Symbolic Characterizations and Algorithms for Hyperequivalence

Björn Victor

December 1998

Abstract

We present a symbolic transition system for the fusion calculus together with alternative characterizations of strong and weak hyperequivalence, without universal quantification over substitutions. We show that this symbolic bisimulation equivalence coincides with the standard hyperequivalence. Based on the symbolic semantics we present an algorithm for deciding the hyperequivalences and its implementation in the Mobility Workbench. The space requirements are significantly reduced compared to earlier work, making it reasonable to use a normal workstation for verification even of large examples.

1 Introduction

A formal method is practically useless if it does not come with tools for automated analysis. With the recent development of the fusion calculus [PV98, VP98] as a simplification of the π-calculus with a significant gain in expressiveness, it is natural to work towards automated tools. In this paper we develop the theory necessary for automated reasoning about equivalence in the fusion calculus, and describe our implementation in the Mobility Workbench.

In the fusion calculus, the effects of communication can be anywhere from local, involving only one of the two communicating agents, to global, involving all processes in the system. This is possible because input actions do not bind names. The communication actions, input and output, are completely symmetric; neither has a binding effect. The *scope* operator limits the effects of communication, and restricts the use of the scoped name as subject of actions. The communication effects are called *fusion* actions; the objects of the communicating input and output actions are "fused", making them identical even in agents not participating in the communication. The bisimulation congruence on fusion calculus agents, *hyperequivalence*, must thus be closed under arbitrary substitutions after each transition in the bisimulation, since in an arbitrary context any fusion action is possible at any time.

Most bisimulation equivalences in value-passing calculi are defined with universal quantification of variable instantiations. The three major equivalences in π-calculus (or value-passing CCS) differ from hyperequivalence as follows. For two bisimilar agents P and Q, if P can perform an input transition, in *early* bisimulation, a simulating transition must be found for each possible value of the input name or variable; in *late* bisimulation, a transition which simulates the transition of P for all possible values must be found; in *open* bisimulation, the agents must be bisimilar for any substitution which respects the uniqueness of new names after each transition.

Computing bisimulation equivalence using these universal quantifications can be very inefficient. But using *symbolic transition systems*, the bisimulation equivalences are characterized in a computationally more efficient way, without universal quantification. We can then give decision procedures for efficient implementations in automated tools.

In the first part of the paper, we present a symbolic transition system and a symbolic characterization of hyperequivalence, and show that the symbolic bisimulation equivalence coincides with hyperequivalence. Based on this, a decision procedure has been implemented.

Related work. Symbolic semantics for process calculi was first introduced for data-independent value-passing CCS by Jonsson and Parrow [JP93] and for full value-passing CCS by Hennessy and Lin [HL95]; based on their work, symbolic semantics for the π-calculus were given by Sangiorgi [San96], Liu [Liu94], Lin [Lin94, Lin95], and Boreale & De Nicola [BD96]. The work on value-passing CCS was later continued and improved on by Lin [Lin96]. Characterizations of π-calculus bisimulations suitable for partition-refinement algorithms were developed by Montanari and Pistore [MP95] and by Pistore and Sangiorgi [PS96].

Overview. In Section 2 we recall the fusion calculus and its semantics; in Section 3 we develop a symbolic semantics. Section 4 describes the implementation of decision procedures based on the symbolic semantics, and finally in Section 5 we conclude and present some ideas for further research.

2 The Fusion Calculus

In this section we first review the syntax and operational semantics of a version of the fusion calculus with match, mismatch and recursion. We then review the strong and weak bisimulation equivalences and congruences from [PV98, VP98].

2.1 Syntax and Semantics

We assume an infinite set $\mathcal{N}$ of *names* ranged over by $u, v, \ldots, z$. Like in the π-calculus, names represent communication channels, which are also the values transmitted. We write $\widetilde{x}$ for a (possibly empty) finite sequence $x_1 \cdots x_n$ of names. φ ranges over total equivalence relations over $\mathcal{N}$ (i.e. equivalence relations with $\mathrm{dom}(\varphi) = \mathcal{N}$) with only finitely many non-singular equivalence classes. If x and y are related by φ we write $x\ \varphi\ y$. We write $\{\widetilde{x} = \widetilde{y}\}$ to mean the smallest such equivalence relation relating each x_i with y_i, and write $\mathbf{1}$ for the identity relation. As a consequence, a fusion written $\{x = x\}$ is the same as $\{y = y\}$, namely $\mathbf{1}$, and $\{x = y, x = z\}$ is the same as $\{x = y, y = z\}$.

Definition 1 *The* free actions*, ranged over by α, and the* agents*, ranged over by $P, Q, \ldots$, are defined by*

$$
\begin{array}{llllll}
\alpha ::= & u\widetilde{x} & \text{(Input)} & P ::= & \mathbf{0} & \text{(Inaction)} \\
 & \overline{u}\widetilde{x} & \text{(Output)} & & \alpha\,.\,Q & \text{(Prefix)} \\
 & \varphi & \text{(Fusion)} & & Q + R & \text{(Summation)} \\
 & & & & Q \mid R & \text{(Composition)} \\
 & & & & (x)Q & \text{(Scope)} \\
 & & & & [x = y]Q & \text{(Match)} \\
 & & & & [x \neq y]Q & \text{(Mismatch)} \\
 & & & & A(\widetilde{x}) & \text{(Identifier)}
\end{array}
$$

Input and output actions are collectively called *communication* actions. In these, the names $\widetilde{x}$ are the *objects* of the action, and the name u is the *subject*. We write a to stand for either u or $\overline{u}$, thus $a\widetilde{x}$ is the general form of a communication action. Fusion actions have neither subject nor objects.

Prefixing an agent Q means that the agent must perform the action α before acting like Q. Performing a fusion here represents an obligation to treat all related names as identical. We often omit a trailing $\mathbf{0}$ and write α for $\alpha\,.\,\mathbf{0}$ if no confusion can arise. Like in the π-calculus, Summation is alternative choice and Composition lets agents act in parallel.

The Scope $(x)Q$ limits the scope of x to Q; no visible communication action of $(x)Q$ can have x as its subject, and fusion effects with respect to x are limited to Q. Restriction and input binding of the π-calculus can be recovered as special cases of Scope [PV98].

A Match $[x = y]Q$ acts like Q if x and y are the same name; a Mismatch $[x \neq y]Q$ acts like Q if x and y are *not* the same. We let M, N range over Match or Mismatch operators and A over a possibly infinite set of identifiers, each with an associated nonnegative arity k and agent definition $A(x_1, \ldots, x_k) \stackrel{def}{=} P$ with pairwise distinct x_i.

The name x is said to be *bound* in $(x)P$. We write $(\widetilde{x})P$ for $(x_1)\cdots(x_n)P$. The *free names* in P, denoted $\mathrm{fn}(P)$, are the names in P with a non-bound occurrence; here the names occurring in the fusion φ is defined to be the names in the non-singular equivalence classes, i.e., in the relation $\varphi - \mathbf{1}$. Notice that since the syntax $\{x = x\}$ means $\mathbf{1}$, the free names of that fusion do not include x, but is the empty set. As usual we will not distinguish between alpha-variants of agents, i.e., agents differing only in the choice of bound names. We write "match sequence" for a sequence of match *and* mismatch operators, ranged over by $\widetilde{M}$, $\widetilde{N}$, and we say that $\widetilde{M}$ *implies* $\widetilde{N}$, written $\widetilde{M} \Rightarrow \widetilde{N}$, if the conjunction of all matches and mismatches in $\widetilde{M}$ logically implies all elements in $\widetilde{N}$, and that $\widetilde{M} \Leftrightarrow \widetilde{N}$ if $\widetilde{M}$ and $\widetilde{N}$ imply each other. We write $\sum_{i=1}^{n} P_i$ for finite general summation, $P_1 + \cdots + P_n$.

The action of a transition may be free or bound:

Definition 2 *The* actions*, ranged over by γ, consist of the fusion actions and of communication actions of the form $(z_1)\cdots(z_n)a\widetilde{x}$ (written $(\widetilde{z})a\widetilde{x}$), where $n \geq 0$ and all names in $\widetilde{z}$ are also in $\widetilde{x}$. If $n > 0$ we say it is a* bound *action.*

Note that there are no bound fusion actions. In the bound communication actions above, $\widetilde{z}$ are the *bound objects* and the names in $\widetilde{x}$ that are not in $\widetilde{z}$ are the *free objects*. Free actions have no bound objects. We further write $\mathrm{n}(\gamma)$ to mean all names occurring in γ (i.e., also including the subject of communication actions and the names in non-singular equivalence classes in fusion actions).

For convenience we define $\varphi\backslash z$ to mean $\varphi \cap (\mathcal{N} - \{z\})^2 \cup \{(z, z)\}$, i.e., the equivalence relation φ with all references to z removed (except for the identity). For example,
$\{x = y\}\backslash y = \mathbf{1}$, and $\{x = z, z = y\}\backslash z = \{x = y\}$.

We now define a structural congruence which equates all agents we will never want to distinguish for any semantic reason, and then use this when giving the transitional semantics.

Definition 3 *The* structural congruence*, $\equiv$, between agents is the least congruence satisfying the abelian monoid laws for Summation and Composition (associativity, commutativity and $\mathbf{0}$ as identity), and the scope laws*

$$(x)\mathbf{0} \equiv \mathbf{0}, \quad (x)(y)P \equiv (y)(x)P, \quad (x)(P + Q) \equiv (x)P + (x)Q$$
$$(x)MP \equiv M(x)P, \quad \textit{if } x \notin \mathrm{n}(M)$$

$$\text{PREF}\ \frac{-}{\alpha\,.\,P \xrightarrow{\alpha} P} \qquad \text{SUM}\ \frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'} \qquad \text{PAR}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\text{COM}\ \frac{P \xrightarrow{u\widetilde{x}} P',\ \ Q \xrightarrow{\overline{u}\widetilde{y}} Q',\ \ |\widetilde{x}| = |\widetilde{y}|}{P \mid Q \xrightarrow{\{\widetilde{x}=\widetilde{y}\}} P' \mid Q'} \qquad \text{SCOPE}\ \frac{P \xrightarrow{\varphi} P',\ z\,\varphi\,x,\ z \neq x}{(z)P \xrightarrow{\varphi \backslash z} P'\{x/z\}}$$

$$\text{PASS}\ \frac{P \xrightarrow{\alpha} P',\ z \notin \mathrm{n}(\alpha)}{(z)P \xrightarrow{\alpha} (z)P'} \qquad \text{OPEN}\ \frac{P \xrightarrow{(\widetilde{y})a\,\widetilde{x}} P',\ z \in \widetilde{x} - \widetilde{y},\ a \notin \{z, \overline{z}\}}{(z)P \xrightarrow{(z\widetilde{y})a\,\widetilde{x}} P'}$$

$$\text{MATCH}\ \frac{P \xrightarrow{\alpha} P'}{[x=x]P \xrightarrow{\alpha} P'} \qquad \text{MISMATCH}\ \frac{P \xrightarrow{\alpha} P',\ x \neq y}{[x \neq y]P \xrightarrow{\alpha} P'}$$

Table 1: The Fusion Calculus: Laws of action.

and also the scope extension *law* $P \mid (z)Q \equiv (z)(P \mid Q)$ *where* $z \notin \mathrm{fn}(P)$, *and the law for process identifiers:* $A(\widetilde{y}) \equiv P\{\widetilde{y}/\widetilde{x}\}$ *if* $A(\widetilde{x}) \stackrel{def}{=} P$.

Definition 4 *The family of* transitions $P \xrightarrow{\gamma} Q$ *is the least family satisfying the laws in Table 1. In this definition structurally equivalent agents are considered the same, i.e., if* $P \equiv P'$ *and* $Q \equiv Q'$ *and* $P \xrightarrow{\gamma} Q$ *then also* $P' \xrightarrow{\gamma} Q'$.

Use of the SCOPE rule entails a substitution of the scoped name z for a nondeterministically chosen name x, different from z but related to it by φ. For the purpose of the equivalences defined below it will not matter which such x replaces z. The only rule dealing with bound actions is OPEN. Using structural congruence, pulling the relevant scope to the top level, we can still infer e.g. $P \mid (x)ayx\,.\,Q \xrightarrow{(x)ayx} P \mid Q$ using PREF, PAR and OPEN (provided $x \notin \mathrm{fn}(P)$, otherwise an alpha-conversion is necessary). It is clear that for the purpose of the semantics, fusion prefixes could be regarded as derived forms since $\{\widetilde{y} = \widetilde{z}\}\,.\,P$ has exactly the same transitions as $(u)(\overline{u}\widetilde{y}\,.\,\mathbf{0} \mid u\widetilde{z}\,.\,P)$ when $u \notin \mathrm{fn}(P)$. (In the same way the τ prefix could be regarded as derived in CCS and in the π-calculus.)

2.2 Equivalences

From [PV98] we recall the definition of strong bisimulation in the fusion calculus, and from [VP98] the weak bisimulation.

Definition 5 *A substitution σ* agrees with *the fusion φ if $\forall x, y : x\,\varphi\,y \Leftrightarrow \sigma(x) = \sigma(y)$. A substitution σ is* idempotent *if $\sigma\sigma = \sigma$. A* substitutive effect *of a fusion φ is an idempotent substitution σ agreeing with φ (i.e., σ sends all members of each equivalence class of φ to one representative in the class). The only substitutive effect of a communication action is the identity substitution.*

For example, the substitutive effects of $\{x = y\}$ are $\{x/y\}$ and $\{y/x\}$. Note that not all substitutions which agree with φ are substitutive effects of φ. For example any injective substitution agrees with $\mathbf{1}$, but the only substitutive effect of $\mathbf{1}$ is the identity substitution.

Definition 6 *A* bisimulation *is a binary symmetric relation $\mathcal{S}$ between agents such that $P\;\mathcal{S}\;Q$ implies:*

If $P \xrightarrow{\gamma} P'$ with $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ *then*

$Q \xrightarrow{\gamma} Q'$ and $P'\sigma\;\mathcal{S}\;Q'\sigma$ for some substitutive effect σ of γ.

P is bisimilar to Q, written $P \dot{\sim} Q$, if $P\;\mathcal{S}\;Q$ for some bisimulation $\mathcal{S}$.

The interesting point in this definition is the treatment of fusion actions. A fusion $\{x = y\}$ represents an obligation to make x and y equal everywhere. Therefore, if γ above is such a fusion, it only makes sense to relate P' and Q' when a substitution $\{y/x\}$ or $\{x/y\}$ has been performed. Note that it does not matter which substitution we choose, since $P\{x/y\} \dot{\sim} Q\{x/y\}$ implies $P\{y/x\} \dot{\sim} Q\{y/x\}$, by the simple fact that $(P\{x/y\})\{y/x\} \equiv P\{y/x\}$ and that bisimilarity is closed under injective substitutions.

The *weak* bisimulation and bisimilarity $\dot{\approx}$ are in [VP98] defined by replacing $Q \xrightarrow{\gamma} Q'$ by $Q \overset{\gamma}{\Longrightarrow} Q'$ in Definition 6. For the case that γ is a fusion the definition of $\overset{\gamma}{\Longrightarrow}$ requires some care. The intuition is that $\overset{\varphi}{\Longrightarrow}$ represents a sequence of actions with "observable content" φ. But fusions are "observable" only through their substitutive effects. They have no subject and the environment cannot synchronize with them and keep track of, e.g., how many have been performed. Therefore we allow $\overset{\varphi}{\Longrightarrow}$ to be a sequence of fusions whose aggregated effect is φ.

Definition 7 *Define the* composition *of two transitions, $\circ$, by $P(\xrightarrow{\gamma} \circ \xrightarrow{\gamma'})Q$ iff there exists an agent P' such that $P \xrightarrow{\gamma} P'$ and $P'\sigma_\gamma \xrightarrow{\gamma'} Q$, where σ_γ is a substitutive effect of γ. Define the* conjunction *of two fusions φ and ψ, written $\varphi \wedge \psi$, to be the least equivalence relation containing φ and ψ, i.e.,*

$(\varphi \cup \psi)^*$. *Define the* weak transition $\overset{\gamma}{\Longrightarrow}$ *by the following:* $P \overset{\gamma}{\Longrightarrow} Q$ *means that for some* $n \geq 0$, $P \xrightarrow{\gamma_1} \circ \cdots \circ \xrightarrow{\gamma_n} Q$ *and either of*

1. γ *is a communication and* $\gamma = \gamma_i$ *for some* i *and* $\gamma_j = \mathbf{1}$ *for all* $j \neq i$, *or*

2. γ *and all* γ_i *are fusions and* $\gamma = \gamma_1 \wedge \cdots \wedge \gamma_n$. *Here we allow* $n = 0$ *where the empty conjunction is* $\mathbf{1}$, *in other words* $P \overset{\mathbf{1}}{\Longrightarrow} P$ *holds for all* P.

Definition 8 *A* weak bisimulation *is a binary symmetric relation* $\mathcal{S}$ *between agents such that* $P\ \mathcal{S}\ Q$ *implies:*

If $P \xrightarrow{\gamma} P'$ *with* $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ *then*

$Q \overset{\gamma}{\Longrightarrow} Q'$ *and* $P'\sigma_\gamma\ \mathcal{S}\ Q'\sigma_\gamma$ *for some substitutive effect* σ_γ *of* γ.

Two agents P *and* Q *are* weakly bisimilar, *written* $P \dot{\approx} Q$, *if they are related by a weak bisimulation.*

An illuminating exercise is to verify that

$$\{x=y\}\,.\,[x=y]\{v=w\}\,.\,P \quad \dot{\approx} \quad \{x=y\}\,.\,\{v=w\}\,.\,P + \{x=y, v=w\}\,.\,P$$

but not

$$\{x=y\}\,.\,\{v=w\}\,.\,P \quad \dot{\approx} \quad \{x=y, v=w\}\,.\,P$$

since the RHS cannot simulate the transition $\xrightarrow{\{x=y\}}$.

From [PV98] and [VP98] we know that $\dot{\sim}$ and $\dot{\approx}$ are indeed equivalences and that they are not congruences. To find the congruences we close the bisimulations under arbitrary substitutions:

Definition 9 *A* (weak) hyperbisimulation *is a substitution closed (weak) bisimulation, i.e., a (weak) bisimulation* $\mathcal{S}$ *with the property that* $P\ \mathcal{S}\ Q$ *implies* $P\sigma\ \mathcal{S}\ Q\sigma$ *for any substitution* σ. *Two agents* P *and* Q *are* (weakly) hyperequivalent, *written* $P \sim Q$ *(*$P \approx Q$*), if they are related by a (weak) hyperbisimulation.*

Theorem 10 [PV98] *Hyperequivalence is the largest congruence contained in bisimilarity.*

The corresponding relation does not quite hold for weak hyperequivalence, but it holds if we replace the summation operator by *guarded* summation $\sum_{i \in I} \alpha_i\,.\,P_i$. (This is as expected; also in CCS and in the π-calculus weak bisimulation equivalence fails to be a congruence, and for the same reason.)

Theorem 11 [VP98] *With guarded summation, weak hyperequivalence is the largest congruence in weak bisimilarity.*

3 Symbolic Semantics

In this section we present a symbolic semantics for the fusion calculus, along the lines of symbolic semantics for the π-calculus. Symbolic semantics are used to give efficient characterizations of bisimulation equivalences for value-passing calculi: Hennessy and Lin [HL95] did this for value-passing CCS, Sangiorgi [San96, published as a technical report in 1993] gave symbolic semantics for the π-calculus and a symbolic characterization of open bisimulation equivalence, which was later used by the present author and Faron Moller [VM94] in the Mobility Workbench, an automated verification tool for mobile processes (see below). Lin [Lin94, Lin95] used symbolic transition systems to get efficient characterizations of late and early bisimulation equivalences in the π-calculus.

$$\textsc{s-pref}\ \frac{-}{\alpha\,.\,P \xmapsto{\emptyset,\alpha} P} \qquad \textsc{s-sum}\ \frac{P \xmapsto{\widetilde{M},\alpha} P'}{P+Q \xmapsto{\widetilde{M},\alpha} P'}$$

$$\textsc{s-par}\ \frac{P \xmapsto{\widetilde{M},\alpha} P'}{P \mid Q \xmapsto{\widetilde{M},\alpha} P' \mid Q}$$

$$\textsc{s-com}\ \frac{P \xmapsto{\widetilde{M},u\widetilde{x}} P',\ \ Q \xmapsto{\widetilde{N},\overline{v}\widetilde{y}} Q',\ \ |\widetilde{x}| = |\widetilde{y}|,\ \widetilde{M}\widetilde{N}[u=v] \text{ consistent}}{P \mid Q \xmapsto{\widetilde{M}\widetilde{N}[u=v],\{\widetilde{x}=\widetilde{y}\}} P' \mid Q'}$$

$$\textsc{s-scope}\ \frac{P \xmapsto{\widetilde{M},\varphi} P',\ z\,\varphi\,x,\ z \neq x,\ \forall y \neq z : \neg(\widetilde{M} \Rightarrow [z=y])}{(z)P \xmapsto{\widetilde{M}\backslash z,\varphi\backslash z} P'\{x/z\}}$$

$$\textsc{s-pass}\ \frac{P \xmapsto{\widetilde{M},\alpha} P',\ z \notin \mathrm{n}(\widetilde{M^{+}},\alpha)}{(z)P \xmapsto{\widetilde{M}\backslash z,\alpha} (z)P'}$$

$$\textsc{s-open}\ \frac{P \xmapsto{\widetilde{M},(\widetilde{y})a\,\widetilde{x}} P',\ z \in \widetilde{x} - \widetilde{y},\ a \notin \{z,\overline{z}\},\ z \notin \mathrm{n}(\widetilde{M^{+}})}{(z)P \xmapsto{\widetilde{M}\backslash z,(z\widetilde{y})a\,\widetilde{x}} P'}$$

$$\textsc{s-match}\ \frac{P \xmapsto{\widetilde{M},\alpha} P',\ \widetilde{M}[x=y] \text{ consistent}}{[x=y]P \xmapsto{\widetilde{M}[x=y],\alpha} P'}$$

$$\textsc{s-mismatch}\ \frac{P \xmapsto{\widetilde{M},\alpha} P',\ \widetilde{M}[x\neq y] \text{ consistent}}{[x\neq y]P \xmapsto{\widetilde{M}[x\neq y],\alpha} P'}$$

Table 2: The Fusion Calculus: Symbolic transition system.

In Table 2 we present the symbolic transition system. We say that a match sequence is *consistent* if $\widetilde{M}\sigma \Leftrightarrow \mathsf{true}$ for some σ. $\widetilde{M}$ is *consistent with* $\widetilde{N}$ if $\widetilde{M}\widetilde{N}$ is consistent, and $\widetilde{M}$ is consistent with φ if it is consistent with a sequence of matches corresponding to the non-singular equivalence classes of φ. We write $\widetilde{M}^+$ for the match part of $\widetilde{M}$, and $\widetilde{M}^-$ for the mismatch part, and $\widetilde{M}\backslash z$ for a match sequence such that $\widetilde{M}\backslash z \Rightarrow [x \Box y]$ iff $\widetilde{M} \Rightarrow [x \Box y]$ and $z \notin \{x, y\}$, for $\Box \in \{=, \neq\}$, i.e., an equivalent match sequence but without mention of z. For example, $[x = y][y = z]\backslash y \Leftrightarrow [x = z]$. We write $\emptyset$ for the empty match sequence.

A transition in the symbolic transition system is on the form $P \xmapsto{\widetilde{M},\gamma} Q$, where $\widetilde{M}$ is the enabling condition of the action γ in the sense that $\widetilde{M}$ represents the equalities and inequalities a minimal substitution $\sigma_{\widetilde{M}}$ must make true in order for $P\sigma_{\widetilde{M}}$ to perform the corresponding action in the original labelled transition system. For example,

$$[x \neq y]u\,.\,P \xmapsto{[x\neq y],u} P \qquad \text{and} \qquad [x \neq y]u\,.\,P \mid \overline{v}\,.\,Q \xmapsto{[x\neq y,u=v],\mathbf{1}} P \mid Q$$

Using the symbolic transition system, we define a bisimulation equivalence without universal quantification over substitutions. A transition of one process may need to be simulated by a *set* of transitions of the other process. Consider the following example:

$$a\,.\,P \sim [x = y]a\,.\,P + [x \neq y]a\,.\,P$$

For each substitution, the $\xrightarrow{a}$ transition of the LHS is simulated by *different* summands of the RHS depending on whether the substitution makes x and y the same or not. In symbolic hyperbisimulation, the action $\xmapsto{\emptyset,a}$ of the LHS must therefore be matched by the *set* containing the two actions $\xmapsto{[x=y],a}$ and $\xmapsto{[x\neq y],a}$ of the RHS. The conditions of the LHS (here $\emptyset$ or "true") imply the disjunction of the conditions of the RHS actions (here $x = y \vee x \neq y$). The important point when simulating with a set of symbolic transitions is that any substitution which satisfies the LHS condition satisfies one of the RHS conditions. Symbolic bisimulation thus uses a set of process relations, indexed by match sequences representing assertions of equalities between names.

Definition 12 *A substitution σ* satisfies *a match sequence $\widetilde{M}$, written $\sigma \models \widetilde{M}$, if for all x, y, $\widetilde{M} \Rightarrow [x = y]$ implies $\sigma(x) = \sigma(y)$ and $\widetilde{M} \Rightarrow [x \neq y]$ implies $\sigma(x) \neq \sigma(y)$.*

Definition 13 *The set $\mathcal{S} = \{S^C : C \in \mathcal{C}\}$ (for $\mathcal{C}$ a set of sequences of match operators) of binary symmetric process relations is a* symbolic hyperbisimulation *if for each $S^C \in \mathcal{S}$, $(P, Q) \in S^C$ implies:*

If $P \xmapsto{\widetilde{M},\gamma} P'$ *with* $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ *and* $\widetilde{M}$ *consistent with* C*, then*
for some $T \subseteq \{Q \xmapsto{\widetilde{N}_i,\delta_i} Q'_i : \gamma\sigma = \delta_i\sigma$*, for all* σ *which agree with* $C\widetilde{M}\widetilde{N}_i\}$
$C\widetilde{M} \Rightarrow \bigvee \widetilde{N}_i$*, and for all* i*,* $(P', Q'_i) \in S^{(L_i^+)} \in \mathcal{S}$*, where*
$L_i = C\widetilde{M}\widetilde{N}_i\widetilde{M}_\gamma$ *and* $\widetilde{M}_\gamma$ *a sequence of matches such that*
any substitutive effect of γ *agrees with* $\widetilde{M}_\gamma$*.*

P *is* symbolically C-hyperequivalent *to* Q*, written* $P \simeq_C Q$*, if* $(P, Q) \in S^C \in \mathcal{S}$ *for some symbolic hyperbisimulation* $\mathcal{S}$*. We write* $P \simeq Q$ *if* $P \simeq_\emptyset Q$*.*

The *weak* symbolic hyperbisimulation and hyperequivalence $\approxeq$ are defined by replacing $Q \xmapsto{\widetilde{N}_i,\delta_i} Q'_i$ by $Q \mathrel{\big|\!\!\xLongrightarrow{\widetilde{N}_i,\delta_i}} Q'_i$ in Definition 13. We define $\mathrel{\big|\!\!\Longrightarrow}$ as follows:

Definition 14

$$P \mathrel{\big|\!\!\xLongrightarrow{\widetilde{M},\gamma}} Q \text{ if } \begin{cases} P \xmapsto{\widetilde{M}_1,\mathbf{1}} \cdots \xmapsto{\widetilde{M}_n,\mathbf{1}} \xmapsto{\widetilde{L},\gamma} \xmapsto{\widetilde{N}_1,\mathbf{1}} \cdots \xmapsto{\widetilde{N}_m,\mathbf{1}} Q \\ \quad \textit{if } \gamma \textit{ a communication action, } m,n \geq 0, \\ \quad \widetilde{M} \Leftrightarrow (\bigwedge_{i=1}^{n} \widetilde{M}_i) \wedge \widetilde{L} \wedge (\bigwedge_{i=1}^{m} \widetilde{N}_i) \\ P \xmapsto{\widetilde{M}_1,\psi_1} \cdots \xmapsto{\widetilde{M}_n,\psi_n} Q \\ \quad \textit{if } \gamma \textit{ a fusion action, } n \geq 0,\ \widetilde{M}_{k+1} \textit{ is consistent with} \\ \quad \psi_1 \cdots \psi_k,\ \widetilde{M} \Leftrightarrow \bigwedge_{i=1}^{n} (\widetilde{M}_i \sigma_{\psi_1 \cdots \psi_{i-1}}) \textit{ is consistent, where} \\ \quad \sigma_{\psi_1 \cdots \psi_{i-1}} \textit{ is a substitutive effect of } \psi_1 \wedge \cdots \wedge \psi_{i-1} \textit{ and} \\ \quad \gamma = \psi_1 \wedge \cdots \wedge \psi_n. \end{cases}$$

Note that in Definition 7 of the standard weak transitions a substitution is performed on each intermediary agent in the case of a fusion action. In Definition 14 above the substitution needs only be done on the conditions of the intermediary (strong) transitions. This is much more efficient from the point of view of an implementation, but makes the proof of Lemma 24 below a little more complicated, requiring Lemmas 20 and 21.

The adequacy of the symbolic equivalences is given by showing that they coincide with their non-symbolic counterparts. To show that, we use the following Lemmas.

Definition 15 *A substitution* σ respects *a match sequence* $\widetilde{M}$ *if* $\widetilde{M} \Rightarrow [x \neq y]$ *implies* $\sigma(x) \neq \sigma(y)$*, for all* x, y*.*

It may be helpful to relate the concepts of *agree with*, *satisfies* and *respects*: "σ agrees with $\widetilde{M}$" means σ makes exactly those names equal which $\widetilde{M}$ tests for equality; "σ satisfies $\widetilde{M}$" means σ may make more names equal as long as it does not make $\widetilde{M}$ inconsistent; "σ respects $\widetilde{M}$" only means that σ does not make $\widetilde{M}$ inconsistent.

In the fusion calculus with mismatch, the monotonicity property of [MPW92] does not hold: $P \xrightarrow{\alpha} P'$ does not imply $P\sigma \xrightarrow{\alpha\sigma} P'\sigma$. However, in the symbolic semantics we can formulate a limited monotonicity property:

Lemma 16 *If $P \xmapsto{\widetilde{M},\gamma} P'$ then $P\sigma \xmapsto{\widetilde{M}\sigma,\gamma\sigma} P'\sigma$ for any σ which respects $\widetilde{M}$.*

PROOF: Easy induction on the depth of inference. □

Next we establish the correspondence between symbolic and standard transitions.

Lemma 17

1. *If $P \xmapsto{\widetilde{M},\gamma} P'$ then $P\sigma \xrightarrow{\gamma\sigma} P'\sigma$ for any $\sigma \models \widetilde{M}$.*

2. *If $P\sigma \xrightarrow{\gamma'} P''$ then $\exists \widetilde{M}, \gamma, P' : P \xmapsto{\widetilde{M},\gamma} P'$ such that $\sigma \models \widetilde{M}$, $\gamma' = \gamma\sigma$, and $P'' \equiv P'\sigma$.*

PROOF:

1. By induction on the depth of the inference of the transition. The only non-trivial case is S-COM:

   Suppose $P \equiv P_1 \mid P_2 \xmapsto{\widetilde{M},\{\tilde{x}=\tilde{y}\}} P_1' \mid P_2'$ because $P_1 \xmapsto{M_1,u\tilde{x}} P_1'$ and $P_2 \xmapsto{\widetilde{M}_2,\overline{v}\tilde{y}} P_2'$, with $\widetilde{M} \Leftrightarrow \widetilde{M}_1\widetilde{M}_2[u = v]$. For any σ s.t. $\sigma \models \widetilde{M}_1\widetilde{M}_2[u = v]$, σ respects both $\widetilde{M}_1$ and $\widetilde{M}_2$, so by Lemma 16, $P_1\sigma \xmapsto{\widetilde{M}_1\sigma,(u\tilde{x})\sigma} P_1'\sigma$ and $P_2\sigma \xmapsto{\widetilde{M}_2\sigma,(\overline{v}\tilde{y})\sigma} P_2'\sigma$. And since $\sigma \models \widetilde{M}_1$ and $\sigma \models \widetilde{M}_2$, $P_1\sigma \xrightarrow{(u\tilde{x})\sigma} P_1'\sigma$ and $P_2\sigma \xrightarrow{(\overline{v}\tilde{y})\sigma} P_2'\sigma$ and $\sigma \models [u = v]$, it follows that $(P_1 \mid P_2)\sigma \xrightarrow{\{\tilde{x}=\tilde{y}\}\sigma} (P_1' \mid P_2')\sigma$.

2. Simple induction on the depth of inference. □

In the rest of this subsection, we write σ_φ for a substitutive effect of φ. Note that all substitutive effects are idempotent substitutions.

Lemma 18 *For any idempotent substitution σ and any x and y, $x\sigma\sigma_{\varphi\sigma} = y\sigma\sigma_{\varphi\sigma}$ iff $x\sigma\sigma_\varphi = y\sigma\sigma_\varphi$.*

The adequacy of the symbolic equivalence is given by showing that it coincides with its non-symbolic counterpart, using the correspondence between symbolic and non-symbolic transitions.

Theorem 19 *$P \sim Q$ iff $P \simeq Q$.*

PROOF:

$\Rightarrow$: We show that $\mathcal{S} = \{S^C = \{(P,Q) : P\sigma \sim Q\sigma \text{ for all } \sigma \models C\}\}$ is a symbolic hyperbisimulation.

For $(P,Q) \in S^C$, if $P \xmapsto{\widetilde{M},\gamma} P'$, then for any $\sigma \models C\widetilde{M}$, by Lemma 17 $P\sigma \xrightarrow{\gamma\sigma} P'\sigma$, and $P\sigma \sim Q\sigma$, so $Q\sigma \xrightarrow{\gamma\sigma} Q''$ such that $P'\sigma\sigma_{\gamma\sigma} \sim Q''\sigma_{\gamma\sigma}$.

By Lemma 17 there exist $Q', \gamma', \widetilde{N}$ such that $Q \xmapsto{\widetilde{N},\gamma'} Q'$ and $\sigma \models \widetilde{N}$, $\gamma\sigma = \gamma'\sigma$ and $Q'' \equiv Q'\sigma$, so, $P'\sigma\sigma_{\gamma\sigma} \sim Q'\sigma\sigma_{\gamma\sigma}$.

Let $\widetilde{M_\gamma}$ be a sequence of matches such that any substitutive effect of γ agrees with $\widetilde{M_\gamma}$. We know $\sigma \models C\widetilde{M}\widetilde{N}$ and $\sigma\gamma \models \widetilde{M_\gamma}$, so $\sigma\sigma_\gamma \models (C\widetilde{M}\widetilde{N}\widetilde{M_\gamma})^+$, and by Lemma 18 $\sigma\sigma_{\gamma\sigma} \models (C\widetilde{M}\widetilde{N}\widetilde{M_\gamma})^+$, so $(P', Q') \in S^{(\widetilde{L}^+)}$ where $\widetilde{L} = C\widetilde{M}\widetilde{N}\widetilde{M_\gamma}$.

Collecting these transitions of Q, we can form a set $T = \{Q \xmapsto{N_i,\delta_i} Q'_i : \gamma\sigma_i = \delta_i\sigma_i, \sigma_i$ agrees with $C\widetilde{M}\widetilde{N_i}\}$. Since for any σ such that $\sigma \models C\widetilde{M}$ there exists an element i of T such that $\sigma \models N_i$, we know $C\widetilde{M} \Rightarrow \bigvee \widetilde{N_i}$. Also for all i, $(P', Q'_i) \in S^{(\widetilde{L}^+)}$ for $\widetilde{L}$ as above.

Thus $\mathcal{S}$ is a symbolic hyperbisimulation.

$\Leftarrow$: We show that $\mathcal{R} = \{(P\sigma, Q\sigma) : (P, Q) \in S^C \in \mathcal{S}$ for all $\sigma \models C\}$ is a hyperbisimulation, where $\mathcal{S}$ is a symbolic hyperbisimulation.

For $(P\sigma, Q\sigma) \in \mathcal{R}$, if $P\sigma \xrightarrow{\gamma} P'$, then by Lemma 17 $P \xmapsto{\widetilde{M},\gamma'} P''$ s.t. $\sigma \models \widetilde{M}$, $\gamma = \gamma'\sigma$ and $P' \equiv P''\sigma$. Then for some $T \subseteq \{Q \xmapsto{\widetilde{N_i},\delta_i} Q'_i : \gamma'\sigma_i = \delta_i\sigma_i, \sigma_i$ agrees with $C\widetilde{M}\widetilde{N_i}\}$ such that $\forall i \in T : (P'', Q'_i) \in S^{(\widetilde{L}^+)}$ for $\widetilde{L} = C\widetilde{M}\widetilde{N_i}\widetilde{M_{\gamma'}}$ and $C\widetilde{M} \Rightarrow \bigvee \widetilde{N_i}$. So for any $\sigma \models C\widetilde{M}$ there is an element i in T s.t. $\sigma \models C\widetilde{M}\widetilde{N_i}$, and thus by Lemma 17 $Q\sigma \xrightarrow{\delta_i\sigma} Q'_i\sigma$, $\delta_i\sigma = \gamma'\sigma = \gamma$ and $(P'\sigma_\gamma, Q'_i\sigma\sigma_\gamma) \in \mathcal{R}$, where σ_γ a substitutive effect of γ. $\square$

Lemma 20 *For all φ and ψ there is an injective substitution ρ such that $\sigma_{\varphi\sigma_\psi}\rho = \sigma_\varphi\sigma_\psi$.*

Lemma 21 *For all φ and ψ there is an injective substitution ρ such that $\sigma_\varphi\sigma_\psi\rho = \sigma_{(\varphi\wedge\psi)}$.*

Lemma 22 $\sigma_\varphi\sigma_{\varphi\wedge\psi} = \sigma_{\varphi\wedge\psi}$.

Lemma 23 *$P \equiv Q$ iff $P\rho \equiv Q\rho$ for any injective substitution ρ.*

We can now relate the symbolic and non-symbolic weak transitions:

Lemma 24

1. *If $\sigma \models \widetilde{M}$ and $P \overset{\widetilde{M},\gamma}{\Longmapsto} P'$, then $P\sigma \overset{\gamma\sigma}{\Longrightarrow} P''$ s.t. $P''\sigma_{\gamma\sigma} \equiv P'\sigma\sigma_{\gamma\sigma}$.*

2. *If $P\sigma \overset{\gamma}{\Longrightarrow} P''$ then there are $\widetilde{M}, \gamma', P'$ such that $P \overset{\widetilde{M},\gamma'}{\Longmapsto} P'$ and $\sigma \models \widetilde{M}$, $\gamma = \gamma'\sigma$, and $P''\sigma_\gamma \equiv P'\sigma\sigma_{\gamma'\sigma}$.*

PROOF: By induction on the number of strong transitions in the weak transitions.

1. The base cases are simple, using Lemma 16, as are the induction cases where γ is a communication action. The interesting case is where γ is a fusion action:

   Then $\gamma = \varphi_1 \wedge \varphi_2$ and $\widetilde{M} \Leftrightarrow \widetilde{M}_1(\widetilde{M'}\sigma_{\varphi_1})$ where σ_{φ_1} is a substitutive effect of φ_1, and $P \xmapsto{\widetilde{M}_1,\varphi_1} P'_1$, $P'_1 \mathrel{|}\!\!\xRightarrow{\widetilde{M'},\varphi'} P'$.

   By Lemma 16 $P\sigma \xrightarrow{\varphi_1\sigma} P'_1\sigma$, and by induction $P'_1\sigma' \xRightarrow{\varphi'\sigma'} P''$ for any $\sigma' \models \widetilde{M'}$, s.t. $P''\sigma_{\varphi'\sigma'} \equiv P'\sigma'\sigma_{\varphi'\sigma'}$. So if $\sigma\sigma_{\varphi_1\sigma} \models \widetilde{M'}$, we know that $P'_1\sigma\sigma_{\varphi_1\sigma} \xRightarrow{\varphi'\sigma\sigma_{\varphi_1\sigma}} P''$ such that $P''\sigma_{\varphi'\sigma\sigma_{\varphi_1\sigma}} \equiv P'\sigma(\sigma_{\varphi_1\sigma})(\sigma_{\varphi'\sigma(\sigma_{\varphi_1\sigma})})$. By Lemma 18 $\sigma\sigma_{\varphi_1\sigma} \models \widetilde{M'}$ iff $\sigma\sigma_{\varphi_1} \models \widetilde{M'}$ iff $\sigma \models \widetilde{M'}\sigma_{\varphi_1}$, and since $\sigma \models \widetilde{M}$ and $\widetilde{M} \Leftrightarrow \widetilde{M}_1(\widetilde{M'}\sigma_{\varphi_1})$ this holds.

   We need to show that $P''\sigma_{\varphi\sigma} \equiv P'\sigma\sigma_{\varphi\sigma}$, i.e., $P''\sigma_{(\varphi_1\wedge\varphi')\sigma} \equiv P'\sigma\sigma_{(\varphi_1\wedge\varphi')\sigma}$. We know $P''\sigma_{\varphi'\sigma(\sigma_{\varphi_1\sigma})} \equiv P'\sigma(\sigma_{\varphi_1\sigma})(\sigma_{\varphi'\sigma(\sigma_{\varphi_1\sigma})})$, and by Lemmas 20 and 21, $P''\sigma_{\varphi'\sigma(\sigma_{\varphi_1\sigma})} \equiv P''\sigma_{(\varphi_1\wedge\varphi')\sigma}\rho$ for an injective substitution ρ. Again by Lemmas 20 and 21, $P'\sigma(\sigma_{\varphi_1\sigma})(\sigma_{\varphi'\sigma(\sigma_{\varphi_1\sigma})}) \equiv P'\sigma(\sigma_{\varphi_1\sigma})(\sigma_{(\varphi_1\wedge\varphi')\sigma})\rho$ for the *same* injective substitution ρ, and by Lemma 22 $P'\sigma(\sigma_{\varphi_1\sigma})(\sigma_{(\varphi_1\wedge\varphi')\sigma})\rho \equiv P'\sigma(\sigma_{(\varphi_1\wedge\varphi')\sigma})\rho$. We now have $P''\sigma_{(\varphi_1\wedge\varphi')\sigma}\rho \equiv P'\sigma(\sigma_{(\varphi_1\wedge\varphi')\sigma})\rho$, and by Lemma 23, $P''\sigma_{(\varphi_1\wedge\varphi')\sigma} \equiv P'\sigma(\sigma_{(\varphi_1\wedge\varphi')\sigma})$ as required.

2. Similar. □

Definition 25 *The set $\mathcal{S} = \{S^C\}_\mathcal{C}$ (for $\mathcal{C}$ a set of sequences of match operators) of binary symmetric process relations is a* weak symbolic hyperbisimulation *if for each $S^C \in \mathcal{S}$, $(P,Q) \in S^C$ implies:*

If $P \xmapsto{\widetilde{M},\gamma} P'$ with $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ and $\widetilde{M}$ consistent with C, then

for some $T \subseteq \{Q \mathrel{|}\!\!\xRightarrow{\widetilde{N}_i,\delta_i} Q'_i : \gamma\sigma = \delta_i\sigma$, for all σ s.t. σ agrees with $C\widetilde{M}\widetilde{N}_i\}$ $C\widetilde{M} \Rightarrow \bigvee \widetilde{N}_i$, and for all i, $(P', Q'_i) \in S^{(L_i^+)} \in \mathcal{S}$, where $L_i = C\widetilde{M}\widetilde{N}_i\widetilde{M}_\gamma$ and $\widetilde{M}_\gamma$ agrees with the substitutive effect of γ.

P is weak symbolically C-hyperequivalent *to Q, written $P \approxeq_C Q$, if $(P,Q) \in S^C \in \mathcal{S}$ for some symbolic hyperbisimulation $\mathcal{S}$. We write $P \approxeq Q$ if $P \approxeq_\emptyset Q$.*

Theorem 26 *$P \approx Q$ iff $P \approxeq Q$.*

PROOF: Like the proof of Theorem 19, using Lemma 24 in place of Lemma 17 at appropriate places. □

4 Implementation

The main motivation for the symbolic equivalences presented in section 3 is to efficiently decide the equivalences in an automated tool, and we have implemented algorithms to do so in the Mobility Workbench (MWB). The implementation is available through the URL `http://www.docs.uu.se/~victor/mwb.html`.

The Mobility Workbench is an automated tool for analyzing systems of mobile processes, initially developed for the π-calculus by the present author and Faron Moller [VM94] based on the symbolic characterization of open bisimulation equivalences by Davide Sangiorgi [San96]. It was later extended with weak open bisimulation checking [Vic94], a sort inference algorithm [Gay93], and by Mads Dam and Lars-Henrik Eriksson with a model checker [Dam96]. Recently the model checker was reimplemented using symbolic semantics [Fra96, Bes98], with a significant speedup as a result.

The bisimulation checking algorithms in the MWB are "on-the-fly" algorithms which build the state spaces of the agents as they are explored, in contrast with the partition-refinement algorithms algorithms [PT87, KS90] usually employed for finite-state systems in, e.g., CCS. There, the state spaces of the agents are constructed independently of each other, and then their union is partitioned in successively smaller parts, terminating when a bisimulation relation is found. Because of the side conditions on bound names in bisimulation equivalences in fusion calculus or π-calculus it is not simple to generate the state spaces of agents independently. Pistore and Sangiorgi [PS96] have developed a partition refinement algorithm for checking open bisimulation for π-calculus agents with finite control. As future work, we intend to investigate their work in the setting of the fusion calculus; we expect to find simplifications since we do not have the complication of distinctions. The advantages of an on-the-fly algorithm are that it is easy to implement, and that it may return a negative answer even for non-finite-control agents. It can also be more efficient than partition refinement algorithms on agents with a limited degree of non-determinism.

The algorithm follows Definitions 13 and 25 closely. Given two agents P and Q, a match/mismatch sequence C and a relation R, check if there is a triple $(P', Q', C') \in R$ such that $C \Rightarrow C'$ and $C' \Rightarrow (P \equiv P' \wedge Q \equiv Q')$, and if so, return the relation unchanged. Otherwise, for each transition $P \xmapsto{M,\gamma} P'$, find a subset T of the (strong or weak) transitions of Q such that the conditions match appropriately and the actions are equivalent under a substitution agreeing with the conditions. First each singleton subset of the transitions of Q is tried, then incrementally larger subsets not containing transitions which have been seen not to simulate; if there is no such subset P and Q are not bisimilar. To efficiently decide the implication $CM \Rightarrow \bigvee N_i$, we first eliminate all tautological pairs ($[x = y] \vee [x \neq y]$) in $\bigvee N_i$, and then see if CM implies one of the remaining matches. If so then assume that P and Q are bisimilar under the condition C (by adding the triple (P, Q, C) to R), and recurse over the derivatives with the condition L_i. This either fails, causing the current recursion to try another subset T (or fail if there is none), or returns a relation containing (P, Q, C) which is used in subsequent recursive calls when matching the other transitions of P. Finally, when all transitions of P have been simulated, find simulating transition sets of P for each transition of Q in the same way, returning the resulting relation.

To improve efficiency in the MWB, agents are represented using De Bruijn indexing for names, making alpha-conversion unnecessary, and computed transitions are recorded in hash-tables. These techniques are described in more detail in [Vic94]. In the new version implementing the equivalences of this paper, the number of substitutions performed on agents is minimized by representing equivalence between names explicitly in the condition C of the bisimulation definition. This avoids copying of agents, and reduces space requirements significantly for many large examples.

4.1 A Simple Example: Critical Sections

As a simple example of using the MWB for fusion calculus examples, consider the well-known Critical Section problem: two processes share some data, and need to have exclusive access to the data while modifying them. This can be achieved by defining a *critical section* in the processes with the properties that if one process is in its critical section, no other process can be in its critical section, and if a process wants to enter its critical section, it cannot be stopped by processes not wanting to enter theirs.

An abstract specification in the fusion calculus of a solution to the problem for two processes is $\mathit{Spec} \stackrel{def}{=} \mathbf{1} \,.\, b_1 \,.\, \mathbf{1} \,.\, e_1 \,.\, \mathit{Spec} + \mathbf{1} \,.\, b_2 \,.\, \mathbf{1} \,.\, e_2 \,.\, \mathit{Spec}$ which states that without external control, a process may enter its critical section (b_i for "begin"), then modify the data (the internal or invisible action $\mathbf{1}$), and exit the critical section (e_i for "end"), whereupon any process may again enter its critical section.

A simple protocol for solving the critical section problem is by using a *test-and-set* instruction, which atomically tests and sets a (shared) flag variable. A type of test-and-set be specified in the fusion calculus by combining mismatch and fusion prefix. The specification of the solution would then be

```
fagent A2(flag,set,b,e) = \
        [flag#set]{flag=set}.b.1.e. \
                (g)'flag<g>.A2(g,set,b,e) \
      + (g)flag<g>.A2(g,set,b,e)

fagent Sys2(b1,b2,e1,e2) = \
        (flag,set)(A2(flag,set,b1,e1) | A2(flag,set,b2,e2))

fagent Spec2(b1,b2,e1,e2) = \
        1.b1.e1.Spec2(b1,b2,e1,e2) + 1.b2.e2.Spec2(b1,b2,e1,e2)
```

Figure 1: The MWB code for the critical section problem.

$$\begin{aligned} P_i(f) \stackrel{def}{=} & \quad [f \neq s]\{f = s\} \,.\, b_i \,.\, \mathbf{1} \,.\, e_i \,.\, (f')\overline{f}\, f' \,.\, P_i(f') \\ & +(f')f\, f' \,.\, P_i(f') \end{aligned}$$

A process P_i can atomically test if the flag is not set ($[f \neq s]$) and if so set it ($\{f = s\}$). Then it enters its critical section (b_i), modifies the data ($\mathbf{1}$) and exits the critical section (e_i). It must then reset the flag: this is done by creating a new flag and sending it to the other process, itself continuing with the new flag. A process which has not entered its critical section can receive a new flag from another process (which has been in its critical section and thus "used up" the old flag). It must then retry entering its critical section using the new flag.

We expect the implementation to be equivalent to the specification, stated as $Spec \approx (f)(P_1(f) \mid P_2(f))$.

In Figure 1 the MWB code to specify the specification and system of processes can be seen. The syntax of a prefix $a\widetilde{x} \,.\, P$ is a`<x1,...,xn>.P`, where a is written simply `a` for an input or `'a` for an output. The command `fagent` defines a fusion agent; note that the definition must be closed — in our formal definitions we sometimes let names be implicit in the argument lists (as in $Spec$ or P_i above), but the MWB must be explicitly told about all names in argument lists. It will complain if a definition is not closed.

The code in Figure 1 can be entered interactively or into a file, say `cs.mwb`, and then read by MWB using the `input` command. Once the agents have been defined, we can ask the MWB to verify the correctness of the implementation by issuing a `fweq` command (for `f`usion `w`eak `eq`uivalence). The MWB tries to find a symbolic hyperbisimulation and if it does, it reports its size. In Figure 2 we give an example session showing how this can all be done.

```
% mwb

   The Mobility Workbench
   (Version 4.133, built Mon Apr 13 14:11:45 MET DST 1998)

MWB>input "cs.mwb"
MWB>fweq Sys2(b1,b2,e1,e2) Spec2(b1,b2,e1,e2)
The two agents are equal.
Bisimulation relation size = 10.
MWB>quit
```

Figure 2: Running the MWB.

5 Conclusion

We have presented a symbolic transition system for the fusion calculus, along with symbolic characterizations of strong and weak hyperequivalence. The characterizations were proven to coincide with their non-symbolic counterparts, and were used as a basis for an algorithm to automatically check hyperequivalence for finite-control fusion calculus agents.

The algorithms were implemented in the Mobility Workbench, and proved to significantly decrease the space requirements for some large verifications, compared to the earlier implementation. Applying the techniques of this paper to the open bisimulation equivalence of the pi-calculus, the verification of the GSM handover protocol specified in [OP92], which in the previous version of the MWB used 274 megabyte of heap space, in the new version needs only 50 megabyte, making it reasonable to use a normal workstation for verifying such examples without a large penalty for virtual memory handling.

In future work a modal logic for the fusion calculus should be developed, and implemented in the MWB using symbolic semantics. More efficient state representations should also be developed, in order to increase the size of applicable examples. An attractive line of work would be to apply the work on partition refinement algorithms for the π-calculus [MP95, PS96] on the fusion calculus. We would expect to simplify their work since we do not have the complication of distinctions.

References

[BD96] M. Boreale and R. De Nicola. A symbolic semantics for the π-calculus. *Journal of Information and Computation*, 126(1):34–52, 1996. Available as Report SI 94 RR 04, Università "La Sapienza" di Roma; an extended abstract appeared in *Proceedings of CONCUR '94*, pages 299–314, LNCS 836.

[Bes98] F. B. Beste. The model prover - a sequent-calculus based modal μ-calculus model checker tool for finite control π-calculus agents. M.Sc. thesis DoCS 98/97, Department of Computer Systems, Uppsala University, Sweden, March 1998.

[Dam96] M. Dam. Model checking mobile processes. *Journal of Information and Computation*, 129(1):35–51, 1996. Also available as Research Report R94:01, SICS, Sweden. A summary appeared in *Proceedings of CONCUR '93*, LNCS 715.

[Fra96] T. Franzén. A theorem-proving approach to deciding properties of finite-control agents (extended abstract). Technical Report T96:02, Swedish Institute of Computer Science, Kista, 1996.

[Gay93] S. J. Gay. A sort inference algorithm for the polyadic π-calculus. In *Proceedings of POPL '93*. ACM, January 1993.

[HL95] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138(2):353–389, 1995. Earlier version published as Technical Report 1/92, School of Cognitive and Computing Sciences, University of Sussex, UK.

[JP93] B. Jonsson and J. Parrow. Deciding bisimulation equivalences for a class of non-finite-state programs. *Information and Computation*, 107:272–302, 1993.

[KS90] P. Kanellakis and S. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *Journal of Information and Computation*, 86(1):43–68, May 1990.

[Lin94] H. Lin. Symbolic bisimulation and proof systems for the π-calculus. Technical Report 7/94, School of Cognitive and Computing Sciences, University of Sussex, UK, 1994.

[Lin95] H. Lin. Complete inference systems for weak bisimulation equivalences in the π-calculus. In P. D. Mosses, M. Nielsen and M. I. Schwarzbach, eds, *Proceedings of TAPSOFT '95*, volume 915 of *LNCS*, pages 187–201. Springer, 1995. Presented in the CAAP-section. Available as Technical Report ISCAS-LCS-94-11, Institute of Software, Chinese Academy of Sciences, 1994.

[Lin96] H. Lin. Symbolic transition graph with assignment. In U. Montanari and V. Sassone, eds, *Proceedings of CONCUR'96*, volume 1119 of *LNCS*, pages 50–65. Springer, 1996.

[Liu94] X. Liu. Characterizing bisimulation congruence in the π-calculus. In B. Jonsson and J. Parrow, eds, *Proceedings of CONCUR '94*, volume 836 of *LNCS*, pages 331–350. Springer, Aug. 1994. Extended version as Report 2/94, University of Sussex, Brighton.

[MP95] U. Montanari and M. Pistore. Checking bisimilarity for finitary π-calculus. In I. Lee and S. A. Smolka, eds, *Proceedings of CONCUR '95*, volume 962 of *LNCS*, pages 42–56. Springer, 1995.

[MPW92] R. Milner, J. Parrow and D. Walker. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 100:1–77, Sept. 1992.

[OP92] F. Orava and J. Parrow. An algebraic verification of a mobile network. *Journal of Formal Aspects of Computing*, 4:497–543, 1992.

[PS96] M. Pistore and D. Sangiorgi. A partition refinement algorithm for the π-calculus. In R. Alur, ed, *Proceedings of CAV '96*, volume 1102 of *LNCS*, pages 38–49, 1996. Full version to appear in *Journal of Information and Computation*.

[PT87] R. Paige and R. Tarjan. Three partition refinement algorithms. *SIAM Journal of Computing*, 16(6):973–989, 1987.

[PV98] J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS '98*, 1998.

[San96] D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996. Earlier version published as Report ECS-LFCS-93-270, University of Edinburgh. An extended abstract appeared in the *Proceedings of CONCUR '93*, LNCS 715.

[Vic94] B. Victor. *A Verification Tool for the Polyadic π-Calculus*. Licentiate thesis, Department of Computer Systems, Uppsala University, Sweden, May 1994. Available as report DoCS 94/50.

[VM94] B. Victor and F. Moller. The Mobility Workbench — a tool for the π-calculus. In D. Dill, ed, *Proceedings of CAV '94*, volume 818 of *LNCS*, pages 428–440. Springer, 1994.

[VP98] B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In K. G. Larsen, S. Skyum and G. Winskel, eds, *Proceedings of ICALP '98*, volume 1443 of *LNCS*, pages 455–469. Springer, July 1998.