Quantitative Game Semantics for Linear Logic

Ugo Dal Lago[1,⋆] and Olivier Laurent[2,⋆⋆]

[1] Dipartimento di Scienze dell'Informazione
Università di Bologna
dallago@cs.unibo.it
[2] Preuves Programmes Systèmes
CNRS – Université Paris 7
Olivier.Laurent@pps.jussieu.fr

Abstract. We present a game-based semantic framework in which the time complexity of any IMELL proof can be read out of its interpretation. This gives a compositional view of the geometry of interaction framework introduced by the first author. In our model the time measure is given by means of slots, as introduced by Ghica in a recent paper. The cost associated to a strategy is polynomially related to the normalization time of the interpreted proof, in the style of a complexity-theoretical full abstraction result.

1 Introduction

Implicit computational complexity (ICC) is a very active research area lying at the intersection between mathematical logic, computational complexity and programming language theory. In the last years, a myriad of systems derived from mathematical logic (often through the Curry-Howard correspondence) and characterizing complexity classes (e.g. polynomial time, polynomial space or logarithmic space) has been proposed.

The techniques used to analyze ICC systems are usually ad-hoc and cannot be easily generalized to other (even similar) systems. Moreover, checking whether extending an existing ICC system with new constructs or new rules would break the correspondence with a given complexity class is usually not easy: soundness must be (re)proved from scratch. Take, for example, the case of subsystems of Girard's linear logic capturing complexity classes: there are at least three distinct subsystems of linear logic corresponding to polynomial time, namely bounded linear logic, light linear logic and soft linear logic. All of them can be obtained by properly restricting the rules governing the exponential connectives. Even if they have *not* been introduced independently, correspondence with polynomial time had to be reproved thrice. We need to understand *why* certain restrictions on the usual comonoidal exponential discipline in linear logic lead to characterizations of certain complexity classes.

⋆ Partially supported by the Marie Curie EIF grant "ASFIX" and by FIRB grant RBIN04M8S8, "Intern. Inst. for Applicable Math.".
⋆⋆ Partially supported by the French ANR "NO-CoST" project (JC05_43380).

This is the typical situation where semantics can be useful. And, indeed, some proposals for semantic frameworks into which some existing ICC systems can be interpreted have already appeared in the literature. Moreover, there are some proposals for semantic models in which the interpretation "reveals" quantitative, intensional, properties of proofs and programs. One of them [5] is due to the first author and is based on context semantics. There, the complexity of a proof is obtained by a global analysis of its interpretation as a set of paths.

In this paper, we show that the above mentioned context semantics can be put into a more interactive form by defining a game model for multiplicative and exponential linear logic and showing a quantitative correspondence between the interpretation of any proof and the time needed to normalize the proof itself. This correspondence can be thought of as a complexity-theoretic full-abstraction result, where proofs with the same complexity (rather than observationally equivalent proofs) are equated in the semantics.

Context semantics is a model of Girard's geometry of interaction. As a consequence, turning it into an AJM game model should not be difficult (at least in principle), due to the well-known strong correspondence between the two frameworks (see [3], for example). But there are at least two problems: first of all, the context semantics framework described in [5] is *slightly* different from the original one and, as such, it is not a model of the geometry of interaction. This is why we introduce a lifting construction in our game model.

Moreover, the global analysis needed in [5] to extract the complexity of a proof from its interpretation cannot be easily turned into a more interactive analysis, in the spirit of game semantics. The extraction of time bounds from proof interpretations is somehow internalized here through the notion of *time analyzer* (see Section 2). One of the key technical lemmas towards the quantitative correspondence cited above is proved through a game-theoretical reducibility argument (see Section 4).

Another semantic framework which has been designed with similar goals is Ghica's slot games [8]. There, however, the idea is playing slots in correspondence with any potential redex in a program, while here we focus on exponentials. On the other hand, the idea of using slots to capture intensional properties of proofs (or programs) in an interactive way is one of the key ingredients of this paper. In Section 5 the reader can find a more detailed comparison with Ghica's work. To keep the presentation simple, we preferred to adopt Ghica's way of introducing cost into games, rather than Leperchey's time monad.

In Baillot and Pedicini's geometry of interaction model [4], the "cost" of a proof is strongly related to the length of regular paths in its interpretation. But this way, one can easily define a family of terms which normalize in linear time but have exponential cost.

2 Syntax

We here introduce multiplicative exponential linear logic as a sequent calculus. It would be more natural to deal with proof-nets instead of the sequent calcu-

$$\frac{}{A \vdash A} A \quad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\varsigma(\Gamma, \Delta) \vdash B} U \quad \frac{\Gamma \vdash B}{\varsigma(\Gamma, !A) \vdash B} W \quad \frac{\Gamma, !A, !A \vdash B}{\varsigma(\Gamma, !A) \vdash B} C$$

$$\frac{\Gamma, A \vdash B}{\varsigma(\Gamma) \vdash A \multimap B} R_{\multimap} \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\varsigma(\Gamma, \Delta, A \multimap B) \vdash C} L_{\multimap} \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\varsigma(\Gamma, \Delta) \vdash A \otimes B} R_{\otimes} \quad \frac{\Gamma, A, B \vdash C}{\varsigma(\Gamma, A \otimes B) \vdash C} L_{\otimes}$$

$$\frac{A_1, \ldots, A_n \vdash B}{\varsigma(!A_1, \ldots, !A_n) \vdash !B} P_! \quad \frac{\Gamma, A \vdash B}{\varsigma(\Gamma, !A) \vdash B} D_! \quad \frac{\Gamma, !!A \vdash B}{\varsigma(\Gamma, !A) \vdash B} N_!$$

Fig. 1. A sequent calculus for IMELL

lus, but our semantic constructions will rely on a precise sequentiality in proof constructions that we would have to rebuild in a proof-net setting.

The language of *formulas* is defined by the following productions:

$$A ::= \alpha \mid A \multimap A \mid A \otimes A \mid !A$$

where α ranges over a countable set of *atoms*. A context is a sequence $\Gamma = A_1, \ldots, A_n$ of formulas. If $\Gamma = A_1, \ldots, A_n$ is a context and $\varsigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a permutation, then $\varsigma(\Gamma)$ stands for the context $A_{\varsigma(1)}, \ldots, A_{\varsigma(n)}$.

The rules in Figure 1 define a sequent calculus for (intuitionistic) multiplicative and exponential linear logic, IMELL, with an exchange rule integrated in the other ones. Our presentation uses an explicit *digging* rule $N_!$ as often done in the geometry of interaction setting (see [9] for some comments).

Given any proof $\pi : \Gamma \vdash A$, we can build another (cut-free) proof $[\pi] : \Phi, \Gamma \vdash A$, where Φ is a sequence in the form $!^{k_1}(A_1 \multimap A_1), \ldots, !^{k_n}(A_n \multimap A_n)$. We say that "cuts are exposed" in $[\pi]$. It is defined as follows, by induction on the structure of π:

- If the last rule in π is not U and the immediate subproofs of π are $\rho_1, \ldots, \rho_n$, then $[\pi]$ is obtained from $[\rho_1], \ldots, [\rho_n]$ in the natural way. For a promotion rule, as an example, π and $[\pi]$ are given by:

$$\frac{\rho : A_1, \ldots, A_n \vdash B}{\varsigma(!A_1, \ldots, !A_n) \vdash !B} P_! \qquad \frac{[\rho] : \Phi, A_1, \ldots, A_n \vdash B}{!\Phi, \varsigma(!A_1, \ldots, !A_n) \vdash !B} P_!$$

- For a cut rule, π and $[\pi]$ are given by:

$$\frac{\rho : \Gamma \vdash B \quad \sigma : \Delta, B \vdash A}{\varsigma(\Gamma, \Delta) \vdash A} U \qquad \frac{[\rho] : \Phi, \Gamma \vdash B \quad [\sigma] : \Psi, \Delta, B \vdash A}{\Phi, B \multimap B, \Psi, \varsigma(\Gamma, \Delta) \vdash A} L_{\multimap}$$

The cut-elimination steps $\pi \rightsquigarrow \rho$ are an easy adaptation of the usual ones. We just have to take care of the exchange parts, but they can be handled without any particular problem. To avoid stupid loops, we allow a cut rule c to commute upwards with another cut rule d during reduction only if d introduces the left premise of c (and not if d introduces the right premise of c).

For our complexity analysis to make sense, we will restrict the cut elimination procedure to a particular strategy of reduction called *surface reduction*. From a proof-net point of view it corresponds to reducing cuts at depth 0 only. In a sequent calculus setting, we only apply a reduction step to a cut rule if it is not above a promotion rule $P_!$. There are several reasons why surface reduction has been considered here:

- It corresponds to various reduction strategies for the lambda calculus. In particular, if lambda terms are encoded into IMELL via the cbn encoding $A \to B \equiv !A \multimap B$, then surface reduction simulates head reduction. On the other hand, the cbv encoding $A \to B \equiv !(A \multimap B)$ induces a simulation of (weak) call-by-value reduction by surface reduction.
- An upper bound to the time complexity of cut-elimination can be obtained by considering the so-called level-by-level strategy [5]. But the level-by-level strategy is nothing more than an iteration of surface reduction. As a consequence, our semantic interpretation could be applied itself iteratively to obtain bounds on the time complexity of ordinary cut-elimination.
- Any proof whose conclusion does not contain the modal operator ! in *positive* position can be normalized using surface reduction. In the cbn encoding, formulas for infinite datatypes *do* contain ! in positive position, but those positive occurrences can be "linearized" with appropriate coercion maps. As an example, natural numbers are encoded as $N = !(!A \multimap A) \multimap !A \multimap A$, but there is an easy coercion $N \multimap N_{lin}$, where $N_{lin} = !(A \multimap A) \multimap !A \multimap A$.

For practical reasons, we introduce a particular atomic formula $\mathbb{U}$ and we extend the IMELL system with the following "pseudo"-rules (which are not valid from a logical point of view):

$$\frac{}{\vdash X}\,a \qquad\qquad \frac{\Gamma \vdash A}{\varsigma(X,\Gamma) \vdash A}\,w$$

where X is any atomic formula: α or $\mathbb{U}$.

This allows us to define a proof TA_A of $A \vdash \mathbb{U}$ and a proof TA^0_A of $\vdash A$. TA_A is called the *time analyzer* of A. They are defined by mutual induction on A:

$$\frac{\dfrac{}{\vdash \mathbb{U}}\,a}{\mathsf{TA}_\alpha : \alpha \vdash \mathbb{U}}\,w \qquad \frac{}{\mathsf{TA}^0_\alpha : \vdash \alpha}\,a \qquad \frac{\dfrac{}{\vdash \mathbb{U}}\,a}{\mathsf{TA}_{!A} : !A \vdash \mathbb{U}}\,W \qquad \frac{\mathsf{TA}^0_A : \vdash A}{\mathsf{TA}^0_{!A} : \vdash !A}\,P_!$$

$$\frac{\dfrac{\mathsf{TA}_A : A \vdash \mathbb{U} \quad \dfrac{\mathsf{TA}_B : B \vdash \mathbb{U}}{\mathbb{U}, B \vdash \mathbb{U}}\,w}{A, B \vdash \mathbb{U}}\,U}{\mathsf{TA}_{A\otimes B} : A \otimes B \vdash \mathbb{U}}\,L_\otimes \qquad \frac{\mathsf{TA}^0_A : \vdash A \quad \mathsf{TA}^0_B : \vdash B}{\mathsf{TA}^0_{A\otimes B} : \vdash A \otimes B}\,R_\otimes$$

$$\frac{\mathsf{TA}^0_A : \vdash A \quad \mathsf{TA}_B : B \vdash \mathbb{U}}{\mathsf{TA}_{A\multimap B} : A \multimap B \vdash \mathbb{U}}\,L_\multimap \qquad \frac{\dfrac{\mathsf{TA}_A : A \vdash \mathbb{U} \quad \dfrac{\mathsf{TA}^0_B : \vdash B}{\mathbb{U} \vdash B}\,w}{A \vdash B}\,U}{\mathsf{TA}^0_{A\multimap B} : \vdash A \multimap B}\,R_\multimap$$

3 Game Semantics

The game model we use is based on the constructions presented in [1]. We extend it with the simplest exponential construction (by enriching moves with copy indexes [10, 2], except that we use a presentation with exponential signatures in the spirit of the geometry of interaction [6]) together with a lifting operation (adding two fresh moves at the beginning of a game).

3.1 Games

A *game* A consists in:

- A set of *moves* M_A.
- A function $\lambda_A : M_A \to \{\mathsf{P}, \mathsf{O}\}$. $\overline{\lambda_A}$ denotes the function from M_A to $\{\mathsf{P}, \mathsf{O}\}$ defined by $\overline{\lambda_A}(m) \neq \lambda_A(m)$. $M_A^{\circledast}$ denotes the subset of M_A^* containing alternating, opponent-initiated sequences only, i.e., $\lambda_A(m) = \mathsf{O}$ whenever $ms \in M_A^{\circledast}$ and, moreover, $\lambda_A(m) \neq \lambda_A(n)$ whenever $smnr \in M_A^{\circledast}$. M_A^{P} and M_A^{O} are subsets of M_A defined in the natural way.
- A set P_A of *valid plays* such that $P_A \subseteq M_A^{\circledast}$ and P_A is closed under prefixes.

The language $\mathscr{E}$ of *exponential signatures* is defined by induction from the following set of productions: $t, s, u ::= \mathsf{e} \mid \mathsf{l}(t) \mid \mathsf{r}(t) \mid \mathsf{p}(t) \mid \mathsf{n}(t, t)$.

3.2 Constructions on Games

To each connective corresponds a game construction. In the particular case of the exponential connective !, we decompose its interpretation in our model into a "sequentiality construction" given by lifting and a "copying construction" given by a traditional exponential construction with copy indexes given by exponential signatures.

- **Atomic game** α:
  - $M_\alpha = \{\alpha^{\mathsf{P}}, \alpha^{\mathsf{O}}\}$.
  - $\lambda_\alpha(\alpha^{\mathsf{P}}) = \mathsf{P}$ and $\lambda_\alpha(\alpha^{\mathsf{O}}) = \mathsf{O}$.
  - P_α is $\{\varepsilon, \alpha^{\mathsf{O}}, \alpha^{\mathsf{O}} \cdot \alpha^{\mathsf{P}}\}$.

  One particular atomic game is called $\mathbb{U}$ with moves denoted by a (instead of α^{P}) and q (instead of α^{O}).
- **Tensor game** $A \otimes B$:
  - $M_{A\otimes B} = M_A + M_B$. If $s \in M_{A\otimes B}^*$, then s_A denotes the subsequence of s consisting of moves in M_A. Similarly for s_B.
  - $\lambda_{A\otimes B} = \lambda_A + \lambda_B$.
  - The elements of $P_{A\otimes B}$ are sequences $s \in M_{A\otimes B}^{\circledast}$ such that $s_A \in P_A$, $s_B \in P_B$.
- **Arrow game** $A \multimap B$:
  - $M_{A\multimap B} = M_A + M_B$.
  - $\lambda_{A\multimap B} = \overline{\lambda_A} + \lambda_B$.
  - The elements of $P_{A\multimap B}$ are sequences $s \in M_{A\multimap B}^{\circledast}$ such that $s_A \in P_A$, $s_B \in P_B$.
- **Lifting game** $\downarrow A$:
  - $M_{\downarrow A} = M_A + \{\mathsf{open}, \mathsf{close}\}$.
  - $\lambda_{\downarrow A}(m) = \lambda_A(m)$ whenever $m \in M_A$, $\lambda_{\downarrow A}(\mathsf{open}) = \mathsf{O}$, $\lambda_{\downarrow A}(\mathsf{close}) = \mathsf{P}$.
  - $P_{\downarrow A}$ is $\{\varepsilon, \mathsf{open}\} \cup \{\mathsf{open} \cdot \mathsf{close} \cdot s \mid s \in P_A\}$.
- **Exponential game** $\#A$:
  - $M_{\#A} = \mathscr{E} \times M_A$. Given any sequence s in $M_{\#A}^*$ and any exponential signature t, s_t denotes the subsequence of s consisting in moves in the form (t, m). Given any sequence s in M_A^* and any exponential signature t, $t \times s$ denotes the sequence in $M_{\#A}^*$ obtained by pairing each move in s with t.

- $\lambda_{\#A}(t, m) = \lambda_A(m)$.
- The elements of $P_{\#A}$ are sequences $s \in M_{\#A}^{\circledast}$ such that for every $t \in \mathscr{E}$, $s_t = t \times r$ with $r \in P_A$.

We will often use the notation $!A$ for $\#\downarrow A$.

3.3 Strategies

Proofs are interpreted as particular strategies over games. However since we are not looking for full completeness results (but for complexity full abstraction instead), we are not particularly restrictive on the kind of strategies we deal with. There is no particular notion of uniformity on strategies such as history-freeness, innocence, etc. Important properties of strategies coming from proofs will be recovered through realizability (see Section 4).

A *strategy* σ over a game A is a non-empty set of even-length plays in P_A satisfying the following conditions:

- σ is *even-prefix-closed*;
- σ is *deterministic*: if $smn \in \sigma$, $sml \in \sigma$, then $n = l$.

A strategy σ over A is *total* if $s \in \sigma$ and $sm \in P_A$ implies $smn \in \sigma$ for some $n \in M_A$.

Composition of strategies can be defined in the usual way. Given a strategy σ over $A \multimap B$ and τ over $B \multimap C$, we can first define $\sigma \parallel \tau$ as follows:

$$\sigma \parallel \tau = \{s \in (M_A + M_B + M_C)^* \mid s_{A,B} \in \sigma \wedge s_{B,C} \in \tau\}.$$

where $s_{A,B}$ denotes the subsequence of s consisting of moves in $M_A + M_B$ and similarly for $s_{B,C}$ and for $s_{A,C}$.

The *composition* of σ and τ, denoted $\sigma;\tau$ is simply $\sigma;\tau = \{s_{A,C} \mid s \in \sigma \parallel \tau\}$.

Proposition 1. *If σ is a strategy over $A \multimap B$ and τ is a strategy over $B \multimap C$, then $\sigma;\tau$ is a strategy over $A \multimap C$.*

A useful restriction on strategies is given by *history-free strategies* σ satisfying: if $sm \cdot next_\sigma(m) \in P_A$ then $sm \cdot next_\sigma(m) \in \sigma$ if and only if $s \in \sigma$ where $next_\sigma$ is the generating partial function from M_A^{O} to M_A^{P}. The composition of two history-free strategies is an history-free strategy generated by the composition of generating functions. Some of the strategies we use happen to be history-free, but not all of them are.

The history-free *identity strategy* $\mathbf{id}_A$ over $A \multimap A$ is given by the generating function (assume $A \multimap A$ is $A_1 \multimap A_2$):

$$\forall m \in M_A^{\mathsf{O}}.next_{\mathbf{id}_A}(m_{A_2}) = m_{A_1}$$
$$\forall m \in M_A^{\mathsf{P}}.next_{\mathbf{id}_A}(m_{A_1}) = m_{A_2}$$

According to [1], games and strategies define a symmetric monoidal closed category (SMCC).

3.4 Constructions on Strategies

We describe elementary constructions on strategies which, once plugged together, will allow us to interpret proofs in the game model.

- **Left-lifting Strategy**: Given a strategy σ over the game $A \otimes B \multimap C$, the subset $\mathbf{ll}(\sigma)$ of $P_{\downarrow A \otimes B \multimap C}$ is defined as follows:

$$\mathbf{ll}(\sigma) = \{\varepsilon\} \cup \{m \cdot \mathsf{open}_{\downarrow A} \mid \exists ms \in \sigma\} \cup \{m \cdot \mathsf{open}_{\downarrow A} \cdot \mathsf{close}_{\downarrow A} \cdot s \mid ms \in \sigma\}$$

  In the same spirit, we can define $\mathbf{ll}_B(\sigma)$ over $A \otimes \downarrow B \multimap C$ (so that $\mathbf{ll}_A(\sigma) = \mathbf{ll}(\sigma)$).
- **Right-lifting Strategy**: Given a strategy σ over the game A, the subset $\mathbf{rl}(\sigma)$ of $P_{\downarrow A}$ is defined as follows:

$$\mathbf{rl}(\sigma) = \{\varepsilon\} \cup \{\mathsf{open}_{\downarrow A} \cdot \mathsf{close}_{\downarrow A} \cdot s \mid s \in \sigma\}$$

  Using the immediate bijection between $M_{\downarrow(A \multimap B)}$ and $M_{A \multimap \downarrow B}$, if σ is a strategy over $A \multimap B$, we will often use $\mathbf{rl}(\sigma)$ as a strategy over $A \multimap \downarrow B$.
- **Lifting Strategy**: Given a strategy σ over the game $A_1 \otimes \cdots \otimes A_n \multimap B$, the subset $\mathbf{l}(\sigma)$ of $P_{\downarrow A_1 \otimes \cdots \otimes \downarrow A_n \multimap \downarrow B}$ is defined by $\mathbf{l}(\sigma) = \mathbf{ll}_{A_1}(\ldots \mathbf{ll}_{A_n}(\mathbf{rl}(\sigma)))$.
- **Dereliction Strategy**: The subset $\mathbf{d}_A$ of $P_{\#A \multimap A}$ is the one induced by the following (assume $\#A \multimap A$ is $\#A_1 \multimap A_2$):

$$\forall m \in M_A^{\mathsf{O}}.next_{\mathbf{d}_A}(m_{A_2}) = (\mathsf{e}, m)_{\#A_1}$$
$$\forall m \in M_A^{\mathsf{P}}.next_{\mathbf{d}_A}((\mathsf{e}, m)_{\#A_1}) = m_{A_2}$$

- **Digging Strategy**: The subset $\mathbf{n}_A$ of $P_{\#\downarrow A \multimap \downarrow\#\downarrow\#\downarrow A}$ is the one induced by the following (assume $\#\downarrow A \multimap \downarrow\#\downarrow\#\downarrow A$ is $\#\downarrow A_1 \multimap \downarrow\#\downarrow\#\downarrow A_2$):

$$next_{\mathbf{n}_A}(\mathsf{open}_{\downarrow\#\downarrow\#\downarrow A_2}) = (\mathsf{e}, \mathsf{open})_{\#\downarrow A_1}$$
$$next_{\mathbf{n}_A}((\mathsf{e}, \mathsf{close})_{\#\downarrow A_1}) = \mathsf{close}_{\downarrow\#\downarrow\#\downarrow A_2}$$
$$next_{\mathbf{n}_A}((t, \mathsf{open})_{\downarrow\#\downarrow\#\downarrow A_2}) = (\mathsf{p}(t), \mathsf{open})_{\#\downarrow A_1}$$
$$next_{\mathbf{n}_A}((\mathsf{p}(t), \mathsf{close})_{\#\downarrow A_1}) = (t, \mathsf{close})_{\downarrow\#\downarrow\#\downarrow A_2}$$
$$\forall m \in M_A^{\mathsf{O}}.next_{\mathbf{n}_A}((t, (s, m))_{\downarrow\#\downarrow\#\downarrow A_2}) = (\mathsf{n}(t, s), m)_{\#\downarrow A_1}$$
$$\forall m \in M_A^{\mathsf{P}}.next_{\mathbf{n}_A}((\mathsf{n}(t, s), m)_{\#\downarrow A_1}) = (t, (s, m))_{\downarrow\#\downarrow\#\downarrow A_2}$$

- **Contraction Strategy**: The subset $\mathbf{c}_A$ of $P_{\#\downarrow A \multimap \downarrow\#\downarrow A \otimes \#\downarrow A}$ is the one induced by the following (assume $\#\downarrow A \multimap \downarrow\#\downarrow A \otimes \#\downarrow A$ is $\#\downarrow A_1 \multimap \downarrow\#\downarrow A_2 \otimes \#\downarrow A_3$):

$$next_{\mathbf{c}_A}(\mathsf{open}_{\downarrow\#\downarrow A_2}) = (\mathsf{e}, \mathsf{open})_{\#\downarrow A_1}$$
$$next_{\mathbf{c}_A}((\mathsf{e}, \mathsf{close})_{\#\downarrow A_1}) = \mathsf{close}_{\downarrow\#\downarrow A_2}$$
$$\forall m \in M_A^{\mathsf{O}}.next_{\mathbf{c}_A}((t, m)_{\downarrow\#\downarrow A_2}) = (\mathsf{l}(t), m)_{\#\downarrow A_1}$$
$$\forall m \in M_A^{\mathsf{P}}.next_{\mathbf{c}_A}((\mathsf{l}(t), m)_{\#\downarrow A_1}) = (t, m)_{\downarrow\#\downarrow A_2}$$
$$\forall m \in M_A^{\mathsf{O}}.next_{\mathbf{c}_A}((t, m)_{\#\downarrow A_3}) = (\mathsf{r}(t), m)_{\#\downarrow A_1}$$
$$\forall m \in M_A^{\mathsf{P}}.next_{\mathbf{c}_A}((\mathsf{r}(t), m)_{\#\downarrow A_1}) = (t, m)_{\#\downarrow A_3}$$

- **Promotion Strategy**: Given a strategy σ over the game $A_1 \otimes \cdots \otimes A_n \multimap B$, the subset $\mathbf{p}(\sigma)$ of $P_{\#A_1 \otimes \cdots \otimes \#A_n \multimap \#B}$ is defined as follows:

$$\mathbf{p}(\sigma) = \{s \in P_{\#A_1 \otimes \cdots \otimes \#A_n \multimap \#B} \mid \forall t. \exists r \in \sigma. s_t = t \times r\}$$

We use the notation $\mathbf{pl}(\sigma)$ for $\mathbf{p}(\mathbf{l}(\sigma))$.

Proposition 2. *For any game A, $\mathbf{d}_A$, $\mathbf{n}_A$ and $\mathbf{c}_A$ are strategies. Let σ be a strategy over $A_1 \otimes \cdots \otimes A_n \multimap B$. Then $\mathbf{rl}(\sigma)$ and $\mathbf{p}(\sigma)$ are strategies and, if $n \geq 1$, $\mathbf{ll}(\sigma)$ is a strategy.*

3.5 Interpretation of Proofs

We define the strategy $[\![\pi]\!]$ interpreting a proof π.

The multiplicative rules are interpreted according to the symmetric monoidal closed structure of the category of games and strategies. The interpretation of the exponential rules is based on the constructions described above.

- **Weakening**: if σ is a strategy over $\Gamma \multimap B$, it is also a strategy over $A \otimes \Gamma \multimap B$ and we can build $(\mathbf{d}_{\downarrow A} \otimes \mathbf{id}_\Gamma); \mathbf{ll}(\sigma)$ as a strategy over $!A \otimes \Gamma \multimap B$.
- **Contraction**: if σ is a strategy over $!A \otimes !A \otimes \Gamma \multimap B$, we can build $(\mathbf{c}_A \otimes \mathbf{id}_\Gamma); \mathbf{ll}(\sigma)$ as a strategy over $!A \otimes \Gamma \multimap B$.
- **Promotion**: if σ is a strategy over $A_1 \otimes \cdots \otimes A_n \multimap B$, we can build $\mathbf{pl}(\sigma)$ as a strategy over $!A_1 \otimes \cdots \otimes !A_n \multimap !B$.
- **Dereliction**: if σ is a strategy over $A \otimes \Gamma \multimap B$, we can build $(\mathbf{d}_{\downarrow A} \otimes \mathbf{id}_\Gamma); \mathbf{ll}(\sigma)$ as a strategy over $!A \otimes \Gamma \multimap B$.
- **Digging**: if σ is a strategy over $!!A \otimes \Gamma \multimap B$, we can build $(\mathbf{n}_A \otimes \mathbf{id}_\Gamma); \mathbf{ll}(\sigma)$ as a strategy over $!A \otimes \Gamma \multimap B$.

The main difference between weakening and dereliction comes from the original strategy: over $\Gamma \multimap B$ for weakening and considered over $A \otimes \Gamma \multimap B$, while "really" over $A \otimes \Gamma \multimap B$ for dereliction.

Theorem 1 (Soundness). *If $\pi \rightsquigarrow \rho$ then $[\![\pi]\!] = [\![\rho]\!]$.*

Proof. The multiplicative steps are given by the SMCC structure. The permutations of formulas are handled by the symmetry of the SMCC structure. The key properties required for the other cases are:

- If $\sigma : A_0 \multimap A$ and $\tau : A \otimes B \multimap C$ then $(\sigma \otimes \mathbf{id}_{\downarrow B}); \mathbf{ll}_B(\tau) = \mathbf{ll}_B((\sigma \otimes \mathbf{id}_B); \tau)$.
- If $\sigma : A_1 \otimes \cdots \otimes A_n \multimap A$ and $\tau : A \otimes A_{n+1} \otimes \cdots \otimes A_m \multimap B$ then $(\mathbf{p}(\sigma) \otimes \mathbf{id}_{\#A_{n+1}} \otimes \cdots \otimes \mathbf{id}_{\#A_m}); \mathbf{p}(\tau) = \mathbf{p}((\sigma \otimes \mathbf{id}_{A_{n+1}} \otimes \cdots \otimes \mathbf{id}_{A_m}); \tau)$.
- If $\sigma : A_1 \otimes \cdots \otimes A_n \multimap B$ then $\mathbf{p}(\sigma); \mathbf{d}_B = (\mathbf{d}_{A_1} \otimes \cdots \otimes \mathbf{d}_{A_n}); \sigma$.
- If $\sigma : A_1 \otimes \cdots \otimes A_n \multimap B$ then $\mathbf{pl}(\sigma); \mathbf{c}_B = (\mathbf{c}_{A_1} \otimes \cdots \otimes \mathbf{c}_{A_n}); \mathbf{l}(\mathbf{pl}(\sigma)) \otimes \mathbf{pl}(\sigma)$ (up to some permutation in the second composition turning $\downarrow !A_1 \otimes !A_1 \otimes \cdots \otimes \downarrow !A_n \otimes !A_n$ into $\downarrow !A_1 \otimes \cdots \otimes \downarrow !A_n \otimes !A_1 \otimes \cdots \otimes !A_n$).
- If $\sigma : A_1 \otimes \cdots \otimes A_n \multimap B$ then $\mathbf{pl}(\sigma); \mathbf{n}_B = (\mathbf{n}_{A_1} \otimes \cdots \otimes \mathbf{n}_{A_n}); \mathbf{l}(\mathbf{pl}(\mathbf{pl}(\sigma)))$. □

We extend the interpretation to the formula $\mathbb{U}$ and to pseudo-rules. The pseudo-rule a is interpreted by the strategy $\{\varepsilon, X^{\mathsf{O}} \cdot X^{\mathsf{P}}\}$. If σ is the interpretation of the premise of an application of the pseudo-rule w, its conclusion is interpreted by $\{\varepsilon\} \cup \{m \cdot X^{\mathsf{O}} \mid \exists ms \in \sigma\} \cup \{m \cdot X^{\mathsf{O}} \cdot X^{\mathsf{P}} \cdot s \mid ms \in \sigma\}$. X^{O} denotes α^{O} if $X = \alpha$ and q if $X = \mathbb{U}$. X^{P} denotes α^{P} if $X = \alpha$ and a if $X = \mathbb{U}$.

If σ is the interpretation of a (pseudo)-proof, then σ is *total*.

4 Realizability

In order to prove properties of the strategies interpreting proofs, we are going to define a notion of realizability between strategies and formulas.

The relations "σ P-realizes A", $\sigma \Vdash^P A$, (with σ strategy over A) and "τ O-realizes A", $\tau \Vdash^O A$, (with τ strategy over $A \multimap \mathbb{U}$) are defined in a mutually recursive way by induction on A:

- $\sigma \Vdash^P \alpha$ if $\sigma = \{\varepsilon, \alpha^{\mathsf{O}} \cdot \alpha^{\mathsf{P}}\}$
- $\tau \Vdash^O \alpha$ if $\tau = \{\varepsilon, \mathsf{q} \cdot \alpha^{\mathsf{O}}, \mathsf{q} \cdot \alpha^{\mathsf{O}} \cdot \alpha^{\mathsf{P}} \cdot \mathsf{a}\}$
- $\sigma \Vdash^P \mathbb{U}$ if $\sigma = \{\varepsilon, \mathsf{q} \cdot \mathsf{a}\}$
- $\tau \Vdash^O \mathbb{U}$ if $\tau = \mathbf{id}_{\mathbb{U}}$
- $\sigma \Vdash^P A \otimes B$ if $\sigma_A \Vdash^P A$ and $\sigma_B \Vdash^P B$ with $\sigma_A = \{s_A \mid s \in \sigma\}$. (We ask in particular that σ_A and σ_B are strategies over A and B, respectively.)
- $\tau \Vdash^O A \otimes B$ if for any $\sigma \Vdash^P A$, $\sigma; \tau \Vdash^O B$ and for any $\sigma \Vdash^P B$, $\sigma; \tau \Vdash^O A$. (Using that, up to the curryfication isomorphisms, τ can also be seen as a strategy over $A \multimap (B \multimap \mathbb{U})$ or over $B \multimap (A \multimap \mathbb{U})$.)
- $\sigma \Vdash^P A \multimap B$ if for any $\delta \Vdash^P A$, $\delta; \sigma \Vdash^P B$ and for any $\tau \Vdash^O B$, $\sigma; \tau \Vdash^O A$
- $\tau \Vdash^O A \multimap B$ if $\tau_A \Vdash^P A$ and $\tau_{B \multimap \mathbb{U}} \Vdash^O B$
- $\sigma \Vdash^P\, !A$ if for any exponential signature t, $\sigma\restriction_t \Vdash^P A$ with $\sigma\restriction_t = \{s\restriction_t \mid s \in \sigma\}$ and $s\restriction_t$ is obtained from s_t by replacing any move (t, m) by m and by then erasing the initial open and close moves if they appear (we ask in particular that $\sigma\restriction_t$ is a strategy over A for any t).
- $\tau \Vdash^O\, !A$ if τ contains the play $\mathsf{q} \cdot (\mathsf{e}, \mathsf{open})$.

An adequacy property relates proofs, strategies and realizability:

Proposition 3. *For every proof π, the strategy $[\![\pi]\!]$ P-realizes the conclusion of π.*

Proof. A first remark is that if $\sigma \Vdash^P A$ then σ contains a non-empty play and if $\tau \Vdash^O A$ then τ contains a play with a move in A (by induction on A). We now do the proof by induction on π. We only give a few typical cases.

- Right tensor: if $\sigma_1 \Vdash^P \Gamma \multimap A$ and $\sigma_2 \Vdash^P \Delta \multimap B$, and if $\delta \Vdash^P \Gamma \otimes \Delta$ then $\delta_\Gamma \Vdash^P \Gamma$ and $\delta_\Delta \Vdash^P \Delta$ so that $\delta_\Gamma; \sigma_1 \Vdash^P A$ and $\delta_\Delta; \sigma_2 \Vdash^P B$, and finally $\delta; (\sigma_1 \otimes \sigma_2) \Vdash^P A \otimes B$. If $\tau \Vdash^O A \otimes B$, $\delta_1 \Vdash^P \Gamma$ and $\delta_2 \Vdash^P \Delta$, we have: $(\delta_1 \otimes \mathbf{id}_\Delta); (\sigma_1 \otimes \sigma_2); \tau = (\delta_1; \sigma_1) \otimes \sigma_2; \tau = \sigma_2; ((\delta_1; \sigma_1); \tau)$, but $\delta_1; \sigma_1 \Vdash^P A$ thus $(\delta_1; \sigma_1); \tau \Vdash^O B$ and $(\delta_1 \otimes \mathbf{id}_\Delta); (\sigma_1 \otimes \sigma_2); \tau \Vdash^O \Delta$. In a similar way $(\mathbf{id}_\Gamma \otimes \delta_2); (\sigma_1 \otimes \sigma_2); \tau \Vdash^O \Gamma$.
- Promotion: if $\sigma \Vdash^P A_1 \otimes \cdots \otimes A_n \multimap B$ (with σ' obtained from σ by interpreting the promotion rule) and if $\delta_i \Vdash^P\, !A_i$ $(1 \leq i \leq n)$, for any exponential signature t we have $\delta_i\restriction_t \Vdash^P A_i$ thus $((\delta_1 \otimes \cdots \otimes \delta_n); \sigma')\restriction_t = (\delta_1\restriction_t \otimes \cdots \otimes \delta_n\restriction_t); \sigma \Vdash^P B$. If $\tau \Vdash^O\, !B$ and $\delta_i \Vdash^P\, !A_i$ $(1 \leq i \leq n)$, for any

$1 \leq i \leq n$, $(\delta_1 \otimes \cdots \otimes \delta_{i-1} \otimes \mathbf{id}_{!A_i} \otimes \delta_{i+1} \otimes \cdots \otimes \delta_n); \sigma'; \tau$ plays $(\mathsf{e}, \mathsf{open})$ as first move in $!A_i$ since τ plays $(\mathsf{e}, \mathsf{open})$ as first move in $!B$ and each δ_i contains the play $(\mathsf{e}, \mathsf{open}) \cdot (\mathsf{e}, \mathsf{close})$. $\square$

As a consequence, $[\![\mathsf{TA}_A]\!] \Vdash^P A \multimap \mathbb{U}$ thus $[\![\mathsf{TA}_A]\!] \Vdash^O A$ (since $\mathbf{id}_{\mathbb{U}} \Vdash^O \mathbb{U}$).

A *complete set of moves* for any game A is a subset of M_A defined by induction on the structure of A:

- If $A = \alpha$, the only complete set of moves for A is $\{\alpha^{\mathsf{P}}, \alpha^{\mathsf{O}}\}$.
- If $A = B \otimes C$ or $A = B \multimap C$, C_B is a complete set of moves for B and C_C is a complete set of moves for C, then $C_A = C_B + C_C$ is a complete set of moves for A.
- If $A = !B$, then any subset of M_A containing the move $(\mathsf{e}, \mathsf{close})$ is a complete set of moves for A.

Proposition 4. *If σ P-realizes A, τ O-realizes A and $\sigma; \tau$ is total, then the maximal sequence in $\sigma \parallel \tau$ (seen as a set of moves of A) is complete.*

5 Complexity

In this Section, we show how to instrument games with slots, in the same vein as in Ghica's framework [8]. The idea is simple: slots are used by the player to communicate some quantitative properties of the underlying proof to the opponent. But while in Ghica's work slots are produced in correspondence with any potential redex, here the player raises a slot in correspondence with boxes, i.e. instances of the promotion rule. In Ghica's slot games, the complexity of a program can be read out of any complete play in its interpretation, while here the process of measuring the complexity of proofs is internalized through the notion of time analyzer (see Section 2): the complexity of π (with conclusion A) is simply the number of slots produced in the interaction between $[\![\pi]\!]$ and $[\![\mathsf{TA}_A]\!]$. Notice that the definition of TA_A only depends on the formula A.

The symbol $\bullet$ is a special symbol called a *slot*. In the new setting, the set of moves for A, will be the usual M_A, while the notion of a play should be slightly changed. Given a game A and a sequence s in $(M_A + \{\bullet\})^*$, we denote by s° the sequence in M_A^* obtained by deleting any occurrence of $\bullet$ in s. Analogously, given any subset σ of $(M_A + \{\bullet\})^*$, σ° will denote $\{s^\circ \mid s \in \sigma\} \subseteq M_A^*$.

A *play-with-costs* for A is a sequence s in $(M_A + \{\bullet\})^*$ such that $s^\circ \in P_A$, whenever $s = r \bullet mq$ it holds that $\lambda_A(m) = \mathsf{P}$ and the last symbol in s (if any) is a move in M_A. A *strategy-with-costs* for the game A is a set σ of plays-with-costs for A such that σ° is a strategy (in the usual sense) for A and, moreover, σ is *slot-deterministic*: if $sm \bullet^k n \in \sigma$ and $sm \bullet^h n \in \sigma$, then $k = h$.

Composition of strategies-with-costs needs to be defined in a slightly different way than the one of usual strategies. In particular, we need two different notions of projections: first of all, if $s \in (M_A + M_B + \{\bullet\})^*$, we can construct s_{A^X} (where $X \subseteq \{\mathsf{P}, \mathsf{O}\}$) by extracting from s any move $m \in M_A$ together with the slots immediately before any such m provided $\lambda_A(m) \in X$. But we can even construct $s_{A^\bullet}$, by only considering the slots which precede moves in M_A *but not the moves* *themselves*. Given strategies-with-costs σ over $A \multimap B$ and τ over $B \multimap C$, we can first define $\sigma \parallel \tau$ as follows:

$$\sigma \parallel \tau = \{s \in (M_A + M_B + M_C + \{\bullet\})^* \mid s_{A^{\mathsf{P},\mathsf{O}},B^{\mathsf{P}}} \in \sigma \wedge s_{B^{\mathsf{O}},C^{\mathsf{P},\mathsf{O}}} \in \tau\}.$$

The *composition* of σ and τ, denoted $\sigma;\tau$ is now simply

$$\sigma;\tau = \{s_{A^{\mathsf{P},\mathsf{O}},B^{\bullet},C^{\mathsf{P},\mathsf{O}}} \mid s \in \sigma \parallel \tau\}.$$

In other words, we forget the moves in M_B, but we keep all the slots produced by them.

Proposition 5. *If σ is a strategy-with-costs over $A \multimap B$ and τ is a strategy-with-costs over $B \multimap C$, then $\sigma;\tau$ is a strategy-with-costs over $A \multimap C$.*

The strategy constructions we have seen so far can be turned into strategy-with-costs constructions. In the basic strategies, slots come into play only in $\mathbf{rl}(\sigma)$: in particular, $\mathbf{rl}^i(\sigma) = \{\varepsilon\} \cup \{\mathsf{open}_{\downarrow A} \cdot \bullet^i \cdot \mathsf{close}_{\downarrow A} \cdot s \mid s \in \sigma\}$. This way, the interpretation $[\![\pi]\!]^i$ of any proof π is parametrized on a natural number i.

In the particular case of a cut-free proof π with axiom rules only introducing !-free formulas, $[\![\pi]\!]^i$ can be easily deduced from $[\![\pi]\!]$ by adding $\bullet^i$ before each P-move of the shape (t, close) in each play of $[\![\pi]\!]$.

We are in a position to define the complexity $\mathcal{C}(\pi)$ of any proof π. First, consider the shape of any non-trivial play-with-costs s in a strategy-with-costs σ for $\mathbb{U}$: it must have the following shape $\mathsf{q}\ \bullet^i\ \mathsf{a}$. But observe that this play is the *only* non-trivial play-with-costs in σ, due to (slot) determinacy. The integer i is called the *complexity* of σ, denoted $\mathcal{C}(\sigma)$. This way we can define the complexity $\mathcal{C}(\pi)$ of any proof π with conclusion A as simply the complexity of π when composed with the time analyzer: $\mathcal{C}([\![\pi]\!]^1; [\![\mathsf{TA}_A]\!]^0)$. The complexity of π is defined for every π because $[\![\pi]\!]; [\![\mathsf{TA}_A]\!]$ P-realizes $\mathbb{U}$ (by Proposition 3) and, as a consequence, contains a non-empty play. Given any play-with-costs s, $\mathcal{C}(s)$ is simply the number of occurrences of $\bullet$ in s.

5.1 Dynamics under Exposed Cuts

In this Section, we will prove some lemmas about the preservation of semantics when cuts are exposed as in the $[\cdot]$ construction (see Section 2). With $[\![\pi]\!]_{eca}$ we denote the (unique) maximal (wrt the prefix order) sequence in $\iota \parallel ([\![\pi]\!]^1_{ec}; [\![\mathsf{TA}_B]\!]^0)$, where $[\pi] : !^{k_1}(A_1 \multimap A_1), \ldots, !^{k_n}(A_n \multimap A_n), \Gamma \vdash C$, $[\![\pi]\!]^i_{ec} = [\![[\pi]]\!]^i$, $\iota = [\![!^{k_1} id_{A_1} \otimes \cdots \otimes !^{k_n} id_{A_n}]\!]^0$ and $B = \bigotimes \Gamma \multimap C$. $!^k id_A$ is the proof for $!^k(A \multimap A)$ obtained by applying k times the promotion rule (with an empty context) to the trivial proof id_A of $A \multimap A$. We are interested in studying how $[\![\pi]\!]_{eca}$ evolves during cut elimination for any proof $\pi : \Gamma \vdash C$. This will lead us to full abstraction. Indeed:

Remark 1. Please notice that the strategy from which we obtain the complexity of π is:

$$\tau = [\![\pi]\!]^1; [\![\mathsf{TA}_B]\!]^0 = (\iota; [\![\pi]\!]^1_{ec}); [\![\mathsf{TA}_B]\!]^0 = \iota; ([\![\pi]\!]^1_{ec}; [\![\mathsf{TA}_B]\!]^0).$$

This implies that $[\![\pi]\!]_{eca}$ contains exactly $\mathsf{C}(\pi)$ slots and, moreover, it contains a complete set of moves for $D = !^{k_1}(A_1 \multimap A_1) \otimes \cdots \otimes !^{k_n}(A_n \multimap A_n)$. This, in particular, is a consequence of Proposition 4, since $\iota \Vdash^P D$, $([\![\pi]\!]^1_{ec}; [\![\mathsf{TA}_B]\!]^0) \Vdash^O D$ and their composition is total.

The cut-elimination relation $\rightsquigarrow$ can be thought of as the union of nine reduction relations $\overset{x}{\rightsquigarrow}$ where x ranges over the set $\mathcal{R} = \{\mathcal{T}, \mathcal{X}, \otimes, \multimap, \mathcal{C}, \mathcal{D}, \mathcal{N}, \mathcal{W}, !{-}!\}$. They correspond to commuting, axiom, tensor, linear arrow, contraction, dereliction, digging, weakening and promotion-promotion cut-elimination steps. If $X \subseteq \mathcal{R}$ or $x \in \mathcal{R}$, then $\overset{X}{\rightsquigarrow}$ and $\overset{x}{\rightsquigarrow}$ have the obvious meaning. We can consider a reduction relation that postpones $!{-}!$-cuts to the very end of the computation. The resulting reduction relation is denoted with $\hookrightarrow$. Again, $\overset{X}{\hookrightarrow}$ and $\overset{x}{\hookrightarrow}$ (where $x \in \mathcal{R}$ and $X \subseteq \mathcal{R}$) have their natural meaning.

We need to analyze how $[\![\rho]\!]_{eca}$ differs from $[\![\pi]\!]_{eca}$ if $\pi \overset{x}{\rightsquigarrow} \rho$. Clearly, this crucially depends on $x \in \mathcal{R}$, since cuts are exposed in $[\pi]$ and $[\rho]$. Due to lack of space, we report just one particular case here, namely $x = \mathcal{D}$:

Lemma 1 (Dereliction). *If $\pi \overset{\mathcal{D}}{\rightsquigarrow} \rho$, then $\mathcal{C}([\![\pi]\!]_{eca}) = \mathcal{C}([\![\rho]\!]_{eca}) + 1$.*

Proof. We only consider the case where the cut reduced in π is the last rule of π. The other cases can be reduced to this one by an easy induction. With this hypothesis, $[\pi]$ is

$$\dfrac{\dfrac{[\sigma] : A_1, \ldots, A_n, D_1, \ldots, D_m \vdash B}{!A_1, \ldots, !A_n, \varsigma(!D_1, \ldots, !D_m) \vdash !B}\,P_! \qquad \dfrac{[\theta] : \Phi, \Gamma, B \vdash C}{\Phi, \vartheta(\Gamma), !B \vdash C}\,D_!}{!A_1, \ldots, !A_n, !B \multimap !B, \Phi, \varpi(!D_1, \ldots, !D_m, \Gamma) \vdash C}\,L_{\multimap}$$

and $[\rho]$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\sigma] : A_1, \ldots, A_n, D_1, \ldots, D_m \vdash B \quad [\theta] : \Phi, \Gamma, B \vdash C}{A_1, \ldots, A_n, B \multimap B, \Phi, \Gamma, D_1, \ldots, D_m \vdash C}\,L_{\multimap}}{A_1, \ldots, A_n, B \multimap B, \Phi, !D_m, \Gamma, D_1, \ldots, D_{m-1} \vdash C}\,D_!}{\vdots}\,D_!}{A_1, \ldots, A_n, B \multimap B, \Phi, !D_2, \ldots, !D_m, \Gamma, D_1 \vdash C}\,D_!}{A_1, \ldots, A_n, B \multimap B, \Phi, \varpi(!D_1, \ldots, !D_m, \Gamma) \vdash C}\,D_!$$

Observe that: $[\pi] :!A_1, \ldots, !A_n, !B_1 \multimap !B_2, \Phi_1, \Gamma_1 \vdash C$ and $[\rho] : A_1, \ldots, A_n, B_1 \multimap B_2, \Phi_1, \Gamma_1 \vdash C$. Now, consider $\iota \parallel ([\![\rho]\!]^1_{ec}; [\![\mathsf{TA}_E]\!])$ and $\iota' \parallel ([\![\pi]\!]^1_{ec}; [\![\mathsf{TA}_E]\!])$, where E is the conclusion of ρ and π. It is easy to realize that $[\![\rho]\!]_{eca}$ can be simulated by the $[\![\pi]\!]_{eca}$, in such a way that

$$[\![\rho]\!]_{eca} = [\![\pi]\!]_{eca}\{m_{B_i}/(\mathsf{e}, m_{B_i}), \cdot / \bullet^a\ (\mathsf{e}, \mu)_{!B_i}, m_{A_i}/(\mathsf{e}, m_{A_i}), \cdot / \bullet^b\ (\mathsf{e}, \mu)_{!A_i}\}$$

where μ is a metavariable for either open or close. Observe that $a = 1$ when $\mu = \mathsf{close}$ and $i = 1$ (a promotion in π raises a slot), $a = 0$ otherwise and $b = 0$ (ι does not raise any slot). But there is exactly one $(\mathsf{e}, \mathsf{close})_{!B_1}$ in $[\![\pi]\!]_{eca}$: at most one (the same move is not played twice); at least one from Proposition 4 (since strategies interpreting (pseudo)proofs are total). The thesis easily follows. □

5.2 Full Abstraction

We now have all the required material to give our key result: full abstraction of the game model with respect to the reduction length (Theorems 2 and 3).

Given a proof π and any reduction relation $\rightarrow$, $[\pi]_{\rightarrow}$ and $||\pi||_{\rightarrow}$ denote the maximum length of a reduction sequence starting in π (under $\rightarrow$) and the maximum size of any reduct of π (under $\rightarrow$), respectively. We note $|\pi|$ the size of a proof π.

Lemma 2. *For every proof π, $[\pi]_{\rightsquigarrow} = [\pi]_{\hookrightarrow}$ and $||\pi||_{\rightsquigarrow} = ||\pi||_{\hookrightarrow}$.*

Proof. Whenever $\pi \overset{!-!}{\rightsquigarrow} \rho \overset{x}{\rightsquigarrow} \sigma$ and $x \neq\, !-!$, there are $\theta_1, \ldots, \theta_n$ (where $n \geq 1$) such that $\pi \overset{x_1}{\rightsquigarrow} \theta_1 \overset{x_2}{\rightsquigarrow} \cdots \overset{x_n}{\rightsquigarrow} \theta_n \overset{x_{n+1}}{\rightsquigarrow} \sigma$, and $x_{i+1} = \,!-!$ whenever $x_i = \,!-!$. For example, if $\pi \overset{!-!}{\rightsquigarrow} \rho \overset{W}{\rightsquigarrow} \sigma$ and the box erased in the second step is exactly the one created by the first step, then clearly $\pi \overset{W}{\rightsquigarrow} \theta \overset{W}{\rightsquigarrow} \sigma$. As a consequence, for any sequence $\pi_1 \rightsquigarrow \cdots \rightsquigarrow \pi_n$ there is another sequence $\rho_1 \hookrightarrow \cdots \hookrightarrow \rho_m$ such that $\pi_1 = \rho_1$, $\pi_n = \rho_m$ and $m \geq n$. This proves the first claim. Now, observe that for any $1 \leq i \leq n$ there is j such that $|\rho_j| \geq |\pi_i|$: a simple case analysis suffices. □

Proposition 6. *If $\pi \overset{\{\mathcal{C},\mathcal{D},\mathcal{N},\mathcal{W},!-!\}}{\hookrightarrow} \rho$ then $\mathcal{C}(\pi) = \mathcal{C}(\rho) + 1$.*

Proof. From Remark 1, we know that $\mathcal{C}(\pi) = \mathcal{C}([\![\pi]\!]_{eca})$. We apply Lemma 1 (and similar statements for contraction, digging and weakening). □

Proposition 7. *If $\pi \overset{\{\mathcal{T},\mathcal{X},\otimes,\multimap\}}{\hookrightarrow} \rho$ then $\mathcal{C}(\pi) = \mathcal{C}(\rho)$.*

The mismatch between the statements of Proposition 6 and Proposition 7 can be informally explained as follows. After any "exponential" reduction step (see Proposition 6) one slot is missing, namely the one raised by the promotion rule involved in the reduction step when faced with the $(\mathsf{e}, \mathsf{open})$ move raised by the left rule interacting with the promotion rule itself. Clearly, this does not happen when performing "linear" reduction steps (see Proposition 7).

Lemma 3. *If π is cut-free, then $\mathcal{C}(\pi) \leq |\pi|$.*

Proposition 8. *If π rewrites to ρ in n steps by the $\mathcal{T}$ rule, then $|\pi| = |\rho|$ and $n \leq 2|\pi|^2$.*

Proof. The equality $|\pi| = |\rho|$ can be trivially verified whenever $\pi \overset{\mathcal{T}}{\hookrightarrow} \rho$. Now, for any proof π, define $|\pi|_{comm}$ as the sum, over all instances of the U rule inside π, of $|\sigma|_{cut} + |\sigma| + |\theta|$, where σ (respectively, θ) is the left (respectively, right) premise of the cut and $|\sigma|_{cut}$ is simply the number of instances of the cut rule in σ. For obvious reasons, $0 \leq |\pi|_{comm} \leq 2|\pi|^2$. Moreover, if $\pi \overset{\mathcal{T}}{\hookrightarrow} \rho$, then $|\pi|_{comm} > |\rho|_{comm}$. For example, consider the following commutative reduction step:

$$\dfrac{\dfrac{\pi : \Gamma \vdash A \quad \rho : \Delta, A \vdash B}{\varsigma(\Gamma, \Delta) \vdash B}\,U \qquad \sigma : \Omega, B \vdash C}{\theta : \vartheta(\Gamma, \Delta, \Omega) \vdash C}\,U \quad \rightsquigarrow \quad \dfrac{\pi : \Gamma \vdash A \qquad \dfrac{\rho : \Delta, A \vdash B \quad \sigma : \Omega, B \vdash C}{\varpi(\Delta, \Omega), A \vdash C}\,U}{\xi : \vartheta(\Gamma, \Delta, \Omega) \vdash C}\,U$$

Clearly, $|\theta| = |\xi|$ but $|\theta|_{comm} > |\xi|_{comm}$. Other cases are similar. □

Theorem 2. *For every proof* π, $\mathcal{C}(\pi) \leq [\pi]_{\leadsto} + ||\pi||_{\leadsto}$

Theorem 3. *There is a polynomial* $p : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *such that for every proof* π, $[\pi]_{\leadsto} \leq p(\mathcal{C}(\pi), |\pi|)$ *and* $||\pi||_{\leadsto} \leq p(\mathcal{C}(\pi), |\pi|)$.

Proof. By Lemma 2, the thesis easily follows from $[\pi]_{\hookrightarrow}, ||\pi||_{\hookrightarrow} \leq p(\mathcal{C}(\pi), |\pi|)$. Our first task will be to analyze the shape of any box you can find during the normalization of π by $\hookrightarrow$ up to the point where you begin to fire $!-!$ cuts. But it is easy to prove that any such box is just a subproof of π, possibly endowed with n promotions rules (where n is less than the total number of $N_!$ cuts fired during normalization). As a consequence, any such box has at most size $|\pi| + \mathcal{C}(\pi)$. Now, we can easily bound $||\pi||_{\hookrightarrow}$: at any C or N normalization step, the size of the underlying proof increases by at most $|\pi| + \mathcal{C}(\pi)$ (but the complexity strictly decreases), while in any other case the size decreases. As a consequence, $||\pi||_{\hookrightarrow} \leq \mathcal{C}(\pi)(|\pi| + \mathcal{C}(\pi))$. Now, the total number of non-commuting reduction steps is at most $\mathcal{C}(\pi) + \mathcal{C}(\pi)(|\pi| + \mathcal{C}(\pi))$. Between any of them, there are at most $2||\pi||^2_{\hookrightarrow}$ commuting steps. As a consequence:

$$\begin{aligned}[\pi]_{\hookrightarrow} &\leq \mathcal{C}(\pi) + \mathcal{C}(\pi)(|\pi| + \mathcal{C}(\pi)) + (\mathcal{C}(\pi) + \mathcal{C}(\pi)(|\pi| + \mathcal{C}(\pi)))\, 2||\pi||^2_{\hookrightarrow} \\ &\leq (\mathcal{C}(\pi) + \mathcal{C}(\pi)(|\pi| + \mathcal{C}(\pi)))\,(1 + 2(\mathcal{C}(\pi)(|\pi| + \mathcal{C}(\pi)))^2).\end{aligned}$$ □

6 Further Work

The main defect of our approach is the strong use of sequentiality information from sequent calculus proofs in the game interpretation. The two main approaches to get rid of this sequentiality are the use of non-deterministic strategies or of clusters of moves (when interpreting the promotion rule). This way we would be able to directly interpret proof-nets. In a similar spirit, we have used an exponential construction for games based on a grammar of exponential signatures, as usually done with context semantics. This is known to lead to not-very-satisfactory properties for !: for example, weakening is not neutral with respect to contraction, contraction is not commutative, etc. However, an answer to this problem should easily come from the solution proposed in the AJM setting with the notion of equivalence of strategies [2]. All these ingredients would probably allow us to turn our game model into a true categorical model of intuitionistic linear logic.

Another weakness is the restriction to surface reduction. We think adaptations to head reduction or to reduction strategies leading to normal forms should be possible by modifying the time analyzer in order to interactively access to "deeper" parts of proofs.

The notion of realizability we have introduced is tuned to reach the result we need, namely Proposition 4. However, it seems possible to modify it in various ways and to use it for very different applications in the more general context of game semantics.

Very recently, another proposal leading to similar observations but being based on relational semantics has appeared [7]. The authors give an exact measure of the number of steps required for surface reduction (and then level-by-level reduction). This should be also possible in our setting by adding lifting constructions to all the connectives (not only to the exponential ones). However an important difference comes from the notion of cut elimination under consideration: while they use β-reduction style exponential steps (coming from contractions of unbounded arity in particular), we consider standard exponential steps (based on binary contractions), and this may lead to an exponential blowup. Possible correspondences between our game-theoretical analysis and the analysis done in [7] could come from works about projecting strategies into relations.

Acknowledgements

The authors would like to thank the anonymous referees for their useful comments.

References

1. Samson Abramsky. Semantics of interaction. In Peter Dybjer and Andrew Pitts, editors, *Semantics and Logics of Computation*, Publications of the Newton Institute, pages 1–32. Cambridge University Press, 1997.
2. Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, December 2000.
3. Patrick Baillot. *Approches dynamiques en sémantique de la logique linéaire : jeux et géométrie de l'interaction*. Thèse de doctorat, Université Aix-Marseille II, 1999.
4. Patrick Baillot and Marco Pedicini. Elementary complexity and geometry of interaction. *Fundamenta Informaticae*, 45(1-2):1–31, 2001.
5. Ugo Dal Lago. Context semantics, linear logic and computational complexity. In *Proc. 21st Symposium on Logic in Computer Science*, pages 169–178. IEEE, 2006.
6. Vincent Danos, Hugo Herbelin, and Laurent Regnier. Games semantics and abstract machines. In *Proc. 11th Symposium on Logic In Computer Science*, pages 394–405. IEEE, 1996.
7. Daniel de Carvalho, Michele Pagani, and Lorenzo Tortora de Falco. A semantic measure of the execution time in linear logic. Technical Report 6441, INRIA, 2007.
8. Dan Ghica. Slot games: A quantitative model of computation. In *Proc. 32nd ACM Symposium on Principles of Programming Languages*, pages 85–97, 2005.
9. Jean-Yves Girard. Geometry of interaction III: accommodating the additives. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 329–389. Cambridge University Press, 1995.
10. Martin Hyland. Game semantics. In Peter Dybjer and Andrew Pitts, editors, *Semantics and Logics of Computation*, Publications of the Newton Institute, pages 131–184. Cambridge University Press, 1997.