Leader Election in Rings of Ambient Processes

Iain Phillips and Maria Grazia Vigliotti [1,2]

Department of Computing
Imperial College London

Abstract

Palamidessi has shown that the π-calculus with mixed choice is powerful enough to solve the leader election problem on a symmetric ring of processes. We show that this is also possible in the calculus of Mobile Ambients (MA), without using communication or restriction. Following Palamidessi's methods, we deduce that there is no encoding satisfying certain conditions from MA into CCS. We also show that the calculus of Boxed Ambients is more expressive than its communication-free fragment.

Key words: Ambient calculi, electoral systems, rings, expressiveness

1 Introduction

The π-calculus [12] is a simple, yet extremely powerful, formalism that models concurrency and the passing around of resources that can later be used by other processes. It is based on a very simple and uniform concept of *names*. Names are both *channels*, on which communication takes place, and *values*, i.e. the resources passed around. Names sent as values can be used later as channels for communication. This particular feature seems unique to the π-calculus, and allows processes to establish connection during computation. This seems to add extra power that was not previously available in CCS [11] or other similar calculi such as CSP [8] or ACP [2].

In fact, while CCS (with value-passing) may be regarded as a subcalculus of the π-calculus, Palamidessi [13] has exploited the possibility of creating new connections in the π-calculus by showing that (under certain conditions) there exists no encoding from the π-calculus to CCS. Palamidessi establishes

[1] We wish to thank Catuscia Palamidessi for helpful discussions and the anonymous referees for their helpful comments. M.G. Vigliotti was supported by the EPSRC grant PROFORMA GR/545140.

[2] Email: {iccp,mgv98}@doc.ic.ac.uk

This is the final version. It will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

her result within the framework of the leader election problem. Leader election problems arise in the field of distributed systems, where they are widely studied for practical reasons, and are also used to differentiate models of computation. The problem is stated as follows: given a symmetric network, a composition of processes that differ in their free variables only, one process has to be elected a leader without the help of a centralised server.

CCS and the π-calculus with mixed choice (where input and output can occur in the choice operator together) can both elect a leader in a fully connected symmetric network. This differentiates these two calculi from the π-calculus with separate choice (meaning that inputs and outputs cannot be mixed in the same choice), where such election is not possible. Moreover, the π-calculus can also solve the problem of electing a leader in a symmetric *ring* of processes, in other words, in a network where each process is connected only to its two neighbours in the ring. Palamidessi's algorithm works in two phases. In phase one the processes pass names around the ring so that every process becomes directly connected to every other process. Here there is an essential use of the π-calculus, though without any use of choice. CCS would not do, since it cannot increase connectivity. In the second phase the processes elect a leader. Here there is an essential use of mixed choice, but CCS would suffice, rather than the π-calculus. Building on the work of Angluin [1] and Bougé [4], Palamidessi proves that CCS cannot perform leader election on symmetric rings, by showing that there is a maximal computation where symmetry is never broken, so that no single leader emerges. She deduces that there is no encoding (of a certain kind) from the π-calculus with mixed choice into CCS.

In the present work we explore how Palamidessi's techniques apply to rings in ambient calculi, studying how new connections between processes can be established. In previous work [15], we have shown that in Mobile Ambients (MA) [6] without the communication primitives, the open capability and restriction, the leader election problem can be solved in a fully-connected symmetric network. We might call this fragment of MA the *minimal fragment*. This fragment, which is also a sub-calculus of Boxed Ambients (BA) [5], is choice-free; the solution to electing a leader in symmetric network is achieved through the pre-emptive power of migration inside ambients [15,17]. The communication primitives of MA have the same operational semantics as the π-calculus, except that they are *anonymous*, in the sense that there are no channels on which communication happens (in the π-calculus one would write $a(x).P$ for an input on the channel a, while in MA one would write $(x).P$ for an anonymous input). Thus, since the communication primitives in ambients are very similar to those of the π-calculus, it would be not surprising if Palamidessi's algorithm for rings could be formulated in MA. However, in this paper we solve the leader election problem for symmetric rings in *pure public* MA, i.e. MA without communication primitives and restriction. The link-passing in this case has to be simulated, since there is no explicit way of

passing names in the absence of communication. This yields immediately the result that pure public MA cannot be encoded into CCS.

The second major result that we present here, is that, even if we add communication and restriction to the minimal fragment of MA, the leader election problem in a ring cannot be solved. This clearly shows that in MA, the open capability (but not communication) is crucial in order to pass resources around. In connection with our results, we recall that Zimmer [18] proved that the synchronous choice-free π-calculus can be encoded into pure Safe Ambients (SA) [9], showing that link-passing can be simulated in pure SA. The encoding uses the open capability. Thus the open capability seems quite powerful. This is in contrast with other expressiveness results based on Turing completeness [10,3], where it was shown that the open capability is not crucial, since the minimal fragment is still Turing-complete.

The situation is different for BA, where the open capability is missing as a design choice. Communications between parent and child ambients are allowed, and the synchronous choice-free π-calculus can be encoded, and with that, clearly, the power of creating new links. Thus, in BA the leader election problem in a ring can be solved by converting the ring into a fully-connected network and then using the algorithm of [15]. However, *pure* BA is less expressive than the full calculus, since only with the presence of the open capability can MA elect a leader in rings.

In distributed systems, leader election problems are categorised according to the connectivity of the network, the knowledge of the size of the network and the methods of election. In this paper we present a solution in MA for a ring of four processes (the smallest interesting case). We conjecture that the generalisation of the algorithm to any size of ring should be possible, providing that the processes are given information about the size of the ring. Palamidessi's algorithm also uses this information. However, for the Push and Pull Ambient Calculus [14], we present a solution for rings where the processes do not know the size of the ring. Thus a single uniform solution will work for any size of ring. As far as we know, this is the first time that such an algorithm has been devised in the setting of process calculi. It remains for future work to find suitable conditions that differentiate those calculi that admit a solution to the leader election problem without having to know the size of the ring from those that do need to know the size.

The rest of the paper is structured as follows. In Section 2 we describe the calculi we are considering, and in Section 3 we discuss electoral systems. In Section 4 we consider calculi which admit symmetric electoral systems of rings of processes, while in Section 5 we show that MA without the open capability does not admit symmetric electoral systems for certain rings. In Section 6 we examine the consequences of our results for expressiveness of ambient calculi. Finally we draw some conclusions in Section 7.

2 Calculi

In this section we recall Cardelli and Gordon's Mobile Ambients (MA) and related calculi.

2.1 Mobile Ambients

We follow [6], except for communication, as noted below. Let $P, Q, \ldots$ range over processes and $M, \ldots$ over capabilities. We assume a set of names $\mathcal{N}$, ranged over by $m, n, \ldots$. Processes are defined as follows:

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

Here $\mathbf{0}$ is the nil process which is inactive; $P \mid Q$ is the parallel composition of processes P and Q; $\nu n\, P$ is P with name n restricted; $!\, P$ (replication) is a process which can spin off as many copies of P as are required; $n[\, P \,]$ is an ambient named n containing process P; $M.P$ performs capability M before continuing as P; and $(n).P$ receives input on an anonymous channel, with the input name replacing free occurrences of name n in P; and finally $\langle n \rangle$ is a process which outputs name n. Notice that output is *asynchronous*, that is, it has no continuation. Restriction and input are name-binding. We let $\mathsf{fn}(P)$ denote the set of free names of P. We omit trailing $\mathbf{0}$s and write $n[\,]$ instead of $n[\, \mathbf{0} \,]$.

Capabilities are defined as follows:

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

Capabilities allow movement of ambients ($\mathsf{in}\, n$ and $\mathsf{out}\, n$) and dissolution of ambients ($\mathsf{open}\, n$).

We confine ourselves in this paper to communication of names, rather than full communication including capabilities (as in [6]). This serves to streamline the presentation; the results would also hold for full communication.

Structural congruence $\equiv$ allows rearrangement of processes; it is the least congruence generated by the following laws:

$$\begin{array}{ll}
P \mid Q \equiv Q \mid P & \nu n\, \nu m\, P \equiv \nu m\, \nu n\, P \\
(P \mid P') \mid P'' \equiv P \mid (P' \mid P'') & \nu n\, (P \mid Q) \equiv P \mid \nu n\, Q \ \text{ if } n \notin \mathsf{fn}(P) \\
P \mid \mathbf{0} \equiv P & \nu n\, m[\, P \,] \equiv m[\, \nu n\, P \,] \ \text{ if } n \neq m \\
!\, P \equiv P \mid !\, P & \nu n\, \mathbf{0} \equiv \mathbf{0} \\
!\, \mathbf{0} \equiv \mathbf{0} &
\end{array}$$

The *reduction* relation $\rightarrow$ is generated by the following rules:

$$\begin{array}{ll}
\text{(Open)} & \mathsf{open}\, n.P \mid n[\, Q \,] \rightarrow P \mid Q \\
\text{(In)} & n[\, \mathsf{in}\, m.P \mid P' \,] \mid m[\, Q \,] \rightarrow m[\, n[\, P \mid P' \,] \mid Q \,] \\
\text{(Out)} & m[\, n[\, \mathsf{out}\, m.P \mid P' \,] \mid Q \,] \rightarrow n[\, P \mid P' \,] \mid m[\, Q \,] \\
\text{(Comm)} & \langle m' \rangle \mid (m).P \rightarrow P\{m'/m\}
\end{array}$$

$$\text{(Amb)}\ \frac{P \to P'}{n[\,P\,] \to n[\,P'\,]} \qquad \text{(Par)}\ \frac{P \to P'}{P \mid Q \to P' \mid Q}$$

$$\text{(Res)}\ \frac{P \to P'}{\nu n\, P \to \nu n\, P'} \qquad \text{(Str)}\ \frac{P \equiv P' \quad P' \to Q' \quad Q' \equiv Q}{P \to Q}$$

Here $P\{m'/m\}$ denotes P with m' substituted for every free occurrence of m. Notice that movement in MA is *subjective*: ambients move themselves using the in and out capabilities. We write $\twoheadrightarrow$ for the reflexive and transitive closure of $\to$.

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 $P \equiv \nu\vec{m}\,(n[\,P'\,] \mid P'')$ with $n \notin \vec{m}$. Here $\vec{m}$ represents a tuple of names. A process P *eventually exhibits barb* n, written $P \Downarrow n$, iff $P \twoheadrightarrow Q$ and $Q \downarrow n$ for some Q.

We shall be interested in various subcalculi: *pure* MA is MA without communication; *public* MA is MA without restriction; and *boxed* MA is MA without the open capability.

2.2 The Push and Pull Ambient Calculus

The Push and Pull Ambient Calculus (PAC) [14,17] is a variant of MA where the subjective moves enabled by the in and out capabilities are replaced by objective moves whereby ambients can be pulled in or pushed out by other ambients. The syntax of processes is the same as for MA. Capabilities are defined as follows:

$$M ::= \mathsf{pull}\, n \mid \mathsf{push}\, n \mid \mathsf{open}\, n$$

The reduction rules are the same as for MA, except that (In) and (Out) are replaced by the following:

$$\text{(Pull)}\quad n[\,\mathsf{pull}\, m.P \mid P'\,] \mid m[\,Q\,] \to n[\,P \mid P' \mid m[\,Q\,]\,]$$

$$\text{(Push)}\ n[\,m[\,P\,] \mid \mathsf{push}\, m.Q \mid Q'\,] \to n[\,Q \mid Q'\,] \mid m[\,P\,]$$

2.3 Boxed Ambients

The calculus of Boxed Ambients [5] is derived from MA by removing the open capability and allowing parent-child communication as well as same-level communication. Processes are defined as follows:

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

Here $\vec{n}$ denotes a tuple of names, and η ranges over *locations*, defined as follows:

$$\eta ::= n \mid \uparrow \mid \star$$

The "local" location $\star$ is elided. Notice that output $\langle\vec{n}\rangle^{\eta}.P$ is synchronous, unlike in MA. Capabilities M are defined as for MA but without open. The reduction rules are the same as for boxed MA, except for communication,

where the rule (Comm) is replaced by the following five rules:

$$\begin{array}{ll}
\text{(Local)} & (\vec{m}).P \mid \langle \vec{m'} \rangle.Q \to P\{\vec{m'}/\vec{m}\} \mid Q \\
\text{(Input } n\text{)} & (\vec{m})^n.P \mid n[\, \langle \vec{m'} \rangle.Q \mid Q' \,] \to P\{\vec{m'}/\vec{m}\} \mid n[\, Q \mid Q' \,] \\
\text{(Input } \uparrow\text{)} & n[\, (\vec{m})^{\uparrow}.P \mid P' \,] \mid \langle \vec{m'} \rangle.Q \to n[\, P\{\vec{m'}/\vec{m}\} \mid P' \,] \mid Q \\
\text{(Output } n\text{)} & n[\, (\vec{m}).P \mid P' \,] \mid \langle \vec{m'} \rangle^n.Q \to n[\, P\{\vec{m'}/\vec{m}\} \mid P' \,] \mid Q \\
\text{(Output } \uparrow\text{)} & (\vec{m}).P \mid n[\, \langle \vec{m'} \rangle^{\uparrow}.Q \mid Q' \,] \to P\{\vec{m'}/\vec{m}\} \mid n[\, Q \mid Q' \,]
\end{array}$$

Clearly, rule (Local) extends rule (Comm), so that communication in BA is at least as powerful as communication in MA.

3 Electoral Systems and Rings

All the notions of this section apply equally to MA, PAC and BA.

3.1 Networks and Electoral Systems

We briefly recall electoral systems as formulated in [15], building on [13]. 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. The notation $P \not\to$ means that there does not exist a process to which P can reduce.

Definition 3.1 Let P be a process. A *computation* $\mathcal{C}$ of P is a (finite or infinite) sequence $P = P_0 \to P_1 \to \cdots$. It is *maximal* if it cannot be extended, i.e. either $\mathcal{C}$ is infinite, or else it is of the form $P_0 \to \cdots \to P_h$ where $P_h \not\to$.

Definition 3.2 Let $\mathcal{C}$ be a computation $P_0 \to \cdots \to P_h \to \cdots$. We define the *observables* of $\mathcal{C}$ as $\mathsf{Obs}(\mathcal{C}) = \{\omega \in \mathsf{Obs} : \ \exists h \ \ P_h \downarrow \omega\}$.

Networks are just collections of processes running in parallel:

Definition 3.3 A *network* Net of size k is a process in the form $\nu\vec{n}\,(P_0 \mid \ldots \mid P_{k-1})$.

A *permutation* is a bijection $\sigma : \mathcal{N} \to \mathcal{N}$ 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 3.4 Let $\mathsf{Net} = \nu\vec{n}(P_0 \mid \ldots \mid P_{k-1})$ 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. $n \in \vec{n}$ iff $\sigma(n) \in \vec{n}$.

Definition 3.5 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 $\hat{\sigma}^h(i) = i$. If every orbit has the same size then σ is *well-balanced*.

Definition 3.6 Let $\mathsf{Net} = \nu\vec{n}\,(P_0 \mid \ldots \mid P_{k-1})$ 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 $P_{\hat{\sigma}(i)} = \sigma(P_i)$.

Intuitively an electoral system is a network which reports a unique winner, no matter how the computation proceeds.

Definition 3.7 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 $\mathsf{Obs}(\mathcal{C}) = \{\omega_i\}$.

3.2 Rings and Independence

In this paper we are interested in the connectivity between processes, and in rings of processes in particular. Given a network $\mathsf{Net} = \nu\vec{n}\,(P_0 \mid \cdots \mid P_{k-1})$, we can associate a graph with Net by letting the set of nodes be $\{0, \ldots, k-1\}$ and letting $i, j < k$ be adjacent iff $\mathsf{fn}(P_i) \cap \mathsf{fn}(P_j) \neq \emptyset$. A network forms a ring if the processes can be arranged in a cycle, and each node i is adjacent to at most its two neighbours in the cycle.

Definition 3.8 A *ring* is a network $\mathsf{Net} = \nu\vec{n}\,(P_0 \mid \cdots \mid P_{k-1})$ which has a single-orbit automorphism σ such that for all $i, j < k$, if $\mathsf{fn}(P_i) \cap \mathsf{fn}(P_j) \neq \emptyset$ then one of $i = j$, $\hat{\sigma}(i) = j$ or $\hat{\sigma}(j) = i$ must hold. A ring is *symmetric* if it is symmetric with respect to such an automorphism σ.

Recall that an *independent set* in a graph is a set of nodes such that no two nodes of the set are adjacent.

Definition 3.9 Two processes P and Q are *independent* if they do not share any free names: $\mathsf{fn}(P) \cap \mathsf{fn}(Q) = \emptyset$.

Definition 3.10 Let σ be an automorphism on a network $\mathsf{Net} = \nu\vec{n}\,(P_0 \mid \cdots \mid P_{k-1})$. Then Net is *independent* with respect to σ if every orbit forms an independent set, in the sense that if $i, j < k$ are in the same orbit of $\hat{\sigma}$ with $i \neq j$, then P_i and P_j are independent.

4 Calculi with Electoral Systems for Rings

In this section we show that we can solve leader election on symmetric rings in ambient calculi. We present solutions for MA, BA and PAC. We start with PAC, since it is the simplest.

4.1 Pure Public PAC

We show that using push and pull we can build a symmetric ring of processes which can elect a leader. What is more, the construction is such that individual processes do not know the size of the ring.

Theorem 4.1 *For any $k \geq 1$, there is a symmetric ring of size k which is an electoral system in pure public PAC.*

Proof. (Sketch) Let $k \geq 1$. For $i = 0, \ldots, k-1$, let P_i be defined as follows:

$$P_i \stackrel{\text{df}}{=} n_i[\, n_i[\, \omega_i[\,]\,] \mid \mathsf{push}\ \omega_i \mid \mathsf{pull}\ n_{i+1} \mid \mathsf{open}\ n_{i+1}\,]$$

Let $\mathsf{Net} \stackrel{\text{df}}{=} P_0 \mid \cdots \mid P_{k-1}$. Note that Net belongs to pure public PAC—there is no use of communication or restriction. Moreover, the construction of P_i does not depend on k. It can be checked that Net forms a symmetric ring.

We claim that Net forms an electoral system. The idea is that a process can pull in its neighbour and open it. The neighbour thereby loses and drops out of the ring, which now has one fewer process. Eventually only one process P_i' is left, which has the capability to open $n_i[\, \omega_i[\,]\,]$ and push $\omega_i[\,]$ to the top level, announcing i as the winner. More details can be found in [16]. □

We do not see how to express this algorithm using the different movement capabilities available in MA, or in Safe Ambients [9], or in ROAM [7].

4.2 Pure Public MA

We now solve the leader election problem for rings in pure public MA. We restrict ourselves to a ring of size four, which is the smallest interesting case and will be enough for establishing separation results between calculi (Section 6).

Theorem 4.2 *There is a symmetric ring of size four which is an electoral system in pure public MA.*

Proof. (Sketch) Unlike in the case of PAC, where during the computation the ring contracts as each losing process gets eliminated, we follow Palamidessi's solution for the π-calculus with mixed choice by first converting the ring into a complete graph and then running an election on this graph. We write $n^r[\, P\,]$ as a shorthand for $n[\, n[\, \ldots n[\, P\,] \ldots]\,]$ (r embedded ambients named n, with P as the contents of the innermost ambient). We define:

$$\begin{aligned} P_i \stackrel{\text{df}}{=}\ & r_i[\, n_i[\, \omega_i^4[\, \mathsf{out}\ n_i\,] \mid Q_{i,i+1} \mid \mathsf{open}\ e_i.s_i[\, \mathsf{out}\ n_i.\mathsf{out}\ r_i\,]\,]\,] \\ & \mid \mathsf{open}\ s_i.\mathsf{open}\ r_i \mid b_{i+1}[\, \mathsf{in}\ d_{i+1}.c_{i+1}[\, Q_i\,]\,] \mid d_i[\, \mathsf{open}\ b_i.\mathsf{in}\ e_{i+1}\,] \\ & \mid e_{i+1}[\, \mathsf{open}\ d_i.\mathsf{in}\ r_{i+1}.\mathsf{open}\ c_i.\mathsf{in}\ n_{i+1}.(Q_i' \mid Q_{i+1,i})\,] \end{aligned}$$

where

$$Q_{ij} \stackrel{\text{df}}{=} \mathsf{in}\ n_j \mid \mathsf{open}\ n_j.\mathsf{dm}_j[\ ] \mid \mathsf{open}\ \mathsf{dm}_j.(\mathsf{dm}_j[\ ] \mid \mathsf{open}\ \omega_i)$$
$$Q_i \stackrel{\text{df}}{=} \mathsf{in}\ n_i \mid \mathsf{open}\ n_i.\mathsf{dm}_i[\ ] \mid \mathsf{open}\ \mathsf{dm}_i.(\mathsf{dm}_i[\ ] \mid a_{i+1}[\ ])$$
$$Q'_i \stackrel{\text{df}}{=} \mathsf{open}\ a_i.\mathsf{open}\ \omega_{i+1}$$

Then $P_0 \mid P_1 \mid P_2 \mid P_3$ forms a ring, since $\mathsf{fn}(P_i) \cap \mathsf{fn}(P_{i+2}) = \emptyset$. It can be shown that $P_0 \mid P_1 \mid P_2 \mid P_3$ is guaranteed to reduce to the network $R_0 \mid R_1 \mid R_2 \mid R_3$, where:

$$R_i \stackrel{\text{df}}{=} n_i[\,\omega_i^4[\,\mathsf{out}\ n_i\,] \mid Q_{i,i+1} \mid Q_{i-2} \mid Q'_{i-1} \mid Q_{i,i-1}\,]$$

This network forms a complete symmetric graph, and is an electoral system. The idea is that a process j loses by entering another process i. At this point ambient n_j is opened, unleashing the "dummy" ambient dm_j within ambient n_i. The winner will be the single process i that has absorbed all the other processes, and has been able to open dm_j (all $j \neq i$) and thereby strip off the three outer ω_i ambients to allow $\omega_i[\ ]$ to emerge at the top level. More details can be found in [16]. □

We conjecture that the construction in the proof of Theorem 4.2 can be generalised to build symmetric rings forming electoral systems of any size k.

4.3 Boxed Ambients

We now consider Boxed Ambients. The solutions for MA and PAC depend on open, which is not available in BA. However it turns out that the parent-child communication of BA enables the construction of symmetric rings forming electoral systems.

Theorem 4.3 *For any $k \geq 1$, there is a symmetric ring of size k which is an electoral system in BA.*

Proof. (Sketch) As in the proof of Theorem 4.2, we follow Palamidessi's method of first distributing names round the ring to create a complete graph and then running the election on it. Palamidessi shows how to distribute the names in choice-free synchronous π-calculus. Suppose that process P_i has a channel y_i initially known only to itself, and is joined to P_{i+1} by channel x_i. Then the names y_i are passed around the ring so that all processes share them and can use them in the election phase. Since BA can encode choice-free synchronous π-calculus [5], we can carry out the distribution phase in BA. We use the following translation of the π-calculus input and synchronous output:

$$x(y).P \stackrel{\text{df}}{=} (y,z)^x.(z[\ ] \mid P)$$
$$\bar{x}\langle y\rangle.P \stackrel{\text{df}}{=} x[\,\langle y,z\rangle\,] \mid ()^z.P$$

where z is fresh. Note that we do not need restriction.

The algorithm for the election phase is the same as the one presented in [15] for pure public boxed MA.

Here is what P_i looks like for $k = 4$:

$$P_i(x_i, x_{i+3}, y_i) \stackrel{\text{df}}{=} \bar{x}_i\langle y_i\rangle.x_{i+3}(y_{i+3}).\bar{x}_i\langle y_{i+3}\rangle.x_{i+3}(y_{i+2}).\bar{x}_i\langle y_{i+2}\rangle.x_{i+3}(y_{i+1}).$$
$$Q_i\langle y_i, y_{i+1}, y_{i+2}, y_{i+3}\rangle$$

Here Q_i is ready to carry out the election:

$$\begin{aligned}
Q_i(y_i, y_{i+1}, y_{i+2}, y_{i+3}) \stackrel{\text{df}}{=} n_i[\,&\mathsf{in}\ y_{i+1} \mid \mathsf{in}\ y_{i+2} \mid \mathsf{in}\ y_{i+3} \mid \\
&\omega_i[\,\mathsf{in}\ y_{i+1}.\mathsf{in}\ y_{i+2}.\mathsf{in}\ y_{i+3}.\mathsf{out}\ y_{i+3}.\mathsf{out}\ y_{i+2}.\mathsf{out}\ y_{i+1}.\mathsf{out}\ n_i\,] \mid \\
&\omega_i[\,\mathsf{in}\ y_{i+1}.\mathsf{in}\ y_{i+3}.\mathsf{in}\ y_{i+2}.\mathsf{out}\ y_{i+2}.\mathsf{out}\ y_{i+3}.\mathsf{out}\ y_{i+1}.\mathsf{out}\ n_i\,] \mid \\
&\omega_i[\,\mathsf{in}\ y_{i+2}.\mathsf{in}\ y_{i+1}.\mathsf{in}\ y_{i+3}.\mathsf{out}\ y_{i+3}.\mathsf{out}\ y_{i+1}.\mathsf{out}\ y_{i+2}.\mathsf{out}\ n_i\,] \mid \\
&\omega_i[\,\mathsf{in}\ y_{i+3}.\mathsf{in}\ y_{i+1}.\mathsf{in}\ y_{i+2}.\mathsf{out}\ y_{i+2}.\mathsf{out}\ y_{i+1}.\mathsf{out}\ y_{i+3}.\mathsf{out}\ n_i\,] \mid \\
&\omega_i[\,\mathsf{in}\ y_{i+3}.\mathsf{in}\ y_{i+2}.\mathsf{in}\ y_{i+1}.\mathsf{out}\ y_{i+1}.\mathsf{out}\ y_{i+2}.\mathsf{out}\ y_{i+3}.\mathsf{out}\ n_i\,] \mid \\
&\omega_i[\,\mathsf{in}\ y_{i+2}.\mathsf{in}\ y_{i+3}.\mathsf{in}\ y_{i+1}.\mathsf{out}\ y_{i+1}.\mathsf{out}\ y_{i+3}.\mathsf{out}\ y_{i+2}.\mathsf{out}\ n_i\,]\,]
\end{aligned}$$

□

5 Calculi without Electoral Systems for Rings

In this section, we show that the open capability is crucial for electing a leader in symmetric networks. If fact, if the open capability is dropped, then election in symmetric rings is not possible. We present below the proof for MA; however our technique can be easily adapted to PAC and SA. Thus, PAC and SA do not admit a solution for electoral system for rings without the open capability.

Recall that by *boxed* MA we mean MA without the open capability.

Theorem 5.1 *For any composite $k > 1$, boxed MA does not have a symmetric ring of size k with no globally-bound names which is an electoral system.*

For the proof, of which full details can be found in [16], we consider a network Net symmetric with respect to an automorphism σ with independent orbits. Whatever reduction Net makes, we can retain symmetry and independence with respect to σ by propagating that move round the orbit(s) concerned. If a process ever declares itself a winner, then by symmetry all processes in the same orbit can declare themselves winners on the same round. With orbits of size greater than 2 this means that there is a computation of Net which does not declare a unique winner, so that Net is not an electoral system. The significance of Net being independent with respect to σ is that if a reduction involves two processes interacting then they must come from different orbits. This means that when the reduction is propagated round the two orbits concerned we have restored symmetry with respect to σ. So far, the method we have outlined essentially follows Palamidessi's proof for CCS [13, Theorem 6.1], which builds on the work of Angluin [1] and Bougé [4]. How-

ever, that proof relies on the orbits of σ remaining independent throughout the computation. This may not be the case for boxed MA, since processes can acquire new free names through communication and through other ambients entering. We are therefore obliged to weaken the notion of independence for the proof. We label processes to keep track of which ambients truly belong to a process, and which ambients have entered from another process. "Foreign" ambients can move around, but they can never transfer their capabilities to the host process, since the open capability is not available. Thus, independence and symmetry with respect to σ is preserved during computation.

The condition in Theorem 5.1 requiring a ring of composite size is stronger than the condition used by Palamidessi for CCS, which is that the network has a well-balanced automorphism with independent orbits. We need the single orbit of the ring automorphism in order to deal with anonymous communication.

6 Separation Results

We use the results of Sections 4 and 5 to show that certain languages cannot be encoded in others.

We recall the following from [15] (building on [13]):

Definition 6.1 Let L, L' be process languages. An encoding $[\![-]\!] : L \to L'$ is

(i) *distribution-preserving* if for all processes P, Q of L, $[\![P \mid Q]\!] = [\![P]\!] \mid [\![Q]\!]$;

(ii) *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 $\sigma(\omega_i) = \theta(\omega_i)$;

(iii) *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*.

Unlike in [15], we are considering encodings which map rings to rings. We therefore need a further property:

Definition 6.2 An encoding is *independence-preserving* if for any processes P, Q, if P and Q are independent then $[\![P]\!]$ and $[\![Q]\!]$ are also independent.

Palamidessi says that such an encoding "does not increase the level of connectivity of the network". Not all encodings preserve independence. For instance, Zimmer's [18] encoding of the synchronous π-calculus without choice into pure Safe Ambients [9] introduces a new global ambient whose name is shared by all processes.

Lemma 6.3 *Suppose* $[\![-]\!] : L \to L'$ *is a uniform, observation-respecting and independence-preserving encoding. Suppose that* Net *is a symmetric ring of size* $k \geq 1$ *with no globally-bound names which is an electoral system. Then* $[\![\mathsf{Net}]\!]$ *is also a symmetric ring of size* k *with no globally-bound names which is an electoral system.*

The proof can be found in [16].

Theorem 6.4 *There is no uniform, observation-respecting and independence-preserving encoding from pure public MA into CCS (with value passing).*

Proof. This follows from Theorem 4.2, Lemma 6.3 and the fact that CCS does not have a symmetric ring of four processes which is an electoral system [13]. Although CCS usually uses labelled transition systems, it can be regarded as a subcalculus of the π-calculus and therefore can be fitted into the unlabelled approach to electoral systems of Section 3, as was done for the π-calculus in [15]. □

Theorem 6.5 *There is no uniform, observation-respecting and independence-preserving encoding from pure public MA into boxed MA.*

Proof. From Theorem 4.2, Theorem 5.1 and Lemma 6.3. □

It follows from Theorem 6.5 that the open capability of MA does indeed add expressive power not present in the other operators of MA.

Theorem 6.6 *There is no uniform, observation-respecting and independence-preserving encoding from BA into boxed MA (and therefore into pure BA).*

Proof. From Theorem 4.3, Theorem 5.1 and Lemma 6.3. □

It follows from Theorem 6.6 that the parent-child communication in BA does indeed add expressive power (without it, BA would be essentially boxed MA).

7 Conclusions

In this paper we have shown how to elect a leader in a symmetric ring of processes in MA and its variants. We have seen that it can be done in pure public MA for a ring of size 4, so that for that case communication is unnecessary. On the other hand, the open capability is essential, since the election cannot be carried out in boxed MA (in fact the in and out capabilities are also essential—cf. [15]). Thus, simulating link-passing requires the open capability, but does not require the (anonymous) communication of MA. This shows that pure MA cannot be encoded either into CCS or into pure BA. In the case of BA, however, (parent-child) communication *is* necessary in order to elect a leader in rings, since the open capability is not present. While our results shed light on the expressive power provided by the open capability,

in the presence of the latter, leader election problems in both rings and fully connected graphs do not give any separation results between MA with communication primitives and pure MA. In this framework one could regard them as equal, since, when it comes to passing names around, pure MA can do just as well as the full calculus.

It is worth spending a few words on Theorem 5.1, which says that MA without the open capability cannot solve the election problem on rings with a composite number of processes. If the number of processes is prime, then any well-balanced automorphism different from the identity has one orbit only, and our proof methods would not apply. This would be true for Palamidessi's work as well. Nevertheless, we expect that election is impossible in rings of any size greater than three. Furthermore, we claim that Theorem 5.1 also holds for PAC (or SA) without the open capability. In connection with this, and recalling that Zimmer has encoded the synchronous choice-free π-calculus into pure SA, we conjecture that for SA without the open capability such an encoding would not be possible, even in the presence of communication. For if it were possible, then it would seem that SA without open *could* perform election on rings, much as shown for BA (Theorem 4.3).

A challenge for the future is to find suitable conditions that differentiate those calculi that admit a solution to the leader election problem without having to know the size of the ring (such as PAC) from those that do need to know the size.

References

[1] Angluin, D., *Local and global properties in networks of processors*, in: *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, 1980, pp. 82–93.

[2] Bergstra, J. and J. Klop, *Process algebra for synchronous communication*, Information and Control **60** (1984), pp. 109–137.

[3] Boneva, I. and J.-M. Talbot, *When ambients cannot be opened*, in: *Proceedings of 6th International Conference on Foundations of Software Science and Computation Structures, FoSSaCS 2003*, Lecture Notes in Computer Science **2620** (2003), pp. 169–184, full version to appear in Theoretical Computer Science.

[4] Bougé, L., *On the existence of symmetric algorithms to find leaders in networks of communicating sequential processes*, Acta Informatica **25** (1988), pp. 179–201.

[5] 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.

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

[7] Guan, X., Y. Yang and J. You, *Making ambients more robust*, in: *Proceedings of International Conference on Software: Theory and Practice, Beijing, China, August 2000*, 2000, pp. 377–384.

[8] Hoare, C., "Communicating Sequential Processes," Prentice-Hall, 1985.

[9] Levi, F. and D. Sangiorgi, *Mobile safe ambients*, ACM Transactions on Programming Languages and Systems **25** (2003), pp. 1–69.

[10] Maffeis, S. and I. Phillips, *On the computational strength of pure ambient calculi*, Theoretical Computer Science (2005), accepted.

[11] Milner, R., "Communication and Concurrency," Prentice-Hall, 1989.

[12] Milner, R., J. Parrow and D. Walker, *A calculus of mobile processes*, Information and Computation **100** (1992), pp. 1–77.

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

[14] Phillips, I. and M. Vigliotti, *On reduction semantics for the push and pull ambient calculus*, in: *Proceedings of IFIP International Conference on Theoretical Computer Science (TCS 2002), IFIP 17th World Computer Congress, August 2002, Montreal* (2002), pp. 550–562.

[15] Phillips, I. 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.

[16] Phillips, I. and M. Vigliotti, *Leader election in rings of ambient processes*, Technical report, Department of Computing, Imperial College London (2004).

[17] Vigliotti, M., "Reduction semantics for ambient calculi," Ph.D. thesis, Imperial College, University of London (2004).

[18] Zimmer, P., *On the expressiveness of pure safe ambients*, Mathematical Structures in Computer Science **13** (2003), pp. 721–770.