A Ground-Complete Axiomatization of Stateless Bisimilarity over Linda[☆]

Luca Aceto[a,b,*], Anna Ingolfsdottir[a], Eugen-Ioan Goriac[a]

[a] *ICE-TCS, School of Computer Science, Reykjavik University, Menntavegur 1, IS 101 Reykjavik, Iceland*
[b] *SysMA, IMT Lucca Institute for Advanced Studies, Lucca 55100, Italy*

Abstract

This paper offers a finite, ground-complete axiomatization of stateless bisimilarity over the tuple-space-based coordination language Linda. As stepping stones towards that result, axiomatizations of stateless bisimilarity over the sequential fragment of Linda without the *nask* primitive, and over the full sequential sublanguage are given. It is also shown that stateless bisimilarity coincides with standard bisimilarity over the sequential fragment of Linda without the *nask* primitive.

Keywords: Concurrency, process algebra, stateless bisimilarity, Linda, equational logic

1. Introduction

The goal of this paper is to contribute to the study of equational axiomatizations of behavioural equivalences for processes with data—see, e.g., the references [10, 14, 15, 16] for earlier contributions to this line of research. Specifically, we present a ground-complete axiomatization of stateless bisimilarity from [5, 8, 13, 19] over the well-known, tuple-space-based coordination language Linda [7, 11].

Linda is a, by now classic, example from a family of coordination languages that focus on the explicit control of interactions between parallel processes. A communication between Linda processes takes place by accessing tuples in a shared memory, called the *tuple space*, which is a multiset of tuples. The communication mechanism in Linda is asynchronous, in that send operations

[☆] The authors have been partially supported by the project 'Meta-theory of Algebraic Process Theories' (nr. 100014021) of the Icelandic Research Fund. Eugen-Ioan Goriac is also funded by the project 'Extending and Axiomatizing Structural Operational Semantics: Theory and Tools' (nr. 1102940061) of the Icelandic Research Fund.

[*] Corresponding author
Email addresses: `luca@ru.is` (Luca Aceto), `annai@ru.is` (Anna Ingolfsdottir), `eugen.goriac@me.com` (Eugen-Ioan Goriac)

are non-blocking. Our presentation of the syntax and the semantics of Linda follows those given in [6, 19].

In the light of its intuitive appeal and impact, Linda has received a fair amount of attention within the concurrency theory community. For instance, the relative expressive power of fragments of Linda has been studied in [6] and the paper [10] studies testing semantics, in the sense of De Nicola and Hennessy [9], over applicative and imperative process algebras that are inspired by Linda. The paper [9] also provides complete inequational axiomatizations of the studied calculi with respect to testing semantics.

Testing semantics can be viewed as the most natural notion of behavioural equivalence for a language from the programmer's perspective. Indeed, it is the formalization of the motto that 'two program fragments should be considered equivalent unless there is a context/test that tells them apart.' Testing semantics is, however, not very robust. In particular, if one extends a language with new features that increase the observational power of tests, the resulting notion of 'testing equivalence' for the extended language will be finer than the one for the original language. This means that the results one had worked hard to establish for the original language will have to be established anew.

Stateless bisimilarity [5, 8, 13, 19] is a variation on the classic notion of bisimilarity [17, 20] that is suitable for reasoning compositionally about open, concurrent and state-bearing systems. It is the finest notion of bisimilarity for state-bearing processes that one can find in the literature and comes equipped with a congruence rule format for operational rules [19]. It is therefore interesting to study its equational theory in the setting of a seminal language like Linda, not least because equational axiomatizations of stateless bisimilarity may form the core of axiom systems for coarser notions of equivalence over that language.

The main contribution of this paper is a ground-completeness result for stateless bisimilarity over Linda given in Section 3. We first present a complete axiom system for stateless bisimilarity over the sequential fragment of Linda without the *nask* primitive, which tests for the *absence* of a tuple in the tuple space (Theorem 3.2). Interestingly, it turns out that stateless bisimilarity over this fragment of Linda has the same axiomatization of standard bisimilarity, when the considered sub-language of Linda is viewed as Basic Process Algebra (BPA) with deadlock and the empty process [21]. We formalize the connection between the two languages and their respective semantics, culminating in Theorem 3.4.

Next we offer a ground-complete axiomatization of stateless bisimilarity over the full sequential fragment of Linda (Section 3.1). In this setting, we have to deal with the subtle interplay of *ask* and *nask* primitives, which test for the presence and absence of some tuple in the tuple space, respectively. In Theorem 3.5, we show that two equation schemas are enough to capture equationally the effect that combinations of *ask* and *nask* primitives may have on the behaviour of Linda terms.

Following rather standard lines, we give a ground-complete axiomatization of stateless bisimilarity over the full Linda language we consider in this paper in Section 3.2.

We end the paper with some concluding remarks and a suggestion for future research (Section 4).

2. Preliminaries

In this section we present the syntax and operational semantics for the classic, tuple-space-based coordination language Linda [7, 11]. (Our presentation follows those given in [6, 19].) Moreover, we introduce the notion of stateless bisimilarity and the basic definitions from equational logic used in this paper.

Linda's signature Σ_D for data (the so-called tuple space) consists of the constant $\emptyset$ for the empty tuple space, a (possibly infinite) set $\mathcal{U}$ of constants standing for memory tuples and a binary separator $\_\ \_$ that is associative and commutative, but not idempotent, and has $\emptyset$ as left and right unit. (The store is a multiset of tuples.) The set $T(\Sigma_D)$ of closed *data terms* is given, therefore, by the following BNF grammar:

$$d ::= \emptyset \mid u \mid d\ d,$$

where $u \in \mathcal{U}$. Each data term d determines a multiset $\{u_1, \ldots, u_k\}$ of tuples in the obvious way. In what follows, we write $u \in d$ when there is at least one occurrence of the tuple u in the multiset denoted by d.

Following [6], the signature Σ_P for Linda is implicitly given by the BNF grammar defining the set $\mathbb{T}(\Sigma_P)$ of open *process terms* over a countably infinite set V_P of *process variables*:

$$t ::= x \mid \delta \mid \varepsilon \mid \mathit{ask}(u) \mid \mathit{nask}(u) \mid \mathit{tell}(u) \mid \mathit{get}(u) \mid t + t \mid t;t \mid t \parallel t,$$

where $x \in V_P$ and $u \in \mathcal{U}$. Closed terms are terms without occurrences of variables. The set of closed process terms is denoted by $T(\Sigma_P)$. A substitution σ is a function of type $V_P \to \mathbb{T}(\Sigma_P)$. A closed substitution is a substitution whose range is included in $T(\Sigma_P)$. We write $\sigma(t)$ for the term resulting by replacing each occurrence of a variable x in t with the term $\sigma(x)$. Note that $\sigma(t)$ is a closed term whenever σ is a closed substitution.

Intuitively, δ is a constant process that symbolizes deadlock, which satisfies no predicates and performs no actions. The constant ε denotes a process that satisfies the successful termination predicate, denoted by $\downarrow$ in what follows, and performs no action. The constants *ask*, *nask*, *tell*, and *get* are the basic Linda instructions for operating with the data component. $\mathit{ask}(u)$ and $\mathit{nask}(u)$ check whether tuple u is and, respectively, is not in the store. $\mathit{tell}(u)$ adds tuple u to the store, while $\mathit{get}(u)$ removes one of its occurrences if it is present. The $\mathit{ask}(u)$, $\mathit{get}(u)$ and $\mathit{nask}(u)$ operations are blocking, in the sense that a process executing them blocks if u is not in the tuple space for *ask* and *get*, and if it is in the tuple space for *nask*. The operations $\_+\_$, $\_;\_$ and $\_\|\_$ are, respectively, the standard alternative, sequential and interleaving parallel composition operations familiar from process algebras—see, for instance, [2].

Definition 2.1 (Transition System Specification for Linda)**.** *The operational semantics of Linda is given in terms of a unary immediate termination predicate* $\downarrow$ *and a binary transition relation* $\rightarrow$ *over configurations of the form* (p, d)*, with* $p \in T(\Sigma_P)$ *and* $d \in T(\Sigma_D)$*. Intuitively,* $(p, d) \downarrow$ *means that the process term* p *can terminate immediately in the context of the tuple space* d*, whereas*

$$(p, d) \rightarrow (p', d')$$

indicates that the configuration (p, d) *can evolve into* (p', d') *in one computational step. Formally,* $\downarrow$ *and* $\rightarrow$ *are the least relations over configurations satisfying the following set of rules.*

$$\frac{}{(\varepsilon, d)\downarrow} \qquad \frac{(x,d)\downarrow}{(x+y,d)\downarrow} \qquad \frac{(y,d)\downarrow}{(x+y,d)\downarrow} \qquad \frac{(x,d)\downarrow \quad (y,d)\downarrow}{(x\,;y,d)\downarrow} \qquad \frac{(x,d)\downarrow \quad (y,d)\downarrow}{(x \parallel y,d)\downarrow}$$

$$\frac{}{(ask(u), d\ u) \rightarrow (\varepsilon, d\ u)} \qquad \frac{}{(tell(u), d) \rightarrow (\varepsilon, d\ u)}$$

$$\frac{}{(get(u), d\ u) \rightarrow (\varepsilon, d)} \qquad \frac{}{(nask(u), d) \rightarrow (\varepsilon, d)}\ [u \notin d]$$

$$\frac{(x,d) \rightarrow (x',d')}{(x+y,d) \rightarrow (x',d')} \qquad \frac{(y,d) \rightarrow (y',d')}{(x+y,d) \rightarrow (y',d')}$$

$$\frac{(x,d) \rightarrow (x',d')}{(x\,;y,d) \rightarrow (x'\,;y,d')} \qquad \frac{(x,d)\downarrow \quad (y,d) \rightarrow (y',d')}{(x\,;y,d) \rightarrow (y',d')}$$

$$\frac{(x,d) \rightarrow (x',d')}{(x \parallel y,d) \rightarrow (x' \parallel y,d')} \qquad \frac{(y,d) \rightarrow (y',d')}{(x \parallel y,d) \rightarrow (x \parallel y',d')}$$

Note that the predicate $\downarrow$ is independent of the data component in a configuration, that is, if $(p, d) \downarrow$ for some $p \in T(\Sigma_P)$ and $d \in T(\Sigma_D)$, then $(p, d') \downarrow$ also holds for each $d' \in T(\Sigma_D)$.

Throughout the paper we use the notion of *stateless bisimilarity* from [5, 8, 13, 19] as our notion of behavioural equivalence over closed Linda process terms. Stateless bisimilarity is the finest notion of bisimilarity for state-bearing processes that one can find in the literature. It is a variation on strong bisimilarity for processes with data in which the behaviour of process terms is compared in the context of all possible data terms, and that allows for interference from 'the environment' in the data part after each transition. This makes stateless bisimilarity suitable for reasoning compositionally about open concurrent systems.

Definition 2.2 (Stateless Bisimilarity)**.** *A relation* $R \subseteq T(\Sigma_P) \times T(\Sigma_P)$ *is a* stateless bisimulation *if, and only if, it is symmetric and the following conditions hold for each* $(p, q) \in R$*:*

- *for all* $p' \in T(\Sigma_P)$ *and* $d \in T(\Sigma_D)$*, if* $(p, d) \rightarrow (p', d')$ *then there is some* $q' \in T(\Sigma_P)$ *such that* $(q, d) \rightarrow (q', d')$ *and* $(p', q') \in R$*;*

- *for each $d \in T(\Sigma_D)$, if $(p,d)\downarrow$ then $(q,d)\downarrow$.*

Two closed process terms p and q are stateless bisimilar*, denoted by $p \underline{\leftrightarrow}_{\mathrm{sl}}\, q$, if there exists a stateless bisimulation R such that $(p,q) \in R$. Stateless bisimilarity is extended to open terms in the standard way: two open terms $t, t' \in \mathbb{T}(\Sigma_P)$ are stateless bisimilar when $\sigma(t) \underline{\leftrightarrow}_{\mathrm{sl}}\, \sigma(t')$ holds for each closed substitution σ.*

Example 1. *The processes $\mathit{tell}(u) \parallel \mathit{get}(u)$ and $\mathit{tell}(u); \mathit{get}(u) + \mathit{get}(u); \mathit{tell}(u)$ are stateless bisimilar for each tuple u. Indeed, using the rules in Definition 2.1, it is not hard to check that the symmetric closure of the relation R, consisting of the pair*

$$(\mathit{tell}(u) \parallel \mathit{get}(u), \mathit{tell}(u); \mathit{get}(u) + \mathit{get}(u); \mathit{tell}(u))$$

and the pairs

$$(\varepsilon \parallel \mathit{get}(u), \varepsilon; \mathit{get}(u)),\ (\mathit{tell}(u) \parallel \varepsilon, \varepsilon; \mathit{tell}(u)),\ (\varepsilon \parallel \varepsilon, \varepsilon),$$

is a stateless bisimulation.

Example 2. *Consider the terms $\mathit{ask}(u) + \mathit{nask}(u)$ and $\mathit{ask}(v) + \mathit{nask}(v)$, where u and v are (possibly different) tuples. By Definition 2.2, these terms are stateless bisimilar as they both transition to ε, independently of the data term $d \in T(\Sigma_D)$ they are paired up with, leaving the data term d unchanged.*

Definition 2.3 (Congruence)**.** *Let Σ be a signature. An equivalence relation $\sim$ over Σ-terms is a* congruence *if, for all $f \in \Sigma$ and closed terms $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$, where n is the arity of f, if $p_i \sim q_i$ for each $i \in \{1, \ldots, n\}$ then $f(p_1, \ldots, p_n) \sim f(q_1, \ldots, q_n)$.*

The following result is easy to show.

Proposition 3. $\underline{\leftrightarrow}_{\mathrm{sl}}$ *is a congruence for Linda.*

Definition 2.4 (Axiom system, Derivability [2])**.** *An* axiom system *is a pair (Σ, E), where Σ is a signature and E is a set of* axioms *(*equations*) of the form $s = t$, where $s, t \in \mathbb{T}(\Sigma)$.*

By $\vdash$ we denote the well known notion of derivability *in equational logic—closure under substitutions and contexts, and the fact that equality is an equivalence relation are the means through which one can derive equations.*

An axiom system (Σ, E) is often identified with the set of equations E when the signature Σ is clear from the context.

Example 4. *The following axiom system E^1 over the signature for Linda coincides with the one for BPA with the empty process and δ proposed in [21, Table 4, page 291].*

$$\begin{array}{rcll}
x+y &=& y+x & (e_1)\\
x+(y+z) &=& (x+y)+z & (e_2)\\
\varepsilon+\varepsilon &=& \varepsilon & (e_3)\\
\varepsilon+\delta &=& \varepsilon & (e_4)\\
(x+y);z &=& (x;z)+(y;z) & (e_5)\\
x;(y;z) &=& (x;y);z & (e_6)\\
\delta;x &=& \delta & (e_7)\\
\varepsilon;x &=& x & (e_8)\\
x;\varepsilon &=& x & (e_9)
\end{array}$$

Using it, one can derive, for example, the following equations, which state that the + operation is idempotent and has δ as unit element:

$$x+x = x \tag{1}$$

$$x+\delta = x. \tag{2}$$

The basic sanity criterion for an axiom system is that it only allows one to derive valid equalities between terms. This is the so-called soundness property, which is formalized in the following definition in the setting of Linda modulo stateless bisimilarity.

Definition 2.5 (Soundness). *An axiom system E over Σ_P is* sound *when, for all $s,t \in \mathbb{T}(\Sigma_P)$, if $E \vdash s = t$ then $s \underline{\leftrightarrow}_{\mathrm{sl}}\, t$.*

The following result can be shown following standard lines.

Lemma 5. *E^1 is sound for stateless bisimilarity over Linda.*

Given a finite index set $I = \{i_1, \ldots, i_n\}$ and an indexed set of terms $\{t_i\}_{i\in I}$, we write $\sum_{i\in I} t_i$ for $t_{i_1} + \cdots + t_{i_n}$. An empty sum stands for δ. (The generalized sum notation is justified since, by the above lemma and the derivability of equation (2), the + operation is associative, commutative and has δ has unit element, modulo stateless bisimilarity.)

Ideally, one would like to have axiom systems that are strong enough to prove all the equalities that are valid with respect to the chosen notion of equivalence over Linda. As is customary in the literature on process calculi, in what follows we will focus on ground-complete axiom systems.

Definition 2.6 (Ground Completeness). *An axiom system E is* ground complete *when, for all $p,q \in T(\Sigma_P)$, if $p \underline{\leftrightarrow}_{\mathrm{sl}}\, q$ then $E \vdash p = q$.*

3. Axiomatization

In this section we provide a sound and ground complete axiomatization for stateless bisimilarity over the collection of Linda process terms. The axiom system is finite if the tuple names mentioned in the axioms are taken to be variables ranging over tuples. We build the axiomatization incrementally starting with all the operations except for *nask* and $\_\|\_$. Then we discuss the issues *nask* poses and how to overcome them by means of two equations (Section 3.1). In order to avoid cluttering previous explanations, the axiomatization for $\_\|\_$ comes only at the end and will be given following standard lines (Section 3.2).

We proceed by introducing the notion of *normal form* of a Linda process term, which plays a crucial role in proving the completeness of the proposed axiomatization.

Definition 3.1 (Normal Form)**.** *A term $t \in T(\Sigma_P)$ is in* normal form *if it is of the form $(\sum_{i\in I} a_i(u_i); t_i)[+\varepsilon]$ (that is, either $\sum_{i\in I} a_i(u_i); t_i$ or $(\sum_{i\in I} a_i(u_i); t_i)+ \varepsilon$), with I a finite, possibly empty, index set, $a_i \in \{ask, nask, tell, get\}$, $u_i \in \mathcal{U}$ and t_i is in normal form for each $i \in I$.*

The terms $a_i(u_i); t_i$ $(i \in I)$ and ε, if present, are called the summands *of $(\sum_{i\in I} a_i(u_i); t_i)[+\varepsilon]$.*

An axiom system E over Σ_P is normalizing *for $t \in T(\Sigma_P)$ if there exists a term $t' \in T(\Sigma_P)$ in normal form such that $E \vdash t = t'$.*

In what follows, we let Σ^1_P be Σ_P without the operations $\_\|\_$ and *nask*(u), for all $u \in U$, and Σ^2_P be Σ_P without the operation $\_\|\_$. The following lemma can be shown following standard lines.

Lemma 6. *E^1 is normalizing for each closed term in Σ^2_P.*

Theorem 3.2. *E^1 is sound and ground complete for stateless bisimilarity over $T(\Sigma^1_P)$.*

Proof. The soundness of the axiom system is given in Lemma 5.

In order to establish the ground completeness of E^1, we shall prove that, for all $p, q \in T(\Sigma^1_P)$,

$$p \underline{\leftrightarrow}_{\mathrm{sl}}\, q \Rightarrow E^1 \vdash p = q.$$

(Note that we may assume, by Lemma 6, that p and q are in normal form.) To this end, we first define the function *height* that computes the height of the syntax tree associated with a term $t \in T(\Sigma^1_P)$:

$$height(p) = \begin{cases} 0 & \text{if } p \text{ is a constant,} \\ 1 + \max(height(p_1), height(p_2)) & \text{if } p = p_1 + p_2 \text{ or } p_1; p_2. \end{cases}$$

We prove the claim by induction on $M = \max(height(p), height(q))$.

Base case: If $M = 0$ then $p = q = \delta$, because none of the terms in the set $\{\varepsilon\} \cup \{ask(u), tell(u), get(u) \mid u \in \mathcal{U}\}$ is in normal form, and the claim follows immediately by reflexivity.

Inductive step, $M > 0$: In order to show that $p = q$ we argue that each summand of p is provably equal to a summand of q. We proceed by examining the possible forms a summand of p may have.

- Assume that ε is a summand of p. Then $(p, \emptyset)\downarrow$. As $p \underline{\leftrightarrow}_{\mathrm{sl}}\, q$, it holds that $(q, \emptyset)\downarrow$. Since q is in normal form, ε is also a summand of q.

- Let $\mathit{ask}(u); p'$ be a summand of p. This yields that $(p, u) \to (\varepsilon; p', u)$. As $p \underline{\leftrightarrow}_{\mathrm{sl}}\, q$ and q is in normal form, we have that $(q, u) \to (\varepsilon; q', u)$ for some q' such that $\varepsilon; p' \underline{\leftrightarrow}_{\mathrm{sl}}\, \varepsilon; q'$. This means that q has the summand $\mathit{ask}(u); q'$. Indeed, the primitives $\mathit{tell}(u')$, for each $u' \in \mathcal{U}$, and $\mathit{get}(u)$ alter the tuple space u, and, if $u' \neq u$, neither $\mathit{ask}(u')$ nor $\mathit{get}(u')$ can be performed in the context of the tuple space u. Since $\max(\mathit{height}(p'), \mathit{height}(q')) < M$ and $p' \underline{\leftrightarrow}_{\mathrm{sl}}\, q'$, we may use the induction hypothesis to infer that $E^1 \vdash p' = q'$. Hence, by substitutivity, $E^1 \vdash \mathit{ask}(u); p' = \mathit{ask}(u); q'$.

- Let $\mathit{tell}(u); p'$ be a summand of p. This yields that $(p, \emptyset) \to (\varepsilon; p', u)$. As $p \underline{\leftrightarrow}_{\mathrm{sl}}\, q$ and q is in normal form, we have that $(q, \emptyset) \to (\varepsilon; q', u)$ for some q' such that $\varepsilon; p' \underline{\leftrightarrow}_{\mathrm{sl}}\, \varepsilon; q'$. This means that q has the summand $\mathit{tell}(u); q'$, as $\mathit{tell}(u)$ is the only Linda primitive that can transform the empty tuple space into u. The proof now proceeds as in the above case.

- Let $\mathit{get}(u); p'$ be a summand of p. This yields that $(p, u) \to (\varepsilon; p', \emptyset)$. As $p \underline{\leftrightarrow}_{\mathrm{sl}}\, q$ and q is in normal form, we have that $(q, u) \to (\varepsilon; q', \emptyset)$ for some q' such that $\varepsilon; p' \underline{\leftrightarrow}_{\mathrm{sl}}\, \varepsilon; q'$. This means that q has the summand $\mathit{get}(u); q'$, as $\mathit{get}(u)$ is the only Linda primitive that can transform u into the empty tuple space. The proof now proceeds as above.

As each summand of p is provably equal to a summand of q, we have that $E^1 \vdash q = p + q$. (Note that, in the case that $p = \delta + \varepsilon$, this equality can be derived using equations (1) and (2).) By symmetry, $E^1 \vdash p = p + q$ too, and therefore $E^1 \vdash p = q$. □

The import of the above result is that stateless bisimilarity over $T(\Sigma_P^1)$ has the same axiomatization as standard strong bisimilarity [17, 20] over BPA with the empty process and δ. This may seem surprising at first sight, since the definition of stateless bisimilarity over Linda given in Definition 2.2 is based on an unlabelled transition system semantics, whereas strong bisimilarity is based on a labelled transition system semantics. However, as the proof of the ground-completeness result given above indicates, the effect on the tuple space of the execution of the primitive operations in Linda considered so far, in combination with the definition of stateless bisimilarity, essentially encodes the primitive operation that is being executed in an unlabelled computational step. We now make this intuition precise, by showing how the problem of axiomatizing stateless bisimilarity over $T(\Sigma_P^1)$ can be reduced to that of axiomatizing ordinary bisimilarity over that language.

Let $\mathcal{A}^+ = \{\mathit{ask}(u), \mathit{tell}(u), \mathit{get}(u) \mid u \in \mathcal{U}\}$. We define a partial function

$$\frac{}{\varepsilon\downarrow} \qquad \frac{x\downarrow}{x+y\downarrow} \qquad \frac{y\downarrow}{x+y\downarrow} \qquad \frac{x\downarrow \quad y\downarrow}{x\;;\;y\downarrow}$$

$$\frac{}{\mathit{ask}(u) \xrightarrow{\mathit{ask}(u)} \varepsilon} \qquad \frac{}{\mathit{tell}(u) \xrightarrow{\mathit{tell}(u)} \varepsilon} \qquad \frac{}{\mathit{get}(u) \xrightarrow{\mathit{get}(u)} \varepsilon}$$

$$\frac{x \xrightarrow{\alpha} x'}{x+y \xrightarrow{\alpha} x'} \qquad \frac{y \xrightarrow{\alpha} y'}{x+y \xrightarrow{\alpha} y'} \qquad \frac{x \xrightarrow{\alpha} x'}{x\;;\;y \xrightarrow{\alpha} x'\;;\;y} \qquad \frac{x\downarrow \quad y \xrightarrow{\alpha} y'}{x\;;\;y \xrightarrow{\alpha} y'}$$

Figure 1: SOS rules for the labelled transition system semantics for $T(\Sigma_P^1)$ ($\alpha \in \mathcal{A}^+$)

upd : $\mathcal{A}^+ \times T(\Sigma_D) \to T(\Sigma_D)$ as follows, where $d \in T(\Sigma_D)$ and $u \in \mathcal{U}$:

$$\begin{aligned} \mathrm{upd}(\mathit{ask}(u), d\ u) &= d\ u, \\ \mathrm{upd}(\mathit{get}(u), d\ u) &= d \qquad \text{and} \\ \mathrm{upd}(\mathit{tell}(u), d) &= d\ u. \end{aligned}$$

Intuitively, $\mathrm{upd}(\alpha, d) = d'$ holds for some $\alpha \in \mathcal{A}^+$ and $d, d' \in T(\Sigma_D)$ if, and only if, the primitive operation α can be executed in the context of the tuple space represented by d and its execution results in the tuple space represented by d'. For example, the first equation in the definition of the function upd specifies that $\mathrm{upd}(\mathit{ask}(u), d)$ is only defined if $u \in d$, and that the execution of $\mathit{ask}(u)$ leaves d unchanged.

The following lemma connects the transition system semantics for $T(\Sigma_P^1)$ given in Definition 2.1 with the standard labelled transition system semantics that $T(\Sigma_P^1)$ inherits when viewed as BPA, with ε and δ, over the set of actions $\mathcal{A}^+$, which is given in Figure 1.

Lemma 7. *For all $p, p' \in T(\Sigma_P^1)$ and $d, d' \in T(\Sigma_D)$,*

$$(p, d) \to (p', d') \Leftrightarrow \exists \alpha \in \mathcal{A}^+.\ p \xrightarrow{\alpha} p' \text{ and } \mathrm{upd}(\alpha, d) = d'.$$

Proof. Both implications can be shown by induction on the proof of the relevant transition. We omit the straightforward details. □

As an easy, but useful, corollary of the above lemma and of the definition of the upd function, we have the following observations.

Corollary 8. *For all $p, p' \in T(\Sigma_P^1)$ and $u \in \mathcal{U}$, the following statements hold:*

1. *$(p, u) \to (p', u)$ if, and only if, $p \xrightarrow{\mathit{ask}(u)} p'$;*
2. *$(p, u) \to (p', \emptyset)$ if, and only if, $p \xrightarrow{\mathit{get}(u)} p'$; and*
3. *$(p, \emptyset) \to (p', u)$ if, and only if, $p \xrightarrow{\mathit{tell}(u)} p'$.*

For the sake of clarity, we now recall the standard definition of bisimilarity in the presence of the termination predicate.

Definition 3.3 (Bisimilarity). *A relation $R \subseteq T(\Sigma_P^1) \times T(\Sigma_P^1)$ is a* bisimulation *if and only if it is symmetric and, whenever $(p, q) \in R$, the following conditions hold:*

- $\forall p' \in T(\Sigma_P^1).\ p \xrightarrow{\alpha} p' \Rightarrow \exists q' \in T(\Sigma_P^1).\ q \xrightarrow{\alpha} q' \wedge (p', q') \in R$;
- $p \downarrow \ \Rightarrow \ q \downarrow$.

Two closed process terms p and q are bisimilar*, denoted by $p \underline{\leftrightarrow} q$, if there exists a bisimulation R such that $(p, q) \in R$. Bisimilarity is extended to open terms in the standard way: two open terms $t, t' \in \mathbb{T}(\Sigma_P^1)$ are bisimilar when $\sigma(t) \underline{\leftrightarrow} \sigma(t')$ holds for each closed substitution σ.*

Theorem 3.4. *Stateless bisimilarity and bisimilarity coincide over $T(\Sigma_P^1)$—that is, $p \underline{\leftrightarrow}_{\mathrm{sl}} q$ if, and only if, $p \underline{\leftrightarrow} q$, for all $p, p' \in T(\Sigma_P^1)$.*

Proof. Using Corollary 8, it is not hard to show that $\underline{\leftrightarrow}_{\mathrm{sl}}$ is a bisimulation and, using Lemma 7, one proves that $\underline{\leftrightarrow}$ is a stateless bisimulation. In establishing both claims, we use the simple observation that, for all $p \in T(\Sigma_P^1)$,

$$p \downarrow \text{ if, and only if, } (p, d) \downarrow \text{ for all } d \in T(\Sigma_D),$$

and that, as remarked earlier, the predicate $\downarrow$ from Definition 2.1 is independent of the data component in a configuration. $\square$

The above result yields an alternative, but less direct, proof of Theorem 3.2 by reducing the problem of axiomatizing stateless bisimilarity over $T(\Sigma_P^1)$ to that of axiomatizing bisimilarity over that language. It is well known that bisimilarity is axiomatized by the axiom system E^1 over $T(\Sigma_P^1)$—see, e.g., [3].

3.1. Adding the nask operations

As we proved above, the axiom system E^1 is ground complete for stateless bisimilarity over $T(\Sigma_P^1)$. However, when we add the *nask* operations to Σ_P^1, E^1 is *not* ground complete any more. To see this, recall that in Example 2 we argued that $\mathit{ask}(u) + \mathit{nask}(u) \underline{\leftrightarrow}_{\mathrm{sl}} \mathit{ask}(v) + \mathit{nask}(v)$ holds, even when the tuples u and v are distinct. The axiom system E^1, however, does not suffice to prove that equality when $u \neq v$.

Consider the axiom system E^2, which is obtained by adding the following equations to E^1.

$$\begin{array}{rcll} \mathit{ask}(u) + \mathit{nask}(u) + \mathit{ask}(v) & = & \mathit{ask}(u) + \mathit{nask}(u) & \text{for all } u, v \in \mathcal{U} \quad (e_{10}) \\ \mathit{ask}(u) + \mathit{nask}(u) + \mathit{nask}(v) & = & \mathit{ask}(u) + \mathit{nask}(u) & \text{for all } u, v \in \mathcal{U} \quad (e_{11}) \end{array}$$

Note that, using the above equations, one may derive the following ones:

$$\begin{array}{rcll} \mathit{ask}(u) + \mathit{nask}(u) + \mathit{ask}(w) & = & \mathit{ask}(v) + \mathit{nask}(v) & \text{for all } u, v, w \in \mathcal{U}, \\ \mathit{ask}(u) + \mathit{nask}(u) + \mathit{nask}(w) & = & \mathit{ask}(v) + \mathit{nask}(v) & \text{for all } u, v, w \in \mathcal{U}. \end{array}$$

Our order of business in the remainder of this subsection is to show that E^2 is sound and ground complete modulo stateless bisimilarity over $T(\Sigma_P^2)$. (Recall that Σ_P^2 is Σ_P without the operation $\_\|\_$.)

For each $p = (\sum_{i\in I} a_i(u_i); p_i)[+\varepsilon] \in T(\Sigma_P^2)$ in normal form, let:

- $p_{an} = \sum_{i\in I,\ a_i \text{ is } ask \text{ or } nask} a_i(u_i); p_i$,
- $p_{gt} = \sum_{i\in I,\ a_i \text{ is } get \text{ or } tell} a_i(u_i); p_i$.

The following 'decomposition lemma' will be useful in establishing the desired completeness result.

Lemma 9. *Let $p, q \in T(\Sigma_P^2)$ be two terms in normal form. Then p and q are stateless bisimilar if, and only if,*

1. $p_{an} \underline{\leftrightarrow}_{\mathrm{sl}}\, q_{an}$,
2. $p_{gt} \underline{\leftrightarrow}_{\mathrm{sl}}\, q_{gt}$, *and*
3. ε *is a summand of p if, and only if, it is a summand of q.*

Proof. The 'if' implication follows because, by Proposition 3, stateless bisimilarity is preserved by $+$. Assume now that p and q are stateless bisimilar normal forms. We shall show that statements 1–3 hold. Statement 3 is immediate since, for each term t in normal form, $(t, \emptyset)\downarrow$ if, and only if, ε is a summand of t. To prove the other two statements, it suffices to show that the relation

$$R = \{(p_{an}, q_{an}), (p_{gt}, q_{gt}) \mid p, q \text{ are in normal form and } p \underline{\leftrightarrow}_{\mathrm{sl}}\, q\} \cup \underline{\leftrightarrow}_{\mathrm{sl}}$$

is a stateless bisimulation. First of all, note that R is symmetric since so is $\underline{\leftrightarrow}_{\mathrm{sl}}$. We limit ourselves to presenting the verification of the stateless bisimulation conditions for a pair of the form $(p_{an}, q_{an}) \in R$. The proof for pairs of the form $(p_{gt}, q_{gt}) \in R$ is similar and we omit it.

Let $d \in T(\Sigma_D)$. By definition of p_{an}, we have that $(p_{an}, d)\downarrow$ does not hold. Hence the second clause in Definition 2.2 is met vacuously. Assume now that $(p_{an}, d) \rightarrow (p', d')$ for some p' and d'. Then there is a summand $a_i(u_i); p_i$ of p_{an} such that '$a_i(u_i)$ is enabled in the context of d', $p' = \varepsilon; p_i$ and $d = d'$. Since $a_i(u_i); p_i$ is also a summand of p, we have that $(p, d) \rightarrow (p', d)$. From our assumption that p and q are stateless bisimilar, we infer that $(q, d) \rightarrow (q', d)$ for some q' such that $p' \underline{\leftrightarrow}_{\mathrm{sl}}\, q'$. Since the execution of *get* and *tell* primitives modifies the tuple space d, the above transition is due to a summand of q that is also a summand of q_{an}. Thus, $(q_{an}, d) \rightarrow (q', d)$ for some q' such that $p' \underline{\leftrightarrow}_{\mathrm{sl}}\, q'$. Since $\underline{\leftrightarrow}_{\mathrm{sl}}$ is included in R, we are done. □

Theorem 3.5. *E^2 is sound and ground complete modulo stateless bisimilarity over $T(\Sigma_P^2)$.*

Proof. It is easy to check that equations (e_{10}) and (e_{11}) are sound.

In order to establish the ground completeness of E^2, we shall prove that, for all $p, q \in T(\Sigma_P^2)$,

$$p \underline{\leftrightarrow}_{\mathrm{sl}}\, q \Rightarrow E^2 \vdash p = q.$$

Assume that $p \underline{\leftrightarrow}_{\mathrm{sl}} q$. We shall prove the claim above by induction on $M = \max(\mathit{height}(p), \mathit{height}(q))$, where the function *height* from the proof of Theorem 3.2 is extended to $T(\Sigma_P^2)$ by setting

$$\mathit{height}(\mathit{nask}(u)) = 0.$$

By Lemma 6, we may assume that p, q are in normal form. Our induction hypothesis is that $E^2 \vdash p' = q'$ for all p', q' such that $p' \underline{\leftrightarrow}_{\mathrm{sl}} q'$ and $M > \max(\mathit{height}(p'), \mathit{height}(q'))$.

By Lemma 9, p_{gt} and q_{gt} are stateless bisimilar and they can be proved equal by mimicking the proof of Theorem 3.2. Using again Lemma 9, we have that p_{an} and q_{an} are also stateless bisimilar. We claim that

$$E^2 \vdash p_{an} = q_{an}, \tag{3}$$

from which the equality $p = q$ follows by substitutivity, also in the light of the last statement in Lemma 9.

In order to show the above claim, note, first of all, that, modulo E^2, we may assume that p_{an} and q_{an} are in the adapted normal forms

$$\begin{aligned} p_{an} &= \sum_{i \in I}((\sum_{j \in J_i} a_j(u_j))); p_i \quad \text{and} \\ q_{an} &= \sum_{i \in I}((\sum_{k \in K_i} b_k(v_k))); p_i, \end{aligned}$$

where J_i and K_i are non-empty index sets ($i \in I$), $a_j, b_k \in \{\mathit{ask}, \mathit{nask}\}$ and $E^2 \not\vdash p_i = p_j$ (or, equivalently, $p_i \underline{\not\leftrightarrow}_{\mathrm{sl}} p_j$), for all $i, j \in I$ such that $i \neq j$. To see this, assume that $p_{an} = \sum_{\ell \in L} a_\ell(u_\ell); p_\ell$ and that $p_{\ell_1} \underline{\leftrightarrow}_{\mathrm{sl}} p_{\ell_2}$, for some $\ell_1, \ell_2 \in L$. Since

$$\max(\mathit{height}(p_{\ell_1}), \mathit{height}(p_{\ell_2})) < \mathit{height}(p_{an}) \leq \mathit{height}(p) \leq M,$$

we may use the inductive hypothesis to obtain that $E^2 \vdash p_{\ell_1} = p_{\ell_2}$. Therefore, modulo E^2,

$$p_{an} = \sum_{\ell \in L} a_\ell(u_\ell); p'_\ell,$$

where $p'_\ell \in \{p_h \mid h \in L \text{ and } p_\ell \underline{\leftrightarrow}_{\mathrm{sl}} p_h\}$ is a canonical representative of the equivalence class of the 'suffixes' of p_{an} that are stateless bisimilar to p_ℓ, for each $\ell \in L$. For this reason, we can group all the summands of p_{an} with the same suffix p_i modulo E^2 by applying equation (e_5) form right to left as needed. This puts p_{an} in the desired form, namely

$$p_{an} = \sum_{i \in I}((\sum_{j \in J_i} a_j(u_j))); p_i.$$

(Note that each J_i is non-empty.) Let $q_{an} = \sum_{h \in H} b_h(v_h); q_h$. Since p_{an} and q_{an} are stateless bisimilar, it is not hard to see that, for each $h \in H$, there is some $i \in I$ such that $p_i \underline{\leftrightarrow}_{\rm sl} q_h$. Using the inductive hypothesis as above, the equality $p_i = q_h$ can be proved from E^2. This means that q_{an} can be put in the form $\sum_{i \in I}((\sum_{k \in K_i} b_k(v_k))); p_i$, by substitutivity and applying equation (e_5) form right to left as needed. (Note that, since $p_{an} \underline{\leftrightarrow}_{\rm sl} q_{an}$, each K_i is non-empty.)

In order to show claim (3), it therefore suffices to show that $C_i = \sum_{j \in J_i} a_j(u_j)$ is provably equal to $D_i = \sum_{k \in K_i} b_k(v_k)$, for each $i \in I$.

For a fixed $i \in I$, we consider the two sums of *ask* and *nask* terms C_i and D_i. Since E^2 proves equation (1), to the effect that + is idempotent, we may assume in what follows that all the summands of C_i and D_i are different. We shall prove that $E^2 \vdash C_i = D_i$ by case analysis on their possible form.

1. $C_i = ask(u) + nask(u) + C'$ and $D_i = ask(v) + nask(v) + D'$, for some $u, v \in \mathcal{U}$ and C', D'. (Note that, in the light of equation (2), C' and D' may be δ.) Then the equality $C_i = D_i$ can be shown by using equations (e_{10}) and (e_{11}) repeatedly.
2. $C_i = ask(u)+nask(u)+C$, for some $u \in \mathcal{U}$ and C, and $D_i = \sum_{k \in K_i} b_k(v_k)$, with $v_{k_1} \neq v_{k_2}$ whenever $k_1 \neq k_2$, for all $k_1, k_2 \in K_i$. We shall argue that this case is impossible, since it contradicts the assumption that p_{an} and q_{an} are stateless bisimilar.
   By using equations (e_{10}) and (e_{11}) repeatedly, we can derive the equation $C_i = ask(u) + nask(u)$. It is easy to see that $(C_i, d) \to (\varepsilon, d)$, for each $d \in T(\Sigma_D)$. Therefore $(p_{an}, d) \to (\varepsilon; p_i, d)$, for each $d \in T(\Sigma_D)$. However, since $D_i = \sum_{k_1 \in K_i^1} ask(v_{k_1}) + \sum_{k_2 \in K_i^2} nask(v_{k_2})$ with $K_i^1 \cap K_i^2 = \emptyset$ and $K_i^1 \cup K_i^2 = K_i$, the tuple space $d' = \{v_{k_2} \mid k_2 \in K_2\}$ 'blocks' D_i (no transition can be performed from (D_i, d')). This means that (q_{an}, d') does not have a transition leading to $(\varepsilon; p_i, d')$, which contradicts the assumption that p_{an} and q_{an} are stateless bisimilar.
3. $D_i = ask(u)+nask(u)+D$, for some $u \in \mathcal{U}$ and D, and $C_i = \sum_{k \in K_i} b_k(v_k)$, with $v_{k_1} \neq v_{k_2}$ whenever $k_1 \neq k_2$, for all $k_1, k_2 \in K_i$. This case is symmetric to the one above.
4. Assume that none of the previous cases applies. Let

   $$C_i = \sum_{j_1 \in J_i^1} ask(v_{j_1}) + \sum_{j_2 \in J_i^2} nask(v_{j_2}),$$

   with $J_i^1 \cap J_i^2 = \emptyset$ and $J_i^1 \cup J_i^2 = J_i$, and

   $$D_i = \sum_{k_1 \in K_i^1} ask(v_{k_1}) + \sum_{k_2 \in K_i^2} nask(v_{k_2}),$$

   with $K_i^1 \cap K_i^2 = \emptyset$ and $K_i^1 \cup K_i^2 = K_i$. We have that $v_{j_1} \neq v_{j_2}$, for each $j_1 \in J_i^1$ and $j_2 \in J_i^2$, and $v_{k_1} \neq v_{k_2}$, for each $k_1 \in K_i^1$ and $k_2 \in K_i^2$.
   We show that each summand in C_i appears in D_i, and vice versa, by reductio ad absurdum. We consider the following four cases, each of which contradicts the assumption that p_{an} and q_{an} are stateless bisimilar.

- Suppose $J_i^1 \setminus K_i^1 \neq \emptyset$. Then $d = \{v_l \mid l \in (J_i^1 \setminus K_i^1) \cup K_i^2\}$ 'blocks' D_i, but $(C_i, d) \to (\varepsilon, d)$. This means that (q_{an}, d) does not have a transition leading to $(\varepsilon; p_i, d)$, whereas (p_{an}, d) does. This contradicts the assumption that p_{an} and q_{an} are stateless bisimilar.
- Suppose $K_i^1 \setminus J_i^1 \neq \emptyset$. The proof for this case is similar to that for the previous one.
- Suppose $J_i^2 \setminus K_i^2 \neq \emptyset$. Then $d = \{v_l \mid l \in K_i^2\}$ 'blocks' D_i, but $(C_i, d) \to (\varepsilon, d)$. This means that (q_{an}, d) does not have a transition leading to $(\varepsilon; p_i, d)$, whereas (p_{an}, d) does. This contradicts the assumption that p_{an} and q_{an} are stateless bisimilar.
- Suppose $K_i^2 \setminus J_i^2 \neq \emptyset$. The proof for this case is similar to that for the previous one.

Concluding, $E^2 \vdash C_i = D_i$ for each $i \in I$. This means that $E^2 \vdash p_{an} = q_{an}$, and we are done. $\square$

3.2. Adding parallel composition

Our goal in this section is to axiomatize stateless bisimilarity over the full Linda language studied in this paper. Consider the signature Σ_P^3, an extension of Σ_P with the binary left merge operation $\_ \mathbin{|\!\!\lfloor} \_$, which stems from [4], defined by the rules:

$$(r_{16})\ \frac{(x,d) \to (x',d')}{(x \mathbin{|\!\!\lfloor} y, d) \to (x' \parallel y, d')} \qquad (r_{17})\ \frac{(x,d)\downarrow \quad (y,d)\downarrow}{(x \mathbin{|\!\!\lfloor} y, d)\downarrow}$$

As is well known (see, e.g., [18]), the left merge operation is necessary in order to obtain finite equational axiomatizations of bisimilarity in process algebras.

Consider the axiom system E^3 which is E^2 enriched with the following equations.

$$\begin{array}{rcll}
x \parallel y & = & x \mathbin{|\!\!\lfloor} y + y \mathbin{|\!\!\lfloor} x & (e_{12})\\
(x+y) \mathbin{|\!\!\lfloor} z & = & (x \mathbin{|\!\!\lfloor} z) + (y \mathbin{|\!\!\lfloor} z) & (e_{13})\\
(a(u);x) \mathbin{|\!\!\lfloor} y & = & a(u);(x \parallel y) \ \text{ for all } a \in \{\mathit{ask}, \mathit{nask}, \mathit{tell}, \mathit{get}\} \text{ and } u \in \mathcal{U} & (e_{14})\\
\varepsilon \mathbin{|\!\!\lfloor} (x+y) & = & \varepsilon \mathbin{|\!\!\lfloor} x + \varepsilon \mathbin{|\!\!\lfloor} y & (e_{15})\\
\varepsilon \mathbin{|\!\!\lfloor} (a(u);y) & = & \delta \ \text{ for all } a \in \{\mathit{ask}, \mathit{nask}, \mathit{tell}, \mathit{get}\} \text{ and } u \in \mathcal{U} & (e_{16})\\
\varepsilon \mathbin{|\!\!\lfloor} \varepsilon & = & \varepsilon & (e_{17})\\
\varepsilon \mathbin{|\!\!\lfloor} \delta & = & \delta & (e_{18})\\
\delta \mathbin{|\!\!\lfloor} x & = & \delta & (e_{19})
\end{array}$$

Theorem 3.6. *E^3 is sound and ground complete modulo stateless bisimilarity over $T(\Sigma_P^3)$.*

Proof. The soundness of E^3 modulo stateless bisimilarity can be shown following standard lines. The ground completeness of E^3 modulo stateless bisimilarity can be reduced to that of E^2 over $T(\Sigma_P^2)$. Indeed, by induction on the size of terms, one can show that the equations $(e_{12} - e_{19})$ can be used to eliminate each occurrence of $\parallel$ and $\mathbin{|\!\!\lfloor}$ from terms. $\square$

4. Conclusions

In this paper, we have presented a sound and ground-complete axiomatization for stateless bisimilarity over the collection of Linda process terms. The axiom system is finite if the tuple names mentioned in the axioms are taken to be variables ranging over tuples. In fact, as can be shown using the technique presented in [12], the axiom systems E^1 and E^2 for the two fragments of the sequential sub-language of Linda we consider in this paper are ω-complete as well as complete for stateless bisimilarity. (We refer the interested reader to [1] for information on (ω-)complete axiom systems and the connections between completeness and ω-completeness.)

The axiom systems we present in this paper may form the core of axiom systems for coarser notions of equivalence over Linda. In particular, an interesting direction for future research is the development of a complete axiomatization for the notion of behavioural equivalence over Linda studied in [6].

References

[1] L. Aceto, W. Fokkink, A. Ingólfsdóttir, and B. Luttik. Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles*, volume 3838 of *Lecture Notes in Computer Science*, pages 338–367. Springer, 2005.

[2] J. C. Baeten, T. Basten, and M. A. Reniers. *Process Algebra: Equational Theories of Communicating Processes*, volume 50 of *Cambridge Tracts in Theoretical Computer Science*. Cambrdige University Press, 2010.

[3] J. C. M. Baeten. Embedding untimed into timed process algebra: the case for explicit termination. *Mathematical Structures in Computer Science*, 13(4):589–618, 2003.

[4] J. A. Bergstra and J. W. Klop. Fixedpoint semantics in process algebra. Technical Report IW 206/82, Center for Mathematics, Amsterdam, The Netherlands, 1982.

[5] V. Bos and J. J. Kleijn. Redesign of a systems engineering language — formalisation of χ. *Formal Aspects of Computing*, 15(4):370–389, Dec. 2003.

[6] A. Brogi and J.-M. Jacquet. On the expressiveness of Linda-like concurrent languages. In I. Castellani and C. Palamidessi, editors, *Proceedings of the 5th International Workshop on Expressiveness in Concurrency (EXPRESS'98)*, volume 16 of *Electronic Notes in Theoretical Computer Science (ENTCS)*. Elsevier Science, Dordrecht, The Netherlands, 1998, 1998.

[7] N. Carriero and D. Gelernter. Linda in context. *Communications of the ACM*, 32(4):444–458, 1989.

[8] P. J. Cuijpers and M. A. Reniers. Hybrid process algebra. *Journal of Logic and Algebraic Programming*, 62(2):191–245, 2005.

[9] R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.

[10] R. De Nicola and R. Pugliese. Linda-based applicative and imperative process algebras. *Theoretical Computer Science*, 238(1–2):389–437, 2000.

[11] D. Gelernter. Generative communication in linda. *ACM Transactions on Programming Languages and Systems*, 7(1):80–112, 1985.

[12] J. F. Groote. A new strategy for proving ω-completeness applied to process algebra. In J. C. M. Baeten and J. W. Klop, editors, *CONCUR '90, Theories of Concurrency: Unification and Extension, Amsterdam, The Netherlands, August 27–30, 1990, Proceedings*, volume 458 of *Lecture Notes in Computer Science*, pages 314–331. Springer, 1990.

[13] J. F. Groote and A. Ponse. Process algebra with guards: Combining Hoare logic with process algebra. *Formal Aspects of Computing*, 6(2):115–164, 1994.

[14] M. Hennessy and A. Ingólfsdóttir. Communicating processes with value-passing and assignments. *Formal Aspects of Computing*, 5(5):432–466, 1993.

[15] M. Hennessy and A. Ingólfsdóttir. A theory of communicating processes with value passing. *Information and Computation*, 107(2):202–236, 1993.

[16] M. Hennessy, H. Lin, and J. Rathke. Unique fixpoint induction for message-passing process calculi. *Science of Computer Programming*, 41(3):241–275, 2001.

[17] A. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[18] F. Moller. The importance of the left merge operator in process algebras. In M. Paterson, editor, *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, July 16–20, 1990, Proceedings*, volume 443 of *Lecture Notes in Computer Science*, pages 752–764. Springer, 1990.

[19] M. R. Mousavi, M. A. Reniers, and J. F. Groote. Notions of bisimulation and congruence formats for SOS with data. *Information and Computation*, 200(1):107–147, 2005.

[20] D. M. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[21] J. L. M. Vrancken. The algebra of communicating processes with empty process. *Theoretical Computer Science*, 177(2):287–328, 1997.