(P'\,|\,R)\,|\,A) \sim (\tilde{x})(P'\,|\,(R\,|\,A))\ \mathcal{R}\ (\tilde{x})(Q'\,|\,(R\,|\,A)) \sim (x)((\tilde{x'})(Q'\,|\,R)\,|\,A)$$

The proof is completed. □

It appears from Definition 22 that it is very untractable to check if two processes are local open bisimilar. The universal quantification over all processes in the fourth clause of the definition is a minus for the equivalence. As a matter of fact the local open bisimilarity is not that untractable. The following lemma is suggestive.

Lemma 24. *If $(x)(P\,|\,\overline{a}x) \approx_{lo} (x)(Q\,|\,\overline{a}x)$ for $a \notin fn(P\,|\,Q)$ then $(x)(P\,|\,A) \approx_{lo} (x)(Q\,|\,A)$ for each A.*

Proof. Let σ be a substitution and let σ' be defined as follows:

$$\sigma'(x) \overset{\text{def}}{=} \begin{cases} \sigma(x), & x \neq a \\ x, & x = a \end{cases}$$

Then $(x)(P\,|\,\overline{a}x)\sigma' \approx_{lo} (x)(Q\,|\,\overline{a}x)\sigma'$. Hence $(x)(P\sigma'\,|\,\overline{a}x) \approx_{lo} (x)(Q\sigma'\,|\,\overline{a}x)$. Thus $(x)(P\sigma\,|\,\overline{a}x) \approx_{lo} (x)(Q\sigma\,|\,\overline{a}x)$. Assume that A is a process. Let b, z be fresh names. According to Lemma 23 one has

$$(x)(P\sigma\,|\,\overline{a}x)\,|\,a(x).\overline{b}z.A\sigma \approx_{lo} (x)(Q\sigma\,|\,\overline{a}x)\,|\,a(x).\overline{b}z.A\sigma$$

Observe that

$$\begin{aligned} (x)(P\sigma\,|\,\overline{a}x)\,|\,a(x).\overline{b}z.A\sigma &\xrightarrow{\tau} (x)((P\sigma\,|\,\mathbf{0})\,|\,\overline{b}z.A\sigma) \\ &\xrightarrow{\overline{b}z} (x)((P\sigma\,|\,\mathbf{0})\,|\,A\sigma) \end{aligned}$$

In order to match these actions there must be some Q' such that

$$(x)(Q\sigma\,|\,\overline{a}x)\,|\,a(x).\overline{b}z.A\sigma \overset{\overline{b}z}{\Longrightarrow} (x)((Q'\,|\,\mathbf{0})\,|\,A') \approx_{lo} (x)((P\sigma\,|\,\mathbf{0})\,|\,A\sigma)$$

which can obviously be rearranged as

$$\begin{aligned} (x)(Q\sigma\,|\,\overline{a}x)\,|\,a(x).\overline{b}z.A\sigma &\xrightarrow{\tau} (x)((Q\sigma\,|\,\mathbf{0})\,|\,\overline{b}z.A\sigma) \\ &\xrightarrow{\overline{b}z} (x)((Q\sigma\,|\,\mathbf{0})\,|\,A\sigma) \\ &\Longrightarrow (x)((Q'\,|\,\mathbf{0})\,|\,A') \end{aligned}$$

Similarly some P', A'' exist such that

$$(x)((P\sigma\,|\,\mathbf{0})\,|\,A\sigma) \Longrightarrow (x)((P'\,|\,\mathbf{0})\,|\,A'') \approx_{lo} (x)((Q\sigma\,|\,\mathbf{0})\,|\,A\sigma)$$

It follows easily that

$$(x)(P\,|\,A)\sigma \Longrightarrow (x)(P'\,|\,A'') \approx_{lo} (x)(Q\,|\,A)\sigma$$
$$(x)(Q\,|\,A)\sigma \Longrightarrow (x)(Q'\,|\,A') \approx_{lo} (x)(P\,|\,A)\sigma$$

Using this property it is easy to see that $\{((x)(P\,|\,A)\sigma, (x)(Q\,|\,A)\sigma)\} \cup \approx_{lo}$ is a local open bisimulation. □

In view of Lemma 24, it is tempting to 'simplify' Definition 22 in the following manner:

A symmetric relation $\mathcal{R}$ on π-processes is an l-bisimulation if whenever $P\mathcal{R}Q$ then the following properties hold:
(i) If $P \xrightarrow{\lambda} P'$ then some Q' exists such that $Q \stackrel{\lambda}{\Longrightarrow} Q'$ and $P'\mathcal{R}Q'$;
(ii) If $P \xrightarrow{\overline{a}(x)} P'$ then, for each fresh name b, some Q' exists such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(x)(P' \,|\, \overline{b}x)\mathcal{R}(x)(Q' \,|\, \overline{b}x)$.
The l-bisimilarity $\approx_l'$ is the largest l-bisimulation.

It turns out that the relation $\approx_l'$ is not a good one. It identifies for example $(x)(\overline{a}x \,|\, \overline{b}x)$ and $(x)\overline{a}x \,|\, (x)\overline{b}x$ and is therefore not closed under composition. It is clear that, from an observational point of view, the process $a(v).b(w).(v(z).z(u) \,|\, \overline{w}c)$, where c is fresh, is able to detect the difference between the two processes.

Another implication of Lemma 24 is that if we replace the fourth requirement of Definition 22 by the following stronger requirement (iv'), then we would obtain the same local bisimilarity.

(iv') If $P \xrightarrow{\overline{a}(x)} P'$ then some Q' exists such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(x)(P' \,|\, A)\mathcal{R}(x)(Q' \,|\, A)$ for every process A.

The reason could be given as follows: Suppose $P \approx_{lo} Q$ and $P \xrightarrow{\overline{a}(x)} P'$. For a fresh name b, there must be some Q' such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ and $(x)(P' \,|\, \overline{b}x) \approx_{lo} (x)(Q' \,|\, \overline{b}x)$. Now Lemma 24 implies that $(x)(P' \,|\, A) \approx_{lo} (x)(Q' \,|\, A)$ actually holds for every process A. So $\approx_{lo}$ is a relation that satisfies (iv'). On the other hand the reverse inclusion is obvious by definition.

Now we show that the local open bisimilarity is another characterization of the quasi open bisimilarity.

Theorem 25. *The relations $\approx_{lo}$ and $\approx_q$ coincide.*

Proof. Suppose $P \approx_q Q$ and $P \xrightarrow{\overline{a}(x)} P_1$. Then Q_1 exists such that $Q \stackrel{\overline{a}(x)}{\Longrightarrow} Q_1$ and $P_1 \approx_q^x Q_1$. By Lemma 8,

$$(x)(P_1 \,|\, A) \approx_q (x)(Q_1 \,|\, A)$$

holds for every A. This should be enough to convince the reader that $\approx_q$ is a local open bisimulation.

Now let $\mathcal{R}_{x_1 \ldots x_n}$ be

$$\left\{ (P,Q) \,\middle|\, \begin{array}{l} (x_1 \ldots x_n)(P \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \approx_{lo} (x_1 \ldots x_n)(Q \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \\ \text{for some } a_1, \ldots, a_n \notin fn(P \,|\, Q) \end{array} \right\}$$

Suppose σ respects $x_1 \ldots x_n$. Notice that $(\widetilde{x})(P \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)\sigma \equiv (\widetilde{x})(P\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$. Three cases need be considered:

- $P\sigma \xrightarrow{\lambda} P'$ where λ is a τ, or an input action, or a free output action and $n(\lambda) \cap \widetilde{x} = \emptyset$. Then

  $$(\widetilde{x})(P\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \xrightarrow{\lambda} (\widetilde{x})(P' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$$

  It follows that Q' exists such that

  $$\begin{array}{rcl} (\widetilde{x})(Q\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) & \stackrel{\lambda}{\Longrightarrow} & (\widetilde{x})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \\ & \approx_{lo} & (\widetilde{x})(P' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \end{array}$$

  Then $Q\sigma \stackrel{\lambda}{\Longrightarrow} Q'\mathcal{R}_{x_1 \ldots x_n}P'$.

- $P\sigma \xrightarrow{\overline{a}x_i} P'$ where $i \in \{1, \ldots, n\}$. Then

  $$(\widetilde{x})(P\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \xrightarrow{\overline{a}(x_i)} (\widetilde{x'})(P' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$$

  where $\widetilde{x'}$ is obtained from $\widetilde{x}$ by removing x_i. It follows that Q' exists such that

  $$(\widetilde{x})(Q\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \stackrel{\overline{a}(x_i)}{\Longrightarrow} (\widetilde{x'})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$$

  and

  $$(\widetilde{x})(P' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \approx_{lo} (\widetilde{x})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$$

  Then $Q\sigma \stackrel{\overline{a}(x_i)}{\Longrightarrow} Q'\mathcal{R}_{x_1 \ldots x_n}P'$.

- $P\sigma \xrightarrow{\overline{c}(x)} P'$. Then
$$(\widetilde{x})(P \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \xrightarrow{\overline{c}(x)} (\widetilde{x})(P_1 \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \tag{6}$$
By definition some Q' exists such that
$$(\widetilde{x})(Q\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \stackrel{\overline{c}(x)}{\Longrightarrow} (\widetilde{x})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$$
and for fresh a
$$(x)((\widetilde{x})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \,|\, \overline{a}x) \approx_{lo} (x)((\widetilde{x})(P' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \,|\, \overline{a}x)$$
Consequently
$$(x)(\widetilde{x})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n \,|\, \overline{a}x) \approx_{lo} (x)(\widetilde{x})(P' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n \,|\, \overline{a}x)$$
It follows that $Q\sigma \stackrel{\overline{c}(x)}{\Longrightarrow} Q'\mathcal{R}_{x_1\ldots x_n x}P'$. Notice that (6) can not be matched up by
$$(\widetilde{x})(Q\sigma \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n) \stackrel{\overline{c}(x)}{\Longrightarrow} (\widetilde{x'})(Q' \,|\, \overline{a_1}x_1 \,|\, \ldots \,|\, \overline{a_n}x_n)$$
induced by a free output action $Q\sigma \stackrel{\overline{c}x}{\Longrightarrow} Q'$.

So $\mathcal{R}$ is a quasi open bisimulation. □

7 Complete System

An algebraic investigation of an equivalence relation would be incomplete without discussing the axiomatization aspect of the corresponding congruence. Given an observational equivalence $\approx$ one of the most interesting questions to ask is if there exists an equational axiomatic system that is both sound and complete for $\approx$. In general one should not expect that a complete system works for all processes due to the limitation of Turing computability. The restrictions to the finite processes ([9]) or the finite control processes ([7, 8, 5, 6]) are imposed in order to obtain a completeness result.

In this section we take a look at the equational axioms for the quasi open congruence. The main result is a complete system for the congruence. This system differs from the system for the weak open congruence [2] in that it has an additional law S6. The line of our investigation follows closely the one advocated in [2]. We refer the reader to [2] for a detailed discussion on the axiomatization problems of the π-calculus. We begin with some comments on some of the laws for the π-calculus:

- Milner, Parrow and Walker proposed, among others, two well known types of equivalences for the π-processes, the early equivalence and the late equivalence. The difference between the two can be seen from the following fact: $a(x).P{+}a(x)$ is early equivalent to $a(x).P{+}a(x){+}a(x).[x{=}y]P$; but the two are not late equivalent. This is because in the late semantics the action
$$a(x).P{+}a(x){+}a(x).[x{=}y]P \xrightarrow{a(x)} [x{=}y]P$$
can not be simulated by any action from $a(x).P{+}a(x).Q$. In the presence of the mismatch operator this pair can be generalized to
$$a(x).P{+}a(x).Q$$
and
$$a(x).P{+}a(x).Q{+}a(x).([x{=}y]P{+}[x{\neq}y]Q)$$
They are early equivalent but not late equivalent. For the *weak open* bisimilarities, there is also a difference between the early and the late approaches. This early/late dichotomy does not appear in the π-calculus without the mismatch operator. But it does appear in the π-calculus with the mismatch operator, as pointed out by the present author and Yang in [2]. A pair that are early open bisimilar but not late open bisimilar are the following π-processes:
$$a(x).[x{=}y]\tau.P + a(x).[x{\neq}y]\tau.P$$

and

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

This is because in the early framework the action $a(x).[x{=}y]\tau.P{+}a(x).[x{\neq}y]\tau.P{+}a(x).P \xrightarrow{a(x)} P$ can be simulated either by the summand $a(x).[x{=}y]\tau.P$ or by the summand $a(x).[x{\neq}y]\tau.P$, depending on whether x is y or not. Notice the important role played by the tau prefix. The quasi open bisimilarity of this paper is the early version.

- A combinator that causes a major concern is the localization operator. Take for instance the π-process $\overline{a}(x).(P{+}[x{=}y]Q)$. Intuitively this process is equivalent to $\overline{a}(x).P$ since y will never be identified to x. From an algebraic point of view, one needs equational law(s) that could support the inference of such equalities. It is difficult to conceive that there are simple structural laws to fulfill this. For instance the localization operator (x) in $(x)\overline{a}x.C[[x{=}y]P]$ could not be pushed inside the output prefix $\overline{a}x$ but the match operator $[x{=}y]$ could be very deep inside the context $C[\_]$. So obviously the following axiom

$$(x)[x{=}y]P = \mathbf{0} \tag{7}$$

where $x \neq y$, is far from sufficient. What we have come up with is the following schematic law:

$$(x)C[[x{=}y]P] = (x)C[\mathbf{0}] \tag{8}$$

where neither x nor y is bound in $C[]$ and $x \neq y$. In the full π-calculus this law is equivalent to the following law

$$(x)C[[x{\neq}y]P] = (x)C[P] \tag{9}$$

under the same condition, the proof of which can be found in [2].

- Since the quasi open bisimilarity is looser than the open bisimilarity, there must be equalities that are valid for the former but are invalid for the latter. We are looking for a law or laws that could characterize the difference between the two equivalences. For this purpose we propose the following axiom:

$$(y)C[a(x).P] = (y)C[a(x).([x{=}y]P{+}[x{\neq}y]P^y)] \tag{10}$$

where P is composition free. Here the most important thing to notice is that y is local. When it is the time for the input prefix $a(x)$ to act, it may receive a name z. After the action the subprocess $a(x).P$ becomes $P\{z/x\}$ and the subprocess $a(x).([x{=}y]P{+}[x{\neq}y]P^y)$ becomes

$$[z{=}y]P\{z/x\} + [z{\neq}y]P^y\{z/x\} \tag{11}$$

There are two cases:

  - $z = y$. Then (11) is actually $[y{=}y]P\{y/x\} + [y{\neq}y]P^y\{y/x\}$, which is equivalent to $P\{z/x\}$.
  - $z \neq y$. In this case (11) is equivalent to $P^y\{z/x\}$ using (8) and (9). Here we have used in an essential way that y is local. But if y is local then P^y is operationally the same as P.

So (10) is sound for the quasi open bisimilarity.

- In process algebra Miler's three tau laws are well known for their role in characterizing the observational aspect of the equivalences. Without paying any attention to the match/mismatch operators, these laws are as follows:

$$\pi.\tau.P = \pi.P \tag{12}$$
$$P{+}\tau.P = \tau.P \tag{13}$$
$$\pi.(P{+}\tau.Q) = \pi.(P{+}\tau.Q) + \pi.Q \tag{14}$$

In the presence of the mismatch operator the law (14) should be generalized to

$$\pi.(P{+}\psi\tau.Q) = \pi.(P{+}\psi\tau.Q) + \psi\pi.Q \tag{15}$$

Now (14) and (15) are equivalent in the equational systems that admit the following two laws:

$$[x{=}y]\pi.P = [x{=}y]\pi.[x{=}y]P \tag{16}$$
$$[x{\neq}y]\pi.P = [x{\neq}y]\pi.[x{\neq}y]P \tag{17}$$

For the early and the late equivalences both (16) and (17) are valid. But for the open bisimilarities (17) is not admissible. Another complication that the open bisimilarities add to the algebraic theory is that Milner's three tau laws are not sufficient to obtain a completeness result. It is discovered by Fu in [2] that the following tau law is crucial:

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

For the π-calculus with the mismatch operator, the above law should be generalized to

$$\tau.P = \tau.(P+\psi\tau.P) \tag{18}$$

For the early and the late equivalences the above tau laws are derivable. In the open semantics it has been shown indispensable in some special case [2]. In more complicated languages, the establishment of such a negative result has not been achieved. The most complex tau law used in this paper is the following one, where I is a finite indexing set and $x \notin n(\psi)$

$$\sum_{i\in I} a(x).(P_i + \psi_i\tau.Q) \quad = \quad \sum_{i\in I} a(x).(P_i + \psi_i\tau.Q) + \psi a(x).Q \tag{19}$$

The above axiom holds under the condition:

$$\bigvee_{i\in I} \psi_i \Leftrightarrow \psi$$

The point is that ψ_i might contain the bound name x but ψ does not. The intuition behind (19) is this: Suppose the substitution σ validates ψ. Then $\bigvee_{i\in I} \psi_i\sigma \Leftrightarrow \top$. If $\psi\sigma a\sigma(x).Q\sigma$ receives the name y through communication, then $\bigvee_{i\in I} \psi_i\sigma\{y/x\} \Leftrightarrow \top$ and therefore there must be some $i \in I$ such that $\psi_i\sigma\{y/x\} \Leftrightarrow \top$. So the summand $(a(x).(P_i + \psi_i\tau.Q))\sigma$ can put into action.

We are now in a position to present the equational system. The basic laws for the quasi open congruence are summarized in Figure 1. These laws are valid for the *strong* open congruence. We will refer to this set of laws as AS. Apart from L8 and S6, the laws of AS have appeared in [12]. Some derived laws of AS are stated in Figure 2. The proofs that they are derivable can be found in [10, 13, 2]. Figure 3 lists all the tau laws used in this paper.

For the π-calculus with the mismatch the normal forms defined in the standard manner are not quite useful in the algebraic investigation. One needs the complete normal forms defined below.

Definition 26. Let V be a finite set of names. A condition ψ is complete on V if for any condition ϕ such that $n(\phi) \subseteq V$ either $\psi\phi \Leftrightarrow \psi$ or $\psi\phi \Leftrightarrow \bot$.

A complete condition on V is a maximal condition on V that can not be extended or strengthened by any condition on V.

Definition 27. 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 for each $i \in I$ the following conditions are met:

- $bn(\lambda_i) \cap V = \emptyset$;
- ψ_i is complete on V;
- and P_i is a complete normal form on $V \cup bn(\lambda_i)$.

The standard property required for normal forms, irrespective of their definitions, is that every finite (or finite control) process is convertible to a normal form process. This is established by the next lemma, whose proof is routine using structural induction.

Lemma 28. *For every process P there is a complete normal form P' such that $AS \vdash P = P'$.*

The next lemma describes a useful equational property about the operation $(\_)^{\tilde{z}}$.

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

Figure 1: *AS*: Basic Rules and Axioms for the Full Pi Calculus

Lemma 29. *Suppose that none of the free names appearing in the match/mismatch operations of P is bound by an input prefix in $C[]$. Then $AS \vdash (z)C[P^z] = (z)C[P]$.*

Proof. The proof is carried out by an induction on P. We take a look at the major cases:

- P is of the form $[x{=}y]P_1$. There are three subcases:
  - $z \notin \{x, y\}$. Then $(z)C[P^z] = (z)C[[x{=}y]P_1^z] \stackrel{I.H.}{=} (z)C[[x{=}y]P_1] = (z)C[P]$ where "I.H." means "induction hypothesis".
  - $x = z = y$. Then $(z)C[P^z] = (z)C[P_1^z] \stackrel{I.H.}{=} (z)C[P_1] = (z)C[P]$.
  - $z \in \{x, y\}$ and $x \neq y$. Then $(z)C[P^z] = (z)C[\mathbf{0}] \stackrel{L8}{=} (z)C[P]$.
- P is of the form $[x{\neq}y]P_1$. The situation is similar, using D8.
- P is of the form $a(x).P_1$. Then

$$\begin{aligned} (z)C[P^z] &= (z)C[a(x).([x{=}z](P_1\{z/x\})^z + [x{\neq}z]P_1^z)] \\ &\stackrel{I.H.}{=} (z)C[a(x).([x{=}z]P_1\{z/x\} + [x{\neq}z]P_1^z)] \\ &= (z)C[a(x).([x{=}z]P_1 + [x{\neq}z]P_1^z)] \\ &\stackrel{S6}{=} (z)C[a(x).P_1] \\ &= (z)C[P] \end{aligned}$$

This completes the proof. □

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 $x \notin fn(P)$
D8	$(x)C[[x{\neq}y]P]$	$=$	$(x)C[P]$	$x,y \notin bn(C[])$ and $x{\neq}y$

Figure 2: Some Laws Derivable from AS

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

Figure 3: Tau Laws

In the rest of this paper, we will let AS_q be the set $AS \cup \{T1, T2, T3, T4, T5\}$. The purpose of the remaining section is to prove that AS_q is a complete system for the quasi open congruence. The congruence is defined in the standard manner ([10, 13]):

Definition 30. P and Q are quasi open congruent, notation $P \simeq_q^l Q$, if $P \approx_q Q$ and the following properties hold:
(i) If $P \xrightarrow{\tau} P'$ then $Q \stackrel{\tau}{\Longrightarrow} Q' \approx_q P'$ for some Q';
(ii) If $Q \xrightarrow{\tau} Q'$ then $P \stackrel{\tau}{\Longrightarrow} P' \approx_q Q'$ for some P'.

7.1 Saturation

Saturation properties say that additional internal actions (tau actions) can be ironed out by equational rewriting. In CCS this property takes the following form:

> If $P \stackrel{\lambda}{\Longrightarrow} P'$ then $P{+}\lambda.P'$ and P are provably equal (in some equational system).

Intuitively the property suggests that one can think of $P \stackrel{\lambda}{\Longrightarrow} P'$ as a single step action $P \xrightarrow{\lambda} P'$ from the viewpoint of an equational theory that is rich enough. When one has established this property for an equational system, one is half way through in proving a completeness result. For the π-calculus without the match operator the property is termed as follows:

> If σ is a substitution induced by ψ and $P\sigma \stackrel{\lambda}{\Longrightarrow} P'$ then $P{+}\psi\lambda.P'$ and P are provably equal (in some equational system).

In the presence of the mismatch operator it should be refined to the following:

> If σ is a substitution induced by ψ, ψ is complete on $n(P)$ and $P\sigma \stackrel{\lambda}{\Longrightarrow} P'$ then $P{+}\psi\lambda.P'$ and P are provably equal (in some equational system).

The completeness requirement can not be left out. Take for instance $A \stackrel{\text{def}}{=} \overline{a}a.(\overline{b}b{+}[x{\neq}y]\tau.\overline{c}c)$. Obviously $A \stackrel{\overline{a}a}{\Longrightarrow} \overline{c}c$. But $A{+}\overline{c}c$ can not be equal to A since $(A{+}\overline{c}c)\{y/x\} \xrightarrow{\overline{c}c} \mathbf{0}$ can not be matched up by any action of A.

The next lemma establishes some basic saturation properties that are independent of any particular observational equality, meaning that they hold of all the observational equalities we are aware of.

Lemma 31 (saturation 1). *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 $Q\sigma \stackrel{\tau}{\Longrightarrow} Q'$ then $AS_q \vdash Q = Q + \psi\tau.Q'$.

(ii) If $Q\sigma \stackrel{\overline{a}x}{\Longrightarrow} Q'$ *then* $AS_q \vdash Q = Q + \psi\overline{a}x.Q'$.
(iii) If $Q\sigma \stackrel{\overline{a}(x)}{\Longrightarrow} Q'$ *then* $AS_q \vdash Q = Q + \psi\overline{a}(x).Q'$.
(iv) If $Q\sigma \Longrightarrow\stackrel{ax}{\longrightarrow} Q'$, *for* $x \notin fn(Q\sigma)$, *then* $AS_q \vdash Q = Q + \psi a(x).Q'$.

Proof. The proofs are routine inductions on the lengths of the action sequences. We sketch a proof of (iii). Suppose $Q\sigma \Longrightarrow\stackrel{\overline{a}(x)}{\longrightarrow} Q''\sigma \stackrel{\tau}{\Longrightarrow} Q'$. It is easy to show that

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

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

$$AS \cup \{T1, T2, T3\} \vdash Q'' = Q'' + [x{\notin}V]\psi\tau.Q'$$

Therefore

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

We are done. □

Notice that the (iv) of Lemma 31 can not be strengthened to

(iv') If $Q\sigma \stackrel{ax}{\Longrightarrow} Q'$, for $x \notin fn(Q\sigma)$, then $AS_q \vdash Q = Q + \psi a(x).Q'$.

This is because Q' might contain x free and therefore in general the substitution σ is not induced by a condition ψ complete on $V \cup \{x\}$. A careful analysis of the input actions leads to the following lemma.

Lemma 32 (saturation 2). *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*

$Q\sigma \Longrightarrow\stackrel{ax}{\longrightarrow} Q'\sigma$ *for* $x \notin fn(Q\sigma)$,
$Q'\sigma\{y_1/x\} \Longrightarrow Q_1$,
$Q'\sigma\{y_2/x\} \Longrightarrow Q_2$,
$\vdots$
$Q'\sigma\{y_k/x\} \Longrightarrow Q_k$,
$Q'\sigma \Longrightarrow Q_{k+1}$

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

Proof. By the assumption of the lemma and Lemma 31

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

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

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

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 $Q'\sigma \overset{\tau}{\Longrightarrow} Q_{k+1}$ 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. □

Intuitively the saturation property for an input action of Q must be based upon the knowledge of what the process evolves into after receiving a name in $fn(Q)$ and what it evolves into after receiving a name not in $fn(Q)$. In other words, the saturation property must take into account all the possible consequences the input action might incur.

7.2 Promotion

Promotion refers to the fact that a pair of observable processes can be lifted to a closely related pair of proof theoretical equal processes. This fact is crucial to the proof of the completeness theorem.

Lemma 33 (promotion). *If $P \approx_{qo} Q$ then $AS_q \vdash \tau.P = \tau.Q$.*

Proof. Suppose $P \approx_{qo} Q$. By Lemma 28, we may assume that both P and Q are complete normal forms on $V \overset{\text{def}}{=} fn(P{+}Q){=}\{y_1,\ldots,y_k\}$. Let them be $\sum_{i\in I}\phi_i\lambda_i.P_i$ and $\sum_{j\in J}\psi_j\lambda_j.Q_j$ respectively. Then $P\sigma \overset{\lambda_i\sigma}{\longrightarrow} P_i\sigma$ for a non input action and $P\sigma \overset{(a_i\sigma)x}{\longrightarrow} P_i\sigma$ for input action with fresh x. Let σ be a substitution induced by ϕ_i.

- λ_i is a restricted output action $\overline{a}(x)$. Then Q' exists such that $Q\sigma \overset{\overline{a\sigma}(x)}{\Longrightarrow} Q'$ and $(P_i\sigma)^x \approx_{qo} (Q')^x$. By induction hypothesis one has $AS_q \vdash \tau.(P_i\sigma)^x = \tau.(Q')^x$. By Lemma 29, one has $\overline{a}(x).P = \overline{a}(x).P^x$. Therefore by Lemma 31 one gets

$$
\begin{aligned}
AS_q \vdash Q{+}\phi_i\overline{a}(x).P_i &= Q{+}\phi_i\overline{a\sigma}(x).P_i\sigma \\
&= Q{+}\phi_i\overline{a\sigma}(x).\tau.(P_i\sigma)^x \\
&= Q{+}\phi_i\overline{a\sigma}(x).\tau.(Q')^x \\
&= Q{+}\phi_i\overline{a\sigma}(x).Q' \\
&= Q
\end{aligned}
$$

- λ_i is an input action $(a_i\sigma)x$. It follows from $P \approx_{qo} Q$ that Q' exists such that $Q\sigma \Longrightarrow\overset{(a_i\sigma)x}{\longrightarrow} Q'\sigma$ and the following conditions hold:
  - For each $l \in \{1,\ldots,k\}$, Q_{i_l} exists such that $Q'\sigma\{y_l/x\} \Longrightarrow Q_{i_l} \approx_{qo} P_i\sigma\{y_l/x\}$.

– $Q_{i_{k+1}}$ exists such that $Q'\sigma \Longrightarrow Q_{i_{k+1}} \approx_{qo} P_i\sigma$.

It follows by induction hypothesis that $\tau.Q_{i_l} = \tau.P_i\sigma\{y_l/x\}$ for $l \in \{1,\ldots,k\}$ and $\tau.Q_{i_{k+1}} = \tau.P_i\sigma$. Now by Lemma 32 one gets

$$\begin{array}{rcl} AS_q \vdash Q & = & Q + \phi_i a_i(x).(\tau.Q' + \sum_{l=1}^{k} \phi_i[x{=}y_l]\tau.Q_{i_l}\sigma\{y_l/x\} + \phi_i[x{\notin}V]\tau.Q_{i_{k+1}}\sigma) \\ & = & Q + \phi_i a_i(x).(\tau.Q' + \sum_{l=1}^{k} \phi_i[x{=}y_l]\tau.P_i\sigma\{y_l/x\} + \phi_i[x{\notin}V]\tau.P_i\sigma) \\ & = & Q + \phi_i a_i(x).(\tau.Q' + \sum_{l=1}^{k} \phi_i[x{=}y_l]\tau.P_i + \phi_i[x{\notin}V]\tau.P_i) \\ & = & Q + \phi_i a_i(x).(\tau.Q' + \phi_i\tau.P_i) \\ & \stackrel{T3}{=} & Q + \phi_i a_i(x).P_i \end{array}$$

where the fourth equality holds because $x{\notin}V \vee (\bigvee_{l=1}^{k} x{=}y_l)$ is a tautology.

The other cases are simpler. The rest of the proof is standard. We refer the reader to [2] for details. □

7.3 Completeness

Using the fact that $P \approx_{qo} Q$ iff $\tau.P \simeq_{qo} \tau.Q$, one can rephrase the promotion lemma as follows:

If $\tau.P \simeq_{qo} \tau.Q$ then $AS_q \vdash \tau.P = \tau.Q$.

This reiteration makes it obvious that promotion is a restricted form of completeness. As it turns out the proof of the promotion lemma can be modified slightly to a proof of the following completeness theorem. The modification follows the following slogan:

> "Wherever we use the induction hypothesis in the proof of the promotion lemma, we use the promotion lemma in the proof of the completeness theorem."

The reader who is not quite familiar with the routine is referred to [2].

Theorem 34 (completeness). *If $P \simeq_{qo} Q$ then $AS_q \vdash P = Q$.*

8 Conclusion

The bisimulation approach was first applied to the pure CCS ([9]) and later to the value-passing CCS ([3, 4]). It is fair to say that the former is generalized, in the framework of the π-calculus, to the open bisimulation whereas the latter is related to the early/late bisimulation equivalence. The value passing calculi can be classified into two groups according to what can be communicated. For the calculi in the first group, the contents of communications may contain *anything but* channel names. The value passing CCS is one such calculus. For those in the second group the situation is the opposite. The π family belong to the second group. For the (first order) π-calculus, the contents of communications can be *nothing but* channel names. Now for the value passing CCS the only way environments may affect processes is through exchanging data. Data may change the control flow of the computation but not the communication topology. On the other hand environments have a much stronger impact in the π-calculus. A consequence of this stronger observation power is that bisimulations must be closed under every bisimulation step. This also points out that the value passing view does not give rise to equivalences that take into account of the dynamic updating power of the environments. So the open bisimulations have the edge. The only problem is that it assumes that an environment has something it really does not have, that is the ability to modify local names. If we rectify this, we arrive at the authentic equivalence for the π-calculus, the quasi open bisimilarity. In this paper we have achieved a better understanding of the equivalence.

References

[1] Y. Fu, On Quasi Open Bisimulation, *Theoretical Computer Science*, **338** (2005) 96-126.

[2] Y. Fu and Z. Yang, Tau Laws for Pi Calculus, *Theoretical Computer Science*, **308** (2003) 55-130.

[3] M. Hennessy, H. Lin, Proof Systems for Message Passing Process Algebra, *CONCUR '93*, Lecture Notes in Computer Science 715 (Springer-Verlag, 1993) 202-216.

[4] M. Hennessy, H. Lin, Symbolic Bisimulations, *Theoretical Computer Science*, **138** (1995) 137-161.

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

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

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

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

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

[10] 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).

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

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

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

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

[15] D. Sangiorgi and D. Walker, The Pi Calculus–A Theory of Mobile Processes, CUP, 2001.