Math. Struct. in Comp. Science (2018), *vol.* 28, *pp.* 1097–1125. © Cambridge University Press 2016
doi:10.1017/S0960129516000402 First published online 21 November 2016

The true concurrency of differential interaction nets

DAMIANO MAZZA

CNRS, UMR 7030, Laboratoire d'Informatique de Paris Nord
Université Paris 13, Sorbonne Paris Cité, F-93430 Villetaneuse, France
Email: damiano.mazza@lipn.univ-paris13.fr

Received 15 May 2010; revised 31 July 2012

We analyse the reduction of differential interaction nets from the point of view of so-called 'true concurrency,' that is, employing a non-interleaving model of parallelism. More precisely, we associate with each differential interaction net an event structure describing its reduction. We show how differential interaction nets are only able to generate confusion-free event structures, and we argue that this is a serious limitation in terms of the concurrent behaviours they may express. In fact, confusion is an extremely elementary phenomenon in concurrency (for example, it already appears in CCS with just prefixing and parallel composition) and we show how its presence is preserved by any encoding respecting the degree of distribution and the reduction semantics. We thus infer that no reasonably expressive process calculus may be satisfactorily encoded in differential interaction nets. We conclude with an analysis of one such encoding proposed by Ehrhard and Laurent, and argue that it does not contradict our claims, but rather supports them.

1. Introduction

1.1. *Proof-nets and process algebras: an unsuccessful story*

Ever since Girard (1987) introduced proof-nets, there has been a widespread belief that process algebras and linear logic are somehow related. Essentially, the reason for such a belief has to do with how proof-nets represent the associativity of logical deduction. Suppose we are able to prove the three formulas $A \Rightarrow B$, $B \Rightarrow C$ and $C \Rightarrow D$; no reasonable mathematician would ever claim the existence of two distinct proofs of $A \Rightarrow D$ simply because there are two different orders in which the three lemmas may be composed. In proof-theoretic terms, composition corresponds to the cut rule, so a formalism capable of accounting for the associativity of deduction must be able to abstract from the order in which cut rules are applied. This is precisely what proof-nets, as opposed to sequent calculus, are able to do, and it is why Girard calls them 'the parallel syntax for proof theory' (Girard 1996).

The reader acquainted with standard process calculi (CSP, CCS, the π-calculus...) will have probably recognized a familiar situation in the above description: The three lemmas may be thought of as three processes P, Q, R, each capable of independently receiving on channel a, b, c, respectively, and sending on b, c, d, respectively. Then, the logical composition of the three lemmas corresponds to the process $\nu(b, c)(P \mid Q \mid R)$. This analogy, first pointed out by Abramsky (1994), was formally developed at least by

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

Abramsky (1993) himself and Bellin and Scott (1994), confirming that process calculi are indeed a suitable framework for expressing the 'parallelism' of linear logic proofs-nets.

However, although linear logic has kept providing, even in recent times, useful tools for ensuring properties of process algebras, especially via type systems (Caires and Pfenning 2010; Honda and Laurent 2010; Kobayashi *et al.* 1999; Yoshida *et al.* 2004), all further investigations have failed to bring any deep logical insight into concurrency theory, in the sense that no concurrent primitive has found a convincing counterpart in linear logic, or anything even remotely resembling the perfect correspondence between functional languages and intuitionistic logic. In our opinion, we must simply accept that linear logic is not the right framework for carrying out Abramsky's 'proofs as processes' programme (which, in fact, more than 20 years after its inception has yet to see a satisfactory completion).

The reason for such a failure must ultimately be sought in the strong determinism of the cut-elimination procedure of linear logic, which, so to speak, makes composition excessively associative: The cut rule is associative not only statically (i.e., a chain of lemmas forms, as a static object, only one proof) but also dynamically, i.e., the normal form of the chain does not depend on the order in which the cuts are eliminated (this is nothing but the Church–Rosser property). Using the process-algebraic example above, if $v(b)(P \mid Q)$ reduces to S and $v(c)(Q \mid R)$ reduces to T, dynamic associativity would mean that $v(b,c)(P \mid Q \mid R)$ has the same behaviour as $v(c)(S \mid R)$ and also as $v(b)(P \mid T)$, which cannot hold in general in a genuinely concurrent system.

In synthesis, we may put it this way: Linear logic proofs are processes, but processes are far from being linear logic proofs and, what is more important, it is unclear at this time whether what is missing has any natural proof-theoretic counterpart.

1.2. *Differential interaction nets: a new hope*

An entirely new perspective on the relationship between concurrent computation and proof theory was brought forth by the introduction of differential linear logic. This latter arose from the study of finiteness spaces, a denotational model defined by Ehrhard (2005) which achieves the long-sought goal (already foreshadowed in Girard (1987)) of interpreting the connectives of linear logic as operations on topological vector spaces.

In a nutshell, finiteness spaces endow the exponential modalities of linear logic with additional structure, introducing symmetries which are absent in linear logic: The structural rules, weakening and contraction, have symmetric counterparts, called coweakening and cocontraction (categorically, the objects of the form $!A$, which are coalgebras in linear logic, become bialgebras in differential linear logic); furthermore, the other fundamental operation on exponential modalities, called dereliction, finds a symmetric counterpart in the derivative operator, which may be defined thanks to the linear-topological structure of finiteness spaces (and is the reason behind the 'differential' adjective).

At the level of proof-nets, this translates into the *differential interaction nets* of Ehrhard and Regnier (2006), which rapidly attracted the attention of those seeking to relate linear logic and concurrency, because their cut-elimination procedure is *non-deterministic*, a true

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

novelty in the world of proof-nets. Even better, the non-determinism at work in differential interaction nets is in some sense structured, controlled, as opposed to the 'wild' nature of the non-determinism present in the cut-elimination of Gentzen's classical sequent calculus. This gives some serious hope that differential interaction nets may help us overcome the stalemate of linear logic with respect to concurrent computation.

As a matter of fact, after some time, Ehrhard and Laurent (2010b) managed to find what seems to be a satisfactory encoding of the π-calculus, with all of its main concurrent and mobile features, into differential interaction nets, in which the reduction semantics of processes is mimicked by the cut-elimination of differential interaction nets. This would pave the way for the development of a 'concurrent' Curry–Howard isomorphism, an achievement of arguably great interest.

1.3. *Our contribution: differential interaction nets are still not enough*

And yet, a close inspection of the encoding of Ehrhard and Laurent (2010b) reveals that it is not as convincing as one would hope. We give a detailed analysis of it in Section 5; for the time being, let us ignore the existence of such an encoding, and let us try to address, in an abstract way, the question of whether differential interaction nets are sufficiently expressive to be considered as a model of concurrent computation.

We start by observing that, at the intuitive level, the non-determinism of differential interaction nets seems to be too weak for general concurrency. In fact, if one allows formal sums of nets (as is the case in Ehrhard and Regnier (2006)), cut-elimination in this system still enjoys the Church–Rosser property, which means that differential interaction nets morally verify the dynamic associativity to which we alluded above.

In order to pinpoint the problem in a formal way, our idea is to turn to the most fine-grained models of parallelism, the so called 'truly concurrent,' or non-interleaving semantics of computation. The main feature of these models is that they do not reduce parallelism to a sum of sequentializations, i.e., the presence of several concurrent actions is not equated with the non-deterministic superposition of their possible linear orderings: In CCS terms, $x \mid y$ is not considered to be the same as $x.y + y.x$.

Amongst all non-interleaving models, event structures (Winskel 1982) are quite appealing, because of their simple and yet very expressive mathematical structure. The basic assumption underlying event structures is that a computational process may be described as a collection of *events*, which are related by *causality* and *conflict*. Causality is an abstraction of time in computation, and allows one to speak of sequentiality and parallelism. Conflict describes non-determinism: If two events are in conflict, we are in front of a choice between the two, as the occurrence of either one of them excludes the occurrence of the other.

In Section 3.3, we show how the reduction of differential interaction nets may easily be described in terms of event structures: Each reduction step is an event, with causal order given by necessity (a reduction step whose execution is possible only after another reduction step is executed), and conflict given by non-deterministic choice (two reduction steps coming from two ways of reducing the same redex). Once we know how to interpret differential interaction nets in event structures, we proceed as follows.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

Differential interaction nets are confusion-free. We start by observing something remarkable, namely that the event structures generated by differential interaction nets have a very special form: They are all *confusion-free* (Theorem 3.8). The notion of confusion-freeness arose in the study of Petri nets (Rozenberg and Engelfriet 1996) and, intuitively, reflects the fact that the non-determinism of a concurrent system is always 'localized': In a confusion-free system, non-deterministic choices are made by the components of the system regardless of what happens elsewhere. A typical example is the non-determinism generated by a coin toss. On the contrary, the non-determinism induced by the different possible *interactions* between the components of a distributed system generally results in confusion. The reformulation of confusion-freeness in the context of event structures which we use in this paper is due to Varacca *et al.* (2006).

The confusion-freeness of differential interaction nets is a form of dynamic associativity and, in our opinion, must be seen as the deep reason why it is possible to formulate cut-elimination with formal sums and obtain a Church–Rosser system. Anyhow, our initial question may now be reformulated as: How important is confusion in concurrency? In other words:

a. Is confusion present in the process calculi usually employed in concurrency theory?
b. If so, is it essential?

For what concerns question (a), we may find a simple example of confusion in the CCS process

$$\overline{x} \mid x.\overline{y} \mid y \mid \overline{y}$$

(see the opening of Section 4 for an explanation). Note that the above process uses only two primitives: parallel composition and prefixing. These primitives are so basic that one can hardly imagine how an acceptable model of concurrent computation may disregard them.[†] Therefore, we conclude that question (a) must be answered positively.

Acceptable encodings preserve confusion. Section 4 is devoted to give a technical argument which should convince the reader that question (b) must also be answered positively. What we should like to do is proving a statement of the form 'the CCS process above cannot be encoded in differential interaction nets.' However, we stumble here on the extremely thorny problem of defining a good notion of 'encoding' in a concurrent setting (see Parrow (2008), Gorla (2010) for a survey of the existing literature on the subject, which is quite vast).

This is where our choice of using a non-interleaving semantics turns out to be essential. In fact, we are able to avoid the elusive question of what is a concurrent encoding by reasoning directly at the level of event structures. We define one notion, called *bisimilar embedding*, which captures, in terms of events, two minimal requirements that an acceptable encoding must arguably satisfy

[†] The solos calculus (Laneve and Victor 2003) has no explicit prefixing in its syntax and uses name restriction to express causality. Nevertheless, it does not contradict our claim: What really matters is a way of expressing causality, and confusion is present in the solos calculus as well, *cf.* the example after Theorem 5.2.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

— preservation of the degree of distribution, i.e., the fact that coordinating agents (such as a scheduler) are not introduced when encoding parallel, independent events of the source language;
— operational correspondence, i.e., the fact that the reduction steps of the source language must be faithfully simulated in the target language.

These two properties may be considered 'minimal' because they appear more or less ubiquitously in the literature on encodings (Gorla 2010).

Our last step is to prove that bisimilar embeddings preserve confusion (Theorem 4.6). This makes us conclude that confusion-free calculi (such as differential interaction nets) are fundamentally less expressive than what is desirable for a model of concurrency.

Other systems of interaction nets. After discussing the encoding of Ehrhard and Laurent (2010b) and arguing that, instead of contradicting, it actually supports our negative results, we conclude the paper by applying the event structure analysis to other systems of interaction nets. The interest of these systems is in that confusion may be expressed in them. Nevertheless, they do not bring us any further in the 'proofs as processes' quest because their relationship with logic is obscure (and may be inexistent). We study in particular so-called *multiport interaction nets* (Alexiev 1999; Mazza 2005), and give a partially negative result on the existence of universal multiport systems, in contrast with the (positive) result of Lafont (1997) on universal systems for *deterministic* interaction nets.

2. Preliminaries: event structures and confusion

In what follows, if $(A, \leqslant)$ is a poset and $u \subseteq A$, we denote by $\downarrow u$ the downward closure of u, i.e., $\downarrow u = \{a' \in A \mid \exists a \in u.a' \leqslant a\}$, and we write $\downarrow a$ for the principal ideal $\downarrow\{a\}$.

Definition 2.1 (Event structure (Winskel and Nielsen 1995)). An *event structure* is a triple $E = (|E|, \leqslant_E, \smile_E)$, where

— $|E|$ is a set, the elements of which are called *events* and are ranged over by a, b, c;
— $\leqslant_E$ is a partial order on $|E|$, called *causal order*, such that, for all $a \in |E|$, $\downarrow a$ is finite;
— $\smile_E$ is an anti-reflexive symmetric relation on $|E|$, called *conflict relation*, such that, for all $a, b, c \in |E|$, $a \smile_E b \leqslant_E c$ implies $a \smile_E c$;

The complement of $\smile_E$, called *coherence relation*, is denoted by $\frown_E$. When no confusion may arise, we omit the subscript in the notations for causal order and conflict.

Let E be an event structure, and let $u \subseteq |E|$. We say that u is a *configuration* of E iff $\downarrow u = u$ and $a, b \in u$ implies $a \frown b$. The set of finite configurations of E is denoted by $\mathcal{C}(E)$, and ranged over by u, v, w. If u is a configuration and a an event such that $a \notin u$ and $u' = u \cup \{a\}$ is a configuration, we say that u *enables* a. The smallest configuration enabling a generic $a \in |E|$ is clearly $\downarrow a \setminus \{a\}$, which we denote by $\lceil a \rceil$.

We now recall the definition of confusion-free event structure, taken from Varacca *et al.* (2006).

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

Definition 2.2 (Immediate conflict). Let E be an event structure. We say that $a, a' \in |E|$ are in *immediate conflict*, and we write $a \# a'$, iff $a \smile a'$ and there exists a configuration enabling both a and a'.

Immediate conflict actually generates all other conflicts by 'propagating upwards':

Lemma 2.3 (Varacca *et al.* 2006). Let E be an event structure, and let $a, b \in |E|$. Then, $a \smile b$ iff there exist $a_0, b_0 \in |E|$ such that $a_0 \# b_0$ and $a_0 \leqslant a$, $b_0 \leqslant b$.

Definition 2.4 (Confusion). Let E be an event structure. A *confusion of type I* in E is a triple $(a, b, c) \in |E|^3$ such that $a \neq c$, $a \# b$, $b \# c$ and a and c are *not* in immediate conflict. A *confusion of type II* in E is a pair $(a, b) \in |E|^2$ such that $a \# b$ and $\lceil a \rceil \neq \lceil b \rceil$.

An event structure E is *confusion-free* if it contains no confusion or, equivalently,

— the reflexive closure of $\#$ is transitive, i.e., it is an equivalence relation;
— for every $a, b \in |E|$, $a \# b$ implies $\lceil a \rceil = \lceil b \rceil$.

3. Differential interaction nets

3.1. *Algebraic presentation*

The following definition of differential interaction nets is due to a joint work with Andrei Dorman (to be included in Dorman (2013)). It differs from the graphical presentation usually adopted in the literature (recalled in Section 3.2) in that it is more in the style of process algebras. This has the advantage of providing more concise (although less visual and hence, perhaps, less intuitive) notations which are useful for our present purposes (for example, in Definition 3.4).

We must also mention that, since they are 'the proof-nets of differential linear logic,' differential interaction nets usually come with types, which are formulas of linear logic or variants thereof (for instance, Ehrhard and Laurent (2010b) use Danos–Regnier recursive types, i.e., based on a type o satisfying $o = !o \multimap o$). However, the dynamics of nets in terms of event structures is not affected by the presence of types: Definition 3.7 makes sense and Theorem 3.8 holds in any kind of typed nets. Therefore, we completely ignore types in our presentation.

Definition 3.1 (Net). A *cell* is an expression of one of the following 10 forms:

Multiplicative cells: $\bot(x)$, $⅋(x; y, z)$, $1(x)$, $\otimes(x; y, z)$,
Exponential cells: $?\mathrm{d}(x; y)$, $?\mathrm{w}(x)$, $?\mathrm{c}(x; y, z)$, $!\mathrm{d}(x; y)$, $!\mathrm{w}(x)$, $!\mathrm{c}(x; y, z)$,

where x, y, z range over a denumerably infinite set of *ports*. Multiplicative cells are called *bottom*, *par*, *one* and *tensor*, respectively. Exponential cells are called *dereliction*, *weakening*, *contraction*, *codereliction*, *coweakening* and *cocontraction*.[†]

A *wire* is a multiset containing exactly two (not necessarily distinct) ports x, y, which we denote by $x \leftrightarrow y$ (or $y \leftrightarrow x$).

[†] Observe the symmetries between $\bot/1$, $⅋/\otimes$, $?/!$; these are dual connectives in linear logic.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

A *net* is a finite multiset of cells and wires in which every port appears at most twice. The set of *free ports* of a net μ, denoted by $\mathrm{fp}(\mu)$, is the set of ports appearing exactly once in μ. The ports appearing twice in a net are called *bound*. We identify any two nets which may be obtained one from the other by an injective renaming of their bound ports (this is α-*equivalence*).

Given two nets μ, ν, we denote by $\mu \mid \nu$ the net obtained by renaming (using α-equivalence) the bound ports of μ and ν so that the two nets have no bound name in common, and by taking then the standard multiset union. The operation $\mid$ is obviously commutative and has the empty net, denoted by $*$, as neutral element. It is not associative in general; however, for $\mu \mid (\nu \mid \rho)$ and $(\mu \mid \nu) \mid \rho$ to be equal, it is enough to suppose that $\mathrm{fp}(\mu) \cap \mathrm{fp}(\nu) \cap \mathrm{fp}(\rho) = \varnothing$. More generally, if $\mu_1, \ldots, \mu_n$ are such that, for any pairwise distinct i, j, k, $\mathrm{fp}(\mu_i) \cap \mathrm{fp}(\mu_j) \cap \mathrm{fp}(\mu_k) = \varnothing$, then the expression $\mu_1 \mid \cdots \mid \mu_n$ is not ambiguous. In the sequel, we shall always assume this to be the case.

We now introduce a notion corresponding to what is usually called *structural congruence* in process calculi. We adopt the standard notation $\equiv$ but call it ω-*equivalence* to avoid conflict with the terminology *structural reduction* (*cf.* Definition 3.3) employed in the literature on differential interaction nets, which is owed to the well-established proof-theoretic tradition of calling weakening and contraction *structural rules*.

Definition 3.2 (ω-equivalence). We write $\mu \to_\omega \mu'$ if $\mu = \nu \mid x \leftrightarrow y$, $x \in \mathrm{fp}(\nu)$ and $\mu' = \nu\{y/x\}$, where the latter denotes the net ν in which the only free occurrence of x is replaced by y. The relation $\to_\omega$ is obviously confluent and strongly normalizing. We denote by $\equiv$ the induced equivalence, i.e., $\mu \equiv \mu'$ if μ and μ' have the same normal form with respect to $\to_\omega$.

Definition 3.3 (Reduction). We define the following rewriting rules:
Multiplicative reduction:

$$\begin{aligned} 1(x) \mid \bot(x) &\to *, \\ \otimes(x; y_1, y_2) \mid \parr(x; z_1, z_2) &\to y_1 \leftrightarrow z_1 \mid y_2 \leftrightarrow z_2. \end{aligned}$$

Communication reduction:

$$!\mathrm{d}(x; y) \mid ?\mathrm{d}(x; z) \to y \leftrightarrow z.$$

Non-deterministic reduction:

$$\begin{aligned} !\mathrm{d}(x; y) \mid ?\mathrm{c}(x; z_1, z_2) &\to \begin{cases} !\mathrm{d}(z_1; y) \mid !\mathrm{w}(z_2) \\ !\mathrm{w}(z_1) \mid !\mathrm{d}(z_2; y), \end{cases} \\ ?\mathrm{d}(x; y) \mid !\mathrm{c}(x; z_1, z_2) &\to \begin{cases} ?\mathrm{d}(z_1; y) \mid ?\mathrm{w}(z_2) \\ ?\mathrm{w}(z_1) \mid ?\mathrm{d}(z_2; y). \end{cases} \end{aligned}$$

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

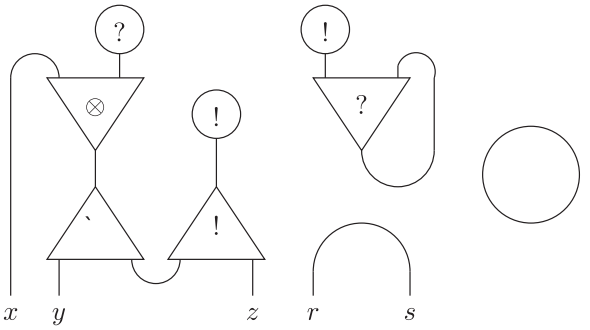


Fig. 1. Graphical representation of a net.

Structural reduction:

$$\begin{aligned}
!\mathrm{w}(x) \mid ?\mathrm{w}(x) &\to *,\\
!\mathrm{c}(x; y_1, y_2) \mid ?\mathrm{c}(x; z_1, z_2) &\to ?\mathrm{c}(y_1; r, p) \mid ?\mathrm{c}(y_2; s, q) \mid !\mathrm{c}(z_1; r, s) \mid !\mathrm{c}(z_2; p, q),\\
!\mathrm{w}(x) \mid ?\mathrm{c}(x; y_1, y_2) &\to !\mathrm{w}(y_1) \mid !\mathrm{w}(y_2),\\
?\mathrm{w}(x) \mid !\mathrm{c}(x; y_1, y_2) &\to ?\mathrm{w}(y_1) \mid ?\mathrm{w}(y_2).
\end{aligned}$$

The nets on the left side of the above rewriting rules are called *active pairs*.

The *reduction* relation of differential interaction nets, denoted by $\to$, is defined by closing the above rules under parallel composition and ω-equivalence:

$$\frac{\mu \to \mu'}{\mu \mid \nu \to \mu' \mid \nu}, \qquad \frac{\mu \equiv \mu' \quad \mu' \to \nu' \quad \nu' \equiv \nu}{\mu \to \nu}.$$

3.2. *Graphical presentation*

As mentioned above, nets admit a nice and intuitive graphical representation, in the style of string diagrams for monoidal categories. Although not as concise, the graphical representation has the advantage not only of automatically handling α-equivalence, but also of internalizing ω-equivalence: Two nets are ω-equivalent iff they have the same graphical representation. For instance, if we set

$$\begin{aligned}
\nu_1 &= \otimes(t; x, w) \mid \parr(t; u, y) \mid ?\mathrm{w}(w) \mid !\mathrm{c}(v; z, u) \mid !\mathrm{w}(v),\\
\nu_2 &= !\mathrm{w}(p) \mid !\mathrm{c}(q; p, q),\\
\mu &= \nu_1 \mid \nu_2 \mid r \leftrightarrow s \mid t \leftrightarrow t,\\
\mu' &= \nu_1 \mid \nu_2 \mid r \leftrightarrow s' \mid s' \leftrightarrow s \mid t \leftrightarrow t' \mid t' \leftrightarrow t,
\end{aligned}$$

then it is readily seen that $\mu \equiv \mu'$ and that both μ and μ' are represented graphically as in Figure 1.

The graphical form of the rewriting rules of Definition 3.3 is given in Figures 2–5. Note how the symmetries of differential interaction nets appear in an even more conspicuous way.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

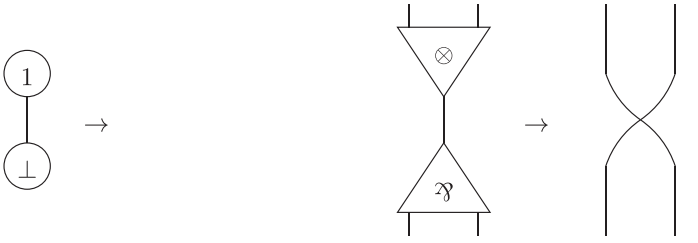


Fig. 2. Multiplicative reduction.

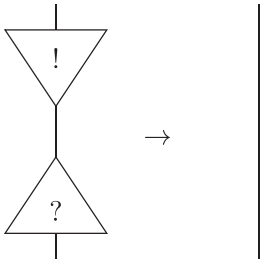


Fig. 3. Communication reduction.

3.3. *Differential interaction nets and event structures*

We are now going to associate with every net μ an event structure $\mathrm{Ev}(\mu)$ which describes the dynamics of its reduction. Intuitively, each reduction step which may be applied to μ or to any of its reducts constitutes an event. A reduction step a causally precedes a reduction step b if a must be performed in order for the active pair inducing b to appear. Two reduction steps are in conflict when they mutually exclude each other; it is the case of every pair of reduction steps reducing the same non-deterministic active pair (Figure 4). In fact, such a choice is actually the only source of conflict in differential interaction nets, and we shall see that this causes the event structure generated by a net to be always confusion-free.

In order to proceed, it is technically convenient to associate a unique label with every cell of a net, and then to adapt the definition of reduction in such a way that uniqueness is preserved. For this, we define a *label* to be a finite binary word, i.e., an element of $\{0,1\}^*$, which we consider as a poset under the prefix order.

Definition 3.4 (Labelled net). A *labelled cell* is a cell endowed with a label, i.e., an expression of one the following 10 forms:

Multiplicative cells: $\bot[l](x)$, $⅋[l](x;y,z)$, $1[l](x)$, $\otimes[l](x;y,z)$,
Exponential cells: $?\mathrm{d}[l](x;y)$, $?\mathrm{w}[l](x)$, $?\mathrm{c}[l](x;y,z)$, $!\mathrm{d}[l](x;y)$, $!\mathrm{w}[l](x)$, $!\mathrm{c}[l](x;y,z)$,

where l ranges over labels and, as usual, x,y,z range over ports.

A *labelled net* is a multiset of labelled cells and wires in which, as usual, every port appears at most twice and, moreover:

— every labelled cell appears at most once;
— any two distinct labels are incomparable.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

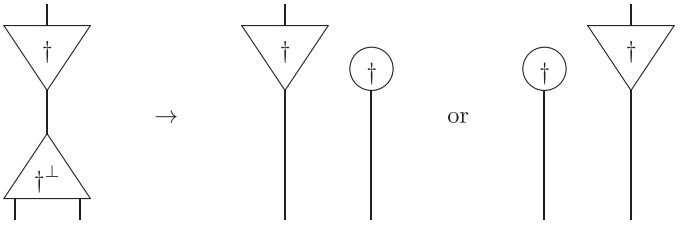


Fig. 4. Non-deterministic reduction. In the figure, † stands for ! or ?, and $!^{\perp} = ?$, $?^{\perp} = !$.

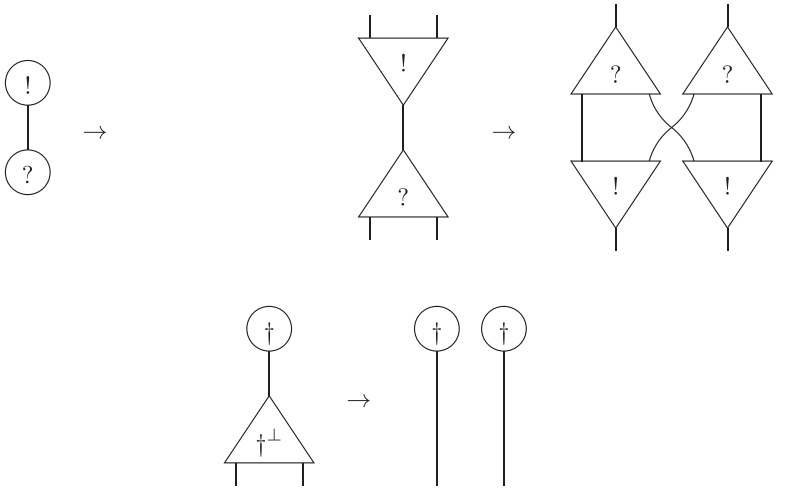


Fig. 5. Structural reduction. In the figure, † stands for ! or ?, and $!^{\perp} = ?$, $?^{\perp} = !$.

The definitions of free, bound port and of α-equivalence apply unchanged to labelled nets. For parallel composition, in forming $\mu \mid \nu$, we stipulate that the labels of μ are all prefixed with 0 and those of ν with 1 (so the operation is no longer commutative, but this has no consequence for our purposes). The notion of ω-equivalence applies as it is to labelled nets. Given a labelled net μ, we may define its *erasure* by forgetting all of its labels; this yields a net which is denoted by μ^{-}.

Definition 3.5 (Reduction of labelled nets). We modify the rules of Definition 3.3 as follows:

$$
\begin{aligned}
!\mathrm{d}[l](x;y) \mid ?\mathrm{c}[m](x;z_1,z_2) &\to \begin{cases} !\mathrm{d}[l0](z_1;y) \mid !\mathrm{w}[l1](z_2) \\ !\mathrm{w}[l0](z_1) \mid !\mathrm{d}[l1](z_2;y), \end{cases} \\
?\mathrm{d}[l](x;y) \mid !\mathrm{c}[m](x;z_1,z_2) &\to \begin{cases} ?\mathrm{d}[l0](z_1;y) \mid ?\mathrm{w}[l1](z_2) \\ ?\mathrm{w}[l0](z_1) \mid ?\mathrm{d}[l1](z_2;y), \end{cases} \\
!\mathrm{c}[l](x;y_1,y_2) \mid ?\mathrm{c}[m](x;z_1,z_2) &\to ?\mathrm{c}[m0](y_1;r,p) \mid ?\mathrm{c}[m1](y_2;s,q) \mid \\
&\quad\ !\mathrm{c}[l0](z_1;r,s) \mid !\mathrm{c}[l1](z_2;p,q), \\
!\mathrm{w}[l](x) \mid ?\mathrm{c}[m](x;y_1,y_2) &\to !\mathrm{w}[l0](y_1) \mid !\mathrm{w}[l1](y_2), \\
?\mathrm{w}[l](x) \mid !\mathrm{c}[m](x;y_1,y_2) &\to ?\mathrm{w}[l0](y_1) \mid ?\mathrm{w}[l1](y_2),
\end{aligned}
$$

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

where l, m range over labels and $l0, l1$ denote the result of appending $0, 1$, respectively, to the label l, and similarly for m. The other rules, in which some cells (and their labels) disappear, are transported to the labelled setting in the obvious way. It is clear that if μ is a labelled net which is rewritten into μ' by means of the application of one of the above rules (in any context and modulo ω-equivalence), then μ' is also a labelled net, i.e., the constraints of Definition 3.4 on labels are preserved.

We may now attach a label to each single reduction step, as follows. Let μ be a labelled net containing an active pair composed by two cells labelled by l and m, and let μ' be the net obtained from μ by reducing such an active pair. In case the active pair is of the multiplicative, communication or structural kind, we write $\mu \rightarrow_{(l,m)} \mu'$. Otherwise, if it is of the non-deterministic kind, we write $\mu \rightarrow_{(l,m,i)} \mu'$, where i is either 0 or 1, according to the choice made to obtain μ' (it is irrelevant which choice is attached to 0 and which to 1). The constraints on cell labels guarantee that each reduction step of μ or of any of its reducts has its own unique label. In the sequel, we shall use a, b, c to range over the labels of reductions.

The following result is immediate; it relates reduction of labelled nets with reduction of (unlabelled) nets:

Lemma 3.6. Let μ be a labelled net. Then, $\mu \rightarrow_a \nu$ implies $\mu^- \rightarrow \nu^-$; conversely, $\mu^- \rightarrow \mu'$ implies that there are a label a and a labelled net ν such that $\mu \rightarrow_a \nu$ and $\nu^- = \mu'$.

Definition 3.7 (Event structure of a labelled net). Let μ be a labelled net. We define

$$A_\mu = \{a \mid \mu \rightarrow^* \mu' \rightarrow_a \mu''\},$$

that is, A_μ is the set of the labels of all reductions applicable to μ and its reducts. Given $a, b \in A_\mu$, we set $a \leqslant_\mu b$ just if, for all reductions of the form $\mu \rightarrow^* \mu' \rightarrow_b \mu''$, one of the reduction steps in $\mu \rightarrow^* \mu'$ is labelled by a. Moreover, we define $a \#_\mu b$ exactly when $a = (l, m, 0)$ and $b = (l, m, 1)$, for some labels l, m. Finally, we set $\mathrm{Ev}(\mu) = (A_\mu, \leqslant_\mu, \smile_\mu)$, where $a \smile_\mu b$ just if there exist $a_0 \leqslant_\mu a$, $b_0 \leqslant_\mu b$ such that $a_0 \#_\mu b_0$ (i.e., $\smile_\mu$ is the conflict relation inherited from $\#_\mu$ through $\leqslant_\mu$, as in the statement of Lemma 2.3).

Theorem 3.8. For every labelled net μ, $\mathrm{Ev}(\mu)$ is a confusion-free event structure. Moreover, if μ, ν are such that $\mu^- = \nu^-$, then $\mathrm{Ev}(\mu)$ and $\mathrm{Ev}(\nu)$ are isomorphic.

Proof. The fact that $\leqslant_\mu$ is a partial order follows straightforwardly from its definition. The finiteness of its principal ideals is a consequence of the fact that the reductions we consider are all of finite length. That $\smile_\mu$ is symmetric is obvious. For what concerns its anti-reflexivity, $c \smile_\mu c$ would imply the existence of $a \#_\mu b$ such that $a, b \leqslant_\mu c$; but this latter condition means that all reductions starting from μ which yield the active pair inducing c must at some point reduce both a and b, which is impossible by definition of $\#_\mu$. Finally, the property that $a \smile_\mu b \leqslant_\mu c$ implies $a \smile_\mu c$ is automatically ensured by the definition of $\smile_\mu$ in terms of $\#_\mu$. Therefore, $\mathrm{Ev}(\mu)$ is indeed an event structure.

Now, the relation $\#_\mu$ is easily seen to be the immediate conflict relation of $\mathrm{Ev}(\mu)$: By definition, $a \#_\mu b$ means that a and b represent the two distinct choices of reducing the same (non-deterministic) active pair; let μ' be any reduct of μ containing such an active

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

pair; we have, by definition of reduct, $\mu \to_{c_1} \dots \to_{c_n} \mu'$ for some reduction steps $c_1, \dots, c_n$; it is then straightforward to check that $\{c_1, \dots, c_n\}$ is a configuration of $\mathrm{Ev}(\mu)$, which enables both a and b. Moreover, since a and b are induced by the same active pair, if a reduction step c is necessary for such an active pair to appear, then we will have both $c \leqslant_\mu a$ and $c \leqslant_\mu b$, which proves the absence of confusions of type II. For what concerns the absence of confusions of type I, simply observe that, since the non-deterministic choices only come in pairs, $a \#_\mu b \#_\mu c$ implies $a = c$.

The final point, i.e., that $\mathrm{Ev}(\mu)$ and $\mathrm{Ev}(\nu)$ are isomorphic, is a straightforward consequence of Lemma 3.6. □

The second part of Theorem 3.8 guarantees that, no matter how we label the cells of a net, we end up with the same event structure. Therefore, we may forget about labelled nets and speak directly of the event structure of a(n unlabelled) net μ, which we still denote by $\mathrm{Ev}(\mu)$.

3.4. *An example and some side notes*

Let us give an example. Consider the net

$$\begin{aligned} \mu \;=\; & 1(p) \mid 1(q) \mid \otimes(r; p, q) \mid \\ & !\mathrm{d}(s; r) \mid ?\mathrm{d}(s; t) \mid !\mathrm{d}(w; t) \mid ?\mathrm{d}(w; x) \mid \\ & \parr(x; y, z) \mid \bot(y) \mid \bot(z). \end{aligned}$$

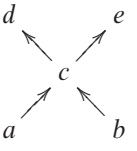
There are two active pairs in μ, both consisting of a dereliction and a codereliction. Let us call them a and b. Obviously, they are independent; after reducing them, we obtain the net

$$1(p) \mid 1(q) \mid \otimes(x; p, q) \mid \parr(x; y, z) \mid \bot(y) \mid \bot(z),$$

which contains a $\otimes/\parr$ active pair we denote by c. Reducing c gives us the net

$$1(y) \mid 1(z) \mid \bot(y) \mid \bot(z),$$

containing two further active pairs d and e which, after reducing both, yield the empty net. It is clear that a and b are both necessary for c to appear, and that this is in turn necessary for the appearance of d and e. Since there are no non-deterministic active pairs, we conclude that there are no conflicts, and $\mathrm{Ev}(\mu)$ is



where arrows denote causal precedence. In fact, we may prove that differential interaction nets are able to generate a fairly vast range of conflict-free event structures.

Let $(A, \leqslant)$ be a finite partial order. Observe that this is automatically a conflict-free event structure (indeed, *finite* conflict-free event structures and finite posets are exactly the same thing). We write $a <_1 b$ if $a < b$ and there is no c such that $a < c < b$, and we

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

define $\mathrm{succ}(a) = \{b \in A \mid a <_1 b\}$. The *out-degree* of A is the least $n \leqslant \omega$ such that, for all $a \in A$, the cardinality of $\mathrm{succ}(a)$ is at most n.

Proposition 3.9. Let $(A, \leqslant)$ be a finite partial order of out-degree at most 2. Then, there exists a differential interaction net μ such that $\mathrm{Ev}(\mu) = (A, \leqslant)$.

Proof. Throughout the proof, we use α, β to range over the cell symbols $1, \otimes$, !d, !w, and we denote by $\overline{\alpha}$ the dual cell, i.e., if α is equal to $1, \otimes$, !d, !w, then $\overline{\alpha}$ stands for $\bot, \parr$, ?d, ?w, respectively.

We shall prove by induction on the cardinality of A that there exists μ such that:

i. for each $a \in A$, μ contains exactly one pair of dual cells associated with a, of the form
$$\alpha(p; z_1, \ldots, z_m) \mid \overline{\alpha}(q; w_1, \ldots, w_m),$$
where m is the cardinality of $\mathrm{succ}(a)$;

ii. if a is minimal, then $p = q$, so the pair corresponding to a is actually an active pair, the reduction of which gives $\mu \to \mu'$ such that $\mathrm{Ev}(\mu') = (A \setminus \{a\}, \leqslant)$.

We invite the reader to check that such a net μ satisfies $\mathrm{Ev}(\mu) = (A, \leqslant)$ (it is a straightforward consequence of the definition of event structure of a net). Intuitively, μ is built from the maximal elements, proceeding 'downwards.' Each time we add a new minimal element a, we insert an active pair whose cells are connected to the cells corresponding to the elements b which are immediately above a. In doing this, if necessary we 'break up,' the active pair corresponding to b, so that it needs the execution of the active pair corresponding to a to 'become' an active pair again.

Let us start the proof. If $A = \varnothing$, then $\mu = *$ will do. Assume that $A \neq \varnothing$, and let a be a minimal element (which must exist by the finiteness of A). Let μ_0 be a net such that $\mathrm{Ev}(\mu_0) = (A \setminus \{a\}, \leqslant)$, which exists by induction hypothesis. We have now three different cases, depending on the cardinality of $\mathrm{succ}(a)$.

If a has no immediate successor, then it is easy to check that $\mu = \mu_0 \mid !\mathrm{w}(x) \mid ?\mathrm{w}(x)$ satisfies the required properties.

Suppose now that $\mathrm{succ}(a) = \{b\}$. By induction hypothesis, we know that there exists a net ν such that
$$\mu_0 = \nu \mid \beta(p; \vec{z}) \mid \overline{\beta}(q; \vec{w}),$$
where the pair $\beta/\overline{\beta}$ is the one associated with b. Then, we set
$$\mu = \nu \mid \beta(p; \vec{z}) \mid \overline{\beta}(r; \vec{w}) \mid !\mathrm{d}(x; q) \mid ?\mathrm{d}(x; r).$$
Let us check that μ meets the requirements. It clearly satisfies (i), with the !d/?d active pair being the one associated with a. For what concerns (ii), let a' be a minimal element of A. If $a' = a$, then we have $\mu \to \mu_0$, so we conclude by induction hypothesis. If $a' \neq a$, observe that in any case $a' \neq b$ (because b is certainly not minimal in A), so the active pair associated with a' is in ν, and was therefore already in μ_0, so we conclude once again by invoking the induction hypothesis.

The last case is that in which $\mathrm{succ}(a)$ has cardinality 2. This is treated analogously to the previous case, except that we introduce a $\otimes/\parr$ active pair. □

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

By inspecting the proof of Proposition 3.9, it is clear how differential interaction nets may be modified so as to generate *all* finite conflict-free event structures. It suffices to replace the cells 1, !d, $\otimes$ with a family of cells of the form $\mathbf{M}_n(x; y_1, \ldots, y_n)$, with $n \in \mathbb{N}$, and similarly for their dual cells, which are replaced by the family $\mathbf{M}_n^*(x; y_1, \ldots, y_n)$. The interaction between these cells is defined by

$$\mathbf{M}_n(x; y_1, \ldots, y_n) \mid \mathbf{M}_n^*(x; z_1, \ldots, z_n) \quad \rightarrow \quad y_1 \leftrightarrow z_1 \mid \cdots \mid y_n \leftrightarrow z_n.$$

Quite obviously, the interaction rules for $1/\bot$, !d/?d, $\otimes/\parr$ are the special cases where $n = 0, 1, 2$, respectively. From the (linear) logical point of view, allowing these generalized cells corresponds to considering proof-nets in which the connectives $\otimes$ and $\parr$ have arbitrary arity.

This generalization of Proposition 3.9 shows that the reduction dynamics of Lafont interaction nets (see Section 6.1) is able to generate all finite conflict-free event structures. Since conflict-free event structures may be thought of as an abstract model of deterministic asynchronous computation, this result supports the following statement from Lafont (1997):

> [...] we can say that [Lafont] interaction nets are a deterministic and asynchronous model of computation. In fact, we think that any computation of that kind can be modeled by means of [Lafont] interaction nets, but of course, an assertion of this kind cannot be proved.

Moreover, if we write $x\vec{y}$ for $\mathbf{M}_n(x; \vec{y})$ and $\overline{x}\vec{y}$ for $\mathbf{M}_n^*(x; \vec{y})$, then it is clear how these generalized cells are nothing but *solos* as defined in Laneve and Victor (2003). Indeed, the interaction rule above is consistent with that of the solos calculus, except that it introduces *explicit fusions* (Wischik and Gardner 2005). The fact that multiplicative proof-nets (with generalized arities) may be seen as a (deterministic) fragment of the solos calculus was already remarked in Laneve *et al.* (2001).

4. On the existence of encodings of process calculi into differential interaction nets

We now come to the central question of the paper: Are differential interaction nets sufficiently expressive to be considered as a model of concurrent computation? We first explain why, as mentioned in the introduction, all process calculi usually employed in concurrency theory admit the presence of confusion.

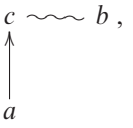
Consider the following CCS process, which is also trivially a π-calculus process or a fusion calculus process (Parrow and Victor 1998):

$$P = \nu(x, y)(\overline{x} \mid x.\overline{y} \mid y \mid \overline{y}).$$

The standard event structure semantics of P (originally defined in Winskel (1982) and reviewed, for example, in Winskel and Nielsen (1995)) consists of three events: an event a representing the synchronization on channel x, an event b representing the synchronization on channel y immediately available in P and an event c representing the 'other' synchronization on y which may occur after the event a. Graphically, the

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

relationship between these three events may be represented by



where the 'wiggly' line represents immediate conflict, so the event structure has a confusion of type II (it is indeed the smallest of this kind).

Note that name restriction is present in P only to make its event structure be as small as possible; it is not needed for generating confusion. Indeed, if we consider the reductions of P, the presence or absence of restriction is irrelevant. Therefore, confusion really arises only through parallel composition and prefixing. Although the event structure semantics of more sophisticated process calculi, especially those involving mobility, is far from trivial (for instance, it is only recently that Crafa *et al.* (2012) have succeeded in defining an extension of Winskel's CCS semantics for the π-calculus), for fragments so simple as to include only parallel composition and prefixing there is no difficulty, even in presence of mobility features, in defining a semantics of reduction which extends Winskel's CCS interpretation in a natural way, so that the presence of confusion in virtually all process calculi may be asserted not just at an intuitive but also at a technical level.

4.1. *Bisimilar embeddings of event structures*

Let E, E' be two event structures, and let $R \subseteq |E| \times |E'|$. We denote by $\pi_1(R), \pi_2(R)$ the first-component and second-component projections of R, respectively. If $u \in \mathcal{C}(E)$, $a \in |E|$ is enabled by u, and $v = u \cup \{a\}$, we write $u \xrightarrow{a}_R v$ if $a \in \pi_1(R)$ (we call this a *computational transition* labelled by a), and $u \xrightarrow{\tau}_R v$ otherwise (we call this an *administrative transition*). We denote by $\Longrightarrow_R$ the reflexive–transitive closure of $\xrightarrow{\tau}_R$, and we write $u \overset{a}{\Longrightarrow}_R v$ iff there exist $u_1, v_1 \in \mathcal{C}(E)$ such that $u \Longrightarrow_R u_1 \xrightarrow{a}_R v_1 \Longrightarrow_R v$. We apply the same notations to E', with π_2 replacing π_1, i.e., we write $u' \xrightarrow{a'}_R v'$ if $a' \in \pi_2(R)$, and $u' \xrightarrow{\tau}_R v'$ otherwise, with $u', v' \in \mathcal{C}(E')$ and $a' \in |E'|$. Moreover, given $u \in \mathcal{C}(E)$ and $u' \in \mathcal{C}(E')$, we set $\mathrm{supp}_R(u) = u \cap \pi_1(R)$ and $\mathrm{supp}_R(u') = u' \cap \pi_2(R)$.

In what follows, we denote by $\mathcal{P}_{\mathrm{fin}}(A)$ the set of finite subsets of a set A.

Definition 4.1 (R-bisimulation). Let $E = (|E|, \leqslant, \smile), E' = (|E'|, \leqslant', \smile')$ be event structures, and let $R \subseteq |E| \times |E'|$. An *$R$-bisimulation* between E and E' is a relation $\mathcal{B} \subseteq \mathcal{C}(E) \times \mathcal{P}_{\mathrm{fin}}(R) \times \mathcal{C}(E')$, such that $(\varnothing, \varnothing, \varnothing) \in \mathcal{B}$ and, whenever $(u, \phi, u') \in \mathcal{B}$, we have

i. ϕ is (the graph of) a poset isomorphism between $(\mathrm{supp}_R(u), \leqslant)$ and $(\mathrm{supp}_R(u'), \leqslant')$;
ii. $u \xrightarrow{a}_R v$ implies $u' \overset{a'}{\Longrightarrow}_R v'$ with $(v, \phi \cup \{(a, a')\}, v') \in \mathcal{B}$;
iii. $u \xrightarrow{\tau}_R v$ implies $u' \Longrightarrow_R v'$ with $(v, \phi, v') \in \mathcal{B}$;
iv. $u' \xrightarrow{a'}_R v'$ implies $u \overset{a}{\Longrightarrow}_R v$ with $(v, \phi \cup \{(a, a')\}, v') \in \mathcal{B}$;
v. $u' \xrightarrow{\tau}_R v'$ implies $u \Longrightarrow_R v$ with $(v, \phi, v') \in \mathcal{B}$.

We say that E and E' are R-bisimilar, and we write $E \approx_R E'$, if there exists an R-bisimulation between them.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

R-bisimulations are a variant of history-preserving bisimulations. These were originally defined on *labelled* event structures. Our definition is a generalization of this: If $\ell_E, \ell_{E'}$ are labelling functions for E, E', such that some events are assigned the special label τ, we may set $R_\ell = \{(a, a') \in |E| \times |E'| \mid \ell_E(a) = \ell_{E'}(a') \neq \tau\}$, and we have $E \approx_{R_\ell} E'$ exactly when E and E' are weakly bisimilar in the original definition of Rabinovitch and Traktenbrot (1988) and van Glabeek and Goltz (1989).

Of course, the meaningfulness of an R-bisimulation depends on R: For example, we invite the reader to check that, for all even structures E, E', $\{(u, \varnothing, u') \mid u \in |E|, u' \in |E'|\}$ is a $\varnothing$-bisimulation. The idea is to avoid this kind of degeneracy by considering a special case of R-bisimulations, in which R is 'as big as possible.'

Definition 4.2 (Bisimilar embedding). Let E, E' be two event structures. A *bisimilar embedding* of E into E' is a relation $\iota \subseteq |E| \times |E'|$ such that

totality: $\pi_1(\iota) = |E|$;
injectivity: for all $a, b \in |E|$, $\iota(a) \cap \iota(b) \neq \varnothing$ implies $a = b$;
bisimilarity: $E \approx_\iota E'$; a ι-bisimulation proving this is said to be *associated with* ι.

We write $E \overset{\iota}{\hookrightarrow} E'$ to denote the fact that ι is an embedding of E into E', or simply $E \hookrightarrow E'$ to state the existence of an embedding.

Proposition 4.3 (Embeddings compose). If $E \overset{\iota'}{\hookrightarrow} E'$ and $E' \overset{\iota''}{\hookrightarrow} E''$, then $E \overset{\iota'' \circ \iota'}{\hookrightarrow} E''$, where $\circ$ denotes standard composition of relations.

Proof. One checks that, if $\mathcal{B}', \mathcal{B}''$ are bisimulations associated with ι', ι'', respectively, then $\{(u, \phi, u'') \mid (u, \phi_1, u') \in \mathcal{B}', (u', \phi_2, u'') \in \mathcal{B}'', \phi = \phi_2 \circ \phi_1\}$ is a bisimulation associated with $\iota'' \circ \iota'$. □

Note that E can be embedded into E' precisely when, once we consider the events of E to be labelled by themselves, there is a way of labelling the events of E' over $|E| \cup \{\tau\}$ so that E and E' are weakly history-preserving bisimilar in the sense of Rabinovitch and Traktenbrot (1988), van Glabeek and Goltz (1989).

Although there is no general agreement on what an embedding of concurrent calculi should be, there is a certain number of desirable properties which are recurrently adopted in the literature. For instance, Gorla (2010) isolates five criteria yielding a sensible notion of encoding on 'abstract' concurrent systems (sets of processes with a reduction relation and a behavioural equivalence), which is general enough to encompass a great deal of known encodability results, both positive and negative. Amongst these criteria, there are two which are of special interest to us:

Preservation of distribution: The encoding should preserve the degree of distribution of the source process; technically, on process calculi one usually imposes that the encoding be a homomorphism of parallel composition.

Operational correspondence: The target process should simulate the source process in a faithful way, i.e., it should be

- **complete:** if P reduces to P', then the encoding of P should reduce to something equivalent to the encoding of P';

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

sound: if the encoding of P reduces to Q, then such a Q is not just anything: There must exist P' such that P reduces to P' and Q is able to reduce to something equivalent to the encoding of P'.

Bisimilar embeddings are conceived with the above two properties in mind. Indeed, the requirement that the embedding induces a bisimulation is exactly the principle of operational correspondence formulated on events, which are morally reductions. Preservation of distribution is ensured by the non-interleaving character of event structures, which translates technically as the fact that if two configurations are part of a bisimulation, then they are order isomorphic. Typically, this will detect the presence of coordinating agents inserted by a non-homomorphic encoding.

Of the other three criteria of Gorla (2010), two (*name invariance* and *success sensitiveness*) are specific to name-passing (or name-based) calculi and/or are hard to formulate in the abstract context of event structures. The third one, *divergence reflection*, i.e., the fact that the encoding does not introduce divergence, is perhaps less agreed upon in the literature. We shall consider it in Section 6.3. In any case, since Theorem 4.6 rests only on preservation of distribution and operational correspondence, it would obviously hold if more stringent notions of embedding (incorporating these additional criteria) were to be introduced.

We hope to have thus convinced the reader that bisimilar embeddings provide an abstract way, formulated directly at the level of event structures, to speak of the minimal properties which a reasonable encoding of process calculi should satisfy.

4.2. *Bisimilar embeddings preserve confusion*

In this section, we fix two generic event structures $E = (|E|, \leqslant, \smile)$ and $E' = (|E'|, \leqslant', \smile')$.

Lemma 4.4. Let $E \overset{\iota}{\hookrightarrow} E'$, and let $d, e \in |E|$. Then, $d \smile e$ implies that, for all $d' \in \iota(d)$ and $e' \in \iota(e)$, $d' \smile' e'$.

Proof. An immediate consequence of the properties of ι-bisimulations. □

Let us introduce a useful terminology. Suppose that $E \overset{\iota}{\hookrightarrow} E'$, and let $e' \in |E'|$ and $u' \in \mathcal{C}(E')$. We say that u' *eventually enables* e' if $u' \Longrightarrow_\iota v'$ such that v' enables e'.

Lemma 4.5. Let $E \overset{\iota}{\hookrightarrow} E'$, let $d' \smile' e'$, and let $u' \in \mathcal{C}(E')$ enabling d' and eventually enabling e'. Then, there exists $e'_0 \leqslant' e'$ such that $d' \#' e'_0$.

Proof. Lemma 2.3 gives us two events d'_0, e'_0 of E' such that $d' \geqslant' d'_0 \#' e'_0 \leqslant' e'$. We contend that $d'_0 = d'$. Indeed, suppose, for the sake of contradiction, that $d'_0 <' d'$. We then have $d'_0 \in u'$, because u' enables d, but $d'_0 \smile' e'$ by propagation of conflict, in contradiction with our hypothesis that u' eventually enables e'. □

Theorem 4.6. Let E, E' be event structures with E containing a confusion. Then, $E \hookrightarrow E'$ implies that also E' contains a confusion.

Proof. Let ι be an embedding of E into E', and let $\mathcal{B}$ be an associated ι-bisimulation. Suppose there is a pair (a, b) which is a confusion of type II in E, and assume w.l.o.g. that

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

there exists $c \in \lceil a \rceil \setminus \lceil b \rceil$. Note that $u = \lceil a \rceil \cup \lceil b \rceil$ is necessarily a configuration of E; hence, there must exist $u'_0 \in \mathcal{C}(E')$ such that $(u, \phi, u'_0) \in \mathcal{B}$, with ϕ a poset isomorphism. Now, since u enables a, we must have $u'_0 \Longrightarrow_\iota u'$ such that u' enables $a' \in \iota(a)$. Moreover, since all events in that transition are administrative, we have $(u, \phi, u') \in \mathcal{B}$. But u also enables b, so u' must eventually enable an event $b' \in \iota(b)$. Observe that, by Lemma 4.4, we have $a' \smile' b'$, so we are in position of applying Lemma 4.5, which gives us an event $b'_0 \leqslant' b'$ such that $a' \#' b'_0$. Let $u' \overset{a'}{\Longrightarrow} v'_a$ and $u' \overset{b'}{\Longrightarrow} v'_b$ be two transitions simulating the transitions $u \xrightarrow{a} u \cup \{a\}$ and $u \xrightarrow{b} u \cup \{b\}$, respectively, i.e., such that $(u \cup \{a\}, \phi \cup \{(a, a')\}, v'_a), (u \cup \{b\}, \phi \cup \{(b, b')\}, v'_b) \in \mathcal{B}$. Since $\phi \cup \{(a, a')\}$ and $\phi \cup \{(b, b')\}$ are poset isomorphisms, we have that $c \leqslant a$ and $c \not\leqslant b$ imply $\phi(c) \leqslant' a'$ and $\phi(c) \not\leqslant' b'$, which implies $\phi(c) \not\leqslant' b'_0$, so (a', b'_0) is a confusion of type II in E'.

Suppose now there is no confusion of type II in E, and let (a, b, c) be a confusion of type I. By absence of type II confusion, we have $\lceil a \rceil = \lceil b \rceil = \lceil c \rceil = u$, and there must exist $(u, \phi, u'_0) \in \mathcal{B}$ such that $u'_0 \Longrightarrow_\iota u'$ with u' enabling an event $b' \in \iota(b)$, because u enables b. Again, since all events in that transition are administrative, we actually have $(u, \phi, u') \in \mathcal{B}$. But u also enables the simultaneous occurrence of a and c, so u' must eventually enable some $a' \in \iota(a)$ and $c' \in \iota(c)$ satisfying $a' \frown' c'$. Additionally, Lemma 4.4 gives us $a' \smile' b' \smile' c'$. Observe that we are now in position of applying Lemma 4.5 twice: with $d' = b'$ and $e' = a'$, and with $d' = b'$ and $e' = c'$. This gives us $a'_0 \leqslant' a'$ and $c'_0 \leqslant' c'$ such that (a'_0, b', c'_0) is a confusion of type I in E'. □

5. About Ehrhard and Laurent's encoding

Ehrhard and Laurent (2010a) exhibited an encoding of a fragment of the solos calculus, called *acyclic solos*, into differential interaction nets. Acyclic solos are expressive enough to encode the π-calculus (the image of the encoding of Laneve and Victor (2003) is contained in the acyclic fragment), so by composing the two encodings, and simplifying, Ehrhard and Laurent (2010b) were also able to show directly how the π-calculus may be encoded in differential interaction nets. These results are apparently in contradiction with our claims of Section 4, a situation which requires an explanation.

First of all, let us recall the results of Ehrhard and Laurent (2010a), from which those of Ehrhard and Laurent (2010b) follow. As in the present paper, Ehrhard and Laurent concentrate on the *reduction* of solos processes, and their goal is to show that it may be faithfully simulated in differential interaction nets. To express 'faithfulness' at a technical level, Ehrhard and Laurent opt for bisimulations, a standard choice in concurrency theory. However, since reduction is not labelled, they make a technical adjustment: They consider processes in which each solo has a unique label, much like we did in Definition 3.4, so that a reduction $P \to Q$ resulting from the synchronization of an input solo labelled by l with an output solo labelled by m may be written $P \xrightarrow{l\overline{m}} Q$. In this way, one may define a labelled transition system $\mathbb{S}$ whose states are *acyclic* solo processes and whose transitions are the ones just described (the definition of acyclic process is irrelevant here; it is enough to know that, as mentioned above, the acyclicity condition does not compromise the expressiveness of the solos calculus).

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

The next step is to endow also differential interaction nets with labels. However, unlike our Definition 3.4, Ehrhard and Laurent only label certain dereliction and codereliction cells; all other cells (including some derelictions and coderelictions) are left unlabelled. This greatly simplifies reduction of labelled nets, because it is no longer necessary, as in Definition 3.5, to append 0's and 1's to labels in order to preserve their uniqueness: Labels are just left unchanged in non-deterministic reductions and, as in Definition 3.5, disappear in communication reductions. At this point, Ehrhard and Laurent give the following definitions:

Definition 5.1 (Ehrhard and Laurent 2010a). We write $\mu \rightsquigarrow_{l,m} \nu$ if $\mu \to \nu$ by means of one of the following:

— a multiplicative reduction;
— a communication reduction involving two unlabelled cells;
— a non-deterministic reduction involving a cell labelled by l or m;
— a structural reduction.

We define the labelled transition system $\mathbb{D}_{\mathrm{EL}}$ whose states are nets and in which there is a transition $\mu \xrightarrow{l\overline{m}} \nu$ exactly when $\mu \rightsquigarrow^*_{l,m} \mu' \to \nu$, where the last step is a communication reduction involving a codereliction labelled by l and a dereliction labelled by m.

The main result of Ehrhard and Laurent (2010a) is the definition of a relation $P \leftrightsquigarrow \mu$, to be read 'the net μ is an encoding of the process P,' having the following property:

Theorem 5.2 (Ehrhard and Laurent 2010a). The relation $\leftrightsquigarrow$ is a bisimulation between $\mathbb{S}$ and $\mathbb{D}_{\mathrm{EL}}$, up to bisimilarity in $\mathbb{D}_{\mathrm{EL}}$. (More precisely, if we denote by $\sim_d$ the equivalence relation on labelled nets induced by the union of multiplicative reduction, structural reduction and unlabelled communication reduction, then $\sim_d$ may be proved to be a bisimulation in $\mathbb{D}_{\mathrm{EL}}$, such that $\leftrightsquigarrow\sim_d$ is a bisimulation between $\mathbb{S}$ and $\mathbb{D}_{\mathrm{EL}}$).

The key point is understanding the definition of $\mathbb{D}_{\mathrm{EL}}$, which is somewhat fishy. In fact, an arguably more natural labelled transition system for nets would be the following:

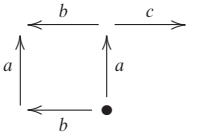
Definition 5.3. We define the labelled transition system $\mathbb{D}$ whose states are nets and in which there is a transition $\mu \xrightarrow{\alpha} \nu$ exactly when $\mu \to \nu$, the label α being determined as follows: If the reduction step is a communication involving a codereliction labelled by l and a dereliction labelled by m, then $\alpha = l\overline{m}$; otherwise, $\alpha = \tau$.

However, Theorem 5.2 fails if we replace $\mathbb{D}_{\mathrm{EL}}$ with $\mathbb{D}$. A counterexample may be found, as expected, by considering a solos process exhibiting confusion, for instance, a process P whose behaviour is identical to the CCS process $\nu(x,y)(\overline{x} \mid x.\overline{y} \mid y \mid \overline{y})$ already discussed in the opening of Section 4.[†] If l_0 is the label of the output on x, l the label of the output on y, m_0 the label of the input on x, m_1, m_2 the labels of the two inputs on y, and if we set $a = l_0\overline{m_0}$, $b = l\overline{m_1}$, $c = l\overline{m_2}$, the labelled transition system of P within $\mathbb{S}$ is the following,

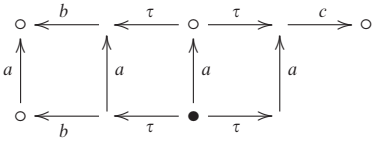
[†] In the solos calculus, we may take $P = \nu(x,y,z)(\overline{x}y \mid xz \mid \overline{z} \mid y \mid \overline{y})$, but the CCS description is more readable.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

which we denote by $\mathbb{T}_P$:



On the other hand, it is possible to show that, whenever $P \leftrightarrow \mu$, the labelled transition system of μ within $\mathbb{D}$ is the following, which we denote by $\mathbb{T}_\mu$:



Both in $\mathbb{T}_P$ and $\mathbb{T}_\mu$ we denoted by $\bullet$ the initial state. Although $\mathbb{T}_\mu$ does simulate $\mathbb{T}_P$, the two are obviously not bisimilar, not even up to bisimilarity in $\mathbb{T}_\mu$. The problem is that the non-deterministic choice between b and c appearing when we reduce P through a is already present in μ (represented by the pair of τ transitions) and it is 'blind,' i.e., is independent of a. This misbehaviour is a consequence of the absence of confusion in μ and is predicted by Theorem 3.8. The definition of $\mathbb{D}_{\mathrm{EL}}$ solves the problem by 'compressing' $\mathbb{T}_\mu$ in such a way that only the states marked by $\circ$ are attainable, yielding a labelled transition system which is bisimilar (indeed, isomorphic) to $\mathbb{T}_P$.

The aforementioned notion of 'compression' may be generalized as follows:

Definition 5.4 (Compression). Let $\mathbb{T}$ be a labelled transition system, let S be a state of $\mathbb{T}$, and let α be a label other than τ. The α-range of S in $\mathbb{T}$, denoted by $\mathbb{T}_\alpha(S)$, is the set of all states T of $\mathbb{T}$ such that $S \Longrightarrow\!\!\!\!\stackrel{\alpha}{\longrightarrow} T$, where, as usual, $\Longrightarrow$ denotes the reflexive–transitive closure of $\stackrel{\tau}{\longrightarrow}$.

Let $\mathbb{T}'$ be a labelled transition system with the same states as $\mathbb{T}$ and containing no τ transition. We say that $\mathbb{T}'$ is a *compression* of $\mathbb{T}$ if, for every state S and every label $\alpha \neq \tau$, we have

— $\mathbb{T}'_\alpha(S) \subseteq \mathbb{T}_\alpha(S)$;
— $\mathbb{T}'_\alpha(S) = \varnothing$ implies $\mathbb{T}_\alpha(S) = \varnothing$.

The second condition excludes the trivial compression, i.e., a system with no transitions at all, as soon as $\mathbb{T}$ has at least one visible transition. Therefore, there is in general no smallest compression, but there is always the largest, namely the system $\mathbb{T}'$ in which $\mathbb{T}'_\alpha(S) = \mathbb{T}_\alpha(S)$ for all S and all α.

It is obvious that $\mathbb{D}_{\mathrm{EL}}$ is a compression of $\mathbb{D}$. It is not the largest; indeed, the example discussed above shows that Theorem 5.2 would fail if we took the largest compression of $\mathbb{D}$ in place of $\mathbb{D}_{\mathrm{EL}}$ (we invite the reader to check that the largest compression of $\mathbb{T}_\mu$ is not bisimilar to $\mathbb{T}_P$). Instead, $\mathbb{D}_{\mathrm{EL}}$ looks rather like a 'minimal' compression: We only consider transitions $\mu \stackrel{\alpha}{\longrightarrow} \nu$ such that $\mu \Longrightarrow \mu' \stackrel{\alpha}{\longrightarrow} \nu$ in $\mathbb{D}$ and the reduction $\mu \Longrightarrow \mu'$ is 'minimal' in the sense that it does not contain τ transitions which are unnecessary to attain the action α (*cf.* Definition 5.1). As a matter of fact, one may abstractly define a notion of 'optimal compression' based on minimizing the length of the sequences of τ

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

transitions leading to visible actions, and it is true that $\mathbb{D}_{\mathrm{EL}}$ is an optimal compression of $\mathbb{D}$. However, neither is this optimal compression unique in general, nor does it seem to capture what actually goes on in Definition 5.1, which looks quite *ad hoc*.

In light of the above discussion, even though we are lacking a general construction relating $\mathbb{D}_{\mathrm{EL}}$ and $\mathbb{D}$, the value of Theorem 5.2 appears to be doubtful. On one side, we have a labelled transition system $\mathbb{S}$ which accurately reflects the reduction semantics of the solos calculus, in the sense that each reduction step gives rise to a transition, and vice versa. The same cannot be said about $\mathbb{D}_{\mathrm{EL}}$ on the other side; if we really want to describe the reduction of nets with the same accuracy as for solos processes, the system $\mathbb{D}$ arguably does a much better job. Therefore, the least we may say is that Theorem 5.2 is 'biased' in favour of differential interaction nets. On the other hand, the fact that the theorem fails when this 'bias' is eliminated cannot but confirm the value of the negative results of Section 4, especially because a counterexample is found as soon as we consider a non-confusion-free process.

The reader will have noticed how the event structures semantics introduced above does not play any role in the objections to Ehrhard and Laurent's encoding. In fact, these appear already on the basis of a labelled transition system semantics. In this respect, we believe that event structures are a step forward in pointing out the problem and its essential nature, independently of specific encodings (as far as they are reasonable). Indeed, the confusion-freeness of differential interaction nets is arguably a precise (if abstract) description of the nature of their non-determinism, and we are currently unable to provide such a description using other semantics (such as labelled transition systems).

6. Other systems of interaction nets

6.1. *Interaction net systems*

Differential interaction nets are only a particular case of a more general notion of *interaction net system.*

Definition 6.1 (Interaction net system). An *alphabet* is a pair $\Sigma = (|\Sigma|, \deg)$, where $|\Sigma|$ is a set and $\deg : |\Sigma| \to \mathbb{N}$ is the *degree function.*

A *cell* on the alphabet Σ is an expression of the form $\alpha(x_1, \ldots, x_n)$, where $\alpha \in |\Sigma|$, $x_1, \ldots, x_n$ are ports and $n = \deg(\alpha)$. The notions of *wire* and *net* over the alphabet Σ are defined just as in Definition 3.1, *mutatis mutandis.* All the other definitions (free and bound ports, α-equivalence, parallel composition, ω-equivalence) apply verbatim.

If $\alpha, \beta \in |\Sigma|$, $m = \deg(\alpha)$, $n = \deg(\beta)$, $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$, we define an (α_i, β_j)-*active pair* to be a net of the form

$$\alpha_i \bowtie \beta_j = \alpha(x_1, \ldots, z, \ldots, x_m) \mid \beta(y_1, \ldots, z, \ldots, y_n),$$

where the bound port z is the ith port in the α cell and the jth port in the β cell, and all other ports are free. An *interaction rule* on the alphabet Σ is a rewriting rule of the form

$$\alpha_i \bowtie \beta_j \to \nu$$

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

where v is a net such that $\mathrm{fp}(v) = \mathrm{fp}(\alpha_i \bowtie \beta_j)$. In case $\alpha = \beta$ and $i = j$, we say that the above rule is a *self-rule*, and we say that it is *symmetric* if v is invariant under the renaming that exchanges the ports x_1 and y_1, x_2 and y_2, and so on.

An *interaction net system* is a pair $(\Sigma, \mathcal{R})$ where Σ is an alphabet and $\mathcal{R}$ is a set of interaction rules on Σ. Given $\alpha \in |\Sigma|$, we say that the ith port of α is *principal* if $\mathcal{R}$ contains a rule whose left member is $\alpha_i \bowtie \beta_j$, for some $\beta \in |\Sigma|$. An interaction net system is *multiport* if it has a symbol with more than one principal port; otherwise, it is *uniport*. An interaction net system is *unambiguous* if $\mathcal{R}$ contains at most one rule for each active pair and every self-rule is symmetric; it is *ambiguous* otherwise. A *Lafont system* is an unambiguous, uniport interaction net system.

Differential interaction nets are thus an example of ambiguous uniport interaction net system (also known in the literature as a *multirule* system (Alexiev 1999)). Multiplicative nets, i.e., differential interaction nets restricted to the symbols $\bot$, $⅋$, 1 and $\otimes$, are the prototypical example of Lafont system (it is indeed this system that suggested to Lafont (1990), his definition of interaction nets). Another example is the system with the symbols $\mathbf{M}_n, \mathbf{M}_n^*$ discussed after Proposition 3.9, which generalizes multiplicative proof-nets.

To improve readability, especially in the case of uniport systems, it is convenient to assume principal ports to be always the 'leftmost' in the list of ports of a cell, and to use the notation $\alpha(x_1, \ldots, x_m; y_1, \ldots, y_n)$ for a cell whose symbol α is of degree $m + n$ and has m principal ports, which are $x_1, \ldots, x_m$. We already adopted this notation in Definition 3.1.

6.2. *Interaction nets and event structures*

Given a net μ of a generic interaction net system, it is possible to associate with it an event structure $\mathrm{Ev}(\mu)$ which describes the dynamics of its reduction, much as what we did for differential interaction nets in Section 3.3. Intuitively, an event is an active pair appearing during the reduction of μ, together with a choice of how to reduce it (in case the system is ambiguous); an event a causally precedes b if the active pair yielding b only appears after the occurrence of a; and a and b are in conflict when they overlap, i.e., the active pairs inducing them have at least one cell in common.

Although the above idea is simple and easy to grasp, formalizing it for interaction nets in their full generality requires non-trivial techniques from rewriting theory, which have been developed in the literature in several different settings, including Mazurkiewicz traces (Mazurkiewicz 1986), domains (Nielsen *et al.* 1981), concurrent transition systems (Stark 1989), concurrent automata (Boldi *et al.* 1993), term rewriting (Clark and Kennaway 1996), abstract rewriting systems (Khasidashvili and Glauert 2005), graph rewriting (Baldan *et al.* 2007) and asynchronous graphs (Melliès 2004; Mimram 2008). It is out of the scope of this paper to give an account of such techniques; for our purposes, the above informal description will be just enough.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

Let us give an example. Consider the alphabet consisting of four symbols $\mathbf{0}, \mathbf{1}, \iota, \chi$ of respective degree $1, 1, 2, 4$, and consider the interaction rules

$$\begin{aligned} \iota(x;y) \mid \iota(x,z) &\rightarrow y \leftrightarrow z, \\ \chi(x,y;s,t) \mid \alpha(x) &\rightarrow \alpha(s) \mid y \leftrightarrow t, \\ \chi(x,y;s,t) \mid \alpha(y) &\rightarrow \alpha(s) \mid x \leftrightarrow t \end{aligned}$$

with $\alpha \in \{\mathbf{0}, \mathbf{1}\}$. Observe that the system thus defined, although multiport, is unambiguous. A cell of type χ may be seen as a server handling two concurrent requests (on ports x, y) from cells of type $\mathbf{0}$ and $\mathbf{1}$, and granting access to port s to the cell that interacts first. The interaction between two cells of type ι may be seen as the abstraction of some internal action (like a τ transition).

Consider now the net

$$\mu = \chi(x,y;s,t) \mid \iota(w;x) \mid \iota(w;z) \mid \mathbf{0}(z) \mid \mathbf{1}(y).$$

The net μ has two active pairs: the two ι cells may interact on port w, and the $\mathbf{1}$ cell may interact with the χ cell on its second principal port. This gives us two causally unrelated events, which we call a and b, respectively. The occurrence of a gives us the net

$$\chi(x,y;s,t) \mid x \leftrightarrow z \mid \mathbf{0}(z) \mid \mathbf{1}(y) \equiv \chi(x,y;s,t) \mid \mathbf{0}(x) \mid \mathbf{1}(y),$$

in which the event b is still available, showing that a and b are not in conflict. We also witness the appearance of a third event c, corresponding to the interaction between the $\mathbf{0}$ cell and the first principal port of the χ cell. Now, this event c is in conflict with b, because they both concern the χ cell, so that one excludes the other: If we choose b, we obtain the net $\mathbf{1}(s) \mid \mathbf{0}(t)$, if we choose c, we obtain $\mathbf{0}(s) \mid \mathbf{1}(t)$. The event structure $\mathrm{Ev}(\mu)$ is then exactly the same as that of the process P discussed in the opening of Section 4. Therefore, unambiguous multiport interaction nets are able to express confusion.

In fact, unambiguous multiport interaction nets systems may be shown to be able to encode several standard process calculi, such as the π-calculus and the solos calculus, in a sense compatible with our bisimilar embeddings. This is part of an ongoing work with Andrei Dorman, which will be the subject of future publication (Dorman 2013). Here, we concentrate on the internal theory of interaction nets and prove an interesting result about the existence of universal systems of multiport interaction nets.

6.3. *On the existence of universal systems of multiport interaction nets*

Lafont (1997) introduced a notion of translation between his interaction net systems (that is, unambiguous and uniport). We recall it briefly. A *principal net* is a net $\pi(x;\vec{y})$ whose free ports are $x, y_1, \ldots, y_n$ (we use $\vec{y}$ as a shorthand for the list $y_1, \ldots, y_n$) such that exactly x appears as the principal port of a cell. A *pre-translation* from a Lafont system $\mathcal{S}$ to a Lafont system $\mathcal{T}$ is a map Φ assigning to every cell $\alpha(x;\vec{y})$ of $\mathcal{S}$ a principal net $\Phi(\alpha)(x;\vec{y})$. A pre-translation may obviously be lifted to a map from nets of $\mathcal{S}$ to nets of $\mathcal{T}$ by applying it homomorphically with respect to parallel composition and making it behave as the

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

identity on wires:

$$\Phi(\alpha_1(x_1;\vec{y}_1) \mid \cdots \mid \alpha_c(x_c;\vec{y}_c) \mid \vec{z} \leftrightarrow \vec{w}) \;=\; \Phi(\alpha_1)(x_1;\vec{y}_1) \mid \cdots \mid \Phi(\alpha_c)(x_c;\vec{y}_c) \mid \vec{z} \leftrightarrow \vec{w},$$

where we used the shorthand $\vec{z} \leftrightarrow \vec{w} = z_1 \leftrightarrow w_1 \mid \cdots \mid z_k \leftrightarrow w_k$. A *translation* is a pre-translation such that, whenever $\alpha \bowtie \beta \to v$ is a rule of $\mathcal{S}$ (we omit the subscripts in the active pair, because all cells have exactly one principal port), we have $\Phi(\alpha \bowtie \beta) \to^* \Phi(v)$ in $\mathcal{T}$.

Observe that, by definition, a Lafont translation preserves the degree of distribution. Moreover, thanks to the fact that Lafont systems are strongly confluent, it also ensures operational correspondence, so it fits the criteria required by bisimilar embeddings.

Lafont (1997) went on to prove the existence of a *universal system* with respect to this notion of translation: There is a Lafont system in which every other Lafont system may be translated. Such a universal system, which Lafont calls the *interaction combinators*, is quite elegant: Its alphabet consists of only three symbols, two of degree 3 and one of degree 1, and the six interaction rules between them are remarkably simple.

It is natural to ask whether unambiguous multiport interaction nets also admit a universal system. Of course, one must first fix a notion of translation according to which universality is sought. Although there may not exist a notion as natural as that given by Lafont for the uniport case, we may still argue that the criteria underlying bisimilar embeddings should be considered as minimal requirements. In that case, we are able to prove a partially negative result.

Definition 6.2 (Degree of non-determinism). An *anti-clique* of an event structure E is a finite set $A \subseteq |E|$ such that $a \smile b$ for any two distinct $a, b \in A$, and there exists $u \in \mathcal{C}(E)$ enabling all events of A.

The *degree of non-determinism* of an event structure E is the smallest $m \leqslant \omega$ such that the cardinality of all anti-cliques of E is bounded by m.

A bisimilar embedding $E \overset{\iota}{\hookrightarrow} E'$ is said to *introduce divergence* if, whenever $\mathcal{B}$ is a bisimulation associated with ι, there exists $(u, \phi, u') \in \mathcal{B}$ such that there is an infinite sequence of administrative transitions $u' \xrightarrow{\tau}_\iota u'_1 \xrightarrow{\tau}_\iota u'_2 \xrightarrow{\tau}_\iota \cdots$ in E'. Embeddings not introducing divergence preserve the degree of non-determinism:

Proposition 6.3. Let $E \overset{\iota}{\hookrightarrow} E'$ without introducing divergence. Then, E' has at least the same degree of non-determinism as E.

Proof. Let $\mathcal{B}$ be a bisimulation associated with ι, and let $u \in \mathcal{C}(E)$ be a configuration enabling an anti-clique $A = \{a_1, \ldots, a_n\}$ of E. There must be a configuration u' of E' such that $(u, \phi, u') \in \mathcal{B}$ for some isomorphism ϕ and, since ι does not introduce divergence, there exists a maximal sequence of administrative transitions $u' \Longrightarrow_\iota u''$, i.e., such that there is no administrative transition starting from u''. Then, since $\mathcal{B}$ is a bisimulation, and since for all $1 \leqslant i \leqslant n$, we have $u \xrightarrow{a_i}_\iota u \cup \{a_i\}$, we must have, for all $1 \leqslant i \leqslant n$, some $a'_i \in \iota(a_i)$ such that $u'' \xrightarrow{a'_i}_\iota u'' \cup \{a'_i\}$; moreover, when $i \neq j$, $a_i \smile a_j$ implies $a'_i \smile' a'_j$ (this is Lemma 4.4), so $\{a'_1 \ldots, a'_n\}$ is an anti-clique of E'. □

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

We now show that the degree of non-determinism of an unambiguous interaction net system on a finite alphabet is strictly smaller than ω. In the following, we refer to the number of principal ports of a symbol as its *coarity*.

Lemma 6.4. Let μ be a net of an unambiguous interaction net system on a finite alphabet Σ, and let m be the maximum coarity of the symbols in Σ. Then, $\mathrm{Ev}(\mu)$ has degree of non-determinism bounded by $\lfloor \frac{3m}{2} \rfloor$ (where $\lfloor x \rfloor$ is the greatest integer not greater than x).

Proof. An anti-clique of cardinality $n > 1$ in $\mathrm{Ev}(\mu)$ is given by a reduct μ' of μ containing n active pairs $a_1, \ldots, a_n$ such that, for all i, j, a_i and a_j share at least one cell. Since the cells involved in an active pair are always distinct, we can represent the situation by saying that each a_i is a set containing exactly two cells, and that we have $a_i \cap a_j \neq \varnothing$ for all i, j. Then, the minimum coarity required by a cell $c \in C = \bigcup_{i=1}^{n} a_i$ is equal to the number of sets a_i such that $c \in a_i$; we denote this number by $\flat c$. We shall give a lower bound to $h = \max_{c \in C} \flat c$ in terms of n, which will give us the desired bound. Choose some $1 \leqslant i \leqslant n$, and let $a_i = \{c', c''\}$; for $1 \leqslant j \leqslant n$, let $x = a_i \cap a_j$. We have three mutually exclusive possibilities: $x = a_i$, $x = \{c'\}$, or $x = \{c''\}$. Let q, p', p'' be the number of a_j which fall in the first, second and third case, respectively. Note that $q > 0$ (at least $a_i \cap a_i = a_i$), and that obviously $q + p' + p'' = n$. Suppose $p'' = 0$; then all sets intersect in c', so we have $h = \flat c' = n$. The case $p' = 0$ is symmetric. On the other hand, if none of p', p'' is null, we have that necessarily all a_j with $j \neq i$ intersect in a third cell d, and we have $\flat d = p' + p''$, $\flat c' = q + p'$, $\flat c'' = q + p''$. The minimum value for h is obtained when n is equally split into three parts: $\frac{2n}{3} \leqslant h$. This lower bound is smaller than the previous $h = n$, so we may conclude that in all cases $n \leqslant \frac{3h}{2}$, as desired. □

The upper bound of Lemma 6.4 can be reached. The net

$$\alpha(w, x, y, z;) \mid \alpha(y, z, s, t;) \mid \alpha(s, t, w, x;)$$

gives an example with $m = 4$. Additionally observe that the result holds also for ambiguous systems, as long as there is a finite number of rules for each active pair.

If we combine Proposition 6.3 with Lemma 6.4, we obtain that, under any reasonable notion of translation, a universal system of multiport interaction nets either has an infinite alphabet, or divergence may be introduced when translating other systems into it.

Observe that this does not exclude the possibility of simple universal systems on a finite number of symbols plus a 'parametric' symbol α_n which gives technically infinitely many symbols (one for each $n \in \mathbb{N}$) but which is defined and interacts in a uniform, finitary way. It is for example the case of the system proposed in Mazza (2006), whose universality, however, has only been proved in a limited form. 'Parametric' symbols have already been evoked in this paper at the end of Section 3.4. The situation, if we may make analogy, is hardly unheard of: first-order Peano arithmetic has infinitely many axioms, but finitely many axiom *schemata*.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

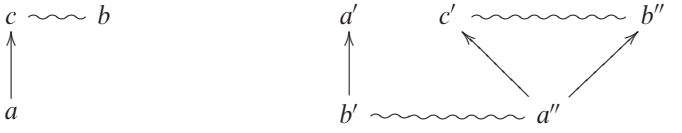
7. Concluding remarks

7.1. *On the necessity of a non-interleaving semantics*

We would like to observe that it is essential to resort to the full power of non-interleaving semantics in order to obtain the negative results of Section 4. To see this, let us define a notion of *interleaving bisimilar embedding* of event structures, which is just like Definition 4.2 but in which the bisimilarity condition only requires a 'plain' bisimulation, i.e., not a history-preserving bisimulation.

We recall that a labelled event structure E naturally induces a labelled transition system $\mathrm{lts}(E)$ whose states are the finite configurations of E, the initial state being the empty configuration, and whose transitions are given by $u \xrightarrow{\alpha} v$ iff $v = u \cup \{a\}$ for some event $a \notin u$ whose label is α. Then, in short, there is an interleaving bisimilar embedding of an (unlabelled) event structure E into E' if, when we consider the events of E to be labelled by themselves, there exists a labelling of the events of E' over $|E| \cup \{\tau\}$ such that $\mathrm{lts}(E)$ and $\mathrm{lts}(E')$ are weakly bisimilar in the usual sense.

It is not hard to see that interleaving bisimilar embeddings *do not* preserve confusion. Indeed, the minimal type II confusion, already encountered in the opening of Section 4 and recalled below on the left, is interleaving bisimilarly embeddable into the event structure below on the right, which is confusion-free:



We invite the reader to check that an interleaving bisimilar embedding is given by

$$\{(a, a'), (a, a''), (b, b'), (b, b''), (c, c')\}.$$

Using plain bisimulations instead of history-preserving bisimulations amounts to disregarding the non-interleaving features of event structures and corresponds to discarding the requirement that encodings preserve distribution, i.e., we content ourselves with encodings which respect the reduction semantics. As a matter of fact, it is obvious that the event structure above on the right simulates the one on the left by 'sequentializing' the parallelism between the events a and b. This implies the presence of a coordinating agent which alters the degree of distribution of the original process.

Therefore, we may not exclude the existence of encodings of process calculi in differential interaction nets enjoying operational correspondence thanks to the adoption of some form of 'scheduler,' whose acceptability in a concurrent setting is however debatable.

7.2. *On changes in the granularity of non-determinism*

One last remark concerning bisimilar embeddings. It may be objected that the breadth of our negative results is undermined by the fact that bisimulations (history-preserving or not) are perhaps too strong. In particular, they do not allow for a change in the 'granularity' of the non-determinism present in computational models. Therefore, bisimilar

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

embeddings do not accept, for example, that a single die cast (a six-ary internal choice) is simulated by five coin tosses (concatenated binary internal choices – of course, we are ignoring probabilities and considering only non-determinism). In this respect, the acquainted reader will immediately think of *coupled simulations* (Parrow and Sjodin 1992), which allow precisely for this kinds of granularity adjustments.

It is plausible that Definition 4.1 may be meaningfully adapted so as to define a notion of 'coupled R-simulation.' However, that would not change much: When embedding an event structure E into E', *every* event of E is considered to be relevant, i.e., E is considered to have no τ transitions. In that case, coupled bisimulations are useless: A coupled bisimulation between $\mathbb{T}$ and $\mathbb{T}'$ in which $\mathbb{T}$ has no silent transition is exactly a bisimulation.

Considering all events of E to be 'computational,' as we deemed them in opposition to 'administrative' events, is the only sensible thing to do when the information about silent transitions is missing, which is the case of unlabelled event structures. Unlabelled event structures arise naturally when one wants to describe, in terms of events, the *reduction* of a (concurrent) process, which is virtually always unlabelled, as in the π-calculus, the fusion calculus, the solos calculus, etc., not to mention all rewriting systems in general. And being able to faithfully simulate the reduction of a process is widely accepted to be an essential requirement for an encoding to be acceptable (Gorla 2010; Parrow 2008).

We do not say that changing the granularity of non-determinism is not acceptable; on the contrary, we think it would be extremely interesting if we were able to capture this phenomenon with our notion of embedding. However, for the time being, we prefer to keep this issue aside and to leave improvements to future work.

The author is grateful to Olivier Laurent and Samuel Mimram for discussions which were essential to the development of this work, and to the anonymous referees for their comments and suggestions on an earlier version of this paper. This work was partially supported by the *Fondation Sciences Mathématiques de Paris*, by Digiteo project COLLODI (2009-28HD) and by ANR projects CHOCO (07-BLAN-0324), PANDA (09-BLAN-0169-02) and LOGOI (10-BLAN-0213-02).

References

Abramsky, S. (1993). Computational interpretations of linear logic. *Theoretical Computer Science* **111** (1–2) 3–57.

Abramsky, S. (1994). Proofs as processes. *Theoretical Computer Science* **135** (1) 5–9.

Alexiev, V. (1999). *Non-deterministic Interaction Nets*. Ph.D. Thesis, University of Alberta.

Baldan, P., Corradini, A., Montanari, U. and Ribeiro, L. (2007). Unfolding semantics of graph transformation. *Information Computation* **205** (5) 733–782.

Bellin, G. and Scott, P.J. (1994). On the pi-calculus and linear logic. *Theoretical Computer Science* **135** (1) 11–65.

Boldi, P., Cardone, F. and Sabadini, N. (1993). Concurrent automata, prime event structures and universal domains. In: Droste, M. and Gurevich, Y. (eds.) *Semantics of Programming Languages and Model Theory*, Algebra, Logic and Applications, vol. 5, Gordon and Breach, 89–108.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

Caires, L. and Pfenning, F. (2010). Session types as intuitionistic linear propositions. In: Gastin, P. and Laroussinie, F. (eds.) *Proceedings of CONCUR 2010*, Lecture Notes in Computer Science, vol. 6269, Springer, 222–236.

Clark, D. and Kennaway, R. (1996). Event structures and non-orthogonal term graph rewriting. *Mathematical Structures in Computer Science* **6** (6) 545–578.

Crafa, S., Varacca, D. and Yoshida, N. (2012). Event structure semantics of parallel extrusion in the pi-calculus. In: Birkedal, L. (ed.) *Proceedings of FoSSaCS 2012*, Lecture Notes in Computer Science, vol. 7213, Springer, 225–239.

Dorman, A. (2013). *Concurrency in Interaction Nets and Graph Rewriting*. Ph.D. Thesis, Università degli Studi Roma Tre / Université Paris Nord-Sorbonne Paris Cité.

Ehrhard, T. (2005). Finiteness spaces. *Mathematical Structures in Computer Science* **15** (4) 615–646.

Ehrhard, T. and Laurent, O. (2010a). Acyclic solos and differential interaction nets. *Logical Methods in Computer Science* **6** (3).

Ehrhard, T. and Laurent, O. (2010b). Interpreting a finitary Pi-calculus in differential interaction nets. *Information and Computation* **208** (6) 606–633.

Ehrhard, T. and Regnier, L. (2006). Differential interaction nets. *Theoretical Computer Science* **364** (2) 166–195.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science* **50** (1) 1–102.

Girard, J.-Y. (1996). Proof-nets: The parallel syntax for proof-theory. In: Ursini and Agliano (eds.), Logic and Algebra. Marcel Dekker, Inc.

Gorla, D. (2010). Towards a unified approach to encodability and separation results for process calculi. *Information and Computation* **208** (9) 1031–1053.

Honda, K. and Laurent, O. (2010). An exact correspondence between a typed pi-calculus and polarised proof-nets. *Theoretical Computer Science* **411** (22–24) 2223–2238.

Khasidashvili, Z. and Glauert, J.R.W. (2005). The conflict-free reduction geometry. *Theoretical Computer Science* **347** (3) 465–497.

Kobayashi, N., Pierce, B.C. and Turner, D.N. (1999). Linearity and the Pi-Calculus. *ACM Transactions on Programming Languages and Systems* **21** (5) 914–947.

Lafont, Y. (1990). Interaction nets. In: *Conference Record of POPL'90*, ACM Press, 95–108.

Lafont, Y. (1997). Interaction combinators. *Information and Computation* **137** (1) 69–101.

Laneve, C., Parrow, J. and Victor, B. (2001). Solo diagrams. In: Kobayashi, N. and Pierce, B.C. (eds.) *Proceedings of TACS 2001*, Lecture Notes in Computer Science, vol. 2215, Springer, 127–144.

Laneve, C. and Victor, B. (2003). Solos in concert. *Mathematical Structures in Computer Science* **13** (5) 657–683.

Mazurkiewicz, A.W. (1986). Trace theory. In: Brauer, W., Reisig, W. and Rozenberg, G. (eds.) *Advances in Petri Nets*, Lecture Notes in Computer Science, vol. 255, Springer, 279–324.

Mazza, D. (2005). Multiport interaction nets and concurrency. In: Abadi, M. and de Alfaro, L. (eds.) In: *Proceedings of CONCUR 2005*, Lecture Notes in Computer Science, Springer, 21–35.

Mazza, D. (2006). Interaction nets: Semantics and concurrent extensions. Ph.D. Thesis, Université de la Méditerranée / Università degli Studi Roma Tre.

Melliès, P.-A. (2004). Asynchronous games 2: The true concurrency of innocence. *Theoretical Computer Science* **358** (2–3) 200–228.

Mimram, S. (2008). Sémantique des jeux asynchrone et réécriture 2-dimensionelle. Ph.D. Thesis, Université Paris-Diderot (Paris 7).

Nielsen, M., Plotkin, G.D. and Winskel, G. (1981). Petri nets, event structures and domains, part I. *Theoretical Computer Science* **13** (1) 85–108.

Parrow, J. (2008). Expressiveness of process algebras. *Electronic Notes in Theoretical Computer Science* **209** 173–186.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press

Parrow, J. and Sjodin, P. (1992). Multiway synchrinizaton verified with coupled simulation. In: Cleaveland, R. (ed.) *Proceedings of CONCUR'92*, Lecture Notes in Computer Science (LNCS), vol. 630, Springer, 518–533.

Parrow, J. and Victor, B. (1998). The fusion calculus: Expressiveness and symmetry in mobile processes. In: *Proceedings of LICS*, IEEE Computer Society, 176–185.

Rabinovitch, A. and Traktenbrot, B. (1988). Behaviour structures and nets. *Fundamenta Informatica* **11** (4) 357–404.

Rozenberg, G. and Engelfriet, J. (1996). Elementary net systems. In: *Dagstuhl Lectures on Petri Nets*, Lecture Notes in Computer Science, vol. 1491, Springer, 12–121.

Stark, E.W. (1989). Concurrent transition systems. *Theoretical Computer Science* **64** (3) 221–269.

van Glabeek, R.J. and Goltz, U. (1989). Equivalence notions for concurrent systems and refinement of actions. In: *Proceedings of MFCS 1989*, Lecture Notes in Computer Science (LNCS), vol. 379, Springer-Verlag.

Varacca, D., Völzer, H. and Winskel, G. (2006). Probabilistic event structures and domains. *Theoretical Computer Science* **358** (2–3) 173–199.

Winskel, G. (1982). Event structure semantics of CCS and related languages. In: Nielsen, M. and Schmidt, E.M. (eds.), *Proceedings of ICALP 1982*, Lecture Notes in Computer Science vol. 140, Springer, 561–576.

Winskel, G. and Nielsen, M. (1995). Models for concurrency. In: *Handbook of Logic in Computer Science*, vol. 4, Oxford University Press.

Wischik, L. and Gardner, P. (2005). Explicit fusions. *Theoretical Computer Science* **340** (3) 606–630.

Yoshida, N., Berger, M. and Honda, K. (2004). Strong normalisation in the pi-calculus. *Information and Computation* **191** (2) 145–202.

https://doi.org/10.1017/S0960129516000402 Published online by Cambridge University Press