Journal of Applied Logic 11 (2013) 52–62

Contents lists available at SciVerse ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

A logic of non-monotonic interactions

Giovanni Boniolo [a,b], Marcello D'Agostino [c,*], Mario Piazza [d], Gabriele Pulcini [d]

[a] Dipartimento di Scienze della Salute, Università di Milano, Italy
[b] Istituto Europeo di Oncologia (IEO), Milano, Italy
[c] Dipartimento di Economia e Management, Università di Ferrara, Italy
[d] Dipartimento di Filosofia, Università di Chieti, Italy

ARTICLE INFO

Article history:
Received 26 April 2012
Accepted 27 September 2012
Available online 28 September 2012

Keywords:
Zsyntax
Substructural logics
Non-associative sequent calculi
Non-monotonicity
Proof-nets
Molecular biology

ABSTRACT

In this paper, which is part of the Zsyntax project outlined in Boniolo et al. (2010) [2], we provide a proof-theoretical setting for the study of context-sensitive interactions by means of a non-monotonic conjunction operator. The resulting system is a non-associative variant of MLL_{pol} (the multiplicative polarised fragment of Linear Logic) in which the monotonicity of interactions, depending on the context, is governed by specific devices called *control sets*. Following the spirit of Linear Logic, the ordinary sequent calculus presentation is also framed into a theory of proof-nets and the set of sequential proofs is shown to be sound and complete with respect to the class of corresponding proof-nets. Some possible biochemical applications are also discussed.

1. Introduction

Jane has a strong preference for red wine over white and for white wine over beer. She also has a weak preference for fish over beef. If presented with a menu that contains *beef* (*bf*), *fish* (*fs*), *white wine* (*ww*) and *beer* (*br*), she will probably choose according to her best available choice in each category. Schematically:

$$fs \otimes ww \otimes bf \otimes br \vdash (fs \odot ww) \otimes bf \otimes br \tag{1}$$

where (i) $\otimes$ is a resource-sensitive conjunction, indicating that the conjoined items are simultaneously available for ordering, each operand representing a single consumable item of the given type (so that the number of times the name of an item-type occurs in the $\otimes$-expression reflects the number of consumable items of that type that are available),[1] (ii) "$\odot$" denotes a "meal composition" operator, and (iii) the relation symbol "$\vdash$" denotes the pseudo-consequence relation that obtains between states A and B when B is reachable from A by a (possibly empty) sequence of choosing acts (so that, for every A, it holds that $A \vdash A$ as a result of the empty sequence of choosing acts). Hence, in the above example we are assuming that exactly one item of each dish and drink is available. In the right-hand side, as a result of Jane's choosing act, the units of fish and white wine have been used up to fulfil Jane's order, while the others remain available for further consumption.

What happens if one unit of red wine (*rw*) is added to the menu? It may well be that Jane chooses to order beef and red wine despite her weak preference for fish over beef, because to her taste red wine (for which she has a strong preference) "binds better" to beef than to fish. That is, her choice in the new extended menu is represented by

* Corresponding author.
E-mail addresses: giovanni.boniolo@ieo.eu (G. Boniolo), dgm@unife.it (M. D'Agostino), mpiazza@unich.it (M. Piazza), gab.pulcini@gmail.com (G. Pulcini).

[1] Here we make the simplifying assumption that the order in which the menu items are presented is immaterial, while it is well-known that it may well affect one's choice. However the approach developed in the following sections is largely independent of this simplifying assumption.

1570-8683/$ – see front matter
http://dx.doi.org/10.1016/j.jal.2012.09.002

$$fs \otimes ww \otimes bf \otimes br \otimes rw \vdash (bf \odot rw) \otimes fs \otimes ww \otimes br. \tag{2}$$

On the other hand, given that Jane dislikes cider (*cd*), its addition to the original menu in (1), instead of red wine, would by no means alter her initial choice:

$$fs \otimes ww \otimes bf \otimes br \otimes cd \vdash (fs \odot ww) \otimes bf \otimes br \otimes cd \tag{3}$$

and the same happens if the menu is extended with any item (e.g., chicken, tea, etc.) other than red wine.

Now, observing (1), (3) and the like, the $\otimes$ operator appears to be *monotonic*, i.e., to satisfy the following condition:

$$\frac{A \vdash C}{A \otimes B \vdash C \otimes B} \tag{4}$$

where A, B, C are arbitrary expressions representing states of the process, constructed using the resource-sensitive operators $\otimes$ and $\odot$. However, observing (1) and (2), it turns out to be *non-monotonic* and violate (4).

If we insist that an operator $\otimes$ should be classified as "monotonic" only if it *always* satisfies (4), then we should conclude that our $\otimes$ operator is non-monotonic. However, this involves some loss of information. For instance, on the basis of Jane's preferences we may *know in advance* that she will always make the same order as in (2) when presented with any menu that contains red wine and beef, quite independent of whatever else is available, that is:

$$bf \otimes rw \otimes A \vdash (bf \odot rw) \otimes A \tag{5}$$

whatever A may be. Moreover we may also know, still on the basis of her preferences, that she will always order fish and white wine *unless* beef and red wine are both present in the menu. This kind of information may be schematically represented as follows:

$$fs \otimes ww \otimes A \vdash (fs \odot ww) \otimes A \quad \text{provided that} \quad rw \text{ and } bf \text{ are not in } A. \tag{6}$$

Observe that the kind of information displayed in (5) and (6) can be obtained and used without knowing all the details of Jane's preferences.

The main purpose of this paper is to envisage a formalism for adequately representing, without loss of information, processes that involve this kind of *context-sensitive interactions*, which prompt for a *controlled monotonicity* of the $\otimes$ operator. We maintain that this approach may turn out to be useful in a variety of applications. A prominent one is the representation of biochemical processes where this kind of controlled monotonicity allows for expressing the empirical regularities that are typically observed in the laboratory, which often involve context-sensitive, non-monotonic interactions between objects.

An example is given by the all-important mechanisms concerning concurrent enzyme inhibition. An enzyme inhibitor is a molecule that binds to enzymes and decreases their activity. Since blocking an enzyme's activity can, for example, kill a pathogen or correct a metabolic imbalance, many drugs are enzyme inhibitors. Let us indicate by E the enzyme and by S its substratum, that is the molecule that should bind to the enzyme. Then, in most contexts, we have that $E \otimes S \vdash E \odot S$. However, if the inhibitor I is present, we observe that $E \otimes S \otimes I \vdash (E \odot I) \otimes S$ and $E \otimes S \otimes I \nvdash (E \odot S) \otimes I$, violating (4). Since most biological "regularities" are strongly context-dependent, the standard monotonicity of $\otimes$ cannot hold in general. However, simply saying that it does not hold would hide the important information that certain reactions may be assumed to take place in virtually all contexts, *except for* some known ones in which another kind of reaction is observed. With respect to this class of applications, the present paper can be seen as a development of the *Z-Syntax* project outlined in [2].

Following a tradition started with the Curry–Howard correspondence we shall present our formalism as a *logical* one, in which formulas are interpreted as *types* of objects and proofs are interpreted as *processes*. Under this interpretation, as in the examples above, the relation $\vdash$ is informally explained as follows:

$A \vdash B$ ("B is reachable from A") iff there exists a process (possibly the null one) that,

starting from an aggregate of type A, yields an aggregate of type B. (7)

Such processes typically involve (context-sensitive) non-monotonic interactions of objects that are destroyed to generate new ones. So, this approach requires that once a formula has been used in a proof, it is no longer available for further processing unless a fresh copy of it is provided, that is, the classical (and intuitionistic) *contraction* rule:

$$\frac{A \otimes A \vdash B}{A \vdash B} \tag{Contraction}$$

is *not* valid. Given the general failure of contraction, the logic we are concerned with belongs to the family of *substructural logics* [17,13,4,12,15,1,16], but departs from it for the important fact that (4), which is valid in all known substructural logics, is *not* valid in general.

As customary, we can identify an expression $A_1 \otimes \cdots \otimes A_n$, when occurring on the left of $\vdash$, with the *multiset* $[A_1, \ldots, A_n]$ and often represent such multisets by simply listing their elements (in an arbitrary order). Then, the monotonicity of $\otimes$ can be expressed by

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \tag{8}$$

where the comma in the antecedent of a sequent expression represents multiset union. This is a valid inference rule in all known contraction-free logics (when $\otimes$ is interpreted as "multiplicative" conjunction), but in the present approach its unrestricted validity is dropped in favour of the following *controlled* version:

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \quad \text{provided that} \quad \forall \Lambda \in \mathfrak{S}, \Lambda \nsubseteq \Gamma \cup \Delta \tag{9}$$

where $\subseteq$ is multiset inclusion, $\cup$ is multiset union and $\mathfrak{S}$ is a set of multisets, called *control set*, representing the known multisets of formulas whose inclusion in the input multiset has the effect of preventing the derivation of A from Γ or the derivation of B from Δ.

The role played by the control set is somewhat analogous to the role played by the background knowledge in a typical defeasible rule of the kind investigated in the field of *Default Logic* [14]. From this point of view, our approach can be seen as a practical way of introducing a default mechanism to control the monotonicity of a resource-sensitive conjunction operator.[2]

It is important to notice that, in most practical applications, the negative information stored in the control set is not fixed once and for all, but is open to accommodate empirical results, to the effect that proofs are not definitive, but may loose their validity depending on changes in the composition of the control sets which are dictated by empirical research. In this perspective, suitable extensions of this kind of formalism can be useful to represent the dynamics of empirical knowledge especially in areas — like molecular biology and empirical economics — in which the massive and continuous growth of data is not (yet) matched by the emergence of lawlike generalisations.

An unrestricted version of (8), on the other hand, can be obtained by assuming that the contexts Γ and Δ *do not interact*, which can be expressed by enclosing them in brackets:

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{(\Gamma), (\Delta) \vdash A \otimes B}. \tag{10}$$

A straightforward consequence of the use of such unrestricted rules with separation of contexts is that the $\otimes$ operator is *not associative*, a fact that connects our formalism to the class of logical calculi with non-associative conjunction [7,11,6].

The idea of using a contraction-free logic as the primary inference device in modelling molecular biology has been recently explored in [3], where hybrid connectives are integrated into the proof-theoretical machinery of Intuitionistic Linear Logic to deal with constraints that may restrict, depending on the application, the validity of a linear proof. In connection with molecular biology, this approach allows for effectively representing temporal and stochastic constraints that yield an adequate encoding of the synchronous stochastic π calculus. While our approach shares the same heuristic idea — using some conservative extension of Linear Logic to model constrained transitions — it focuses on the constraints associated with context-sensitive transitions, which are accounted for by means of control sets rather than hybrid connectives. It appears that the two approaches explore complementary aspects of complex transitions — context-sensitivity on the one hand, and temporal and probabilistic constraints on the other — that may be interestingly integrated in some unifying framework. Moreover, following the spirit of Linear Logic, we provide a theory of proof-nets [9,10] for context-sensitive interactions. Proof-nets constitute a graph-theoretical representation of sequent derivations which ignore trivial differences between proofs (for instance, when rules are applied in a different order). What is important to emphasise is that our proof-net formalism deals with proper axioms representing background empirical knowledge, and allows for managing this extra-logical information by means of linear logic links. The advantage of dealing with these geometric objects consists in expressing the sophisticated machinery of controlled monotonicity in a compact way, so as to facilitate the study of applications outside the field of proof-theory. The brief discussion of enzymatic inhibition is just a simple example of such potential applications.

The paper is organised as follows. In Section 1 we introduce the calculus of Context-Sensitive Interactions (CSI). Then, we provide an easy example of deduction describing a generic case of concurrent enzyme inhibition. In Section 2, we describe the proof-net formalism and prove that it is complete with respect to the sequent calculus. (We stress, however, that proof-nets are conceptually independent of the sequent calculus used to construct them.) Finally, in Section 3, some research issues and further developments are briefly discussed.

2. Language and sequent calculus

Henceforth, we will refer to our Logic of Context-Sensitive Interactions with the abbreviation CSI.

---

[2] Of course, the non-monotonic behaviour of some biochemical reactions is not a new topic in formal biology. For instance, in [5] a system is considered in which proteins are subject to two kinds of transformations: *complexation* (proteins bind to each other) and *decomplexation* (two proteins unbind in order to bind again in a new way). This latter transformation implies a form non-monotonic causality since the number of links may decrease during a formal reaction. In a different way, our calculus takes into account the specific non-monotonicity related to concurrent inhibition phenomena which does not imply any kind of non-monotonic causality.

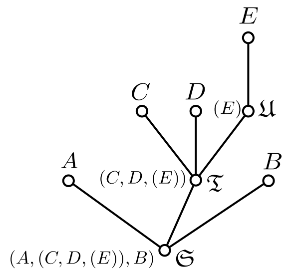

Fig. 1. The tree-structure of the context $\mathcal{C} = (A, (C, D, (E)_{\mathfrak{U}})_{\mathfrak{T}}, B)_{\mathfrak{S}}$.

Definition 1 *(Formulas).* CSI-formulas are inductively defined as follows:

- set of atoms $\mathscr{A} = \{a, b, c, \ldots\}$,
- bonding language $\mathscr{F}_{\odot} ::= \mathscr{A} \mid \mathscr{F}_{\odot} \odot \mathscr{F}_{\odot}$,
- logical language $\mathscr{F} ::= \mathscr{F}_{\odot} \mid \mathscr{F} \otimes \mathscr{F}$.

Technically speaking, the bonding operator $\odot$ is not a logical connective, since the CSI sequent calculus does not encompass any logical rule for introducing it. In fact, $\odot$ is a connective that expresses the existence of some extralogical interaction (e.g., a biochemical interaction) between objects of the types specified by the operands that yields a new object of some other type. Under this interpretation, a proposition such as, for instance, $(A \otimes B) \odot (C \otimes D)$ would be meaningless, and this is why the formulas of the bonding language cannot contain any operator other than $\odot$. In this sense the set of the *logically atomic* formulas is given by $\mathscr{F}_{\odot}$.

Definition 2 *(Control sets).* By *control set* we mean a finite set $\{\Gamma_1, \ldots, \Gamma_n\}$ of finite multisets of formulas such that, for all $1 \leqslant i \leqslant n$, $\Gamma_i \subset \mathscr{F}_{\odot}$. The empty set $\varnothing$ is a control set. Control sets are denoted by Gothic capital letters $\mathfrak{S}, \mathfrak{T}, \mathfrak{U}$ and so on.

Definition 3 *(Contexts, subcontexts, supports). Precontexts* are rooted trees recursively defined as follows:

- every element of $\mathscr{F}$ is a precontext,
- if $T_1, \ldots, T_n$ are precontexts, then $(T_1, \ldots, T_n)$ is a precontext.

Let us say that a precontext is *proper* when it is not an element of $\mathcal{F}$. A *CSI-context* $\mathcal{C}$ is an ordered pair $\langle T, f \rangle$ such that T is a proper precontext and f a function which assigns a control set to each node of T other than its leaves.

The *subcontexts* of a CSI-context $\mathcal{C} = \langle T, f \rangle$ are the CSI-contexts $\mathcal{C}' = \langle T', f' \rangle$ such that T' is a subtree of T and f' is the function obtained by restricting the domain of f to the non-leaf nodes of T'. We use the notation $\mathcal{C}[\mathcal{D}]$ to mean that $\mathcal{D}$ occurs as a subcontext of $\mathcal{C}$.

The *support* of $\mathcal{C}$, denoted by $|\mathcal{C}|$, is the multiset of the formulas labelling the leaves of $\mathcal{C}$.

Contexts are written by stressing the usual compact notation through nested parentheses. In order to represent the additional information afforded by the function f, right parentheses come indexed with a control set.

Remark 1. Loosely speaking, a CSI-context is nothing but a rooted tree having its leaves single-labelled with a formula from $\mathscr{F}$ and the other nodes double-labelled with both a precontext and a control set. It is worth noting that the unrestricted tree structure of sequents allows us to generalise the usual approach to logical non-associativity based on binary trees [7,6]. Moreover, as we will see later, the monotonicity of derivations is specifically regulated by the information injected into precontexts by the function f.

Definition 4 *(Immediately acting formulas).* An occurrence of a formula A is said to be *immediately acting* in a context $\mathcal{C}$ if it labels a leaf of $\mathcal{C}$ which is directly connected to the root. For any context $\mathcal{C}$, the multiset $\texttt{imac}(\mathcal{C})$ gathers all the immediately acting occurrences of formulas in $\mathcal{C}$.

Example 1. The tree structure of the context $\mathcal{C} = (A, (C, D, (E)_{\mathfrak{U}})_{\mathfrak{T}}, B)_{\mathfrak{S}}$ is displayed in Fig. 1. $(C, D, (E)_{\mathfrak{U}})_{\mathfrak{T}}$ and $(E)_{\mathfrak{U}}$ are the subcontexts of $\mathcal{C}$. In order to exemplify the notion of immediately acting formula, let's notice that $\texttt{imac}(\mathcal{C}) = [A, B]$.

For any $A \in \mathscr{F}$, $A^{\star}$ denotes the multiset of the logically atomic formulas occurring in A. The $\star$-operation can be straightforwardly extended to any multiset of formulas Γ as follows: $\Gamma^{\star} = \bigcup_{A \in \Gamma} A^{\star}$.

Table 1
Inference schemata.

Axiom:

$$\frac{}{(A)_{\varnothing} \vdash A}\ ax.$$

Cut-rules:

$$\frac{\mathcal{C}[(\mathcal{C}_1,\ldots,\mathcal{C}_n,A)_{\mathfrak{S}}] \vdash B \qquad \mathcal{D} \vdash A}{\mathcal{C}'[(\mathcal{C}_1,\ldots,\mathcal{C}_n,\mathcal{D})_{\mathfrak{S}}] \vdash B}\ surgical\ cut$$

$$\frac{\mathcal{C}[(\mathcal{C}_1,\ldots,\mathcal{C}_n,A)_{\mathfrak{S}}] \vdash B \qquad (\mathcal{D}_1,\ldots,\mathcal{D}_m)_{\mathfrak{T}} \vdash A}{\mathcal{C}'[(\mathcal{C}_1,\ldots,\mathcal{C}_n,\mathcal{D}_1,\ldots,\mathcal{D}_m)_{\mathfrak{S}\cup\mathfrak{T}}] \vdash B}\ deep\ cut^{\dagger}$$

(†) provided that $(\mathtt{imac}(\mathcal{C}_1\ldots,\mathcal{C}_n,\mathcal{D}_1,\ldots,\mathcal{D}_m))^{\star} \parallel \mathfrak{S}\cup\mathfrak{T}$

Structural rules:

$$\frac{\mathcal{C}[(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{H},\mathcal{K})_{\mathfrak{S}}] \vdash A}{\mathcal{C}'[(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{K},\mathcal{H})_{\mathfrak{S}}] \vdash A}\ \times change$$

Multiplicative conjunctions:

$$\frac{\mathcal{C}[(\mathcal{D}_1,\ldots,\mathcal{D}_n,A,B)_{\mathfrak{S}}] \vdash C}{\mathcal{C}'[(\mathcal{D}_1,\ldots,\mathcal{D}_n,A\otimes B)_{\mathfrak{S}}] \vdash C}\ \otimes_{\mathcal{L}} \qquad \frac{\mathcal{C} \vdash A \qquad \mathcal{D} \vdash B}{(\mathcal{C},\mathcal{D})_{\varnothing} \vdash A\otimes B}\ \otimes_{\mathcal{R}}$$

$$\frac{(\mathcal{C}_1,\ldots,\mathcal{C}_n)_{\mathfrak{S}} \vdash A \qquad (\mathcal{D}_1,\ldots,\mathcal{D}_m)_{\mathfrak{T}} \vdash B}{(\mathcal{C}_1,\ldots,\mathcal{C}_n,\mathcal{D}_1,\ldots,\mathcal{D}_m)_{\mathfrak{S}\cup\mathfrak{T}} \vdash A\otimes B}\ deep\text{-}\otimes_{\mathcal{R}}^{\ddagger}$$

(‡) provided that $(\mathtt{imac}(\mathcal{C}_1\ldots,C_n,\mathcal{D}_1,\ldots,\mathcal{D}_m))^{\star} \parallel \mathfrak{S}\cup\mathfrak{T}$

Table 2
Proper axioms.

Axioms of type σ:	**Axioms of type ρ:**
$\dfrac{}{(E\otimes F)_{\mathfrak{S}_i} \vdash E\odot F}\ \sigma_i$	$\dfrac{}{(E\odot F)_{\mathfrak{R}_i} \vdash E\otimes F}\ \rho_i$

Definition 5 *(Compatibility).* A multiset of formulas Δ is said to be *compatible* with a control set $\mathfrak{S} = \{\Gamma_1,\ldots,\Gamma_n\}$ — in symbols, $\Delta \parallel \mathfrak{S}$ — if, for all $\Gamma_i \in \mathfrak{S}$, $\Gamma_i \nsubseteq \Delta^{\star}$.

Definition 6 *(Monotonicity soundness).* A context $\mathcal{C}$ is said to be *monotonically sound* in case that, for any subcontext $\mathcal{D}$ of $\mathcal{C}$, $(\mathtt{imac}(\mathcal{D})) \parallel \mathfrak{T}$ where $\mathfrak{T}$ is the control set attached to the root of $\mathcal{D}$.

Definition 7 *(Sequents, related notions).* A *CSI-sequent* is an ordered pair $\langle \mathcal{C}, A\rangle$ such that $\mathcal{C}$ is a CSI context and $A \in \mathscr{F}$. Following the usual logical notation, a sequent $\langle \mathcal{C}, A\rangle$ will be henceforth written as $\mathcal{C} \vdash A$. Formulas in $|\mathcal{C}|$ are the *premises* of the sequent, whereas A is the *conclusion*.

Definition 8 *(Proofs).* A *CSI-proof* is a sequence of CSI-sequents such that each sequent is derivable from the sequents appearing earlier in the sequence by means of the rules displayed in Table 1 and of the proper axioms displayed in Table 2.

Remark 2. It is easy to check that CSI, stripped from the proper axioms and the rules of surgical cut and $\otimes_{\mathcal{R}}$, corresponds to MLL_{pol}, the multiplicative fragment of the polarised Linear Logic [10].

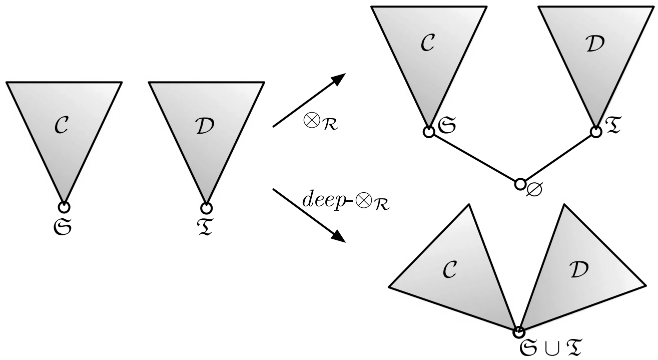

Fig. 2. Tensor combinatorics on trees.

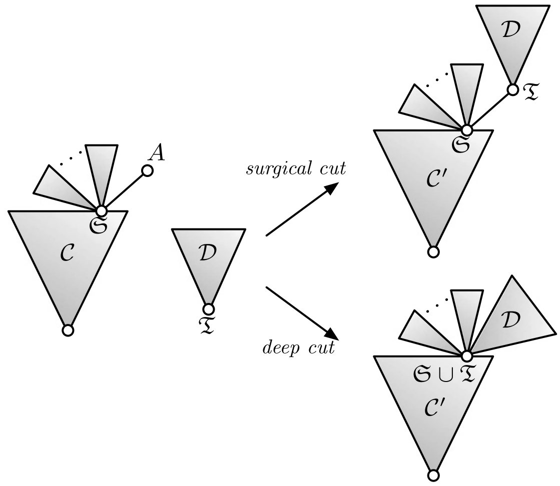

Fig. 3. Cut combinatorics on trees.

Figs. 2 and 3 illustrate the basic combinatorics on contexts respectively induced by the two right tensor rules ($\otimes_{\mathcal{R}}$ and *deep*-$\otimes_{\mathcal{R}}$) and the two cut rules (*surgical* and *deep*).

Whereas the $\otimes$-connective expresses a sort of proto-conjunction, the operator $\odot$ indicates that a proto-conjunction has been, so to speak, *activated* by virtue of certain (non-logical) conditions. Of course, the word "activated" changes its meaning according to the specific empirical context under focus. The need for distinguishing these two operators becomes apparent when one considers that $\otimes$ is only a multiset-constructor, that may combine occurrences of arbitrary formulas and has the same meaning as the comma in the antecedent of a sequent, while $\odot$ expresses the fact that two objects of specified types *interact* to generate a new object.

Together with the familiar inference schemata, the deductive apparatus of CSI comes equipped with a set of proper axioms encoding the information acquired in the specific empirical context we are concerned with. Unlike inference schemata which provide a general pattern of inference, proper axioms refer to specific types. In order to have a better grasp of CSI proper axioms, let's consider the specific empirical context afforded by biochemistry. Suppose that for a certain $i \in \mathbb{N}$ the axiom σ_i expresses the fact that two molecules of hydrogen bind so as to form the compound H_2:

$$\frac{}{(H \otimes H)_{\mathfrak{S}_i} \vdash H \odot H}\ \sigma_i.$$

The axiom σ_i does not mean that, for any type $A \in \mathscr{F}$, $A \otimes A \vdash A \odot A$: it just says that, for the *specific* type H, we have empirical evidence allowing us to state that $H \otimes H \vdash H \odot H$. Being our empirical information finite, CSI proper axioms are conceived as finite in number. Proper axioms are partitioned into two classes: the set of σ-axioms $\{\sigma_1, \ldots, \sigma_n\}$ and the set of ρ-axioms $\{\rho_1, \ldots, \rho_m\}$. Whereas σ-axioms regulate the "activation" of proto-conjunctions, ρ-axioms symmetrically manage the reverse process of "deactivation". As far as biochemical applications are concerned, reactions are standardly classified as synthesis, decomposition, single and double replacement. The CSI-calculus is able to formally reproduce these reactions only in a partial way in so far as σ- and ρ-axioms are meant to represent synthesis and decomposition, respectively. In order to recover the whole class of biochemical reactions, the set of CSI proper axioms must be completed by a third class of axioms, say ϵ, shaped as follows:

$$\frac{}{(E \otimes F)_{\mathfrak{E}_i} \vdash G \otimes H}\ \epsilon_i.$$

Notice that the logical results here proved for CSI can be easily extended so as to cover the enriched calculus including ϵ-axioms.

Example 2. A case of non-monotonic interaction is given by the extremely important mechanisms concerning the concurrent enzyme inhibition. Let us indicate by E the enzyme under consideration, by I the inhibitor and by S its substratum, that is the molecule that should bind the enzyme. In such a case, it is expected to have at disposal, among the set of biochemical axioms, a specific σ-axiom

$$\dfrac{}{(E \otimes S)_{\{[I],\ldots\}} \vdash E \odot S}\,\sigma_i$$

introducing the control set $\mathfrak{S}_i = \{[I], \ldots\}$. In order to provide an example of a CSI deduction, we report below a proof of the sequent $((E \otimes S)_{\{\{I\},\ldots\}}, (I)_\varnothing)_\varnothing \vdash (E \odot S) \otimes I$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(E)_\varnothing \vdash E}\,ax. \quad \dfrac{}{(S)_\varnothing \vdash S}\,ax.}{(E,S)_\varnothing \vdash E \otimes S}\,deep\text{-}\otimes_{\mathcal{R}} \quad \dfrac{}{(E \otimes S)_{\{[I],\ldots\}} \vdash E \odot S}\,\sigma_i}{(E,S)_{\{[I],\ldots\}} \vdash E \odot S}\,deep\text{-}cut \quad \dfrac{}{(I)_\varnothing \vdash I}\,ax.}{((E,S)_{\{[I],\ldots\}}, (I)_\varnothing)_\varnothing \vdash (E \odot S) \otimes I}\,\otimes_{\mathcal{R}}}{((E \otimes S)_{\{[I],\ldots\}}, (I)_\varnothing)_\varnothing \vdash (E \odot S) \otimes I}\,\otimes_{\mathcal{L}}$$

Let's notice that $(E \otimes S, I)_{\{[I],\ldots\}} \vdash (E \odot S) \otimes I$ is not a CSI-theorem because any derivation leading to this sequent would be unable to pass the checkpoint imposed by the control set $\mathfrak{S}_i$.

3. Proof-nets: the natural deduction for CSI

Let us start this section by remarking that there are essentially two ways of guaranteeing the monotonicity soundness along CSI derivations. One way consists in performing a check on contexts which parallels, critical step by critical step, the construction of proofs. This approach is followed when providing the CSI sequent calculus with the two metalogical conditions indicated in Table 1 with (†) and (‡). Another way consists in requiring the monotonicity soundness of the context to be checked just in the final sequent of the proof. In this way, logical correctness and monotonicity soundness turn out to be fully disentangled. Whereas, the first solution should be preferred in the traditional framework of sequent calculus, we will opt for the second alternative when dealing with CSI proof-nets.

The syntax of CSI proof-nets is defined by the set of links displayed in Fig. 4. Following the standard terminology, for any link ℓ, incident edges are called the *premises* of ℓ, whereas emergent edges are the *conclusions* of ℓ. At the level of sequent calculus, the distinction between input types and output types is clearly established by the very structure of the sequent. Passing from sequential proofs to deductions in term of nets, such a distinction needs to be recovered by attaching a polarity to formulas: negative formulas represent the inputs, whereas positive formulas are the outputs. A formal definition is provided below. Edges labelled with positive (resp. negative) formulas are called *positive* (resp. *negative*). Notice that, in the graph-theoretical setting we are proposing, control sets are taken into account by attaching them to positive edges.

Definition 9 *(Polarised formulas).* CSI-formulas can be partitioned into *positive* and *negative* as follows:

- *positive formulas*:
  - set of positive atoms $\mathscr{F}^+_\odot = \{A^+ \mid A \in \mathscr{F}_\odot\}$,
  - $\mathscr{F}^+ ::= \mathscr{F}^+_\odot \mid \mathscr{F}^+ \otimes \mathscr{F}^+$;
- *negative formulas*:
  - set of negative atoms $\mathscr{F}^-_\odot = \{A^- \mid A \in \mathscr{F}_\odot\}$,
  - $\mathscr{F}^- ::= \mathscr{F}^-_\odot \mid \mathscr{F}^- \otimes \mathscr{F}^-$.

Definition 10 *(Proof-structure).* A *proof-structure* is an oriented graph built over the set of links displayed in Fig. 4, which does not contain oriented cycles.

Proof-structures will be denoted by $\Pi, \Pi_0, \Pi_1, \ldots$. For any proof-structure Π, emergent negative (resp. positive) edges which are the premises of no link represent the *inputs* (resp. the *outputs*) of Π.

For the sake of clarity, proof-structures will be drawn by satisfying a sort of "gravity condition" according to which negative edges are always "lighter" than positive ones, that is, we will always draw positive (resp. negative) edges downwardly (resp. upwardly) oriented.

We indicate with CSI$^-$ and CSI$^=$ the two subsystems of CSI obtained, respectively, by removing the deep-cut rule from CSI and by removing the $\otimes_{\mathcal{R}}$ rule from CSI$^-$.

We will provide, at first, a theory of proof-nets limited to the fragment CSI$^-$. The difficulty of defining an homogeneous treatment for the whole class of CSI proof-nets lies in the fact that, in spite of their different combinatorial behaviour, the

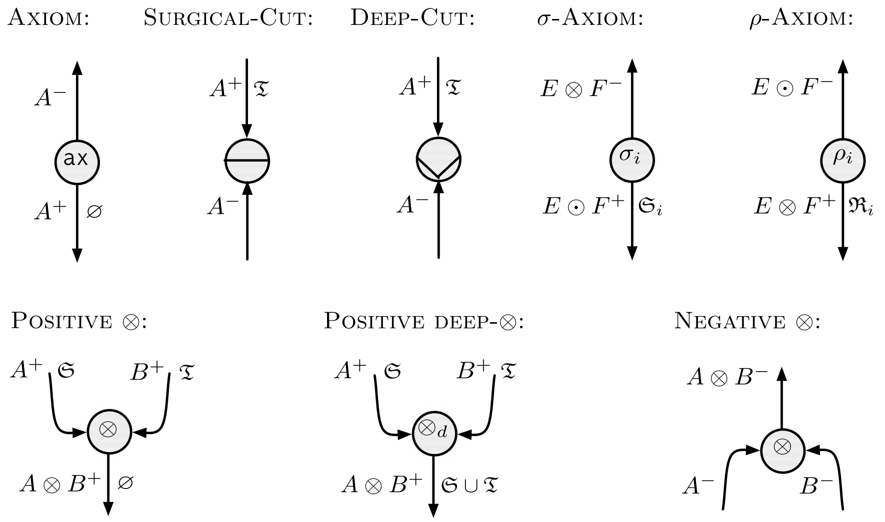

Fig. 4. The set of CSI links.

two cut links, surgical and deep, are identical from a syntactical point of view. Nonetheless, we will sketch in a final remark how to slightly modify our theory in order to encompass the deep-cut link.

Definition 11 *(Splitting links).* Given a proof-structure Π, a link $\ell \in \Pi$ is said to be *splitting* when its premises come from two disconnected proof-structures.

Definition 12 *(Proof-nets).* A CSI^- proof-structure Π is a *proof-net* only in case that:

- Π has exactly one output,
- each cut or positive $\otimes$-link in Π is splitting.

Remark 3. Notice that *only* positive $\otimes$-links are required to be splitting in a proof-net. This point appears to be clearly motivated when we recall that the positive $\otimes$-link corresponds to the $\otimes_{\mathcal{R}}$-rule in CSI sequent calculus whose combinatorial mechanism is just to connect two trees by merging the root of one with a leaf of the other. Thus, any CSI^- proof-net can be seen as a tree in which each node is either a splitting link or a $\mathsf{CSI}^=$ proof-net.

Definition 13 *(Cut-decomposition).* The *cut-decomposition* of a certain proof-net Π is given by the set of proof-nets $\Pi_0, \ldots, \Pi_n$ obtained from Π by disconnecting all the n cut links occurring in Π. We assume that a cut-decomposition comes with information enough for allowing an unambiguous recomposition of the former net.

Theorem 1 *(Context-extraction). The following instructions provide an algorithm for turning any* CSI^- *proof-net* Π *into a context* $\mathcal{C}_\Pi$.

- *Consider the cut-decomposition* $\Pi_0, \ldots, \Pi_n$ *of* Π. *For each* Π_i *occurring in the sequence compute* $\mathcal{C}_{\Pi_i}$ *as follows*:
  1. *for each positive deep-*$\otimes$ *link, contract its premises*;
  2. *contract all the edges directly connecting two negative* $\otimes$*-links*;
  3. *rewrite the resulting net through the contraction rule* $\Rrightarrow_{\mathsf{cnt}}$ (*see* Fig. 5) *until the net is no longer rewritable*;
  4. *"close" all the input edges of the net with a terminal node labelled with the type of the incident edge stripped from its polarity*;
  5. *remove all the polarised types labelling the edges*;
  6. *for each positive edge, shift the attached control set to the above node from which it emerges*;
  7. *contract the output edge.*
- *Compute the context* $\mathcal{C}_\Pi$ *through the recomposition of all the contexts* $\mathcal{C}_{\Pi_0}, \ldots, \mathcal{C}_{\Pi_n}$. *Such a recomposition is intended to be performed by stressing in the obvious way* (i) *the information gathered during the cut-decomposition and* (ii) *the instructions suggested by the sequent calculus.*

Proof. It is easy to verify that our procedure reduces any $\mathsf{CSI}^=$ proof-net Π to a fan-like tree $\mathcal{C}_\Pi$ where the inputs of Π label the leaves and the root is labelled by the union of all the control sets appearing in Π. This fact together with Remark 3 straightforwardly leads to the claim of our theorem. □

We write $\pi : \mathcal{C} \vdash A$ to denote a sequential proof π ending with the sequent $\mathcal{C} \vdash A$.

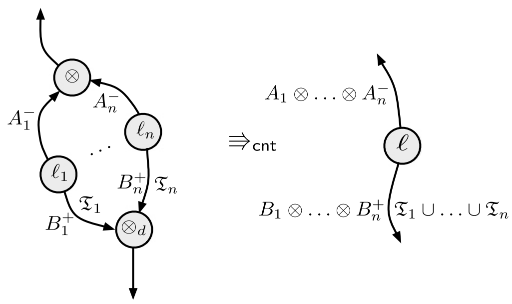

Fig. 5. Associativity of the negative ⊗-link.

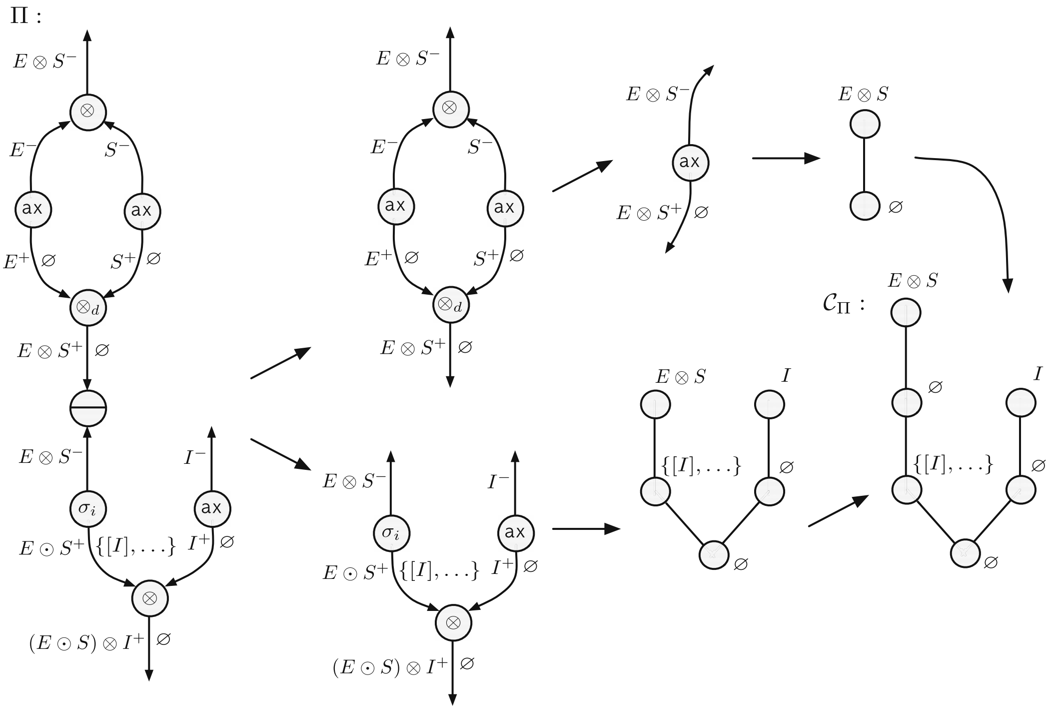

Fig. 6. Context-extraction procedure at work.

Theorem 2 *(Soundness). Any* CSI^- *sequential proof* $\pi : \mathcal{C} \vdash A$ *can be recursively turned into a monotonically sound proof-net* Π *having the output labelled with* A^+ *and such that* $\mathcal{C}_\Pi = \mathcal{C}$.

Proof. The proof can be easily achieved by induction on the length of the proof π. We consider here only the trickiest case in which the proof π is obtained by lengthening π' through a $\otimes_{\mathcal{L}}$-rule*:*

$$\dfrac{\begin{matrix}\pi' \\ \vdots \\ \mathcal{C}[(\mathcal{D}_1, \ldots, \mathcal{D}_n, A, B)_{\mathfrak{S}}] \vdash C\end{matrix}}{\mathcal{C}'[(\mathcal{D}_1, \ldots, \mathcal{D}_n, A \otimes B)_{\mathfrak{S}}] \vdash C} \otimes_{\mathcal{L}}.$$

In terms of nets, Π is obtained from Π' by connecting the two inputs labelled with A^- and B^- by means of a negative ⊗-link. Since Π' has exactly one output, the last inserted link will form a new cycle. By induction hypothesis we know that Π' is a proof-net such that $\mathcal{C}_{\Pi'} = \mathcal{C}[(\mathcal{D}_1, \ldots, \mathcal{D}_n, A, B)_{\mathfrak{S}}]$ so the new-formed cycle will not involve any cut or positive ⊗-link, otherwise the leaves A and B would not emerge, in $\mathcal{C}_{\Pi'}$, by the same node. Moreover, it is easy to check that $\mathcal{C}_\Pi$ differs from $\mathcal{C}_{\Pi'}$ just in making the two leaves A and B coincide. In this way, the monotonicity soundness turns out to be trivially preserved. □

Example 3. In Fig. 6, it is shown how to extract the context $\mathcal{C}_\Pi$ from the proof-net Π by stressing Theorem 1.

Theorem 3 *(Completeness). Any* CSI$^-$ *proof-net* Π *having the output edge labelled with* A^+ *can be turned into a monotonically sound sequential proof* $\pi : \mathcal{C}_\Pi \vdash A$.

Proof. The proof consists in providing a recursive procedure for "sequentialising" any monotonically sound proof-net Π yielding the output A^+ into a proof $\pi : \mathcal{C}_\Pi \vdash A$. According to Remark 3, the algorithm proceeds modularly by sequentialising, at first, each of the proof-nets belonging to the cut-decomposition of Π. The sequentialisation for a cut-free CSI proof-net Π_i is an easy task for $\mathcal{C}_{\Pi_i}$ immediately provides the sequential structure (quotiented under the associativity of the connectives $\otimes_\mathcal{L}$ and *deep*-$\otimes_\mathcal{R}$) of the proof π_i. Once we have at disposal the sequence of proofs $\pi_0, \ldots, \pi_n$, the entire derivation π can be "composed" by connecting the sub-proofs in the sequence according to the information gathered while performing the cut-decomposition. Notice that the monotonicity soundness does not raise any special difficulty, for the proof π is built over a context which is, by hypothesis, monotonically sound. □

Remark 4 (CSI *proof-nets*). The theory of proof-nets provided in this section for the fragment CSI$^-$ can be slightly modified in order to encompass the deep-cut link, and so the whole deductive apparatus of CSI. As far as logical correctness is concerned, it is sufficient to update Definition 12 through the following additional requirement:

- the oriented graph obtained from Π by reversing the orientation of all the positive edges is acyclic [10].

Indeed, CSI proof-nets without surgical cuts and positive $\otimes$-links geometrically behave as MLL$_{pol}$ proof-nets. The monotonicity soundness of a proof-net Π can be easily checked by turning it into a sequential proof. Note that, in this case, the sequentialisation procedure just consists in the trivial combination of the above-provided algorithm for CSI$^-$ proof-nets with the well-established procedure for MLL$_{pol}$ proof-nets [10].

4. Future work

This paper is a first attempt to address the problem of controlled monotonicity from a proof-theoretical and a graph-theoretical points of view. Its main aim is to illustrate the general control mechanism in a basic logical setting that can be extended in a variety of ways. A natural extension consists in increasing the expressivity of the underlying logical language to include suitable conditional and negation operators. Others crucially depend on the application context. For example, in biochemical applications more realistic logical models may require, as suggested in [2], to shift from standard formulas to *labelled formulas* as basic units of the contexts. By "labelled formulas" we mean expressions of the form $A : \alpha, B : \beta, C : \gamma$ where α, β, γ are labelling strings, specifying the values of suitable parameters related to any kind of additional information concerning the entities to which the formulas refer.[3] This idea has been recently investigated in some detail in [3] using Hybrid Linear Logic and an interesting line of research could be that of combining this approach with the one put forward in the present paper to deal with context-sensitive reactions.

On a more technical side, many proof-theoretical questions are opened, for example the search for a logical setting without non-associativity decorations so that the non-associative structure of processes can be taken into account by the tree-structure of proofs. On this view, the monotonicity soundness of derivation could be controlled through the well-established machinery of proof-nets and empires. A further step is to achieve a multiconclusion system not limited to the minimal sequents.

Finally, as concerns the cut elimination theorem, it is clear that it fails for sequent calculi with proper axioms. Nevertheless, the question of the eliminability of cut remains for a calculus with controlled monotonicity using logical axioms only, and will be addressed in a future work.

References

[1] J.-M. Andreoli, G. Pulcini, P. Ruet, Permutative logic, in: CSL, pp. 184–199.
[2] G. Boniolo, M. D'Agostino, P.D. Fiore, Zsyntax: A formal language for molecular biology with projected applications in text mining and biological prediction, PLoS ONE 5 (2010) e9511.
[3] K. Chauduri, J. Despeyroux, A Hybrid linear logic for constrained transition systems with applications to molecular biology, Technical Report, INRIA-HAL, 2010. Available at http://www-sop.inria.fr/members/Joelle.Despeyroux/papers/hyll_report.pdf.
[4] M. D'Agostino, D.M. Gabbay, K. Broda, Tableau methods for substructural logics, in: M. D'Agostino, D. Gabbay, R. Hähnle, J. Posegga (Eds.), Handbook of Tableaux Methods, Kluwer Academic Publishers, 1999, pp. 397–468.
[5] V. Danos, C. Laneve, Formal molecular biology, Theoret. Comput. Sci. 325 (2004) 69–110.
[6] P. de Groote, F. Lamarche, Classical non-associative Lambek calculus, Studia Logica 71 (2002) 355–388.
[7] K. Dosen, Sequent-systems and groupoid models I, Studia Logica 47 (1988) 353–385.
[8] D. Gabbay, Labelled Deductive Systems, Oxford University Press, 1996.
[9] J. Girard, Linear logic, Theoret. Comput. Sci. 50 (1987) 1–101.
[10] O. Laurent, Polarized proof-nets and lambda-μ-calculus, Theoret. Comput. Sci. 290 (2003) 161–188.
[11] R. Moot, M. Piazza, Linguistic applications of first order intuitionistic linear logic, J. Log. Lang. Inf. 10 (2001) 211–232.

[3] For the general paradigm of "Labelled Deduction", the reader is referred to [8].

[12] F. Paoli, Substructural Logics: A Primer, Kluwer Academic Publishers, Dordrecht, 2002.
[13] M. Piazza, M. Castellan, Quantales and structural rules, J. Logic Comput. 6 (1996) 709–724.
[14] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1980) 81–132.
[15] G. Restall, An Introduction to Substructural Logics, Routledge, London, 2000.
[16] G. Restall, Entry “Substructural Logics”, in: Stanford Encyclopedia of Philosophy, http://plato.stanford.edu/entries/logic-substructural/, 2008.
[17] P. Schroeder-Heister, K. Dôsen (Eds.), Substructural Logics, Clarendon Press, Oxford, 1993.