Desequentialization of Games and Experiments

Pierre Boudes

Institut de mathématiques de Luminy UMR 6206,
campus de Luminy case 907,
13288 Marseille cedex 9, France,
`boudes@iml.univ-mrs.fr`*

Abstract. We relate the dynamic semantics (games, dealing with interactions) and the static semantics (dealing with results of interactions) of linear logic with polarities, in the spirit of Timeless Games [BDER97].
We carefully analyze Laurent's polarized proof-nets (which for the game semantics is full and faithful [Lau03]) and we detail the precise correspondence between cut free proof-nets and innocent strategies, in a framework related to Böhm trees. We then introduce a notion of thick subtree that is used to define a desequentializing operation, forgetting time in games. This allows use to show a deep relation between plays in games and Girard's experiments on proof-nets. We then obtain our main result: desequentializing the game interpretation of a polarized proof-net yields its generic static interpretation.

Introduction

Denotational semantics interprets a *program* (a proof or a λ-term) as a structure representing all its possible interactions (via cut elimination or via β-reduction) with others programs. In static semantics only the *result* of the interaction is represented. In dynamic semantics (games) an interaction is fully represented by a sequence (play) of actions (moves) of the program (the Player) and the environment (the Opponent).

There are many references for game semantics. For an introduction, see [AM98]. In this paper, we use Hyland-Ong style polarized games [Lau04]. In such games, a move can justify itself by *pointing* to a preceding move. Laurent recently proved that polarized game semantics is full and faithful [Lau03], for the *proof-nets'* syntax of *linear logic with polarities*, LLpol. Proof-nets have been introduced together with linear logic [Gir87] as a more parallel syntax than sequent calculus. The static interpretation of a proof-net is the set of *results of experiments* on this proof-net.

The comparison between static and dynamic semantics is strongly motivated by Ehrhard's result ([Ehr96]) stating that the extensional collapse of sequential algorithms (a game model) is the hypercoherences semantics (a static semantics). In [Mel03], by introducing a suitable game semantics (extensional games), P.-A. Mellies gives a new proof of this result which better details the extensional content of games.

The *relational model* is the *generic* static semantics of linear logic, in the sense that the others are generally derived from this one by introducing new ingredients (like various coherence relations, see [Bou03]). In that very simple semantics, formulæ are sets and proofs are relations.

* This work was partly supported by the CNR-CNRS *Interaction and Complexity* project.

A *naive* approach to the comparison is to consider an operation D which maps a play (an interaction) to an element in a set of *results*, as in Figure 1. The difficulties are then: (i) to build a static semantics with sets of *results* (a natural candidate is the relational model); (ii) to turn D into a *logical* map, *i.e.* such that the diagram commutes for proofs.

$$\begin{tikzcd}[column sep=small] & \textit{Syntax} \arrow[dl] \arrow[dr] & \\ \textit{games} \arrow[rr, "D"] & & \textit{static semantics} \\ \text{play} \arrow[rr, mapsto] & & \text{result} \end{tikzcd}$$

Fig. 1. Projection D

This approach is successfully used in [BDER97], by introducing an ad hoc static semantics, the *(bi)-polarized pointed relational model*, and in [Bou04], by introducing an ad hoc game semantics, *bordered games* where plays explicitly carries results of interactions.

In this paper, our aim is to clarify the relation between syntax, static and dynamic semantics, without using ad hoc interpretations. We introduce a desequentialization D of justified plays, which for the source is Laurent's polarized games, and the target is the relational model. The desequentialization maps a play to the tree of its justification pointers.

By seeing the relational model (the target) as a game semantics without time (through the time forgetful projection D), we also pursue the goal of introducing static semantics to the operationally rich point of view on computation of game semantics.

In polarized games, proofs are interpreted as finite *innocent strategies*. Such strategies can be presented as finite *trees of Player's views*. Trees of P-views are particular instances of *abstract Böhm trees* ([Cur98,CH98]). When it comes from the interpretation of a proof, a tree of P-views can be thought of as an abstract presentation of the Böhm tree of a simply typed λ-term. (Pointers represent variables binding).

Since the polarized game semantics is full and faithful, trees of P-views are in a bijective correspondence with cut free polarized proof-nets of the same type. By analyzing the shape of cut free proof-nets, we detail this correspondence Ψ in a very direct way (comparatively to [Lau03]). To do so, we restrict ourselves to the additive free fragment, MELLpol, of LLpol. This minimize the complexity of the definition of proof-nets, at a low cost, since additives can (almost) be encoded in MELLpol.

Here is a sketch of the correspondence Ψ. Obviously, the Reader unfamiliar with proof-nets and games will find the definitions in the body of the paper.

A tree of P-views ϕ is a finite tree t_ϕ, together with two more datum: a naming f_ϕ of nodes by moves and a relation $\leftarrow_\phi$ specifying justification pointers between nodes. A MELLpol proof-net π is also a finite tree T_π (representing the nesting of !-boxes) but together with: a labeling R_π of nodes by *flat proof structures* (which are finite oriented graphs with pending edges) and a structure (S_π, B_π) relating flat proof structures to each others (this data can be considered as the frontiers of the !-boxes). When π is in normal form (cut free), the correspondence Ψ establishes a tree isomorphism between T_π and t_ϕ. Moreover the flat proof structures of π have a very particular shape: each flat proof structure consists of one *combined* positive connective and one *combined* negative connective together with edges connecting them or going through the frontiers of !-boxes. Through Ψ, moves of t_ϕ correspond to these *combined* connectives and pointers are just another way to draw the connecting edges.

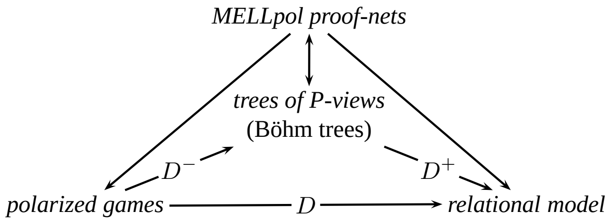


Fig. 2. Desequentialization

The core ingredient of the paper is the notion of *thick subtree*, a generalization of the usual notion of rooted subtree, which we present in Section 1, devoted to trees. We use thick subtrees both at the term level and at the type level.

Type level (Section 2). A MELLpol formula A can be thought as a tree (the *arena* in games). The desequentialization $D(p)$ of a play p in A is a particular thick subtree of A. And there is a (bijective) encoding of (classes up to isomorphism of) thick subtrees of A into the set of results of type A.

Term level (Section 3). Thick subtrees are used at the term level both to represent experiments in proof-nets and to express the *intrinsic dynamic* of plays of a strategy.

The desequentialization D factors into a negative part D^- followed by a positive part D^+ (Fig.2).

If p is a play in an innocent strategy ϕ then $D^-(p)$ is an even thick subtree s of t_ϕ. Conversely, any even thick subtree t of t_ϕ can be lifted into many plays in ϕ. This is just a matter of extending the tree order of s into a well-shaped total order. Intuitively, the tree order of s corresponds to the internal dynamic of the program (positive/Player) that one will find in any interaction between this program and an environment. The new part of the order is then provided by the environment (negative/Opponent) during a possible interaction. The thick subtree s *together with* the total order is called a multiplexed position in recent Curien's terminology (unpublished).

An experiment e in a proof-net π is just an even thick subtree s of T_π, together with an arbitrary *valuation* v of axioms. By extending the correspondence Ψ between proof-nets and trees of P-views to their thick subtrees, we show that the positive desequentialization D^+ of s (together with v) is the result of the experiment e. This proves that the desequentialization is a functor which maps a finite innocent strategy (a set of plays) to the static interpretation of the corresponding proof-net.

Figure 2 sketches the full picture we then obtain.

The notion of valuation (not represented in the figure) may interfere with the understanding of the full picture. In polarized games, an atom is represented by a single element (a move labeled by the name of the atom) while, in the relational model, it is a set of results, a *web*. Valuations of atoms fills this gap by associating webs to moves coming from atoms. For a first reading, one can restrict to the case of MELLpol without atoms, where valuations are useless.

1 Trees

(1.a) Trees. A **finite tree** t is a partial order $(I_t, \leq_t)$, where I_t is a finite set, the **indexing set**, having a least element, the **root** of the tree, and such that if $a \leq_t c$ and $b \leq_t c$ then $a \leq_t b$ or $b \leq_t a$. *In the sequel, trees will all be finite.* The cardinality $\sharp t$ of a tree $t = (I_t, \leq_t)$ is the cardinality of I_t. Elements of a tree are **nodes** and maximal elements are **leaves**. The associated precedence relation is denoted by $<^1_t$ (so $a <^1_t b$ means that $a <_t b$ and $a \leq_t c \leq_t b \implies c = a$ or $c = b$). The **tree!sons**, $\mathrm{son}(a)$, of a node a are the nodes b such that $a <^1_t b$. If a is a node of t then $t{\upharpoonright}a$ is the tree defined on $\{b \in I_t \mid a \leq b\}$ by $\leq_t$. A **labeling** in a tree is a function from I_t to a set of *labels*. An **ordered tree** is a tree together with, for each node a, a total order $<_a$ on the sons of a.

(1.b) Thick subtrees. A **tree morphism** $f : t \to t'$ is a function $f : I_t \to I_{t'}$ which maps the root of t to the root of t' and such that if $a <^1_t b$ then $f(a) <^1_{t'} f(b)$. An ordered tree morphism is a tree morphism preserving the order on sons.

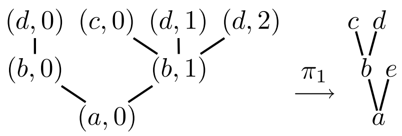


Fig. 3. An example of thick subtree

A **thick subtree** of a tree t is a tree s together with a tree morphism $f : s \to t$. The terminology is reminiscent from the fact that different nodes of s can be mapped to the same node of t. Typically, in Figure 3, the l.h.s. tree together with the first projection function, is a thick subtree of the r.h.s. tree. Observe that when f is injective, s is just a non empty rooted subtree of t (up to an explicit renaming of nodes). A **thick subtree morphism** between two thick subtrees (s, f) and (s', f') of a (same) ordered tree t is a tree morphism $g : s \to s'$ such that $f' = f \circ g$. If there exists an injective thick subtree morphism $g : (s, f) \to (s', f')$ then (s, f) is less than (s', f'). This defines an order on thick subtrees.

(1.c) Re-indexing. *In this paper, indexing sets are irrelevant:* we work on trees, sequences, ordered trees and thick subtrees up to isomorphism (for the respective notions of morphism). A concrete representation of ordered trees *up to re-indexing* is given by the grammar: $t := (t_1, \ldots, t_n)$ (an ordered tree is a tuple of ordered trees.) We also use the following convention for *canonically* localize an ordered tree. The indexing set I_t is a set of words of integers. The root of t is the empty word ε and if w is a node having n sons, then these sons are $w \cdot 1$, ..., $w \cdot n$, in this order. The tree order on I_t is then the prefix order. More generally, a set of words defines a tree (by prefix closure). We denote $w \cdot a$ the concatenation of a word w with an element a.

2 Types

Formulæ of multiplicative exponential linear logic with polarities (MELLpol) are given by:

$$N := ?X^\perp \mid \perp \mid N \parr N \mid ?P \qquad \text{(negative formulæ)}$$
$$P := !X \mid 1 \mid P \otimes P \mid !N \qquad \text{(positive formulæ)}$$

with the usual De Morgan laws for the orthogonal $(-)^\perp$ and where X is any element of a given set of atoms $\mathcal{V}$. Here, as in [Lau03], atoms (X, $X^\perp$) are not formulæ. This restriction is necessary for obtaining the faithfulness of the game semantics. For the same reason, MELLpol proof-nets (see Section 3) require the introduction of a *flat* ($\flat$) modality which does not belong to MELLpol and which can be thought of as a *why not* ($?$) modality, in semantics.

(2.a). The relational interpretation of a formula A is a countable set, denoted $|A|$ and called the **web** of A. The web of $A^\perp$ is always the same as the web of A. The web of 1 (and the web of $\perp$) is the singleton set $\{*\}$, (this set is intended to be *the* neutral element of the Cartesian product of sets, so $*$ shall be thought of as a notation for the empty tuple). The web of $A \otimes B$ (or of $A \parr B$) is $|A| \times |B|$. The web of $!A$ (or of $?A$) is the set of finite multisets of elements of $|A|$. For each atom $X \in \mathcal{V}$, an arbitrary enumerable set is chosen as web (both for X and its orthogonal). For convenience, we also set $|\flat A| = |?A|$.

To avoid some bureaucratic aspects, we will work on MELLpol up to associativity and neutrality of multiplicatives. In the relational model, this amounts to working up to associativity of the Cartesian product and neutrality of $\{*\}$.

2.1 Arenas and the desequentialization

An **arena** A is a labeled ordered finite forest together with a polarity: positive or negative. For the game semantics of MELLpol, we restrict ourselves to finite trees. The labeling function is denoted α_A. The labels of leaves are elements of $\mathcal{V} \cup \{*\}$ and the labels of others nodes are all equal to $*$.

The polarity of the arena is extended to moves by choosing the polarity of the arena for the root and by saying that two successive nodes have different polarities. This corresponds to the usual Player/Opponent polarity as follows: positive corresponds to Player and negative to Opponent.

Basically, in MELLpol, the arena associated with a formula A is the syntactic tree of this formula, up to associativity and neutrality of multiplicatives and where exponentials shift polarities.

(2.b) Arena of a formula. Let A be a formula. The arena of A is defined as follows. The polarity of the arena associated to A is the polarity of the formula. The tree of A and of $A^\perp$ are equal. The tree of 1 or of an atom X is the tree reduced to one node: $()$. If t is the tree of N then (t) is the tree of $!N$. If $(t_1, \ldots, t_p)$ is the tree of P and $(t'_1, \ldots, t'_q)$ is the tree of P' then the tree of $P \otimes P'$ is $(t_1, \ldots, t_p, t'_1, \ldots, t'_q)$. We adopt the canonical localization of ordered trees on arenas. The labels are chosen such that the label of a node coming from an atom X or $X^\perp$ is X and the labels of the others nodes are $*$. Conversely, every arena is the arena of a unique formula (up to re-indexing of labeled ordered trees). We further identify arenas and formulæ. Figure 4 shows the arena of $N_0 = ?!(?1 \parr ?X^\perp) \parr ?(!X \otimes !\perp) \parr ?1$ with the relevant part of the labeling.

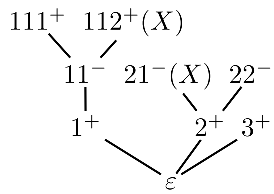


Fig. 4. The arena of N_0

(2.c). In the arena of a formula A, each sub-formula of A corresponds to a move. Two sub-formulæ can correspond to a same move a, but, for each move, their is a **maximal sub-formula** $F(a)$ of A corresponding to a. For instance, the first occurrence of $?1$ in N_0 corresponds to the move 11, but $F(11)$ is $?1 \parr ?X^\perp$.

(2.d). A **legal justified tree** (LJT, for short) on A is a finite tree $(I, \leq)$ together with a labeling function $f : I \to I_A$ and a *pointing* relation $\leftarrow$ such that: (i) $(I, \leftarrow^*, f)$ is a thick subtree of A; (ii) $\leq$ extends the order $\leftarrow^*$ (*i.e.* $\leftarrow^* \subseteq \leq$); and (iii) $<^1$ alternates between positives and negatives. We consider that polarities extend to elements of I by saying that the polarity of $a \in I$ is the polarity of $f(a)$. So the set I is the disjoint union of a set of negative nodes I^- and a set of positive nodes I^+. Observe that (ii) implies that $\leftarrow$ alternates between I^- and I^+ (as $<^1$). The notion of LJT encompasses the game notion of *legal play*. A **legal play** is a LJT where $(I \leq)$ is a total order.

(2.e). Observe that a LJT t has two tree structures $(I_t, \leq_t)$ and $(I_t, \leftarrow_t^*)$. We implicitly generalize some notions on trees (e.g. thick subtrees and prefixes) to LJT by considering that $(I_t, \leq_t)$ is *the tree* of the LJT t. If t is a LJT, a thick subtree $(I_{t'}, \leq_{t'}, g)$ of $(I_t, \leq_t)$ inherits a LJT structure $(\leftarrow_{t'}, f_{t'})$ from t by setting: $f_{t'} = f_t \circ g$ and if $a \leq_{t'} b$ and $g(a) \leftarrow_t g(b)$ then $a \leftarrow_{t'} b$.

Definition 1. *The **desequentialization** $D(t)$ of a LJT $t = (I, \leq, \leftarrow, f)$ is just the thick subtree $(I, \leftarrow^*)$, seen up to re-indexing (§1.c).*

(2.f). A thick subtree (t, f) is **equitable** when t has as much positive nodes as negative nodes. Observe that the thick subtree associated with an even-length, legal play is equitable. Conversely if a thick subtree (t, f) of an arena A is equitable then there exists a total order extending t into an even length legal play. (Proof by cases on the number of leaves and strict nodes of t of each polarity).

(2.g) Valuation (atoms). Let A be a formula. Let (t, f) be a thick subtree of A. A **valuation** v of (t, f) is a partial labeling of nodes of t given by the choice of an element x of the web of X, for each node a of t such that $\alpha_A(f(a)) = X$ (in that case $f(a)$ is a leaf of A and a is a leaf of t). When $f(a)$ is a leaf of A and $\alpha_A(f(a)) = *$ we set $v(a) = *$. So, each node a of t such that $f(a)$ is a leaf of A and no other is labeled.

A result of type A can be seen as a concrete representation of an equivalence class of valuated thick subtrees of A for the equivalence by re-indexing (§1.c). This representation commutes to the orthogonal. An element x of $|X|$ (resp. the element $*$ of $|1|$), is simply the unique thick subtree of the tree $()$, together with the valuation mapping its unique node to x (resp. $*$). Let $a = ([a_1^1, \ldots, a_{n_1}^1], \ldots, [a_1^k, \ldots, a_{n_k}^k])$ be an element of $|P|$ where P is $!N_1 \otimes \ldots \otimes !N_k$ (N_i can be an atom). For each $1 \leq i \leq k$ and for each $1 \leq j \leq n_i$, a_j^i is an element of the web of N_i, so a_j^i represents a valuated thick subtree (t_i^j, f_i^j, v_i^j) of N_i. The valuated thick subtree represented by a is then the tree $(t_1^1, \ldots, t_{n_k}^k)$ (seen *unordered*) together with: the function f mapping its root to the root of the arena of P and equal to $\sum f_i^j$ on the other nodes; and the valuation $\sum v_i^j$.

Proposition 2. *Let A be a formula. The set $\mathrm{VTST}(A)$ of valuated thick subtrees of A (up to re-indexing) is equal to the web of A.*

Direct, by induction on A. So, the desequentialization of a legal play on A together with a valuation is an element of the web of A.

3 Terms (proof-nets)

(3.a). A **flat proof structure** R is a finite directed graph, built using links of Figure 5, with at least one pending outgoing edges, called the conclusion edges. A label of an edge is either a formula, positive (P, Q) or negative (N, M) or an atom (X) or its orthogonal ($X^\perp$), or a *flat formula* $\flat F$, where F (or G) is either a positive formula or the orthogonal $X^\perp$ of an atom (so $F^\perp$ or $G^\perp$ is either a negative formula or an atom X). When connecting two links by an edge, the two labels of the edge must match.

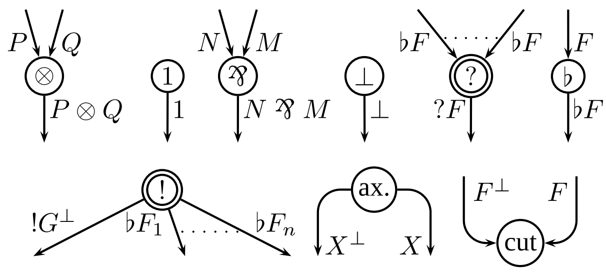


Fig. 5. Links of flat proof structures

Polarities of MELLpol formulæ extend to labels as follows. Atoms X are negative, their orthogonal are positive and flat formulæ are negative.

In a link, an outgoing edge is a *conclusion* and an incoming edge is a *premise*. In a !-link, the edge labeled $!G^\perp$ is the *front conclusion* and the others edges are the *auxiliary conclusions*. There is one !-link (resp. ?-link) for each natural number of auxiliary conclusions (resp. premises). For ?-links and !-link the ordering of incoming and outgoing edges is irrelevant (to remind it we draw them with a double line).

Observe that R is acyclic, because for each link, the label of each conclusion is strictly bigger than the label of each premise.

(3.b) Correctness criterion. Given a flat proof structure R, we define a new directed graph, , . A flat proof structure R is **correct** if: (i) the graph obtained (starting from R) by inversion of every edge with a positive label is acyclic; and (ii) either R contains no flat link ($\flat$) and has exactly one positive conclusion, or R contains exactly one flat link and has only negative conclusions.

(3.c). A **!-box** (R_L, B_L) for a !-link L is a correct flat proof structure R_L together with a one to one correspondence B_L between the conclusion edges of R_L and the conclusions of L such that: the conclusion labeled $!G^\perp$ of L (its front conclusion) is the image of a conclusion edge of R labeled by $G^\perp$; and the other edges' labels are preserved.

Definition 3. *A* ***proof-net*** π *is a finite tree* T *and three labeling functions* R, S, B *of nodes of* T *such that:*

- *for each node* n *of* T, $R(n)$ *is a correct flat proof structure and* $S(n)$ *is a one to one correspondence between the sons of* n *and the* !*-links of* $R(n)$*;*
- *if* n' *is a son of* n *in* T *then* $(R(n'), B(n'))$ *is a* !*-box for the* !*-link* $S(n)(n')$*.*

We do not make any requirement on the label $f(r)$ of the root r of π: this label is just here to ease the writing of the definition and it can be safely forgotten.

Observe that, if n is a node of a proof-net $\pi = (T, R, S, B)$ and if T_n is the maximal subtree T_n of t with root n, then $\pi_n = (T_n, R_{|T_n}, S_{|T_n}, B_{|T_n})$ is a proof-net.

The conclusions of a proof-net are the conclusions of its root's flat proof structure. A **MELLpol proof-net** is a proof-net where conclusions are not atoms or flat formulæ.

We do not describe the cut elimination procedure on proof-nets.

3.1 Relational semantics

(3.d). An **experiment** on a flat proof structure R is a labeling function e on edges of R such that:

- if a is a conclusion of an axiom link introducing an atom X, and if its other conclusion is b then $e(a) = e(b)$ and $e(a) \in |X|$;
- if a is the front conclusion of a !-link and $b_1, \ldots, b_n$ are the auxiliary conclusions, labeled respectively by $!N, \flat F_1, \ldots, \flat F_n$ then for each i, $e(b_i)$ is a multiset of points of $|F_i|$ and $e(a)$ is a multiset of points of $|N|$;
- if a is the conclusion of a 1-link or of a $\perp$-link then $e(a) = *$;
- if a_1 and a_2 are the first and the second premises and a is the conclusion of a $\otimes$-link or of a $\parr$-link then $e(a) = (e(a_1), e(a_2))$;
- if a is the premise and b is the conclusion of a $\flat$-link then $e(b) = [e(a)]$;
- if $a_1, \ldots, a_n$ are the premises and b is the conclusion of a ?-link then $e(a_1), \ldots, e(a_n)$ and $e(b)$ are multisets and $e(b) = e(a_1) + \ldots + e(a_n)$;
- if a is a premise of a cut link, and if its other premise is b then $e(a) = e(b)$.

Observe that $e(a)$ is always an element of the web of the label of the edge a.

An experiment on a flat proof structure can be considered as a choice of labels for axiom links and !-links which satisfies the constraint $e(a) = e(b)$ on cut links, when propagated by other links.

(3.e). If R has only negative conclusions and if e is an experiment on R then $r(e)$, the **result** of e, is the family $a \mapsto e(a)$ indexed by conclusions of R. This notion of result extends to any flat proof structure R' and to any experiment e' on R' by setting $r(e') = r(e)$ where e is an experiment on a flat proof structure R defined as follows. If R' has a positive conclusion we add below a $\flat$-link. Then, for each conclusion of type a flat-formula we add below a unary ?-link. The resulting proof structure is R and there is a unique extension of e' into an experiment of R which is e.

(3.f). Experiments on proof-nets and their results are defined inductively on π as follows. If the root of π is the flat proof structure R then an experiment e_π on π is an experiment e on R together with, for each proof-net π_v associated with a !-link v of R, a multiset $[e^1_{\pi_v}, \ldots, e^{k_v}_{\pi_v}]$ ($k_v \in \mathbb{N}$) of experiments on π_v which satisfies the following. If a is the front conclusion and $b_1,\ldots,b_n$ are the auxiliary conclusions of v and if, for each i, the result of $e^i_{\pi_v}$ is $(x_i, \nu_i^{1,v}, \ldots, \nu_i^{n,v})$ then $e(b_1) = \sum_{i=1}^{k_v} \nu_i^{1,v}, \ldots, e(b_n) = \sum_{i=1}^{k_v} \nu_i^{n,v}$ and $e(a) = [x_1, \ldots, x_{k_v}]$. The result $r(e_\pi)$ of e_π is the result of e.

Hence on a proof-net, an experiment consists of two choices: (i) a copying choice for !-boxes, inductively given by: taking one copy of the root of π and, for each !-link of the root, choosing an arbitrary finite number of copies of the proof-net above, then starting again for each of these proof-nets; (ii) a choice of labels for axioms links in each (copy of) flat proof structure which have been selected during the first choice. Once propagated, this choice have to obey to the only constraint of equality of labels on cut links.

Observe that the first choice (i) is just the choice of an arbitrary thick subtree of T_π and that there is no constraint on (i) and (ii) when there is no cut link.

(3.g). To summarize, in this paper, an **experiment on a cut-free proof-net** π is regarded as given by: a thick subtree s of T_π; together with, for each axiom link in s introducing an atom X, the choice of an element of the web of X. We call this last choice a **valuation of axioms**.

(3.h). The result of an experiment on a MELLpol proof-net π with only one conclusion N is an element of the web of N. The **relational interpretation of a proof-net** π is the set of results of experiments on π, for all possible experiments.

3.2 Cut free MELLpol proof-nets

In this section, we describe and simplify the structure of cut free proof-nets. We start by introducing two simplifications, there will be a third one.

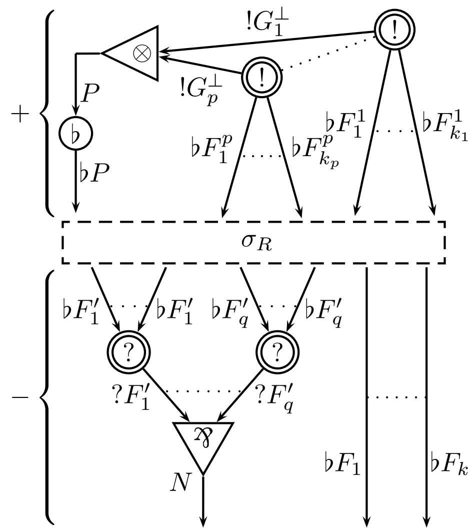


Fig. 6. What is in the box?

(3.i). First, we work with multiplicative connective up to neutrality and associativity. In flat proof structures there are trees of tensor links and 1-links with front conclusions of !-links above. We identify maximal such trees, called $\otimes$**-trees**, to links (drawn with a triangle). The same for trees of $\parr$-links and $\perp$-links with $?$-links above. We then speak about $\parr$**-trees**.

Second, we only consider MELLpol proof-nets with only one negative conclusion. If needed we can always transform a (cut free) proof-net into such a MELLpol proof-net by adding well chosen links to the flat proof structure at its root (the same way as in §3.e).

Observe that, if the conclusions of a cut free proof-net are known (before simplification) then we can recover this proof-net from its simplified version.

Let π be a (simplified) MELLpol cut free proof-net. We detail the shape of the flat proof structures contained in π. Let R be a flat proof structure of π. There are cases: one with exactly one axiom-link (Fig. 7) and one without axiom (Fig. 6).

Each flat proof structure occurring in π is either in a !-box or at the root of π. Hence R has conclusion edges labeled $\flat F_1, \ldots, \flat F_k$ and exactly one negative conclusion edge e^- labeled $G^\perp$ (the only conclusion if R is at the root of π).

According to the correctness criterion (§3.b), R has only one $\flat$-link $L_\flat$. Since this is the only link which has a positive premise and a negative conclusion, all the links with positive conclusions must be above $L_\flat$ in R. There are two cases: (i) either the premise of $L_\flat$ is the (positive) conclusion of an axiom link $L_{\text{ax.}}$ and there is no other link with some positive conclusions in R; (ii) or the premise of $L_\flat$ is the conclusion of a $\otimes$-tree $t_\otimes$ (possibly reduced to an edge or to a 1-link) with front conclusions of !-links above and there is no other link in R introducing a positive formula (we already found the unique $\flat$-link).

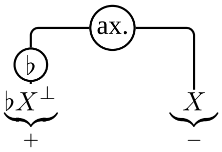


Fig. 7. Axiom's case

In the first case, the axiom also introduces a negative atom X which cannot be the premise of any link. So, R has no other link than $L_\flat$ and $L_{\text{ax.}}$ (in particular, R is a leaf of π).

Now the second case. The conclusion $G^{\perp}$ is a MELLpol formula N (if $G^{\perp}$ was a atom X then there will be an axiom introducing it in R). Above the edge e^{-}, labeled by N, there is a $⅋$-tree $t_{⅋}$ with $?$-links above. There is no other !-link introducing negative conclusions than the above mentioned. If there was one, this must be a $\flat$ formula (because we already found the $G^{\perp}$ conclusion) but flat formulæ are only introduced by $\flat$-links (there is only one, $L_\flat$) and !-links (another one than the above mentioned will also introduce a positive conclusion). Since the premises of the $?$-link are $\flat$ formulæ they have to be chosen among the conclusion of $L_\flat$ or of the !-links. The others conclusions of these last links which are not premises of $?$-links are the conclusions $\flat F_1, \ldots, \flat F_k$ of R. So there is a pairwise connection σ_R of: the conclusion of $t_{\otimes}$ and the auxiliary conclusions of the !-links with: the premises of the $?$-links and the conclusions of R different from e^{-}.

The third simplification we consider is the following. Observe that the !-links occurring in a flat proof structure R of a cut free proof-net π are totally ordered by mean of the ordering of premises of the unique $\otimes$-tree of R. As a consequence, rather than using S for matching !-boxes with !-links, we consider T_π as an ordered tree where the ordering of sons of a node n is the same as the ordering of the !-links of $R(n)$. (This cannot be done in a canonical way when there are cut links).

3.3 Game semantics

In polarized games, a MELLpol proof-net of conclusion A is interpreted as a *finite balanced total innocent strategy* in the arena A, called further a **MELLpol strategy**. In the sequel, we restrict ourselves to a negative type (the extension to positive types is easy).

(3.j). A **Player's view** (P-view for short) is a legal play s such that if $s_i <^1 s_j$ and s_j is an Opponent's move then $s_i \leftarrow s_j$ (the Opponent always points to the last move).

Traditionally a strategy is a set of legal plays satisfying some properties (e.g. prefix-closure, *determinism*). Composition of strategies is then defined *pointwise* on legal plays: two interacting legal plays are interleaved and, in the resulting sequence, the part on which the plays have interacted is hidden. We do not recall all the definitions of game semantics and polarized games. It is well known that, when a strategy is innocent, all its legal plays are determined by its P-views. This allows for an alternative description of innocent strategies which only uses P-views which we next relate to the traditional presentation (§3.k and Prop. 5).

Definition 4. *A* ***finite innocent strategy*** ϕ *in a negative arena* A *is an even prefix-closed set of P-views which is finite and* deterministic*: the longest common prefix of every two elements of the set is of even length. We further consider* ϕ *as the prefix tree of its P-views regarded, up to re-indexing, as a particular LJT* $(t_\phi, \leftarrow_\phi, f)$ *(in which every branch is a P-view).*

(3.k) Traditional presentation. The set of legal plays $P(\phi)$ associated with a finite innocent strategy ϕ on A is the smallest set such that: (i) the P-views of ϕ are in $P(\phi)$; (ii) if there is a visible legal play $s \cdot ab$ such that $s \in P(\phi)$ and $v^+(s \cdot ab)$ is a P-view of ϕ then $s \cdot ab \in P(\phi)$. Observe that $P(\phi)$ is an even-prefix closed set of visible even length legal plays which is not, in general, finite.

(3.l). When $(I, \leq, \leftarrow, f)$ is a LJT, we further consider the following relations: the *Player's pointers* $\leftarrow^+ = \leftarrow \cap (I^- \times I^+)$; the *Opponent's pointers* $\leftarrow^- = \leftarrow \cap (I^+ \times I^-)$; the *Player's precedence* $<^+ = <^1 \cap (I^- \times I^+)$; the *Opponent's precedence* $<^- = <^1 \cap (I^+ \times I^-)$; and the *Player's order*: $(<^+ \cup \leftarrow^-)^*$. The Player's order on I is still a tree because $\leftarrow^* \subseteq \leq$.

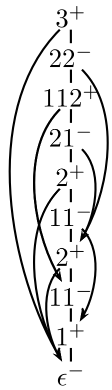


Fig. 8. p

(3.m). A LJT is **visible** when its Player's order contains the Player's pointers.

(3.n). The **negative desequentialization** $D^-((s, \leftarrow_s))$ of a visible legal play (more generally, of a visible LJT) is the LJT $(I_s, (<^+ \cup \leftarrow_s^-)^*, \leftarrow_s)$. Each branch of this tree is a P-view. The **view function** v^+ maps visible legal plays to P-views: a legal play s with last move a is mapped to the unique branch of the tree $D^-(s)$ with leaf a. The **positive desequentialization** D^+ is defined dually, by reversing the roles of Player and Opponent, on *co-visible* LJTs (the dual notion of visible LJTs). We will only use D^+ on the image of D^- where all LJTs are co-visible (because the Opponent always points to the last move). On LJTs the desequentialization D factors through D^- and D^+, moreover D and D^+ coincide on the image of D^-. So, for a visible legal play s,

$$D(s) = D^+(D^-(s)) = D(D^-(s)).$$

Figure 8 shows a legal play p in the formula N_0 (of Fig. 4) which is visible (but not co-visible). If the Player's move 112 was set to point to the second occurrence of 11 (from bottom to top) then the play will not be visible. Figure 9 shows the negative desequentialization of p (the Opponent's pointers, trivial, are omitted).

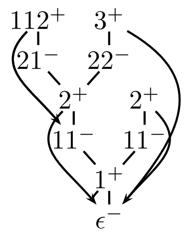
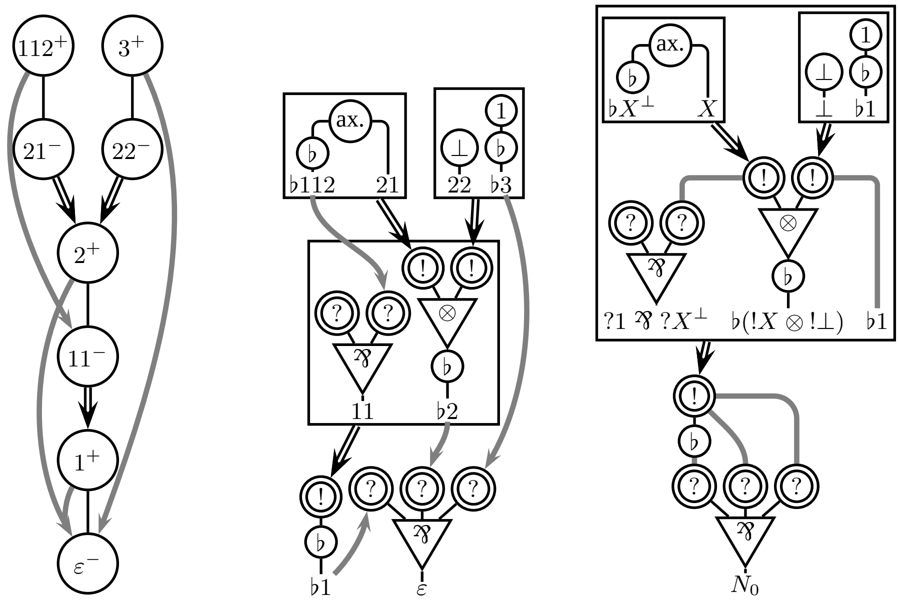


Fig. 9. $D^-(p)$

(3.o). If a LJT t is such that each Opponent's move has exactly one son then its **compact presentation** is itself but where, in the tree $(I_t, \leq_t)$, each pair of successive nodes $a <^+ b$ is regarded as one node (a, b). An **even thick subtree** t of a finite innocent strategy ϕ is a thick subtree of ϕ such that each Opponent's move has exactly one son (so, t is given by an arbitrary thick subtree of the compact presentation of ϕ). The set of even thick subtrees of ϕ is denoted $\mathrm{ETST}(\phi)$.

Proposition 5. *Let ϕ be a finite innocent strategy. If s is a legal play of ϕ (ie $s \in P(\phi)$) then $D^-(s)$ is an even thick subtree of ϕ. Conversely, if t (together with a tree morphism f) is an even thick subtree of ϕ then any total order $\leq_s$ on I_t which extends $\leq_t$ and preserves the Player's precedence of t (*i.e. *$<^+_s = <^+_t$) uniquely defines a legal play $(I_t, \leq_s, \leftarrow_t, f_t)$ which is an element of $P(\phi)$.*

Proof by induction on the cardinal of s (resp. t).



(a) Strategy (b) *Go-between* (c) Proof-net

Fig. 10. Example

(3.p). A finite innocent strategy ϕ in a negative arena A is: **total** when for each Player's move a of ϕ, the sons of a in ϕ are all the sons of a in A; **balanced** when for each Opponent's move a, if b is the son of a then a and b are mapped to the same label by α_A. A MELLpol strategy is a balanced total finite innocent strategy.

3.4 MELLpol proof-nets and strategies

Let N be a negative formula. A MELLpol strategy ϕ in N, together with N, uniquely defines a cut free MELLpol proof-net $\pi = \Psi_N^{-1}(\phi)$ as follows.

Figure 10 gives an example of the construction for the type N_0 (of Fig. 4).

The tree T_π is the tree of the compact presentation (§3.o) of ϕ, so the Opponent's pointers (§3.l) give the tree order of T_π. This relation will also gives the correspondence S_π between !-boxes and !-links (it is drawn with double lines in the figure). Totality of ϕ (§3.p) ensures that every !-link has an associated !-box. Each pair of moves (n^-, n^+) associated with a node n partially defines a flat proof structure $R(n)$ as follows. If $\alpha_N(n^-) = \alpha_N(n^+)$ is an atom X, $R(n)$ is a flat proof structure consisting of: one axiom introducing X with a ♭-link $n(\flat)$ connected to its conclusion $X^\perp$ (as in Fig.7). Otherwise $\alpha_N(n^-) = \alpha_N(n^+) = *$ (since ϕ is balanced) and $R(n)$ is a partial flat proof structure, similar to the one of Fig. 6. The maximal sub-formula $F(n^-)$ of N (§2.c) determines a ⅋-tree $n(⅋)$ with, for each of its premises $a_1, \ldots, a_q$, a ?-link W_i of conclusion a_i. The premises of these ?-links are yet unknown. The maximal sub-formula $F(n^+)$ of N defines a $\otimes$-tree $n(\otimes)$ with: for each of its premises $b_1, \ldots, b_p$, a

!-link with principal conclusion b_i; and a $\flat$-link $n(\flat)$ having the conclusion of the $n(\otimes)$. The auxiliary doors of the !-links and the permutation σ_{R_n} are not yet defined. In the partial proof-net we then obtain (Figure 10(b)), we still represent the Player's pointer: if n_1^+ points to n_2^- and if $F(n_1)$ occurs in $F(n_2)$ at place i then we draw an edge from $n_1(\flat)$ to the ith ?-link W_i above $n_2(\parr)$.

Next, we slice the pointers to reconstruct the missing edges of the flat proof structures, by working inductively on t_ϕ, from leaves to root. When $n(\flat)$ points to a ?-link W_i above $n(\parr)$ we draw an edge from the conclusion of $n(\flat)$ to W_i. Otherwise $n(\flat)$ is a conclusion of R_n. It is then *passed* as an auxiliary door to the associated !-link L of the flat proof structure below n: a conclusion edge is drawn from the actual source of the pointer, and this source is changed into L. It is passed again, until the source of the pointer is in the same flat proof structure as its target, a ?-link. We then draw an edge from the source to the target of the pointer. At the end of this process we obtain π.

Conversely, let πbe a cut free proof-net of conclusion N. We construct $\phi = \Psi_N(\pi)$ as follows.

We define a labeling M_π of the edges of the flat proof structures of π by moves of N. Intuitively this labeling is just a way to identify the occurrences of sub-formulæ of N to the places where they are created in the proof-net π. The (unique) conclusion edge of π is labeled by the empty word (the root of N). Going upward through a ?-link, a $\flat$-link or through a !-link and its associated !-box do not change labels. If L is a $\otimes$-tree or a $\parr$-tree and w is the label of its conclusion then its k premises are labeled $w \cdot 1, \dots,$ $w \cdot k$ (in that order).

We use M_π to associate to each node n of π an ordered pair made of one Opponent's move n^- and one Player's move n^+. For a flat proof structure containing one axiom (Fig. 7) this is respectively the move labeling the negative conclusion and the move labeling the positive conclusion of the axiom. For a flat proof structure without axiom (Fig. 6) this is respectively the move labeling the conclusion of the $\parr$-tree and the move labeling the conclusion of the $\otimes$-tree.

The tree T_π equipped with the labeling M_π will be the compact presentation of ϕ. The pointing relation $\leftarrow_\phi$ is not yet defined. We first set $n_1^+ \leftarrow_\phi n_2^-$ each time $n_1^- <_\phi^+ n_2^+$ (that is, each time $(n_1^-, n_1^+) <^1 (n_2^-, n_2^+)$ in the compact presentation of ϕ). Each flat proof structure $R_\pi(n)$ associated to a node n of π contains a unique flat link. We denote $\flat(n)$ its conclusion edge. There exists a unique chain $n_1 <_\pi^1 \dots <_\pi^1 n_k = n$ in π such that

$$B_\pi(n_2)(\dots B_\pi(n_k)(\flat(n))\dots)$$

is the premise of a ?-link of n_1. We set $n_1^- \leftarrow_\phi n^+$. We then obtain ϕ.

3.5 Experiments and strategies

The detailed correspondence between proof-net and strategies shows that a thick subtree of a cut free proof-net π of N can be regarded as an even thick subtree of the corresponding strategy $\phi = \Psi_N(\pi)$ (and conversely). But an experiment in π is just a thick subtree of T_π together with a valuation of axioms (§3.g). We now define valuations of axioms directly in games, to obtain a notion of experiment on MELLpol strategies.

The correspondence Ψ_N then extends into a one to one correspondence between experiments in π and experiments in $\phi = \Psi_N(\pi)$ and that allows to show that the result of a proof-net's experiment is the desequentialization of the corresponding strategy's experiment.

(3.q) Valuation of axioms (2). If (t, f) is an even thick subtree of ϕ (equivalently, an element of $P(\phi)$, §3.k) then a **valuation of axioms** in (t, f) is the choice, for each pair $a <_t^+ b$ such that $\alpha_N(a) = \alpha_N(b)$ is an atom X, of an element of the web of X. The set of valuated thick subtrees of ϕ is denoted $\mathrm{VETST}(\phi)$ and the set of valuated legal plays of $P(\phi)$ is denoted $V(P(\phi))$. Proposition 5 extends into a correspondence between these two sets.

Even thick subtrees of a strategy inherit the labeling by moves and pointers from the strategy (§2.e). We do the same for thick subtrees of proof-nets. If (t, g) is a thick subtree of T_π then we define three labeling functions R_t, S_t and B_t on t as follows. For each node n of t, $R_t(n)$ is a copy of the flat proof structure $R_\pi(g(n))$. If $n <_t^1 n'$ then $S_\pi(g(n))(g(n'))$ is a !-link L of $R_\pi(g(n))$ which has a corresponding copy L_c in $R_t(n)$. We set $S_t(n)(n') = L_c$. The one to one correspondence $B_t(n')$ between conclusions of $R_t(n')$ and conclusions of L_c is then simply a *copy* of $B_\pi(g(n'))$.

Lemma 6. *Let N be a negative MELLpol formula, Ψ_N extends into a one to one correspondence between the experiments on π and the experiments on $\phi = \Psi_N(\pi)$. Moreover, if e is an experiment on π then the valuated thick subtree $D^+(\Psi_N(e))$ (up to re-indexing) is the result of e.*

The extension of Ψ_N is straightforward, because the condition that the functions $S_\pi(n)$ are one to one in the definition of proof-nets (Def. 3) is not necessary to make Ψ_N work. But one needs to be careful for the slicing of Player's pointers when reconstructing a proof-net experiment e. If there are two pointers $n^- \leftarrow^+ n_1^+$, and $n^- \leftarrow^+ n_2^+$ such that n_1^+ and n_2^+ correspond to the same node in the strategy ϕ then, when going through the same partial flat proof structure R, these pointers define in R the same edge a (from a !-link to a ?-link or a conclusion) rather than two edges. This identification corresponds to a sum of multisets in the label $e(a)$. There is a labeling of pointers of $\Psi_N(e)$ which coincides with e on sources of pointers. It is then easy to check that $D^+(\Psi_N(e))$ is the result of e.

As an immediate consequence of this last Lemma we have that:

Proposition 7. *If π is a cut-free MELLpol proof-net of negative conclusion N, then the set $D^+(\mathrm{VETST}(\Psi_N(\pi))) = D(V(P(\Psi_N(\pi))))$ is the relational interpretation of π.*

This result proves that the desequentialization (together with valuations) defines a logical functor from polarized games to the relational model. By using techniques presented in [Bou03], D can be composed with a functor forgetting some results, to obtain a logical functor from polarized games to coherence spaces or to hypercoherences.

The results presented here extend to the full fragment of LLpol by shifting from trees to forests and by using the isomorphism $!(N \mathbin{\&} M) = !N \otimes !M$ and the distributivity laws of linear logic.

An important technical point concerns the co-visibility condition. Co-visibility of legal plays is required in original polarized games but, according to [Lau03], it does not play any role for proving fullness and faithfulness (as previously observed, P-views are always co-visible). Here it is mandatory to not require this condition on elements of $P(\phi)$, otherwise Prop. 5 will not be valid and D will be a lax functor. To our opinion, there is nothing deep here, just the usual flexibility of definitions in game semantics.

Since the desequentialization D provides good results for the game semantics of MELLpol, we hope that it can be applied to others game semantics, for instance for syntaxes with imperative features, in order to obtain static semantics of these syntaxes.

A corollary of Proposition 7 is that all the results of experiments are equitable. This property, coming from alternation of moves in plays, surely allows for narrowing the relational model to equitable results. But others properties of plays such as visibility are much harder to trap on the side of the relational model (at least we fail).

Another direction to look at is the faithfulness of the relational model. Factoring the interpretation through abstract Böhm trees by mean of thick subtrees allows for a more combinatoric approach of this long standing conjecture. But for the moment, we only have some limited results in that direction. Shifting to an untyped setting with real Böhm trees shall provides a faithfulness result (at least by using Böhm's theorem) which would be to be related with the typed setting.

References

[AM98] Samson Abramsky and Guy McCusker. Game semantics. In *Proceedings of Marktorberdorf of the 1997 Summer School*. Springer-Verlag, 1998. Lecture notes.

[BDER97] Patrick Baillot, Vincent Danos, Thomas Ehrhard, and Laurent Regnier. Timeless games. In Mogens Nielsen and Wolfgang Thomas, editors, *Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science*, pages 56–77, Aahrus, Denmark, August 1997. European Association for Computer Science Logic, Springer-Verlag.

[Bou03] Pierre Boudes. Non uniform hypercoherences. In Rick Blute and Peter Selinger, editors, *Proceedings of CTCS 2002, Electronic Notes in Theoretical Computer Science*, volume 69. Elsevier, 2003.

[Bou04] Pierre Boudes. Projecting games on hypercoherences. In *ICALP*, volume 3142 of *Lecture Notes in Computer Science*, pages 257–268. Springer, 2004.

[CH98] Pierre-Louis Curien and Hugo Herbelin. Computing with abstract Böhm trees. In Masahiko Sato and Yoshihito Toyama, editors, *Fuji International Symposium on Functional and Logic Programming (FLOPS '98), Kyoto, Japan, April 2-4, 1998*, pages 20–39. World Scientific, Singapore, 1998.

[Cur98] Pierre-Louis Curien. Abstract Böhm trees. *Mathematical Structures in Computer Science*, 8(6), 1998.

[Ehr96] Thomas Ehrhard. Projecting sequential algorithms on strongly stable functions. *Annals of Pure and Applied Logic*, 77, 1996.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Lau03] Olivier Laurent. Syntax vs. semantics: a polarized approach. Prépublication électronique PPS//03/04//n$^\circ$17 (pp), Laboratoire Preuves, Programmes et Systèmes, March 2003. Submitted for publication.

[Lau04] Olivier Laurent. Polarized games. *Annals of Pure and Applied Logic*, 130(1–3):79–123, December 2004.

[Mel03] Paul-André Melliès. Sequential algorithms and strongly stable functions. To appear in the special issue of TCS: Game Theory Meets Theoretical Computer Science, 2003.

Index