Complete Proof Systems for Observation Congruences in Finite-Control π-Calculus*

H. Lin
Laboratory for Computer Science
Institute of Software, Chinese Academy of Sciences
E-mail: lhm@ox.ios.ac.cn

Abstract

Proof systems for weak bisimulation congruences in the finite-control π-calculus are presented and their completeness proved. This study consists of two major steps: first complete proof systems for *guarded* recursions are obtained; then sound laws sufficient to remove any unguarded recursions are formulated. These results lift Milner's axiomatisation for observation congruence in regular pure-CCS to the π-calculus. The completeness proof relies on the *symbolic bisimulation* technique.

1 Introduction

Axiomatisation for the π-calculus has received extensive studies since the infancy of this calculus. Different bisimulation equivalences in the recursion-free subset of the π-calculus have been successfully axiomatised: late ground bisimulation [9], late/early strong bisimulation congruences [10, 1, 4], open bisimulation [11], late/early weak bisimulation congruences [5]; styles of proof systems have been exploited: equational axiomatisation [10, 11] and *symbolic* inference systems [1, 4, 5]. To deal with recursion, [6] proposed a version of *Unique fixpoint induction*, thus obtained complete proof systems for both late and early strong bisimulation congruence in finite-control π-calculus with *guarded* recursions. The main contributions of the present paper are: (1) Presenting proof systems for weak bisimulation congruences in guarded finite-control π-calculus and proving their completeness; (2) Formulating sound axioms to transform unguarded recursions to guarded ones, thus extending these proof systems to the full language of finite-control π-calculus.

These results can be seen as "lifting" Milner's axiomatisation for observation congruence in regular pure-CCS ([8]) to the π-calculus. For the lifting to work three major problems must be tackled.

1. The proof of the complete axiomatisation in [8] relies on the finite-branching and finite-state properties of regular CCS. However in the π-calculus input actions may result in infinitely many outgoing transitions, and recursively defined processes are parameterised on names and therefore may have infinite states. Such infinity can be avoided by appealing

*Supported by grants from the National Science Foundation of China and the Chinese Academy of Sciences.

to the *symbolic* approach, originally developed for general message-passing processes ([3]) and has been successfully adapted to the π-calculus in [1, 4, 5, 6].

2. The key inference rule to deal with recursion in [7, 8] is *unique fixpoint induction*:

$$\frac{P = Q[P/X]}{P = \mathbf{fix} X Q} \qquad X \text{ guarded in } Q$$

Here $[P/X]$ is a first order substitution: X is a process variable and P a process term. But in the π-calculus process variables may take name parameters, hence substitutions on process variables must cater for parameter-passing. This paper inherits the solution proposed in [6] of using *second order substitution*. The syntactic form of the unique fixpoint induction will remain the same, but P, Q will be *process abstractions*, namely functions from names to process terms, and X a process abstraction variable. In $Q[P/X]$, passing to P the actual parameters of the occurrences of X in Q is realized by β-conversion.

3. The last problem concerns unguarded recursions. Let us look at a simple example:

$$(\mathbf{fix} X(x,y,z)(\overline{x}y.\mathbf{0} + X(y,x,z) + X(x,z,y)))(a,b,c)$$

This process consists of a recursion abstraction with formal parameters (x,y,z), applied to the actual parameter (a,b,c). It contains two (immediate) unguarded recursions, $X(y,x,z)$ and $X(x,z,y)$. We can not use the law $\mathbf{fix} X(T+X) = \mathbf{fix} XT$, which is sound for pure-CCS [7, 8], to reduce the term to $\mathbf{fix} X(x,y,z)(\overline{x}y.\mathbf{0})(a,b,c)$, because, for instance, $X(y,x,z)$ may contribute to the behaviour of the original term by causing the transition $\xrightarrow{\overline{b}a}$ (resulted from unfolding the recursion once), and the combined effects of $X(y,x,z)$ and $X(x,z,y)$ may produce further transitions $\xrightarrow{\overline{c}a} \mathbf{0}$, $\xrightarrow{\overline{b}c} \mathbf{0}$ *etc.* General laws for eliminating immediate as well deeply nested unguarded recursions are formulated in Section 5, which also take care for conditions and restrictions.

Due to the presence of additional constructs in the π-calculus (more sophisticated actions, the conditional construct, name parameters, ...), it should not be surprising that proof of the completeness theorem is more complicated than in pure-CCS. However much of the complexity has been brought under control by the systematic employment of *maximally consistent extensions* of conditions.

The rest of the paper is organised as follows: The calculus and the symbolic semantics are introduced in the next section. Section 3 presents the inference system and summaries its properties, while the completeness proof is the subject of the subsequent section. The axioms for removing unguarded recursions are studied in Section 5, and the paper is concluded with Section 6.

2 The Language and Weak Bisimulations

We presuppose an infinite set $\mathcal{N}$ of *names*, ranging over by a, b, x, $y, \ldots$, and a countably infinite set of *process abstraction variables* (or just *process variables* for short), *Var*, ranged over by $X, Y, Z, W, \ldots$. Each process abstraction variable is associated with an *arity* indicating the number of names it should take to form a process term. The language of the

π-calculus we consider in this paper can then be given by the following BNF grammar

$$\begin{array}{lcl} T & ::= & \mathbf{0} \mid \alpha.T \mid T+T \mid \phi T \mid \nu x T \mid F\tilde{x} \\ F & ::= & X \mid (\tilde{x})T \mid \mathbf{fix} XF \\ \alpha & ::= & \tau \mid a(x) \mid \overline{a}x \\ \phi & ::= & [x=y] \mid \phi \wedge \phi \mid \neg\phi \end{array}$$

This language is the π-calculus analogue to *regular CCS*. By adding parallel composition while keeping it outside the scope of the **fix** operator we obtain the fragment known as "finite-control π-calculus" ([2]). We will concentrate on the regular subset in the main body of this paper, and will indicate how our results can be extended to the finite-control fragment at the end of Section 4.

Most of the operators are taken from [9]. The only construct worth mentioning is *abstraction* which is either a process variable, a process abstraction $(\tilde{x})T$ (where $\tilde{x}$ is a vector of distinct names), or a recursion abstraction $\mathbf{fix}XF$. Three constructs, input prefixing $a(x).T$, restriction νxT and process abstraction $(\tilde{x})T$, introduce bound names, and we use the customary notations $bn(T)$ and $fn(T)$ for bound and free names in T. For any expression e, $n(e)$ denotes the set of all names appearing in e. We shall only use name-closed abstractions, so in $(\tilde{x})T$ it is always required that $fn(T) \subseteq \{\tilde{x}\}$. A recursion abstraction $\mathbf{fix}XF$ binds X in F, and we use $FV(T)$ to denote the set of free process variables of T. We shall assume that all the expressions we write are syntactically well-formed. A recursion $\mathbf{fix}XF$ is *guarded* if each occurrence of X in F is within the scope of some prefix operator $\alpha.-$ with $\alpha \not\equiv \tau$.

Bound names and bound variables induce the notion of α-equivalence as usual. In the sequel we will not distinguish between α-equivalent terms and will use $\equiv$ for both syntactical equality and α-equivalence.

We use the customary notation for name substitution, ranged over by σ. $[F_i/X_i|1 \leq i \leq m]$, with X_i:s distinct, is a process substitution that sends X_i to F_i for $1 \leq i \leq m$. We will often simply write $T[F_i/X_i|i]$ when the range of i is clear from the context. In name and process substitutions bound names and bound variables will be renamed when necessary to avoid capture.

Conditions, ranged over by ϕ, ψ, are boolean expressions over *matches* of the form $[x = y]$. Sometimes we will abbreviate $\phi \wedge \psi$ as $\phi\psi$. *True* stands for $x = x$ and *false* for $\neg true$. We write $\sigma \models \phi$ if $\phi\sigma = true$, and $\phi \Rightarrow \psi$ if $\sigma \models \phi$ implies $\sigma \models \psi$ for any substitution σ. It is not difficult to see that the relation $\phi \Rightarrow \psi$ is decidable.

A condition ϕ is consistent if there are no $x,\ y \in \mathcal{N}$ such that $\phi \Rightarrow x = y$ and $\phi \Rightarrow x \neq y$. ϕ is *maximally consistent* on $V \subset \mathcal{N}$ if ϕ is consistent and for any $x,\ y \in V$ either $\phi \Rightarrow x = y$ or $\phi \Rightarrow x \neq y$.

Let the restriction of ϕ on V be

$$\phi|_V = \bigwedge\{\, x = y \mid x, y \in V, \phi \Rightarrow x = y \,\} \wedge \bigwedge\{\, x \neq y \mid x, y \in V, \phi \Rightarrow x \neq y \,\}$$

Lemma 2.1 *If ϕ is maximally consistent on V' and $V \subseteq V'$, then $\phi|_V$ is maximally consistent on V.*

$$\text{Pre}\ \frac{}{\alpha.P \xrightarrow{true,\alpha} P} \qquad \text{Cond}\ \frac{P \xrightarrow{\phi,\alpha} P'}{\psi P \xrightarrow{\phi\wedge\psi,\alpha} P'} \qquad \text{Rec}\ \frac{F\tilde{v}[\mathbf{fix}XF/X] \xrightarrow{\phi,\alpha} P'}{(\mathbf{fix}XF)\tilde{v} \xrightarrow{\phi,\alpha} P'}$$

$$\text{Sum}\ \frac{P \xrightarrow{\phi,\alpha} P'}{P+Q \xrightarrow{\phi,\alpha} P'} \qquad \text{Res}\ \frac{P \xrightarrow{\phi,\alpha} P'}{\nu xP \xrightarrow{\nu_x\phi,\alpha} \nu xP'} x \notin n(\alpha) \qquad \text{Open}\ \frac{P \xrightarrow{\phi,\overline{a}x} P'}{\nu xP \xrightarrow{\nu_x\phi,\overline{a}(x)} P'} x \neq a$$

Figure 1: π-Calculus Symbolic Transitional Semantics

ψ is a *maximally consistent extension* of ϕ on V, written $\psi \in MCE_V(\phi)$, if $n(\psi) \subseteq V$, $\psi \Rightarrow \phi|_V$ and ψ is maximally consistent on V. Up-to logical equivalence, the set of maximally consistent extensions of a given condition on a finite set of names V is finite. $MCE_V(true)$ will be abbreviated as MC_V.

Lemma 2.2 $\bigvee MCE_V(\phi) = \phi|_V$.

Due to space limitation we refer to the standard references of the π-calculus for the definitions of late ground weak bisimulation ($\dot{\approx}_l$) and observation congruence ($\simeq_l$), and shall only give the *symbolic* versions of these equivalences. The symbolic transitional semantics of the π-calculus is reported in Figure 1, where ν_x is an operation on conditions defined thus ([1]):

$$\begin{array}{lll} \nu_x true = true & \nu_x[x=x] = true & \nu_x[x=y] = false \\ \nu_x[y=z] = [y=z] & \nu_x\neg\phi = \neg\nu_x\phi & \nu_x(\phi\wedge\psi) = \nu_x\phi \wedge \nu_x\psi \end{array}$$

Definition 2.3 The symbolic late double arrows are defined as the least relations satisfying the following rules: (1) $T \overset{true,\varepsilon}{\Longrightarrow}_L T$; (2) $T \xrightarrow{\phi,\alpha}_L U$ implies $T \overset{\phi,\alpha}{\Longrightarrow}_L U$; (3) $T \xrightarrow{\phi,\tau}_L \overset{\psi,\alpha}{\Longrightarrow}_L U$ implies $T \overset{\phi\psi,\alpha}{\Longrightarrow}_L U$; (4) If α does not have the form $a(x)$ then $T \overset{\phi,\alpha}{\Longrightarrow}_L \xrightarrow{\psi,\tau}_L U$ implies $T \overset{\phi\psi,\alpha}{\Longrightarrow}_L U$.
We shall write $\overset{\phi,\hat{\tau}}{\Longrightarrow}_L$ for $\overset{\phi,\varepsilon}{\Longrightarrow}_L$, and $\overset{\phi,\hat{\alpha}}{\Longrightarrow}_L$ for $\overset{\phi,\alpha}{\Longrightarrow}_L$ when $\alpha \not\equiv \tau$. □

In the following definition of symbolic bisimulation we use $\alpha =^\phi \beta$ to mean

if $\alpha \equiv \tau$ then $\beta \equiv \tau$	if $\alpha \equiv \overline{a}x$ then $\beta \equiv \overline{b}y$ and $\phi \Rightarrow a = b \wedge x = y$
if $\alpha \equiv \overline{a}(x)$ then $\beta \equiv \overline{b}(x)$ and $\phi \Rightarrow a = b$	if $\alpha \equiv a(x)$ then $\beta \equiv b(x)$ and $\phi \Rightarrow a = b$

Definition 2.4 A condition indexed family of symmetric relations $\mathcal{S} = \{S^\phi\}$ is a symbolic late weak bisimulation if $(T,U) \in S^\phi$ implies for each $\psi \in MCE_{fn(T,U)}(\phi)$

whenever $T \xrightarrow{\phi',\alpha} T'$ with $bn(\alpha) \cap fn(T,U) = \emptyset$ and $\psi \Rightarrow \phi'$, then there is a $U \overset{\psi',\hat{\beta}}{\Longrightarrow}_L U'$ such that $\psi \Rightarrow \psi'$, $\alpha =^\psi \beta$, and

- If $\alpha \equiv a(x)$ then for each $\psi'' \in MCE_{fn(T,U)\cup\{x\}}(\psi)$ there is a $U' \overset{\phi''}{\Longrightarrow}_L U''$ such that $\psi'' \Rightarrow \phi''$ and $(T',U'') \in S^{\psi''}$.
- If $\alpha \equiv \overline{a}(x)$ then $(T',U') \in S^{\psi\cup\{\, x\neq y | y\in fn(\alpha.T',\beta.U')\,\}}$

$$\text{CHOICE}\ \frac{\phi \rhd T_i = U_i}{\phi \rhd T_1 + T_2 = U_1 + U_2} \qquad \text{COND}\ \frac{\phi \wedge \psi \rhd T = U \quad \phi \wedge \neg\psi \rhd \mathbf{0} = U}{\phi \rhd \psi T = U}$$

$$\text{TAU}\ \frac{\phi \rhd T = U}{\phi \rhd \tau.T = \tau.U} \qquad \text{RES}\ \frac{\phi \wedge \bigwedge_{y \in fn(\nu xT, \nu xU)} x \neq y \rhd T = U}{\phi \rhd \nu xT = \nu xU} x \notin n(\phi)$$

$$\text{INPUT}\ \frac{\phi \rhd T = U}{\phi \rhd a(x).T = b(x).U} \begin{array}{l} \phi \Rightarrow a = b \\ x \notin n(\phi) \end{array} \qquad \text{CUT}\ \frac{\phi_1 \rhd T = U \quad \phi_2 \rhd T = U}{\phi \rhd T = U} \phi \Rightarrow \phi_1 \vee \phi_2$$

$$\text{OUTPUT}\ \frac{\phi \rhd T = U}{\phi \rhd \overline{a}x.T = \overline{b}y.U} \begin{array}{l} \phi \Rightarrow a = b \\ \phi \Rightarrow x = y \end{array} \qquad \text{ABSURD}\ \frac{}{\mathit{false} \rhd T = U}$$

Figure 2: RP: The Inference Rules for Process Terms

- Otherwise $(T', U') \in S^{\psi}$

Let $\approx_L$ be the largest symbolic late weak bisimulation.
For abstractions, $F \approx_L^{true} G$ iff $F\tilde{x} \approx_L^{true} G\tilde{x}$ for any $\tilde{x}$. For process terms T, U with free process variables in $\tilde{X}$, $T \approx_L^{\phi} U$ iff $T[\tilde{F}/\tilde{X}] \approx_L^{\phi} U[\tilde{F}/\tilde{X}]$ for any vector $\tilde{F}$ of process-closed abstractions.

T, U are *symbolic late observation congruent* with respect to ϕ, written $T \simeq_L^{\phi} U$, if $T \approx_L^{\phi} U$, any initial τ move from T (U) is matched by at least one τ move from U (T), and the residuals are weak bisimilar. □

Notation We will write $(T, U)^{\psi} \in \mathcal{S}$ to mean that there is $S^{\phi} \in \mathcal{S}$ such that $(T, U) \in S^{\phi}$ and $\psi \Rightarrow \phi$.

The following theorem asserts that $\approx_L$ and $\simeq_L$ capture $\dot{\approx}$ and $\dot{\simeq}_l$, respectively:

Theorem 2.5 *1. $T \approx_L^{\phi} U$ iff $T\sigma \dot{\approx}_l U\sigma$ for any $\sigma \models \phi$.*

2. $T \simeq_L^{\phi} U$ iff $T\sigma \dot{\simeq}_l U\sigma$ for every $\sigma \models \phi$.

3 The Proof Systems

The proof system for $\simeq_L$ is presented as a set of inference rules (RP in Figure 2 for process terms, and RA in Figure 3 for abstractions), together with the standard equational axioms in Figure 4. The judgements are *conditional equations* of the form

$$\phi \rhd T = U$$

meaning "T and U are equal when their free names satisfying ϕ". $\vdash \mathit{true} \rhd T = U$ is abbreviated as $\vdash T = U$.

Several subsets of this inference system are of interest. Let $\vdash^{fs} = RP \cup \{S1 - S4, R1 - R6\}$, $\vdash^{fw} = \vdash^{fs} \cup \{T1 - T3\}$, $\vdash^{grs} = \vdash^{fs} \cup RA$, and $\vdash^{grw} = \vdash^{grs} \cup \{T1 - T3\}$. $\vdash^{fs}$ ($\vdash^{fw}$) is sound and complete for late strong (weak) bisimulation congruence in recursion-free π-calculus ([4, 5]), and $\vdash^{grs}$ is sound and complete for late strong

$$\text{ABS}\ \frac{true \rhd T = U}{true \rhd (\tilde{x})T = (\tilde{x})U} \qquad \text{APP}\ \frac{true \rhd F = G}{true \rhd F\tilde{w} = G\tilde{w}}$$

$$\text{CONGR-}\mathbf{fix}\ \frac{true \rhd F = G}{true \rhd \mathbf{fix}XF = \mathbf{fix}XG} \qquad \text{REC}\ \frac{}{true \rhd \mathbf{fix}XF = F[\mathbf{fix}XF/X]}$$

$$\text{UFI}\ \frac{true \rhd G = F[G/X]}{true \rhd G = \mathbf{fix}XF}$$

Figure 3: RA: The Inference Rules for Abstractions

bisimulation in guarded finite-control π-calculus ([6])[1]. In the next section we shall show that $\vdash^{grw}$ is sound and complete for $\simeq_L$ in guarded finite-control π-calculus. In Section 5 we will extend $\vdash^{grs}$ and $\vdash^{grw}$ to the full finite-control π-calculus. To simplify the notation, in this and next sections we will write $\vdash$ for $\vdash^{grw}$.

Some usefull properties of this proof system are summerised in the following propositions:

Proposition 3.1 *1. $\vdash \phi \wedge \psi \rhd T = U$ iff $\vdash \phi \rhd \psi T = \psi U$.*

2. If $\phi \Rightarrow \psi$ and $\vdash \phi \rhd T = U$ then $\vdash \phi \rhd \psi T = U$.

3. If $\phi \wedge \psi \Rightarrow false$ then $\vdash \phi \rhd \psi T = \mathbf{0}$.

4. $\vdash \psi(T + U) = \psi T + \psi U$.

5. $\vdash \nu x(\psi T) = (\nu_x \psi)\nu x T$.

Proposition 3.2 *Suppose $fn(T, U) \subseteq V$. If $\vdash \psi \rhd T = U$ for each $\psi \in MCE_V(\phi)$ then $\vdash \phi \rhd T = U$.*

Theorem 3.3 *(Soundness) If $\vdash \phi \rhd T = U$ then $T \simeq_L^{\phi} U$.*

4 Completeness of $\vdash$ over Guarded Terms

The structure of the completeness proof follows that of [8], but there are extra complications due to the presences of name parameters, the conditional construct, and bound actions. The first step is to demonstrate that any guarded process term can be proved to satisfy a standard set of equations. The next step is to show if two process terms are symbolically bisimilar then an equation set can be constructed which is provably satisfied by both. Finally we show that two process terms satisfying the same set of equations can be proved equal.

[1]By adding an "early" axiom (see Section 6), all these proof systems are extended to the corresponding early equivalences.

S1 $T + \mathbf{0} = T$	S2 $T + U = U + T$	S3 $(T + U) + V = T + (U + V)$
S4 $T + T = T$		
R1 $\nu x\mathbf{0} = \mathbf{0}$	R2 $\nu x\alpha.T = \alpha.\nu xT \;\; x \notin n(\alpha)$	R3 $\nu x\alpha.T = \mathbf{0} \;\; x$ is the port of α
R4 $\nu x\nu yT = \nu y\nu xT$	R5 $\nu x(T + U) = \nu xT + \nu xU$	R6 $\nu xT(\tilde{w}) = T(\tilde{w}) \quad x \notin \{\tilde{w}\}$
T1 $\alpha.\tau.T = \alpha.T$	T2 $T + \tau.T = \tau.T$	T3 $\alpha.(T + \tau.U) + \alpha.U = \alpha.(T + \tau.U)$

Figure 4: The Equational Axioms

Let $\tilde{X} = \{X_1, X_2, ..., X_m\}$ and $\tilde{W} = \{W_1, W_2, ..., W_n\}$ be disjoint sets of process variables. Then

$$E : X_i(\tilde{x}_i) = H_i \qquad 1 \le i \le m,$$

where $fn(H_i) \subseteq \{\tilde{x}_i\}$ and $FV(H_i) \subseteq \tilde{X} \cup \tilde{W}$, is a set of equations with formal process variables in $\tilde{X}$ and free process variables in $\tilde{W}$. E is *standard* if each H_i has the form

$$\sum_{\psi_{ik} \in MC_{\{\tilde{x}\}}} \psi_{ik}\Big(\sum_{p \in P_{ik}} \alpha_{ikp}.X_{f(i,k,p)}(\tilde{x}_{ikp}) + \sum_{p' \in P'_{ik}} \nu\tilde{w}'_{ikp'} W_{f'(i,k,p')}(\tilde{w}_{ikp'})\Big)$$

with $bn(\alpha_{ikp}) \cap \{\tilde{x}_i\} = \emptyset$ and $\{\tilde{w}'_{ikp'}\} \subseteq \{\tilde{w}_{ikp'}\}$. Given a standard equation set E we define the formal relations $\longrightarrow_F$ and $\succ_F$ by letting $X_i(\tilde{x}_i) \stackrel{\psi_{ik},\alpha_{ikp}}{\longrightarrow}_F X_{f(i,k,p)}(\tilde{x}_{ikp})$ and $X_i(\tilde{x}_i) \succ_F^{\psi_{ik}} \nu\tilde{w}'_{ikp'} W_{f'(i,k,p')}(\tilde{w}_{ikp'})$. Also write $\stackrel{\phi_1\phi_2...\phi_n}{\Longrightarrow}_F$ for $\stackrel{\phi_1,\tau}{\longrightarrow}_F \stackrel{\phi_2,\tau}{\longrightarrow}_F \ldots \stackrel{\phi_n,\tau}{\longrightarrow}_F$. E is *guarded* if there is no circle $X_i(\tilde{x}_i) \stackrel{\phi}{\Longrightarrow}_F X_i(\tilde{x}_i)$. E is τ*-saturated* if for all $1 \le i \le m$,

1. If α_{ikp} is a bound action, $X_i(\tilde{x}_i) \stackrel{\phi}{\Longrightarrow}_F \stackrel{\psi_{ik},\alpha_{ikp}}{\longrightarrow}_F X'(\tilde{x}')$ implies $X_i(\tilde{x}_i) \stackrel{\phi\psi_{ik},\alpha_{ikp}}{\longrightarrow}_F X'(\tilde{x}')$.
2. For other actions $X_i(\tilde{x}_i) \stackrel{\phi}{\Longrightarrow}_F \stackrel{\psi_{ik},\alpha_{ikp}}{\longrightarrow}_F \stackrel{\phi'}{\Longrightarrow}_F X'(\tilde{x}')$ implies $X_i(\tilde{x}_i) \stackrel{\phi\psi_{ik}\phi',\alpha_{ikp}}{\longrightarrow}_F X'(\tilde{x}')$.
3. $X_i(\tilde{x}_i) \stackrel{\phi}{\Longrightarrow}_F \succ_F^{\psi} \nu\tilde{w}'W(\tilde{w})$ implies $X_i(\tilde{x}_i) \succ_F^{\phi\psi} \nu\tilde{w}'W(\tilde{w})$.

A process term T provably $\tilde{\phi}$-satisfies an equation set E if there exist T_i with $fn(T_i) \subseteq \{\tilde{x}_i\}$, $1 \le i \le m$, and $T_1 \equiv T$ s.t.

$$\vdash \phi_i \rhd T_i = H_i[(\tilde{x}_k)\phi_k T_k / X_k | 1 \le k \le m] \qquad 1 \le i \le m$$

We will simply say "T provably satisfies E" when $\phi_i \equiv true$ for all i.

Lemma 4.1 *Any guarded process term T with free process variables in $\tilde{W}$ provably satisfies a standard, guarded and τ-saturated equation set E with free process variables in $\tilde{W}$.*

Proposition 4.2 *Suppose T, U are guarded process terms with free process variables in $\tilde{W}$. If $T \simeq_L^{\phi} U$ then there exists $\tilde{\phi}$ with $\phi_1 \equiv \phi$ and a set of guarded equations E with free process variables in $\tilde{W}$ such that both T and U provably $\tilde{\phi}$-satisfy E.*

Proof (Sketch) For simplicity we only consider the case $\{\tilde{W}\} = \emptyset$ here.

Let $E_1,\ E_2$ be the standard, guarded and τ-saturated equation sets provably satisfied by $T,\ U$, respectively:

$$\begin{array}{llll} E_1: & X_i(\tilde{x}_i) = \displaystyle\sum_{\phi_{ik}\in MC_{\{\tilde{x}_i\}}} \phi_{ik} \sum_{p\in P_{ik}} \alpha_{ikp}.X_{f(i,k,p)}(\tilde{x}_{ikp}) & & 1\le i\le m \\ E_2: & Y_j(\tilde{y}_j) = \displaystyle\sum_{\psi_{jl}\in MC_{\{\tilde{y}_j\}}} \psi_{jl} \sum_{q\in Q_{jl}} \beta_{jlq}.Y_{g(j,l,q)}(\tilde{y}_{jlq}) & & 1\le j\le n \end{array}$$

Without losing generality we may assume that each $\tilde{x}_i$ is a vector over $\{\tilde{x}\}$, each $\tilde{y}_j$ a vector over $\{\tilde{y}\}$, $\{\tilde{x}\}\cap\{\tilde{y}\}=\emptyset$, and in $E_1,\ E_2$ all input and bound output prefixes use the same name $v\notin\{\tilde{z}\}$ where $\tilde{z}=\tilde{x}\tilde{y}$.

Thus there exist $T_i,\ 1\le i\le m$, with $fn(T_i)\subseteq\{\tilde{x}_i\},\ T_1\equiv T$, and $U_j,\ 1\le j\le n$, with $fn(U_j)\subseteq\{\tilde{y}_j\},\ U_1\equiv U$, such that

$$\vdash T_i \ = \sum_{\phi_{ik}\in MC_{\{\tilde{x}_i\}}} \phi_{ik} \sum_{p\in P_{ik}} \alpha_{ikp}.T'_{f(i,k,p)} \tag{1}$$

$$\vdash U_j \ = \sum_{\psi_{jl}\in MC_{\{\tilde{y}_j\}}} \psi_{jl} \sum_{q\in Q_{jl}} \beta_{jlq}.U'_{g(j,l,q)} \tag{2}$$

where $T'_{f(i,k,p)}\equiv T_{f(i,k,p)}[\tilde{x}_{ikp}/\tilde{x}_{f(i,k,p)}]$, $U'_{g(j,l,q)}\equiv U_{g(j,l,q)}[\tilde{y}_{jlq}/\tilde{y}_{g(j,l,q)}]$.

Since $T\simeq_L^{\phi} U$, there exists a symbolic late weak bisimulation $\mathcal{S}$ such that $(T,\ U)^{\phi}\in\mathcal{S}$, and $\mathcal{S}$ has the properties stated in Definition 2.4, namely the first τ moves from T (U) are matched by proper τ moves from U (T). $\mathcal{S}$ will be used to guide the construction of the new equation set.

Our goal is to construct an equation set E, out of E_1 and E_2, satisfied by both T and U. Two factors complicate this task:

1. In general E will only be satisfied by T and U *conditionally*, over some vector of conditions $\tilde{\phi}$ with $\phi_1=\phi$.
2. As T and U are observation congruent, we need to take special care of the first τ moves.

Our solution to the first complication is to find *the weakest* such $\tilde{\phi}$: For each pair of $i,\ j$, let

$$\Phi_{ij} = \{\,\psi\in MC_{\{\tilde{z}\}}\mid (T_i,U_j)^{\psi}\in\mathcal{S}\,\}$$

and $\phi_{ij}=\bigvee\Phi_{ij}$. By construction ϕ_{ij} satisfies: $(T_i,U_j)^{\phi_{ij}}\in\mathcal{S}$ and for any ϕ' s.t. $(T_i,U_j)^{\phi'}\in\mathcal{S}$ it holds $\phi'\Rightarrow\phi_{ij}$.

By Lemma 2.1, for each $\psi\in\Phi_{ij}$, $\psi|_{\{\tilde{x}\}}\in MC_{\{\tilde{x}_i\}}$ and $\psi|_{\{\tilde{y}\}}\in MC_{\{\tilde{y}_j\}}$. Therefore there is a unique k and a unique l s.t. $\psi|_{\{\tilde{x}\}}=\phi_{ik}$ and $\psi|_{\{\tilde{y}\}}=\psi_{jl}$. To simplify the notation we will leave such choice of k and l implicit in the sequel.

In constructing E we need to treat four kinds of actions differently. Below we shall only consider τ and bound output, i.e. we assume in E_1 and E_2 α_{ikp} and β_{jlq} are either τ or bound output. The case for free output is straightforward, while the case for input is slightly more complicated and requires some auxiliary notation, but the basic idea is the same.

For each $\psi\in\Phi_{ij}$, let $\mathcal{C}_{\psi}$ be the equivalence classes on $\{\tilde{z}\}$ determined by ψ, namely $x,y\in\{\tilde{z}\}$ are in the same equivalence class if and only if $\psi\Rightarrow x=y$. Let $[e]$ range over $\mathcal{C}_{\psi}$. Define

$$\begin{array}{ll} I_{\psi}^{\tau} & = \{\,(p,q)\mid \alpha_{ikp}\equiv\tau,\ \beta_{jlq}\equiv\tau,\ (T'_{f(i,k,p)},U'_{g(j,l,q)})^{\psi}\in\mathcal{S}\,\} \\ I_{\psi}^{1} & = \{\,q\mid \beta_{jlq}\equiv\tau,\ (T_i,U'_{g(j,l,q)})^{\psi}\in\mathcal{S}\,\} \\ I_{\psi}^{2} & = \{\,p\mid \alpha_{ikp}\equiv\tau,\ (T'_{f(i,k,p)},U_j)^{\psi}\in\mathcal{S}\,\} \\ I_{\psi}^{BO_{[e]}} & = \{\,(p,q)\mid \alpha_{ikp}=^{\psi}\beta_{jlq}=^{\psi}\overline{e}(v),\ (T'_{f(i,k,p)},U'_{g(j,l,q)})^{\psi\cup\{\,z'\neq v\mid z'\in\{\tilde{z}\}\,\}}\in\mathcal{S}\,\} \end{array}$$

The three cases for τ reflect the three different ways to matching a τ move: it can be matched by a proper τ move (I_ψ^τ), or by no move (I_ψ^1 and I_ψ^2) when it is not a first τ move. This is our solution to the second complication mentioned above. Note that $I_\psi^1 = I_\psi^2 = \emptyset$ for $\psi \in \Phi_{11}$.

Let $\{ Z_{ij} \mid 1 \le i \le m,\ 1 \le j \le n \}$ be a set of new process variables and $\tilde{z}_{ij} = \tilde{x}_i \tilde{y}_j$. Consider the equation set

$$E: \ Z_{ij}(\tilde{z}_{ij}) = \sum_{\psi \in \Phi_{ij}} \psi(A_\psi^\tau + \sum_{[e]} A_\psi^{BO_{[e]}})$$

where

$$\begin{aligned} A_\psi^\tau &\equiv \sum_{(p,q)\in I_\psi^\tau} \tau.Z_{f(i,k,p)g(j,l,q)}(\tilde{x}_{ikp}\tilde{y}_{jlq}) + \sum_{q \in I_\psi^1} \tau.Z_{ig(j,l,q)}(\tilde{x}_i \tilde{y}_{jlq}) + \sum_{p \in I_\psi^2} \tau.Z_{f(i,k,p)j}(\tilde{x}_{ikp}\tilde{y}_j) \\ A_\psi^{BO_{[e]}} &\equiv \sum_{(p,q)\in I_\psi^{BO_{[e]}}} \alpha_{ikp}.Z_{f(i,k,p)g(j,l,q)}(\tilde{x}_{ikp}\tilde{y}_{jlq}) \end{aligned}$$

Informally, for each $\psi \in \Phi_{ij}$, the right-hand side of the ij^{th} equation of E consists of the summands in the i^{th} equation of E_1 and the summands in the j^{th} equation of E_2 which are bisimilar over ψ. Note that E is guarded because both $E1$ and $E2$ are. Set

$$V_{ij} \equiv T_i + \sum_{\psi \in \Phi_{ij}} \psi \sum_{q \in I_\psi^1} \tau.T_i$$

Then $V_{11} \equiv T_1 \equiv T$ because $I_\psi^1 = \emptyset$ for $\psi \in \Phi_{11}$. Also, for any $\psi \in \Phi_{ij}$

$$\vdash \psi \rhd V_{ij} = \begin{cases} T_i & \text{if } I_\psi^1 = \emptyset \\ \tau.T_i & \text{otherwise} \end{cases} \tag{3}$$

We claim that E is provably $\{ \phi_{ij} \mid i,j \}$-satisfied by T when each Z_{ij} is instantiated with $(\tilde{z}_{ij})V_{ij}$ over ϕ_{ij}, *i.e.* for each $i,\ j$

$$\vdash \phi_{ij} \rhd V_{ij} = (\sum_{\psi \in \Phi_{ij}} \psi(A_\psi^\tau + \sum_{[e]} A_\psi^{BO_{[e]}}))\theta \tag{4}$$

where $[e]$ ranges over $\mathcal{C}_\psi$ and $\theta \equiv [(\tilde{z}_{ij})^{\phi_{ij}} V_{ij}/Z_{ij} | i,j]$. By Propositions 3.2 and 3.1, and the fact that the elements of Φ_{ij} are mutually disjoint, (4) can be reduced to: for each $\psi \in \Phi_{ij}$

$$\vdash \psi \rhd V_{ij} = A_\psi^\tau \theta + \sum_{[e]} A_\psi^{BO_{[e]}}\theta \tag{5}$$

By examining each summand of the right-hand side, one can show

$$\begin{aligned} \vdash \psi \rhd A_\psi^\tau \theta &= \sum_{(p,q)\in I_\psi^\tau} \tau.T'_{f(i,k,p)} + \sum_{q \in I_\psi^1} \tau.T_i + \sum_{p \in I_\psi^2} \tau.T'_{f(i,k,p)} && (6) \\ \vdash \psi \rhd A_\psi^{BO_{[e]}}\theta &= \sum_{(p,q)\in I_\psi^{BO_{[e]}}} \alpha_{ikp}.T'_{f(i,k,p)} && (7) \end{aligned}$$

With (6) and (7) we go back to equation (5). If $I_\psi^1 = \emptyset$, then, by (3), its left-hand side $V_{ij} = T_i$, and the right-hand side contains exactly the terms $\alpha_{ikp}.T'_{f(i,k,p)}$ and/or $\alpha_{ikp}.\tau.T'_{f(i,k,p)}$ (with possible repetitions), hence (5) follows from (1); Otherwise, the left-hand side $V_{ij} = \tau.T_i$, and the right-hand side additionally contains the terms $\tau.T_i$ and/or $\tau.\tau.T_i$ (rising from the second component of the right-hand side of (6)), therefore (5) also follows from (1) by axiom T2.

Symmetrically we can prove E is provably $\{ \phi_{ij} \mid i,j \}$-satisfied by U. $\square$

Proposition 4.3 *Let* $E : X_i(\tilde{x}_i) = H_i,\ 1 \le i \le m$, *be a guarded equation set with free process variables in* $\tilde{W}$. *If* $T,\ U$ *both provably* $\tilde{\phi}$*-satisfy* E *then* $\vdash \phi_1 \rhd T = U$.

Putting Lemma 4.1 and Propositions 4.2, 4.3 together we obtain

Theorem 4.4 *For guarded process terms* T *and* U, $T \simeq_L^{\phi} U$ *implies* $\vdash \phi \rhd T = U$.

To extend this result to the finite-control π-calculus where parallel composition is allowed to combine regular processes, what needed is to add a version of expansion law ([10, 5]) to the proof system. Then Lemma 4.1 still holds, and the proofs of the other propositions are not affected.

5 Removing Unguarded Recursions

In [8] Milner formulated three elegant laws which are sufficient to remove all unguarded recursions in regular CCS:

$$
\begin{array}{ll}
\mathrm{R3} & \mathbf{fix}X(X+T) = \mathbf{fix}XT \\
\mathrm{R4} & \mathbf{fix}X(\tau.X+T) = \mathbf{fix}X\tau.T \\
\mathrm{R5} & \mathbf{fix}X(\tau.(X+T)+U) = \mathbf{fix}X(\tau.X+T+U)
\end{array}
$$

As has been illustrated by the example in Introduction, these laws can not be naively adopted to the π-calculus. To motivate the axioms to be formulated in the sequel, let us examine this example in more detail. The transition $(\mathbf{fix}X(x,y,z)(\overline{x}y.\mathbf{0} + X(y,x,z) + X(x,z,y)))(a,b,c) \xrightarrow{\overline{a}b} \mathbf{0}$ originates from the subterm $\overline{x}y.\mathbf{0}$ in the recursion body. On the other hand, the unguarded occurrences $X(y,x,z)$ and $X(x,z,y)$ can not contribute to the behaviours of the top-level term by generating transitions from themselves, but they can do so by permuting the recursion parameters appearing in the "real" action derived from $\overline{x}y.\mathbf{0}$. Thus $X(y,x,z)$ is responsible for the transition $\xrightarrow{\overline{b}a} \mathbf{0}$, and the transition $\xrightarrow{\overline{a}c} \mathbf{0}$ is also possible because of $X(x,z,y)$. Furthermore, the combined effects of $X(y,x,z)$ and $X(x,z,y)$ produce more transitions: $\xrightarrow{\overline{c}b} \mathbf{0}$ (first $X(y,x,z)$, followed by $X(x,z,y)$, then $X(y,x,z)$ again), ... As each of $X(y,x,z)$ and $X(x,z,y)$ may be travelled through arbitrary times before a transition is derived, it seems this process may have infinitely many transitions.

Fortunately this is not true. Each recursion unwinding amounts to a permutation of the formal parameter (x,y,z), and there are only finite many permutations over (x,y,z) can be generated by composing the two permutations (y,x,z) and (x,z,y). So what is needed for this example is to generate the permutation closure from (y,x,z) and (x,z,y), applying every permutation in this closure to $\overline{x}y.\mathbf{0}$, and then removing the unguarded occurrences $X(y,x,z)$ and $X(x,z,y)$.

The actual formulation of the axioms below is complicated by the possible presence of *conditions* and *restrictions* before unguarded occurrences of process variables – a typical unguarded occurrence has the form $\phi\nu\tilde{x}'X(\tilde{x})$, not just $X(\tilde{x})$, but the basic idea remains the same.

Call $< \phi_i, \tilde{x}'_i, \tilde{x}_i >$ a *crp-triple over* $\tilde{x}$ if $\tilde{x}_i$ is a permutation of $\tilde{x}$, $\{\tilde{x}'_i\} \subseteq \{\tilde{x}\}$, and $n(\phi_i) \subseteq \{\tilde{x}\}$. The composition of two crp-triples over $\tilde{x}$, $< \phi_i, \tilde{x}'_i, \tilde{x}_i >$ and $< \phi_j, \tilde{x}'_j, \tilde{x}_j >$

$$\frac{}{\nu\tilde{w}'X(\tilde{w}) \succ^{true} \nu\tilde{w}'X(\tilde{w})} \qquad \frac{T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})}{\tau.T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})} \qquad \frac{T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})}{T+U \succ^{\phi} \nu\tilde{w}'X(\tilde{w})}$$

$$\frac{U \succ^{\phi} \nu\tilde{w}'X(\tilde{w})}{T+U \succ^{\phi} \nu\tilde{w}'X(\tilde{w})} \qquad \frac{T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})}{\psi T \succ^{\psi\phi} \nu\tilde{w}'X(\tilde{w})} \qquad \frac{T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})}{(\tilde{x})T \succ^{\lambda\tilde{x}\phi} \lambda\tilde{x}\nu\tilde{w}'X(\tilde{w})}$$

$$\frac{F \succ^{\lambda\tilde{x}\phi} \lambda\tilde{x}\nu\tilde{w}'X(\tilde{w})}{F\tilde{x} \succ^{\phi} \nu\tilde{w}'X(\tilde{w})} \qquad \frac{F \succ^{\lambda\tilde{x}\phi} \lambda\tilde{x}\nu\tilde{w}'X(\tilde{w})}{\mathbf{fix}YF \succ^{\lambda\tilde{x}\phi} \lambda\tilde{x}\nu\tilde{w}'X(\tilde{w})}\ Y \not\equiv X$$

Figure 5: The Conditional Unguarded Relation

is defined as $<\phi_k, \tilde{x}'_k, \tilde{x}_k>$ where $\phi_k = (\nu_{\tilde{x}'_j}\phi_i)[\tilde{x}_j/\tilde{x}] \wedge \phi_j$, $\tilde{x}'_k = \tilde{x}'_j \circ (\tilde{x}_i[\tilde{x}_j/\tilde{x}])$, and $\tilde{x}_k = \tilde{x}_i \circ \tilde{x}_j$ ($\circ$ is permutation composition).

A set of crp-triples over $\tilde{x}$, K, is *closed* if the composition of any two elements of K is also in K. An arbitrary finite set of crp-triples over $\tilde{x}$ K can be closed up by repeatedly composing its elements, and this process will terminate in finite number of steps because only finite number of crp-triples over $\tilde{x}$ can be generated from K. We denote the closure of K by K^*.

Using the following axiom all "strongly" unguarded recursions can be removed:

$$\text{UNG1} \quad \mathbf{fix}X(\tilde{x})(\sum_{i\in I}\phi_i\nu\tilde{x}'_iX(\tilde{x}_i)+T) = \mathbf{fix}X(\tilde{x})(\sum_{<\phi_j,\tilde{x}'_j,\tilde{x}_j>\in K^*}\phi_j\nu\tilde{x}'_jT[\tilde{x}_j/\tilde{x}]+T)$$

where $K = \{<\phi_i, \tilde{x}'_i, \tilde{x}_i>\mid i \in I\}$.

A similar axiom takes care for immediate τ-loops:

$$\text{UNG2} \quad \mathbf{fix}X(\tilde{x})(\sum_{i\in I}\phi_i\tau.\nu\tilde{x}'_iX(\tilde{x}_i)+T) = \mathbf{fix}X(\tilde{x})(\sum_{<\phi_j,\tilde{x}'_j,\tilde{x}_j>\in K^*}\phi_j\tau.\nu\tilde{x}'_jT[\tilde{x}_j/\tilde{x}]+T)$$

To deal with deep-nested τ-loops, we need an auxiliary relation. The conditional free-unguarded relation $\succ^{\phi}$ is defined by the rules in Figure 5, where λ notation is used as purly syntactic device. Intuitively $T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})$ means $\nu\tilde{w}'X(\tilde{w})$ occurs free and unguarded in T with (accumulated) context condition ϕ. Now the last axiom for removing unguarded recursions:

$$\text{UNG3} \quad \mathbf{fix}X(\tilde{x})(\phi\tau.T+U) = \mathbf{fix}X(\tilde{x})(\phi\tau\psi.\nu\tilde{w}'X(\tilde{w})+\phi\tau.T[\mathbf{0}]+U)$$

where $T \succ^{\phi} \nu\tilde{w}'X(\tilde{w})$ and $T[\mathbf{0}]$ denotes the term obtained by replacing this occurrence of $\nu\tilde{w}'X(\tilde{w})$ with $\mathbf{0}$.

Let $\vdash^{rw} = \vdash^{grw} \cup\{UNG1, UNG2, UNG3\}$. Given any term which contain unguarded recursions, we can first use UNG3 to "lift" deep-nested τ-loops to the top-level which can then be excised using UNG1 and UNG2. So we have

Proposition 5.1 *For any term T with $fn(T) \subseteq \{\tilde{x}\}$, there exists a guarded term T' with $fn(T') \subseteq \{\tilde{x}\}$, such that $\vdash^{rw} T = T'$.*

Combining this proposition with Theorem 4.4 gives the main result of this paper:

Theorem 5.2 *$T \simeq^{\phi}_{L} U$ implies $\vdash^{rw} \phi \rhd T = U$.*

6 Conclusions

We have presented complete proof systems for late weak observation congruence in the finite-control π-calculus. These results are achieved in two steps: First we work with guarded recursions and the proof system is obtained by combining that for weak congruence in finite π-calculus ([5]) and that for strong congruence in guarded finite-control π-calculus ([6]); We then formulate sound laws sufficient to remove arbitrary unguarded recursions, thus extend the result to the whole language of finite-control π-calculus.

Because of space limitation we have only discussed the proof system for late weak bisimulation. There are two ways to adopt it to the early weak equivalence: replacing the INPUT rule with the more general one:

$$\text{E-INPUT}\ \frac{\phi \rhd \sum_{i\in I} \tau.T_i = \sum_{j\in J} \tau.U_j}{\phi \rhd \sum_{i\in I} a_i(x).T = \sum_{j\in J} b_j(x).U} \quad \begin{array}{l} \phi \Rightarrow a_i = b_j,\ i \in I,\ j \in J \\ x \notin n(\phi) \end{array}$$

or adding the following "early" axiom, due to Parrow and Sangiorgi [10]:

$$\text{EA} \quad a(x).T + a(x).U = a(x).T + a(x).U + a(x).([x=y]T + [x \neq y]U)$$

It is not difficult to see that EA can be derived from E-INPUT in the presence of the other rules.

References

[1] Michele Boreale and Rocco De Nicola. A symbolic semantics for the π-calculus. *Information and Computation*, 126(1):34–52, 1996.

[2] M. Dam. Model checking mobile processes. *Information and Computation*, 129:35–51, 1996.

[3] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.

[4] H. Lin. Symbolic bisimulations and proof systems for the π-calculus. Report 7/94, Computer Science, University of Sussex, 1994.

[5] H. Lin. Complete inference systems for weak bisimulation equivalences in the π-calculus. In *TAPSOFT'95*, LNCS 915. Springer–Verlag, 1995.

[6] H. Lin. Unique fixpoint induction for the π-calculus. In *CONCUR'95*, LNCS 962. Springer–Verlag, 1995.

[7] R. Milner. A complete inference system for a class of regular behaviours. *J. Computer and System Science*, 28:439–466, 1984.

[8] R. Milner. A complete axiomatisation for observational congruence of finite-state behaviours. *Information and Computation*, 81:227–247, 1989.

[9] R. Milner, J. Parrow, and D. Walker. A calculus of mobile proceses, part I,II. *Information and Computation*, 100:1–77, 1992.

[10] J. Parrow and D. Sangiorgi. Algebraic theories for name-passing calculi. *Information and Computation*, 120(2):174–197, 1995.

[11] Davide Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996.