Under consideration for publication in Math. Struct. in Comp. Science

Measuring the Expressiveness of Rewriting Systems through Event Structures

Damiano Mazza

Laboratoire d'Informatique de Paris Nord
UMR 7030 CNRS - Université Paris 13, F-93430 Villetaneuse, France
damiano.mazza@lipn.univ-paris13.fr
http://www-lipn.univ-paris13.fr/∼mazza

Received 15 May 2010

We develop a methodology for studying and comparing the expressiveness of computational models which are based on rewriting. We consider a class of rewriting systems, which we call *normal*, admitting a natural interpretation in terms of event structures; this is done by building up on work of Mazurkiewicz, Nielsen, Plotkin, Winskel, Melliès, and Mimram. Then, we introduce the notion of *bisimilar embedding* of event structures, which allows us to say when a computational system is "at least as expressive" as another one, as soon as these two are described in terms of event structures. Finally, we prove a few separation results for normal rewriting systems based on this notion, and we give an application of these to interaction nets and their non-deterministic variants, including Ehrhard and Regnier's differential interaction nets.

Introduction

Interaction nets (Lafont, 1990) are a computational model based on linear logic proof nets (Girard, 1996). In their original formulation, they express only deterministic computation; however, Alexiev (1999) and other authors (Khalil, 2003; Mazza, 2005; Beffara and Maurel, 2006; Ehrhard and Regnier, 2006) independently introduced several non-deterministic extensions, either on purely proof-theoretic grounds, or with the explicit aim of broadening the computational paradigm captured by interaction nets, in particular towards interaction-net-based models of mobile and concurrent systems.

The question that motivated this work is: what is the expressive power of interaction nets? An obvious answer is "they are Turing-complete"; but it is even more obvious that we do not consider such answer to be meaningful at all, otherwise there would be no point in studying extensions of Lafont's original model (which is already Turing-complete (Lafont, 1997)). If we are interested in such extensions, it is because we look not only at *what* we compute, but also at *how* we compute it.

We are therefore seeking some kind of measure of the *intensional* expressiveness of a computational model, instead of simply considering the set of functions it is able to compute. We point out that there is no generally accepted way of doing this; our assumption will be 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.

This view of computation has long been considered by Winskel in his *event structures* (Winskel, 1980), which formalize precisely the concepts of causality and conflict discussed above. The actual meaning of "event" depends on the computational model which is being interpreted as an event structure: for instance, in Turing machines or Petri nets, it is usually convenient to identify events with the executions of transitions. In the case of Turing machines, causality and conflict are trivial: each transition depends on the occurrence of the previous transition, and is necessary for the execution of the next (*i.e.*, we have a linear order), and there is no conflict. By contrast, causality and conflict between transitions can be much more intricate in the case of Petri nets.

The idea now is that the intensional expressiveness of a computational model is given by the complexity of the causality and conflict structure of its events. Defining the "complexity" of such structure in an absolute way is difficult, and probably meaningless; instead, we may take a relative, comparative approach. One of the most widely used techniques to compare computational systems in presence of non-determinism and concurrency is that of relating them through bisimulations. In the context of event structures, a notion of bisimulation, called *history-preserving bisimulation*, was independently introduced by Rabinovitch and Traktenbrot (1988) and van Glabeek and Goltz (1989); later, Joyal et al. (1996) introduced a stronger notion, which has the advantage of fitting into a broad categorical framework.

In this paper, we consider the weaker notion, because we are mostly interested in separation results, whose meaningfulness is inversely proportional to the strength of the equivalences considered. Based on this, we introduce a notion of *bisimilar embedding* of event structures (*cf.* Sect. 3). Intuitively, an event structure E embeds in an event structure E' if this latter contains one or more copies of E, up to bisimilarity. This gives us an arguably natural way of saying that a model is "at least as expressive" as another, as soon as these are described as event structures.

Our next step is finding out how the computational process represented by an interaction net can be given a description in terms of event structures. For this, we abstract from interaction nets, and develop an analysis of events in the more general context of rewriting systems. Such analysis broadens considerably the scope of our work; indeed, virtually any model in which there is some notion of "state" and "transition" may be described as an abstract rewriting system: the "states" are the objects of the system, which are rewritten through the "transitions". In this formulation, causality and conflict are intuitively given by the notion of *residue structure* of the rewriting system. Thus, addressing our original question becomes a simple application of more general results.

Of course, we still need to make some restrictions on rewriting systems to be able to meaningfully describe them in terms of event structures. This is the reason behind the introduction of what we call *normal rewriting systems* (*cf.* Sect. 2): these admit a well behaved definition of homotopy of reductions, from which the events may be easily retrieved (as homotopy classes). Our theory of normal rewriting systems may be seen as an alternative presentation of Mimram's (2008) work on asynchronous graphs, and has its roots in the theory of Mazurkiewicz traces (Winskel and Nielsen, 1995), as well as in work by Nielsen et al. (1981) and Melliès (2004).

A further contribution of this paper is the identification of a subclass of normal rewriting systems, which we call *confusion-free*, exactly corresponding to the confusion-free event structures of Varacca et al. (2006). The prototypical example of confusion-free normal rewriting system is given by non-deterministic Turing machines; another example, which is of great importance in the current development of linear logic proof theory, is given by the differential interaction nets of Ehrhard and Regnier (2006). It is possible to prove that, according to our notion of bisimilar embedding, confusion-free systems are strictly less expressive than unrestricted normal rewriting systems (Corollary 3.10). Similarly, we give a negative result about the embedding of a normal rewriting systems into another system with a smaller degree of non-determinism (Corollary 3.12).

When applied to interaction nets, these give us the first strong, convincing separation results concerning classes of non-deterministic interaction nets. Moreover, we are able to give a nice characterization of the event structures arising from deterministic interaction nets (those originally introduced by Lafont) and multirule interaction nets (of which differential interaction nets are a special case).
[Note: for briefness, most proofs are omitted; the important ones are given in the appendices.]

Acknowledgments. The author is grateful to Samuel Mimram, who pointed out that normal rewriting systems could be defined using the cube property. This work was partially supported by Digiteo project COLLODI (2009-28HD) and ANR project PANDA.

1. Event Structures and Configuration Posets

1.1. *The groupoid of event structures*

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

Definition 1.1 (Event structure (Winskel and Nielsen, 1995)). An *event structure* is a triple $E = (|E|, \leq, \smile)$ where:

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

The complement of $\smile$, called *coherence relation*, is denoted by $\frown$.

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 $\lceil a \rceil$.

There is a standard notion of morphism of event structures (Winskel and Nielsen, 1995), given by partial functions between events which preserve configurations and are injective on coherent events. However, in this work we are only interested in event structures as "static" objects, and not in their transformations; this amounts to saying that we only consider the subcategory of event structures and their isomorphisms.

Definition 1.2 (Isomorphism of event stuctures). Let $E = (|E|, \leq, \smile)$ and $E' = (|E'|, \leq', \smile')$ be event structures. An *isomorphism* between E and E' is a bijection $\beta : |E| \to |E'|$ such that, for all $a, b \in |E|$, $a \leq b$ iff $\beta(a) \leq' \beta(b)$ and $a \smile b$ iff $\beta(a) \smile' \beta(b)$.

Event structures and their isomorphisms obviously form a groupoid, which we denote by $\mathbf{E_{Grp}}$.

1.2. *The groupoid of configuration posets*

Given an event structure E, it is possible to recover its causality and conflict relations from the poset (ordered by inclusion) of its of finite configurations $\mathcal{C}(E)$. This is precisely the contents of Theorem 1.6 below, which is a reformulation of well known results by Nielsen, Winskel and Plotkin (Nielsen et al., 1981; Winskel, 1982). In order to proceed, we need to introduce the notion of *configuration poset*, along with a few preliminary order-theoretic notions.

Let $(X, \leq)$ be a poset. We say that $u \subseteq X$ is *bounded* if there exists $z \in X$ such that, for all $x \in u$, $x \leq z$; in the special case where $u = \{x, y\}$, we say that x and y are *compatible*, and we write $x \uparrow y$. We denote by $x \vee y$ and the supremum of two elements x, y of X, whenever this exists; similarly, if $u \subseteq X$, we denote by $\bigvee u$ the supremum of all elements of u, if it exists.

A poset $(X, \leq)$ is *bounded complete* if for all $x, y \in X$, $x \uparrow y$ implies that $x \vee y$ exists; in other words, in a bounded complete poset it is equivalent, for any finite $u \subseteq X$, to say that u is bounded or that $\bigvee u$ exists.

A subset of a poset is a *clique* if its elements are pairwise compatible. We say that a poset is *coherent* if every finite clique has a supremum. Observe that a coherent poset is necessarily bounded complete. The converse does not hold; however, we have the following:

Proposition 1.3. A poset $(X, \leq)$ is coherent iff it has a least element, and is *locally coherent*, *i.e.*, for every $x, y, z \in X$, $x \uparrow y$, $x \uparrow z$, and $y \uparrow z$ implies that $\bigvee\{x, y, z\}$ exists.

Let $(X, \leq)$ be a poset, and let $x, y \in X$. We say that y *covers* x if $x < y$, and there is no z such that $x < z < y$; we say that $a \in X$ is *prime* if it covers exactly one element of X. We denote by $\mathfrak{P}(X)$ the set of prime elements of X. The terminology is justified by the fact that a prime element is "indecomposable" with respect to suprema: it is not hard to check that, if $a \in \mathfrak{P}(X)$ and $u \subseteq X$ is such that $\bigvee u = a$, then $a \in u$ (*cf.* Lemma A.1). As an example, we characterize the prime elements of the poset $(\mathcal{C}(E), \subseteq)$, *i.e.*, the set of finite configurations of an event structure E, ordered by inclusion.

Lemma 1.4. For every event structure $E = (|E|, \leq, \smile)$, we have $\mathfrak{P}(\mathcal{C}(E), \subseteq) = \{\downarrow a \mid a \in |E|\}$, *i.e.*, the principal ideals of $(|E|, \leq)$.

Let $(X, \leq)$ be a poset, and let $x \in X$. We define $\mathfrak{p}(x) = \{a \in \mathfrak{P}(X) \mid a \leq x\}$. We say that X is *prime algebraic* if for all $x \in X$, $\bigvee \mathfrak{p}(x)$ exists and is equal to x.

Definition 1.5 (Configuration poset). A *configuration poset* is a coherent, prime algebraic poset $(X, \leq)$ such that, for all $x \in X$, $\downarrow x$ is finite.

Once again, we shall not be interested in morphisms of configuration posets, but only in their isomorphisms, which are usual poset isomorphisms, *i.e.*, bijections preserving the order in both directions. We denote by $\mathbf{Conf_{Grp}}$ the groupoid of configuration posets and their isomorphisms.

1.3. *The representation theorem for event structures*

As anticipated above, we have the following result:

Theorem 1.6 (Representation). The groupoids $\mathbf{E_{Grp}}$ and $\mathbf{Conf_{Grp}}$ are equivalent.

We recall that an equivalence between $\mathbf{E_{Grp}}$ and $\mathbf{Conf_{Grp}}$ is a quadruple $(\Phi, \Psi, \eta, \varepsilon)$ such that $\Phi : \mathbf{E_{Grp}} \to \mathbf{Conf_{Grp}}$, $\Psi : \mathbf{Conf_{Grp}} \to \mathbf{E_{Grp}}$ are functors, and $\eta : \mathrm{Id}_{\mathbf{E_{Grp}}} \Rightarrow \Psi \circ \Phi$, $\varepsilon : \Phi \circ \Psi \Rightarrow \mathrm{Id}_{\mathbf{Conf_{Grp}}}$ are natural isomorphisms.

The functors Φ and Ψ are given by the following results:

Lemma 1.7. Let E, E' be event structures, and let $\beta : E \to E'$ be an isomorphism. Then:

1. $\Phi(E) = (\mathcal{C}(E), \subseteq)$ is a configuration poset (as is $\Phi(E') = (\mathcal{C}(E'), \subseteq)$);
2. the function $\Phi(\beta) : \mathcal{C}(E) \to \mathcal{P}_{\mathrm{fin}}(E')$ defined by $\Phi(\beta)(x) = \{\beta(a) \mid a \in x\}$, for all $x \in \mathcal{C}(E)$, is an isomorphism between $\Phi(E)$ and $\Phi(E')$.

Lemma 1.8. Let $(X, \leq), (X', \leq')$ be configuration posets, and let $\phi : X \to X'$ be an isomorphism. Define $\Psi(X) = (\mathfrak{P}(X), \leq, \not\uparrow)$, where $\not\uparrow$ is the negation of compatibility, *i.e.*, $a \not\uparrow b$ if there is no $z \in X$ such that $a, b \leq z$. Then:

1. $\Psi(X)$ is an event structure (as is $\Psi(X')$, which can be defined similarly);
2. the restriction $\Psi(\phi)$ of ϕ to $\mathfrak{P}(X)$ is an isomorphism between $\Psi(X)$ and $\Psi(X')$.

The natural isomorphisms are given by the following results:

Lemma 1.9. Let E be an event structure. Define $\eta_E : |E| \to \mathcal{C}(E)$ by $\eta_E(a) = \downarrow a$, for all $a \in |E|$. Then, η_E is an isomorphism between E and $\Psi(\Phi(E))$.

Lemma 1.10. Let X be a configuration poset, let u be a finite clique of prime elements of X, and set $\varepsilon_X(u) = \bigvee u$. This defines an isomorphism ε_X between $\Phi(\Psi(X))$ and X.

Note that the original results of Nielsen et al. (1981) and Winskel (1982) prove an equivalence between the category $\mathbf{E}$ of event structures and their morphisms (not just isomorphisms) and a certain subcategory of the category of dI-domains (Berry, 1979) and linear functions. As stated above, the reason why we restrict to $\mathbf{E_{Grp}}$ is that we are not interested in general morphisms of event structures. The motivation for introducing the notion of configuration poset instead of using dI-domains will be clear in the sequel. At this time, we can justify this by observing that, since every dI-domain D is algebraic, it is entirely determined by its set compact elements $\mathcal{K}(D)$. This is not itself a domain of any kind (in particular, it is generally not a cpo); however, when D is coherent, as is the case with dI-domains arising from event structures, $\mathcal{K}(D)$ is a configuration poset.

2. Normal Rewriting Systems

2.1. *Pre-normal rewriting systems and homotopy*

In the following, we use the standard categorical definition of graph, *i.e.*, a graph is a tuple $\mathcal{G} = (\mathcal{G}_0, \mathcal{G}_1, \mathrm{src}, \mathrm{trg})$ where:

— $\mathcal{G}_0$ is a set of *nodes*, ranged over by μ, ν, o;
— $\mathcal{G}_1$ is a set of *arrows*, ranged over by r, s, t;
— $\mathrm{src} : \mathcal{G}_1 \to \mathcal{G}_0$ and $\mathrm{trg} : \mathcal{G}_1 \to \mathcal{G}_0$ are the *source* and *target* function, respectively.

We denote by $\mathcal{G}^*$ the free category over $\mathcal{G}$, which we recall has as objects the nodes of $\mathcal{G}$ and as morphisms the directed paths of $\mathcal{G}$, *i.e.*, finite sequences $r_1 \cdots r_n$ of arrows of $\mathcal{G}$ such that, for all $1 \leq i \leq n-1$, $\mathrm{trg}(r_i) = \mathrm{src}(r_{i+1})$, plus, for every node μ, an identity morphism id_μ, with composition defined as path concatenation. The source and target functions are extended to paths in the obvious way; two paths are said to be *coinitial* (resp. *cofinal*) if they have the same source (resp. target). We shall simply write id for an identity morphism whose source (and target) is irrelevant. Paths will be ranged over by $f, g, h, \ldots$; the length of a path f is denoted by $\|f\|$, with $\|f\| = 0$ iff $f = id$. We shall identify arrows of $\mathcal{G}_1$ with paths of length 1, and we shall denote composition by juxtaposition and in the reverse order with respect to usual categorical composition, *i.e.*, rs will denote the path composed first of the arrow r, then the arrow s.

Definition 2.1 (Rewriting system, radical, reduction, residue). Let $\mathcal{G}$ be a graph. A *residue structure* on $\mathcal{G}$ is a relation $R \subseteq \mathcal{G}_1^3$ such that, whenever $(r, s, t) \in R$, we have $\mathrm{src}\, r = \mathrm{src}\, s$ and $\mathrm{src}\, t = \mathrm{trg}\, r$. If $(r, s, t) \in R$, t is said to be a *residue* of s through r (according to R).

A *rewriting system* is a pair $\mathcal{S} = (\mathcal{G}, R)$, where $\mathcal{G}$ is a graph and R a residue structure on $\mathcal{G}$.

In the context of rewriting systems, the nodes of $\mathcal{G}$ are called *objects*, the arrows of $\mathcal{G}$ *radicals*, and the morphisms of $\mathcal{G}^*$ *reductions*. Given a reduction f and a coinitial radical s, we define the set of residues of s through f, denoted by $[f]s$, by induction on $\|f\|$, as follows: $[id]s = \{s\}$, and $[rg]s = \bigcup_{s' \in [g]s} \{t \in \mathcal{G}_1 \mid (r, s', t) \in R\}$. Given two reductions f, g, we say that they are *equivalent*, and we write $f \rightleftharpoons g$, whenever they are coinitial, cofinal, and, for all $t \in \mathcal{G}_1$ coinitial with f and g, we have $[f]t = [g]t$.

Definition 2.2 (Pre-normal rewriting system). A rewriting system $(\mathcal{G}, R)$ is called *pre-normal* whenever, for any coinitial radicals r, s, the following conditions hold:

affinity: $\sharp[r]s \leq 1$; in case this set is a singleton, we denote its only element by s^r;

symmetry: $\sharp[r]s = \sharp[s]r$; in case these sets are singletons, we say that r and s are *independent*;

tiling: $rs^r \rightleftharpoons sr^s$.

Definition 2.3 (Homotopy). Let $\mathcal{S}$ be a pre-normal rewriting system. We define a sequence $(\sim_n)_{n<\omega}$ of binary relations on the reductions of $\mathcal{S}$ by induction on n, as follows:

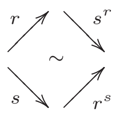
— $f \sim_0 g$ iff $f = g$;

— $f \sim_{n+1} g$ iff there exist two reductions h, h' and two independent radicals r, s such that $f \sim_n hrs^r h'$ and $g = hsr^s h'$.

Homotopy, denoted by $\sim$, is the union of all $\sim_n$ for $n < \omega$. This is an equivalence relation; we denote by $\mathcal{H}(\mathcal{S})$ the set of homotopy classes of $\mathcal{S}$, and we range it over by x, y, z. If f is a reduction of $\mathcal{S}$, we denote by $[f]$ its homotopy class.

Note that, given $x \in \mathcal{H}(\mathcal{S})$, any $f, g \in x$ are coinitial (and also cofinal, equivalent, and of equal length); hence, we can obviously extend the source (and target) function to homotopy classes. Then, if μ is an object of $\mathcal{S}$, we denote by $\mathcal{H}_\mu(\mathcal{S})$ the largest subset of $\mathcal{H}(\mathcal{S})$ such that $x \in \mathcal{H}_\mu(\mathcal{S})$ implies $\mathrm{src}(x) = \mu$.

Observe that $f \sim g$ implies $\|f\| = \|g\|$ and $f \rightleftharpoons g$, a fact that we shall use in the sequel. It is often convenient to depict homotopy relations as "tiles" between reductions:



In fact, pre-normal rewriting systems may be equivalently defined in terms of a class of 2-graphs called *asynchronous graphs*, as done by Melliès (2004); Mimram (2008). The 2-cells of these 2-graphs, which are all invertible, are generated by composing tiles, and they represent the homotopy relation.

Definition 2.4 (Homotopy order). Let $\mathcal{S}$ be a pre-normal rewriting system, and let f, g be reductions of $\mathcal{S}$. We write $f \lesssim g$ iff there exists h such that $fh \sim g$. This is easily seen to be a preorder, whose associated equivalence is exactly homotopy, *i.e.*, $f \lesssim g$ and $g \lesssim f$ iff $f \sim g$. Hence, $\mathcal{H}(\mathcal{S})$ becomes a poset by setting $[f] \leq [g]$ whenever $f \lesssim g$.

2.2. *Normal rewriting systems and event structures*

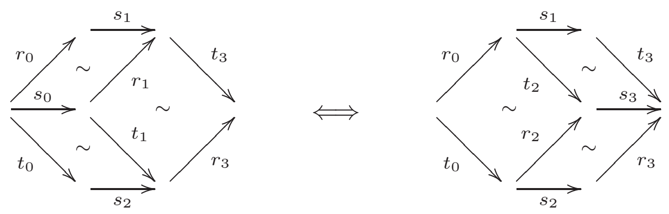
In what follows, when we say that a rewriting system "contains" a certain graphical structure (presented through "tiles"), we mean that its underlying graph contains the given arrows and paths and, according to its residue structure, these are in the depicted homotopy relation.

Definition 2.5 (Normal rewriting system). A pre-normal rewriting system $\mathcal{S}$ is *normal* if the following conditions hold:

self-conflict: for every radical r, $[r]r = \emptyset$;

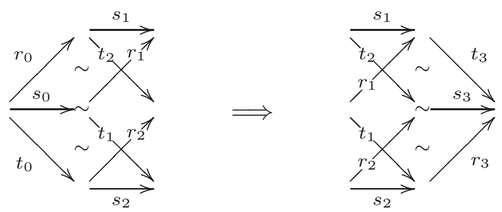
injectivity: for all radicals r, s, t with r, t and s, t independent, $r^t = s^t$ implies $r = s$; graphically, a path of length 2 can be the border of at most one tile;

cube property: for all radicals $r_0, r_3, s_1, s_2, t_0, t_3$, there exist r_1, s_0, t_1 such that $\mathcal{S}$ contains the structure below on the left iff there exist r_2, s_3, t_2 such that $\mathcal{S}$ contains the structure on the right:



Note that the cube property is actually the sum of two properties: the implication from left to right, which we call *forward cube property*, and the implication from right to left, which we call *backward cube property* (the terminology is borrowed from Mimram (2008)).

cubic pushout: for all radicals $r_0, r_1, r_2, s_0, s_1, s_2, t_0, t_1, t_2$, if $\mathcal{S}$ contains the structure below on the left, then there exist radicals r_3, s_3, t_3 such that $\mathcal{S}$ contains the structure on the right:



Normal rewriting systems have the following fundamental property:

Theorem 2.6. Let $\mathcal{S}$ be a normal rewriting system. Then, for every object μ of $\mathcal{S}$, $(\mathcal{H}_\mu(\mathcal{S}), \leq)$ is a configuration poset, where $\leq$ is the partial order of Definition 2.4 restricted to $\mathcal{H}_\mu(\mathcal{S})$.

This result explains the need to consider configuration posets instead of dI-domains: finite reductions are most natural in the context of rewriting systems; however, the poset of all finite reductions modulo homotopy is unlikely to be directed complete.

The reminder of this section is devoted to showing the key results needed to prove Theorem 2.6. The missing proofs are given in Appendix B. In what follows, all objects, radicals, and reductions considered belong to a generic, fixed normal rewriting system $\mathcal{S}$.

We start by extending the notion of residue to *compatible* reductions, *i.e.*, coinitial reductions f, g such that there exists h such that $f, g \lesssim h$. In that case, we shall define the residue of g after f, which will be a reduction coinitial with the target of f, denoted by g^f. If g is a radical r, we already have a partial definition: r^f is defined and is equal to a radical t iff $[f]r = \{t\}$. The new definition will extend this one: it will always be defined, and will coincide with the old one when this latter is defined.†

The first step is proving the following fundamental result, which uses the cube property:

Lemma 2.7. Let r, s be two distinct radicals, and let f, g be reductions. Then:

1 $rf \sim sg$ implies that r and s are independent, and there exists a reduction h such that $f \sim s^r h$ and $g \sim r^s h$;

2 $frt_1 \sim gst_2$ implies that $t_1 = p^q$ and $t_2 = q^p$ for some independent radicals p, q, and there exists a reduction h such that $fr \sim hq$ and $gs \sim hp$.

We proceed to define g^r, where r is a radical such that $\mathrm{src}(r) = \mu$, $\mathrm{trg}(r) = \nu$. Observe that, if $g = sh$ with $s \neq r$, the fact that r and g are compatible implies, by point (1) of Lemma 2.7, that r and s are independent, so s^r and r^s are defined. Then, we give the definition by induction on $\|g\|$:

$$g^r = \begin{cases} id_\nu & \text{if } g = id_\mu \\ h & \text{if } g = rh \\ s^r h^{r^s} & \text{if } g = sh, s \neq r \end{cases}$$

In the third case, h^{r^s} is defined because $\|h\| < \|g\|$.

Finally, we extend the definition to all compatible reductions f, g, by induction on $\|f\|$:

$$g^f = \begin{cases} g & \text{if } f = id \\ (g^r)^h & \text{if } f = rh \end{cases}$$

The next result collects some useful properties of residues:

Lemma 2.8. Let f, g be compatible reductions. Then:

1 if $\mathrm{src}\, f = \mu$ and $\mathrm{trg}\, f = \nu$, then $id_\mu^f = id_\nu$;

2 if $f = hk$, $f^g = (hk)^g = h^g k^{g^h}$;

3 $fg^f \sim gf^g$.

The residue operation is a sort of left adjoint to path concatenation:

Lemma 2.9 (Adjunction). For all reductions f, g, h, we have

1 $f \lesssim gh$ iff $f^g \lesssim h$;

2 $f \sim gh$ implies $f^g \sim h$.

Note that the converse of point (2) of Lemma 2.9 fails, as it easily seen by taking two independent radicals r, s and setting $f = r$, $g = s$, and $h = r^s$.

We are now ready to prove Theorem 2.6, which will be a consequence of Corollary 2.12 (coherence), Proposition 2.16 (the poset is prime algebraic), and Proposition 2.18 (the principal ideals are finite).

† In particular, we shall see that the extended definition will give $r^r = id_\nu$ for every radical r, where $\nu = \mathrm{trg}(r)$; the only purpose of the self-conflict axiom is precisely to ensure that such equation does not contradict the residue structure of $\mathcal{S}$, which may otherwise attribute any radical of source ν as the residue of r through itself.

Proposition 2.10 (Bounded completeness). For all reductions f, g, $[f] \uparrow [g]$ implies $[f] \vee [g] = [fg^f] = [gf^g]$.

Proof. Using point (3) of Lemma 2.8, we have $[f], [g] \leq [fg^f] = [gf^g]$. Let now h be such that $[f], [g] \leq [h]$. By definition, there exists f' such that $h \sim ff'$; then, we have $g \lesssim ff'$, so Lemma 2.9 gives us $g^f \lesssim f'$, so $fg^f \lesssim ff' \sim h$, which implies $[fg^f] \leq [h]$. □

Proposition 2.11 (Local coherence). For all reductions f, g, h, $[f] \uparrow [g], [f] \uparrow [h], [g] \uparrow [h]$ implies that $\bigvee\{[f], [g], [h]\}$ exists.

Proof. The proof is by induction on $\|f\| + \|g\| + \|h\|$. Instead of giving the (tedious) details, we shall simply give the basic intuition, which should be enough to convince the reader of the correctness of the statement. Observe that by hypothesis, and by Proposition 2.10, our rewriting system $\mathcal{S}$ contains a structure which is similar to that on the left of the statement of the cubic pushout property (Definition 2.5), but composed of reductions instead of radicals: r_0, r_1, r_2 are replaced by f, f^g, f^h, respectively; s_0, s_1, s_2 are replaced by g, g^f, g^h, respectively; and t_0, t_1, t_2 are replaced by h, h^g, h^f, respectively. Now, the induction does nothing but iterating the cubic pushout property, "closing" this cube and giving the supremum of $[f], [g], [h]$, which is, for instance, $[fg^f h^{fg^f}]$. □

Corollary 2.12 (Coherence). For every object μ, the sub-poset $(\mathcal{H}_\mu(\mathcal{S}), \leq)$ is coherent.

Proof. We simply apply Proposition 1.3, using Proposition 2.11 and the fact that $\{id_\mu\}$ is obviously the least element of $(\mathcal{H}_\mu(\mathcal{S}), \leq)$. □

Note that $(\mathcal{H}(\mathcal{S}), \leq)$ fails to be coherent for a rather trivial reason: the only clique not having a supremum is the empty set, because $\mathcal{H}(\mathcal{S})$ lacks a least element!

Definition 2.13 (Essential homotopy class). Let $\mathcal{S}$ be a pre-normal rewriting system, and let f be a reduction of $\mathcal{S}$. We say that f is *essential* if $f = f'r$ for some radical r and reduction f' and, whenever $f' \sim f''s$ for some radical s and reduction f'', there is no radical t such that $r = t^s$. We then say that $x \in \mathcal{H}(\mathcal{S})$ is essential if $x = [f]$ for some essential reduction f.

The following motivates the introduction of essential homotopy classes:

Lemma 2.14 (Prime elements). $x \in \mathfrak{P}(\mathcal{H}(\mathcal{S}))$ iff x is essential.

The next result crucially uses the backward cube property:

Lemma 2.15. Let fr be a reduction, with r a radical. Then, there exists a unique essential homotopy class $[ht]$, with t a radical, such that $f \sim hg$ and $r = t^g$ for some reduction g.

Proposition 2.16 (A prime algebraic poset). The poset $(\mathcal{H}_\mu(\mathcal{S}), \leq)$ is prime algebraic.

Proof. Let $x \in \mathcal{H}_\mu(\mathcal{S})$. We set $x = [r_1 \cdots r_n]$ with $n \in \mathbb{N}$ and $r_1, \ldots, r_n$ radicals ($x = [id_\mu]$ in case $n = 0$). If we set, for each $0 \leq k \leq n$, $f_k = r_1 \cdots r_k$ (again, $f_0 = id_\mu$ if $n = 0$), applying Lemma 2.15 to each f_i with $1 \leq i \leq n$ yields a sequence of essential homotopy classes $[e_1],\dots,[e_n]$ such that, for all $1 \le i \le n$, $e_i = h_i t_i$, $f_{i-1} \sim h_i g_i$, and $r_i = t_i^{g_i}$, where h_i, g_i are reductions and t_i radicals; moreover, by uniqueness, these are actually *all* the essential homotopy classes below x, so, by Lemma 2.14, $\mathfrak{p}(x) = \{[e_1],\dots,[e_n]\}$. Since these classes are all pairwise compatible (they are bounded by x), we may compute their supremum as in Proposition 2.10, *i.e.*, $\bigvee_{i=1}^{n}[e_i] = [d_n]$, where d_n is defined inductively, as follows: $d_0 = id_\mu$, and $d_{k+1} = d_k e_{k+1}^{d_k}$, where $0 \le k < n$. All that is left to do is prove that $[d_n] = x$, which is equivalent to $d_n \sim f_n$. We reason by induction: given $0 \le k < n$, we prove $d_{k+1} \sim f_{k+1}$ supposing that $d_j \sim f_j$ for all $j \le k$. We start by considering $e_{k+1}^{d_k}$; by hypothesis, this is equal to $(h_{k+1}t_{k+1})^{d_k} = h_{k+1}^{d_k} t_{k+1}^{d_k^{h_{k+1}}}$, where we applied point (2) of Lemma 2.8. Now, the induction hypothesis gives us $d_k \sim f_k$, and by hypothesis $f_k \sim h_{k+1}g_{k+1}$, so: on the one hand, $h_{k+1} \lesssim d_k$, which implies $h_{k+1}^{d_k} = id$ by point (1) of Lemma 2.9; on the other hand, by point (2) of Lemma 2.9, we have $d_k^{h_{k+1}} \sim g_{k+1}$, so $d_k^{h_{k+1}} \rightleftharpoons g_{k+1}$ by the tiling axiom, which implies $t_{k+1}^{d_k^{h_{k+1}}} = t_{k+1}^{g_{k+1}} = r_{k+1}$ by hypothesis. Summing up, in the end we obtain $e_{k+1}^{d_k} = r_{k+1}$; then, by applying once more the induction hypothesis, we have $d_{k+1} = d_k e_{k+1}^{d_k} \sim f_k e_{k+1}^{d_k} = f_k r_{k+1} = f_{k+1}$, as claimed. ☐

Again, $(\mathcal{H}(\mathcal{S}), \le)$ fails to be prime algebraic for the simple reason that identity homotopy classes, *i.e.*, classes of the form $[id]$, are such that $\mathfrak{p}([id]) = \emptyset$, which in general does not have a supremum in $\mathcal{H}(\mathcal{S})$ (there is no least element).

We now complete the proof by establishing the finiteness of principal ideals, which instead holds for all of $\mathcal{H}(\mathcal{S})$. The following lemma uses the injectivity axiom:

Lemma 2.17. For every reduction f, $\sharp[f] \le \|f\|!$.

Proposition 2.18 (Principal ideals are finite). For all $x \in \mathcal{H}(\mathcal{S})$, $\downarrow x$ is finite.

Proof. It is enough to prove that, for every reduction f, there are only finitely many g such that $g \lesssim f$. This, by definition, implies that there exists h such that $gh \sim f$, *i.e.*, g is a prefix of a reduction in $[f]$; but these are finitely many by Lemma 2.17. ☐

Theorem 2.6 allows us, modulo the Representation Theorem 1.6, to associate an event structure with any object of a normal rewriting system. More precisely, if $\mathcal{S}$ is a normal rewriting system and μ is an object of $\mathcal{S}$, we may associate with μ the event structure $\mathrm{Ev}(\mu) = \Psi(\mathcal{H}_\mu(\mathcal{S}), \le)$, where Ψ is the functor of Theorem 1.6. Morally, the events of $\mathrm{Ev}(\mu)$ are the radicals which progressively appear when one reduces μ; two events are in conflict when the corresponding radicals are not independent. Lemma 2.19 below formalizes this intuition.

Observe that, still thanks to Theorem 1.6, $\mathcal{C}(\mathrm{Ev}(\mu))$ is isomorphic to $\mathcal{H}_\mu(\mathcal{S})$; this means in particular that each finite configuration u of $\mathrm{Ev}(\mu)$ corresponds to a unique homotopy class $\varepsilon(u)$ of source μ.

Lemma 2.19. Let μ be an object of a normal rewriting system, let $[ht] \in |\mathrm{Ev}(\mu)|$, and let $u \in \mathcal{C}(\mathrm{Ev}(\mu))$, with $\varepsilon(u) = [f]$. Then, u enables $[ht]$ iff there exists a radical r such that $[ht]$ is the unique homotopy class given by Lemma 2.15 applied to fr.

As a concluding remark, let us point out that, if $\mathcal{S}$ is a normal rewriting system, the failure of $(\mathcal{H}(\mathcal{S}), \leq)$ to be a configuration poset is so trivial that it can be easily fixed: simply consider all identity homotopy classes to be equivalent, so that the poset now has a least element. We can thus associate an event structure $\mathrm{Ev}(\mathcal{S})$ to the whole system; this may be seen to be isomorphic to $\bigoplus_{\mu\in\mathcal{S}} \mathrm{Ev}(\mu)$, where $\oplus$ denotes the coproduct of event structures, which is basically a disjoint union (Winskel and Nielsen, 1995).

3. Measuring the Expressiveness of Normal Rewriting Systems

3.1. *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 $x \in \pi_1(R)$ (we call this a *computational transition* labelled by a), and $u \longrightarrow_R v$ otherwise (we call this an *administrative transition*). We denote by $\Longrightarrow_R$ the reflexive-transitive closure of $\longrightarrow_R$, 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 $u' \xrightarrow{x'}_R v'$ if $x' \in \pi_2(R)$, and $u' \longrightarrow_R v'$ otherwise, with $u', v' \in \mathcal{C}(E')$ and $x' \in |E'|$. 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 $\mathcal{P}_{\mathrm{fin}}(X)$ the set of finite subsets of a set X.

Definition 3.1 (R-bisimulation). Let $E = (|E|, \leq, \smile), E' = (|E'|, \leq', \smile')$ be event structures, and let $R \subseteq |E| \times |E'|$. A *$R$-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 $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}$ and, whenever $(u, \phi, u') \in \mathcal{B}$, we have:

i. ϕ is a poset isomorphism between $(\mathrm{supp}_R(u), \leq)$ and $(\mathrm{supp}_R(u'), \leq')$;
ii. $u \xrightarrow{a}_R v$ implies $u' \overset{a'}{\Longrightarrow}_R v'$ with $(v, \phi \cup \{(a, a')\}, v') \in \mathcal{B}$;
iii. $u \longrightarrow_R v$ 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. $u' \longrightarrow_R v'$ 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.

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', $\{(u, \emptyset, u') \mid u \in |E|, u' \in |E'|\}$ is a $\emptyset$-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 3.2 (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 $a, b \in |E|$, $\iota(a) \cap \iota(b) \neq \emptyset$ 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 3.3 (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.

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).

The notion of bisimilar embedding must be seen as a sort of "faithful simulation": if we regard E, E' as descriptions of computational processes, $E \hookrightarrow E'$ means that the process described by E' may "simulate" the process described by E covering every computational dynamics of E and not adding new dynamics. It is important to observe that embeddings are *not* symmetric, *i.e.*, E may be embedded into E' without E' being embeddable into E. This is simply because ι is required to be injective, and obviously $|E|$ may be injected into $|E'|$ without the converse being possible.

In our opinion, this asymmetry is a desirable feature: for instance, it is reasonable to say that a Lafont interaction net (see Sect. 4.4) faithfully simulates a Turing machine, but the contrary seems unreasonable, if we want to take parallelism into account—the event structure of a Turing machine is a linear order, while that of a Lafont interaction net may be much richer (see Proposition 4.14).

3.2. *Confusion-free rewriting systems*

It is possible to define a class of normal rewriting systems whose objects generate event structures of an interesting form, introduced by Varacca et al. (2006) and called *confusion-free*. We recall the definition below.

Definition 3.4 (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 3.5 (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 $a_0 \leq a$, $b_0 \leq b$.

Below, we denote by $\overset{\#}{\sim}$ the reflexive closure of immediate conflict.

Definition 3.6 (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,

— $\overset{\#}{\sim}$ is transitive, *i.e.*, it is an equivalence relation;
— for every $a, b \in |E|$, $a \# b$ implies $\lceil a \rceil = \lceil b \rceil$.

The equivalence classes of $\overset{\#}{\sim}$ are called *switches*.‡

Definition 3.7 (Confusion-free rewriting system). Let r, s be two coinitial radicals of a normal rewriting system. We say that r and s are *separated* if every radical t coinitial with r, s is independent with at least one of r, s. We say that r and s are *contemporary* if, for all radical r_0 and reduction h such that $r = r_0^h$, there exists a radical s_0 such that $s = s_0^h$. We say that r and s are in *simple conflict* if they are contemporary and not independent. A normal rewriting system $\mathcal{S}$ is *confusion-free* if all coinitial radicals are either separated or in simple conflict.

Observe that two separated radicals must be independent; in fact, if r and s are coinitial and not independent, then by the self-conflict axiom r is not independent with any of r, s (and similarly for s). The following justifies the terminology:

Proposition 3.8. A normal rewriting system $\mathcal{S}$ is confusion-free iff, for every object μ of $\mathcal{S}$, $\mathrm{Ev}(\mu)$ is confusion-free.

3.3. *Some separation results using embeddings*

A fundamental property of embedding is that they preserve confusion:

Proposition 3.9. Let E, E' be event structures with E containing a confusion. Then, $E \hookrightarrow E'$ implies that also E' contains a confusion.

We therefore have an immediate separation result: a non-confusion-free normal rewriting system $\mathcal{S}$ is strictly more expressive than a confusion-free normal rewriting system $\mathcal{S}'$, in the sense that some objects of $\mathcal{S}$ have computational dynamics which cannot be simulated by any object of $\mathcal{S}'$. Formally:

Corollary 3.10. Let $\mathcal{S}, \mathcal{S}'$ be normal rewriting systems, with $\mathcal{S}$ non-confusion-free and $\mathcal{S}$ confusion-free. Then, there is no embedding of $\mathrm{Ev}(\mathcal{S})$ into $\mathrm{Ev}(\mathcal{S}')$, where the event structures associated to the two systems are those defined at the end of Sect. 2.2.

Another easy separation result may be obtained by restricting a bit the notion of embedding. 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 $u' \longrightarrow_\iota u'_1 \longrightarrow_\iota u'_2 \longrightarrow_\iota \cdots$ in E'.

‡ They are called *cells* in the original definition (Varacca et al., 2006), but this would be in conflict with the standard terminology for interaction nets (*cf.* Sect. 4).

Embeddings not introducing divergence preserve groups of events in pairwise immediate conflicts. In the following, an *anticlique* of an event structure E is a finite set $A \subseteq |E|$ such that $a, b \in A$ implies $a \smile b$, and there exists $u \in \mathcal{C}(E)$ enabling all events of A.

Proposition 3.11. Let $E \overset{\iota}{\hookrightarrow} E'$ without introducing divergence, and let A be an anticlique of E. Then, there exists an anticlique of E' of the same cardinality as A.

Proof. Let $\mathcal{B}$ be a bisimulation associated with ι, and let $u \in \mathcal{C}(E)$ be the configuration enabling $A = \{a_1, \ldots, a_n\}$. We must have $(u, \phi, u') \in \mathcal{B}$ and, since ι does not introduce divergence, we may take a sequence of administrative transitions of maximal length $u' \Longrightarrow_\iota u''$, *i.e.*, such that there is no administrative transition starting from u''. Then, since $\mathcal{B}$ is a bisimulation, and since for all $1 \leq i \leq n$, we have $u \xrightarrow{a_i}_\iota u \cup \{a_i\}$, we must have, for all $1 \leq i \leq n$, 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 $a'_i \smile a'_j$, so $\{a'_1 \ldots, a'_n\}$ is an anticlique of E'. □

We may define the *degree of non-determinism* of a normal rewriting system $\mathcal{S}$ as the smallest $M \in \mathbb{N} \cup \{\infty\}$ such that the cardinality of all anticliques of $\mathrm{Ev}(\mathcal{S})$ is bounded by M (equivalently, for every net μ of $\mathcal{S}$, no anticlique of $\mathrm{Ev}(\mu)$ has cardinality above M). Then, we immediately have the following:

Corollary 3.12. Let $\mathcal{S}, \mathcal{S}'$ be normal rewriting systems, with respective degrees of non-determinism $M > M'$. Then, every embedding of $\mathrm{Ev}(\mathcal{S})$ into $\mathrm{Ev}(\mathcal{S}')$ introduces divergence.

4. An Application to Interaction Nets

4.1. *Nets*

There are several possibilities for formally defining interaction nets. Originally, Lafont (1990) proposed a term calculus, later developed by Fernández and Mackie (1999); he also gave a formalization closer in spirit to the graphical representations which are often used in the literature (Lafont, 1997). More recently, de Falco (2009) proposed another formalization in terms of permutations.

However, none of these definitions covers all of the non-deterministic variants that have been introduced for interaction nets, especially the multiport (Alexiev, 1999; Mazza, 2005), and multiwire (Alexiev, 1999; Beffara and Maurel, 2006) extensions. Hence, we propose here a general formalization, borrowing ideas from Lafont's original definition and de Falco's work. Of course, we shall also present the usual graphical representation of nets, which is the one we use in practice.

Let us start by introducing some notations. In what follows, we fix a denumerably infinite set $\mathbb{P}$ of *ports*, ranged over by p, q, which we assume to contain the set of non-negative integers $\mathbb{N}$. A function $f : \mathbb{P} \to \mathbb{P}$ is said to be *finite* if it is the identity almost everywhere, in which case we write $f : \mathbb{P} \overset{\text{fin}}{\to} \mathbb{P}$.

We denote by $\mathcal{P}_{\text{fin}}(X)$ and $\mathcal{M}(X)$ the set of finite subsets and finite multisets of a set X, respectively. Given a finite set or multiset X, we denote by $\sharp X$ its cardinality. A multiset containing the (non necessarily distinct) elements $x_1, \ldots, x_n$ will be denoted by

$[x_1, \dots, x_n]$. Given $M, M' \in \mathcal{M}(X)$, we denote by $M + M'$ their multiset union. If X, Y are sets, we use the notation $X \uplus Y$ to denote the union of X and Y and to simultaneously state that they are disjoint.

We denote by $\mathcal{L}(X)$ the set of repetition-free finite lists of elements of a set X. Given $L \in \mathcal{L}(X)$, we denote by $\sharp L$ its length and, if $1 \leq i \leq \sharp L$, we denote by L_i its i-th element. We also "overload" set-theoretic symbols so that they operate on lists: if $x \in X$ and $L, L' \in \mathcal{L}(X)$, $x \in L$ means that there exists i such that $L_i = x$, whereas $L \cup L'$ and $L \cap L'$ are the sets resulting from, respectively, the union and the intersection of the sets underlying L and L', with $L \uplus L'$ meaning also that L, L' have no common element. Finally, given $L, L' \in \mathcal{L}(X)$ such that $L \cap L' = \emptyset$, we denote by $L \cdot L'$ their concatenation.

If X, Y are sets and $f : X \to Y$, we denote by $\mathcal{M}(f)$ and $\mathcal{L}(f)$ the functions from $\mathcal{M}(X)$ to $\mathcal{M}(Y)$ and from $\mathcal{L}(X)$ to $\mathcal{L}(Y)$, respectively, which result from the pointwise application of f.

Definition 4.1 (Alphabet). An *alphabet* is a triple $\Sigma = (|\Sigma|, \mathrm{ar}, \mathrm{coar})$, where $|\Sigma|$ is a set of *symbols*, ranged over by α, β, $\mathrm{ar} : |\Sigma| \to \mathbb{N}$ is the *arity* function, and $\mathrm{coar} : |\Sigma| \to \mathbb{N} \setminus \{0\}$ is the *coarity* function. An alphabet Σ is said to be *finite* if $|\Sigma|$ is finite.

Definition 4.2 (Cell base). A *cell base* on an alphabet Σ is quadruple $C = (|C|, \ell, \mathrm{pal}, \mathrm{pax})$, where $|C|$ is a finite set, $\ell : |C| \to |\Sigma|$, and $\mathrm{pal}, \mathrm{pax} : |C| \to \mathcal{L}(\mathbb{P})$, such that:

— for all $c \in |C|$, $\sharp\,\mathrm{pal}(c) = \mathrm{coar}(\ell(c))$, $\sharp\,\mathrm{pax}(c) = \mathrm{ar}(\ell(c))$, and $\mathrm{pal}(c) \cap \mathrm{pax}(c) = \emptyset$;
— for all $c, c' \in |C|$, with $c \neq c'$, $(\mathrm{pal}(c) \cup \mathrm{pax}(c)) \cap (\mathrm{pal}(c') \cup \mathrm{pax}(c')) = \emptyset$.

The elements of $|C|$ are called *hypercells*; given $c \in |C|$, $\ell(c)$ is its *symbol*, and $\mathrm{pal}(c), \mathrm{pax}(c)$ are its lists of *principal* and *auxiliary* ports, respectively. A hypercell is simply called *cell* if the coarity of its symbol is equal to 1; otherwise, it is called a *multicell*. Given a cell base C, we define its *support* to be $\mathrm{supp}\, C = \bigcup_{c \in |C|} (\mathrm{pal}(c) \cup \mathrm{pax}(c))$.

Given two cell bases C, C' on the same alphabet Σ and a pair of functions $\varphi : |C| \to |C'|$, $\xi : \mathbb{P} \overset{\mathrm{fin}}{\to} \mathbb{P}$, we say that (φ, ξ) is a *morphism* from C to C' whenever $\ell' \circ \varphi = \ell$, $\mathrm{pal}' \circ \varphi = \mathcal{L}(\xi) \circ \mathrm{pal}$, and $\mathrm{pax}' \circ \varphi = \mathcal{L}(\xi) \circ \mathrm{pax}$.

Definition 4.3 (Wire structure). A *wire structure* is a pair $W = (|W|, \partial)$, where $|W|$ is a finite set and $\partial : |W| \to \mathcal{M}(\mathbb{P})$, such that each $p \in \mathbb{P}$ occurs at most twice in the multiset $\nabla = \sum_{w \in |W|} \partial(w)$. The elements of $|W|$ are called *hyperwires*; if $w \in |W|$, the non-negative integer $\sharp\partial(w)$ is referred to as its *order*. A hyperwire of order 2 is simply called *wire*; otherwise, we say it is a *multiwire*. Given a wire structure W, we define its *set of ports* (resp. its *support*), denoted by $\mathrm{pts}\, W$ (resp. $\mathrm{supp}\, W$), to be the set of all ports occurring (resp. occurring exactly once) in ∇.

If W, W' are two wire structures, and if $\psi : |W| \to |W'|$ and $\xi : \mathbb{P} \overset{\mathrm{fin}}{\to} \mathbb{P}$ are two functions, we say that (ψ, ξ) is a morphism from W to W' whenever $\partial' \circ \psi = \mathcal{M}(\xi) \circ \partial$.

Definition 4.4 (Net). A *net* on an alphabet Σ is a triple $\mu = (F, C, W)$, where $F \in \mathcal{L}(\mathbb{P})$, C is a cell base on Σ, and W is a wire structure such that $\mathrm{supp}\, W = F \uplus \mathrm{supp}\, C$. F is called the *interface* of μ. The *set of ports* of μ is defined as $\mathrm{pts}\, \mu = \mathrm{pts}\, W$. We denote by $\mathcal{N}(\Sigma)$ the set of all nets on the alphabet Σ.

Fig. 1. Graphical representations of a hypercell (left) and a hyperwire (right).

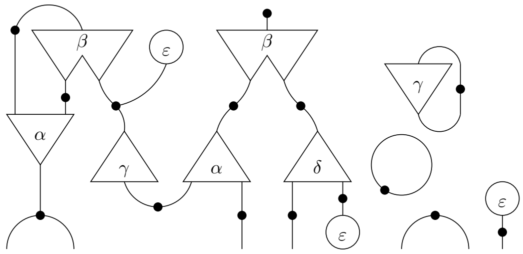


Fig. 2. Graphical representation of a net.

A *morphism* from a net μ to a net μ' is a triple (φ, ψ, ξ) such that $(\varphi, \xi) : C \to C'$ is a morphism of cell bases, $(\psi, \xi) : W \to W'$ is a morphism of wire structures, and, additionally, $F' = \mathcal{L}(\xi)(F)$.

By definition, if $p \in \operatorname{pts} \mu$, exactly one of the following holds: $p \in F$, in which case we say it is *free*; $p \in \operatorname{supp} C$, in which case there exists $c \in |C|$ such that $p \in \operatorname{pal}(c) \uplus \operatorname{pax}(c)$, and we say that p is *principal* (resp. *auxiliary*) if $p \in \operatorname{pal}(c)$ (resp. $p \in \operatorname{pax}(c)$); none of the above holds, in which case we say that p is *hidden*.

For nearly all of our purposes, two isomorphic nets are perfectly equivalent. Indeed, this notion of isomorphism is nothing but a reformulation of what is usually called α-*equivalence* in calculi with variable binders. For this reason, we shall use "α-equivalent" and "isomorphic" synonymously. Most of the literature on interaction nets is based on graphical representations equating isomorphic nets. In principle, this bears some analogy with the string diagram representation of morphisms in monoidal categories (or categories with structures enriching it).

Let c be a hypercell such that $\ell(c) = \alpha$, with $\operatorname{coar}(\alpha) = m, \operatorname{ar}(\alpha) = n$, and let $\operatorname{pal}(c) = (p_0, \ldots, p_{m-1})$, $\operatorname{pax}(c) = (q_1, \ldots, q_n)$; the graphical representation of c is given in Fig. 1, left. Let now w be a hyperwire such that $\partial(w) = [p_1, \ldots, p_k]$; the graphical representation of w is given in Fig. 1, right. Then, an example of graphical representation of a net will look like Fig. 2. Since graphical representations are meant to implicitly deal with isomorphism, the names of ports are omitted, and a port appears simply as a linear stroke. The information concerning the ordering of the free ports is given by making the convention that, in graphical representations, free ports are always ordered "from left to right". Also observe that, to economize space, hypercells with only one principal port and no auxiliary ports are represented by circles.

From now on, nets will always be treated modulo α-equivalence.

Fig. 3. ω-reduction; the hyperwire in the left member of the right rule is of order zero, and the two hyperwires in the left member of the left rule are distinct.

4.2. *Interaction*

The most elementary notion of dynamics in interaction nets concerns wire structures alone. It can be seen as an attempt to eliminate hidden ports; de Falco (2009) refers to it as “port fusion”. It is also related to the elimination of cuts involving axioms in linear logic proof nets.

Definition 4.5 (ω-reduction). Let W, W' be wire structures; we write $W \to_\omega W'$ iff one of the following holds:

— $|W| = |W'| \uplus \{w\}$, $\partial(w) = []$, and $\partial = \partial'$ everywhere else;

— $|W| = X \uplus \{w_1, w_2\}$ for some set X, with $w_1 \neq w_2$, $\partial(w_1) = M_1 + [h]$, $\partial(w_2) = M_2 + [h]$ for some multisets M_1, M_2 and port h, and $|W'| = X \uplus \{w'\}$, $\partial' = \partial$ on X and $\partial'(w') = M_1 + M_2$.

Let now $\mu = (F, C, W)$ be a net, and let $W \to_\omega W'$. It is not hard to verify that $\mu' = (F, C, W')$ is a net; then, we write $\mu \to_\omega \mu'$.

Graphically, ω-reduction is depicted in Fig. 3, with the left and right rules corresponding to the first and second point of Definition 4.5, respectively.

Proposition 4.6. ω-reduction is strongly confluent and strongly normalizing.

Proof. The only kind of critical pair is easily seen to be strongly confluent (modulo α-equivalence), and whenever $W \to_\omega W'$, we have $\sharp|W| > \sharp|W'|$. $\square$

Proposition 4.6 guarantees that every net has a unique ω-normal form. As a matter of fact, the theory of interaction nets is usually concerned with ω-normal nets only. For instance, the net in Fig. 2 is ω-normal. An ω-normal net may still have hidden ports: this happens precisely when it contains some hyperwire w such that $\partial(w) = M + [h, h]$ for some multiset M and some port h, which is obviously hidden. Such configurations are called *loops*.

Even though we ultimately wish to deal only with ω-normal nets, non-ω-normal nets still play a crucial technical role in the definition of the actual computational dynamics of interaction nets. Such dynamics comes from the rewriting of *active pairs*, which, informally, are pairs of distinct hypercells such that a principal port of each is connected to the same hyperwire. Intuitively, if hypercells are agents, the fundamental principle of interaction nets is that such agents can only communicate in pairs, through one of their principal ports.

In the following, if C, C' are two cell bases on the same alphabet such that $|C| \cap |C'| = \emptyset$, we define $C \uplus C' = (|C| \cup |C'|, \ell \uplus \ell', \mathrm{pal} \uplus \mathrm{pal}', \mathrm{pax} \uplus \mathrm{pax}')$, where $f \uplus g$ denotes the function obtained by the union of functions of disjoint domain. Similarly, given two wire structures W, W' such that $|W| \cap |W'| = \emptyset$, we define $W \uplus W' = (|W| \cup |W'|, \partial \uplus \partial')$.

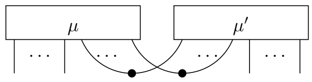


Fig. 4. Glueing of two nets.

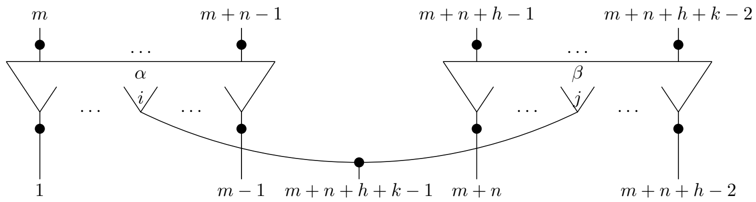


Fig. 5. An active pair. In the picture, $m = \mathrm{coar}(\alpha)$, $n = \mathrm{ar}(\alpha)$, $h = \mathrm{coar}(\beta)$, and $k = \mathrm{ar}(\beta)$

Definition 4.7 (Glueing, context). Let $\mu = (F, C, W), \mu' = (F', C', W')$ be two nets, let $0 \leq n \leq \min\{\sharp F, \sharp F'\}$, and let $F = F_0 \cdot (p_1, \dots, p_n)$, $F' = (p'_1, \dots, p'_n) \cdot F'_0$. By α-equivalence, we can always suppose the intersections $\mathrm{pts}\,\mu \cap \mathrm{pts}\,\mu'$, $|C| \cap |C'|$, and $|W| \cap |W'|$ to be all empty. So let $W_1 = W \uplus W'$; we define a new wire structure W_0 as follows: $|W_0| = |W_1| \uplus \{w_1, \dots, w_n\}$, $\partial_0 = \partial_1$ on $|W_1|$, and $\partial_0(w_i) = [p_i, p'_i]$, for $1 \leq i \leq n$. We then define the *n-glueing* of μ and μ', denoted by $\mu \overset{n}{\leftrightsquigarrow} \mu'$, as $(F_0 \cdot F'_0, C \uplus C', W_0)$, which is easily seen to be a net.

Whenever μ has n free ports and Γ is a net with at least n free ports, we say that Γ is a context for μ, and we use the notation $\Gamma[\mu]$ for the net $\Gamma \overset{n}{\leftrightsquigarrow} \mu$.

Graphically, the glueing of two nets μ, μ' is depicted in Fig. 4. Note that $\mu \overset{0}{\leftrightsquigarrow} \mu'$ is nothing but the "juxtaposition" of μ and μ'. Observe also that, whenever $n \geq 1$, $\mu \overset{n}{\leftrightsquigarrow} \mu'$ need not be ω-normal, even in case μ, μ' are.

Definition 4.8 (Active pair). An *active pair* on the alphabet Σ is a net of the form given in Fig. 5, where, if c, d are the hypercells whose respective symbols are α, β, we have $c \neq d$ and the only hyperwire of order 3 is connected to the ith principal port of c and to the jth principal port of d. For graphical convenience, instead of adopting the usual convention that free ports are ordered "from left to right", we explicitly labelled the free ports with their indices. An active pair is completely determined by a pair of symbols $\alpha, \beta \in \Sigma$, and by two integers $1 \leq i \leq \mathrm{coar}(\alpha)$, $1 \leq j \leq \mathrm{coar}(\beta)$. Thus, we shall denote such an active pair by $\alpha_i \bowtie \beta_j$, and we denote by $\mathrm{Act}(\Sigma)$ the set of active pairs on the alphabet Σ.

In the following, if $\mu = (F, C, W)$ is a net such that $\sharp F = n$, and if σ is a permutation on $\{1, \dots, n\}$, we denote by $\sigma\mu$ the net (F', C, W) such that $F'_i = F_{\sigma(i)}$, $1 \leq i \leq n$. If $X = \{\mu_1, \dots, \mu_n\}$ is a set of nets with interfaces of the same cardinality, we write $\sigma X = \{\sigma\mu_1, \dots, \sigma\mu_n\}$.

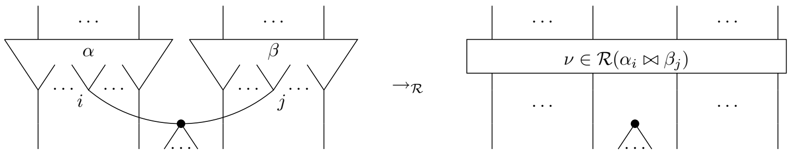


Fig. 6. Reducing an active pair; the two cells on the left must be distinct, and the hyperwire connected to their principal ports i and j is of order $k+2$, whereas the hyperwire on the right is of order k.

Definition 4.9 (Interaction scheme). An *interaction scheme* on the alphabet Σ is a function $\mathcal{R} : \mathrm{Act}(\Sigma) \to \mathcal{P}_{\mathrm{fin}}(\mathcal{N}(\Sigma))$ such that, for all $\alpha_i \bowtie \beta_j \in \mathrm{Act}(\Sigma)$ with $\mathrm{coar}(\alpha) + \mathrm{ar}(\alpha) = m$ and $\mathrm{coar}(\beta) + \mathrm{ar}(\beta) = n$, we have:

— $\mu \in \mathcal{R}(\alpha_i \bowtie \beta_j)$ implies that the interface of μ has cardinality $m+n-1$ (*i.e.*, the same as the interface of $\alpha_i \bowtie \beta_j$);
— $\mathcal{R}(\beta_j \bowtie \alpha_i) = \sigma\mathcal{R}(\alpha_i \bowtie \beta_j)$, where σ is the permutation on $\{1, \ldots, m+n-1\}$ defined by $(m, \ldots, m+n-2, 1, \ldots, m-1, m+n-1)$;
— if F is the interface of a net $\mu \in \mathcal{R}(\alpha_i \bowtie \beta_j)$, then the free port F_{m+n-1} is connected to a hyperwire of order 1.

An interaction scheme on an alphabet Σ induces a rewriting relation on the nets of $\mathcal{N}(\Sigma)$, as follows.

Definition 4.10 (Reduction). Let Σ be an alphabet, let $\mathcal{R}$ be an interaction scheme on Σ, and let $\mu, \mu' \in \mathcal{N}(\Sigma)$. We write $\mu \rightsquigarrow_{\mathcal{R}} \mu'$ if there exist $\mu_0 \in \mathrm{Act}(\Sigma)$ and a net Γ such that $\mu = \Gamma[\mu_0]$ and $\mu' = \Gamma[\nu]$, with $\nu \in \mathcal{R}(\mu_0)$.

Let now μ, μ' be two ω-normal nets on Σ. We write $\mu \to_{\mathcal{R}} \mu'$ whenever there exist μ_1, μ_1' such that $\mu \,{}^{*}_{\omega}\!\!\leftarrow \mu_1 \rightsquigarrow_{\mathcal{R}} \mu_1' \to^{*}_{\omega} \mu'$.

We shall denote reduction simply by $\to$ when the interaction scheme $\mathcal{R}$ is clear from the context or is not important.

Graphically, reduction can be depicted as in Fig. 6, where the context around the active pair is left implicit. Observe that loops may appear through reduction; this is why they have been allowed in the definition of net. One of the insights given by the original definition of interaction nets is that, in a sense, logical correctness is all about avoiding the apparition of loops. We shall not be concerned with this issue here; we refer the reader to Lafont (1990), in which a Danos-Regnier-like correctness criterion (Danos and Regnier, 1989) is used to ensure loop-freeness.

We may end this section by giving the definition of interaction net system:

Definition 4.11 (Interaction net system). An *interaction net system* is a triple $(\Sigma, \mathcal{R}, \mathcal{N})$ where Σ is an alphabet, $\mathcal{R}$ is an interaction scheme on Σ, and $\mathcal{N} \subseteq \mathcal{N}(\Sigma)$ is such that $\mu \in \mathcal{N}$ implies that μ is ω-normal, and $\mu \to_{\mathcal{R}} \mu'$ implies $\mu' \in \mathcal{N}$. The set $\mathcal{N}$ is called the set of *admissible nets* of the system.

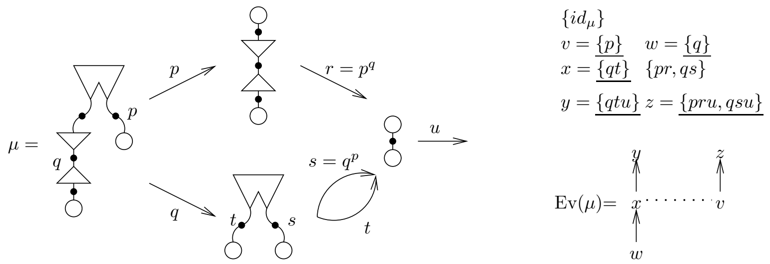


Fig. 7. The event structure associated with a net. The interaction rules are not specified, as they can be easily inferred from the reductions. When more than one radical is present in a net, we annotate it beside its corresponding active pair. On the top right, we enumerate all homotopy classes, underlining and naming those that are essential, which appear as the events of $\mathrm{Ev}(\mu)$. In the graphical representation of $\mathrm{Ev}(\mu)$, an arrow from x to y means $x \leq y$, while the dotted line represents immediate conflict.

4.3. *Interaction nets and event structures*

Given an interaction net system $\mathcal{S} = (\Sigma, \mathcal{R}, \mathcal{N})$, its rewriting relation generates a graph $\mathcal{G}_\mathcal{S}$: the nodes are the nets of $\mathcal{S}$, and there is an arrow r such that $\mathrm{src}(r) = \mu$, $\mathrm{trg}(r) = \nu$ iff there is an active pair a in μ and a reduction rule in $\mathcal{R}(a)$ applying which we obtain $\mu \to_\mathcal{R} \nu$. Note that "multiplicities" count: if an active pair in μ can be reduced in two ways, both yielding ν, or if μ rewrites into the same net ν by reducing two distinct active pairs, then in $\mathcal{G}_\mathcal{S}$ there will be two arrows of source μ and target ν. In other words, an arrow of $\mathcal{G}_\mathcal{S}$ is given by an active pair *and* a way to reduce it.

This graph can be turned into a rewriting system, as follows. Consider a net μ and two arrows r, s of $\mathcal{G}_\mathcal{S}$ coinitial in μ, such that the active pairs associated with r and s are *disjoint*, *i.e.*, they do not share any cell. Then, if $\mu \to_\mathcal{R} \nu$ by r, the locality of interaction rules allows us to unambiguously identify in ν an active pair corresponding to the one associated with s: it is "the same" active pair as that of s, and there is therefore "the same" way to reduce it as that associated with s. This yields an arrow t of source ν in $\mathcal{G}_\mathcal{S}$, which we shall identify with the residue of s through r. The residue structure $R_\mathcal{S}$ is then defined by $(r, s, t) \in R_\mathcal{S}$ iff the active pairs associated with r, s belong to the same net and are disjoint, and t is induced by s after reducing r as described above.

The following is proved by a simple verification, which we leave to the reader:

Proposition 4.12. For every interaction net system $\mathcal{S}$, the rewriting system $(\mathcal{G}_\mathcal{S}, R_\mathcal{S})$ is normal.

In the sequel, we shall abusively denote by $\mathcal{S}$ the rewriting system associated with an interaction net system $\mathcal{S}$.

Proposition 4.12 allows us to associate an event structure $\mathrm{Ev}(\mu)$ with every interaction net μ, as explained in Sect. 2.2. An example is given in Fig. 7. The notions and results of Sect. 3 then offer us an approach to studying the expressiveness of interaction nets.

Also observe that the notion of bisimilar embedding of Sect. 3 may be seen as a very general notion of *translation* between interaction net systems. In fact, let $\mathcal{S}, \mathcal{S}'$ be

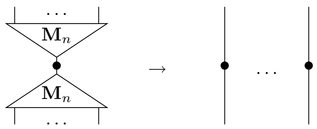


Fig. 8. Interaction scheme $\mathcal{R}$ for the system $\mathcal{M}$: $\mathcal{R}(\mathbf{M}_m \bowtie \mathbf{M}_n)$ is empty if $m \neq n$, and is a singleton containing the net above on the right if $m = n$.

interaction net systems such that $\mathrm{Ev}(\mathcal{S}) \overset{\iota}{\hookrightarrow} \mathrm{Ev}(\mathcal{S}')$; then, every net μ of $\mathcal{S}$ admitting a reduction induces a radical r of source μ; but $[r]$ is essential, so we have a set $\iota([r])$ of essential reductions of $\mathcal{S}'$; the sources of these reductions may be seen as translations of μ; the existence of a ι-bisimulation then guarantees that these translations have "the same" computational dynamics as μ, at least as far as its description in terms of event structures is concerned.

It is worth noticing that the concrete notion of translation defined by Lafont (1997) in the restricted case of what we call Lafont interaction net systems (see Sect. 4.4 below) yields a translation in our sense, *i.e.*, it is not hard to construct from each Lafont translation of $\mathcal{S}$ into $\mathcal{S}'$ a bisimilar embedding $\mathrm{Ev}(\mathcal{S}) \overset{\iota}{\hookrightarrow} \mathrm{Ev}(\mathcal{S}')$ (and, in this particular case, ι will be a function).

4.4. *Lafont interaction nets*

A *Lafont interaction net system* (Lafont, 1990) is a system $(\Sigma, \mathcal{R}, \mathcal{N})$ such that $\mathrm{coar}(\alpha) = 1$ for all $\alpha \in \Sigma$, $\mathcal{R}$ is deterministic, and no net of $\mathcal{N}$ uses multiwires. In such interaction nets, active pairs are always disjoint, so we immediately obtain the following:

Proposition 4.13. Let μ be a net of a Lafont interaction net system. Then, $\mathrm{Ev}(\mu)$ is conflict-free, *i.e.*, it is a poset.

We shall see that Lafont interaction nets are expressive enough to generate all finite posets.

Let M be the alphabet $(\{\mathbf{M}_n \mid n \in \mathbb{N}\}, \mathrm{ar}, \mathrm{coar})$ such that $\mathrm{ar}(\mathbf{M}_n) = n$, $\mathrm{coar}(\mathbf{M}_n) = 1$. We define an interaction net system $\mathcal{M} = (M, \mathcal{R}, \mathcal{N}(M))$ by using the interaction scheme $\mathcal{R}$ described in Fig. 8. A cell of type $\mathbf{M}_n$ interacting as in Fig. 8 is called a *multiplexor*; system $\mathcal{M}$ is basically multiplicative linear logic proof nets, with self-dual connectives of arbitrary arities. In fact, multiplicative linear logic proof nets were the founding example out of which Lafont built his more general idea of interaction net.

Consider now a finite poset $(X, \leq)$. We write $x <_1 y$ when y covers x, *i.e.*, $x < y$ and there is no z such that $x < z < y$. Then, we set $\mathrm{pred}(x) = \{x' \in X \mid x' <_1 x\}$, $\mathrm{succ}(x) = \{x' \in X \mid x <_1 x'\}$. From this, we define a net $\mu_X \in \mathcal{N}(M)$, as follows. We set $|C_X| = \{x^+, x^- \mid x \in X\}$, $|W_X| = \{w_1^x, \ldots, w_{m+1}^x \mid x \in X,\ \sharp\,\mathrm{pred}(x) = m\}$, and we define the cell base $C_X = (|C|, \ell, \mathrm{pal}, \mathrm{pax})$ and the wire structure $W_X = (|W|, \partial)$ such

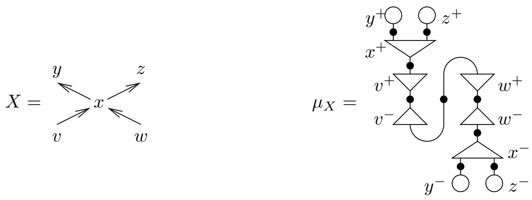


Fig. 9. A poset (left), and its associated net in system $\mathcal{M}$ (right). An arrow from x to y means $x \leq y$.

that, if $\mathrm{succ}(x) = \{y_1, \ldots, y_n\}$ and $\mathrm{pred}(x) = \{z_1, \ldots, z_m\}$, we have

$$\begin{array}{l} \ell(x^+) = \ell(x^-) = \mathbf{M}_n \\ \mathrm{pal}(x^+) = (p_x^+) \qquad \mathrm{pax}(x^+) = (q_{x,y_1}^+, \ldots, q_{x,y_n}^+) \\ \mathrm{pal}(x^-) = (p_x^-) \qquad \mathrm{pax}(x^-) = (q_{x,y_n}^-, \ldots, q_{x,y_1}^-) \\ \partial(w_1^x) = [p_x^+, q_{z_1,x}^+] \quad \partial(w_2^x) = [q_{z_1,x}^-, q_{z_2,x}^+] \quad \ldots \\ \ldots \quad \partial(w_m^x) = [q_{z_{m-1},x}^-, q_{z_m,x}^+] \quad \partial(w_{m+1}^x) = [q_{z_m,x}^-, p_x^-] \end{array}$$

where it is intended that, if $\mathrm{pred}(x) = \emptyset$, $\partial(w_1^x) = [p_x^+, p_x^-]$. We set $\mu_X = (\emptyset, C_X, W_X)$. An example is given in Fig. 9 (symbols of cells are omitted; they are uniquely determined by the arity of cells).

Note that z is a minimal element of X iff z^+, z^- form an active pair in μ_X. Furthermore, if $\mathrm{pred}(x) = \{z_1, \ldots, z_n\}$ such that all z_k are minimal, then there will be an active pair formed by x^+, x^- iff all the active pairs z_k^+, z_k^- in μ_X are reduced. More generally, if we start from the principal port of x^+, we may find a path in μ_X leading to the principal port of x^- and crossing exactly all cells y^+, y^- for all $y \in \downarrow x$. Thanks to the interaction rule of multiplexors (Fig. 8), all y^+ cells successively annihilate with y^- cells, starting from the minimal y's, and such path gradually transforms into a wire, creating an active pair between x^+ and x^- (the acquainted reader will recognize here an *execution path* of the geometry of interaction).

The above description should convince the reader that $\mathrm{Ev}(\mu_X)$ is isomorphic to $(X, \leq)$. Conversely, it is easy to see that no infinite reduction is possible in $\mathcal{M}$ (the number of cells of nets strictly decreases under reduction), so $\mathrm{Ev}(\mu)$ is finite for all nets of $\mathcal{M}$. With the help of Proposition 4.13, we therefore have the following characterization:

Proposition 4.14. An event structure E is finite and conflict-free iff it is isomorphic to $\mathrm{Ev}(\mu)$ for some net μ of $\mathcal{M}$.

Of course capturing all posets is impossible, because there are too many of them; to characterize the posets generated by Lafont interaction nets one must certainly make some recursiveness assumption. At present, we do not have such a characterization.

Nevertheless, Proposition 4.14 gives us an idea of how expressive Lafont interaction nets are within the realm of strictly deterministic computation, *i.e.*, when all conflicts are absent. Quoting Lafont (1997):

"[...] we can say that [Lafont] interaction nets are a deterministic and asynchronous model of

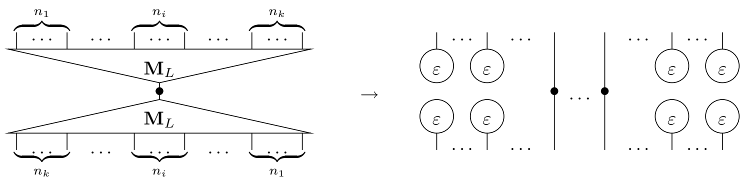


Fig. 10. Interaction scheme $\mathcal{R}$ for system $\mathcal{M}_N$: $\mathcal{R}(\mathbf{M}_{L_1} \bowtie \mathbf{M}_{L_2})$ is non-empty exactly when $L_1 = L_2 = L = (n_1, \dots, n_k)$ with $k > 0$, and in that case it contains k nets defined as above, with $1 \leq i \leq k$.

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".

We may call the above statement "Lafont's thesis"; if we accept the fact that a "deterministic and asynchronous" computational process corresponds to a conflict-free event structure, then Proposition 4.14 provides evidence (albeit limited, because only terminating computations are taken into account) in favor of Lafont's thesis.

4.5. *Multirule interaction nets*

A *multirule interaction net system* (Alexiev, 1999) is a Lafont system $(\Sigma, \mathcal{R}, \mathcal{N})$ in which $\mathcal{R}$ no longer needs to be deterministic. Multirule interaction nets have recently acquired a central position in the proof theory of linear logic, thanks to the introduction of *differential interaction nets* by Ehrhard and Regnier (2006), which are a particular system of multirule interaction nets capable of encoding full linear logic proof nets through the so-called Taylor expansion (de Carvalho, 2007; Tasson, 2009). They were also shown to have promising connections with concurrent computation (Ehrhard and Laurent, 2007).

Let μ be a net of a multirule system $(\Sigma, \mathcal{R}, \mathcal{N})$. If μ contains an active pair $\alpha_0 \bowtie \beta_0$, and if $\mathcal{R}(\alpha_0 \bowtie \beta_0)$ has $k > 1$ elements, we have k radicals $r_1, \dots, r_k$ of source μ; these are not independent, because they all concern the same active pair. Hence, the conflict relation of $\mathrm{Ev}(\mu)$ may be non-empty. However, it always has the structure we introduced in Sect. 3.2, *i.e.*, it is confusion-free:

Proposition 4.15. Let $\mathcal{S}$ be a multirule interaction net system; then, as a normal rewriting system, $\mathcal{S}$ is confusion-free.

Proof. Let r, s be coinitial radicals of $\mathcal{S}$. If μ is their source, there are two cases: either r and s correspond to two distinct active pairs, or they correspond to two different ways of reducing the same active pair. It is easy to see that in the first case, they are separated, and in the second case, they are in simple conflict. □

As in the case of Lafont interaction nets, it is possible to see that multiport interaction nets are complete with respect to finite confusion-free event structures. Let $L = (n_1, \dots, n_k)$ be a finite list of non-negative integers. We set $\|L\| = \sum_{i=1}^{k} n_i$, and we denote by $\mathcal{L}$ the set of all finite list of non-negative integers. Let now N be the alphabet such that $|N| = \{\mathbf{M}_L \mid L \in \mathcal{L}\}$, and let $\mathrm{ar}(\mathbf{M}_L) = \|L\|$, while the coarity of all symbols is 1. We define a multirule interaction net system $\mathcal{M}_N = (N, \mathcal{R}, \mathcal{N}(\mathcal{M}_N))$, where $\mathcal{R}$ is

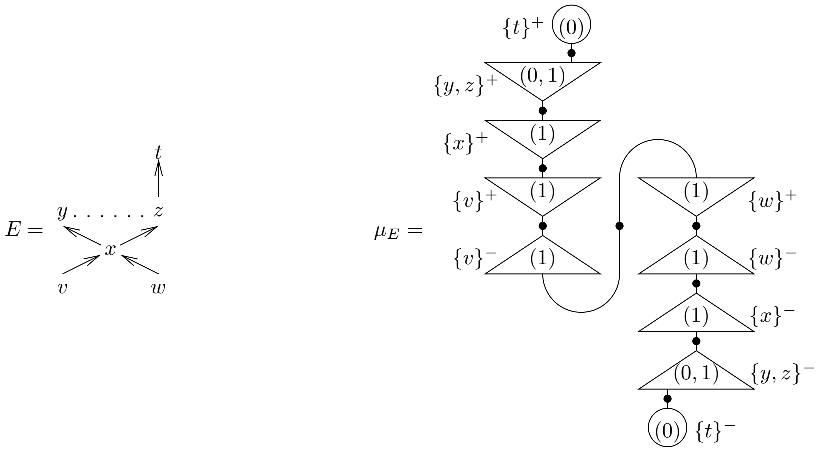


Fig. 11. A finite confusion-free event structure (left), and its associated net in system $\mathcal{M}_N$ (right). In the event structure, causal order is represented as usual by arrows, while the dotted line represents immediate conflict. In the net on the right, the symbols $\mathbf{M}_L$ are shortened into L.

as in Fig. 10, in which, for brevity, we put $\varepsilon = \mathbf{M}_{()}$. The cells of $\mathcal{M}_N$ may be seen as non-deterministic multiplexors; deterministic multiplexors are the special cases $\mathbf{M}_L$ in which L is a singleton.

Now let E be a finite confusion-free event structure. We use the notations pred$(\cdot)$ and succ$(\cdot)$ as in Sect. 4.4, and we denote by sw(E) the set of switches of E (*i.e.*, the equivalence classes of events under the reflexive closure of immediate conflict). If $S = \{x_1, \ldots, x_k\} \in \mathrm{sw}(E)$, we know that, for all $1 \leq i, j \leq k$, $\mathrm{pred}(x_i) = \mathrm{pred}(x_j)$; we denote this set by pred(S). Additionally, for $z \in |E|$, we denote by $S(z)$ the switch of z, *i.e.*, its equivalence class. We then set $|C_E| = \{S^+, S^- \mid S \in \mathrm{sw}(E)\}$, $|W_E| = \{w_1^S, \ldots, w_{m+1}^S \mid S \in \mathrm{sw}(E),\ \sharp\,\mathrm{pred}(S) = m\}$, and we define the cell base $C_E = (|C_E|, \ell, \mathrm{pal}, \mathrm{pax})$ and the wire structure $W_E = (|W_E|, \partial)$ such that, if $S = \{x_1, \ldots, x_k\} \in \mathrm{sw}(E)$, $\mathrm{pred}(S) = \{z_1, \ldots, z_m\}$, and $\mathrm{succ}(x_i) = \{y_1^i, \ldots, y_{n_i}^i\}$, with $1 \leq i \leq k$, we have

$$
\begin{array}{l}
\ell(S^+) = \ell(S^-) = \mathbf{M}_{(n_1,\ldots,n_k)} \\
\mathrm{pal}(S^+) = (p_S^+) \qquad \mathrm{pax}(S^+) = (q^+_{S,S(y_1^1)}, \ldots, q^+_{S,S(y_{n_1}^1)}, \ldots, q^+_{S,S(y_1^k)}, \ldots, q^+_{S,S(y_{n_k}^k)}) \\
\mathrm{pal}(S^-) = (p_S^-) \qquad \mathrm{pax}(S^-) = (q^-_{S,S(y_{n_1}^1)} \ldots, q^-_{S,S(y_1^1)}, \ldots, q^-_{S,S(y_{n_k}^k)}, \ldots, q^-_{S,S(y_1^k)}) \\
\partial(w_1^S) = [p_S^+, q^+_{S(z_1),S}] \quad \partial(w_2^S) = [q^-_{S(z_1),S}, q^+_{S(z_2),S}] \quad \cdots \\
\cdots \quad \partial(w_m^S) = [q^-_{S(z_{m-1}),S}, q^+_{S(z_m),S}] \quad \partial(w_{m+1}^S) = [q^-_{S(z_m),S}, p_S^-]
\end{array}
$$

and it is intended that, if $\mathrm{pred}(S) = \emptyset$, $\partial(w_1^S) = [p_S^+, p_S^-]$. Then, we define $\mu_E = (\emptyset, C_E, W_E)$. Note that the definition is similar to the one given in Sect. 4.4 for Lafont interaction nets; we just replace events with their switches. An example is given in Fig. 11.

Everything we said in the deterministic case carries over to the non-deterministic case, so that, considering Proposition 4.15, we obtain

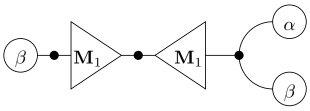


Fig. 12. A multiwire net generating an event structure isomorphic to that of Fig. 7. The cells of type $\mathbf{M}_1$ interact just like described in Fig. 8; the interaction between two β cells is undefined; whereas the interaction between α and β simply erases those two cells.

Proposition 4.16. An event structure E is finite and confusion-free iff it is isomorphic to $\mathrm{Ev}(\mu)$ for some net μ of $\mathcal{M}_N$.

4.6. *Comparing non-deterministic classes of interaction nets*

There are at least two other classes of interaction nets which have been considered in the literature: *multiport interaction nets* (Alexiev, 1999; Mazza, 2005) and *multiwire interaction nets* (Alexiev, 1999; Beffara and Maurel, 2006); both of them have deterministic interaction schemes, but achieve non-determinism thanks to the morphology of nets.

Let $(\Sigma, \mathcal{R}, \mathcal{N})$ be an interaction net system, with $\mathcal{R}$ deterministic. We say that $\mathcal{S}$ is *multiport* if no net of $\mathcal{N}$ contains multiwires; we say that it is *multiwire* if Σ has only symbols of coarity 1. In other words, we take Lafont interaction nets, and in one case, we allow multicells, in the other, we allow multiwires—whence the respective names.

The example of Fig. 7 shows that multiport nets may exhibit confusion; Fig. 12 shows that this is also the case with multiwire nets. Therefore, modulo Proposition 4.15, we have a first concrete application of Corollary 3.10: multirule interaction nets are strictly less expressive than multiport or multiwire interaction nets. This separation result was claimed by Alexiev (1999), but his proof relies on an extremely strict notion of encoding. For example, because he lacked a notion of bisimulation for interaction nets, to ensure soundness Alexiev forced his encodings to preserve the principal or auxiliary nature of ports in nets. Our result is based on more general grounds and, in our opinion, explains more deeply why such separation holds.

An interesting question about "non-Lafont" interaction nets, *i.e.*, interaction nets having some form of non-determinism, concerns the existence of a system of universal combinators, like the interaction combinators (Lafont, 1997) for Lafont interaction nets. A system is *universal* for a class of interaction nets iff all interaction net systems of that class can be translated into it (in the sense of Sect. 4.3). What is interesting about Lafont's interaction combinators is that they are a finite system: the alphabet contains only three symbols, and the interaction scheme defines all six possible interactions between them. Is this possible in the multirule and multiport case?[§] The results of Sect. 3.3 give a partially negative answer, at least according to our notion of embedding.

First of all, observe that, if an alphabet Σ is finite, then any multirule system on Σ yields a normal rewriting system with finite degree of non-determinism (as defined

[§] The question does not make sense for multiwire nets, because their non-determinism does not come from hypercells.

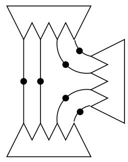


Fig. 13. A maximal anticlique in a multiport system of maximum coarity 4.

in Sect. 3.3), which is given by the active pair having the highest number of possible ways to reduce it. In fact, every non-deterministic active pair gives rise to an anticlique, whose cardinality is equal to the number of its possible reductions. Then, Corollary 3.12 immediately gives us that any universal system of multirule interaction nets must in general introduce divergence, because any such system has a bounded degree of non-determinism, and there will always be a multirule system with a greater degree.

Multiport systems suffer from a similar problem:

Lemma 4.17. Let Σ be a finite alphabet, and let $m = \max\{\mathrm{coar}(\alpha) \mid \alpha \in |\Sigma|\}$. Then, the normal rewriting system induced by any multiport interaction net system on Σ has degree of non-determinism bounded by $\lfloor \frac{3m}{2} \rfloor$ (where $\lfloor x \rfloor$ is the greatest integer not greater than x).

The upper bound of Lemma 4.17 can be reached; Fig. 13 gives an example with $m = 4$.

5. Discussion and Further Work

The significance and scope of the present work depends on the following questions:

(i) how many computational models can be rephrased in terms of normal rewriting systems?

(ii) how sensible is our notion of bisimilar embedding?

We start by observing that, apart from the classical examples of Turing machines (deterministic or not) and Petri nets, several process calculi (CCS, π-calculus, solos (Laneve and Victor, 2003)), when considered under their standard reduction relation, naturally yield normal rewriting systems. However, no system allowing duplication of radicals is normal, as this clashes with the affinity axiom. For instance, the standard residue structure associated with the λ-calculus is not pre-normal: in $\Delta(II) \rightarrow_\beta II(II)$, in the second term there are two residues of the radical induced by the redex II in the first term. Proof nets present an identical issue, because of exponential boxes. Note that, on the contrary, process calculi do not violate affinity, even in presence of replication, because this is formulated precisely so as to avoid duplication of reductions.

We point out that this limitation is somehow intrinsic in event structures: if two events a and b are both enabled, there is no way of saying that the occurrence of a "duplicates" b. Instead, this is modeled by saying that a and b are in conflict, and that, after "destroying" b, the occurrence of a enables two "new" events b_1, b_2, which must be seen as the "copies" of b. Observe that the symmetry axiom is also imposed by event structures, and reflects the symmetry of the conflict relation.

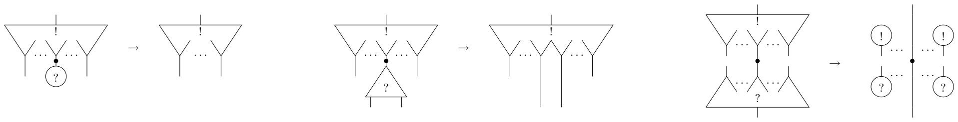


Fig. 14. Rules for multidereliction (?) and multicodereliction (!). The coarity of each symbol is implicitly given by the graphical representation. In the leftmost two rules, the nullary and binary cells are a weakening and a contraction, respectively; these rules have a dual version, which we omit.

There are two possible reactions to this situation. The first, is asserting that event structures describe computation in an accurate way, and that the λ-calculus reduction is somehow "artificial". In fact, these problems disappear as soon as one fixes an evaluation strategy, *e.g.* call-by-name or call-by-value for the λ-calculus. Moreover, the λ-calculus *can* be endowed with a normal residue structure: it is enough to say that the redex $(\lambda x.M)N$ has no residue through reduction of any of the redexes in N, and vice versa.

The second, is to abandon event structures, and work directly with a broader class of rewriting systems, possibly in which a well-behaved notion of homotopy can still be defined. For instance, the above duplication in the λ-calculus may be treated by asserting the existence of an "asymmetric tile" between the two reductions $\Delta(II) \to_\beta II(II) \to_\beta^* II$, of length 3, and the reduction $\Delta(II) \to_\beta \Delta I \to_\beta II$; such a tile cannot be decomposed in terms of the "square tiles" of pre-normal rewriting systems. However, what would replace event structures in such an approach is yet to be seen.

As far as point (ii) is concerned, one would expect the following: if $\mathcal{M}, \mathcal{M}'$ are two computational models for which it is well know that $\mathcal{M}$ can be encoded in $\mathcal{M}'$, and if both admit an interpretation in terms of event structures $\mathrm{Ev}(\mathcal{M}), \mathrm{Ev}(\mathcal{M}')$, then we should have $\mathrm{Ev}(\mathcal{M}) \hookrightarrow \mathrm{Ev}(\mathcal{M}')$. We already remarked that this is true in the case of translations between Lafont interaction nets. It is also possible to see that this holds in a number of other cases: for instance, Girard's standard encoding of the λ-calculus into proof nets yields a bisimilar embedding (when one gives to the λ-calculus and to proof nets the non-standard, normal residue structure mentioned above).

However, there are also surprises. For example, our encoding of the π-calculus in multiport interaction nets (Mazza, 2005) does not induce a bisimilar embedding. An encoding yielding a bisimilar embedding may be given by building on work by Ehrhard and Laurent (2007). One has to turn differential interaction nets into a multiport system (instead of multirule), by introducing two dual families of *multidereliction* and *multicodereliction* symbols; all symbols in these families have arity 1, and there is one symbol for each possible coarity. These symbols interact as in Fig. 14. Then, if one takes Ehrhard and Laurent's encoding, but reduces nets with the rules of Fig. 14 instead of the usual rules for differential interaction nets, one obtains a bisimilar embedding; the details of this are of course out of the scope of this paper.

Actually, the above mentioned encoding by Ehrhard and Laurent (2007) suffers itself from a similar problem: indeed, the π-calculus admits confusion, as shown by the CCS process $\nu(a,b,c)(\overline{a} \,|\, a.\overline{b} \,|\, b.(\overline{c} \,|\, c) \,|\, \overline{b})$, whose τ-transitions generate an event structure isomorphic to that of Fig. 7; then, by Corollary 3.10, no encoding in differential interaction nets can induce a bisimilar embedding. As a matter of fact, to prove the soundness of their encoding, Ehrhard and Laurent take into account only certain reductions of differential interaction nets. If one considers only these reductions and the homotopy classes they generate, then their encoding does yield a bisimilar embedding. The meaningfulness of this "pruning", and its relevance to concurrency, is still a matter of research.

A further unexpected situation is that the folklore encoding of Turing machines with arbitrary degree of non-determinism into Turing machines with degree of non-determinism 2 (simulate a branching of degree $n > 2$ with $n - 1$ successive branchings of degree 2) becomes problematic. In fact, the degree of non-determinism of a Turing machine corresponds exactly with our homonymous notion (*cf.* Sect. 3.3) when these are seen as (confusion-free) normal rewriting systems. Then, since the folklore simulation does not introduce divergence in any reasonable sense (it simply slows down computation by a constant factor), by Corollary 3.12 it cannot induce a bisimilar embedding.

We face here another intrinsic limitation, this time not of event structures, but of how bisimulations deal with non-determinism. In fact, in the folklore encoding, at some point the Turing machine of degree 2 must choose not to simulate a transition of the other machine, without having chosen which of the other transitions it will simulate. This behavior is "forbidden" by bisimulations, and not just for event structures—the same holds for bisimulations on labelled transition systems. In a sense, the current treatment of communication in concurrency theory implies that *the only way to say "no" to someone is to say "yes" to someone else.*

One may object that bisimulations were conceived to treat "open" systems, while Turing machines are "closed". But that brings to light another issue: if Turing machines are indeed closed systems, and all transitions are internal (*i.e.*, τ-transitions), then bisimulations are completely useless (all closed systems are weakly bisimilar!), which leaves us with a fundamental hole: how do we relate expressiveness of "closed" non-deterministic systems? We believe that this question should draw the attention of further research.

References

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

Beffara, E. and Maurel, F. (2006). Concurrent nets: A study of prefixing in process calculi. *Theoretical Computer Science*, 356(3):356–373.

Berry, G. (1979). *Modèles complètement adéquats et stables des λ-calculs typés.* Thèse de doctorat d'état, Université Paris VII.

Danos, V. and Regnier, L. (1989). The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203.

de Carvalho, D. (2007). *Sémantiques de la logique linéaire et temps de calcul.* Ph.d. thesis, Université de la Méditerranée.

de Falco, M. (2009). An explicit framework for interaction nets. In *Proceedings of RTA 2009*, LNCS.

Ehrhard, T. and Laurent, O. (2007). Interpreting a finitary pi-calculus in differential interaction nets. In *Proceedings of CONCUR 2007*, LNCS, pages 333–348.

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

Fernández, M. and Mackie, I. (1999). A calculus for interaction nets. In Nadathur, G., editor, *Proceedings of PPDP '99*, volume 1702 of *Lecture Notes in Computer Science*, pages 170–187. Springer.

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

Joyal, A., Nielsen, M., and Winskel, G. (1996). Bisimulation from open maps. *Information and Computation*, 127(2).

Khalil, L. (2003). *Généralisation des Réseaux d'Interaction avec* amb*, l'agent de McCarthy: propriétés et applications*. Ph.D. Thesis, École Normale Supérieure de Paris.

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

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

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

Mazza, D. (2005). Multiport interaction nets and concurrency. In Abadi, M. and de Alfaro, L., editors, *Proceedings of CONCUR 2005*, LNCS, pages 21–35. Springer.

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). `http://www.pps.jussieu.fr/~smimram`.

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

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

Tasson, C. (2009). *Sémantiques et syntaxes vectorielles de la logique linéaire*. Ph.D. thesis, Université Paris Diderot – Paris 7.

van Glabeek, R. J. and Goltz, U. (1989). Equivalence notions for concurrent systems and refinement of actions. In *Proceedings of MFCS 1989*, volume 379 of *Lecture Notes in Computer Science (LNCS)*. 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. (1980). *Events in Computation*. Ph.D. thesis, Dept. of Computer Science, University of Edinburgh, Edinburgh, Scotland.

Winskel, G. (1982). Event structure semantics of CCS and related languages. In Nielsen, M. and Schmidt, E. M., editors, *Proceedings of ICALP 1982*, volume 140 of *LNCS*, pages 561–576. Springer.

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

Appendix A. Proofs of Sect. 1

A.1. *Proofs of Sect. 1.2*

Proposition 1.3. A poset $(X, \leq)$ is coherent iff it has a least element, and is *locally coherent*, *i.e.*, for every $x, y, z \in X$, $x \uparrow y$, $x \uparrow z$, and $y \uparrow z$ implies that $\bigvee\{x, y, z\}$ exists.

Proof. The forward implication is obvious: the supremum of the empty set, which is a clique, is the least element, and local coherence is just coherence restricted to sets of cardinality 3. For the backward implication, let $u \subseteq X$ be a clique of cardinality n; we must prove that its supremum exists. We do this by induction on n, observing first that we may suppose $n \geq 2$, because if $n = 0$, we have $\bigvee u = \bot$, and if $n = 1$, the only element of u is its supremum. So let $u = u_0 \cup \{x, y\}$. Observe that all subsets of a clique are cliques; therefore, the induction hypothesis gives us the existence of $z = \bigvee u_0$, as well as the existence of $x' = \bigvee(u_0 \cup \{x\})$ and $y' = \bigvee(u_0 \cup \{y\})$. Now, since x' and y' are both upper bounds of u_0, by definition of supremum we have $z \leq x', y'$, which shows that $z \uparrow x$ and $z \uparrow y$. Since $x \uparrow y$ holds by hypothesis, we are in position to apply local coherence, and obtain that $w = \bigvee\{x, y, z\}$ exists. But this is the supremum of u: in fact, it clearly bounds u and, if w' is another upper bound of u, it is also an upper bound of u_0, $u_0 \cup \{x\}$ and $u_0 \cup \{y\}$, so w' is an upper bound of $\{x, y, z\}$, hence $w \leq w'$. □

Lemma A.1. Let $(X, \leq)$ be a poset, let $a \in \mathfrak{P}(X)$, and let $u \subseteq X$ be such that $\bigvee u = a$. Then, $a \in u$.

Proof. We prove the contrapositive statement: let $u \subseteq X$ with $a \notin u$, and suppose that $\bigvee u$ exists. Now either there exists $x \in u$ such that $a < x$, or for all $x \in u$, $x < a$. In the first case, a is obviously not the supremum of u. In the second case, if a_0 is the unique element covered by a, we still have, for all $x \in u$, $x \leq a_0$, which shows once again that a cannot be the supremum of u. □

Lemma A.2. If E is an event structure, $x \in \mathcal{C}(E)$ iff there exists a finite sequence of pairwise coherent events $a_1, \ldots, a_n$ such that $x = \bigcup_{i=1}^{n} \downarrow a_i$.

Proof. Obvious. □

Lemma 1.4. For every event structure $E = (|E|, \leq, \smile)$, we have $\mathfrak{P}(\mathcal{C}(E), \subseteq) = \{\downarrow a \mid a \in |E|\}$, *i.e.*, the principal ideals of $(|E|, \leq)$.

Proof. Let $x \in \mathcal{C}(E)$; by Lemma A.2, we have $x = \bigcup_{i=1}^{n} \downarrow a_i$, where $a_1, \ldots, a_n$ are the maximal elements of x. From this, it can be inferred that x covers exactly n configurations, namely $\lceil a_1 \rceil, \ldots, \lceil a_n \rceil$, so it is prime iff $n = 1$. □

We now give a few auxiliary definitions. Let $(X, \leq)$ be a poset. A *compact* element of $(X, \leq)$ is an element $d \in X$ such that, whenever $d \leq \bigvee u$ for some finite¶ $u \subseteq X$, we have that $d \leq x$ for some $x \in u$.

¶ Compact elements are ususally defined using suprema of *directed* sets, not finite sets. The definition we give here is adjusted to our purposes.

We say that a poset $(X, \leq)$ is a *meet semilattice* if every two elements x, y of X have an infimum, denoted by $x \wedge y$, and there is a least element, denoted by $\bot$; we say that it is *distributive* if it is bounded complete and if, for any three elements x, y, z, $\{x, y, z\}$ bounded implies $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. The following is an immediate consequence of Lemma A.1:

Lemma A.3. Let $(X, \leq)$ be a distributive meet semilattice, and let $a \in \mathfrak{P}(X)$. Then, a is compact.

Proof. Let u be a finite subset of X, and suppose $a \leq \bigvee u$. Consider the set $v = \{a \wedge x \mid x \in u\}$. This set is finite and bounded, so $\bigvee v$ exists by bounded completeness. Moreover, by distributivity we have $\bigvee v = a \wedge \bigvee u = a$. By Lemma A.1, we conclude that $a \in v$, so there exists $x \in u$ such that $a \wedge x = a$, which implies $a \leq x$. □

The following shows that Lemma A.3 applies to configuration posets:

Proposition A.4. A configuration poset is a distributive meet semilattice.

Proof. Let $(X, \leq)$ be a configuration poset. The least element is clearly $\bigvee \emptyset$, which exists because $\emptyset$ is a clique. Let now $x, y \in X$; since $\downarrow x$ and $\downarrow y$ are finite, $\mathfrak{p}(x)$ and $\mathfrak{p}(y)$ are finite too, and so is $\mathfrak{p}(x) \cap \mathfrak{p}(y)$. This set is bounded (by x and y), so by bounded completeness (which is implied by coherence) $z = \bigvee \mathfrak{p}(x) \cap \mathfrak{p}(y)$ exists. We contend that $z = x \wedge y$. First of all, observe that, whenever $a \in \mathfrak{p}(x) \cap \mathfrak{p}(y)$, we have $a \leq x, y$, hence $z \leq x, y$ by definition of supremum. Let now $w \leq x, y$; then, we obviously have $\mathfrak{p}(w) \subseteq \mathfrak{p}(x) \cap \mathfrak{p}(y)$, so $\bigvee \mathfrak{p}(w) \leq z$; but X is prime algebraic, so $\bigvee \mathfrak{p}(w) = w$. For what concerns distributivity, observe that, again by the fact that X is prime algebraic, for all bounded $x, y \in X$, $\mathfrak{p}(x \vee y) = \mathfrak{p}(x) \cup \mathfrak{p}(y)$; therefore, given any $x, y, z \in X$, we have $\mathfrak{p}(x \wedge (y \vee z)) = \mathfrak{p}(x) \cap (\mathfrak{p}(y) \cup \mathfrak{p}(z)) = (\mathfrak{p}(x) \cap \mathfrak{p}(y)) \cup (\mathfrak{p}(x) \cap \mathfrak{p}(z)) = \mathfrak{p}((x \wedge y) \vee (x \wedge z))$, so we conclude by applying once more the fact that X is prime algebraic. □

A.2. *Proofs of Sect. 1.3*

Lemma 1.7. Let E, E' be event structures, and let $\beta : E \to E'$ be an isomorphism. Then:

1 $\Phi(E) = (\mathcal{C}(E), \subseteq)$ is a configuration poset (as is $\Phi(E') = (\mathcal{C}(E'), \subseteq)$);
2 the function $\Phi(\beta) : \mathcal{C}(E) \to \mathcal{P}_{\text{fin}}(E')$ defined by $\Phi(\beta)(x) = \{\beta(a) \mid a \in x\}$, for all $x \in \mathcal{C}(E)$, is an isomorphism between $\Phi(E)$ and $\Phi(E')$.

Proof. Point 1 is an easy verification: infima and, when they exist, suprema are intersections and unions, respectively, and we use Lemmas A.2 and 1.4 to obtain that $\mathcal{C}(E)$ is prime algebraic and coherent. Point 2 is a trivial consequence of the fact that β is bijective and preserves causality and coherence in both directions. □

Lemma 1.8. Let $(X, \leq), (X', \leq')$ be configuration posets, and let $\phi : X \to X'$ be an isomorphism. Define $\Psi(X) = (\mathfrak{P}(X), \leq, \not\uparrow)$, where $\not\uparrow$ is the negation of compatibility, *i.e.*, $a \not\uparrow b$ if there is no $z \in X$ such that $a, b \leq z$. Then:

1 $\Psi(X)$ is an event structure (as is $\Psi(X')$, which can be defined similarly);

2 the restriction $\Psi(\phi)$ of ϕ to $\mathfrak{P}(X)$ is an isomorphism between $\Psi(X)$ and $\Psi(X')$.

Proof. Again, point 1 is an easy verification: given $a \in \mathfrak{P}(X)$, $\downarrow a$ is always finite because X is a configuration poset, and compatibility propagates downwards, *i.e.*, $a \uparrow b \leq c$ implies $a \uparrow c$. For point 2, observe that, since ϕ is an isomorphism, given $a \in X$, $\phi(a)$ is prime iff a is. Then, through $\Psi(\phi)$, $(\mathfrak{P}(X), \leq)$ and $(\mathfrak{P}(X'), \leq')$ are isomorphic subposets of X and X', respectively; the fact that $\Psi(\phi)$ is an isomorphism of event structures is a consequence of the fact that conflict is defined in terms of compatibility. □

Lemma 1.9. Let E be an event structure. Define $\eta_E : |E| \to \mathcal{C}(E)$ by $\eta_E(a) = \downarrow a$, for all $a \in |E|$. Then, η_E is an isomorphism between E and $\Psi(\Phi(E))$.

Proof. Let $E = (|E|, \leq, \smile)$. First of all, Lemma 1.4 shows that η_E is a bijection between $|E|$ and $|\Psi(\Phi(E))|$, as desired. Let now $a, b \in |E|$. Observe that $a \leq b$ iff $a \in \downarrow b$ iff $\downarrow a \subseteq \downarrow b$. Moreover, $a \smile b$ implies that there is no configuration containing both a and b, so $(\downarrow a) \not\uparrow (\downarrow b)$; conversely, $a \frown b$ implies that $(\downarrow a) \cup (\downarrow b)$ is a configuration, so $(\downarrow a) \uparrow (\downarrow b)$. This proves that η_E is indeed an isomorphism of event structures. □

Lemma 1.10. Let X be a configuration poset, let u be a finite clique of prime elements of X, and set $\varepsilon_X(u) = \bigvee u$. This defines an isomorphism ε_X between $\Phi(\Psi(X))$ and X.

Proof. First of all, note that ε_X is well defined, and its image is in X, because X is coherent. Then, we proceed by observing that $u \in \Phi(\Psi(X))$ iff u is a finite configuration of $\Psi(X)$, which is equivalent to saying that it is a downward-closed, finite clique of prime elements of X (by "clique" we mean that $a, b \in u$ implies there exists $x \in X$ such that $a, b \leq x$). We start by proving the injectivity of ε_X on $\Phi(\Psi(X))$. Suppose that $u, v \in \Phi(\Psi(X))$ are such that $\varepsilon_X(u) = \varepsilon_X(v)$, *i.e.*, $\bigvee u = \bigvee v$, and take $a \in u$. Since $a \leq \bigvee u = \bigvee v$, by Lemma A.3 we infer that $a \in v$, because v is downward-closed. By symmetry, we obtain $u = v$. Turning to surjectivity, let $x \in X$, and consider $\mathfrak{p}(x)$. The primes in $\mathfrak{p}(x)$ are all pairwise compatible (they are all bounded by x), and $\mathfrak{p}(x)$ is downward-closed by definition, so $\mathfrak{p}(x) \in \Phi(\Psi(X))$. But then $\varepsilon_X(\mathfrak{p}(x)) = \bigvee \mathfrak{p}(x) = x$, because X is a configuration poset. To see that ε_X is an isomorphism simply observe that, for all $u, v \in \Phi(\Psi(X))$, $u \subseteq v$ iff $\bigvee u \leq \bigvee v$, for which we use again Lemma A.3. □

To finish the proof of Theorem 1.6, the naturality of η and ε remains to be checked; this is left to reader.

Appendix B. Proofs of Sect. 2

As in Sect. 2.2, in what follows we fix a normal rewriting system $\mathcal{S}$; all objects, radicals, and reductions must be assumed to be in $\mathcal{S}$.

Lemma 2.7. Let r, s be two distinct radicals, and let f, g be reductions. Then:

1 $rf \sim sg$ implies that r and s are independent, and there exists a reduction h such that $f \sim s^r h$ and $g \sim r^s h$;

2 $frt_1 \sim gst_2$ implies that $t_1 = p^q$ and $t_2 = q^p$ for some independent radicals p, q, and there exists a reduction h such that $fr \sim hq$ and $gs \sim hp$.

Proof. We actually prove the following statements, by induction on $n > 0$:

1' $rf \sim_n sg$ and $r \neq s$ implies that r and s are independent, and there exists a reduction h and $m, k \in \mathbb{N}$ such that $f \sim_m s^r h$ and $g \sim_k r^s h$, with $m + k = n - 1$;

2' $frt_1 \sim_n gst_2$ implies that $t_1 = p^q$ and $t_2 = q^p$ for some independent radicals p, q, and there exists a reduction h and $m, k \in \mathbb{N}$ such that $fr \sim_m hq$ and $gs \sim_k hp$, with $m + k = n - 1$.

(Note that $n = 0$ is excluded by the fact that r and s are distinct). For point (1') we shall use the forward cube property; point (2') is the dual, and therefore uses the backward cube property. We shall only give the proof of point (1'), the other proof being similar.

If $n = 1$, the result is obvious, with $h = id$ and $m, n = 0$. Let now $n > 1$. Since $r \neq s$, the sequence of "tiles" used to derive $rf \sim sg$ must at some point involve r, *i.e.*, we have $n_0, n' \in \mathbb{N}$, with $n' > 0$, such that $rf \sim_{n_0} rt^r f' \sim_1 tr^t f' \sim_{n'} sg$, for some radical t independent from r. Now, if $t = s$, we are done (simply put $h = f'$, $m = n_0$, $k = n'$). Otherwise, since $n' < n$, we may apply the induction hypothesis and obtain a reduction h' and two integers m', k' such that $r^t f' \sim_{m'} s^t h'$ and $g \sim_{k'} t^s h'$, with $m' + k' = n' - 1$. Now, if we put $p = r^t$, $q = s^t$, the fact that $m' < n'$ allows us once again to apply the induction hypothesis, and infer the existence of a reduction h'' and two integers m'', k'' such that $f' \sim_{m''} q^p h''$ and $h' \sim_{k''} p^q h''$, with $m'' + k'' = m' - 1$. By the forward cube property, we have that r, s are independent, and we further have a radical $t' = t^{rs^r} = t^{sr^s}$. Therefore, if we put $h = t'h''$, composing the homotopies we get

$$f \sim_{n_0} t^r f' \sim_{m''} t^r q^p h'' \sim_1 s^r t' h'' = s^r h,$$

$$g \sim_{k'} t^s h' \sim_{k''} t^s p^q h'' \sim_1 r^s t' h'' = r^s h,$$

which, if we put $m = n_0 + m'' + 1$ and $k = k' + k'' + 1$, prove that $f \sim_m s^r h$ and $g \sim_k r^s h$, with $m + k = n_0 + k' + m'' + k'' + 2 = n_0 + k' + m' + 1 = n_0 + n' = n - 1$. □

Lemma 2.8. Let f, g be compatible reductions. Then:

1 if $\operatorname{src} f = \mu$ and $\operatorname{trg} f = \nu$, then $id_\mu^f = id_\nu$;

2 if $f = hk$, $f^g = (hk)^g = h^g k^{g^h}$;

3 $fg^f \sim gf^g$.

Proof. Point (1) is easily proved by induction on $\|f\|$.

For point (2), we start with the case in which $h = r$, where r is a radical, and we proceed by induction on $\|g\|$. The case $g = id$ follows immediately from the definitions; if $g = sg'$, we have, by applying the above definitions and the induction hypothesis, $(rk)^{sg'} = ((rk)^s)^{g'} = (r^s k^{s^r})^{g'} = (r^s)^{g'} (k^{s^r})^{(g')^{r^s}} = r^g k^{s^r (g')^{r^s}} = r^g k^{(sg')^r} = r^g k^{g^r}$.

The general case of point (2) is proved by induction on $\|h\|$. If $h = id$, we have, thanks to point (1), $(id\, k)^g = k^g = id^g k^{g^{id}}$; if $h = th'$, we apply the restricted result proved above and the induction hypothesis, to obtain $(th'k)^g = t^g (h'k)^{g^t} = t^g h'^{g^t} k^{(g^t)^{h'}} = h^g k^{g^h}$.

For what concerns point (3), we start again by proving it in case $f = r$, where r is a radical; we do this by induction on $\|g\|$. If $g = id$, we immediately have $rg^r = r = gr^g$. Otherwise, we have two cases: either $g = rh$, or $g = sh$ with $s \neq r$. In the first case, observe that, by definition $r^r = id$, so $r^g = r^{rh} = (r^r)^h = id^h = id$ by point (1); hence, we can write $rg^r = r(rh)^r = rh = g = gr^g$. In the second case, by observing that, thanks to point (1) of Lemma 2.7, r and s are independent, and by invoking the induction hypothesis, we have $rg^r = r(sh)^r = rs^r(h^{r^s}) \sim sr^s h^{r^s} \sim sh(r^s)^h = shr^{sh} = gr^g$.

We now prove point (3) in the general case, by induction on $\|f\|$. The case $f = id$ is a straightforward consequence of the definitions and of point (1). If $f = rh$, we have $fg^f = rhg^{rh} = rh(g^r)^h$, which by the induction hypothesis is equal to $rg^r h^{g^r}$; this, by applying the restricted version proved above, is equal to $gr^g h^{g^r}$; but this, thanks to point (2), is equal to $g(rh)^g = gf^g$. □

We now prove a few useful structural properties.

Lemma B.1. Let t be a radical, and let f, g be reductions such that $t \lesssim f, g$. Then, $f \sim g$ implies $f^t \sim g^t$.

Proof. It is enough to prove that $f \sim_1 g$ implies $f^t \sim_1 g^t$; the general result follows by induction on n from the hypothesis $f \sim_n g$. We therefore assume that $f = hrs^r k$ and $g = hsr^s k$ for some independent radicals r, s and some reductions h, k, and reason by induction on $\|h\|$. In the base case, we start by considering the situation in which $t = r$ (or $t = s$, which is perfectly symmetric). Then, we have $f^t = (ts^t k)^t = s^t k = s^t(t^s k)^{t^s} = (st^s k)^t = g^t$. Let now r, s, t be all pairwise distinct; then, point (1) of Lemma 2.7 assures us that they are also pairwise independent, so we have $(rs^r k)^t = r^t(s^r k)^{t^r} = r^t s^{rt^r} k^{t^{rs^r}}$. Now, we invite the reader to check that r, s, t and their residues with respect to each other form a "cube", and that we can write $r^t s^{rt^r} \sim_1 s^t r^{st^s}$; moreover, by the tiling axiom of Definition 2.2, we have $t^{rs^r} = t^{sr^s}$, so we can rewrite the last term above into $r^t s^{rt^r} k^{t^{sr^s}} \sim_1 s^t r^{st^s} k^{t^{sr^s}}$, which by the same computations as above is easily seen to be equal to $(sr^s k)^t$.

Let now $h = ph'$, where p is a radical. If $p = t$, we have $f^t = h' rs^r k \sim_1 h' sr^s k = g^t$; if $p \neq t$, using the induction hypothesis we have $f^t = (ph' rs^r k)^t = p^t(h' rs^r k)^{t^p} \sim_1 p^t(h' sr^s k)^{t^p} = g^t$. □

Another important structural result is the following, which is a direct consequence of the above lemma and states that reductions are "epic modulo homotopy":

Lemma B.2. For all reductions f, g, h, we have that $hf \sim hg$ implies $f \sim g$.

Proof. It is enough to prove the statement when h is a radical; the general case can once again be obtained from this by an easy induction on $\|h\|$. So let t be a radical, and let $tf \sim tg$; we obviously have $t \lesssim tf, tg$, so Lemma B.1 applies, giving $(tf)^t \sim (tg)^t$; but $(tf)^t = f$ and $(tg)^t = g$. □

In order to prove Lemma 2.9, we first show the following:

Lemma B.3. For all reductions f, g, $f \lesssim g$ implies $f^g = id$.

Proof. We first prove the result in case $f = r$, where r is a radical, reasoning by induction on $\|g\|$. Observe that $g \neq id$, so the base case is $g = r$, in which we have $r^r = id$ by definition. If $g = sg'$, observe that r^s is either an identity reduction (in case

$s = r$), or is well defined by point (1) of Lemma 2.7; in both cases, we have $r^s \lesssim g'$, and since $r^{sg'} = (r^s)^{g'}$, we may conclude by applying the induction hypothesis.

To prove the full result, we proceed once again by induction on $\|f\|$. If $f = id$, we conclude by point (1) of Lemma 2.8. If $f = rf'$, we have $f^g = (rf')^g = r^g(f')^{g^r}$ by point (2) Lemma 2.8; since $r \lesssim g$, by the restricted result proved above, this is equal to $(f')^{g^r}$. Now, observe that $\|f'\| < \|f\|$, so if we manage to prove $f' \lesssim g^r$, we may conclude by applying the induction hypothesis. We start by applying point (3) of Lemma 2.8, and write $g = gr^g \sim rg^r$; but $f \lesssim g$ implies that $g \sim rf'h$ for some reduction h; then, by Lemma B.2, we have $g^r \sim f'h$, which proves $f' \lesssim g^r$. □

Lemma 2.9. For all reductions f, g, h, we have

1 $f \lesssim gh$ iff $f^g \lesssim h$;
2 $f \sim gh$ implies $f^g \sim h$.

Proof. We start with point point (1), and consider the backward implication: $f^g \lesssim h$ means that, for some k, $f^g k \sim h$; now, using point (3) of Lemma 2.8, we have $fg^f k \sim gf^g k \sim gh$, which proves $f \lesssim gh$. We move on to the forward implication of point (1), and reason by induction on $\|h\|$. If $h = id$, then we conclude by Lemma B.3; if $h = th'$, by the induction hypothesis we obtain $f^{gt} \lesssim h'$. Then, for some reduction k, using point (3) of Lemma 2.8, we can write $h = th' \sim tf^{gt}k = t(f^g)^t k \sim f^g t^{f^g} k$, which proves that $f^g \lesssim h$, as desired.

Point (2) follows from point (1) by remarking that, for any f', f'' $f' \sim f''$ iff $f' \lesssim f''$ and $f'' \lesssim f'$. Then, thanks to point (1) above, $f \sim gh$ immediately gives us $f^g \lesssim h$; it also implies $g \lesssim f$, so by Lemma B.3 we have $g^f = id$. Now, from $gh \lesssim f$, we can write, for some reduction k and applying by point (3) of Lemma 2.8, $ghk \sim f = fg^f \sim gf^g$, which by Lemma B.2 gives $hk \sim f^g$, which means that $h \lesssim f^g$. □

The last important structural property we prove is a sort of stability result (in the sense of Berry) for residues, which is a consequence of the backward cube property, and is used in the proof of Lemma 2.15:

Lemma B.4 (Stability). Let f, g, h, k be reductions such that $fh \sim gk$, let t be a radical such that $\mathrm{src}(t) = \mathrm{trg}(fh) = \mathrm{trg}(gk)$, and suppose that there exist two radicals t_1, t_2 coinitial with h, k, respectively, such that $t_1^h = t_2^k = t$. Then, there exists a reduction f_0, two coinitial reductions h_0, k_0 and a coinitial radical t_0 such that $f \sim f_0 h_0$, $g \sim f_0 k_0$, $h_0 h \sim k_0 k$, and $t_1 = t_0^{h_0}$ and $t_2 = t_0^{k_0}$.

Proof. By induction on $n = \|f\| + \|h\| = \|g\| + \|k\|$. If $n = 0$, all reductions are equal to identities and the lemma vacuously holds. If $n > 0$, we start by considering the case in which one of h, k is an identity; so suppose, without loss of generality thanks to the symmetry, that $k = id$, which forces $t_2 = t$. It is then easy to verify that the lemma holds with $f_0 = f$, $h_0 = id$, $k_0 = h$ and $t_0 = t_1$. We may therefore suppose that $h = h'r$ and $k = k's$. If $r = s$ and h', k' are identities, then $t_1 = t_2$ and the lemma trivially holds. Hence, if $r = s$, we may suppose that $h' = h''r'$ and $k' = k''s'$. Now, if we still have $r' = s'$, then we may conclude by applying the induction hypothesis to $fh' \sim gk'$. Therefore, we may restrict to the case in which $h = h'r = h''r'r$, $k = k's = k''s's$, and either $r \neq s$, or $r' \neq s'$. In both cases, point (2) of Lemma 2.7 applies, giving us a reduction $\widetilde{h}$ and two independent radicals p, q such that $r = p^q$, $s = q^p$, and $fh' \sim \widetilde{h}q$, $gk' \sim \widetilde{h}p$. Now the reductions $qt_1^{h'} r^{t_1^{h'}} \sim pt_2^{k'} s^{t_2^{k'}}$ and the three radicals r, s, t form a structure to which we can apply the backward cube property, which gives us a radical t' such that $t_1^{h'} = t'^q$ and $t_2^{k'} = t'^p$. Then, we may apply the induction hypothesis to $fh' \sim \widetilde{h}q$ and $gk' \sim \widetilde{h}p$, which gives us six reductions $h_1, k_1, f_1, h_2, k_2, f_2$ and two radicals t_3, t_4 such that: $f \sim f_1h_1$, $\widetilde{h} \sim f_1k_1 \sim f_2h_2$, $g \sim f_2k_2$, $t_1 = t_3^{h_1}$, $t_2 = t_4^{k_2}$, and $t' = t_3^{k_1} = t_4^{h_2}$. This latter equation tells us that we can apply once more the induction hypothesis to $f_1k_1 \sim f_2h_2$; this gives us three reductions $f_0, \widetilde{h}_0, \widetilde{k}_0$ and a radical t_0 such that: $f_1 \sim f_0\widetilde{h}_0$, $f_2 \sim f_0\widetilde{k}_0$, and $t_3 = t_0^{\widetilde{h}_0}$, $t_4 = t_0^{\widetilde{k}_0}$. Composing all this, we can verify that the lemma is satisfied by f_0, $h_0 = \widetilde{h}_0h_1$, $k_0 = \widetilde{k}_0k_2$, and t_0. □

Lemma 2.14. $x \in \mathfrak{P}(\mathcal{H}(\mathcal{S}))$ iff x is essential.

Proof. For the forward implication, consider the contrapositive, and suppose that x is not essential. The first possibility is $x = [id]$, which is not prime, because it covers no other class. Otherwise, $x = [fr]$, and there exists $gs \sim f$ and a radical t such that $r = t^s$. Observe that $[gs], [gt] \leq [fr]$; now suppose $gs \sim gt$; by Lemma B.2 we have $s = t$, which is a contradicts the self-conflict axiom (s and t are independent); so x covers at least two distinct classes, namely $[gs]$ and $[gt]$, and is not prime. Conversely, if $x = [fr]$ is essential, by definition it covers exactly one class, namely $[f]$. □

Lemma 2.15. Let fr be a reduction, with r a radical. Then, there exists a unique essential homotopy class $[ht]$, with t a radical, such that $f \sim hg$ and $r = t^g$ for some reduction g.

Proof. We proceed by induction on $\|f\|$: we prove the statement for fr by supposing it to hold for all reductions $f'r'$ such that $\|f'\| < \|f\|$. There are two cases: either fr is essential, or it is not.

In the first case, induction is not needed: we may put $h = f$, $t = r$, and $g = id$ to meet the requirements. To prove uniqueness, let $h't'$ be an essential reduction such that there exists g' such that $f \sim h'g'$ and $r = t'^{g'}$. If $g' = g''s$, we have that $r = (t'^{g''})^s$, which contradicts the essentiality of fr. So we must have $g' = id$, and $h't' \sim fr$.

In the second case, by definition there exist a reduction f' and a radical s such that $f \sim f's$, and such that $r = p^s$ for some radical p independent from s. We may then apply the induction hypothesis to $f'p$, which gives us an essential reduction $h't'$ and a reduction g' such that $f' \sim h'g'$ and $p = t'^{g'}$. Then, setting $h = h'$, $t = t'$ and $g = g's$ satisfies the requirements. For what concerns uniqueness, suppose $h''t''$ is an essential reduction, with t'' a radical, such that there exists g'' such that $r = t''^{g''}$. Then, by Lemma B.4, there exist three reductions f_0, h_0, k_0 and a radical t_0 coinitial with h_0 such that $h' \sim f_0h_0$, $h'' \sim f_0k_0$, and $t' = t_0^{h_0}$, $t'' = t_0^{k_0}$. If both h_0 and k_0 are identities, then we immediately have $h'' = h'$ and $t'' = t'$, as desired; we shall see that all other cases lead to contradictions. In fact, let $h_0 = h_0'q$; we then have $t' = (t_0^{h_0'})^q$, in contradiction with the essentiality of $h't'$. We may then suppose $h_0 = id$ and $k_0 = k_0'q$; then, we have $t'' = (t_0^{k_0'})^q$, this time contradicting the essentiality of $h''t''$. □

Lemma 2.17. For every reduction f, $\sharp[f] \leq \|f\|!$.

Proof. We start by recalling that the group of permutations over $\{1, \ldots, n\}$, with $n > 0$, which we denote by $\mathfrak{S}_n$, may be presented by $n-1$ generators $\tau_1, \ldots, \tau_{n-1}$ satisfying the equations $\tau_i^2 = 1$ for all $1 \leq i \leq n-1$ and $\tau_i\tau_{i+1}\tau_i = \tau_{i+1}\tau_i\tau_{i+1}$ for all $1 \leq i \leq n-2$; indeed, the generators correspond to the elementary transpositions, *i.e.*, permutations exchanging exactly two consecutive elements and behaving as the identity on the others.

Now, observe that $[id] = \{id\}$, so we may assume $f \neq id$, and $\|f\| = n > 0$. Then, by the injectivity axiom, there are at most $n-1$ reductions g such that $f \sim_1 g$. In fact, these come from "exchanging" pairs of radicals in f by means of a tile, and injectivity imposes at most one tile for every pair. We write $\widetilde{f_i}$, with $i \in \{1, \ldots, n-1\}$, for the tile "exchanging" the i-th and $i+1$-th radical of f (if it exists), and write $f \sim_{\widetilde{f_i}} g$ when $f \sim_1 g$ by means of "applying" the tile $\widetilde{f_i}$. We then invite the reader to check that:

— $f \sim_{\widetilde{f_i}} g \sim_{\widetilde{g}_i} h$ implies $h = f$;
— $f \sim_{\widetilde{f_i}} g \sim_{\widetilde{g}_{i+1}} h \sim_{\widetilde{h}_i} k$ and $f \sim_{\widetilde{f}_{i+1}} g' \sim_{\widetilde{g}'_i} h' \sim_{\widetilde{h}'_{i+1}} k'$ implies $k' = k$.

From this, we may infer that the different g reachable from f through homotopy are at most as many as the elements of $\mathfrak{S}_n$, which are $n!$, as is well known. □

Lemma 2.19. Let μ be an object of a normal rewriting system, let $[ht] \in |\operatorname{Ev}(\mu)|$, and let $u \in \mathcal{C}(\operatorname{Ev}(\mu))$, with $\varepsilon(u) = [f]$. Then, u enables $[ht]$ iff there exists a radical r such that $[ht]$ is the unique homotopy class given by Lemma 2.15 applied to fr.

Proof. For the backward implication, simply observe that under these hypotheses we have $[fr] = [f] \vee [ht]$, which implies $\varepsilon^{-1}([fr]) = u \cup \{[ht]\}$, and $[ht] \notin u$, so u enables $[ht]$. For the forward implication, the hypothesis tells us that $[f] \vee [ht]$ exists. Assume, for the sake of contradiction, that $h \lesssim f$ does not hold, so $h = f'r'$ and $h^f = f_1r_1$ for some radicals r', r_1 and reductions f', f_1. If we apply Lemma 2.15 to $f'r'$, we obtain a unique $[h't']$ and a g' such that $f' \sim h'g'$ and $r' = t'^{g'}$. Consider now the reduction f'^f; we have $ff_1 \sim h'g'f'^f$ and $r_1 = t'^{g'f'^f}$, so $[h't']$ satisfies the conditions of Lemma 2.15 also for $ff_1r_1 = fh^f$; then, by uniqueness of $[h't']$ we have that $h't' \lesssim f$ does not hold, whereas obviously $h't' \lesssim ht$, which means that $[h't']$ is an event below $[ht]$ which is not in u, so u does not enable $[ht]$, contradicting the hypothesis. Then, we must have $h \lesssim f$, so that setting $g = f^h$ meets the conditions of Lemma 2.15 for $[ht]$ with respect to fr. □

Appendix C. Proofs of Sect. 3

C.1. *Proofs of Sect. 3.1*

Proposition 3.3. 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 $\iota'' \circ \iota'$. □

C.2. *Proofs of Sect. 3.2*

The following is a sort of binary reformulation of Lemma 2.19:

Lemma C.1. Let μ be an object of a normal rewriting system, and let $[fr], [gs] \in |\operatorname{Ev}(\mu)|$. Then, there exists $u \in \mathcal{C}(\operatorname{Ev}(\mu))$ enabling both $[fr]$ and $[gs]$ iff there exist a reduction h and two radicals r', s' coinitial in $\operatorname{trg}(h)$ such that $f, g \lesssim h$, $r' = r^{h^f}$ and $s' = s^{h^g}$.

Proof. For the backward implication, we invite the reader to check that $\varepsilon^{-1}([h])$ enables $[fr]$ and $[gs]$, as in Lemma 2.19. For the forward implication, if we set $\varepsilon(u) = [h]$, the existence of r', s' satisfying the requirements immediately follows from Lemma 2.19. □

Proposition 3.8. A normal rewriting system $\mathcal{S}$ is confusion-free iff, for every object μ of $\mathcal{S}$, $\operatorname{Ev}(\mu)$ is confusion-free.

Proof. We start by supposing that $\mathcal{S}$ is confusion-free. Let μ be an object of $\mathcal{S}$, and let $[fr], [gs], [ht] \in |\operatorname{Ev}(\mu)|$. Suppose $[fr] \# [gs]$ and $[gs] \# [ht]$. The first implies, by Lemma C.1, the existence of a reduction k_1 and two radicals r_1, s_1 coinitial in $\operatorname{trg}(k_1)$ such that $f, g \lesssim k_1$ and $r_1 = r^{k_1^f}$, $s_1 = s^{k_1^g}$. Now, since $fr(k_1^f)^r \sim k_1 r'$ and $gs(k_1^g)^s \sim k_1 s'$, the fact that $[fr]$ and $[gs]$ are in conflict implies that r and s are not independent, therefore not separated; hence, by confusion-freeness of $\mathcal{S}$, r and s must be contemporary. The uniqueness provided by Lemma 2.15 then forces $f \sim g$, r, s coinitial and $k_1 = fk_1'$ for some reduction k_1' (we can actually set $k_1 = gk_1'$, there is no loss of generality since $f \sim g$). By the essentiality of $[fr], [gs]$, we immediately have $\lceil [fr] \rceil = \lceil [gs] \rceil$.

A similar reasoning stems from $[gs] \# [ht]$, which leads us to conclude that $[gs] = [fs]$, $[ht] = [ft]$, and there is actually one configuration enabling all three events, namely $\varepsilon^{-1}([k])$ with $k = fk'$ for some reduction k'; additionally, we have three radicals r', s', t' which are the residues of r, s, t, respectively, by reduction of k', and the pairs r', s' and s', t' are *not* independent. These latter two statements imply that, whenever $r' \neq t'$, these are *not* separated, hence in simple conflict by confusion-freeness of $\mathcal{S}$, so in particular they are not independent, which shows that either $[fr] = [ht]$ or $[fr] \# [ht]$, and establishes the fact that $\operatorname{Ev}(\mu)$ is confusion-free.

Let us now consider the converse; let r, s be coinitial radicals of $\mathcal{S}$ which are neither separated nor in simple conflict (note that this latter implies $r \neq s$). This means that there exists t coinitial with r, s such that both pairs r, t and s, t are not independent, and that there exist a radical r_0 and a reduction f such that $r = r_0^f$, but $s = s_0^f$ holds for no radical coinitial with r_0. Let then $\mu = \operatorname{src}(h) = \operatorname{src}(r_0)$. We shall prove that $\operatorname{Ev}(\mu)$ is not confusion free. Lemma 2.15 applied to fs and ft gives us two unique events $[gs_0], [ht_0]$; observe now that $[r_0]$, $[gs_0]$, and $[ht_0]$ are all enabled by $[f]$, and that therefore the non-independence of r, t and s, t implies $[r_0] \# [ht_0]$ and $[ht_0] \# [gs_0]$. Now, if s and r are independent, we immediately have that t must be different from both, and that r_0 and gs_0 are compatible reductions (in fact, $r_0 f^{r_0} s^r \sim gs_0 g_1^{s_0} r^s$, where g_1 is the reduction given by Lemma 2.15 satisfying $f \sim gg_1$ and $s = s_0^{g_1}$), which makes the transitivity of $\overset{\#}{\sim}$ fail for $\operatorname{Ev}(\mu)$. On the other hand, if r and s are *not* independent, then we have

$[r_0]\,\#[gs_0]$, but since by hypothesis $g \neq id$, we have $\lceil[gs_0]\rceil \neq \emptyset$ while $\lceil[r_0]\rceil = \emptyset$, so we have once again an event structure which is not confusion free. $\square$

C.3. *Proofs of Sect. 3.3*

Lemma C.2. Let $E \overset{\iota}{\hookrightarrow} E'$, and let $a \in |E|$. Then, $a', a'' \in \iota(a)$ implies $a' \smile' a''$.

Proof. Let $\mathcal{B}$ be a bisimulation associated with ι, and suppose $a' \frown' a''$. Then, $u' = \downarrow a' \cup \downarrow a'' \in \mathcal{C}(E')$, so there must be $u \in \mathcal{C}(E)$ and a poset isomorphism ϕ such that $(u, \phi, u') \in \mathcal{B}$. Now obviously $a', a'' \in \mathrm{supp}_\iota(u')$; but $\phi \subseteq \iota$, which implies $(a, a'), (a, a'') \in \phi$, contradicting the fact that ϕ is (the graph of) a function. $\square$

Lemma C.3. Let E be an event structure with no confusion of type II, and let (a, b, c) be a confusion of type I in E. Then, $a \frown c$.

Proof. The hypotheses force $u = \lceil a \rceil = \lceil b \rceil = \lceil c \rceil$, so u enables both a and c; if we had $a \smile c$, we would have $a \,\#\, c$, contradicting the definition of type I confusion. $\square$

Lemma C.4. Let ι be an embedding of E into E', and let $a, b \in |E|$ such that $a \,\#\, b$. Then, there exist $a' \in \iota(a), b' \in \iota(b), b'_0 \in |E'|$ such that $a' \,\#'\, b'_0 \leq' b'$.

Proof. Let $\mathcal{B}$ be a bisimulation associated with ι, and let u be a configuration of E enabling both a and b. Then, there must be $u' \in \mathcal{C}(E')$ and a poset isomorphism ϕ such that $(u, \phi, u') \in \mathcal{B}$; moreover, since $u \xrightarrow{a}_\iota u \cup \{a\}$, we must also have u'_1, v'_1, v' such that $u' \Longrightarrow_\iota u'_1 \xrightarrow{a'}_\iota v'_1 \Longrightarrow_\iota v'$, with $(u', \phi[a \mapsto a'], v') \in \mathcal{B}$ and $(a, a') \in \iota$. Consider now u'_1; since there are no administrative transitions in E, we must have $(u, \phi, u'_1) \in \mathcal{B}$. Observe now that from u we also have a computational transition labelled by b, and E' must be able to simulate it from u'_1, *i.e.*, we must have u'_2, v'_2, v'_3 such that $u'_1 \Longrightarrow_\iota u'_2 \xrightarrow{b'}_\iota v'_2 \Longrightarrow_\iota v'_3$, with $b' \in \iota(b)$. Now, if neither the events used by the administrative transitions $u'_1 \Longrightarrow_\iota u'_2$ nor b' are in conflict with a', then we are contradicting the fact that $\mathcal{B}$ is a bisimulation (it would be possible to perform both a and b in E', while these are mutually exclusive in E); therefore, if b'_0 is the "first" event in conflict with a' occurring in these transitions, we have $a' \,\#\, b'_0$. Note that either $b'_0 = b'$, or b'_0 must be necessary for b' to occur, otherwise another transition involving b' would be possible, and we would still obtain a bisimulation associated with ι. $\square$

Proposition 3.9. 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 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 $u' \in \mathcal{C}(E')$ such that $(u, \phi, u') \in \mathcal{B}$, with ϕ a poset isomorphism. Moreover, by Lemma C.4, we have $a' \in \iota(a), b' \in \iota(b)$ and $b'_0 \in |E'|$ such that $a' \,\#'\, b'_0 \leq' b'$, and $u' \overset{a'}{\Longrightarrow}_\iota v'_1$, $u' \overset{b'}{\Longrightarrow}_\iota v'_2$, with b'_0 occurring during this second transition. Now, since $c \leq a$, we have $\phi(c) \leq' a'$; on the other hand, $c \not\leq b$, so $\phi(c) \not\leq' b'$, otherwise the posets $u \cup \{b\}$ and v'_2 would not be isomorphic. But this implies $\phi(c) \not\leq' b'_0$, 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. If we apply Lemma C.4 twice, we obtain $a' \in \iota(a), b', b'' \in \iota(b), c' \in \iota(c)$ and $a'_0, c'_0 \in |E'|$ such that $b' \#' a'_0 \leq' a'$ and $b'' \#' c'_0 \leq' c'$. By Lemma C.2, we have $b' \smile' b''$, so by Lemma 3.5 we have $b'_0, b''_0 \in |E'|$ such that $b'_0 \#' b''_0$ and $b'_0 \leq' b', b''_0 \leq' b''$. Now, the absence of confusion of type II forces $b'_0 \leq a'_0$ and $b''_0 \leq c'_0$, which implies $a' \smile' c'$. But Lemma C.3 tells us that $a \frown c$, so $a' \smile' c'$ would prevent the simulation by a', c' of the occurrence of both a, c; hence, there must be another event $c'' \in \iota(c)$ such that $b'_0 \leq' c''$ and such that $a' \frown' c''$. Of course we must also have $b' \smile' c''$, otherwise E' could perform both b and c; suppose that such conflict is not immediate; then, applying once more Lemma 3.5, we would have $b'_1, c''_1 \in |E'|$ such that $b'_1 \#' c''_1$, $b'_1 \leq' b', c''_1 \leq' c''$. Again by absence of type II confusion, we would have $b'_1 \leq' a'_0$, which implies $a' \smile' c''$, contradicting what we said above. Then, we have $a'_0 \#' b'_0 \#' c''$, with $a'_0 \frown' c''$ and $a'_0 \neq c''$ by injectivity of ι, so (a'_0, b'_0, c'') is a confusion of type I in E'. □

Appendix D. Proofs of Sect. 4

Lemma 4.17. Let Σ be a finite alphabet, and let $m = \max\{\mathrm{coar}(\alpha) \mid \alpha \in |\Sigma|\}$. Then, the normal rewriting system induced by any multiport interaction net system on Σ 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. Let $\mathcal{S}$ be a multiport interaction net system on a finite alphabet. An anticlique of cardinality $n > 1$ in $\mathrm{Ev}(\mathcal{S})$ is given by a net 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 2 cells, and that we have $a_i \cap a_j \neq \emptyset$ for all i, j. Then, the minimum coarity required by a cell $c \in C = \bigcup_{i=1}^{n} a_i$ is equal to to the number of sets a_i such that $c \in a_i$; we denote this number by $\flat c$. We shall prove a lower bound to $m = \max_{c \in C} \flat c$ in terms of n, which will give us the desired bound. Choose some $1 \leq i \leq n$, and let $a_i = \{c', c''\}$; for $1 \leq j \leq n$, 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 one of $p'' = 0$; then all sets intersect in c', so we have $m = \flat c' = n$. 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 m is obtained when n is equally split into three parts: $\frac{2n}{3} \leq m$. This lower bound is smaller than the previous $m = n$, so we may conclude that in all cases $n \leq \frac{3m}{2}$, as desired. □