Electoral Systems in Ambient Calculi

Iain Phillips and Maria Grazia Vigliotti

Department of Computing, Imperial College London, England
{iccp,mgv98}@doc.ic.ac.uk

Abstract. This paper compares the expressiveness of ambient calculi against different dialects of the pi-calculus. Cardelli and Gordon encoded the asynchronous pi-calculus into their calculus of Mobile Ambients (MA). Zimmer has shown that the synchronous pi-calculus without choice can be encoded in pure (no communication) Safe Ambients. We show that pure MA without restriction has symmetric electoral systems, that is, it is possible to solve the problem of electing a leader in a symmetric network. By the work of Palamidessi, this implies that pure MA without restriction is not encodable (under certain conditions) in the pi-calculus with separate choice. We adapt the work of Carbone and Maffeis to show that pure MA cannot be encoded (under certain other conditions) into the pi-calculus with mixed choice (but without matching).

1 Introduction

The π-calculus [16, 15] has acquired a fundamental role in modelling concurrent systems. In particular, the name-passing paradigm, on which the π-calculus is based, has proven to be a powerful and simple framework for describing different scenarios appearing in concurrency. Although the π-calculus remains a cornerstone within the panorama of process calculi, there has been the feeling that explicit constructs are needed for modelling the impressive and fast-growing reality of the Internet.

In recent years many calculi [10, 22, 8] have been proposed in order to represent locations, code mobility, abstract domains and security, which seem to be the main features of computation over the World Wide Web. Cardelli and Gordon introduced Mobile Ambients (MA) [7] as a foundational calculus for representing distributed computation, mobility in terms of software and hardware moving around, authorisation control etc., i.e. phenomena present over the Internet. The main advantage of MA is the simple underpinning unifying concept of *ambient*. Ambients are meant to represent bounded places for computation such as: concrete locations, concrete domains, abstract domains, laptop computers. Ambients move into and out of other ambients bringing along moving code, static processes and possibly other ambients. Due to its simplicity and power MA has been enormously successful (we refer to the ambient web-page: xdguan.freezope.org/wiki/AmbientCalculiOnline); moreover MA has been perceived as a fundamental calculus for representing different issues over the Internet, such as policies, security issues, etc.

Of course, the scientific community has been very interested in the comparison between these two fundamental process calculi. A basic issue is the extent to which the π-calculus (or any of its dialects) can be encoded into MA (or any of its dialects, which we refer to as *ambient calculi*). The asynchronous π-calculus [11, 3] (a fragment without the choice operator, and with no continuation for the output) has been encoded into MA with the use of the communication primitives [7]. The asynchronous π-calculus has also been encoded into the Push and Pull ambient calculus (PAC) [19]. As far as *pure* ambient calculi (i.e. calculi without communication) are concerned, Zimmer [23] has encoded the synchronous π-calculus without choice into pure Safe Ambients (SA) [13]. These encodings imply that ambient calculi are at least as expressive as the π-calculus (without choice). Of course, this poses the question of whether MA (or any of its dialects) can be encoded in any of the dialects of the π-calculus. This paper directly addresses this issue, which has been an open question among ambient calculi researchers.

A seminal result on expressiveness for the π-calculus is due to Palamidessi [18], who established that the π-calculus with mixed choice (i.e. where the summands in a choice can be a mixture of inputs or outputs) is strictly more expressive than the π-calculus with separate choice (i.e. where the summands must be all inputs or all outputs). In this paper we prove that a fragment of MA without restriction, communication primitives and the open capability is not encodable in the π-calculus with separate choice. Following Palamidessi, we achieve this using the problem of electing a leader in a symmetric network (symmetric leader election, or SLE).

For the π-calculus case mixed choice is crucial for writing a program that solves SLE. Choice is not present as a primitive construct in the ambient world. Nevertheless, mobility in MA has the power of *pre-emption*—of inhibiting alternatives—even though it cannot remove alternatives completely, as can a choice operator. This pre-emptive power is enough to break symmetry and elect a leader. More precisely, in standard MA with subjective movement (where ambients move themselves) we show that it is exactly the in capability which breaks symmetry. By contrast, in MA with objective movement (where ambients are moved from outside) there is no power to break symmetry, and so SLE cannot be solved.

Certainly in this framework results crucially depend on the definition of *encoding*. If the criteria for an encoding are too strong, then negative results are meaningless. If the criteria are too weak, nearly every function between languages can be viewed trivially as an encoding. In this paper we consider encodings which are "distribution-preserving", "permutation-preserving" and "observation-respecting", very much following the same criteria as in [18]. "Distribution-preserving" means preserving parallel composition in the encoding, to the end of avoiding that the translation makes use of third parties. "Permutation-preserving" means that the encodings are well-behaved with respect to bijective renamings. "Observation-respecting" means that processes are distinguished if they differ on the observable properties of their maximal computations. These criteria will be made more formal in Section 3.

With a different notion of encoding, Carbone and Maffeis [6] show that the matching operator is a primitive in the π-calculus and cannot be encoded. They require that encodings preserve non-injective substitutions

and not just permutations of names. In this paper, we show that in pure MA the matching operator is not a primitive. Nested locations and restriction allow one to encode faithfully the behaviour of the matching operator. Moreover, with much the same criteria on encoding as used by Carbone and Maffeis [6], we can show that pure MA is not encodable into the π-calculus with mixed choice but without matching.
Although this paper deals mostly with MA, the results can be extended to PAC and SA. For the details we refer to [21].
The rest of the paper is organised as follows: in Section 2 we present the preliminaries for the π-calculus and MA; in Section 3 we look at electoral systems for the two calculi, and thereby obtain separation results; in Section 4 we consider the matching operator and obtain further separation results; conclusions follow.

2 The Calculi

2.1 The π-calculus

We briefly review the basics for the standard π-calculus with mixed choice π_m [16, 15, 20]. We shall assume the existence of an infinite set $\mathcal{N}$ of names, ranged over by $m, n, x, y, \ldots$ and other lower-case letters. The set of processes of π_m is given by the following grammar:

$$P, Q ::= \sum_{i \in I} \alpha_i.P \;\mid\; !P \;\mid\; P \mid Q \;\mid\; (\nu n)P \qquad \alpha ::= m(n) \;\mid\; \overline{m}\langle n\rangle \;\mid\; \tau$$

where I is a finite set. A *summation* process $\sum_{i \in I} \alpha_i.P_i$ represents a choice among the different processes $\alpha_i.P$; $\mathbf{0}$, the inactive process, is an abbreviation for a summation where $I = \emptyset$. We shall feel free to omit trailing $\mathbf{0}$s. *Input* on channel m, $m(n).P$, can be thought of as a channel m that is waiting for an input before acting as P; the name n is bound. *Output* $\overline{m}\langle n\rangle.P$ can be seen as name n sent over channel m before acting as P. τ is the silent action. *Replication* $!P$ simulates recursion by spinning off copies of P, $P \mid Q$ is the *parallel composition* of two processes, and $(\nu n)P$ (*restriction*) creates a new name n in P.
The set $fn(P)$ of *free names* of a process P is defined in the standard way, taking into account that the binding operators are restriction and input. We deem processes to be syntactically equal ($=$) if they are the same up to alpha-conversion of bound names. *Structural congruence* $\equiv$ is the least congruence generated by the following laws:

$$\begin{array}{rl} P \mid \mathbf{0} \equiv P & (\nu n)\mathbf{0} \equiv \mathbf{0} \\ P \mid Q \equiv Q \mid P & (\nu m)(\nu n)P \equiv (\nu n)(\nu m)P \\ (P \mid Q) \mid R \equiv P \mid (Q \mid R) & (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q \text{ if } n \notin fn(P) \\ !P \equiv P \mid\; !P & \end{array}$$

together with reordering of summations. We shall write $\tilde{n}$ for a set of restricted names $\{n_1, \ldots, n_k\}$. The *reduction relation* $\rightarrow$ is defined as follows:

$$\begin{array}{rcll} \tau.P + S & \rightarrow & P & \text{RED TAU} \\ (n(x).P + S) \mid (\overline{n}\langle y\rangle.Q + T) & \rightarrow & P\{y/x\} \mid Q & \text{RED COMM} \\ P \rightarrow P' & \Longrightarrow & P \mid Q \rightarrow P' \mid Q & \text{RED PAR} \\ P \rightarrow P' & \Longrightarrow & (\nu n)P \rightarrow (\nu n)P' & \text{RED RESTR} \\ P \equiv Q \rightarrow Q' \equiv P' & \Longrightarrow & P \rightarrow P' & \text{RED CONG} \end{array}$$

In the above we let S, T range over summations. The notation $P\{y/x\}$ means the substitution of y for every free occurrence of x in P. We write $\twoheadrightarrow$ for the transitive and reflexive closure of $\rightarrow$.
A process P *exhibits barb* n, written as $P \downarrow n$, iff

$$P \equiv (\nu\tilde{p})((\overline{n}\langle q\rangle.P' + S) \mid P'')$$

with $n \notin \tilde{p}$. These barbs represent the most basic observations that we can make of processes. We only consider output barbs; input barbs are not needed, and by omitting them we obtain greater uniformity with MA. A process P *eventually exhibits barb* n, written $P \Downarrow n$, iff $P \twoheadrightarrow Q$ and $Q \downarrow n$ for some Q. If $P \downarrow n$ we say that n is a *strong* barb of P, and if $P \Downarrow n$ we say that n is a *weak* barb of P. We define $P \Downarrow$ iff there is some $n \in \mathcal{N}$ such that $P \Downarrow n$.
The π-calculus with separate choice π_s [17] is the sub-calculus of π_m where summations cannot mix input and output guards. The grammar is the following:

$$\alpha^{\mathrm{I}} ::= m(n) \mid \tau \qquad \alpha^{\mathrm{O}} ::= \overline{m}\langle n\rangle \mid \tau$$

$$P, Q ::= \sum_{i\in I} \alpha_i^{\mathrm{I}}.P \mid \sum_{i\in I} \alpha_i^{\mathrm{O}}.P \mid\ !P \mid P \mid Q \mid (\nu n)P$$

One could regard π_s as having the same expressive strength as the asynchronous π-calculus [11, 3], since π_s can be encoded in asynchronous π-calculus [17].

2.2 Mobile Ambients

As in the π-calculus, we assume the existence of a set $\mathcal{N}$ of names. The processes of MA [7] are given by the following grammar:

$$P, Q ::= \mathbf{0} \mid\ !P \mid P \mid Q \mid (\nu n)P \mid n[\,P\,] \mid M.P \mid (n).P \mid \langle n\rangle$$

Intuitively $\mathbf{0}$ stands for the inactive process, and we shall feel free to omit trailing $\mathbf{0}$s and write empty ambients as $n[\,]$ rather than $n[\,\mathbf{0}\,]$. $n[\,P\,]$ is the ambient n containing the active process P and $M.P$ is process P guarded by the capability M. Here M is defined in the following grammar:

$$M ::= \mathsf{in}\, n \mid \mathsf{out}\, n \mid \mathsf{open}\, n$$

The meaning of the capabilities is intuitively the following: $\mathsf{open}\, n$ dissolves an ambient with name n; $\mathsf{in}\, n$ allows an ambient to enter another ambient named n; finally $\mathsf{out}\, n$ allows an ambient to exit its own parent named n. The anonymous communication primitives are: $(n).P$ will input any message in the top-level ambient, and $\langle n \rangle$ is an output with no continuation. This form of communication is more limited than in standard MA [7], where sequences of capabilities can be passed; we have adopted this formulation for simplicity. The other operators have the same meaning as in the π-calculus.

The free names $fn(P)$ of P are defined in the standard way, taking into account that the binding operators are restriction and input. Processes are syntactically equal ($=$) if they are identical apart from alpha-conversion of bound names. *Structural congruence* $\equiv$ is defined as for the π-calculus with one extra rule:

$$(\nu m)n[\,P\,] \equiv n[\,(\nu m)P\,] \quad \text{if } n \neq m$$

The *reduction relation* $\rightarrow$ is defined as follows:

$$\begin{array}{rcll}
m[\,\mathsf{in}\, n.P \mid Q\,] \mid n[\,R\,] & \rightarrow & n[\,m[\,P \mid Q\,] \mid R\,] & \text{RED IN} \\
n[\,m[\,\mathsf{out}\, n.P \mid Q\,] \mid R\,] & \rightarrow & m[\,P \mid Q\,] \mid n[\,R\,] & \text{RED OUT} \\
\mathsf{open}\, n.P \mid n[\,Q\,] & \rightarrow & P \mid Q & \text{RED OPEN} \\
(m).P \mid \langle n \rangle & \rightarrow & P\{n/m\} & \text{RED COMM} \\
P \rightarrow Q & \Longrightarrow & P \mid R \rightarrow Q \mid R & \text{RED PAR} \\
P \rightarrow Q & \Longrightarrow & n[\,P\,] \rightarrow n[\,Q\,] & \text{RED AMB} \\
P \rightarrow Q & \Longrightarrow & (\nu n)P \rightarrow (\nu n)Q & \text{RED RESTR} \\
P \equiv P' \rightarrow Q' \equiv Q & \Longrightarrow & P \rightarrow Q & \text{RED CONG}
\end{array}$$

Again, $\twoheadrightarrow$ stands for the symmetric and transitive closure of $\rightarrow$.

The most basic observation we can make of an MA process is the presence of an unrestricted top-level ambient. A process P *exhibits barb* n, written as $P \downarrow n$, iff $P \equiv (\nu \tilde{p})(n[\,P'\,] \mid P'')$ with $n \notin \tilde{p}$. As for the π-calculus, we define weak barbs by $P \Downarrow n$ iff $P \twoheadrightarrow Q$ and $Q \downarrow n$ for some Q. We define $P \Downarrow$ iff there is some $n \in \mathcal{N}$ such that $P \Downarrow n$.

We shall be interested in certain fragments of MA. By MA^{io} we mean pure MA without restriction and the open capability. Finally we denote MA with all operators except the in (resp. out) capability by $\text{MA}^{-\mathsf{in}}$ (resp. $\text{MA}^{-\mathsf{out}}$).

3 Electoral Systems

Expressiveness results have positive aspects, where encodings are given. However, one can also show that some encodings do not exist. One way of proceeding is to show that there is a problem that can be solved in one calculus, but not in the other. We have been inspired by Palamidessi's work on expressiveness results for the π-calculus [18]. In order to separate the π-calculus with mixed choice from the π-calculus with separate choice, Palamidessi considers the *symmetric leader election problem* (SLE). The idea is that, given a symmetric network, a leader has to be elected without the help of a centralised server. Symmetry means informally that all the processes in the network can perform the same actions up to some kind of renaming. This kind of problem is widely studied in the literature of distributed computing [14, 2] and process algebra [1, 4, 9, 18]. Problems are categorised by the topology of the network and the

way in which the winner is declared. A network could be a ring or a fully connected graph, while the winner could be announced by one process only, or by all the processes. Palamidessi has shown that a symmetric network in the π-calculus without mixed choice cannot solve the problem of electing a leader in a guaranteed fashion. The idea of the proof is that the symmetry of the network need never be broken.
We present electoral systems for both the π-calculus and MA. We will then show that $\mathrm{MA}^{\mathsf{io}}$ can indeed solve the leader election problem, which is sufficient to separate MA from the π-calculus without mixed choice.

3.1 Networks and Electoral Systems

In this section we define networks and electoral systems. These notions are general enough to be applied to any name-based calculus whose operational semantics is defined in terms of a reduction relation. Hence the definitions below apply equally well to the ambient calculus and to the π-calculus.
We assume that we are dealing with some generic process calculus with a set $\mathcal{N}$ of names, restriction and parallel composition operators, and notions of structural congruence $\equiv$, reduction $\rightarrow$ and barb $\downarrow$.
We assume that $\mathcal{N}$ includes a set of *observables* $\mathsf{Obs} = \{\omega_i : i \in \mathbb{N}\}$ such that for all i, j we have $\omega_i \neq \omega_j$ if $i \neq j$. The observables will be used by networks to communicate with the outside world.

Definition 1. *Let P be a process. A* computation *$\mathcal{C}$ of P is a (finite or infinite) sequence $P = P_0 \rightarrow P_1 \rightarrow \cdots$. It is* maximal *if it cannot be extended, i.e. either $\mathcal{C}$ is infinite, or else it is of the form $P_0 \rightarrow \cdots \rightarrow P_h$ where $P_h \not\rightarrow$.*

Definition 2. *Let $\mathcal{C}$ be a computation $P_0 \rightarrow \cdots \rightarrow P_h \rightarrow \cdots$. We define the* observables *of $\mathcal{C}$ as $\mathsf{Obs}(\mathcal{C}) = \{\omega \in \mathsf{Obs} : \ \exists h \ \ P_h \downarrow \omega\}$.*

Our definition of computation is different from Palamidessi's. She uses labelled transition systems, while we use unlabelled reduction relations (i.e. τ-actions). This seems appropriate for the ambient world, where the intended semantics is defined in terms of a reduction relation and there is no agreement within the scientific community about a good labelled transition system. It also works well for the π-calculus.
Networks are just collections of processes running in parallel:

Definition 3. *A* network Net *of size k is a process in the form $(\nu\tilde{x})(P_0 \mid \ldots \mid P_{k-1})$.*

Networks inherit a notion of computation from processes; we do not assume that a network reduces to a network (although in fact every process is a network of size 1 in a trivial sense). In particular, we do not require that a network maintains its size during a computation. This would be too restrictive, as will become clear when we consider electoral systems in the ambient calculus.
A *permutation* is a bijection $\sigma : \mathcal{N} \rightarrow \mathcal{N}$ such that σ preserves the distinction between observable and non-observable names, i.e. $n \in \mathsf{Obs}$

iff $\sigma(n) \in \mathsf{Obs}$. Any permutation σ gives rise in a standard way to a mapping on processes, where $\sigma(P)$ is the same as P, except that any free name n of P is changed to $\sigma(n)$ in $\sigma(P)$, with bound names being adjusted as necessary to avoid clashes.
A permutation σ induces a bijection $\hat{\sigma} : \mathbb{N} \to \mathbb{N}$ defined as follows: $\hat{\sigma}(i) = j$ where $\sigma(\omega_i) = \omega_j$. Thus for all $i \in \mathbb{N}$, $\sigma(\omega_i) = \omega_{\hat{\sigma}(i)}$. We use $\hat{\sigma}$ to permute the indices of processes in a network.

Definition 4. *Let* $\mathsf{Net} = (\nu\tilde{x})(P_0 \mid \ldots \mid P_{k-1})$ *be a network of size* k*. An* automorphism *on* Net *is a permutation* σ *such that (1)* $\hat{\sigma}$ *restricted to* $\{0, \ldots, k-1\}$ *is a bijection, and (2)* σ *preserves the distinction between free and bound names, i.e.* $x \in \tilde{x}$ *iff* $\sigma(x) \in \tilde{x}$*.*

Definition 5. *Let* σ *be an automorphism on a network of size* k*. For any* $i \in \{0, \ldots, k-1\}$ *the* orbit $\mathcal{O}_{\hat{\sigma}}(i)$ *generated by* $\hat{\sigma}$ *is defined as follows:*

$$\mathcal{O}_{\hat{\sigma}}(i) = \{i, \hat{\sigma}(i), \hat{\sigma}^2(i), \ldots, \hat{\sigma}^{h-1}(i)\}$$

where $\hat{\sigma}^j$ *represents the composition of* $\hat{\sigma}$ *with itself* j *times, and* h *is least such that* $\hat{\sigma}^h(i) = i$*.*

Definition 6 (Symmetric Network). *[18] Let* $\mathsf{Net} = (\nu\tilde{x})(P_0 \mid \ldots \mid P_{k-1})$ *be a network of size* k *and let* σ *be an automorphism on it. We say that* Net *is* symmetric with respect to σ *iff for each* $i = 0, \ldots, k-1$ *we have* $P_{\hat{\sigma}(i)} = \sigma(P_i)$*. We say that* Net *is* symmetric *if it is symmetric with respect to some automorphism with a single orbit (which must have size* k*).*

Our definitions of automorphism and symmetric network differ from those of Palamidessi. She takes the network topology into account, and associates a hypergraph with a network. Automorphisms are defined with respect to this hypergraph, and a network is symmetric if it is symmetric with respect to *every* automorphism. We have chosen our definitions because they seem to capture exactly what is needed for our separation results.
Intuitively an electoral system is a network which reports a unique winner no matter how the computation proceeds.

Definition 7 (Electoral system). *A network* Net *of size* k *is an* electoral system *if for every maximal computation* $\mathcal{C}$ *of* Net *there exists an* $i < k$ *such that* $\mathsf{Obs}(\mathcal{C}) = \{\omega_i\}$*. An electoral system is said to be* symmetric *if the network is symmetric.*

Thus each maximal computation gives exactly one winner. It does not matter which process in the original network displays the observable barb; indeed, in MA this is not even necessarily meaningful, as processes can intermingle using movement capabilities.
Our definition of electoral system is different from Palamidessi's. For Palamidessi the requirement for an electoral system is that *every* process in the electoral system can execute a special action $\overline{out}\langle i\rangle$. In other words everyone is aware of the leader. As she states, her results would hold under the alternative requirement that *exactly one* process announces the winner. Our notion is weaker, in that we merely require that *at least one* process announces the winner, and it is left open how many processes make the announcement.

3.2 Calculi with Symmetric Electoral Systems

The π-calculus with mixed choice can elect a leader in a symmetric network according to Palamidessi's criteria. It is not difficult to see that π_m admits a symmetric electoral system also according to our new and weaker criteria.
The simplest non-trivial symmetric electoral system is for $k = 2$:

$$P_0 \stackrel{\mathrm{df}}{=} x_0(y) + \overline{x_1}\langle z\rangle.\overline{\omega_0}\langle z\rangle \qquad P_1 \stackrel{\mathrm{df}}{=} x_1(y) + \overline{x_0}\langle z\rangle.\overline{\omega_1}\langle z\rangle$$

$$\mathsf{Net} \stackrel{\mathrm{df}}{=} P_0 \mid P_1$$

The process which performs the output wins. Notice that if we were using labelled transitions we would have to restrict x_0 and x_1 globally in order to ensure synchronisation; this is not necessary with unlabelled transitions. The network is symmetric with respect to a single-orbit automorphism σ which swaps ω_0 with ω_1 and x_0 with x_1, with σ the identity on all other names. Hence we have:

Proposition 1. *In π_m there exists a symmetric electoral system of size* 2.

Notice that the link-passing capabilities of the π-calculus play no rôle in the electoral system given above; it is the mixed choice which is important. Palamidessi shows that in π_m there are symmetric electoral systems of size k for every k.
We now turn to MA. Recall that by $\mathrm{MA}^{\mathsf{io}}$ we mean pure MA without restriction and the open capability.

Proposition 2. *In* $\mathrm{MA}^{\mathsf{io}}$ *there exists a symmetric electoral system of size* 2.

Proof (Sketch). Let

$$\mathsf{Net} \stackrel{\mathrm{df}}{=} n_0[\,\mathsf{in}\,n_1.\omega_0[\,\mathsf{out}\,n_0.\mathsf{out}\,n_1\,]\,] \mid n_1[\,\mathsf{in}\,n_0.\omega_1[\,\mathsf{out}\,n_1.\mathsf{out}\,n_0\,]\,]\ .$$

The first process to perform an in wins. Notice that the in capability breaks symmetry to decide the winner, while the out capability enables the winner to be reported at the top level. $\square$

Theorem 1. *In* $\mathrm{MA}^{\mathsf{io}}$*, for any k, there exists a symmetric electoral system of size k.*

Proof (Sketch). For $0 \le i < k$, let $S_i^k \stackrel{\mathrm{df}}{=} \{0, \ldots, i-1, i+1, \ldots, k-1\}$, i.e. the natural numbers less than k excluding i. Let T_i^k be the set of all strings of length $k-1$ using the members of S_i^k exactly once each. Given an element s of T_i^k we denote by s^- the string which is s in reverse order. Let $n_0, \ldots, n_{k-1} \in \mathcal{N}$. With $\mathsf{in}\,(s)$ we mean the sequence of $\mathsf{in}\,n_j$ capabilities for each successive $j \in s$.

$$P_i \stackrel{\mathrm{df}}{=} n_i[\,\prod_{j \in S_i^k} !\mathsf{in}\,n_j \mid \prod_{s \in T_i^k} \omega_i[\,\mathsf{in}\,(s).\mathsf{out}\,(s^-).\mathsf{out}\,n_i\,]\,]$$

$$\mathsf{Net} \stackrel{\text{df}}{=} P_0 \mid \cdots \mid P_{k-1}$$

The idea is that the processes that take part in the election can enter one another, until they form a linear stack. At this point no further movement of the main ambients is possible, and the leader is the ambient n_i which is at the top of the stack. For some $s \in T_i^k$, an ambient ω_i can descend to the bottom of the stack using $\mathsf{in}\,(s)$, and then ascend to the top of the stack using $\mathsf{out}\,(s^-)$ (of course, it may start this process before the stack is fully formed). Finally ω_i uses $\mathsf{out}\,n_i$ to emerge at the top level, and i is declared the winner. Any ω_j ambient ($j \neq i$) will not be able to use up all its $\mathsf{in}\,(s)$ capabilities, and so will not be able to emerge at the top level. Hence, exactly one winner is declared for each computation. As in Proposition 2, the in capability breaks symmetry and chooses the winner, and the out capability is required to report the winner.

We thank Sergio Maffeis for suggesting the construction of the electoral system in this proof, which improves our previous construction. □

3.3 Calculi without Symmetric Electoral Systems

In this subsection we are going to show that there are calculi that do not admit a symmetric electoral system. First of all we reestablish within the present framework Palamidessi's result that the π-calculus with separate choice (π_s) does not admit a symmetric electoral system (Theorem 4.2 of [18]).

Theorem 2. *Let* Net *be a symmetric network of size* $k \geq 2$ *in* π_s. *Then* Net *cannot be an electoral system.*

The proof can be found in [21]. Much as in Palamidessi's proof, the idea is that if Net is symmetric we can find a computation which preserves symmetry, so that every k reductions we have again a symmetric network of size k. This computation can never declare a unique winner.

We have shown in Subsection 3.2 that $\mathrm{MA}^{\mathsf{io}}$ can solve SLE. We have also seen that SLE can be solved in π_m, but not in π_s, so that mixed choice is essential. We now show that the in and out capabilities of $\mathrm{MA}^{\mathsf{io}}$ are both required to solve SLE.

Theorem 3. *Let* Net *be a symmetric network of size* $k \geq 2$ *in* $\mathrm{MA}^{-\mathsf{in}}$. *Then* Net *cannot be an electoral system.*

The proof can be found in [21]. As in the proof of Theorem 2, the idea is that symmetry need never be broken.

Theorem 4. *Let* Net *be a symmetric network of size* $k \geq 2$ *in* $\mathrm{MA}^{-\mathsf{out}}$. *Then* Net *cannot be an electoral system.*

Proof (Sketch). We create a computation without a unique winner as follows: We perform top-level communications and open reductions for as long as possible, maintaining symmetry. Once these are exhausted, there can be no further such reductions as out is not available. By symmetry, so far either no process has won, or else all have, in which case Net is not an electoral system. In the first case, we can then continue freely with in and lower-level reductions. However no winner can emerge, since such reductions do not produce new top-level ambients. □

3.4 Separation Results

Again inspired by Palamidessi's work, we are now going to show that there exists no encoding from MA into π_s which satisfies certain conditions.

Definition 8. *Let L, L' be process languages. An encoding $[\![-]\!] : L \to L'$ is*

1. distribution-preserving *if for all processes P, Q of L, $[\![P \mid Q]\!] = [\![P]\!] \mid [\![Q]\!]$;*
2. permutation-preserving *if for any permutation of names σ in L there exists a permutation θ in L' such that $[\![\sigma(P)]\!] = \theta([\![P]\!])$ and the permutations are compatible on observables, in that for all $i \in \mathbb{N}$ we have $\sigma(\omega_i) = \theta(\omega_i)$;*
3. observation-respecting *if for any P in L,*
   (a) *for every maximal computation $\mathcal{C}$ of P there exists a maximal computation $\mathcal{C}'$ of $[\![P]\!]$ such that $\mathsf{Obs}(\mathcal{C}) = \mathsf{Obs}(\mathcal{C}')$*
   (b) *for every maximal computation $\mathcal{C}$ of $[\![P]\!]$ there exists a maximal computation $\mathcal{C}'$ of P such that $\mathsf{Obs}(\mathcal{C}) = \mathsf{Obs}(\mathcal{C}')$*

An encoding which preserves distribution and permutation is uniform.

In the above L and L' can have different sets of names, but they must have the same observables Obs.
The first two items in Definition 8 (i.e. uniformity) are as in Palamidessi. The condition of preserving distribution is important in ruling out encodings which make use of a central server. The third item requires some comment. Bougé [4] defined an encoding as "reasonable" if it maps electoral systems to electoral systems. The condition of respecting observations is our interpretation of Palamidessi's requirement of "preserving a reasonable semantics". She states that a reasonable semantics should distinguish processes which differ on the observables of their maximal computations. In fact, we only require part (b) to ensure that electoral systems are mapped to electoral systems; part (a) is added to make the condition more natural. In their version of Palamidessi's work, Sangiorgi and Walker [20] use a condition that if the observables of every maximal computation of a process P are singletons, then the same is true for the encoding of P. This obviously relates very directly to the need to preserve electoral systems. Finally, Ene and Muntian [9] use yet another formulation. As it only refers to finite computations, it would not be enough for our purposes.
Symmetric electoral systems are mapped to symmetric electoral systems by encodings satisfying Definition 8:

Lemma 1. *Suppose $[\![-]\!] : L \to L'$ is a uniform observation-respecting encoding. Suppose that Net is a symmetric electoral system of size k with no globally-bound variables. Then $[\![\mathsf{Net}]\!]$ is also a symmetric electoral system of size k.*

Proof. See [21]. □

Recall that the asynchronous π-calculus can be encoded in MA. The next result shows that the converse fails.

Corollary 1. *There does not exist a uniform observation-respecting encoding from* $\mathrm{MA}^{\mathsf{io}}$ *into* π_s.

Proof. By Proposition 2, Theorem 2 and Lemma 1. □

Similarly we have:

Corollary 2. *There does not exist a uniform observation-respecting encoding from* π_m *into* $\mathrm{MA}^{-\mathsf{in}}$, *or from* $\mathrm{MA}^{\mathsf{io}}$ *into* $\mathrm{MA}^{-\mathsf{in}}$ *or* $\mathrm{MA}^{-\mathsf{out}}$.

3.5 Objective Moves

Notice that for a symmetric network to be an electoral system there are two requirements:

1. Every computation has to break symmetry.
2. There must be the possibility of displaying the winner.

The results of Section 3.3 show that to solve SLE in MA both the in and out capabilities are essential, each capability corresponding to one of the above items. However one might wonder whether what is really important is the tree structure of ambients, and the capability to move up and down within it, as reflected in ambient calculi in general. We shall see in this subsection that not every ambient calculus can solve SLE, and so the precise nature of the movement capabilities is important.
We consider a variant of MA with *objective moves*, which we call $\mathrm{MA}^{\mathrm{ob}}$. This was first discussed by Cardelli and Gordon [7]. $\mathrm{MA}^{\mathrm{ob}}$ has two different capabilities with respect to standard MA. Instead of $\mathsf{in}\, n$ and $\mathsf{out}\, n$ there are $\mathsf{mv\ in}\, n$ and $\mathsf{mv\ out}\, n$. We replace RED IN and RED OUT by the following reduction rules:

$$\mathsf{mv\ in}\, n.P \mid n[\,Q\,] \to n[\,P \mid Q\,] \quad \text{RED OBJ-IN}$$
$$n[\,\mathsf{mv\ out}\, n.P \mid Q\,] \to P \mid n[\,Q\,] \quad \text{RED OBJ-OUT}$$

Theorem 5. *Let* Net *be a symmetric network of size* $k \geq 2$ *in* $\mathrm{MA}^{\mathrm{ob}}$. *Then* Net *cannot be an electoral system.*

The proof can be found in [21].

Corollary 3. *There does not exist a uniform observation-respecting encoding from* $\mathrm{MA}^{\mathsf{io}}$ *into* $\mathrm{MA}^{\mathrm{ob}}$, *or from* π_m *into* $\mathrm{MA}^{\mathrm{ob}}$.

Remark 1. Cardelli and Gordon also discuss a variant form of objective moves with a reduction rule of the form

$$\mathsf{mv}\, m\, \mathsf{in}\, n.P \mid m[\,Q\,] \mid n[\,R\,] \to P \mid n[\,m[\,Q\,] \mid R\,]$$

This form of objective move *can* break symmetry (like standard in).

4 Matching

The matching operator $[m=n]P$ was introduced in [16]. It expresses the comparison between two names; $[m=n]P$ behaves like P iff the name n is the same as m; otherwise $[m=n]P$ is inert. If matching is introduced as a primitive operator, structural congruence is augmented with the rule $[n=n]P \equiv P$.
In the case of the π-calculus (with or without mixed choice) Carbone and Maffeis showed that there does not exist a "sensible" encoding of the matching operator [6]. They define an encoding to be *sensible* if it is uniform (Definition 8) with respect to substitutions (not just permutations), preserves a reasonable semantics, and distinguishes deadlocks from livelocks.
We shall see that matching *can* be encoded in MA. We shall adapt the methods of Carbone and Maffeis to achieve a separation between pure MA and π_m.

4.1 The Encoding of Matching

Let $\mathrm{MA}^=$ denote MA with matching. We describe an encoding $[\![-]\!]_=$ from $\mathrm{MA}^=$ to MA. This makes use of the particular semantics of the ambient operator. A restricted ambient, invisible to the outside world, can contain processes with free variables that produce computation that is visible outside. Below is reported only the most important clause of the encoding. For the other operators the encoding is homomorphic.

$$[\![\, [m = n]P \,]\!]_= \stackrel{\mathrm{df}}{=} (\nu x\, y)(x[\, \mathsf{open}\, m.y[\, \mathsf{out}\, x\,] \mid n[\,]\,] \mid \mathsf{open}\, y.\mathsf{open}\, x.[\![P]\!]_=)$$

Observe that in general $\mathit{fn}([\![\, [m=n]P \,]\!]_=) = \mathit{fn}([m=n]P)$. If $m = n$ then the four steps of the encoding proceed in a deterministic fashion within the scope of the restriction, avoiding in this way any form of interference. If $m \neq n$ then the encoded matching is inert.
Let L be any process language with notions of reduction and barb. Assuming that the notion of context C is defined by following the syntax of L, we define a notion of weak bisimulation for reduction semantics.

Definition 9. *[12] A symmetric relation $\mathcal{S} \subseteq L \times L$ is an* contextual barbed bisimulation *if $P \,\mathcal{S}\, Q$ implies:*
- *for each name n, if $P \downarrow n$ then $Q \Downarrow n$;*
- *for any context C, whenever $C[P] \rightarrow P'$ then for some Q', $C[Q] \twoheadrightarrow Q'$ and $P' \,\mathcal{S}\, Q'$.*

Two processes P, Q are said to be contextual barbed equivalent *($P \approx Q$), if $P \,\mathcal{S}\, Q$ for some contextual barbed bisimulation $\mathcal{S}$.*

The following full abstraction theorem shows that our encoding of matching is well-behaved. Let $\approx$ (resp. $\approx^=$) denote contextual barbed equivalence on MA (resp. $\mathrm{MA}^=$).

Theorem 6. *For all $P, Q \in \mathrm{MA}^=$, $P \approx^= Q$ iff $[\![P]\!]_= \approx [\![Q]\!]_=$.*

Proof. See [21]. □

Matching can be encoded in other ambient calculi; this has been carried out for PAC [19], and the above can be easily adapted for SA.

4.2 Separation Results

So far we have shown that pure MA without restriction and open is not encodable in the π-calculus without mixed choice (Corollary 1). We will now show that pure MA is not encodable in the π-calculus with mixed choice and without matching (π_m). The result has been inspired by [6].

Definition 10. *Let L, L' be languages. An encoding $[\![-]\!] : L \to L'$ is*
1. substitution-preserving *if for any substitution of names σ in L there exists a substitution θ in L' such that $[\![\sigma(P)]\!] = \theta([\![P]\!])$;*
2. weak barb-respecting *if for any P in L, $P \Downarrow$ iff $[\![P]\!] \Downarrow$.*

The first condition is to be compared with the permutation-preserving condition of Definition 8. It is stronger, in that we move from permutations to arbitrary substitutions. However we no longer require θ and σ to agree on observables.
The second condition of Definition 10 corresponds to the observation-respecting condition of Definition 8. The two conditions are incompatible. However, if we strengthen the observation-respecting condition by insisting that *all* names are included in the observables, and if we assume that L and L' have the same set of names, then we have $P \Downarrow$ iff P has a computation $\mathcal{C}$ such that $\mathsf{Obs}(\mathcal{C}) \neq \emptyset$. So in this case observation-respecting implies weak-barb respecting.

Lemma 2. *Take $P \in \pi_m$ and any substitution σ. If $\sigma(P) \Downarrow$ then $P \Downarrow$.*

Proof. Similar to that of Proposition 4.1 of [6]. The result depends on *all* names in $\mathcal{N}$ being possible barbs. □

Lemma 2 is not true for MA. Neither does it hold for matching in π-calculus (consider a process $[m = n]\,\overline{x}\langle y\rangle$), and on this is based the proof in [6] that matching is not encodable in π_m.

Theorem 7. *There does not exist a substitution-preserving and weak barb-respecting encoding from pure* MA *into π_m.*

Proof. Assume there exists such an encoding $[\![-]\!]$. Let σ be a substitution with $\sigma(n) = m$ and all other names unchanged. Let $P \overset{\mathrm{df}}{=} (\nu\, r)(r[\,\mathsf{open}\, n.m[\,\mathsf{out}\, r\,] \mid m[\;]\,])$. We have $\sigma(P) \Downarrow$ but $P \not\Downarrow$. There is a substitution θ satisfying $[\![\sigma(P)]\!] = \theta([\![P]\!])$. But since $\sigma(P) \Downarrow$ we have $[\![\sigma(P)]\!] \Downarrow$ (weak barb-respecting condition). So $\theta([\![P]\!]) \Downarrow$. By Lemma 2 we have $[\![P]\!] \Downarrow$. Hence $P \Downarrow$, which is a contradiction. □

Notice that Theorem 7 was achieved without assuming that the encodings preserve distribution (Definition 8).

5 Conclusions and Further Work

We have investigated the relative strengths of the π-calculus and MA. We have used electoral systems to show that MA cannot be encoded in the π-calculus with separate choice π_s (under certain natural conditions on encodings); the other direction is already known to be possible. We have also seen that matching can be encoded in MA. We then saw that pure MA cannot be encoded in the π-calculus without matching (under different conditions on encodings).

We have seen that certain calculi can solve leader election problems in the presence of symmetric networks and others cannot. One way to approach this in a broader context is to categorise operators in languages as *symmetry-breaking* or *symmetry-preserving*. In MA entering another ambient is symmetry-breaking; however it is symmetry-preserving in objective MA ($\mathrm{MA}^{\mathrm{ob}}$). Similarly for the π-calculus, mixed choice is symmetry-breaking while separate choice is symmetry-preserving. As future work, we plan to investigate this difference for process languages in general. This would involve defining some sort of format that characterises symmetry-preserving operators.

We have dispensed with hypergraphs in the present formulation. They were used by Palamidessi to separate the π-calculus from CCS. If we were talking about graphs, in this article it would be fair to say that our examples of electoral systems have all been fully connected graphs. It remains for future work to study electoral systems for MA on graphs that are not fully connected. See [21] for more on this subject.

Finally, we need to consider how this work applies to related languages. From Theorem 1 it follows trivially that Boxed Ambients [5] can solve the symmetric leader election problem. So should the Seal calculus [22], since the (move in) seems to be a symmetry-breaking operator. On the other hand we speculate that the pure version of $D\pi$ [10] cannot solve such a problem. Informally one might think that the operator for flat locations is symmetry-preserving.

Acknowledgements. We thank Sergio Maffeis, Catuscia Palamidessi and Nobuko Yoshida for helpful discussions. We also thank the referees for their suggestions.

References

1. D. Angluin. Local and global properties in networks of processors. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, pages 82–93. ACM, 1980.
2. H. Attiya and J. Welch. *Distributed Computing*. The McGraw-Hill Companies, 1998.
3. G. Boudol. Asynchrony and the π-calculus. Technical Report 1702, INRIA Sophia-Antipolis, 1992.
4. L. Bougé. On the existence of symmetric algorithms to find the leaders in networks of communicating sequential systems. *Acta Informatica*, 25:179–201, 1988.
5. M. Bugliesi, G. Castagna, and S. Crafa. Boxed ambients. In *Proceedings of TACAS'01*, volume 2215 of *LNCS*, pages 38–63. Springer-Verlag, 2001.
6. M. Carbone and S. Maffeis. On the Expressive Power of Polyadic Synchronisation in π-calculus. *Nordic Journal of Computing*, 10(2):70–98, 2003.
7. L. Cardelli and A.D. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000.

8. T. Chothia and I. Stark. Encoding distributed areas and local communication into the pi-calculus. In *Proceedings of HLCL'00*, volume 52 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2000.
9. C. Ene and T. Muntian. Expressiveness of point-to-point versus broadcast communication. In *Proceedings of FCT'99*, volume 1684 of *LNCS*, pages 258–268. Springer-Verlag, 1999.
10. M. Hennessy and J. Riely. Resource access control in systems of mobile agents. In *Proceedings of HLCL'98*, volume 16.3 of *Electronic Notes in Theoretical Computer Science*, pages 3–17. Elsevier Science Publishers, 1998.
11. K. Honda and M. Tokoro. An object calculus for asynchronous communication. In *Proceedings of ECOOP'91*, volume 512 of *LNCS*, pages 133–147. Springer-Verlag, 1991.
12. K. Honda and N. Yoshida. On Reduction-Based Process Semantics. *Theoretical Computer Science*, 151:437–486, 1995.
13. F. Levi and D. Sangiorgi. Controlling Interference for Ambients. *ACM Transactions on Programming Languages and Systems*, 25(1):1–69, 2003.
14. N.A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
15. R. Milner. *Communicating and Mobile Systems: the π-calculus*. Cambridge University Press, 1999.
16. R. Milner, J. Parrow, and D. Walker. A calculus for mobile processes, parts I and II. *Information and Computation*, 100 (1):1–77, 1992.
17. U. Nestmann. What is a 'Good' Encoding of Guarded Choice? *Journal of Information and Computation*, 156:287–319, 2000.
18. C. Palamidessi. Comparing the Expressive Power of the Synchronous and Asynchronous π-calculus. *Mathematical Structures in Computer Science*, 13(5):685–719, 2003.
19. I.C.C. Phillips and M.G. Vigliotti. On Reduction Semantics for the Push and Pull Ambient Calculus. In *Proceedings of IFIP International Conference on Theoretical Computer Science*, pages 550–562, August 2002.
20. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.
21. M.G. Vigliotti. *Reduction Semantics for Ambient Calculi*. PhD thesis, Imperial College, 2004. Forthcoming.
22. J. Vitek and G. Castagna. Seal: A framework for secure mobile computation. In *Proceedings of Internet Programming Languages*, volume 1686 of *LNCS*, pages 47–77. Springer-Verlag, 1999.
23. P. Zimmer. On the Expressiveness of Pure Safe Ambients. *Mathematical Structures in Computer Science*, 13(5):721–770, 2003.