Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 194 (2008) 85–99
www.elsevier.com/locate/entcs

Matching Systems for Concurrent Calculi

Bjørn Haagensen[1]

Aalborg University
Denmark

Sergio Maffeis[2]

Imperial College London

Iain Phillips[3]

Imperial College London

Abstract

Matching systems were introduced by Carbone and Maffeis, and used to investigate the expressiveness of the pi-calculus with polyadic synchronisation. We adapt their definition and investigate matching systems for CCS, the pi-calculus and Mobile Ambients. We show among other results that the asynchronous pi-calculus with matching cannot be encoded (under certain conditions) in CCS with polyadic synchronisation of all finite levels.

Keywords: Matching systems, CCS, pi-calculus, Mobile Ambients

1 Introduction

Matching systems were introduced by Carbone and Maffeis [4]. A matching system is a protocol which ensures that a client matches successfully with a server if and only if both parties have the same sequence of names as parameters. This can be achieved trivially if client and server can synchronise on all names in a single atomic communication. However it may not be possible if, for instance, they can only synchronise on one name at a time, as in standard π-calculus. Carbone and Maffeis used matching systems to establish a hierarchy within ${}^{e}\pi$, the π-calculus with polyadic synchronisation. They show that there is no encoding (satisfying

[1] Email: bh@cs.aau.dk
[2] Email: maffeis@doc.ic.ac.uk
[3] Email: iccp@doc.ic.ac.uk

doi:10.1016/j.entcs.2007.11.004

certain conditions) of the asynchronous calculus with n-adic communication into the synchronous calculus with m-adic communication (for any $m < n$).

In this paper we investigate matching systems further. In particular, we propose a weakened form of matching system, where if client and server agree on their parameters then there is a successful computation, but success is no longer guaranteed, unlike in the original formulation. These weak matching systems enable us to obtain different separations between calculi. We regard matching systems (whether in the weak form or the original strong form) as measuring the capability of particular calculi to perform transactions, in other words to perform a series of operations which can be treated as a single operation. Weak matching systems require only that all the commits are justified, whereas strong matching systems require also that the transaction is not rolled back an unbounded number of times.

Matching systems may be compared with De Nicola-Hennessy testing [6], where processes interact with test processes, and one analyses whether they *can* pass a test ("may" testing), or are *guaranteed* to pass a test ("must" testing). A weak matching system can be seen as a kind of may testing scenario where clients and servers test each other. Similarly, a strong matching system corresponds, in a sense, to a must testing scenario.

In [4], it is shown that there is no "sensible" encoding of matching in the π-calculus with mixed choice. Here we use weak matching systems to give a different separation result involving a language with matching, based on a different notion of encoding. We shall show there there is no encoding (subject to certain conditions) from the asynchronous π-calculus with matching into CCS with n-adic communication (for any n).

This is related to the question of showing that the asynchronous π-calculus cannot be encoded into CCS. As far as we are aware, such a negative result has never been obtained, even though most researchers would presumably expect this to hold, since the asynchronous π-calculus has the ability to send and receive names (objects) and then use them as channels (subjects), and this is disallowed in CCS (even with value passing).

Palamidessi [10] used electoral systems to prove two relevant results. Firstly, she showed that CCS cannot be encoded in the asynchronous π-calculus (the converse of what we are discussing). Secondly, she showed that the π-calculus with mixed choice cannot be encoded in CCS. However her work leaves open the possibility that the asynchronous π-calculus can be encoded in CCS.

Banach and van Breugel [1] encoded the π-calculus into a version of CCS. This involves augmenting CCS with infinite operations (and not just infinite summation).

Sangiorgi [13] defined the π-calculus with internal mobility (πI), where only private names can be transmitted. He gave a hierarchy of typed calculi within πI, such that the bottom level represents "the core of CCS". He showed that higher levels in the hierarchy exhibit a "higher degree" of mobility, in the sense that they admit longer subject-object dependency chains. However he did not assert any result about the non-encodability of higher levels in lower levels of the hierarchy.

Boreale [2] gave an encoding of the asynchronous π-calculus into πI. This en-

coding is in two steps, and goes via an intermediate calculus, *localised* π, or Lπ, the subset of the asynchronous π-calculus where the recipient of a name may only use it in output actions. This terminology is due to Merro and Sangiorgi [8]. They showed that Lπ can be encoded fully abstractly in localised πI using the second step in Boreale's encoding.

After presenting our results on weak matching systems, we recast the separation result concerning polyadic synchronisation of [4] into our current setting, using the notion of replicated strong matching systems. Our new formulation is a slight generalisation of the previous result. Surprisingly, and against previous intuition, we have found that by simply requiring each instance of a matching system to be finite (strong matching systems), the full π-calculus is powerful enough to solve the problem for any degree n. We conjecture that the same is not possible for the asynchronous π-calculus, suggesting a possible new interpretation of the expressive power of the mixed choice construct.

The remainder of the paper is organised as follows. In Section 2 we define the calculi we shall be considering. Then in Sections 3 and 4 we investigate weak and strong matching systems, respectively. We finish with conclusions and further work.

2 Calculi

In this section we define the calculi that we shall be concerned with in this paper.

We let $x, y, \ldots$ range over the set of *names* $\mathcal{N}$. We shall let $\overrightarrow{x}$ denote a tuple of names $x_1, \ldots, x_n$.

Polyadic synchronisation, where e.g. an output process $\overline{x \cdot y}\langle z\rangle.P$ can synchronise with an input process $x \cdot y(w).Q$, was introduced in [4].

Definition 2.1 *The full π_n-calculus (fπ_n) is defined as the polyadic synchronous π-calculus with mixed choice, matching and mismatching, and polyadic synchronisation of degree n, that is*

$$P ::= P \mid Q \;\mid\; \nu x P \;\mid !P \;\mid\; \Sigma_i \alpha_i.P \;\mid\; [x = y]P \;\mid\; [x \neq y]P$$

where each α_i is of the form $x_1 \cdot \ldots \cdot x_n(\overrightarrow{y})$ or $\overline{x_1 \cdot \ldots \cdot x_n}\langle \overrightarrow{y} \rangle$. We let S, T range over summations, and write the empty summation as $\mathbf{0}$.

Note that fπ_1 is the standard full π-calculus. We define the free names $\mathsf{fn}(P)$ as usual, with input and restriction being name-binding.

Definition 2.2 Structural congruence *is the least congruence $\equiv$ on fπ_n processes satisfying the following laws: $P \mid Q \equiv Q \mid P$, $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$, $\mathbf{0} \mid P \equiv P$, $[x = x]P \equiv P$, $[x \neq y]P \equiv P$ if $x \neq y$, $!P \equiv P \mid !P$ and $\nu x(P \mid Q) \equiv P \mid \nu x Q$ if $x \notin \mathsf{fn}(P)$, together with reordering of summations.*

Definition 2.3 *The reduction relation on fπ_n is defined by the following axiom and*

rules

$$(\overline{x_1 \cdot \ldots \cdot x_n}\langle y_1, \ldots, y_m\rangle.P + S) \mid (x_1 \cdot \ldots \cdot x_n(z_1, \ldots, z_m).Q + T)$$
$$\rightarrow P \mid Q\{y_1, \ldots, y_m/z_1, \ldots, z_m\}$$

$$\frac{P \rightarrow Q}{P \mid R \rightarrow Q \mid R} \qquad \frac{P \rightarrow Q}{\nu x P \rightarrow \nu x Q} \qquad \frac{P' \equiv P \quad P \rightarrow Q \quad Q \equiv Q'}{P' \rightarrow Q'}$$

We let $\Rightarrow$ be the reflexive and transitive closure of $\rightarrow$.

Definition 2.4 *Input and output* barbs *are defined by*

$$P \downarrow_{x_1 \cdot \ldots \cdot x_n} \text{ iff } P \equiv \nu \overrightarrow{z}((x_1 \cdot \ldots \cdot x_n(\overrightarrow{y}).R + S) \mid Q) \text{ where } \overrightarrow{x} \cap \overrightarrow{z} = \emptyset$$
$$P \downarrow_{\overline{x_1 \cdot \ldots \cdot x_n}} \text{ iff } P \equiv \nu \overrightarrow{z}((\overline{x_1 \cdot \ldots \cdot x_n}\langle \overrightarrow{y}\rangle.R + S) \mid Q) \text{ where } \overrightarrow{x} \cap \overrightarrow{z} = \emptyset$$

We let $P \Downarrow_{x_1 \cdot \ldots \cdot x_n}$ *iff* $P \Rightarrow \downarrow_{x_1 \cdot \ldots \cdot x_n}$, *and similarly for output barbs.*

Definition 2.5 *The calculus* $\mathrm{a}\pi_n$ *is defined as the polyadic asynchronous* π*-calculus with polyadic synchronisation of degree* n*, that is*

$$P ::= \mathbf{0} \mid P \mid Q \mid \nu x P \mid !P \mid x_1 \cdot \ldots \cdot x_n(\overrightarrow{y}).P \mid \overline{x_1 \cdot \ldots \cdot x_n}\langle \overrightarrow{y}\rangle$$

The only reduction axiom for $\mathrm{a}\pi_n$ *is*

$$x_1 \cdot \ldots \cdot x_n(y_1, \ldots, y_m).P \mid \overline{x_1 \cdot \ldots \cdot x_n}\langle z_1, \ldots, z_m\rangle \rightarrow P\{z_1, \ldots, z_m/y_1, \ldots, y_m\}$$

Note that $\mathrm{a}\pi_1$ is the standard asynchronous π-calculus. The *localised* π-calculus Lπ [2,8] is the subset of $\mathrm{a}\pi_1$ where the recipient of a name may only use it in output actions. We write $\mathrm{a}\pi_n^=$ to denote $\mathrm{a}\pi_n$ with matching $[x = y]P$.

Definition 2.6 *The calculus* CCS_n *is defined as the fragment of* $\mathrm{f}\pi_n$ *which has no name-passing and no matching or mismatching, that is*

$$P ::= P \mid Q \mid \nu x P \mid !P \mid \Sigma_i \alpha_i.P$$

where each α_i *is of the form* $x_1 \cdot \ldots \cdot x_n$ *or* $\overline{x_1 \cdot \ldots \cdot x_n}$.

The CCS_n synchronisation rule is

$$(\overline{x_1 \cdot \ldots \cdot x_n}.P + S) \mid (x_1 \cdot \ldots \cdot x_n.Q + T) \rightarrow P \mid Q\ .$$

Note that CCS_1 is a form of standard CCS. It resembles the CCS of [9] with replication instead of recursion.

Definition 2.7 *The calculus of* Mobile Ambients *(MA) [5] has the following syntax:*

$$P ::= \mathbf{0} \mid P \mid Q \mid \nu x P \mid !P \mid x[\,P\,] \mid \mathsf{in}\, x.P \mid \mathsf{out}\, x.P \mid \mathsf{open} x.P$$
$$\mid \langle x\rangle \mid (x).P$$

Here $x[\,P\,]$ is an ambient named x enclosing P, and in, out, open are the *capabilities* for entering, leaving or dissolving ambients. We also have asynchronous, anonymous (no channel) output and input.[4] The free names $\mathsf{fn}(P)$ of a process P are defined much as for the π-calculus, with input and restriction being name-binding.

[4] Note that for simplicity we have just defined name-passing communication, whereas communication in [5] allows sequences of capabilities to be transmitted.

Structural congruence and reduction rules are adapted from the π-calculus, with the following reduction axioms:

$$x[\,\mathsf{in}\,y.P \mid Q\,] \mid y[\,R\,] \to y[\,x[\,P \mid Q\,] \mid R\,]$$

$$y[\,x[\,\mathsf{out}\,y.P \mid Q\,] \mid R\,] \to x[\,P \mid Q\,] \mid y[\,R\,]$$

$$\mathsf{open}x.P \mid x[\,Q\,] \to P \mid Q$$

$$\langle x\rangle \mid (y).P \to P\{x/y\}$$

Barbs are defined by

$$P \downarrow_x \text{ iff } P \equiv \nu\overrightarrow{z}(x[\,P\,] \mid Q) \text{ where } x \notin \overrightarrow{z}\ .$$

Pure public boxed MA (ppbMA) is got by omitting communication, restriction and the open capability. Recall that the open capability is omitted in the calculus of boxed ambients [3].

Let $\mathrm{MA}^{-\mathsf{in}}$ denote (full) MA with only the in capability omitted.

3 Weak Matching Systems

We present the weakened definition of matching system. Then we show that $\mathrm{a}\pi_2$ has matching systems of every finite degree. We show that CCS_n does not have matching systems of degree $n+1$ or greater. We then show that matching systems are preserved by encodings satisfying certain properties. We deduce that there is no encoding from $\mathrm{a}\pi_2$ to CCS satisfying those properties. We also present analogous results concerning ppbMA and CCS.

In matching systems [4], the idea is that clients C communicate with servers S and try to match their parameters, reporting success if there is a match. We change Carbone and Maffeis's definition of matching system to the following, which applies to all the calculi defined in Section 2:

Definition 3.1 *A* weak matching system (WMS) of degree n *is a tuple* $(C, S, x_1, \ldots, x_n)$ *where* C *and* S *are processes and* $x_1, \ldots, x_n$ *are distinct names, such that for all finite index sets* I *and* J*, and all injective substitutions* σ_i $(i \in I)$ *and* θ_j $(j \in J)$ *where* $\mathrm{dom}(\sigma_i) = \mathrm{dom}(\theta_j) = \{x_1, \ldots, x_n\}$*,*

$$(\prod_{i\in I} C\sigma_i \mid \prod_{j\in J} S\theta_j) \Downarrow_\omega \text{ iff } \exists i \in I, j \in J \text{ such that } \sigma_i = \theta_j\ .$$

Here ω *is a special name used for reporting a successful match. We require that* $\omega \notin \{x_1, \ldots, x_n\}$ *and that substitutions do not change any* x_i *into* ω*. Also substitutions should not map any* x_i *into a free name of* C *or* S*, other than* $x_1, \ldots, x_n$*. When convenient, we display parameters explicitly, writing* $C\sigma$ *as* $C\langle\sigma(x_1), \ldots, \sigma(x_n)\rangle$*.*

It is easy to see that, in a WMS $(C, S, \vec{x})$, all of $\vec{x}$ must be free in both C and S.

There are five changes from the pre-existing notion. Firstly, and most importantly, we do not require that if there is a match then *every* computation leads to success. Thus we have a "may" notion of success, rather than a "must" notion. Secondly, we do not use replication in the definition (for the server). Thirdly, we

omit the identifier for the client, so that client and server are symmetrical. Fourthly, we allow both client and server to contain free names not drawn from $\vec{x}$. Fifthly, we require that parameters are distinct, so that we are dealing with permutations rather than substitutions in general. This last condition will be useful when we show that matching systems are preserved by encodings (Theorem 3.9).

Note that in standard process calculi a weak matching system never needs to use recursion or replication. It must be the case that $(C\langle\vec{x}\rangle \mid S\langle\vec{x}\rangle) \Downarrow_\omega$ by a finite computation. We can unfold recursion or replication enough to get this computation, and then set the recursion or replication part to the nil process $\mathbf{0}$. The modified client and server still give an ω barb when there is a match, and, since we have only reduced behaviour and not added any new behaviour, they still do not yield an ω barb when there is no match.

If a calculus has a WMS of degree n then it has WMSs of all smaller degrees:

Lemma 3.2 *Let $m < n$. If $(C, S, x_1, \ldots, x_n)$ is a WMS of degree n then $(C, S, x_1, \ldots, x_m)$ is a WMS of degree m.* □

We now show that $\mathrm{a}\pi_1^=$ has weak matching systems of every degree:

Theorem 3.3 *For every $n \geq 1$, $\mathrm{a}\pi_1^=$ has a WMS of degree n.*

Proof. (Sketch) We define C_n and S_n as follows:

$$C_n(x_1, \ldots, x_n) \stackrel{\text{df}}{=} x_1(z').(\textstyle\prod_{i=2}^{n} \overline{z'}\langle x_i\rangle)$$

$$S_n(x_1, \ldots, x_n) \stackrel{\text{df}}{=} \nu z(\overline{x_1}\langle z\rangle \mid S'_{n-1}\langle z, x_2, \ldots, x_n\rangle)$$

$$S'_{n-k}(z, x_{k+1}, \ldots, x_n) \stackrel{\text{df}}{=} z(x'_{k+1}).([x_{k+1} = x'_{k+1}]S'_{n-k-1}\langle z, x_{k+2}, \ldots, x_n\rangle)$$

$$\text{for } k = 1, \ldots, n-2$$

$$S'_1(z, x_n) \stackrel{\text{df}}{=} z(x'_n).([x_n = x'_n]\overline{\omega})$$

The server creates a new private name z, which is passed to the client on the first communication on x_1. The client then uses this private channel to send the other names back to the server. As each name is received, the server checks that it matches. Notice that the computation can fail even if conducted entirely between a matching client and server, due to the nondeterminism in the order in which the messages from the client are sent. This does not cause a problem, since we are dealing with weak matching systems—a single successful computation is enough. □

Remark 3.4 *We recall from [4] that matching can be encoded in $\mathrm{a}\pi_2$; the process $[x = y]P$ may be encoded as $\nu z(\overline{z \cdot x} \mid z \cdot y.P)$ where z is fresh. Hence, Theorem 3.3 also holds for $\mathrm{a}\pi_2$. Note also that, in fact, the solution is written in $\mathrm{L}\pi$ with matching.*

If we try to eliminate the use of matching in the proof of Theorem 3.3 by using communication in an obvious manner, then the proof fails. Consider the case for $n = 2$. The client process is

$$C_2(x_1, x_2) \stackrel{\text{df}}{=} x_1(z').\overline{z'}\langle x_2\rangle$$

as before. The revised server process with matching replaced by communication is

$$S_2(x_1, x_2) \stackrel{\text{df}}{=} \nu z(\overline{x_1}\langle z\rangle \mid z(x_2').(\overline{x_2'} \mid x_2.\overline{\omega}))$$

Now consider

$$C_2(a_1, a_2) \mid C_2(a_1', a_2') \mid S_2(a_1, a_2') \mid S_2(a_1', a_2)$$

where a_1, a_2, a_1', a_2' are all distinct. This network does not contain a match. However there is a computation which succeeds erroneously. Suppose that $S_2(a_1, a_2')$ receives a_2 from $C_2(a_1, a_2)$. It should then check a_2 against a_2', which should fail. Suppose also that $S_2(a_1', a_2)$ receives a_2' from $C_2(a_1', a_2')$. It should then check a_2' against a_2, which should again fail. But we can get a crossover between the two checks, so that they both succeed erroneously.

We can also define matching systems using ambients:

Theorem 3.5 *For every $n \geq 1$, ppbMA has a WMS of degree n.*

Proof. Let

$$C_n \stackrel{\text{df}}{=} m[\,\mathsf{in}\, x_1.\cdots.\mathsf{in}\, x_n.\mathsf{out}\, x_n.\ldots.\mathsf{out}\, x_1.\omega[\,\mathsf{out}\, m\,]\,]$$
$$S_n \stackrel{\text{df}}{=} x_1[\,x_2[\ldots x_n[\,]\ldots]\,]\ .$$

The idea is that the client enters successively the stacked $x_1, \ldots, x_n$ ambients of the server, before returning to the top level to report success. The client simply gets stuck if there is no match. □

We next investigate matching systems for CCS.

Theorem 3.6 *Let $m, n \geq 1$. Then CCS_n has a WMS of degree m if and only if $n \geq m$.*

Proof. (Sketch) First suppose that $n \geq m$. We define a WMS of degree m in CCS_n as follows:

$$C_m \stackrel{\text{df}}{=} \overline{x_1 \cdot \ldots \cdot x_m} \qquad S_m \stackrel{\text{df}}{=} x_1 \cdot \ldots \cdot x_m.\overline{\omega}$$

Notice that this WMS is guaranteed to succeed, and so it is in fact a *strong* MS, to be defined in Section 4.

For the converse direction, by Lemma 3.2 it is enough to show that CCS_n does not have a WMS of degree $n+1$. So suppose for a contradiction that $(C, S, x_1, \ldots, x_{n+1})$ is a WMS of degree $n+1$ in CCS_n. We shall show that there is a combination of clients and servers which does not contain a match, and yet erroneously returns success.

There is $k \geq 0$ and there are C_i, S_i $(0 \leq i \leq k)$ such that

$$C \mid S = C_0 \mid S_0 \to \cdots \to C_k \mid S_k \quad \text{where } (C_k \mid S_k)\downarrow_\omega \ .$$

This holds because there is a match between the single client and and the single server (using the identity substitution in both cases). Note that during the computation we may have to extrude the scope of restrictions in order to obtain the necessary redex, but we can then immediately return the scopes so that they lie

entirely within C_{i+1} or S_{i+1}. This returning of scopes would not in general be possible in the π-calculus, where restricted names can be transmitted along channels, resulting in more than one process sharing the same restricted name.

Let $x'_1, \ldots, x'_{n+1}$ be distinct fresh names different from $x_1, \ldots, x_{n+1}$. Let $s = s_1 \cdots s_{n+1}$ range over binary strings in $\{0,1\}^{n+1}$. Let σ_s be the substitution which sets

$$\sigma_s(x_i) \stackrel{\text{df}}{=} \begin{cases} x_i & \text{if } s_i = 0 \\ x'_i & \text{if } s_i = 1 \end{cases}$$

Let $E = \{s \in \{0,1\}^{n+1} : s \text{ has an even number of 1s}\}$ and $O = \{s \in \{0,1\}^{n+1} : s \text{ has an odd number of 1s}\}$. For $i = 0, \ldots, k$ let

$$P_i \stackrel{\text{df}}{=} \prod_{s \in E} C_i\sigma_s \mid \prod_{s \in O} S_i\sigma_s$$

Then P_0 does not contain a match. We show that $P_i \rightarrow^{2^n} P_{i+1}$ for $i = 0, \ldots, k-1$. Since plainly $P_k \downarrow_\omega$, we shall have a contradiction.

There are various cases

(i) Suppose that $C_i \rightarrow C_{i+1}$ with $S_{i+1} = S_i$. Then for each $s \in E$ we have $C_i\sigma_s \rightarrow C_{i+1}\sigma_s$, and for each $s \in O$ we have $S_{i+1}\sigma_s = S_i\sigma_s$.

(ii) The case where $S_i \rightarrow S_{i+1}$ with $C_{i+1} = C_i$ is handled like the preceding case.

(iii) Suppose that $C_i \mid S_i \rightarrow C_{i+1} \mid S_{i+1}$ by a communication on channel $y_1 \cdot \ldots \cdot y_n$. Let j be such that $x_j \notin \overrightarrow{y}$. Let $s \in E$. Let t be the same as s except that $t_j = 1 - s_j$. Then $t \in O$. Also, $C_i\sigma_s$ and $S_i\sigma_t$ can communicate on channel $\sigma_s(y_1) \cdot \ldots \cdot \sigma_s(y_n)$ to produce $C_{i+1}\sigma_s \mid S_{i+1}\sigma_t$. In this way we pair off all clients and servers and we produce P_{i+1} after 2^n reductions.

□

The next result suggests that the in capability is needed to obtain WMSs for MA.

Theorem 3.7 *For $n \geq 2$, there is no WMS of degree n in pure $MA^{-\mathsf{in}}$.*

Proof. (Sketch) We adapt the method used in the proof of Theorem 3.6. We suppose that we have a WMS of degree 2, and show for a contradiction that

$$C\langle x_1, x_2\rangle \mid C\langle x'_1, x'_2\rangle \mid S\langle x'_1, x_2\rangle \mid S\langle x_1, x'_2\rangle$$

has a successful computation. This is possible because the clients and servers can only interact at the top level of the ambient tree, due to the absence of the in capability. □

Conjecture 3.8 *For $n \geq 2$ there is no WMS of degree n in pure MA without the* out *capability.* □

We now establish conditions under which matching systems are preserved when encoding one language in another. Our result (Theorem 3.9) will apply to all the languages defined in Section 2.

We assume that we are dealing with process calculi L with a notion of weak barb $P \Downarrow_x$ such that for any permutation σ, $P \Downarrow_\omega$ iff $\sigma(P) \Downarrow_{\sigma(\omega)}$. This holds for any

process calculus in the π-calculus family (including ambient calculi). We shall also assume that all calculi have the same set of names $\mathcal{N}$, and that $\mathcal{N}$ is infinite.

The next theorem shows that weak matching systems are preserved by encodings satisfying certain conditions. The first two conditions are similar to those used by Palamidessi [10]. In the third condition, the injection φ and its properties are similar to Gorla's "strict renaming policy" [7]. The idea is that names of the source language are mapped across to unique names in the target language by φ, with the names which are not in the range of φ being available as "reserved names" for use in the encoding (so φ could be the identity if the encoding required no reserved names). The encoding of a process P should not depend on the particular names in P, since names have no structure or meaning. This idea is expressed by requiring a property of invariance under injective substitution, mediated by φ.

Theorem 3.9 *Let L and L' be process calculi. Let $[\![-]\!] : L \to L'$ be an encoding satisfying:*

(i) *$P \Downarrow_\omega$ iff $[\![P]\!] \Downarrow_\omega$;*

(ii) *$[\![P \mid Q]\!] = [\![P]\!] \mid [\![Q]\!]$;*

(iii) *There is an injective $\varphi : \mathcal{N} \to \mathcal{N}$ with $\varphi(\omega) = \omega$, such that for all finite injective substitutions σ, if P is such that $\mathrm{rge}(\sigma) \cap \mathsf{fn}(P) = \emptyset$ then we have $[\![P\sigma]\!] = [\![P]\!]\sigma'$, where the injective substitution σ' is defined by*

$$\sigma'(\varphi(x)) = \varphi(\sigma(x)) \qquad \text{if } x \in \mathrm{dom}(\sigma)$$

$$\sigma'(x) \text{ undefined} \qquad \text{otherwise}$$

Let $(C, S, \vec{x})$ be a weak matching system of degree n in L. Then $([\![C]\!], [\![S]\!], \overrightarrow{\varphi(x)})$ is a weak matching system of degree n in L'.

Proof. (Sketch) Consider

$$P \stackrel{\mathrm{df}}{=} \prod_{i \in I} [\![C]\!]\sigma_i \mid \prod_{j \in J} [\![S]\!]\theta_j$$

where $\mathrm{dom}(\sigma_i) = \mathrm{dom}(\theta_j) = \{\varphi(x_1), \ldots, \varphi(x_n)\}$. We need to show that $P \Downarrow_\omega$ iff P has a match, i.e. there are $i \in I$ and $j \in J$ such that $\sigma_i = \theta_j$.

Let $A = \bigcup_{i \in I} \mathrm{rge}(\sigma_i) \cup \bigcup_{j \in J} \mathrm{rge}(\theta_j)$. Let B be a set of names in bijection with A via $f : A \to B$, such that for each $x \in B$, both x and $\varphi(x)$ are fresh. This is always possible, since we assume that $\mathcal{N}$ is infinite.

For each $i \in I$ and each $k = 1, \ldots, n$, let

$$\sigma'_i(x_k) \stackrel{\mathrm{df}}{=} f(\sigma_i(\varphi(x_k))) \ .$$

Similarly, for each $j \in J$ and each $k = 1, \ldots, n$, let

$$\theta'_j(x_k) \stackrel{\mathrm{df}}{=} f(\theta_j(\varphi(x_k))) \ .$$

Then σ'_i, θ'_j are finite injective substitutions. Also $\sigma'_i(x_k)$, $\theta'_j(x_k) \notin \{\omega\} \cup \mathsf{fn}(P) \cup \mathsf{fn}(Q)$, since all $x \in B$ are fresh.

Now P has a match iff

$$Q \stackrel{\text{df}}{=} \prod_{i \in I} C\sigma'_i \mid \prod_{j \in J} S\theta'_j$$

has a match. This is because Q has a match iff $\exists i, j. \forall k.\, f(\sigma_i(\varphi(x_k))) = f(\theta_j(\varphi(x_k)))$ iff $\exists i, j. \forall k.\, \sigma_i(\varphi(x_k)) = \theta_j(\varphi(x_k))$ (since f is a bijection) iff P has a match.

By property (iii) of the encoding, for each $i \in I$ there is σ''_i such that $[\![C\sigma'_i]\!] = [\![C]\!]\sigma''_i$ where $\sigma''_i(\varphi(x_k)) = \varphi(\sigma'_i(x_k))$. Similarly, for each $j \in J$ there is θ''_j such that $[\![S\theta'_j]\!] = [\![S]\!]\theta''_j$ where $\theta''_j(\varphi(x_k)) = \varphi(\theta'_j(x_k))$.

Now Q has a match iff $Q \Downarrow_\omega$ (since $(C, S, \overrightarrow{x})$ is a WMS). Also, $Q \Downarrow_\omega$ iff $[\![Q]\!] \Downarrow_\omega$ (property (i) of the encoding). Using property (ii) of the encoding, we have $[\![Q]\!] = R$, where

$$R \stackrel{\text{df}}{=} \prod_{i \in I} [\![C]\!]\sigma''_i \mid \prod_{j \in J} [\![S]\!]\theta''_j$$

Notice that $\sigma''_i(\varphi(x_k)) = \varphi(\sigma'_i(x_k)) = \varphi(f(\sigma_i(\varphi(x_k))))$. Similarly, $\theta''_j(\varphi(x_k)) = \varphi(\theta'_j(x_k)) = \varphi(f(\theta_j(\varphi(x_k))))$. Also note that $\text{rge}(\sigma'') \cap ((\mathsf{fn}([\![C]\!]) \cup \mathsf{fn}([\![S]\!])) \setminus \{\varphi(x_1), \ldots, \varphi(x_n)\}) = \emptyset$, since all $y \in \varphi(B)$ are fresh. Similarly for $\text{rge}(\theta'')$. Since f and φ are injective, we can extend their composition $\varphi(f(\cdot))$ to a suitable permutation ρ which leaves ω unchanged, such that $R = P\rho$. But then $R \Downarrow_\omega$ iff $P \Downarrow_\omega$ (property of L').

Combining, we have: P has a match iff $P \Downarrow_\omega$, as required. □

We can use Theorem 3.9 and our various preceding positive and negative results to state some non-encodability results:

Theorem 3.10 *There is no encoding satisfying the conditions of Theorem 3.9 from* $\text{a}\pi_1^=$ *to* CCS_n *(all* $n \geq 1$*).*

Proof. By Theorems 3.3, 3.6 and 3.9. □

In connection with Theorem 3.10, we note that Carbone and Maffeis showed that there is no sensible encoding from $\text{a}\pi_1^=$ into the standard π-calculus with mixed choice (and without matching) [4, Theorem 4.1]. Our result here uses different conditions and holds for all levels of polyadic synchronisation in CCS.

Theorem 3.11 *There is no encoding satisfying the conditions of Theorem 3.9 from ppbMA to* CCS_n *(all* $n \geq 1$*).*

Proof. By Theorems 3.5, 3.6 and 3.9. □

Concerning Theorem 3.11, Phillips and Vigliotti [12] showed there is no encoding (under different conditions) from pure public MA (with open) to CCS. Previously they showed that there is no encoding (under yet other conditions) from pure public boxed MA to $\text{a}\pi_1$ [11].

4 Strong Matching Systems

In this section we investigate *strong* matching systems, where if there is a match then every computation is *guaranteed* to succeed. We show that the full π-calculus $\mathrm{f}\pi_1$ has strong matching systems of every finite degree (Theorem 4.4).

Definition 4.1 *A* strong matching system (SMS) of degree n *is a tuple* $(C, S, x_1, \ldots, x_n)$ *where* C *and* S *are processes and* $x_1, \ldots, x_n$ *are distinct names, such that for all finite index sets* I *and* J*, and all substitutions* σ_i $(i \in I)$ *and* θ_j $(j \in J)$ *where* $\mathrm{dom}(\sigma_i) = \mathrm{dom}(\theta_j) = \{x_1, \ldots, x_n\}$*, defining*

$$MS \stackrel{\mathrm{df}}{=} \prod_{i \in I} C\sigma_i \mid \prod_{j \in J} S\theta_j$$

(i) *if* $MS \Downarrow_\omega$ *then* $\exists i \in I, j \in J.\ \sigma_i = \theta_j$*;*

(ii) *if* $\exists i \in I, j \in J.\ \sigma_i = \theta_j$ *then* $\forall MS'.\ MS \Rightarrow MS'$ *implies* $MS' \Downarrow_\omega$*;*

(iii) *there are no infinite reduction sequences starting from* MS*.*

A replicated SMS *(!SMS for short) is defined as an SMS, except that we require the servers to be replicated, so that*

$$MS \stackrel{\mathrm{df}}{=} \prod_{i \in I} C\sigma_i \mid \prod_{j \in J} !S\theta_j$$

Observe that if $(C, S, \overrightarrow{x})$ is a !SMS then $(C, !S, \overrightarrow{x})$ is an SMS. Also, if $(C, S, \overrightarrow{x})$ is an SMS then $(C, S, \overrightarrow{x})$ is a WMS.

The notion of !SMS is quite close to the original formulation of matching system in [4]. It differs from the original MS in two ways: Firstly, we omit the identifier for the client, so that client and server are symmetrical. Secondly, we allow both client and server to contain free names not drawn from $\overrightarrow{x}$.

The next result is similar to [4, Theorem 4.2]:

Theorem 4.2 *For all non-negative integer numbers* n *and* m*, there is a* !*SMS of degree* m *in* $\mathrm{f}\pi_n$ *if and only if* $n \geq m$*.*

Proof. ($\Leftarrow$) Choosing

$$C_m \stackrel{\mathrm{df}}{=} \overline{x_1 \cdot \ldots \cdot x_m}\langle\rangle \qquad S_m \stackrel{\mathrm{df}}{=} x_1 \cdot \ldots \cdot x_m(z).\overline{\omega}\langle\rangle$$

we have that $(C_m, S_m, x_1, \ldots, x_m)$ is a strong matching system of degree m.

($\Rightarrow$) The idea is that a client can be endlessly "fooled" into interaction with servers which only partially match, giving rise to an infinite computation.

Consider the minimal case where $m = n + 1$ and suppose $(C, S, x_1, \ldots, x_m)$ is a !SMS of degree m in π_n. Let σ be an injective substitution of fresh names. Then

$$P \stackrel{\mathrm{df}}{=} C\sigma \mid S\sigma$$

is a matching instance of $(C, S, x_1, \ldots, x_m)$. Note that for any such σ and any R, if there is a Q such that $R\sigma \Rightarrow Q$ then there is also an R' such that $R \Rightarrow R'$ and $Q = R'\sigma$.

By point (iii) of Definition 4.1, there must be C' and S' such that $C\sigma \Rightarrow C'\sigma$ and $C'\sigma \not\rightarrow$, and similarly $S\sigma \Rightarrow S'\sigma$ and $S'\sigma \not\rightarrow$. By point (ii) of Definition 4.1, it

must be the case that $P \Rightarrow (C'\sigma \mid S'\sigma) \Downarrow_\omega$. By the contrapositive of point (i) of Definition 4.1, it must be the case that $C'\sigma \not\downarrow_\omega$ and $S\sigma \not\downarrow_\omega$. Hence, it must be the case that $C'\sigma \mid S'\sigma \rightarrow P' \Downarrow_\omega$ for some appropriate P'.

By definition of reduction, without loss of generality, we can assume that $C'\sigma \downarrow_{\overrightarrow{a_1}}, \ldots, C'\sigma \downarrow_{\overrightarrow{a_k}}$ and $S'\sigma \downarrow_{\overline{\overrightarrow{a_1}}}, \ldots, S'\sigma \downarrow_{\overline{\overrightarrow{a_k}}}$, where $\overrightarrow{a_1}, \ldots, \overrightarrow{a_k}$ are all of the possible channels on which the two processes are ready to communicate. Since $m > n$, for each $j \in [1..k]$ there exists i such that $\sigma(x_i) \notin \overrightarrow{a_j}$. Let ρ_j be defined as $\rho_j(x_i) = d_j$, for a fresh d_j, and $\rho_j(x_h) = \sigma(x_h)$ otherwise. By construction, since both σ and each ρ_j are injective and fresh, we have $S\rho_j \Rightarrow S'\rho_j$. Since σ and ρ_j agree on $\overrightarrow{a_j}$, it must be the case that $S'\rho_j \downarrow_{\overline{\overrightarrow{a_j}}}$. By the contrapositive of point (i) of Definition 4.1, it must be the case that

$$P_0^j \stackrel{\text{df}}{=} (C\sigma \mid S\rho_j) \not\Downarrow_\omega .$$

However, because of the complementary barbs, there must be P_1^j such that $P_0^j \rightarrow P_1^j$. Still, since

$$P_2^j \stackrel{\text{df}}{=} P_0^j \mid S\sigma$$

is a valid instance of a matching system, by point (ii) it must be the case that $P_2^j \Downarrow_\omega$. Moreover, since

$$P_2^j \Rightarrow P_3^j \stackrel{\text{df}}{=} (P_1^j \mid S\sigma)$$

it must be the case that $P_3^j \Downarrow_\omega$.

We have established above that there must be S' such that $S\sigma \Rightarrow S'\sigma$ and $S'\sigma \not\rightarrow$, and $S'\sigma \downarrow_{\overline{\overrightarrow{a_1}}}, \ldots, S'\sigma \downarrow_{\overline{\overrightarrow{a_k}}}$. Similarly, there must be P_4^j such that $P_1^j \Rightarrow P_4^j \downarrow_{\overrightarrow{a_i}}$ for some i in $[1..k]$. By considering now

$$P_0 \stackrel{\text{df}}{=} C\sigma \mid \prod_{j \in [1..k]} !S\rho_j$$

we have a contradiction because the system can enter a loop: each $S\rho_j$ intercepts the corresponding attempt that $C\sigma$ must keep repeating in order to communicate with a potential matching server $S\sigma$. □

Theorem 4.3 *For $n \geq 2$, there is no !SMS of degree n in MA.*

Proof. The idea is similar to the proof of Theorem 4.2. □

By contrast with Theorem 4.2, we show that $\mathrm{f}\pi_1$ is strong enough to have SMSs of all degrees:

Theorem 4.4 *There is a strong matching system of degree n in $\mathrm{f}\pi_1$.*

Proof. (Sketch) Lists and operations on lists can be encoded in $\mathrm{f}\pi_1$ without introducing divergence, using only restricted names (we consider the encoding given by Turner [14]). For example the list $[a, b]$, accessible through channel x, is represented by the process $\nu y, z(!x(n, c).\overline{c}\langle a, y\rangle \mid !y(n, c).\overline{c}\langle b, z\rangle \mid !z(n, c).\overline{n}\langle\rangle)$. Note that this list above can be passed around as a single value by passing the name x. Below, we use the context $L[-] \stackrel{\text{df}}{=} \nu\, \overrightarrow{l}\,(L \mid -)$ to denote the machinery to implement lists and

head, tail, concatenation, etc. operations in $\mathrm{f}\pi_1$. We assume that the names $\overrightarrow{l}$, used to implement the list operations, are fresh, and that $\mathsf{fn}(L[\mathbf{0}]) = \emptyset$. Consider the processes

$$C_n \stackrel{\mathrm{df}}{=} L[\overline{e}\langle[x_1,\ldots,x_n],[\,]\rangle]$$

$$S_m \stackrel{\mathrm{df}}{=} L[\nu a \begin{pmatrix} \overline{a}\langle[\,],[x_1,\ldots,x_n]\rangle \mid !a(x,y).(\overline{e}\langle x,y\rangle + \\ e(z,w).F(x@z,y@w,x',y').\overline{a}\langle x',y'\rangle) \end{pmatrix}]$$

where function F takes as input the lists $x@z, y@w$ and returns the lists x', y' obtained by removing all the matching pairs from $x@z$ and $y@w$ (@ stands for list concatenation). For each matching pair, F produces the barb ω. Such a function can be implemented in $f\pi_1$ without introducing divergence, and is defined in OCAML by the code

```
let rec rd = function
   (l1,[],l3,[]) -> [l1,l3]
  |(l1,[],l3,b::l4) -> rd(l1,[],l3@[b],l4)
  |(l1,a::l2,l3,[]) -> rd(l1@[a],l2,[],l3)
  |(l1,a::l2,l3,b::l4) -> if a=b then barb(omega); rd(l1,l2,[],l3@l4)
                          else rd(l1,a::l2,l3@[b],l4)
```

where *barb* is a user-defined function representing the barb. Note that the last clause defining F uses the conditional with a boolean guard given by name matching; this operation can be easily encoded in $f\pi_1$ because the language contains name matching and mismatching. □

There is an essential use of mixed choice in the proof of Theorem 4.4. We conjecture that without mixed choice it is impossible to get SMSs of degree higher than the level of synchronisation in the language:

Conjecture 4.5 *For $m > n$ there is no SMS of degree m in $\mathrm{a}\pi_n$.*

5 Conclusions and Further Work

We have adapted the notion of matching system from earlier work by Carbone and Maffeis. We have seen that there are two main types of matching system, the weak and the strong, depending on whether successful termination is possible or guaranteed (in the event of a match between some client and some server). In the strong case, there are two subtypes of matching system, depending on whether the server is required to be a replication or not (the former being the stronger of the two).

These notions can be used to "grade" process calculi according to how good they are at treating synchronisation on several different names as a single transaction.

We have seen that the full π-calculus is strong enough to have strong matching systems of all degrees, but not strong enough to have replicated strong matching systems of degree greater than one.

We also showed that the asynchronous π-calculus with matching has weak matching systems of every finite degree. Our work leaves open the question of whether the asynchronous π-calculus has a strong matching system of degree two or higher. We conjecture that the answer is no.

We showed that the calculus of Mobile Ambients has weak matching systems of all finite degrees. Furthermore, MA does not have replicated strong matching systems of degree two or higher. Our work leaves open the question of whether MA has a strong matching system of degree two or higher. Again, we conjecture that the answer is no.

We showed that CCS does not have weak matching systems of degree greater than one. By our result on preservation of weak matching systems by suitable encodings, we could deduce a non-encodability result for the asynchronous π-calculus with matching into CCS with all levels of polyadic synchronisation.

Acknowledgements

We thank Uwe Nestmann, Diletta Cacciagrano and the anonymous referees for their helpful comments and suggestions.

References

[1] Banach, R. and F. van Breugel, *Mobility and modularity: expressing pi-calculus in CCS (extended abstract)* (1998), draft.

[2] Boreale, M., *On the expressiveness of internal mobility in name-passing calculi*, Theoretical Computer Science **195** (1998), pp. 205–226.

[3] Bugliesi, M., G. Castagna and S. Crafa, *Access control for mobile agents: the calculus of Boxed Ambients*, ACM Transactions on Programming Languages and Systems **26** (2004), pp. 57–124.

[4] Carbone, M. and S. Maffeis, *On the expressive power of polyadic synchronisation in π-calculus*, Nordic Journal of Computing **10** (2003), pp. 70–98.

[5] Cardelli, L. and A.D. Gordon, *Mobile ambients*, Theoretical Computer Science **240** (2000), pp. 177–213.

[6] De Nicola, R. and M. Hennessy, *Testing equivalences for processes*, Theoretical Computer Science **34** (1984), pp. 83–134.

[7] Gorla, D., *Comparing calculi for mobility via their relative expressive power*, Technical Report 05/2006, Dip. di Informatica, Univ. di Roma "La Sapienza", Italy (2006).

[8] Merro, M. and D. Sangiorgi, *On asynchrony in name-passing calculi*, Mathematical Structures in Computer Science **14** (2004), pp. 715–767.

[9] Milner, R., "Communicating and Mobile Systems: the π-calculus," Cambridge University Press, 1999.

[10] Palamidessi, C., *Comparing the expressive power of the synchronous and the asynchronous π-calculi*, Mathematical Structures in Computer Science **13** (2003), pp. 685–719.

[11] Phillips, I.C.C. and M. Vigliotti, *Electoral systems in ambient calculi*, in: *Proceedings of 7th International Conference on Foundations of Software Science and Computation Structures, FoSSaCS 2004*, Lecture Notes in Computer Science **2987** (2004), pp. 408–422.

[12] Phillips, I.C.C. and M. Vigliotti, *Leader election in rings of ambient processes*, Theoretical Computer Science **356** (2006), pp. 468–494.

[13] Sangiorgi, D., *π-calculus, internal mobility and agent-passing calculi*, Theoretical Computer Science **167** (1996), pp. 235–274.

[14] Turner, D.N., "The Polymorphic Pi-calculus: Theory and Implementation," Ph.D. thesis, University of Edinburgh (1995).