



**HAL**  
open science

# Symmetric dialogue games in the proof theory of linear logic

Olivier Delande

► **To cite this version:**

Olivier Delande. Symmetric dialogue games in the proof theory of linear logic. Computer Science [cs]. Ecole Polytechnique X, 2009. English. NNT: . pastel-00006196

**HAL Id: pastel-00006196**

**<https://pastel.hal.science/pastel-00006196>**

Submitted on 30 Jun 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE POLYTECHNIQUE  
Thèse de Doctorat  
Spécialité Informatique

SYMMETRIC DIALOGUE GAMES IN THE PROOF THEORY OF  
LINEAR LOGIC

Présentée et soutenue publiquement par

OLIVIER DELANDE

le 15 octobre 2009

devant le jury composé de

Claudia	FAGGIAN	
André	HIRSCHOWITZ	Rapporteur
Olivier	LAURENT	
Paul-André	MELLIÈS	
Dale	MILLER	Directeur de thèse
Laurent	REGNIER	Rapporteur



# Acknowledgements

My first and foremost thanks go to my adviser Dale Miller for his support over the past three years. He not only provided highly valuable advice for the technical aspects of my research, but also the scientific vision that helped me put that work into perspective. On an equally important topic, he was always available and encouraging, and brought me confidence when I needed it. He also taught the first course I ever attended on the topic of logic, so I owe him my taste for this domain.

I am grateful to André Hirschowitz and Laurent Regnier for their extensive work in reviewing my thesis. Many of the ideas developed in this thesis were inspired by the discussions I had with Claudia Faggian and Paul-André Melliès, so I would like to thank them for that. I am honoured that both of them, together with Olivier Laurent, accepted to be part of my jury.

I would like to thank Martin Hyland for welcoming me as a research visitor at the DPMMS in Cambridge, where I designed the concurrent game model described in Chapter 4 of this document. I had the privilege to work at the LIX, and I would like to thank my colleagues for making it a wonderful work environment, starting with the extremely helpful administrative staff, Catherine, Isabelle, James, Lydie, Marie-Jeanne and Matthieu, my other colleagues from the Tarski<sup>+</sup> Church office, Alexis, David, and Vivek, later joined by Alexandre and Ivan. Special thanks go to Alexis whose work was inspirational and who provided useful advice. I also thank my other colleagues at Parsifal, the LIX and beyond, Anne-Laure, Andrew, Bruno, Carlos, Catuscia, David, Denis, Elaine, Florian, François, Frank, Jesus, Joëlle, Kaustuv, Laurent, Lutz, Matteo, Miki, Nicolas, Romain, Samuel, Stefan, Stéphane, Sylvain, Ulrich, Zach, . . . This list is by no means complete.

Last but not least, thanks to my family and friends for being supportive and to Florence who shared the whole adventure with me.



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Preliminaries</b>	<b>3</b>
1.1 First-order classical logic . . . . .	3
1.2 Sequent calculus . . . . .	4
1.3 Linear logic . . . . .	7
1.4 Focalisation . . . . .	9
1.5 Multifocalisation . . . . .	12
1.6 Proof theory and computation . . . . .	13
1.7 Games and logic . . . . .	14
1.8 The neutral approach . . . . .	15
<b>2 Additive neutral games</b>	<b>17</b>
2.1 An additive neutral game . . . . .	17
2.1.1 Hintikka’s additive game for truth . . . . .	17
2.1.2 A neutral presentation . . . . .	20
2.1.3 Computation as dual proof search . . . . .	23
2.1.4 Interaction as proof normalisation . . . . .	24
2.2 Simple games for a fragment of MALL . . . . .	25
<b>3 A sequential neutral game for MALL</b>	<b>29</b>
3.1 Unrestricted multiplicatives . . . . .	29
3.1.1 Incompleteness . . . . .	29
3.1.2 Concurrency and focusing . . . . .	30
3.2 Proof system . . . . .	31
3.3 Basic definitions . . . . .	32
3.4 Neutral expressions . . . . .	32
3.5 Neutral graphs . . . . .	32
3.5.1 Interacting multiplicatives . . . . .	34
3.5.2 A neutral presentation . . . . .	37
3.6 Positions and moves . . . . .	39
3.6.1 Positions . . . . .	39
3.6.2 Micro-moves . . . . .	41
3.6.3 Macro-moves . . . . .	45
3.7 Examples . . . . .	45
3.7.1 Indeterminacy . . . . .	46
3.7.2 Atoms . . . . .	47
3.8 Winning strategies and cut-free proofs . . . . .	48
<b>4 A concurrent neutral game for MALL</b>	<b>55</b>
4.1 Introduction . . . . .	55
4.1.1 Non-uniformity . . . . .	55
4.1.2 Locality and concurrency . . . . .	56
4.1.3 The $[T]$ rule . . . . .	56
4.1.4 Failure . . . . .	57

4.2	Proof system . . . . .	58
4.3	Neutral graphs . . . . .	59
4.4	Subgraphs . . . . .	60
4.5	Micro-moves . . . . .	63
4.6	Macro-moves and positions . . . . .	66
	4.6.1 Definition . . . . .	66
	4.6.2 Macro-moves as phases . . . . .	67
4.7	Strategies . . . . .	71
4.8	Winning strategies and proofs . . . . .	73
	4.8.1 Strategy classes . . . . .	73
	4.8.2 Goal trees . . . . .	73
<b>5</b>	<b>Extensions</b>	<b>75</b>
5.1	Explicit cuts . . . . .	75
	5.1.1 Infinite interactions . . . . .	76
	5.1.2 An alternate cut micro-move . . . . .	78
	5.1.3 Winning strategies and proofs . . . . .	80
5.2	Exponentials . . . . .	81
5.3	First-order quantification . . . . .	82
5.4	Equality . . . . .	84
5.5	Fixed points . . . . .	85
<b>6</b>	<b>Related and future work</b>	<b>87</b>
<b>7</b>	<b>Conclusion</b>	<b>89</b>

# Introduction

There is no absolute, unique notion of truth. In mathematics, we usually think of a true statement as one that can be proved. The late 19th century and the 20th century saw considerable efforts to formalise mathematics, in an attempt to set the bases of sound reasoning that everyone would agree on. As a byproduct of this effort, the development of proof theory provides formal ways to represent mathematical proofs, but also gives the ability to reason about proofs themselves. A proof is a comprehensive collection of arguments establishing the truth of a statement. It is a self-contained object, that can be written once and for all.

However, proofs are not the only way to convince someone of the truth of a statement. Consider two persons called the Player and the Opponent. If the Player wants to convince the Opponent of the truth of a statement, she might opt for an interactive approach by letting the Opponent ask questions and delivering her arguments as needed. Debating is an example of such an interaction, and varies in its rules and degrees of formality. Logicians incorporate elements of game theory to study interaction. In the context of computer science, this line of research provides models of the interaction between a program and the environment it is running in.

Determining the relationship between the static notion of truth defined by proofs, and the dynamic one defined by interaction, is a vast field of research. Paul Lorenzen, who developed dialogue games, even considered games as primitive objects serving as a foundation for logic. A session of interaction, or *play* in the game-theoretic terminology, will typically involve only some of the arguments that can be found in a proof, much like in an examination: the examiner—the Opponent—tests the examinee—the Player—’s knowledge by asking some questions and verifying the answers given by the examinee, but she is not likely to test the whole subject in a single session. The important point is that the examiner has the power to ask any question pertaining to the subject. The only way for the examinee to be absolutely sure to pass the test is to know the whole subject. In the same fashion, a proof can be seen as a *winning strategy* allowing the Player to succeed in every play.

This fundamental asymmetry between the examinee and the examiner, or the Player and the Opponent, is pervasive in mainstream game semantics for logic. If a statement is provable, then the Player is able to defend it against all possible attacks by the Opponent. Conversely, if a statement is not provable, then the Opponent is able to defeat the Player by asking the right questions. In the former case, the Player’s winning strategy can be seen as a proof. In the latter case, nothing much is done with the Opponent’s winning strategy, which is merely seen as a *refutation* of the Player’s arguments. An interesting question is whether the Opponent’s strategy can be seen as a proof of the negation of the Player’s statement. In this thesis, we develop an original *neutral approach* to proof search. In our games, the two players have symmetric roles. Instead of being an examiner and an examinee, they are two parties attempting to prove opposite statements. The winning strategies of either player are seen as proofs of the statement she defends.

Our work is part of the *computation-as-proof-search* tradition. Instead of



considering proofs as preexisting objects that can be seen as winning strategies in the interaction, we see each step of the interaction as a step in two simultaneous exhaustive searches for the proofs of two dual formulae. We will work in the multiplicative and additive fragment of linear logic (MALL), which has two important advantages. Firstly, it is symmetric enough to allow a single process to perform two orthogonal proof searches. Secondly, it is not complete, i.e. the fact that a formula is not provable does not imply that its negation is. In other words, refuting a statement is necessarily different from proving its negation, highlighting the relevance of our approach. The formalism of choice for representing proofs will be *sequent calculus*, as it is the perfect basis for proof search and enjoys the symmetries we require. Moreover, the property of *focalisation* of linear logic will be thoroughly used in this thesis, and we will demonstrate its significance in interaction.

### Outline of the thesis

- Chapter 1 introduces the fundamental notions used in our work. In particular, the reader will find a presentation of sequent calculus, linear logic, and focalisation. We also present briefly the scope of the neutral approach and put it into the perspective of the history of logic and games.
- Chapter 2 develops the neutral approach in the restricted additive setting for which it was first designed by Miller and Saurin [MS06]. The focus is put on the particularities of dual proof search and the properties that will be preserved in the games for MALL.
- Chapter 3 describes a neutral game for MALL which is sequential in nature. The game-theoretic model is simple and the main result establishes the equivalence, for each player, between the provability of the player's statement and the existence of a winning strategy. Some of this material was covered in our publications [DM08, DMS09].
- Chapter 4 introduces a new game model allowing for enough concurrency to establish a correspondence between proofs and winning strategies.
- Chapter 5 investigates a few extensions of our game model. In particular, we discuss the introduction of explicit cut moves in the game.
- Chapters 6 and 7 discuss related and future work and summarise the contributions of the thesis.

# Preliminaries

---

This chapter presents core concepts that will be used throughout this thesis. We introduce linear logic through its sequent calculus, obtained by restricting the structural rules of the sequent calculus LK for classical logic. We also discuss focalisation, perhaps the most outstanding result on linear logic, and one of its extensions, multifocalisation. We finally introduce dialogue games for logic and our neutral approach.

## 1.1 First-order classical logic

Logic manipulates *formulae* which are mathematical statements written in a formal language. The features of a specific language determine the expressiveness of the logic. The more constructions the language allows, the richer the logic will be. Propositional languages allow logics to describe finite behaviour. Expressing finite statements about infinitely many objects requires at least a first-order language. With such a language, it is possible to quantify over the elements of a set, as in “there exists an element of the set such that . . .” or “for all elements of the set, . . .”, which is familiar to any mathematician. Other, higher-order settings exist, in which it is possible to quantify over more complex objects such as relations.

A first-order language consists of

- a countable set of *function symbols*, each one being associated with a natural number called its arity,
- a countable set of *variables*,
- a countable set of *predicate symbols*, each one being associated with a natural number called its arity.

Function symbols with arity 0 are also referred to as constant symbols. We will now assume that a first-order language is fixed. Terms are inductively defined as follows. A variable is a term. If  $f$  is a function symbol with arity  $n$ , and  $t_1, \dots, t_n$  are terms, then  $f(t_1, \dots, t_n)$  is a term.

We can now define formulae. Formulae of first-order classical logic are inductively defined by the following grammar.

$$F, F' ::= P(\vec{t}) \mid F \vee F' \mid \perp \mid F \wedge F' \mid \top \mid F \rightarrow F' \mid \forall x F \mid \exists x F \mid \neg F$$

where  $P$  is a predicate symbol of arity  $n$ ,  $\vec{t} = t_1, \dots, t_n$  is a list of  $n$  terms, and  $x$  is a variable. A formula of the form  $P(\vec{t})$  is an atom.  $\vee$ ,  $\perp$ ,  $\wedge$ ,  $\top$ ,  $\rightarrow$ ,  $\forall$ ,  $\exists$ , and  $\neg$  respectively represent disjunction, falsehood, conjunction, truth, implication, universal quantification, existential quantification, and negation.  $F \equiv G$  is an abbreviation for  $(F \rightarrow G) \wedge (G \rightarrow F)$ .

## 1.2 Sequent calculus

Proof theory is the formal study of mathematical proofs. There are many ways to represent proofs formally, and the one we are particularly interested in is sequent calculus. Sequent calculus was introduced by Gentzen [Gen69] as a proof system for classical logic named LK, and has been adapted to other logics since then. One of the strengths of this formalism is its symmetry.

A sequent is an expression of the form  $x_1, \dots, x_m; F_1, \dots, F_n \vdash G_1, \dots, G_p$ , where  $x_1, \dots, x_m$  is a list of distinct variables called the signature and  $F_1, \dots, F_n$  and  $G_1, \dots, G_p$  are formulae. The sequent  $x_1, \dots, x_m; F_1, \dots, F_n \vdash G_1, \dots, G_p$  is a formal way of expressing the statement  $\forall x_1 \dots \forall x_m F_1 \wedge \dots \wedge F_n \rightarrow G_1 \vee \dots \vee G_p$ , i.e. that the disjunction of the  $G_i$  holds under the assumption of the conjunction of the  $F_i$ . In sequent calculus, reasoning is expressed by *inference rules* of the form

$$\frac{P_1 \quad \dots \quad P_n}{C}$$

where  $P_1, \dots, P_n$  are sequents called the premises of the rule and  $C$  is a sequent called the conclusion of the rule. An inference rule formalises a step of reasoning in a proof, expressing that the conclusion follows from the premises. Figure 1.1 gives the rules of LK.  $\Sigma$  stands for a list of variables,  $\Gamma$  and  $\Delta$  for lists of formulae,  $x$  for a variable, and  $F, F_1$  and  $F_2$  for formulae. Notice that the signature of a sequent is almost never modified by a rule. As a result it is usually omitted.

A proof is built by combining inference rules in a tree structure. For example, we can formally prove in LK the syllogism “all humans are mortal; Socrates is human; therefore Socrates is mortal”. We encode this syllogism as the formula  $(\forall x (H(x) \rightarrow M(x))) \wedge H(s) \rightarrow M(s)$ . The proof is

$$\frac{\frac{\frac{\frac{H(s) \vdash H(s)}{\text{init}} \quad \frac{M(s) \vdash M(s)}{\text{init}}}{H(s) \rightarrow M(s), H(s) \vdash M(s)} \rightarrow\text{L}}{\forall x (H(x) \rightarrow M(x)), H(s) \vdash M(s)} \forall\text{L}}{\frac{\forall x (H(x) \rightarrow M(x)), (\forall x (H(x) \rightarrow M(x))) \wedge H(s) \vdash M(s)}{\wedge\text{L}_2}}{\frac{(\forall x (H(x) \rightarrow M(x))) \wedge H(s), (\forall x (H(x) \rightarrow M(x))) \wedge H(s) \vdash M(s)}{\wedge\text{L}_1}}{\frac{(\forall x (H(x) \rightarrow M(x))) \wedge H(s) \vdash M(s)}{\text{CL}}} \rightarrow\text{R}}{\vdash (\forall x (H(x) \rightarrow M(x))) \wedge H(s) \rightarrow M(s)}$$

in which we omitted the XL and XR rules for the sake of brevity.

The sequent at the bottom of a proof is called its conclusion. The notion of derivation generalises the notion of proof. In a proof, all the sequents at the leaves of the tree are the conclusions of rules with no premises. In a derivation, such a leaf is said to be closed. An open leaf is simply a sequent which is not the conclusion of any rule. The collection of all the open leaves is called the frontier of the derivation. A derivation thus expresses that its conclusion follows from its frontier.

Inference rules are classified into groups. The identity group consists of the rules that identify two formulae: the init and cut rules. The init rule expresses a fundamental axiom of logic: a formula implies itself. The cut rule allows to introduce lemmas in a proof. It combines two proofs, one of them having the lemma as a conclusion, the other one having the lemma as a hypothesis. The logical rules express the logical meaning of the connectives. Each logical

**Identity**

$$\frac{}{\Sigma; F \vdash F} \text{init} \quad \frac{\Sigma; \Gamma_1 \vdash \Delta_1, F \quad \Sigma; \Gamma_2, F \vdash \Delta_2}{\Sigma; \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \text{cut}$$

**Logical rules**

$$\frac{\Sigma; \Gamma, F_1 \vdash \Delta \quad \Sigma; \Gamma, F_2 \vdash \Delta}{\Sigma; \Gamma, F_1 \vee F_2 \vdash \Delta} \vee\text{L} \quad \frac{\Sigma; \Gamma \vdash \Delta, F_1}{\Sigma; \Gamma \vdash \Delta, F_1 \vee F_2} \vee\text{R}_1 \quad \frac{\Sigma; \Gamma \vdash \Delta, F_2}{\Sigma; \Gamma \vdash \Delta, F_1 \vee F_2} \vee\text{R}_2$$

$$\frac{\Sigma; \Gamma \vdash \Delta}{\Sigma; \Gamma, \top \vdash \Delta} \top\text{L} \quad \frac{}{\Sigma; \Gamma \vdash \Delta, \top} \top\text{R}$$

$$\frac{\Sigma; \Gamma, F_1 \vdash \Delta}{\Sigma; \Gamma, F_1 \wedge F_2 \vdash \Delta} \wedge\text{L}_1 \quad \frac{\Sigma; \Gamma, F_2 \vdash \Delta}{\Sigma; \Gamma, F_1 \wedge F_2 \vdash \Delta} \wedge\text{L}_2 \quad \frac{\Sigma; \Gamma \vdash \Delta, F_1 \quad \Sigma; \Gamma \vdash \Delta, F_2}{\Sigma; \Gamma \vdash \Delta, F_1 \wedge F_2} \wedge\text{R}$$

$$\frac{}{\Sigma; \Gamma, \perp \vdash \Delta} \perp\text{L} \quad \frac{\Sigma; \Gamma \vdash \Delta}{\Sigma; \Gamma \vdash \Delta, \perp} \perp\text{R}$$

$$\frac{\Sigma; \Gamma_1 \vdash \Delta_1, F_1 \quad \Sigma; \Gamma_2, F_2 \vdash \Delta_2}{\Sigma; \Gamma_1, \Gamma_2, F_1 \rightarrow F_2 \vdash \Delta_1, \Delta_2} \rightarrow\text{L} \quad \frac{\Sigma; \Gamma, F_1 \vdash \Delta, F_2}{\Sigma; \Gamma \vdash \Delta, F_1 \rightarrow F_2} \rightarrow\text{R}$$

$$\frac{\Sigma, x; \Gamma, F \vdash \Delta}{\Sigma; \Gamma, \exists x F \vdash \Delta} \exists\text{L} \quad \frac{\Sigma; \Gamma \vdash \Delta, F\{t/x\}}{\Sigma; \Gamma \vdash \Delta, \exists x F} \exists\text{R}$$

$$\frac{\Sigma; \Gamma, F\{t/x\} \vdash \Delta}{\Sigma; \Gamma, \forall x F \vdash \Delta} \forall\text{L} \quad \frac{\Sigma, x; \Gamma \vdash \Delta, F}{\Sigma; \Gamma \vdash \Delta, \forall x F} \forall\text{R}$$

$$\frac{\Sigma; \Gamma \vdash \Delta, F}{\Sigma; \Gamma, \neg F \vdash \Delta} \neg\text{L} \quad \frac{\Sigma; \Gamma, F \vdash \Delta}{\Sigma; \Gamma \vdash \Delta, \neg F} \neg\text{R}$$

**Structural rules**

$$\frac{\Sigma; \Gamma \vdash \Delta}{\Sigma; \Gamma, F \vdash \Delta} \text{WL} \quad \frac{\Sigma; \Gamma \vdash \Delta}{\Sigma; \Gamma \vdash \Delta, F} \text{WR}$$

$$\frac{\Sigma; \Gamma, F, F \vdash \Delta}{\Sigma; \Gamma, F \vdash \Delta} \text{CL} \quad \frac{\Sigma; \Gamma \vdash \Delta, F, F}{\Sigma; \Gamma \vdash \Delta, F} \text{CR}$$

$$\frac{\Sigma; \Gamma_1, F_2, F_1, \Gamma_2 \vdash \Delta}{\Sigma; \Gamma_1, F_1, F_2, \Gamma_2 \vdash \Delta} \text{XL} \quad \frac{\Sigma; \Gamma \vdash \Delta_1, F_2, F_1, \Delta_2}{\Sigma; \Gamma \vdash \Delta_1, F_1, F_2, \Delta_2} \text{XR}$$

In the  $\forall\text{L}$  and  $\exists\text{R}$  rules,  $t$  is a term containing variables from  $\Sigma$  only. In the  $\exists\text{L}$  and  $\forall\text{R}$  rules,  $x$  is not free in the conclusion.

Figure 1.1: Sequent calculus for classical logic.

rule deals with the main connective of a specific formula, called the *principal formula*, leaving the other formulae of the sequent, which form the *context*, unchanged. Each inference rule is named after that connective. The suffixes “L” and “R” after the name of an inference rule mean that the principal formula is respectively on the left and on the right of the  $\vdash$  symbol. The structural rules have no logical content, but express properties of the sequents themselves. In particular, they imply that in a sequent  $\Gamma \vdash \Delta$ , the order and multiplicities of the formulae in  $\Gamma$  and  $\Delta$  do not matter. In other words that  $\Gamma$  and  $\Delta$  could be seen as sets.

LK is highly symmetric. For each one of the pairs of connectives  $\vee/\wedge$ ,  $\perp/\top$ , and  $\exists/\forall$ , the right rule of a connective behaves like the left rule of the other. This is the well-known De Morgan duality. The following equivalences are provable.

$$\begin{array}{ll} \neg(F_1 \vee F_2) \equiv \neg F_1 \wedge \neg F_2 & \neg(F_1 \wedge F_2) \equiv \neg F_1 \vee \neg F_2 \\ \neg\perp \equiv \top & \neg\top \equiv \perp \\ \neg\exists xF \equiv \forall x\neg F & \neg\forall xF \equiv \exists x\neg F \\ F_1 \rightarrow F_2 \equiv \neg F_1 \vee F_2 & \neg\neg F \equiv F \end{array}$$

It is therefore possible to put formulae in negation normal form where negation is applied to atoms only, and where there is no implication. An atom or the negation of an atom is called a *literal*.

The fundamental property of LK is cut admissibility. An inference rule is admissible when removing this rule from the proof system does not change the set of provable sequents.

**Theorem 1.2.1** (Cut admissibility). *The cut rule of LK is admissible.*

Although Gentzen originally stated this result in this way, he proved a stronger result: every LK proof can be effectively transformed into a cut-free proof. This procedure pushes cuts higher in a proof. For example pushing a cut through a disjunction is done as follows:

$$\frac{\frac{\frac{\Gamma_1 \vdash \Delta_1, F_i}{\Gamma_1 \vdash \Delta_1, F_1 \vee F_2} \vee R_i \quad \frac{\frac{\frac{\Gamma_2, F_1 \vdash \Delta_2}{\Gamma_2, F_1 \vee F_2 \vdash \Delta_2} \vee L \quad \frac{\Gamma_2, F_2 \vdash \Delta_2}{\Gamma_2, F_1 \vee F_2 \vdash \Delta_2} \vee L}{\Gamma_2, F_1 \vee F_2 \vdash \Delta_2} \text{cut}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \text{cut}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \rightsquigarrow \frac{\frac{\Gamma_1 \vdash \Delta_1, F_i}{\Gamma_1 \vdash \Delta_1, F_i} \text{cut} \quad \frac{\Gamma_2, F_i \vdash \Delta_2}{\Gamma_2, F_i \vdash \Delta_2} \text{cut}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \text{cut}$$

The cut elimination procedure gives a computational interpretation of sequent calculus. In particular, the Curry-Howard isomorphism sees proofs as programs, and cut elimination corresponds to program execution.

An immediate corollary is the consistency of the logic. A logic is inconsistent when two dual formulae  $F$  and  $\neg F$  are provable. If classical logic were inconsistent, then the empty sequent would be provable by the cut rule

$$\frac{\vdash F \quad \vdash \neg F}{\vdash} \text{cut}$$

and, by the admissibility of cut, there would be a cut-free proof of the empty sequent. A case analysis shows that there is no cut-free proof of any sequent of the form  $\vdash \perp, \dots, \perp$ .

Cut admissibility makes cut-free proofs objects worthy of consideration. Provability is preserved when eliminating cut. Cut-free derivations have a useful property.

**Proposition 1.2.2.** *In a cut-free derivation, every formula is a subformula of a formula occurring in the conclusion.*

Here,  $F$  is a subformula of  $G$  if it is a subtree of an instance  $G\sigma$  of  $G$ . This result holds because the cut rule is the only one to introduce brand new formulae (reading derivations bottom-up). With this result we recast cut-free sequent calculus by replacing the notion of formula with the notion of formula *occurrence*, which is formally a location in the syntactic tree of a formula. For example, the formula  $F \vee F$  has two distinct occurrences of the subformula  $F$ .

Another useful property of LK is that the init rule can be restricted to atoms.

**Proposition 1.2.3.** *Restricting the init rule to the atomic case does not affect provability.*

Later on, we will consider a logic without atoms. The init rule will then be admissible.

### 1.3 Linear logic

Linear logic was introduced by Girard [Gir87]. It emerged from a semantic study of polymorphic  $\lambda$ -calculus. We choose a syntactic approach by introducing it through its sequent calculus LL, obtained by restricting the structural rules of LK.

The weakening (WL and WR) and contraction (CL and CR) rules of LK express that one can always weaken a statement by requiring an additional hypothesis or allowing an additional conclusion, and that multiple occurrences of a formula are as good as one. In a sequent  $\Gamma \vdash \Delta$ , the lists  $\Gamma$  and  $\Delta$  can be considered as sets. Linear logic is obtained by removing the weakening and contraction rules. In our presentation we also internalise the XL and XR rules. In other words,  $\Gamma$  and  $\Delta$  are now multisets of formulae.

One of the interesting consequences is that previously interchangeable sets of inference rules for the logical connectives are not equivalent any more. For example we chose the following right rules for disjunction in LK:

$$\frac{\Gamma \vdash \Delta, F_1}{\Gamma \vdash \Delta, F_1 \vee F_2} \vee R_1 \quad \frac{\Gamma \vdash \Delta, F_2}{\Gamma \vdash \Delta, F_1 \vee F_2} \vee R_2$$

but we could have chosen the following rule instead:

$$\frac{\Gamma \vdash \Delta, F_1, F_2}{\Gamma \vdash \Delta, F_1 \vee F_2} \vee R$$

Those two presentations can be proved equivalent only with the structural rules. Therefore we obtain two distinct disjunctions in linear logic, respectively denoted to by  $\oplus$  and  $\wp$ . Similarly, we get two conjunctions  $\&$  and  $\otimes$ :

$$\frac{\Gamma \vdash \Delta, F_1 \quad \Gamma \vdash \Delta, F_2}{\Gamma \vdash \Delta, F_1 \& F_2} \& \quad \frac{\Gamma_1 \vdash \Delta_1, F_1 \quad \Gamma_2 \vdash \Delta_2, F_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, F_1 \otimes F_2} \otimes$$

The same goes for the units  $\perp$  and  $\top$ , which become four units:  $0$  for  $\oplus$ ,  $\perp$  for  $\wp$ ,  $\top$  for  $\&$ , and  $1$  for  $\otimes$ . In that context, the De Morgan dual pairs are  $\oplus/\&$ ,  $0/\top$ ,  $\otimes/\wp$ , and  $1/\perp$ .  $\oplus$ ,  $\&$ ,  $0$ , and  $\top$  make up the additive fragment of linear logic (ALL).  $\otimes$ ,  $\wp$ ,  $1$ , and  $\perp$  make up the multiplicative fragment of linear logic (MLL). All eight of them make up the multiplicative and additive fragment of linear logic (MALL).

We will mostly be interested in MALL, which has a highly symmetric sequent calculus well suited to our approach. Its provability is decidable, more precisely PSPACE-complete [LMSS92, Kan94]. However, we present the full first-order linear logic here. The formulae of first-order linear logic are inductively defined by the following grammar.

$$F, F' ::= P(\vec{t}) \mid P(\vec{t})^\perp \mid F \oplus F' \mid 0 \mid F \& F' \mid \top \mid F \otimes F' \mid 1 \mid F \wp F' \mid \perp \\ \mid !F \mid ?F \mid \forall xF \mid \exists xF$$

where  $P$  is a predicate symbol of arity  $n$ ,  $\vec{t} = t_1, \dots, t_n$  is a list of  $n$  terms, and  $x$  is a variable. A formula of the form  $P(\vec{t})$  is an atom.

In addition to the MALL connectives and first-order quantifiers, linear logic contains two connectives  $!$  and  $?$  called exponentials. Girard included them to recover the weakening and contraction rules in a controlled way. This allows intuitionistic logic to be encoded into linear logic, for example.

We did not include negation and implication as actual connectives. Like in LK, the ability to put formulae in negation normal form makes them superfluous. In the grammar,  $P(\vec{t})^\perp$  is a negated atom. Negation is an involution, inductively defined as

$$\begin{aligned} (F_1 \oplus F_2)^\perp &= F_1^\perp \& F_2^\perp & (F_1 \& F_2)^\perp &= F_1^\perp \oplus F_2^\perp \\ 0^\perp &= \top & \top^\perp &= 0 \\ (F_1 \otimes F_2)^\perp &= F_1^\perp \wp F_2^\perp & (F_1 \wp F_2)^\perp &= F_1^\perp \otimes F_2^\perp \\ 1^\perp &= \perp & \perp^\perp &= 1 \\ (!F)^\perp &= ?F^\perp & (?F)^\perp &= !F^\perp \\ (\exists xF)^\perp &= \forall xF^\perp & (\forall xF)^\perp &= \exists xF^\perp \end{aligned}$$

Linear implication  $F_1 \multimap F_2$  is defined as  $F_1^\perp \wp F_2$ .

The sequent calculus for linear logic, LL, is presented in Figure 1.2. Due to the perfect symmetry of left and right rules, we opted for the mono-sided presentation, i.e. with sequents of the form  $\vdash \Gamma$ . A sequent  $\Gamma \vdash \Delta$  is replaced with  $\vdash \Gamma^\perp, \Delta$ .

Many properties of LK are preserved in LL, including Theorem 1.2.1 and Propositions 1.2.2 and 1.2.3. The cut elimination step for additives is of particular interest. When pushing a cut through a pair of dual formulae  $F_1 \& F_2 / F_1^\perp \oplus F_2^\perp$

$$\frac{\frac{\frac{\pi_1}{\vdash \Gamma_1, F_1} \quad \frac{\pi_2}{\vdash \Gamma_1, F_2}}{\vdash \Gamma_1, F_1 \& F_2} \quad \frac{\pi_3}{\vdash \Gamma_2, F_i^\perp}}{\vdash \Gamma_2, F_1^\perp \oplus F_2^\perp} \oplus_i \quad \text{cut}}{\vdash \Gamma_1, \Gamma_2} \rightsquigarrow \frac{\pi_i \quad \pi_3}{\vdash \Gamma_1, F_i \quad \vdash \Gamma_2, F_i^\perp} \text{cut}$$

only one of  $\pi_1$  and  $\pi_2$  is kept. In the context of such an interaction between the  $\oplus$  and  $\&$  connectives, it is natural to isolate one of the two branches of a

**Identity**

$$\frac{}{\vdash F, F^\perp} \text{init} \quad \frac{\vdash \Gamma_1, F \quad \vdash \Gamma_2, F^\perp}{\vdash \Gamma_1, \Gamma_2} \text{cut}$$

**Additives**

$$\frac{\vdash \Gamma, F_1}{\vdash \Gamma, F_1 \oplus F_2} \oplus_1 \quad \frac{\vdash \Gamma, F_2}{\vdash \Gamma, F_1 \oplus F_2} \oplus_2 \quad \frac{\vdash \Gamma, F_1 \quad \vdash \Gamma, F_2}{\vdash \Gamma, F_1 \& F_2} \& \quad \frac{}{\vdash \Gamma, \top} \top$$

**Multiplicatives**

$$\frac{\vdash \Gamma_1, F_1 \quad \vdash \Gamma_2, F_2}{\vdash \Gamma_1, \Gamma_2, F_1 \otimes F_2} \otimes \quad \frac{}{\vdash \perp} \perp \quad \frac{\vdash \Gamma, F_1, F_2}{\vdash \Gamma, F_1 \wp F_2} \wp \quad \frac{\vdash \Gamma}{\vdash \Gamma, \perp} \perp$$

**Quantifiers**

$$\frac{\vdash \Gamma, F\{t/x\}}{\vdash \Gamma, \exists x F} \exists \quad \frac{\vdash \Gamma, F}{\vdash \Gamma, \forall x F} \forall$$

**Exponentials**

$$\frac{\vdash ?\Gamma, F}{\vdash ?\Gamma, !F} ! \quad \frac{\vdash \Gamma, F}{\vdash \Gamma, ?F} ?D \quad \frac{\vdash \Gamma}{\vdash \Gamma, ?F} ?W \quad \frac{\vdash \Gamma, ?F, ?F}{\vdash \Gamma, ?F} ?C$$

Figure 1.2: Mono-sided sequent calculus for linear logic.

& rule in a derivation. Such an operation is called *slicing* a derivation, and the resulting object is an (additive) slice. We will use the following notation for instances of the & rule where a slice is taken:

$$\frac{\vdash \Gamma, F_1}{\vdash \Gamma, F_1 \& F_2} \&_1 \quad \frac{\vdash \Gamma, F_2}{\vdash \Gamma, F_1 \& F_2} \&_2$$

## 1.4 Focalisation

Andreoli [And92] proved a fundamental result about linear logic. With efficient logic programming in mind, Andreoli developed deep insights into the structure of the cut-free proofs in LL in an effort to reduce non-determinism in proof search.

For some inference rules it is the case that the premises are provable if and only if the conclusion is. Those rules are said to be *invertible*. For example the & rule is invertible since each premise  $\vdash \Gamma, F_i$  can be derived from the conclusion  $\vdash \Gamma, F_1 \& F_2$ :

$$\frac{\frac{}{\vdash F_i^\perp, F_i} \text{init}}{\vdash F_1^\perp \oplus F_2^\perp, F_i} \oplus_i}{\vdash \Gamma, F_i} \text{cut}$$

During proof search, all the invertible rules can be applied eagerly without losing completeness. This inspired Andreoli to establish a classification of connectives



of linear logic:

- the connectives  $\&$ ,  $\top$ ,  $\wp$ ,  $\perp$ ,  $?$  and  $\forall$  are called *asynchronous*; their introduction rules are invertible;
- their duals  $\oplus$ ,  $0$ ,  $\otimes$ ,  $1$ ,  $!$ , and  $\exists$  are called *synchronous*.

A formula which is not a literal is synchronous (resp. asynchronous) iff its main connective is. An inference rule introducing a connective is synchronous (resp. asynchronous) iff that connective is.

Andreoli proved that proof search can be forced to follow a specific strategy without losing completeness.

1. As long as the current sequent contains asynchronous formulae, asynchronous rules can be applied, in any order. Those rules are not only invertible, they also commute.
2. When no more asynchronous rules can be applied, one must select a (synchronous) formula and *focus* on it: hereditarily apply synchronous rules to that formula and its descendants, until they become asynchronous again. Then, go back to the previous point.

Those two modes alternate in a proof. The first mode is called an asynchronous phase and the second one a synchronous phase.

Although the restriction to that behaviour does not affect provability, it drastically constrains the shape of proofs. Andreoli extended the classification of synchronous/asynchronous formulae to literals. For each pair of dual literals  $P(\vec{t})/P(\vec{t})^\perp$ , one arbitrarily chooses a *bias* by declaring one of them to be synchronous (or positive) and the other one to be asynchronous (or negative). Again, this choice does not affect the provability of a formula, but it has a major impact on the shape of its proofs and can be used to control it [LM07].

Extending the classification of formulae to literals makes sense. Logically speaking, an atom is a placeholder for a generic formula that will never be inspected. Therefore, one expects a proof of a formula to be roughly unchanged if an atom is uniformly replaced with an arbitrary formula. With focalisation, the synchrony of a formula determines the shape of a proof. Thanks to the classification of atoms, a focused proof can be trivially adapted if an atom is uniformly replaced with a formula, as long as that formula and that atom have the same synchrony.

For any pair of dual formulae  $F/F^\perp$ ,  $F^+$  will denote the synchronous (or positive) one, and  $F^-$  will denote the asynchronous (or negative) one. Figure 1.3 shows the focused proof system for linear logic. Sequents are of one of the two forms  $\vdash \Theta : \Gamma \Downarrow F$  or  $\vdash \Theta : \Gamma \Uparrow \Delta$ . Those sequents should be read in the unfocused sequent calculus as  $\vdash ?\Theta, \Gamma, F$  and  $\vdash ?\Theta, \Gamma, \Delta$  respectively.  $\Theta$  is a set of formulae,  $\Gamma$  is a multiset of positive formulae and negative literals,  $F$  is a formula, and  $\Delta$  is a multiset of formulae. A sequent of the form  $\vdash \Theta : \Gamma \Downarrow F$  is in a synchronous phase, and focuses on the formula  $F$ . A sequent of the form  $\vdash \Theta : \Gamma \Uparrow \Delta$  is in an asynchronous phase, and  $\Delta$  is the multiset of the potentially asynchronous formulae that need to be taken care of before the end of the phase.

The structure of focused proofs suggests abstracting away from the details of the phases. We may consider synchronous (resp. asynchronous) phases as large synthetic inference rules [Cur05] introducing a full layer of synchronous (resp. asynchronous) connectives. In particular, we will identify two focused proofs which only differ by a permutation of inference rules within a phase. It should be pointed out that although the order in which individual rules are

**Synchronous rules**

$$\begin{array}{c}
\frac{}{\vdash \Theta : L^- \Downarrow L^+} [\text{init}_1] \quad \frac{}{\vdash \Theta, L^- \Downarrow L^+} [\text{init}_2] \\
\frac{\vdash \Theta : \Gamma \Downarrow F_1}{\vdash \Theta : \Gamma \Downarrow F_1 \oplus F_2} [\oplus_1] \quad \frac{\vdash \Theta : \Gamma \Downarrow F_2}{\vdash \Theta : \Gamma \Downarrow F_1 \oplus F_2} [\oplus_2] \\
\frac{\vdash \Theta : \Gamma_1 \Downarrow F_1 \quad \vdash \Theta : \Gamma_2 \Downarrow F_2}{\vdash \Theta : \Gamma_1, \Gamma_2 \Downarrow F_1 \otimes F_2} [\otimes] \quad \frac{}{\vdash \Theta : \Downarrow 1} [1] \\
\frac{\vdash \Theta : \Gamma \Downarrow F\{t/x\}}{\vdash \Theta : \Gamma \Downarrow \exists x F} [\exists] \quad \frac{\vdash \Theta : \Uparrow F}{\vdash \Theta : \Downarrow !F} [!]
\end{array}$$

**Asynchronous rules**

$$\begin{array}{c}
\frac{\vdash \Theta : \Gamma \Uparrow F_1, \Delta \quad \vdash \Theta : \Gamma \Uparrow F_2, \Delta}{\vdash \Theta : \Gamma \Uparrow F_1 \& F_2, \Delta} [\&] \quad \frac{}{\vdash \Theta : \Gamma \Uparrow \top, \Delta} [\top] \\
\frac{\vdash \Theta : \Gamma \Uparrow F_1, F_2, \Delta}{\vdash \Theta : \Gamma \Uparrow F_1 \wp F_2, \Delta} [\wp] \quad \frac{\vdash \Theta : \Gamma \Uparrow \Delta}{\vdash \Theta : \Gamma \Uparrow \perp, \Delta} [\perp] \\
\frac{\vdash \Theta : \Gamma \Uparrow F, \Delta}{\vdash \Theta : \Gamma \Uparrow \forall x F, \Delta} [\forall] \quad \frac{\vdash \Theta, F : \Gamma \Uparrow \Delta}{\vdash \Theta : \Gamma \Uparrow ?F, \Delta} [?]
\end{array}$$

**Phase changes**

$$\begin{array}{c}
\frac{\vdash \Theta : \Gamma \Uparrow F^-}{\vdash \Theta : \Gamma \Downarrow F^-} [\text{R}\Downarrow] \quad \frac{\vdash \Theta : G, \Gamma \Uparrow \Delta}{\vdash \Theta : \Gamma \Uparrow G, \Delta} [\text{R}\Uparrow] \\
\frac{\vdash \Theta : \Gamma \Downarrow F^+}{\vdash \Theta : \Gamma, F^+ \Uparrow} [\text{D}_1] \quad \frac{\vdash \Theta, F^+ : \Gamma \Downarrow F^+}{\vdash \Theta, F^+ : \Gamma \Uparrow} [\text{D}_2]
\end{array}$$

$F^+$  (resp.  $F^-$ ) stands for a positive (resp. negative) formula.  $L^+$  (resp.  $L^-$ ) stands for a positive (resp. negative) literal. In  $[\text{R}\Uparrow]$ ,  $G$  is either a positive formula or a negative literal.

Figure 1.3: Focalisation for linear logic.

applied is nondeterministic, there is only one asynchronous phase with a given conclusion. In other words, all the non-deterministic choices are grouped in synchronous phases, which does not come as a surprise since asynchronous rules are invertible. As we shall see in this thesis, focalisation is adapted to an interactive interpretation: a synchronous phase corresponds to choices that we make, and an asynchronous phase to choices that an opponent makes.

As we will mostly be interested in MALL, the proof systems we will consider will be simpler. The non-linear  $\Theta$  zone is not needed in the absence of exponentials.

## 1.5 Multifocalisation

Although focalisation provides an abstraction from individual inference rules, it forces the decomposition of synchronous formulae to be sequentialised. The  $[D_1]$  and  $[D_2]$  rules start a synchronous phase by picking a formula as focus, and the other synchronous formulae have to wait for their turn. Consider the two following MLL focused proofs.

$$\begin{array}{c}
\frac{}{\vdash L^- \Downarrow L^+} \text{[init}_1\text{]} \\
\frac{}{\vdash L^+, L^- \Uparrow} \text{[D}_1\text{]} \\
\frac{}{\vdash L^+ \Uparrow L^- \wp \perp} \text{[}\wp, \text{R}\uparrow, \perp\text{]} \\
\frac{}{\vdash L^+ \Downarrow (L^- \wp \perp) \otimes 1} \text{[}\otimes, \text{R}\downarrow, 1\text{]} \\
\frac{}{\vdash (L^- \wp \perp) \otimes 1, L^+ \Uparrow} \text{[D}_1\text{]} \\
\frac{}{\vdash (L^- \wp \perp) \otimes 1 \Uparrow L^+ \wp \perp} \text{[}\wp, \text{R}\uparrow, \perp\text{]} \\
\frac{}{\vdash (L^- \wp \perp) \otimes 1 \Downarrow (L^+ \wp \perp) \otimes 1} \text{[}\otimes, \text{R}\downarrow, 1\text{]} \\
\frac{}{\vdash (L^+ \wp \perp) \otimes 1, (L^- \wp \perp) \otimes 1 \Uparrow} \text{[D}_1\text{]}
\end{array}
\qquad
\begin{array}{c}
\frac{}{\vdash L^- \Downarrow L^+} \\
\frac{}{\vdash L^-, L^+ \Uparrow} \\
\frac{}{\vdash L^- \Uparrow L^+ \wp \perp} \\
\frac{}{\vdash L^- \Downarrow (L^+ \wp \perp) \otimes 1} \\
\frac{}{\vdash (L^+ \wp \perp) \otimes 1, L^- \Uparrow} \\
\frac{}{\vdash (L^+ \wp \perp) \otimes 1 \Uparrow L^- \wp \perp} \\
\frac{}{\vdash (L^+ \wp \perp) \otimes 1 \Downarrow (L^- \wp \perp) \otimes 1} \\
\frac{}{\vdash (L^+ \wp \perp) \otimes 1, (L^- \wp \perp) \otimes 1 \Uparrow}
\end{array}$$

Those proofs only differ by the order in which the phases are scheduled. Chaudhuri, Miller and Saurin propose in [CMS08] to extend focalisation by allowing several foci to be selected for a synchronous phase, a feature known as *multifocalisation*. They introduce a sequent calculus for MALL (see Figure 1.4). Saurin provides a multifocused sequent calculus for the full logic in [Sau08b]. Multifocused sequent calculus is clearly sound and complete with respect to LL. Soundness is seen by removing focusing annotations from sequents, and completeness follows from the completeness of single focalisation.

In addition to the two sequentialised proofs above, multifocusing allows a proof in which the synchronous formulae are decomposed concurrently:

$$\begin{array}{c}
\frac{}{\vdash L^- \Downarrow L^+} \text{[init]} \\
\frac{}{\vdash L^+, L^- \Uparrow} \text{[D]} \\
\frac{}{\vdash \Uparrow L^+ \wp \perp, L^- \wp \perp} \text{[}\wp, \text{R}\uparrow, \perp\text{]} \\
\frac{}{\vdash \Downarrow (L^+ \wp \perp) \otimes 1, (L^- \wp \perp) \otimes 1} \text{[}\otimes, \text{R}\downarrow, 1\text{]} \\
\frac{}{\vdash (L^+ \wp \perp) \otimes 1, (L^- \wp \perp) \otimes 1 \Uparrow} \text{[D]}
\end{array}$$

The authors of [CMS08] show that maximally multifocused MLL proofs correspond to MLL proof nets. We will not cover this topic here, but we will

**Additives**

$$\frac{\vdash \Gamma \Downarrow F_i, \Delta}{\vdash \Gamma \Downarrow F_1 \oplus F_2, \Delta} [\oplus_i] \quad \frac{\vdash \Gamma \Uparrow F_1, \Delta \quad \vdash \Gamma \Uparrow F_2, \Delta}{\vdash \Gamma \Uparrow F_1 \& F_2, \Delta} [\&] \quad \frac{}{\vdash \Gamma \Uparrow \top, \Delta} [\top]$$

**Multiplicatives**

$$\frac{\vdash \Gamma_1 \Downarrow F_1, \Delta_1 \quad \vdash \Gamma_2 \Downarrow F_2, \Delta_2}{\vdash \Gamma_1, \Gamma_2 \Downarrow F_1 \otimes F_2, \Delta_1, \Delta_2} [\otimes] \quad \frac{}{\vdash \Downarrow 1} [1]$$

$$\frac{\vdash \Gamma \Uparrow F_1, F_2, \Delta}{\vdash \Gamma \Uparrow F_1 \wp F_2, \Delta} [\wp] \quad \frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \perp, \Delta} [\perp]$$

**Identity**

$$\frac{}{\vdash L^- \Downarrow L^+} [\text{init}]$$

**Phase changes**

$$\frac{\vdash \Gamma \Uparrow \Delta^-}{\vdash \Gamma \Downarrow \Delta^-} [\text{R}\Downarrow] \quad \frac{\vdash G, \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow G, \Delta} [\text{R}\Uparrow] \quad \frac{\vdash \Gamma \Downarrow \Delta^+}{\vdash \Gamma, \Delta^+ \Uparrow} [\text{D}\Downarrow]$$

In  $[\text{R}\Uparrow]$ ,  $G$  is either a positive formula or a negative literal. In  $[\text{D}\Downarrow]$ ,  $\Delta^+$  is not empty.

Figure 1.4: Multifocalisation for MALL.

use multifocalisation as a way to improve the concurrency of focused sequent calculus.

## 1.6 Proof theory and computation

In the general context of the study of computation, approaches to proof theory traditionally fall into two categories.

In the *computation-as-proof-normalisation* approach, proofs model programs in functional languages. The computational content of this model is seen in the correspondence between proof normalisation and program execution. In that setting, a formula is a type, and a proof is a term of that type. A formula is true iff its type is not empty. Implication corresponds to the “arrow” constructor for types and plays a fundamental role. Proofs of  $A \rightarrow B$  are functions mapping proofs of  $A$  to proofs of  $B$ . Proof normalisation corresponds to  $\beta$ -reduction in  $\lambda$ -calculus.

In the *computation-as-proof-search* approach, the state of the computation is seen as a sequent, and the computation itself as the search for a cut-free proof of that sequent. This is the basis for logic programming [MNPS91]. In that tradition, the specific features of a proof system are used to model proof search strategies. For example, the bias assignment to the atoms in a focused proof system can force forward-chaining or back-chaining [LM07].

## 1.7 Games and logic

The connections between logic and games have a long and rich history, although they remained informal until the 1950s. It is known since Aristotle that a debate can be seen as a game. The twentieth century saw the development of game theory, and connections were then made to logic. From the point of view of game theorists in general, the kinds of games studied in logic are quite specific. They typically involve two players with distinct roles, are not concerned with probabilities, and usually have two outcomes: a win for a player or for the opponent.

In a typical game for logic, the two players, called the Player and the Opponent, or Éloïse and Abélard, build a sequence of *moves* called a *play*. For each finite sequence of moves, the rules of the game specify which player shall be the one to extend the sequence with a move, and which moves are at her disposal for doing so. Plays may be infinite in general, but there may also be some finite sequences of moves that cannot be extended with additional moves. In either case, the rules of the game classify maximal plays into those won by Éloïse and those won by Abélard. The plays are the uni-dimensional objects of the games. There are also bi-dimensional objects of interest called *strategies*. Strategies represent move policies that players may follow. A typical strategy is simply a set of plays that a player is prepared to follow. An important question about the game is whether a specific player can play according to some strategy which ensures her victory. If the answer is positive, then such strategies, called *winning* strategies, are formal witnesses to this fact. Similarly, proofs are witnesses to the truth of a formula. The games we are interested in tend to make this connection between games/formulae, provability/the existence of a winning strategy, proofs/winning strategies, and computation/dynamics of interaction.

Dialogue games were formalised by Lorenzen [Lor61]. In those games, the Player tries to establish the truth of a formula while the Opponent tries to establish its falsehood. Informally, the Opponent plays the role of an examiner and the Player plays the role of an examinee. For example, if the formula is a conjunction, the Opponent chooses a conjunct and challenges the Player to prove it; if it is a disjunction, the Player chooses which disjunct to prove.

Lorenzen's work inspired mainstream game semantics in the computation-as-proof-normalisation tradition. The main idea is to interpret proofs as strategies, and to define an operation of composition of two strategies corresponding to the cut of the two proofs. This line of work began with Blass' game semantics for linear logic [Bla92]. Major advances allowed to solve long-standing problems such as the full-abstraction problem for the language PCF [AJM00, HO00]. The objective was to establish a correspondence between strategies and programs, and this required controlling the information available to the players to choose moves. Hyland and Ong developed the adequate notion of innocent strategy to this end. These approaches also provided models for logics capturing the dynamics of cut elimination [AJ94, AM99]. Another major problem investigated in this line of research was the ability to represent concurrent programs in games. Abramsky and Melliès developed concurrent games [AM99] in which moves are performed concurrently in maximal chunks. Melliès also developed asynchronous games in which the traces of computation (sequences of moves) are equipped with a homotopy relation giving a geometric reading of concurrency. In this framework, it is possible to see innocence as positionality [Mel04].

Game-theoretical studies of logic programming are less common. Some of them model Prolog engines, beginning with [vE86] which connected Prolog computations and two-person games using the  $\alpha\beta$ -algorithm. Loddo *et al.* [CLN98, Lod02] further developed this line of research with constraint logic programming [LC00]. Galanaki *et al.* [GRW08] accounted for negation in logic programming. Some other game-theoretic studies are more fundamentally concerned with proof search, as a foundation of logic programming. The games of Pym and Ritter [PR04] model the search for uniform proofs with backtracking. More recently, Saurin [Sau08b, Sau08a] investigated proof search in Ludics [Gir01] guided by interaction with tests.

## 1.8 The neutral approach

The approach we develop in this thesis is inspired by dialogue games, but it clearly lies in the computation-as-proof-search tradition. The major novelty of this work is a change of perspective: instead of considering asymmetric settings in which the Player provides the arguments and the Opponent only aims at refuting them, we adopt a *neutral approach* with players having the same status. Both players attempt to prove a formula and to refute its negation. The computation can be seen as the simultaneous search for two orthogonal proofs, or *dual proof search*.

The neutral approach was first designed by Miller and Saurin in [MS06] in an essentially additive setting (see Chapter 2). A more subtle role assignment than the usual Player/Opponent dichotomy can be motivated by considering a common approach to proving the completeness of first-order classical logic, following, say, Smullyan [Smu95]. Proving completeness can be done by attempting to progressively build a sequent calculus derivation of a formula  $F$ . Each step extends the derivation with inference rules at the open leaves. If this is done in a systematic fashion, all the leaves are eventually closed, or the derivation ends up with a possibly infinite open branch. In the first case, there is a proof of  $F$  and in the second case, the open branch provides a falsifying model of  $F$ . This process can be seen as an interaction where one of the players attempts to complete the proof while the other one looks for an infinite branch. Exactly one of the players succeeds at their task.

This example relates the proof-theoretic notion of provability to the semantic notion of validity. In a decidable logic, models are expected to be finite, and so are the open branches of the derivation of  $F$ . In that case, it may be possible to extract a proof of  $\neg F$  from the counter-model for  $F$  read from an open branch. The interactive process would then be seen as two players building derivations of opposite formulae. However, models and proofs are objects which are different in nature and, even though the development of the derivation and of the counter-model of  $F$  are closely related at each step, there is no reason to expect the derivation of  $\neg F$  extracted from the counter-model to be related to the derivation of  $F$ .

The neutral approach solves this issue with a symmetric two-player game in which both players play with the same rules. If a player has a winning strategy, she is able to construct a proof: for one of the players, this would be a proof of  $F$  while dually for the other player, this would be a proof of  $\neg F$ . Each step extends the two orthogonal derivations with an inference rule of sequent calculus.

The core part of this thesis consists in defining neutral games for MALL. MALL has the advantage of having an extremely symmetric sequent calculus, which is required in our approach. The provability for MALL is PSPACE-complete, and MALL is incomplete. That is, there are formulae  $F$  such that neither  $F$  nor  $F^\perp$  are provable; take, for example,  $F = 1 \wp 0$ . It means that the neutral approach is not trivial in that logic: refuting a formula is not the same as proving its negation. Therefore, the game is not determinate: in some cases no player has a winning strategy. Some plays are not won by any player and, as a result, plays need to continue after one of the player has failed; it has yet to be determined whether the other player will win or not.

Chapter 3 presents a neutral game for MALL with a simple model in which the provability of a formula is equivalent to the existence of a winning strategy for the corresponding player. Chapter 4 switches to a radically different game model allowing for some concurrent behaviour, in which proofs are related to winning strategies.

Note that we follow a conventional angle. We aim at finding the suitable notion of game to bring the neutral approach to an existing logic with an existing notion of proof and refutation. In doing so, we stress the deep dualities of the logic, not only concerning proofs themselves, but also concerning the dynamics of proof search. The individual steps of proof search are performed simultaneously in the two orthogonal derivations. In contrast, Ludics [Gir01], while sharing some similarities with our work, takes another direction by radically departing from logic and making interaction the primitive notion.

# Additive neutral games

---

This chapter introduces the neutral approach first developed by Miller and Saurin [MS06], restricting it to an additive setting in which refuting a formula is the same as proving its negation. Starting from a Hintikka-style dialogue game, we introduce a neutral syntax and discuss the remarkable properties that we wish to preserve in the game for MALL. In addition, we briefly revisit the *simple games*, an extension to a fragment of MALL keeping this additive behaviour, yet expressive enough to encode numerous examples.

In the two-player games we consider, the players will be named 0 and 1. The symbol  $\lambda$  will typically denote a player. For every player  $\lambda \in \{0, 1\}$ ,  $\bar{\lambda} = 1 - \lambda$  will denote the opponent.

## 2.1 An additive neutral game

### 2.1.1 Hintikka's additive game for truth

Jaakko Hintikka (see, for example, [Hod04]) presented a two player game to define a notion of truth of a formula of first-order classical logic. One of Hintikka's key observations was that playing on the negation of a formula  $F$  was the same as playing on  $F$  and exchanging the two players.

Two players, Éloïse and Abélard, play on a single formula. Éloïse tries to establish the truth of the formula while Abélard tries to establish its falsehood. In this section, we present a propositional version of this game, using the propositional additive fragment of linear logic (ALL). We recall the syntax of the formulae of this fragment:

$$F, G ::= F \oplus G \mid 0 \mid F \& G \mid \top$$

Hintikka's original motivation was to provide an interactive definition of truth. In this presentation we take an alternate approach by switching to proof-theoretic notions. This distinction might seem shallow at first, since validity and provability are equivalent notions. However, seeing interaction as orthogonal proof search is at the heart of our approach, and applying this methodology to the richer MALL will provide insights into its proof theory.

The sequent calculus for ALL is straightforward:

$$\frac{\vdash E_i}{\vdash E_1 \oplus E_2} \oplus_i \quad \text{for } i \in \{1, 2\} \quad \frac{\vdash E_1 \quad \vdash E_2}{\vdash E_1 \& E_2} \& \quad \frac{}{\vdash \top} \top$$

Formally, the game consists of an arena, which is a directed graph whose vertices, called positions, are the formulae occurrences, and whose arcs are called moves. Each move has an associated player, which can be either Éloïse or Abélard. The following definition lists the moves of each player.



**Definition 2.1.1.** *Éloïse has two moves from every position of the form  $F \oplus G$ , to  $F$  and  $G$  respectively. Abélard has two moves from every position of the form  $F \& G$ , to  $F$  and  $G$  respectively.*

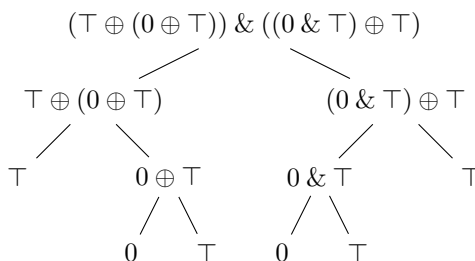
We use the notation  $p \rightarrow p'$  to denote a move from a position  $p$  to a position  $p'$ .

The positions from which there is no move, which are the occurrences of 0 and  $\top$ , are called *terminal*. A *play*  $p$  is a finite sequence  $p_1, \dots, p_n$  of positions such that  $p_{i-1} \rightarrow p_i$  for  $1 < i \leq n$ . A play is *maximal* iff its last position is terminal. A maximal play  $p_1, \dots, p_n$  is won by Éloïse iff  $p_n = \top$ , and by Abélard iff  $p_n = 0$ .

Informally, Éloïse and Abélard discuss the truth of a formula by challenging each other. Each player aims at bringing the other player to a contradiction. Their argument is formally represented by a play, starting from the initial formula, and each move represents a step in the argument. When considering a disjunction  $F \oplus G$ , Éloïse gets to choose a disjunct with which to continue the discussion. Since she tries to establish the truth of  $F \oplus G$ , and since  $F \oplus G$  is true iff  $F$  is or  $G$  is, her task is to point out which one is. Abélard, who tries to establish that  $F \oplus G$  is false, must be able to argue that  $F$  and  $G$  are both false. It seems therefore natural that Éloïse be the one to make the choice between  $F$  and  $G$ : she simultaneously chooses which disjunct she believes to be true, and challenges Abélard to prove her wrong. Dually, when playing on a conjunction, Abélard is the one to make the choice between the conjuncts. When the play reaches a terminal position, representing the undeniable truth ( $\top$ ) or the undeniable falsehood (0), the player who is undeniably right wins the argument.

It is important to understand that there are, in general, many different plays starting from a given position, and that not all of them are won by the same player. In a real-life debate, someone defending a valid point may make some mistakes and end up losing an argument when confronted by a skilled opponent.

**Example 2.1.2.** *The tree of the plays starting at the true formula  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$  is*



*Clearly, some of the maximal ones are won by Éloïse, some others by Abélard.*

As illustrated by this example, a play is a uni-dimensional object which does not contain enough information to decide the truth of a formula. It is, however, a *trace* of a debate which ends with a player failing to prove her point. This does not come as a surprise, since objects establishing the truth of a formula, like proofs, are usually bi-dimensional. In games, a player sometimes has a way to play which ensure her victory, no matter how the opponent chooses to play.

A strategy is an object representing how a player is willing to play. A strategy is winning when the player wins all the terminal plays in which she follows the strategy. Strategies are bi-dimensional objects, formalised here as the set of plays the player is willing to play.

**Definition 2.1.3** (Strategy). *Let  $\lambda$  be a player. A  $\lambda$ -strategy for a position  $p$  is a minimal prefixed-closed set  $\sigma$  of plays such that:*

- $p \in \sigma$ ,
- if  $p_1, \dots, p_n \in \sigma$  and player  $\lambda$  moves at  $p_n$ , then there is exactly one position  $p_{n+1}$  such that  $p_n \rightarrow p_{n+1}$  and the play  $p_1, \dots, p_{n+1}$  is in  $\sigma$ ,
- if  $p_1, \dots, p_n \in \sigma$  and player  $\bar{\lambda}$  moves at  $p_n$ , then for every position  $p_{n+1}$  such that  $p_n \rightarrow p_{n+1}$ , the play  $p_1, \dots, p_{n+1}$  is in  $\sigma$ .

A  $\lambda$ -strategy is winning iff  $\lambda$  wins all its maximal plays.

Here we require strategies to be deterministic (whenever the player has to move, the strategy must specify at most one move) and total (whenever the player has to move, the strategy specifies at least one move; when the opponent has to move, the strategy accepts all possible moves).

In Example 2.1.2, Éloïse has two winning strategies:

- one consisting of the plays  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$ ,  $\top \oplus (0 \oplus \top)$ ,  $\top$  and  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$ ,  $(0 \& \top) \oplus \top$ ,  $\top$ , along with their prefixes;
- one consisting of the plays  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$ ,  $\top \oplus (0 \oplus \top)$ ,  $0 \oplus \top$ ,  $\top$  and  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$ ,  $(0 \& \top) \oplus \top$ ,  $\top$ , along with their prefixes.

Winning strategies allow a player to win against all attacks. The two players cannot both have winning strategies at the same time. Otherwise playing them against each other would yield a play won by the two players, which is impossible. Not surprisingly, having a winning strategy means that the player playing it is correct at establishing the truth.

**Theorem 2.1.4.** *Let  $E$  be a formula.  $E$  is true iff Éloïse has a winning strategy for  $E$ .  $E$  is false iff Abélard has a winning strategy for  $E$*

*Proof.* If  $E = \top$ , Éloïse has a trivial winning strategy. If  $E = 0$ , Éloïse has no winning strategy. If  $E$  is of the form  $F \oplus G$ , Éloïse has a winning strategy iff she has a winning strategy for  $F$  or a winning strategy for  $G$ . If  $E$  is of the form  $F \& G$ , Éloïse has a winning strategy iff she has a winning strategy for  $F$  and a winning strategy for  $G$ . Those cases exactly match the truth semantics of the logic, hence the theorem. The dual result for Abélard is equally obvious.  $\square$

Modelling truth is one of the objectives of the connections between logic and games, but we can do better. Since winning strategies are formal objects establishing the truth of a formula, it is natural to question how they relate to proofs. Depending on the game model and proof system, winning strategies may not correspond to proofs at all, correspond exactly to proofs, or correspond to classes of proofs.

In the case of our additive game, the correspondence is clearly one-to-one.

**Theorem 2.1.5.** *Let  $E$  be a formula. There is a one-to-one correspondence between the cut-free ALL proofs of  $E$  and Éloïse's winning strategies for  $E$ .*

For example, Éloïse's winning strategies for Example 2.1.2 correspond to the two proofs of  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$ :

$$\frac{\frac{\overline{\top} \top}{\vdash \top \oplus (0 \oplus \top)} \oplus_1 \quad \frac{\overline{\top} \top}{\vdash (0 \& \top) \oplus \top} \oplus_2}{\vdash (\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)} \&$$

and

$$\frac{\frac{\overline{\top} \top}{\vdash 0 \oplus \top} \oplus_2 \quad \frac{\overline{\top} \top}{\vdash (0 \& \top) \oplus \top} \oplus_2}{\vdash (\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)} \&$$

**Remark 2.1.6.** *In sequent calculus, the two proofs of  $\vdash \top \oplus \top$  are distinct. We define the positions of the game as formula occurrences instead of plain formulae to make the same distinction. Otherwise the two disjuncts of  $\top \oplus \top$  would be identified and Éloïse would only have one winning strategy for this formula.*

### 2.1.2 A neutral presentation

An important feature of ALL is its completeness. For every formula  $E$ , either  $E$  or  $E^\perp$  is provable. Making this observation explicit in sequent calculus is not easy, but it can be done in the game. The game is symmetric in the sense that disjunction ( $\oplus$  and  $0$ ) is treated in the same way by Éloïse as conjunction ( $\&$  and  $\top$ ) by Abélard. What matters is not the current formula in a play, but who plays. If we negate all formulae in a game, the strategies of Éloïse and Abélard are swapped.

**Corollary 2.1.7.** *Let  $E$  be a formula. There is a one-to-one correspondence between the proofs of  $E^\perp$  and Abélard's winning strategies in  $E$ .*

This is a stronger statement than the mere completeness of the logic: not only is  $E^\perp$  provable if and only if  $E$  is not, but the refutations of  $E$  (witnesses that it is unprovable) are in one-to-one correspondence with the proofs of  $E^\perp$ .

Therefore, the game can be seen as the simultaneous development of two dual derivations, in which Éloïse attempts to derive  $E$  while Abélard attempts to derive  $E^\perp$ . Both derivations are built from the bottom up. Whenever one of the players faces a sequent of the form  $\vdash F_1 \oplus F_2$ , she picks one of the disjuncts  $F_i$  by extending the derivation with the rule  $\oplus_i$ . Simultaneously her opponent faces  $\vdash F_1^\perp \& F_2^\perp$  and, depending on the choice of her opponent, is challenged to prove either  $\vdash F_1^\perp$  or  $\vdash F_2^\perp$ . Even if only one of the conjuncts is picked during a given play, the opponent must be able to accommodate to any of the two choices of the player, and her winning strategies are indeed composed of two separate sub-strategies, exactly like the proofs of  $\vdash F_1^\perp \& F_2^\perp$ . Whenever one of the players faces  $\vdash \top$ , she can close her derivation and wins; while her opponent, who faces  $\vdash 0$ , cannot close her derivation and loses.

Note that since a given play only explores one branch of each conjunction, it is not a derivation per se, rather an additive *slice* of a derivation.

This duality motivates the introduction of a neutral syntax replacing that of formulae. Conjunction and disjunction will be replaced by a single neutral connective  $+$ , its unit will be denoted to by  $\emptyset$ , and we will denote the change of player by  $\ddagger$ .

$$\begin{array}{ll}
[E_1 + E_2]^+ = [E_1]^+ \oplus [E_2]^+ & [E_1 + E_2]^- = [E_1]^- \& [E_2]^- \\
[0]^+ = 0 & [0]^- = \top \\
[\Downarrow E]^+ = [E]^- & [\Downarrow E]^- = [E]^+
\end{array}$$

Figure 2.1: Positive and negative translations of additive neutral expressions into ALL.

**Definition 2.1.8** (Additive neutral expressions). *The additive neutral expressions  $E$  and guarded additive neutral expressions  $G$  are inductively defined as follows.*

$$\begin{array}{l}
G, G' ::= E + E' \mid 0 \\
E, E' ::= G \mid \Downarrow G
\end{array}$$

An additive neutral expression can be seen as a pair of two dual formulae of ALL called its positive and negative *translations*. Figure 2.1 defines them.

We can now reformulate the game in terms of neutral expressions instead of formulae.

**Definition 2.1.9** (Propositional additive game). *The game on a neutral expression  $E$  consists of positions and moves. The positions are the occurrences of the sub-expressions of  $E$ . Consider the rewrite rules*

$$\begin{array}{ll}
E_1 + E_2 \mapsto E_1 & E_1 + E_2 \mapsto E_2
\end{array}$$

*There is a move  $E \rightarrow F$  iff  $E \mapsto^* \Downarrow F$ . A position from which there is no move is terminal. Plays are defined as before.*

**Definition 2.1.10** (Strategy). *A strategy for a position  $p$  is a minimal prefixed-closed set  $\sigma$  of plays such that:*

- $p \in \sigma$ ,
- if  $p_1, \dots, p_n \in \sigma$  is not maximal and  $n$  is odd, then there is exactly one position  $p_{n+1}$  such that  $p_n \rightarrow p_{n+1}$  and the play  $p_1, \dots, p_{n+1}$  is in  $\sigma$ ,
- if  $p_1, \dots, p_n \in \sigma$  is not maximal and  $n$  is even, then for every position  $p_{n+1}$  such that  $p_n \rightarrow p_{n+1}$ , the play  $p_1, \dots, p_{n+1}$  is in  $\sigma$ .

*A strategy is winning iff all its maximal plays have even length, and counter-winning iff all its maximal plays have odd length.*

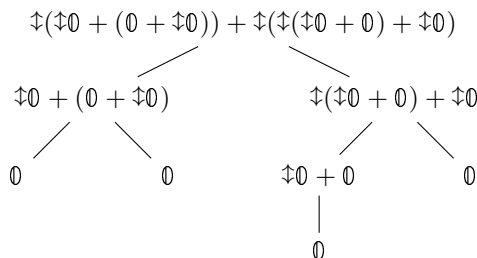
**Theorem 2.1.11.** *Let  $E$  be a neutral expression. There is a one-to-one correspondence between the proofs of  $[E]^+$  and the winning strategies for  $E$ . There is a one-to-one correspondence between the proofs of  $[E]^-$  and the counter-winning strategies for  $E$ .*

Additive behaviour is well captured by this game, with the benefit of improved symmetry. In Hintikka's game, the same player might move several times in a row, as long as the principal connective of the formula remains the same. In our game these moves correspond to individual rewrite steps and a move is a maximal sequence of such steps, which ensures a strict alternation of players. From the point of view of the proof objects, a rewrite step (a.k.a. a *micro-move*)

corresponds to the introduction of an individual connective or unit, while a move (a.k.a. a *macro-move*) corresponds to a full phase. The games for MALL we develop in the next chapters are more complex but their moves still have those two levels.

With this neutral syntax it is even clearer that this game may be seen as a process accounting for the simultaneous development of two dual derivations: starting from a neutral expression  $E$ , the player who begins sees the game as an attempt to derive  $\vdash [E]^+$ , while her opponent sees it as an attempt to derive  $\vdash [E]^-$ . With this in mind, each player develops what she sees as synchronous phases during her turn and leaves it to her opponent to decompose asynchronous phases.

Let us revisit Example 2.1.2 and describe it in terms of neutral expressions. The formula  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$  is replaced with the neutral expression  $\sharp(\sharp 0 + (0 + \sharp 0)) + \sharp(\sharp(\sharp 0 + 0) + \sharp 0)$ , of which it is the negative translation. We draw the tree of plays starting from this neutral expression:



In the original game, Abélard was the one to begin and Éloïse had winning strategies. Therefore the winning player was the one who did not start. It must also be the case in the neutral version of the game. In our terminology, there is a counter-winning strategy.

In the neutral game, there is a strict alternation of players. In contrast, Éloïse would play twice in a row at  $\top \oplus (0 \oplus \top)$  in the original game, for example. Those possible sequences of moves by Éloïse are replaced in the neutral game with the macro-moves from  $\sharp 0 + (0 + \sharp 0)$  to  $0$  and  $0$  (distinct occurrences). Those macro-moves correspond respectively to the sequences of micro-moves  $\sharp 0 + (0 + \sharp 0) \mapsto \sharp 0$  and  $\sharp 0 + (0 + \sharp 0) \mapsto 0 + \sharp 0 \mapsto \sharp 0$ . Since players alternate strictly, the parity of the length of a maximal play determines its winner.

There are two counter-winning strategies, corresponding to the two winning strategies of Éloïse in the original game. The first one is composed of the plays  $\sharp(\sharp 0 + (0 + \sharp 0)) + \sharp(\sharp(\sharp 0 + 0) + \sharp 0)$ ,  $\sharp 0 + (0 + \sharp 0)$ ,  $0$  and  $\sharp(\sharp 0 + (0 + \sharp 0)) + \sharp(\sharp(\sharp 0 + 0) + \sharp 0)$ ,  $\sharp(\sharp 0 + 0) + \sharp 0$ ,  $0$ . Its corresponding proof is, unsurprisingly, the same as the one corresponding to the original winning strategy. If we rewrite it in a focused proof system

$$\frac{\frac{\frac{\frac{\overline{\vdash \uparrow \top} [\top]}{\vdash \downarrow \top} [\mathbf{R}\downarrow]}{\vdash \downarrow \top \oplus (0 \oplus \top)} [\oplus_1]}{\vdash \top \oplus (0 \oplus \top) \uparrow} [\mathbf{D}]}{\vdash \uparrow \top \oplus (0 \oplus \top)} [\mathbf{R}\uparrow]}{\vdash \uparrow (\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)} [\&]}{\frac{\frac{\frac{\frac{\overline{\vdash \uparrow \top} [\top]}{\vdash \downarrow \top} [\mathbf{R}\downarrow]}{\vdash \downarrow (0 \& \top) \oplus \top} [\oplus_2]}{\vdash (0 \& \top) \oplus \top \uparrow} [\mathbf{D}]}{\vdash \uparrow (0 \& \top) \oplus \top} [\mathbf{R}\uparrow]}{\vdash \uparrow (\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)} [\&]}$$

it becomes clear that micro-moves are read as individual inference rules while macro-moves are read as phases. Synchronous phases are macro-moves by the player, and asynchronous phases are macro-moves by the opponent. The introduction of focalisation in ALL for this single remark may seem overkill, even more so because focalisation is trivial in that context. The abstraction provided by focalisation will prove useful when we introduce multiplicatives in the game.

### 2.1.3 Computation as dual proof search

In the above example, the player who starts does not have a winning strategy. Determining it requires to explore the arena much like a search space in proof search. This graph exploration progressively partitions the positions in two sets, starting from the terminal ones, depending on which player has a winning strategy for them. This process not only does this classification, but it also builds winning strategies. Since steps in the game (micro-moves and macro-moves) correspond to steps in proofs (rules and phases), it is clear that the exploration of the arena builds derivations gradually. In this respect, this game is deeply rooted in the tradition of computation as proof search. An interesting aspect of this computation is that we always get a winning strategy in the end (hence a proof), but we do not know for which player until we finish. Hence, the computation can be seen as the simultaneous development of two dual derivations, one of which will eventually be closed to become a proof.

In the above example, the maximal play  $\Downarrow(\Downarrow(0 + (0 + \Downarrow 0)) + \Downarrow(\Downarrow(0 + 0) + \Downarrow 0))$ ,  $\Downarrow(\Downarrow(0 + 0) + \Downarrow 0)$ ,  $\Downarrow 0 + 0$ ,  $0$  is seen as two dual derivation *slices*. From the point of view of the player who begins—and attempts to prove  $(0 \& (\top \& 0)) \oplus ((\top \oplus 0) \& 0)$ —, the play corresponds to the slice

$$\frac{\frac{\frac{\frac{\frac{\frac{\overline{\vdash \uparrow \top} [\top]}{\vdash \downarrow \top} [\mathbb{R}\downarrow]}{\vdash \downarrow \top \oplus 0} [\oplus_1]}{\vdash \top \oplus 0 \uparrow} [\mathbb{D}]}{\vdash \uparrow \top \oplus 0} [\mathbb{R}\uparrow]}{\vdash \uparrow (\top \oplus 0) \& 0} [\&_1]}{\vdash \downarrow (\top \oplus 0) \& 0} [\mathbb{R}\downarrow]}{\vdash \downarrow (0 \& (\top \& 0)) \oplus ((\top \oplus 0) \& 0)} [\oplus_2]}$$

while from the point of view of the player who does not begin—and attempts to prove  $(\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)$ —, the play corresponds to the slice

$$\frac{\frac{\frac{\frac{\frac{\frac{\overline{\vdash \downarrow 0} [\boxtimes]}{\vdash 0 \uparrow} [\mathbb{D}]}{\vdash \uparrow 0} [\mathbb{R}\uparrow]}{\vdash \uparrow 0 \& \top} [\&_1]}{\vdash \downarrow 0 \& \top} [\mathbb{R}\downarrow]}{\vdash \downarrow (0 \& \top) \oplus \top} [\oplus_1]}{\vdash (0 \& \top) \oplus \top \uparrow} [\mathbb{D}]}{\vdash \uparrow (0 \& \top) \oplus \top} [\mathbb{R}\uparrow]}{\vdash \uparrow (\top \oplus (0 \oplus \top)) \& ((0 \& \top) \oplus \top)} [\&_2]}$$



$$\begin{array}{ll}
[E_1 + E_2]^+ = [E_1]^+ \oplus [E_2]^+ & [E_1 + E_2]^- = [E_1]^- \& [E_2]^- \\
[0]^+ = 0 & [0]^- = \top \\
[E_1 \times E_2]^+ = [E_1]^+ \otimes [E_2]^+ & [E_1 \times E_2]^- = [E_1]^- \wp [E_2]^- \\
[\mathbb{1}]^+ = 1 & [\mathbb{1}]^- = \perp \\
[\Downarrow E]^+ = [E]^- & [\Downarrow E]^- = [E]^+
\end{array}$$

Figure 2.3: Positive and negative translations of additive and multiplicative neutral expressions into MALL.

nation procedure.

## 2.2 Simple games for a fragment of MALL

Our objective is to define a neutral game with not only additive but also multiplicative behaviour. In [MS06] the authors investigate such a game for a fragment of multiplicative and additive linear logic. In that fragment, multiplicatives of opposite polarities ( $\otimes$  and  $\wp$ ) are not allowed to interact. We present a slight variation of that fragment here. It is considerably simpler to handle than MALL. Again, we restrict our presentation to the propositional case.

Neutral expressions are extended to incorporate a multiplicative connective  $\times$  and its unit  $\mathbb{1}$ . Figure 2.3 shows their translations into MALL.

$$\begin{array}{l}
G, G' ::= E + E' \mid 0 \mid E \times E' \mid \mathbb{1} \\
E, E' ::= G \mid \Downarrow G
\end{array}$$

The fragment to which our study is restricted is presented further down. In the additive case, sequents only consisted of one formula. Positive and negative phases were of the form

Positive phase	Negative phase
$\frac{\vdash \uparrow [G_k]^-}{\vdash \Downarrow \bigoplus_{i=1}^n [G_i]^-}$	$\frac{\{\vdash [G_i]^+ \mid 1 \leq i \leq n\}}{\vdash \uparrow \&_{i=1}^n [G_i]^+}$
for some $k \in \{1, \dots, n\}$ .	

Plays correspond to additive slices, which consist of a single branch. We want to keep the same additive structure to the game, only introducing multiplicatives *inside* phases. A slice of a phase is allowed at most one premise, which must be a sequent with at most one formula. In other words, a positive phase needs to be either of the form

$$\frac{\overline{\vdash \Downarrow G_1} \quad \dots \quad \overline{\vdash \Downarrow G_{i-1}} \quad \frac{\vdash \uparrow H}{\vdash \Downarrow G_i} \quad \overline{\vdash \Downarrow G_{i+1}} \quad \dots \quad \overline{\vdash \Downarrow G_n}}{\vdash \Downarrow F}$$



or of the form

$$\frac{\frac{\overline{\overline{\vdash \Downarrow G_1}} \quad \dots \quad \overline{\overline{\vdash \Downarrow G_n}}}{\vdash \Downarrow F}}$$

and a slice of a negative phase needs to be of one of the forms

$$\frac{\frac{\vdash H \uparrow}{\vdash \uparrow G_1, \dots, G_n}}{\vdash \uparrow F} \quad \frac{\vdash}{\vdash \uparrow G_1, \dots, G_n}}{\vdash \uparrow F} \quad \frac{\overline{\overline{\vdash \uparrow G_1, \dots, G_n}}}{\vdash \uparrow F}}$$

With this restriction, derivations have the following shape:

- in positive phases, there are one or more branches and sequents contain exactly one formula;
- in negative phase slices, there is exactly one branch; the sequent contains zero or more formulae.

This particular shape suggests using multisets of neutral expression occurrences to represent the state of a game in this fragment. A multiset  $\{E_1, \dots, E_n\}$  will be read positively as the collection of sequents  $\vdash \Downarrow [E_1]^+, \dots, \vdash \Downarrow [E_n]^+$  and negatively as the sequent  $\vdash \uparrow [E_1]^-, \dots, [E_n]^-$ .

The micro-moves of the game are the following rewrite steps on multisets of neutral expression occurrences:

$$\begin{array}{ccc} \Gamma, \mathbb{1} \mapsto \Gamma & & \Gamma, E_1 \times E_2 \mapsto \Gamma, E_1, E_2 \\ \Gamma, E_1 + E_2 \mapsto \Gamma, E_1 & & \Gamma, E_1 + E_2 \mapsto \Gamma, E_2 \end{array}$$

where the comma denotes the multiset union.

Rewriting terminates since the total number of symbols of the neutral expressions in a multiset strictly decreases. Neutral expressions of the form  $\Downarrow G$  do not rewrite (they are to be passed to the other player), and neither does  $\mathbb{0}$ .

The restriction we want to impose on the logic corresponds to neutral expressions which yield, through rewriting, multisets containing at most one neutral expression of the form  $\Downarrow G$ . Formally, we define a measure  $\natural(E)$  representing the maximal number of neutral expressions of the form  $\Downarrow G$  in a multiset that can be obtained from  $E$  through rewriting.

$$\begin{array}{ll} \natural(\mathbb{0}) = \natural(\mathbb{1}) = 0 & \natural(E_1 + E_2) = \max(\natural(E_1), \natural(E_2)) \\ \natural(\Downarrow G) = 1 & \natural(E_1 \times E_2) = \natural(E_1) + \natural(E_2) \end{array}$$

**Definition 2.2.1** (Simple neutral expressions). *A neutral expression  $E$  is simple iff  $\natural(E) \leq 1$  and all sub-expressions of  $E$  are simple.*

Alternatively, simple neutral expressions  $E$  and guarded simple neutral expressions  $G$  are defined with the following grammar:

$$\begin{array}{l} Z, Z' ::= Z + Z' \mid \mathbb{0} \mid Z \times Z' \mid \mathbb{1} \\ G ::= Z \mid E \times Z \mid Z \times E \\ E ::= G \mid \Downarrow G \end{array}$$

We can now define the arena of the game.

**Definition 2.2.2.** *The positions are the simple neutral expression occurrences. If  $\{E\} \mapsto^* \{\}$ , then there is a macro-move from  $E$  to  $\mathbb{0}$ . If  $\{E\} \mapsto^* \{\Downarrow E'\}$ , then there is a macro-move from  $E$  to  $E'$ .*

Plays, strategies, winning and counter winning strategies are defined as in the additive game. The main result is similar.

**Theorem 2.2.3.** *Let  $E$  be a simple neutral expression. There is a one-to-one correspondence between the winning strategies for  $E$  and the focused proofs of  $[E]^+$ . There is a one-to-one correspondence between the counter winning strategies for  $E$  and the focused proofs of  $[E]^-$ .*

We refer to [MS06, DMS09] for further details. In contrast to the additive game, focalisation plays a non-trivial role in this case. The internal multiplicative structure of the macro-moves is abstracted away, and the precise ordering of micro-moves is irrelevant. This corresponds to the level of abstraction provided by focalisation on the proof-theoretic side.

For example, a multiset  $\{(E_1 + E_2) \times (F_1 + F_2)\}$  rewrites to  $\{E_i, F_j\}$  for any  $i, j \in \{1, 2\}$  and by two paths, depending on the order in which the micro-move replacing  $E_1 + E_2$  with  $E_i$  and the one replacing  $F_1 + F_2$  with  $F_j$  are scheduled. This order is irrelevant once we move to the macro-move level. On the proof-theoretic side, the two derivations seen by the players will be

$$\frac{\frac{\vdash \downarrow [E_i]^+}{\vdash \downarrow [E_1]^+ \oplus [E_2]^+} [\oplus_i] \quad \frac{\vdash \downarrow [F_j]^+}{\vdash \downarrow [F_1]^+ \oplus [F_2]^+} [\oplus_j]}{\vdash \downarrow ([E_1]^+ \oplus [E_2]^+) \times ([F_1]^+ \oplus [F_2]^+)} [\otimes]$$

by the player, and

$$\frac{\frac{\frac{\vdash \uparrow [E_1]^-, [F_1]^- \quad \vdash \uparrow [E_1]^-, [F_2]^-}{\vdash \uparrow [E_1]^-, [F_1]^- \& [F_2]^-} [\&] \quad \frac{\vdash \uparrow [E_2]^-, [F_1]^- \quad \vdash \uparrow [E_2]^-, [F_2]^-}{\vdash \uparrow [E_2]^-, [F_1]^- \& [F_2]^-} [\&]}{\vdash \uparrow [E_1]^- \& [E_2]^-, [F_1]^- \& [F_2]^-} [\&]}{\vdash \uparrow ([E_1]^- \& [E_2]^-) \wp ([F_1]^- \& [F_2]^-)} [\wp]$$

by the opponent. Note that, in the latter derivation, we arbitrarily chose a sequential presentation in which  $[E_1]^- \& [E_2]^-$  is introduced below  $[F_1]^- \& [F_2]^-$ , but focalisation precisely equates that derivation with the one obtained by commuting those rules:

$$\frac{\frac{\frac{\vdash \uparrow [E_1]^-, [F_1]^- \quad \vdash \uparrow [E_2]^-, [F_1]^-}{\vdash \uparrow [E_1]^- \& [E_2]^-, [F_1]^-} [\&] \quad \frac{\vdash \uparrow [E_1]^-, [F_2]^- \quad \vdash \uparrow [E_2]^-, [F_2]^-}{\vdash \uparrow [E_1]^- \& [E_2]^-, [F_2]^-} [\&]}{\vdash \uparrow [E_1]^- \& [E_2]^-, [F_1]^- \& [F_2]^-} [\&]}{\vdash \uparrow ([E_1]^- \& [E_2]^-) \wp ([F_1]^- \& [F_2]^-)} [\wp]$$

thus providing the adequate level of abstraction.



# A sequential neutral game for MALL

---

Additive behaviour is built in the structure of games. In a strategy, a player must pick a move among possibly several choices, and must be ready to accommodate to all the possible moves of the opponent. The neutral approach thus fits naturally in additive games. Multiplicative behaviour is more difficult to handle. Extending to the fragment of simple expressions yields a richer setting, expressive enough to encode numerous examples like bisimulation [MS06]. The game is essentially additive and has a simple and elegant structure. In this chapter, we introduce a new game which pushes the neutral approach to the full propositional fragment of multiplicative and additive linear logic. In that fragment, multiplicatives are allowed to interact, which requires the introduction of complex graph structures to represent the state of the game. Moreover, the game is not determinate any more. This chapter covers some of the material published in [DM08, DMS09].

## 3.1 Unrestricted multiplicatives

In this chapter we will be dealing with the propositional multiplicative and additive fragment of linear logic (MALL). We recall the syntax of formulae:

$$F, G ::= A \mid A^\perp \mid F \oplus G \mid 0 \mid F \& G \mid \top \mid F \otimes G \mid 1 \mid F \wp G \mid \perp$$

where  $A$  stands for an atom. For each pair of dual formulae  $F$  and  $F^\perp$  (including literals),  $F^+$  will denote the positive one and  $F^-$  the negative one.

### 3.1.1 Incompleteness

The two neutral games presented so far were based on two fragments of linear logic: the additive fragment and the simple multiplicative and additive fragment. Those fragments have the property of being complete, that is, for every pair of dual formulae in the fragment, one of the formulae is provable. As a result, the two games are determinate, i.e. at each position some player has a winning strategy. A play ends as soon as the current player cannot develop her derivation any more, making her opponent win immediately.

In game semantics for logic players usually play non-symmetric roles. Éloïse (a.k.a. the player) aims at proving a formula while the Abélard (a.k.a. the opponent or environment) aims at refuting it. That kind of game is determinate since a formula is either provable or not. Better yet, one can hope to get countermodels witnessing the non provability of the formula from Abélard's winning strategies.

In contrast, our neutral approach features two players with symmetric roles who aim at proving dual formulae. In the additive game (Section 2.1), this slight change is not visible: the logic is complete and the refutations of a formula are the same as the proofs of its negation. The simple fragment studied in Section 2.2 is no different in this regard. This is not the case any more in an incomplete logic like MALL. For example, the dual formulae  $\perp \otimes \top$  and  $1 \wp 0$  are both unprovable. As a consequence, neutral games can no longer be determinate. We keep the proof-theoretic reading of each move as the simultaneous development of two dual derivations, but, when one of the players fails to complete her derivation, we cannot conclude that her opponent wins as we did before. The play must continue until the opponent completes her derivation (thus winning the play) or fails as well (thus ending the play in a tie). The derivations must therefore be richer objects which leave some room for failure. For example, consider the following dual derivations:

$$\frac{\frac{\frac{\vdash A, \perp \quad \overline{\vdash \top}}{\vdash A, \perp \otimes \top}}{\vdash A \wp (\perp \otimes \top)}}{\vdash A^\perp \quad \frac{\frac{\vdash 1, 0}{\vdash 1 \wp 0}}{\vdash A^\perp \otimes (1 \wp 0)}}$$

Clearly inference rules are applied dually up to the point where  $\top$  is introduced on the left. At that point there is no dual rule to apply on the right. As a matter of fact the sequent  $\vdash 1, 0$  is not provable. However, we still need a way to challenge the left derivation, which may or may not be completed into a proof, depending on  $A$ . In order to preserve the reading of each micro-move as the simultaneous application of a rule in both derivations, we shall enrich our proof system with a rule acting as a “joker” allowing to recover from a failure. We call this rule the “daimon” by analogy with Ludics [Gir01]. A proof is now a closed derivation which does not use the daimon rule. A player who uses daimon cannot win, but may try to make her opponent fail as well. In the last example, the derivation on the left closes the branch with  $\top$ . The dual operation consists in using the daimon rule to get rid of  $0$  in the derivation on the right:

$$\frac{\frac{\frac{\vdash A, \perp \quad \overline{\vdash \top}}{\vdash A, \perp \otimes \top}}{\vdash A \wp (\perp \otimes \top)}}{\vdash A^\perp \quad \frac{\frac{\frac{\vdash 1}{\vdash 1, 0} \wp}{\vdash 1 \wp 0}}{\vdash A^\perp \otimes (1 \wp 0)}}$$

where  $\wp$  stands for the daimon rule. Naturally, the right derivation cannot be completed into a proof because of this daimon rule, but interaction can continue. Indeed, it is now possible to introduce  $\perp$  on the left and, dually,  $1$  on the right:

$$\frac{\frac{\frac{\vdash A}{\vdash A, \perp} \quad \overline{\vdash \top}}{\vdash A, \perp \otimes \top}}{\vdash A \wp (\perp \otimes \top)} \quad \left| \quad \frac{\frac{\overline{\vdash 1}}{\vdash 1, 0} \wp}{\vdash 1 \wp 0}}{\vdash A^\perp \otimes (1 \wp 0)}$$

This is just an informal example however, as our proof system will be focused, as explained below.

### 3.1.2 Concurrency and focusing

In the additive game, moves must be done in sequence: at any point, there is only one connective to consider, which is the main connective of the current formula.



formulae,  $\mathcal{N}$  is a multiset of negative formulae, and  $\mathcal{F}$  is a multiset of formulae. The sequents  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow$  and  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow$  are identified and also denoted to by  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N}$ .

### 3.3 Basic definitions

Name two players 0 and 1. If  $\lambda$  denotes a player,  $\bar{\lambda} = 1 - \lambda$  denotes her opponent. We still base our game on a set of positions  $\mathcal{P}$  and a macro-move binary relation on positions  $\rightarrow$ . A position from which no move is possible is called *terminal*. All terminal positions are classified as 0-wins, 1-wins, and ties, and the non-terminal positions as 0-positions and 1-positions. If  $p$  is a position, a *play from*  $p$  is a path in the arena starting with  $p$ . We usually assume  $\rightarrow$  to be noetherian, so that all plays are finite. A play is *won* by player  $\lambda$  iff its last position is a  $\lambda$ -win, and is a *tie* iff its last position is a tie.

**Definition 3.3.1** ( $\lambda$ -strategy). *Let  $\lambda \in \{0, 1\}$ . A  $\lambda$ -strategy for a position  $p$  is a minimal prefixed-closed set  $\sigma$  of plays such that:*

- $p_1 \in \sigma$ ,
- if  $p_1, \dots, p_n \in \sigma$  where  $p_n$  is a  $\lambda$ -position, then there is exactly one position  $p_{n+1}$  such that  $p_n \rightarrow p_{n+1}$  and the play  $p_1, \dots, p_{n+1}$  is in  $\sigma$ ,
- if  $p_1, \dots, p_n \in \sigma$  where  $p_n$  is a  $\bar{\lambda}$ -position, then for every position  $p_{n+1}$  such that  $p_n \rightarrow p_{n+1}$ , the play  $p_1, \dots, p_{n+1}$  is in  $\sigma$ .

A  $\lambda$ -strategy is winning iff all its maximal plays are won by player  $\lambda$ .

### 3.4 Neutral expressions

So far neutral expressions contained additive and multiplicative connectives along with an operator  $\dagger$  denoting the change of polarity or player. In [DM08] we removed the restriction on multiplicative interactions, essentially treating MALL without atoms. We later included a neutral constructor for atoms in [DMS09].

We extend the syntax of neutral expressions. Figure 3.2 defines their translations into MALL.

**Definition 3.4.1** (Neutral expressions). *The additive and multiplicative neutral expressions  $E$  and guarded additive and multiplicative neutral expressions  $G$  are inductively defined as follows.*

$$\begin{aligned} G, G' &::= a \mid E + E' \mid \mathbb{0} \mid E \times E' \mid \mathbb{1} \\ E, E' &::= G \mid \dagger G \end{aligned}$$

where  $a$  stands for a neutral atom. There is a neutral atom for each pair of dual literals of MALL.

### 3.5 Neutral graphs

In order to account for the complexity and intensional behaviour of the multiplicative connectives and atoms of MALL, we shall not enrich the structure of

**Additives**

$$\frac{\frac{\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F_i, \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F_1 \oplus F_2, \mathcal{F}} [\oplus_i]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow F_1, \mathcal{F}} \quad \frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow F_2, \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow F_1 \& F_2, \mathcal{F}} [\&]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow \top, \mathcal{F}} [\top]$$

**Multiplicatives**

$$\frac{\frac{\frac{\frac{\vdash \mathcal{L}_1; \mathcal{P}_1; \mathcal{N}_1 \Downarrow F_1, \mathcal{F}_1}{\vdash \mathcal{L}_1, \mathcal{L}_2; \mathcal{P}_1, \mathcal{P}_2; \mathcal{N}_1, \mathcal{N}_2 \Downarrow F_1 \otimes F_2, \mathcal{F}_1, \mathcal{F}_2} [\otimes]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow F_1, F_2, \mathcal{F}} [\wp]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow F_1 \wp F_2, \mathcal{F}} [\wp]} \quad \frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow \perp, \mathcal{F}} [\perp]}{\vdash \vdash; \Downarrow 1} [1]$$

**Identity**

$$\frac{}{\vdash L^-; \Downarrow L^+} [\text{init}]$$

**Daimon**

$$\frac{}{\Sigma} [\boxtimes] \quad \frac{\Sigma'}{\Sigma} [\boxtimes]$$

**Phase changes**

$$\frac{\frac{\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N}, \mathcal{N} \Downarrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow N, \mathcal{F}} [\text{R}\Downarrow]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{P}; \mathcal{N} \Uparrow \mathcal{F}} [\text{R}\Uparrow]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow P, \mathcal{F}} [\text{R}\Uparrow]} \quad \frac{\vdash L^-, \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Uparrow L^-, \mathcal{F}} [\text{atomic R}\Uparrow]}{\frac{\frac{\frac{\vdash \mathcal{L}; \mathcal{P}_1; \Downarrow \mathcal{P}_2}{\vdash \mathcal{L}; \mathcal{P}_1, \mathcal{P}_2} [\text{D}\Downarrow]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N}_1 \Uparrow \mathcal{N}_2, \mathcal{F}} [\text{D}\Uparrow]}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N}_1, \mathcal{N}_2 \Uparrow \mathcal{F}} [\text{D}\Uparrow]}}$$

$L$  denotes a literal,  $N$  a negative formula, and  $P$  a positive formula. In  $[\text{D}\Downarrow]$  (resp.  $[\text{D}\Uparrow]$ ),  $\mathcal{P}_2$  (resp.  $\mathcal{N}_2$ ) is not empty. A proof is a closed derivation which does not use  $[\boxtimes]$ .

Figure 3.1: The proof system used in the sequential game for MALL



$$\begin{array}{ll}
[a]^+ = A^+ & [a]^- = A^- \\
[E_1 + E_2]^+ = [E_1]^+ \oplus [E_2]^+ & [E_1 + E_2]^- = [E_1]^- \& [E_2]^- \\
[0]^+ = 0 & [0]^- = \top \\
[E_1 \times E_2]^+ = [E_1]^+ \otimes [E_2]^+ & [E_1 \times E_2]^- = [E_1]^- \wp [E_2]^- \\
[1]^+ = 1 & [1]^- = \perp \\
[\Downarrow E]^+ = [E]^- & [\Downarrow E]^- = [E]^+
\end{array}$$

Figure 3.2: Positive and negative translations of additive and multiplicative neutral expressions into MALL.  $A^+$  and  $A^-$  are the MALL literals corresponding to the neutral atom  $a$ .

arenas and plays. That will be done in the next chapter, in which we introduce a form of concurrency in the game. Instead, we enrich the notion of position by moving from being just simple neutral expressions (as was used in the additive game) to labelled graph structures, which we describe next.

### 3.5.1 Interacting multiplicatives

Figure 3.3 shows an example of two dual derivations in our proof system. We use neutral expressions instead of formulae to emphasise the duality of the two derivations. The conclusions of derivations are the positive and negative translations of the neutral expression  $\Downarrow(\Downarrow E \times \Downarrow F) \times \Downarrow G$ , where  $E$ ,  $F$ , and  $G$  are non atomic neutral expressions.

A first remark is that we only examine purely multiplicative behaviour in that example. Additives will be treated in this game as they were in the two previous games, and we focus on the novelty of having interactions between  $\otimes$  and  $\wp$ . Next, the neutral expression  $\Downarrow(\Downarrow E \times \Downarrow F) \times \Downarrow G$  chosen here is not in the simple fragment addressed in Section 2.2. Indeed, the multiset  $\{\Downarrow(\Downarrow E \times \Downarrow F) \times \Downarrow G\}$  immediately rewrites to  $\{\Downarrow(\Downarrow E \times \Downarrow F), \Downarrow G\}$ , which consists of two neutral expressions beginning with  $\Downarrow$ .

It can be noted that at any point in the simultaneous development of those derivations, each formula present in a frontier has its dual in the other frontier (note: if additives were present, we would have to consider additive slices of the derivations). This correspondence between formulae is one-to-one. For example at the bottom of the derivations the frontier of derivation 3.1) consists of the sequent  $\vdash; ([E]^+ \wp [F]^+) \otimes [G]^-$ ; and the frontier of derivation 3.2 consists of  $\vdash; ([E]^- \otimes [F]^-) \wp [G]^+$ . Clearly there is exactly one formula in each frontier and they are dual. At the top of the two derivations, the frontiers are  $\vdash; [E]^+, [F]^+$ ; and  $\vdash; [G]^-$  for derivation 3.1, and  $\vdash; [G]^+, [E]^-$  and  $\vdash; [F]^-$  for derivation 3.2. Here, the corresponding pairs are  $[E]^+/[E]^-$ ,  $[F]^+/[F]^-$ , and  $[G]^-/[G]^+$ .

We can give an abstract description of the situation with a multiset of neutral expressions equipped with two partitions. We collect all the neutral expressions whose translations are present in the frontiers of the two derivations at a given point in a multiset. Then we associate a partition of that multiset to each derivation which describes how the corresponding formulae are grouped in sequents.

$$\begin{array}{c}
\frac{\frac{\frac{\frac{\frac{\vdash; [E]^+, [F]^+}{\vdash; \uparrow [E]^+, [F]^+} [\text{R}\uparrow]}{\vdash; \uparrow [E]^+ \wp [F]^+} [\wp]}{\vdash; [E]^+ \wp [F]^+} [\text{D}\uparrow]}{\vdash; \downarrow [E]^+ \wp [F]^+} [\text{R}\downarrow]}{\vdash; \downarrow ([E]^+ \wp [F]^+) \otimes [G]^-} [\otimes]} \quad \frac{\vdash; [G]^-}{\vdash; \downarrow [G]^-} [\text{R}\downarrow]}{\vdash; \downarrow ([E]^+ \wp [F]^+) \otimes [G]^-} [\text{D}\downarrow]} \\
\vdash; ([E]^+ \wp [F]^+) \otimes [G]^-; \quad (3.1)
\end{array}$$

$$\begin{array}{c}
\frac{\frac{\frac{\frac{\frac{\frac{\vdash; [G]^+, [E]^-}{\vdash; [G]^+, \downarrow [E]^-} [\text{R}\downarrow]}{\vdash; [G]^+ \downarrow [E]^-} [\otimes]}{\vdash; [G]^+, \downarrow [E]^- \otimes [F]^-} [\text{D}\downarrow]}{\vdash; [E]^- \otimes [F]^-; [G]^+} [\text{R}\uparrow]}{\vdash; \uparrow ([E]^- \otimes [F]^-) \wp [G]^+} [\wp]}{\vdash; ([E]^- \otimes [F]^-) \wp [G]^+} [\text{D}\uparrow]} \quad \frac{\frac{\vdash; [F]^-}{\vdash; \downarrow [F]^-} [\text{R}\downarrow]}{\vdash; \downarrow [F]^-} [\otimes]}{\vdash; ([E]^- \otimes [F]^-) \wp [G]^+} [\text{D}\uparrow]} \\
\vdash; ([E]^- \otimes [F]^-) \wp [G]^+ \quad (3.2)
\end{array}$$

$E$ ,  $F$  and  $G$  are non atomic neutral expressions.

Figure 3.3: Two dual derivations in our proof system.

At the bottom of the derivations, the multiset of the neutral expressions is simply  $\{\downarrow(\downarrow E \times \downarrow F) \times \downarrow G\}$ . Both derivations see the only possible partition  $\{\{\downarrow(\downarrow E \times \downarrow F) \times \downarrow G\}\}$ . At the top of the derivations, the multiset is  $\{E, F, G\}$ , derivation 3.1 sees the partition  $\{\{E, F\}, \{G\}\}$ , and derivation 3.2 sees the partition  $\{\{E, G\}, \{F\}\}$ . Naturally, this information is not enough: we must also specify, for each neutral expression, which derivation sees its positive translation and which sees its negative translation.

A similar observation is made by Danos and Regnier in [DR89]. The authors show that partitions can be associated to multiplicative rules of linear logic. For example, by removing the context from the  $\otimes$  rule, we get

$$\frac{\vdash A \quad \vdash B}{\vdash A \otimes B} \otimes$$

and the associated partition is  $\{\{1\}, \{2\}\}$ , denoting that  $A$  and  $B$  end up in distinct sequents. Dually, the  $\wp$  rule without context

$$\frac{\vdash A, B}{\vdash A \wp B} \wp$$

is associated with the partition  $\{1, 2\}$ , denoting that  $A$  and  $B$  end up in the same sequent. This becomes interesting when mixing multiplicatives. For example the generalised dual rules

$$\frac{\vdash A \quad \vdash B, C}{\vdash (A \otimes B) \wp C} \otimes, \wp \quad \frac{\vdash A, B \quad \vdash C}{\vdash (A \wp B) \otimes C} \wp, \otimes$$

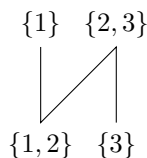
would be associated with the partitions  $\{\{1\}, \{2, 3\}\}$  and  $\{\{1, 2\}, \{3\}\}$  respectively. The authors also consider generalised multiplicative rules, defined by

such partitions, which cannot be decomposed in a layer of  $\otimes$  and  $\wp$  rules. Those generalised multiplicatives are beyond the scope of this chapter. However, the authors make important remarks regarding the computational content of such multiplicative rules. In the description of cut elimination for those extended connectives, they remark that a “key-step” of cut elimination such as

$$\frac{\frac{\frac{\vdash A \quad \vdash B, C}{\vdash (A \otimes B)} \wp, \wp}{\vdash} \quad \frac{\frac{\frac{\vdash A^\perp, B^\perp \quad \vdash C^\perp}{\vdash (A^\perp \wp B^\perp)} \wp, \otimes}{\vdash} \text{ cut}}{\vdash} \rightsquigarrow \frac{\frac{\vdash A \quad \vdash B, C \quad \vdash A^\perp, B^\perp \quad \vdash C^\perp}{\vdash} \text{ cut}}{\vdash}$$

can only be carried out if the partitions associated with the two dual connectives are *orthogonal*.

Orthogonality in this context refers to an acyclicity property which is the matching piece to the correctness criterion for multiplicative proofs nets [Gir87, DR89]. For two partitions of the same set, consider the unoriented graph whose vertices are the classes of the partitions, and in which there is an arc between two classes when they share an element. Note that this is more precisely a multigraph, for two classes are linked by one arc for each element they share. The two partitions are orthogonal iff this *induced graph* is connected and acyclic. For example, the two partitions  $\{\{1\}, \{2, 3\}\}$  and  $\{\{1, 2\}, \{3\}\}$  from the above example are orthogonal:



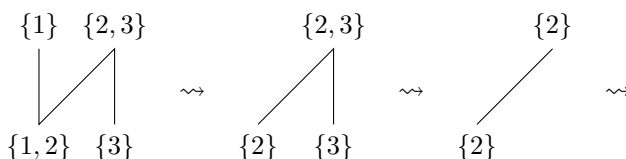
What we represent by a layer of cuts in sequent calculus

$$\frac{\frac{\vdash A \quad \vdash B, C \quad \vdash A^\perp, B^\perp \quad \vdash C^\perp}{\vdash} \text{ cut}}{\vdash}$$

can actually be sequentialised by reading the graph and contracting the arcs corresponding to the pairs of dual formulae being cut. For example, the sequentialisation

$$\frac{\frac{\frac{\vdash A \quad \vdash A^\perp, B^\perp}{\vdash B^\perp} \text{ cut} \quad \frac{\frac{\vdash B, C \quad \vdash C^\perp}{\vdash B} \text{ cut}}{\vdash} \text{ cut}}{\vdash}$$

corresponds to contracting the arcs “1” (corresponding to  $A/A^\perp$ ) and “3” (corresponding to  $C/C^\perp$ ), then “2” (corresponding to  $B/B^\perp$ ) in the induced graph:



We could have chosen to sequentialise those cuts in a different order. This graphical representation therefore abstracts away from that order. In sequent calculus, the same abstraction is provided by considering a layer of cuts as a synthetic “multicut” inference rule. Other ways exist to represent the same information: using cut links between the conclusions of proof structures [Gir87, Gir96], or cut-nets in Ludics [Gir01].

### 3.5.2 A neutral presentation

Inspired by the approach by Danos and Regnier, we restate the invariant we pointed out in the additive and simple multiplicative games. The sequents at the frontier of two (additive slices of) orthogonal derivations can be cut together to infer the empty sequent. In our example, the situation at the bottom of the derivations 3.1 and 3.2 is summarised by the following cut:

$$\frac{\vdash; ([E]^+ \wp [F]^+) \otimes [G]^-; \quad \vdash; ([E]^- \otimes [F]^-) \wp [G]^+}{\vdash;} \text{ cut}$$

Here we trivially extend our proof system with a cut rule; this rule will only be used in meta-level discussions, not in the game itself. Reading the derivations bottom-up, the situation after one phase is summarised by the multicut:

$$\frac{\vdash; [E]^+ \wp [F]^+ \quad \vdash; [G]^- \quad \vdash; [E]^- \otimes [F]^-, [G]^+;}{\vdash;} \text{ cut}$$

and, after another phase, at the top of the derivations:

$$\frac{\vdash; [E]^+, [F]^+; \quad \vdash; [G]^- \quad \vdash; [G]^+; [E]^- \quad \vdash; [F]^-}{\vdash;} \text{ cut}$$

We use labelled graph structures called neutral graphs to represent those successive states, just like induced graphs represent orthogonal partitions. Nodes represent the sequents, and arcs represent pairs of formulae that are cut. There are two kinds of nodes (one for each derivation/player). The arc

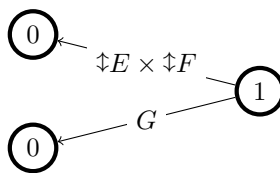
$$\begin{array}{ccc} \textcircled{0} & \xrightarrow{E} & \textcircled{1} \\ u & & v \end{array}$$

labelled with a guarded neutral expression  $E$  means that the formula  $[E]^+$  occurs in the sequent represented by  $u$  belonging to player 0 and that the formula  $[E]^-$  occurs in the sequent represented by  $v$  belonging to player 1. A neutral graph is bipartite: recall that we do not pair two formulae in the same derivation.

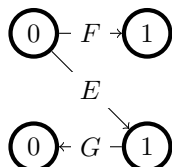
For example, the neutral graph representing the situation at the bottom of derivations (3.1) and (3.2) is

$$\textcircled{0} \text{ --- } \wp(\wp E \times \wp F) \times G \text{ --- } \textcircled{1}$$

where players 0 and 1 see derivations (3.1) and (3.2) respectively. The intermediate situation is represented by



and the situation at the top of the derivations is represented by



Remark that, in the examples given so far, we only observed the situations at phase boundaries, where all the sequents are of the form  $\vdash; \mathcal{P}; \mathcal{N}$ . The transitions between the neutral graphs will be macro-moves in the game. We also need to represent sequents of the forms  $\vdash; \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}$  and  $\vdash; \mathcal{L}; \mathcal{P}; \mathcal{N} \downarrow \mathcal{F}$ . To this end, we will use several types of arcs in neutral graphs. A lower level of transitions between neutral graphs, the micro-moves, will account for individual rule applications.

Although computation in the game can be seen as a form of cut reduction (with peculiarities due to focalisation and failure), it should be noted that the neutral graphs themselves are not proof structures. They merely represent cut links between goals for the players to prove, which are placeholders for proof structures.

**Definition 3.5.1.** A neutral graph  $G$  is a tuple  $(N, A, \Lambda, \tau, \epsilon)$ , where  $N$  is a finite set (possibly empty) of nodes,  $A \subseteq N \times N$  is a set of arcs,  $\Lambda : N \mapsto \{0, 1\}$  associates a player to each node,  $\tau : A \mapsto \{\text{atomic}, \text{normal}, \text{focused}\}$  associates a type to each arc, and  $\epsilon : A \mapsto \mathcal{E}$  associates a neutral expression to each arc. In addition, the following must hold:

- The undirected graph based on  $(N, A)$  is a set of trees none of which are the degenerate (one-node) tree.
- The graph is bipartite, i.e., for every  $(u, v) \in A$ ,  $\Lambda(u) \neq \Lambda(v)$ .
- For every normal arc  $a \in A$ ,  $\epsilon(a)$  is guarded.
- For every atomic arc  $a \in A$ ,  $\epsilon(a)$  is an atom.
- The origins of the focused arcs all belong to the same player (and similarly for the ends, since the graph is bipartite).

A node with player  $\lambda$  is said to *belong* to  $\lambda$  or to be a  $\lambda$ -node.

Arcs are represented using the following convention:



Unsurprisingly, the definition requires that neutral graphs have no undirected cycles. What is less obvious is that we do not require them to be connected, when graphs induced by partitions are. The reason for that will be made clear when we deal with failure, which is a part of the game, but not of proof theory. Also, notice that no node may be isolated (i.e., without neighbours). Equivalently, neutral graphs do not represent the empty sequent.

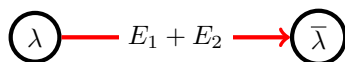
We now relate neutral graphs to frontiers of additive slices of derivations. Each node has an associated sequent, whose formulae are the translations of the neutral expressions labelling the arcs connected to it. The direction of an arc determines which translation (positive or negative) to consider, while its type (atomic, normal or focused) determines in which part of the sequent the formula occurs.

**Definition 3.5.2** (Sequent associated with a node). *Let  $G = (N, A, \Lambda, \tau, \epsilon)$  be a neutral graph and  $v \in N$ . Let*

- $\mathcal{L}^+ = \{[\epsilon(v, w)]^+ : (v, w) \in A, \tau(v, w) = \text{atomic}\},$
- $\mathcal{L}^- = \{[\epsilon(u, v)]^- : (u, v) \in A, \tau(u, v) = \text{atomic}\},$
- $\mathcal{U}^+ = \{[\epsilon(v, w)]^+ : (v, w) \in A, \tau(v, w) = \text{normal}\},$
- $\mathcal{U}^- = \{[\epsilon(u, v)]^- : (u, v) \in A, \tau(u, v) = \text{normal}\},$
- $\mathcal{F}^+ = \{[\epsilon(v, w)]^+ : (v, w) \in A, \tau(v, w) = \text{focused}\},$
- $\mathcal{F}^- = \{[\epsilon(u, v)]^- : (u, v) \in A, \tau(u, v) = \text{focused}\}.$

*Since  $G$  is a neutral graph, at least one of  $\mathcal{F}^+$  and  $\mathcal{F}^-$  is empty. The sequent  $\Sigma_G(v)$  associated with  $v$  is  $\vdash \mathcal{L}^+, \mathcal{L}^-; \mathcal{U}^+; \mathcal{U}^- \uparrow \mathcal{F}^-$  if  $\mathcal{F}^+$  is empty, and  $\vdash \mathcal{L}^+, \mathcal{L}^-; \mathcal{U}^+; \mathcal{U}^- \Downarrow \mathcal{F}^+$  otherwise.*

Focused arcs mark neutral expressions that are scheduled for decomposition in a phase. For each focused arc, the node at the origin sees the positive translation under focus in a synchronous phase, while the node at the end sees the negative translation under focus in an asynchronous phase. In a focused proof system, choices are made during synchronous phases. For example, in the case of the following focused arc



player  $\lambda$  sees the sequent  $\vdash; \Downarrow [E_1]^+ \oplus [E_2]^+$ , while player  $\bar{\lambda}$  sees the sequent  $\vdash; \uparrow [E_1]^- \& [E_2]^-$ . As was the case in the previous games, player  $\lambda$ —the one seeing the additive as the disjunction  $\oplus$ —will have to make a choice between  $E_1$  and  $E_2$ . More generally, all the micro-moves dealing with an arc will be played by the player owning the node at the origin of the arc. In the definition of a neutral graph, it is required that the origins of the focused arcs all belong to the same player. This condition simply states that we only allow one player to be playing at a time.

**Definition 3.5.3** (Source). *A source of a neutral graph is a node which is not the end of a normal or focused arc, but is the origin of some normal arc.*

Sources play a significant role, because they are precisely the nodes associated with sequents of the form  $\vdash \mathcal{L}; \mathcal{P}$ ; with  $\mathcal{P} \neq \emptyset$ , i.e., those which may appear as the conclusion of the  $[D\Downarrow]$  rule, which marks the beginning—reading bottom-up—of a synchronous phase. In other words, they are the places where a player begins to move.

## 3.6 Positions and moves

### 3.6.1 Positions

As in the previous games, we define positions and moves on two levels. A first level is made of *micro-positions* and *micro-moves* between them. A second level

is made of *macro-positions* and *macro-moves* between them.

We first introduce a basic notion which will be used to define micro-positions and macro-positions. A neutral graph contains almost all the state of the game: it represents the frontiers of the additive slices of the derivations being developed. Players rewrite this object as they move, extending their derivations simultaneously. Our game, however, is indeterminate and must account for the possibility of a player failing in the course of a play. When that happens, the play does not end with the victory of the opponent, but this information is recorded in the state of the play. The interaction continues, but can never be won by the player who has failed. A position represents the state of the game.

**Definition 3.6.1** (Position). *A position is a triple  $(G, f_0, f_1)$  where  $G$  is a neutral graph and  $f_0$  and  $f_1$  are Boolean values, satisfying the two following properties:*

- *if  $G$  is empty, then at least one of  $f_0$  and  $f_1$  is true;*
- *if there is some atomic arc  $(u, v)$  in  $G$ , then  $f_\lambda$  must be true, where  $\lambda$  is the player associated with  $u$ .*

Informally, the flag  $f_0$  (resp.  $f_1$ ) is true iff player 0 (resp. 1) has failed. The first property expresses that when a play is over ( $G$  is empty), it is not possible for both players to have succeeded in proving their goals (at least one of them must have failed). The second property is a technical invariant, expressing that the origin of an atomic arc has an associated sequent which will only be obtained through an application of the daimon  $\boxtimes$ . Consequently, the corresponding player will necessarily have failed.

**Definition 3.6.2** (Macro-position). *A macro-position is a position  $(G, f_0, f_1)$  such that  $G$  has no focused arc and all its sources (possibly zero) belong to the same player.*

Informally, a macro-move is seen as a synchronous phase by the player and as an asynchronous phase by the opponent. A macro-position is the state between macro-moves. At a macro-position, no formula is under focus. Moreover, there must be at most one player ready to start a synchronous phase, i.e., ready to move.

**Definition 3.6.3** (Terminal/ $\lambda$ -macro-position). *Let  $p = (G, f_0, f_1)$  be a macro-position. If  $G$  has no source, then  $p$  is terminal. Otherwise, the sources of  $G$  belong to some player  $\lambda$  and  $p$  is a  $\lambda$ -macro-position.*

A terminal macro-position marks the end of a play. At any other macro-position exactly one player can move.

**Definition 3.6.4** (Tie,  $\lambda$ -win). *Let  $p = (G, f_0, f_1)$  be a terminal macro-position. Then at least one of  $f_0$  and  $f_1$  is true. If they are both true,  $p$  is a tie. Otherwise  $p$  is a  $\lambda$ -win, where  $\lambda \in \{0, 1\}$  is such that  $f_\lambda$  is false.*

*Proof.* By Definition 3.6.2,  $G$  has no focused arc. Since  $G$  is acyclic and has no sources,  $G$  has no normal arc. All the arcs of  $G$  are therefore atomic. Whether  $G$  is empty or not, at least one of  $f_0$  and  $f_1$  is true by Definition 3.6.1.  $\square$

In order to win a play, a player must cause her opponent to fail without doing so herself.

Transition	Sync reading	Async reading
$p \xrightarrow{D} p'$	$[D\Downarrow]$	$[D\Uparrow]$
$p \xrightarrow{R} p'$	$[R\Downarrow]$	$[R\Uparrow]$
$p \xrightarrow{\perp} p'$	$[\oplus]$	$[\&]$
$p \xrightarrow{0} p'$	$[\boxtimes]$	$[\top]$
$p \xrightarrow{\times} p'$	$[\otimes]$	$[\wp]$
$p \xrightarrow{\perp} p'$	$[1]$ or $[\boxtimes]$	$[\perp]$ or $[\boxtimes]$
$p \xrightarrow{\text{ax}} p'$	$[\text{init}]$ or $[\boxtimes]$	$[\text{atomic } R\Uparrow]$ or $[\boxtimes]$

Table 3.1: Neutral micro-moves and their two readings.

The arena of the game consists of the macro-positions and the macro-moves, which will be defined later. Player  $\lambda$  plays at a  $\lambda$ -macro-position and wins at a  $\lambda$ -win.

**Definition 3.6.5** ( $\lambda$ -micro-position). *Let  $\lambda \in \{0, 1\}$ . A  $\lambda$ -micro-position is a position  $(G, f_0, f_1)$  such that all the origins of the focused arcs of  $G$  belong to player  $\lambda$ .*

Informally, a  $\lambda$ -micro-position is an intermediate step which may appear during player  $\lambda$ 's macro-moves. The origins of the focused arcs are the nodes associated to sequents in the middle of a synchronous phase.

**Definition 3.6.6** (Playable  $\lambda$ -micro-position). *A  $\lambda$ -micro-position  $(G, f_0, f_1)$  is playable if  $G$  has at least one focused arc or one source belonging to player  $\lambda$ .*

A player continues her turn as long as some of her sequents are in the middle of a synchronous phase (origins of focused arcs) or are ready to start one (sources). Notice that every  $\lambda$ -macro-position is a playable  $\lambda$ -micro-position. Informally, player  $\lambda$  can always play at a  $\lambda$ -macro-position.

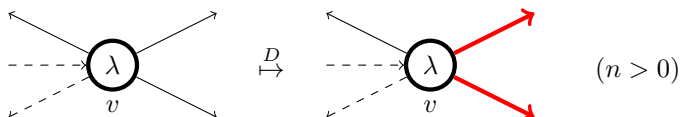
### 3.6.2 Micro-moves

This section describes the transitions on neutral graphs that are the basis of the game. We first introduce six of them, the aforementioned “micro-moves”, that should be interpreted as the simultaneous applications of two dual single rules of the proof system. Table 3.1 lists them along with their interpretations. We subsequently build another transition, which packs a maximal sequence of micro-moves together and should be read as the simultaneous development of two dual phases. Essentially, micro-moves are cut reduction rules for proof structures, with focalisation. A notable difference is that cut reduction operates on readily available proof structures, while our micro-moves can be seen as an attempt to develop those structures as cut reduction goes. It is not always possible to develop them. As a result, failures may arise in some of these transitions; in that case the transition makes the relevant flags  $f_0$  and/or  $f_1$  true.

In the following description of the micro-moves we use figures to illustrate the formal definitions. Each micro-move rewrites a playable  $\lambda$ -micro-position  $p = (G, f_0, f_1)$ . To describe the transitions, let  $G = (N, A, \Lambda, \tau, \epsilon)$ .



**Decision** Assume  $G$  has a  $\lambda$ -source  $v$ . Let  $F$  be a non-empty subset of  $\{(v, w) \mid (v, w) \in A \wedge \tau(v, w) = \mathbf{normal}\}$ . If we then let  $G' = (N, A, \Lambda, \tau', \epsilon)$ , where  $\tau'$  is the same as  $\tau$  except that  $\tau'(a) = \mathbf{focused}$  if  $a \in F$ , we have the labelled transition  $(G, f_0, f_1) \xrightarrow{D} (G', f_0, f_1)$ .



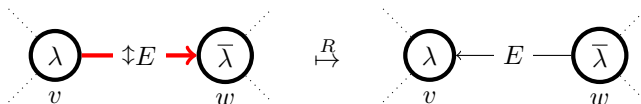
Let us give an informal description of this transition. Recall the decision rules  $[D\Downarrow]$  and  $[D\Uparrow]$  in Figure 3.1).  $[D\Downarrow]$  is applied to a sequent of the form  $\vdash \mathcal{L}; \mathcal{P}$ ; where  $\mathcal{P}$  is not empty. In  $G$ , these sequents exactly correspond to the sources, and the transition corresponds exactly to applying  $[D\Downarrow]$  to one of them, while applying  $[D\Uparrow]$  to some of its neighbours.

To describe the next five labelled transitions, assume  $G$  has a focused arc  $a = (v, w)$ . Since  $p$  is a  $\lambda$ -micro-position,  $v$  belongs to player  $\lambda$  and  $w$  belongs to player  $\bar{\lambda}$ .

**Reaction** If  $\epsilon(a)$  is of the form  $\Downarrow E$ , then one can remove the leading  $\Downarrow$ , reverse the arc, and unfocus it. Formally, let  $\bar{a} = (w, v)$  be the opposite arc to  $a$  and let

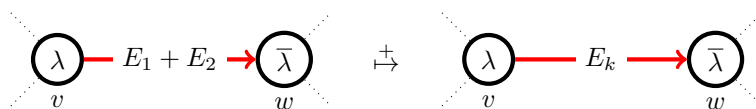
$$G' = (N, (A \setminus \{a\}) \cup \{\bar{a}\}, \Lambda, \tau_{|A \setminus \{a\}} \cup \{(\bar{a}, \mathbf{normal})\}, \epsilon_{|A \setminus \{a\}} \cup \{(\bar{a}, E)\}).$$

Then we have the transition  $(G, f_0, f_1) \xrightarrow{R} (G', f_0, f_1)$ .



In both interpretations, a formula of the wrong polarity is reclassified.

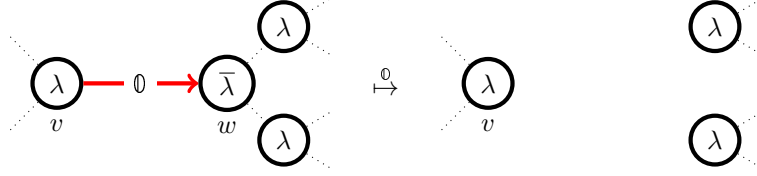
**Additives** If  $\epsilon(a)$  is of the form  $E_1 + E_2$ , then one can replace this expression with one of the operands. Formally, let  $G' = (N, A, \Lambda, \tau, \epsilon')$  where  $\epsilon'$  is the same as  $\epsilon$  except that  $\epsilon'(a) = E_k$  for some  $k \in \{1, 2\}$ . We then have the labelled transition  $(G, f_0, f_1) \xrightarrow{\pm} (G', f_0, f_1)$ .



for some  $k \in \{1, 2\}$ . This treatment of  $+$  is essentially the same as in the additive game presented before. It corresponds exactly to the reduction of a cut link between two MALL proof structures with boxes with conclusions  $[E_1]^+ \oplus [E_2]^+$  and  $[E_1]^- \& [E_2]^-$ : player  $\lambda$ 's choice between  $E_1$  and  $E_2$  corresponds to that between  $[E_1]^+$  and  $[E_2]^+$ .

If  $\epsilon(a) = \mathbf{0}$  (the 0-ary additive), then one can remove  $w$  and all its adjacent arcs. Formally, let  $G'$  be the graph obtained from  $G$  by removing all the arcs

connected to  $w$  and then all the isolated nodes (including  $w$ ), and let  $f'_0$  and  $f'_1$  be the Boolean values defined as follows:  $f'_\lambda = \top$  and  $f'_{\bar{\lambda}} = f_{\bar{\lambda}}$ . Then we have the labelled transition  $(G, f_0, f_1) \xrightarrow{0} (G', f'_0, f'_1)$ .

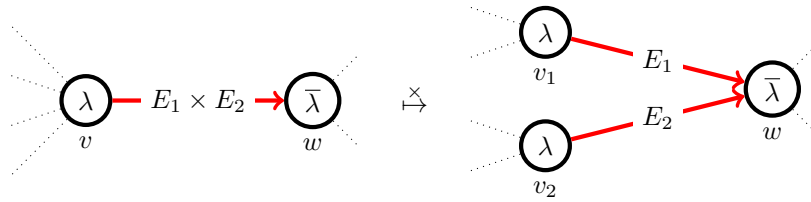


(in the second graph, any isolated node shall be removed.) This last transition is particular: on player  $\bar{\lambda}$ 's side we simply remove a sequent of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \top, \mathcal{F}$ , in other words we apply  $[\top]$ ; on player  $\lambda$ 's side we face an unprovable sequent of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \downarrow 0, \mathcal{F}$  and we must apply  $[\boxtimes]$ . Consequently player  $\lambda$  fails ( $f'_\lambda = \top$ ). Notice that this micro-move may disconnect the neutral graph.

**Multiplicatives** If  $\epsilon(a)$  is of the form  $E_1 \times E_2$ , then one can split  $v$  into two nodes and  $a$  into two arcs, labelling each one with an operand. Formally, define two new nodes  $v_1$  and  $v_2$  and for every  $b = (t, u) \in A \setminus \{a\}$ , define an arc  $b'$  as follows: if  $t \neq v$  and  $u \neq v$ , then  $b' = b$ ; if  $t = v$ , then  $b' = (v_i, u)$  for some  $i \in \{1, 2\}$ ; and if  $u = v$ , then  $b' = (t, v_i)$  for some  $i \in \{1, 2\}$ . Now let  $G' = (N', A', \Lambda', \tau', \epsilon')$  where

- $N' = (N \setminus \{v\}) \uplus \{v_1, v_2\}$ ,
- $A' = \{(v_1, w), (v_2, w)\} \cup \{b' | b \in A \setminus \{a\}\}$ ,
- $\Lambda' = \Lambda_{|V \setminus \{v\}} \cup \{(v_1, \lambda), (v_2, \lambda)\}$ ,
- $\tau'(v_1, w) = \tau'(v_2, w) = \text{focused}$ , and for every  $b \in A \setminus \{a\}$ ,  $\tau'(b') = \tau(b)$ ,
- $\epsilon'(v_1, w) = E_1$  and  $\epsilon'(v_2, w) = E_2$ , and for every  $b \in A \setminus \{a\}$ ,  $\epsilon'(b') = \epsilon(b)$ ,

We then have the labelled transition  $(G, f_0, f_1) \xrightarrow{\times} (G', f'_0, f'_1)$ .



On player  $\lambda$ 's side, the splitting corresponds to that of the  $[\otimes]$  rule. On player  $\bar{\lambda}$ 's side the invertible  $[\wp]$  rule is applied. Here again, this transition is exactly the reduction of a cut link between two proof structures with conclusions  $[E_1]^+ \otimes [E_2]^+$  and  $[E_1]^- \wp [E_2]^-$ .

If  $\epsilon(a) = \mathbb{1}$  (the 0-ary multiplicative), then one can remove  $a$ . Formally, let  $G'$  be the neutral graph obtained from  $G$  by removing  $a$  and then the isolated nodes, and let  $f'_0$  and  $f'_1$  be Boolean values defined as follows:

$$f'_\lambda = \begin{cases} \top & \text{if } v \text{ is a node of } G' \\ f_\lambda & \text{otherwise} \end{cases} \quad f'_{\bar{\lambda}} = \begin{cases} f_{\bar{\lambda}} & \text{if } w \text{ is a node of } G' \\ \top & \text{otherwise} \end{cases}$$

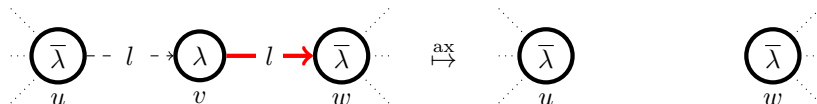
Then we have the labelled transition  $(G, f_0, f_1) \xrightarrow{\perp} (G', f'_0, f'_1)$ .



(in the second graph, any isolated node shall be removed.) In this transition both players may fail. On player  $\lambda$ 's side the transition corresponds to applying  $[1]$ . The sequent associated to  $v$  should thus be  $\vdash; \Downarrow 1$ , therefore the player fails ( $f'_\lambda = \top$ ) if 1 is not the only formula of the sequent. On player  $\bar{\lambda}$ 's side  $[\perp]$  is applied, and if  $w$  is only connected to  $v$  then its associated sequent becomes  $\vdash; \uparrow \perp$  which is unprovable, and the player fails ( $f'_\lambda = \top$ ). Notice that this micro-move may disconnect the neutral graph.

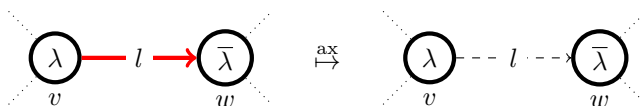
**Atoms** If  $\epsilon(a)$  is an atom  $l$ , then the sequent associated to  $v$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow L^+, \mathcal{F}$ . There are two cases, depending on whether the rule  $[\text{init}]$  may be applied or not.

First case: if there is exactly one arc  $b$  connected to  $v$  beside  $a$ , and  $b$  is of the form  $(u, v)$ , with  $\tau(b) = \text{atomic}$  and  $\epsilon(b) = l$ , then one may remove  $v$ ,  $a$  and  $b$ . Formally, let  $G'$  be the neutral graph obtained from  $G$  by removing  $a$ ,  $b$ , then the isolated nodes (including  $v$ ). Then we have the labelled transition  $(G, f_0, f_1) \xrightarrow{\text{ax}} (G', f_0, f_1)$ .



(in the second graph, any isolated node shall be removed.) In this case the sequent associated to  $v$  is  $\vdash L^-; \Downarrow L^+$  and  $[\text{init}]$  may be applied, which is reflected by the transition. On player  $\bar{\lambda}$ 's side the transition corresponds to applying  $[\boxtimes]$  to the sequents associated with  $u$  and  $w$ . This can be done safely as player  $\bar{\lambda}$  has already failed ( $f_{\bar{\lambda}} = \top$ ) since  $u$  is the origin of an atomic arc. Notice that this micro-move may disconnect the neutral graph.

Second case: if the first case does not apply, then one may make  $a$  atomic and make the player fail in the process. Formally, let  $G' = (N, A, \Lambda, \tau', \epsilon)$ , where  $\tau'$  is the same as  $\tau$  except that  $\tau'(a) = \text{atomic}$ , and let  $f'_0$  and  $f'_1$  be Boolean values defined as follows:  $f'_\lambda = \top$  and  $f'_{\bar{\lambda}} = f_{\bar{\lambda}}$ . Then we have the labelled transition  $(G, f_0, f_1) \xrightarrow{\text{ax}} (G', f_0, f_1)$ .



In this case the sequent associated to  $v$  is not  $\vdash L^-; \Downarrow L^+$ ,  $[\text{init}]$  may not be applied, and the player applies  $[\boxtimes]$  and fails. On player  $\bar{\lambda}$ 's side the transition corresponds to applying  $[\text{atomic R } \uparrow]$  to the sequent associated with  $w$ .

**Definition 3.6.7** ( $\lambda$ -micro-move). *Let  $\lambda \in \{0, 1\}$  and  $p, p'$  be  $\lambda$ -micro-positions. There is a  $\lambda$ -micro-move from  $p$  to  $p'$  (notation  $p \mapsto_\lambda p'$ ) iff one of the following holds:  $p \xrightarrow{D} p'$ ,  $p \xrightarrow{R} p'$ ,  $p \xrightarrow{+} p'$ ,  $p \xrightarrow{0} p'$ ,  $p \xrightarrow{\times} p'$ ,  $p \xrightarrow{1} p'$ , or  $p \xrightarrow{ax} p'$ .*

**Proposition 3.6.8.** *There is a  $\lambda$ -micro-move from a  $\lambda$ -micro-position iff it is playable.*

*Proof.* A  $\lambda$ -micro-position is playable iff its neutral graph has a source or a focused arc. A  $\lambda$ -micro-move  $\xrightarrow{D}$  is possible iff the neutral graph has a source. A  $\lambda$ -micro-move  $\xrightarrow{R}$ ,  $\xrightarrow{+}$ ,  $\xrightarrow{0}$ ,  $\xrightarrow{\times}$ ,  $\xrightarrow{1}$ , or  $\xrightarrow{ax}$  is possible iff the neutral graph has a focused arc (those moves cover all the cases for the neutral expression labelling the arc).  $\square$

### 3.6.3 Macro-moves

We proceed to define the macro-moves, which are the actual moves of the game, as maximal sequences of micro-moves.

**Proposition 3.6.9.** *The length of the sequences of micro-moves starting in a fixed micro-position is bounded.*

*Proof.* Associate a triple  $(s, n, f)$  with each micro-position  $(G, f_0, f_1)$ , where  $s$  is the total number of symbols of the neutral expressions labelling the arcs of  $G$ ,  $n$  is the number of normal arcs of  $G$ , and  $f$  is the number of focused arcs of  $G$ . Every micro-move decreases this triple for the lexicographical ordering. Moreover, these triples verify  $n \leq s$  and  $f \leq s$ .  $\square$

**Definition 3.6.10** (Macro-move). *Let  $p$  be a  $\lambda$ -macro-position and  $p \mapsto_\lambda^* p'$  a maximal sequence of  $\lambda$ -micro-moves from  $p$ . Then  $p'$  is either a  $\bar{\lambda}$ -macro-position or a final macro-position. We say that there is a macro-move from  $p$  to  $p'$  and denote it by  $p \rightarrow p'$ .*

*Proof.*  $p'$  is a  $\lambda$ -micro-position, but it is not playable by Proposition 3.6.8.  $p'$  has no focused arcs and all its sources belong to player  $\bar{\lambda}$ , which makes it a  $\bar{\lambda}$ -macro-position or a final macro-position.  $\square$

Notice that there is a macro-move from every non-final macro-position.

**Proposition 3.6.11.** *The length of the plays starting in a fixed macro-position is bounded.*

*Proof.* Every macro-move expands to a sequence of micro-moves which is non-empty since every non-final  $\lambda$ -macro-position is a playable  $\lambda$ -micro-position. The result thus follows from Proposition 3.6.9.  $\square$

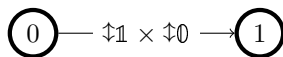
## 3.7 Examples

In this section we illustrate the formal definitions of the game with examples.

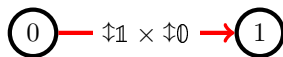
### 3.7.1 Indeterminacy

The dual formulae  $\perp \otimes \top$  and  $1 \wp 0$  are unprovable. The game based on the neutral expression representing that pair,  $\Downarrow 1 \times \Downarrow 0$ , is indeterminate as expected. We describe the plays here.

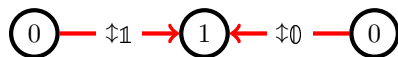
The starting macro-position consists of the neutral graph



along with the two Boolean flags being false, to denote that no player has failed yet. Player 0 aims at proving  $\perp \otimes \top$  and player 1 aims at proving  $1 \wp 0$ . In that macro-position, the only source is the 0-node, and player 0's first micro-move must be  $\overset{D}{\mapsto}$ , where she focuses on the only arc



The second micro-move must be to decompose the multiplicative with  $\overset{\times}{\mapsto}$



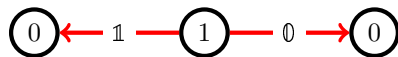
The two last micro-moves of the macro-move release the foci with  $\overset{R}{\mapsto}$



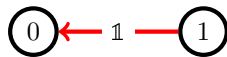
To sum up, this 0-macro-move is seen as the derivations

Player 0	Player 1
$\frac{\frac{\frac{\vdash; \perp}{\vdash; \Downarrow \perp} [\text{R}\Downarrow] \quad \frac{\vdash; \top}{\vdash; \Downarrow \top} [\text{R}\Downarrow]}{\vdash; \Downarrow \perp \otimes \top} [\otimes] \quad \frac{}{\vdash; \perp \otimes \top} [\text{D}\Downarrow]}{\vdash; \perp \otimes \top} [\otimes]$	$\frac{\frac{\frac{\vdash; 1, 0;}{\vdash; \Uparrow 1, 0} [\text{R}\Uparrow]}{\vdash; \Uparrow 1 \wp 0} [\wp]}{\vdash; 1 \wp 0} [\text{D}\Uparrow]}$

The only source is now the 1-node, and player 1 is the next to play. Her first micro-move is  $\overset{D}{\mapsto}$  and she has three choices: focus on 1 alone, on 0 alone, or on both. We present the latter; the others are similar. The neutral graph becomes



and the next micro-move is either  $\overset{1}{\mapsto}$  or  $\overset{0}{\mapsto}$ . If it is  $\overset{0}{\mapsto}$ , player 1 fails and the neutral graph becomes



At that point player 1 plays  $\overset{1}{\mapsto}$  to obtain the empty neutral graph, causing player 0 to fail (because the 0-node becomes isolated and is removed). The play is a tie since both players failed. To sum up, this 1-macro-move is seen as the derivations

Player 0			Player 1	
$\frac{\overline{\vdash; \uparrow \perp}}{\vdash; \perp} \quad [\boxtimes]$	$\frac{\overline{\vdash; \uparrow \top}}{\vdash; \top} \quad [\top]$		$\frac{\overline{\vdash; \downarrow 1}}{\vdash; \downarrow 1, 0} \quad [1]$	$\frac{\overline{\vdash; \downarrow 1, 0}}{\vdash; 1, 0; } \quad [\boxtimes]$
$\frac{}{\vdash; \perp} \quad [\text{D}\uparrow]$	$\frac{}{\vdash; \top} \quad [\text{D}\uparrow]$		$\frac{}{\vdash; \downarrow 1, 0} \quad [\text{D}\downarrow]$	$\frac{}{\vdash; 1, 0; } \quad [\text{D}\downarrow]$

In fact, it can be easily seen that other choices for player 1's micro-moves would have given the same outcome: an empty neutral graph and failure for both players. Therefore, no player has a winning strategy, as expected.

### 3.7.2 Atoms

The next example involves atomic neutral expressions. Consider the game on the neutral expression  $(a_1 + a_2) \times \downarrow(a_1 + a_2)$ , where  $a_1$  and  $a_2$  are two neutral atoms. The negative translation  $(A_1^- \& A_2^-) \wp (A_1^+ \oplus A_2^+)$  of the expression is provable. Starting from

$$\textcircled{0} \text{---} (a_1 + a_2) \times \downarrow(a_1 + a_2) \text{---} \textcircled{1}$$

player 0 applies  $\xrightarrow{D}$  and  $\xrightarrow{\boxtimes}$  to yield

$$\textcircled{0} \text{---} a_1 + a_2 \text{---} \textcircled{1} \text{---} \downarrow(a_1 + a_2) \text{---} \textcircled{0}$$

then she may apply  $\xrightarrow{\perp}$  and  $\xrightarrow{R}$

$$\textcircled{0} \text{---} a_i \text{---} \textcircled{1} \text{---} a_1 + a_2 \text{---} \textcircled{0}$$

for some  $i \in \{1, 2\}$ ; then she has to apply  $\xrightarrow{\text{ax}}$  which makes her fail

$$\textcircled{0} \text{---} \dots a_i \text{---} \textcircled{1} \text{---} a_1 + a_2 \text{---} \textcircled{0}$$

Notice that micro-moves could have been scheduled otherwise without changing the outcome of this macro-move. This 0-macro-move is seen as

$$\frac{\frac{\frac{\vdash A_i^+;}{\vdash; \downarrow A_i^+} \quad [\boxtimes]}{\vdash; \downarrow A_1^+ \oplus A_2^+} \quad [\oplus_i] \quad \frac{\vdash; A_1^- \& A_2^-}{\vdash; \downarrow A_1^- \& A_2^-} \quad [\text{R}\downarrow]}{\frac{\vdash; \downarrow (A_1^+ \oplus A_2^+) \otimes (A_1^- \& A_2^-)}{\vdash; (A_1^+ \oplus A_2^+) \otimes (A_1^- \& A_2^-)} \quad [\otimes]} \quad [\text{D}\downarrow]$$

by player 0 and as

$$\frac{\frac{\frac{\vdash A_i^-; A_1^+ \oplus A_2^+}{\vdash; A_1^+ \oplus A_2^+; \uparrow A_i^-} \quad [\text{atomic R}\uparrow]}{\vdash; \uparrow A_1^- \& A_2^-, A_1^+ \oplus A_2^+} \quad [\&_i, \text{R}\uparrow]}{\frac{\vdash; \uparrow (A_1^- \& A_2^-) \wp (A_1^+ \oplus A_2^+)}{\vdash; (A_1^- \& A_2^-) \wp (A_1^+ \oplus A_2^+)} \quad [\wp]} \quad [\text{D}\uparrow]$$

by player 1.

Then the only source is the 1-node and all player 1 has to do is to apply  $\overset{D}{\mapsto}$  to focus on  $a_1 + a_2$ , then  $\overset{\dagger}{\mapsto}$  to select  $a_i$ , and finally  $\overset{\text{ax}}{\mapsto}$  to get the empty neutral graph and win the play. This 1-macro-move is seen as

Player 0	Player 1
$\frac{}{\vdash A_i^+;;} \text{✕} \quad \frac{\frac{}{\vdash;; \uparrow A_i^-} \text{✕}}{\vdash;; \uparrow A_1^- \& A_2^-} \&_i}{\vdash;; A_1^- \& A_2^-} [\text{D}\uparrow]$	$\frac{}{\vdash A_i^-;; \downarrow A_i^+} [\text{init}] \quad \frac{}{\vdash A_i^-;; \downarrow A_1^+ \oplus A_2^+} [\oplus_i]}{\vdash A_i^-;; A_1^+ \oplus A_2^+} [\text{D}\downarrow]$

Notice that during her macro-move, player 0 may choose any  $i \in \{1, 2\}$ . We showed that player 1 could win the play in both cases. As expected, player 1—the one seeing the provable translation of the neutral expression—has a winning strategy.

### 3.8 Winning strategies and cut-free proofs

In this section we relate the provability of a formula to the existence of a winning strategy in the game. Our proofs effectively show how to construct a winning strategy from a proof. They also show how to construct a proof from a winning strategy, but this construction is not unique in general. Establishing a correspondence between winning strategies and proofs requires a more involved game model and is covered in the next chapter.

The operators  $[\cdot]^+$  and  $[\cdot]^-$  are applied to multisets of neutral expressions in the obvious way. Two focused proofs of the same sequent are equivalent iff they differ by the order in which asynchronous rules are applied within asynchronous phases. This is indeed an equivalence relation.

We begin by formally defining a central notion relating concepts of the game to concepts of the proof system: that of  $\lambda$ -provability (for  $\lambda \in \{0, 1\}$ ).

**Definition 3.8.1** ( $\lambda$ -provability). *Let  $G$  be a neutral graph and  $\lambda \in \{0, 1\}$ .  $G$  is  $\lambda$ -provable iff the sequents associated with its  $\lambda$ -nodes are all provable. A triple  $(G, f_0, f_1)$  where  $G$  is a neutral graph and  $f_0, f_1$  are Boolean values is  $\lambda$ -provable iff  $f_\lambda = \perp$  and  $G$  is  $\lambda$ -provable.*

We relate game moves to derivations by proceeding gradually from small steps (micro-moves and inference rules) to large objects (winning strategies and proofs).

**Proposition 3.8.2.** *Let  $p$  be a playable  $\lambda$ -micro-position. Let  $S = \{p' \mid p \mapsto_\lambda p'\}$ .  $p$  is  $\lambda$ -provable iff there exists  $p' \in S$  which is  $\lambda$ -provable.*

*Proof.* Let us write  $p = (G, f_0, f_1)$  and  $G = (N, A, \Lambda, \tau, \epsilon)$ . We prove this result in two parts: (1) the “if” part, (2) the “only if” part.

(1) Suppose that there exists  $p' = (G', f'_0, f'_1) \in S$  which is  $\lambda$ -provable. Let us show that  $p$  is  $\lambda$ -provable. We have  $f'_\lambda = \perp$  and  $G'$  is  $\lambda$ -provable. Let us write  $G' = (N', A', \Lambda', \tau', \epsilon')$ . We examine the cases for  $p \mapsto_\lambda p'$ .

Case  $p \xrightarrow{D} p'$ . Thus  $f_\lambda = f'_\lambda = \perp$  and, moreover, the only  $\lambda$ -sequent affected by the move is the one associated with the source  $v$  selected for the move. In  $G'$ , this sequent is of the form  $\vdash \mathcal{L}; \mathcal{P}_1; \Downarrow \mathcal{P}_2$ , and in  $G$  it is  $\vdash \mathcal{L}; \mathcal{P}_1, \mathcal{P}_2$ . The result follows from the derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}_1; \Downarrow \mathcal{P}_2}{\vdash \mathcal{L}; \mathcal{P}_1, \mathcal{P}_2} [\text{D}\Downarrow]$$

In each one of the other cases we consider the focused arc  $a = (v, w)$  from the definition of the corresponding transition. First of all, the case  $p \xrightarrow{0} p'$  does not happen, since  $f'_\lambda = \perp$ . In the other cases  $v$  is the only  $\lambda$ -node of  $G$  affected by the transition, hence all we need to show is that  $f_\lambda = \perp$  and  $\Sigma_G(v)$  is provable. For the latter, showing that  $\Sigma_G(v)$  derives from sequents associated with  $\lambda$ -nodes of  $G'$  (which are provable) is enough.

Case  $p \xrightarrow{R} p'$ . Thus  $f_\lambda = f'_\lambda = \perp$  and, moreover,  $\Sigma_G(v)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F, \mathcal{F}$ , where  $F = [\epsilon(a)]^+$ . Remark that  $[\epsilon'(w, v)]^- = F$  and that  $F$  is negative; it is then clear that  $\Sigma_{G'}(v) = \vdash \mathcal{L}; \mathcal{P}; F, \mathcal{N} \Downarrow \mathcal{F}$  and the result follows from the derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}; F, \mathcal{N} \Downarrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F, \mathcal{F}} [\text{R}\Downarrow]$$

Case  $p \xrightarrow{\oplus} p'$ . Thus  $f_\lambda = f'_\lambda = \perp$  and, moreover,  $[\epsilon(a)]^+$  is of the form  $F_1 \oplus F_2$ .  $\Sigma_G(v)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F_1 \oplus F_2, \mathcal{F}$  and  $\Sigma_{G'}(v) = \vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F_i, \mathcal{F}$ , for some  $i \in \{1, 2\}$ . The result follows from the derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F_i, \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F_1 \oplus F_2, \mathcal{F}} [\oplus_i]$$

Case  $p \xrightarrow{\otimes} p'$ . Thus,  $f_\lambda = f'_\lambda = \perp$  and, moreover,  $[\epsilon(a)]^+$  is of the form  $F_1 \otimes F_2$ . The node  $v$  is replaced in  $G'$  with two  $\lambda$ -nodes  $v_1$  and  $v_2$ .  $\Sigma_{G'}(v_1)$  and  $\Sigma_{G'}(v_2)$  are of the forms  $\vdash \mathcal{L}_1; \mathcal{P}_1; \mathcal{N}_1 \Downarrow F_1, \mathcal{F}_1$  and  $\vdash \mathcal{L}_2; \mathcal{P}_2; \mathcal{N}_2 \Downarrow F_2, \mathcal{F}_2$ , and we have  $\Sigma_G(v) = \vdash \mathcal{L}_1, \mathcal{L}_2; \mathcal{P}_1, \mathcal{P}_2; \mathcal{N}_1, \mathcal{N}_2 \Downarrow F_1 \otimes F_2, \mathcal{F}_1, \mathcal{F}_2$ . The result follows from the derivation

$$\frac{\vdash \mathcal{L}_1; \mathcal{P}_1; \mathcal{N}_1 \Downarrow F_1, \mathcal{F}_1 \quad \vdash \mathcal{L}_2; \mathcal{P}_2; \mathcal{N}_2 \Downarrow F_2, \mathcal{F}_2}{\vdash \mathcal{L}_1, \mathcal{L}_2; \mathcal{P}_1, \mathcal{P}_2; \mathcal{N}_1, \mathcal{N}_2 \Downarrow F_1 \otimes F_2, \mathcal{F}_1, \mathcal{F}_2} [\otimes]$$

Case  $p \xrightarrow{1} p'$ . Thus,  $[\epsilon(a)]^+ = 1$ . Since  $f'_\lambda = \perp$ ,  $f_\lambda = \perp$  and  $v$  is not a node of  $G'$ .  $a$  is therefore the only arc connected to it in  $G$  and  $\Sigma_G(v) = \vdash ; ; \Downarrow 1$  is provable:

$$\overline{\vdash ; ; \Downarrow 1} [1]$$

Case  $p \xrightarrow{\text{ax}} p'$ . Thus,  $[\epsilon(a)]^+ = L^+$ . Since  $f'_\lambda = \perp$ , the first case in the definition of  $\xrightarrow{\text{ax}}$  applies,  $f_\lambda = f'_\lambda = \perp$  and  $\Sigma_G(v) = \vdash L^- ; ; \Downarrow L^+$  is provable:

$$\overline{\vdash L^- ; ; \Downarrow L^+} [\text{init}]$$

(2) This is the converse to the previous part. Suppose that  $p = (G, f_0, f_1)$  is  $\lambda$ -provable. We have  $f_\lambda = \perp$  and  $G$  is  $\lambda$ -provable. We will consider several cases, and in each one we will show that there is a  $\lambda$ -provable  $p' = (G', f'_0, f'_1) \in S$ . There will be few  $\lambda$ -nodes affected by the transition  $p \mapsto_\lambda p'$ , hence it will be



enough to show that  $f'_\lambda = \perp$  and that the sequents associated to those nodes are provable. Since  $p$  is a playable  $\lambda$ -micro-position,  $G$  has a source (belonging to player  $\lambda$ ) or a focused arc (whose origin belongs to player  $\lambda$ ).

First case. Suppose that  $G$  has a  $\lambda$ -source  $v$ .  $\Sigma_G(v)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}$ ; with  $\mathcal{P} \neq \emptyset$  and it is provable by assumption, and a proof must end with the  $[D\Downarrow]$  rule:

$$\frac{\vdash \mathcal{L}; \mathcal{P}_1; \Downarrow \mathcal{P}_2}{\vdash \mathcal{L}; \mathcal{P}_1, \mathcal{P}_2; } [D\Downarrow]$$

where  $\mathcal{P}_1$  and  $\mathcal{P}_2$  partition  $\mathcal{P}$ ,  $\mathcal{P}_2$  is not empty and  $\vdash \mathcal{L}; \mathcal{P}_1; \Downarrow \mathcal{P}_2$  is provable. This corresponds to a transition  $p \xrightarrow{D} p' = (G', f'_0, f'_1)$  in which the outgoing normal arcs of  $v$  corresponding to  $\mathcal{P}_2$  become focused. The only affected  $\lambda$ -node is  $v$  and  $\Sigma_{G'}(v)$  is precisely  $\vdash \mathcal{L}; \mathcal{P}_1; \Downarrow \mathcal{P}_2$ . Moreover  $f'_\lambda = f_\lambda = \perp$ , therefore  $p'$  is  $\lambda$ -provable.

Second case. Suppose that  $G = (N, A, \Lambda, \tau, \epsilon)$  has a focused arc, and let  $v$  be its origin.  $\Sigma_G(v)$  is provable. Consider the last rule  $R$  of such a proof.  $\Sigma_G(v)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \Downarrow F, \mathcal{F}$ , where  $F$  is the principal formula of  $R$ . There is a focused arc  $a = (v, w)$  for some  $w$  such that  $[\epsilon(a)]^+ = F$ . We are going to consider all the cases for  $\epsilon(a)$ . In each case we show that there is  $p' = (G', f'_0, f'_1) \in S$  which is  $\lambda$ -provable. More specifically, we show that  $f'_\lambda = \perp$  and that the sequents associated to the  $\lambda$ -nodes of  $G'$  affected by the transition are in fact the premises of  $R$ , and are therefore provable.

Case  $\epsilon(a) = \Downarrow E$ . Thus  $F$  is negative and  $R$  is  $[R\Downarrow]$ . As we did in part (1), we match this rule with the transition  $p \xrightarrow{R} (G', f_0, f_1)$  in which  $a$  is reversed. The only affected  $\lambda$ -node is  $v$ . Since  $F = [E]^-$ ,  $\Sigma_{G'}(v)$  is precisely the premise of  $R$ .

Case  $\epsilon(a) = E_1 + E_2$ . Thus  $F = [E_1]^+ \oplus [E_2]^+$  and  $R$  is  $[\oplus_i]$  for some  $i \in \{1, 2\}$ . As we did in part (1), we match this rule with the transition  $p \xrightarrow{\oplus} (G', f_0, f_1)$  in which  $\epsilon(a)$  is replaced with  $E_i$ . The only affected  $\lambda$ -node is  $v$ , and  $\Sigma_{G'}(v)$  is precisely the premise of  $R$ .

Case  $\epsilon(a) = 0$ . Thus,  $F = 0$ . This case does not happen, since there is no introduction rule for 0.

Case  $\epsilon(a) = E_1 \times E_2$ . Thus  $F = [E_1]^+ \otimes [E_2]^+$  and  $R$  is  $[\otimes]$ . As we did in part (1), we match this rule with a transition  $p \xrightarrow{\otimes} (G', f_0, f_1)$ . The only affected  $\lambda$ -node is  $v$  which is split into  $v_1$  and  $v_2$ , and  $\Sigma_{G'}(v_1)$  and  $\Sigma_{G'}(v_2)$  are precisely the premises of  $R$ .

Case  $\epsilon(a) = \mathbb{1}$ . Thus  $F = 1$  and  $R$  is  $[1]$ . It means that  $\Sigma_G(v) = \vdash; \Downarrow 1$ . As we did in part (1), we match this rule with the transition  $p \xrightarrow{1} (G', f'_0, f'_1)$  in which  $a$  is removed, and it is clear that  $v$  is not a node of  $G'$ , therefore  $f'_\lambda = f_\lambda = \perp$  as needed. There are no new/affected  $\lambda$ -nodes in  $G'$ .

Case  $\epsilon(a) = l$ . Thus  $F = L^+$  and  $R$  is  $[\text{init}]$ . It means that  $\Sigma_G(v) = \vdash L^-; \Downarrow \oplus L$ . As we did in part (1), we match this rule with the transition  $p \xrightarrow{\text{ax}} (G', f'_0, f'_1)$  in which  $v$  is removed (first case of the definition of  $\text{ax}$ ). Then  $f'_\lambda = f_\lambda = \perp$  as needed. There are no new/affected  $\lambda$ -nodes in  $G'$ .  $\square$

**Proposition 3.8.3.** *Let  $\lambda \in \{0, 1\}$  and  $p$  be a playable  $\lambda$ -micro-position. Let  $S = \{p' | p \mapsto_\lambda p'\}$ .  $p$  is  $\bar{\lambda}$ -provable iff every  $p' \in S$  is  $\bar{\lambda}$ -provable.*

*Proof.* Let us write  $p = (G, f_0, f_1)$  and  $G = (N, A, \Lambda, \tau, \epsilon)$ . We are going to prove both directions simultaneously. Let us give names to the two hypotheses:

(A)  $p$  is  $\bar{\lambda}$ -provable, and (B) every  $p' \in S$  is  $\bar{\lambda}$ -provable. Since  $p$  is playable,  $S$  is not empty. Let  $p' = (G', f'_0, f'_1) \in S$ . We examine the cases for  $p \mapsto_{\lambda} p'$ . In each case we show that (A) implies that  $p'$  is  $\bar{\lambda}$ -provable, and that (B) implies (A). This will prove the proposition, since we chose  $p' \in S$  arbitrarily and we cover all the cases for  $p \mapsto_{\lambda} p'$ . Moreover, it will be enough to consider only the  $\bar{\lambda}$ -nodes affected by the transition.

Case  $p \xrightarrow{D} p'$ . Let  $v$  be the source of  $G$  which is selected for the move. Every  $\bar{\lambda}$ -node  $w$  affected by the move is such that the arc  $(v, w)$  is normal in  $G$  and becomes focused in the transition. In other words,  $\Sigma_G(w)$  and  $\Sigma_{G'}(w)$  are of the form  $\vdash \mathcal{L}; \mathcal{P}; F, \mathcal{N} \uparrow \mathcal{F}$  and  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F, \mathcal{F}$ . The derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F, \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; F, \mathcal{N} \uparrow \mathcal{F}} [\text{D}\uparrow]$$

shows that if (B), then each  $\Sigma_{G'}(w)$  has a proof, hence so does each  $\Sigma_G(w)$ ; moreover  $f'_{\bar{\lambda}} = \perp$ , hence  $f_{\bar{\lambda}} = \perp$  and (A). Conversely, if (A), then each  $\Sigma_G(w)$  has a proof, which is equivalent to a proof ending with the above derivation, hence each  $\Sigma_{G'}(w)$  is provable; moreover  $f_{\bar{\lambda}} = \perp$ , hence  $f'_{\bar{\lambda}} = \perp$  and  $p'$  is  $\bar{\lambda}$ -provable.

For each one of the other cases we consider the arc  $a = (v, w)$  from the definition of the corresponding transition.

Case  $p \xrightarrow{\text{ax}} p'$ . There are two cases in the definition of this transition. In the first case, we must have  $f_{\bar{\lambda}} = \top$  since the  $\bar{\lambda}$ -node  $u$  is the origin of the atomic arc  $(u, v)$ . Then  $f'_{\bar{\lambda}} = \top$  and (A) and (B) are both false, which concludes this case. In the second case, the only affected  $\bar{\lambda}$ -node is  $w$ , and  $\Sigma_G(w)$  and  $\Sigma_{G'}(w)$  are of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow L^-, \mathcal{F}$  and  $\vdash L^-, \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}$ . The derivation

$$\frac{\vdash \uparrow L, \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow L^-, \mathcal{F}} [\text{atomic R}\uparrow]$$

shows that if (B), then  $\Sigma_{G'}(w)$  has a proof, hence so does  $\Sigma_G(w)$ ; moreover  $f'_{\bar{\lambda}} = \perp$ , hence  $f_{\bar{\lambda}} = \perp$  and (A). Conversely, if (A), then  $\Sigma_G(w)$  has a proof, which is equivalent to a proof ending with the above derivation, hence  $\Sigma_{G'}(w)$  is provable; moreover  $f_{\bar{\lambda}} = \perp$ , hence  $f'_{\bar{\lambda}} = \perp$  and  $p'$  is  $\bar{\lambda}$ -provable.

In all the remaining cases the only  $\bar{\lambda}$ -node affected by the transition is  $w$ .

Case  $p \xrightarrow{R} p'$ .  $\epsilon(a)$  is of the form  $\dagger E$ .  $\Sigma_G(w)$  and  $\Sigma_{G'}(w)$  are of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F, \mathcal{F}$  and  $\vdash \mathcal{L}; F, \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}$  with  $F = [\dagger E]^- = [E]^+$ . The derivation

$$\frac{\vdash \mathcal{L}; F, \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F, \mathcal{F}} [\text{R}\uparrow]$$

allows us to conclude as in the previous cases.

Case  $p \xrightarrow{\oplus} p'$ .  $\epsilon(a)$  is of the form  $E_1 \oplus E_2$  and a choice is made between  $E_1$  and  $E_2$  in  $P'$ . Consider both choices. They lead to two elements of  $S$ ,  $(G_1, f_0, f_1)$  and  $(G_2, f_0, f_1)$  (one of them is  $p'$ ). Let us write  $G_i = (N', A', \Lambda', \tau', \epsilon'_i)$  for each  $i \in \{1, 2\}$ .  $[\epsilon(a)]^-$  is of the form  $F_1 \& F_2$  and  $[\epsilon'_i(a)]^- = F_i$ .  $\Sigma_G(w)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1 \& F_2, \mathcal{F}$  and  $\Sigma_{G_i}(w) = \vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_i, \mathcal{F}$ . The derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1, \mathcal{F} \quad \vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_2, \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1 \& F_2, \mathcal{F}} [\&]$$

shows that if (B), then  $\Sigma_{G_1}(w)$  and  $\Sigma_{G_2}(w)$  have proofs, hence so does  $\Sigma_G(w)$ ; moreover  $f_{\bar{\lambda}} = \perp$ , hence (A). Conversely, if (A), then  $\Sigma_G(w)$  has a proof, which is equivalent to a proof ending with the above derivation, hence  $\Sigma_{G_1}(w)$  and  $\Sigma_{G_2}(w)$  are provable; moreover  $f_{\bar{\lambda}} = \perp$ , hence  $p'$  is  $\bar{\lambda}$ -provable.

Case  $p \xrightarrow{0} p'$ .  $\Sigma_G(w)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \top, \mathcal{F}$  and it is provable:

$$\frac{}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \top, \mathcal{F}} [\top]$$

$w$  is not a node of  $G'$  and  $f'_{\bar{\lambda}} = f_{\bar{\lambda}}$ , hence  $p$  is  $\bar{\lambda}$ -provable iff  $p'$  is. If (B), then (A). Conversely, if (A), then  $p'$  is  $\bar{\lambda}$ -provable.

Case  $p \xrightarrow{\times} p'$ .  $\Sigma_G(w)$  and  $\Sigma_{G'}(w)$  are of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1 \wp F_2, \mathcal{F}$  and  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1, F_2, \mathcal{F}$ . The derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1, F_2, \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow F_1 \wp F_2, \mathcal{F}} [\wp]$$

allows us to conclude as in the previous cases.

Case  $p \xrightarrow{1} p'$ .  $\Sigma_G(w)$  is of the form  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \perp, \mathcal{F}$ . Consider the derivation

$$\frac{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}}{\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \perp, \mathcal{F}} [\perp]$$

If (B), then  $f'_{\bar{\lambda}} = \perp$  and  $w$  is a node of  $G'$ . Then  $\Sigma_{G'}(w) = \vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}$ , it has a proof, hence so does  $\Sigma_G(w)$  by the above derivation; moreover  $f_{\bar{\lambda}} = \perp$ , hence (A). Conversely, if (A), then  $f_{\bar{\lambda}} = \perp$  and  $\Sigma_G(w)$  has a proof, which is equivalent to a proof ending with the above derivation, hence  $\vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}$  is provable;  $\mathcal{L}, \mathcal{P}, \mathcal{N}$  and  $\mathcal{F}$  cannot all be empty, therefore  $w$  is a node of  $G'$ , and  $\Sigma_{G'}(w) = \vdash \mathcal{L}; \mathcal{P}; \mathcal{N} \uparrow \mathcal{F}$  is provable; moreover  $f'_{\bar{\lambda}} = \perp$ , hence (A).  $\square$

**Lemma 3.8.4.** *Let  $p$  be a  $\lambda$ -macro-position. Let  $S = \{p' | p \rightarrow p'\}$ .  $p$  is  $\lambda$ -provable iff there exists  $p' \in S$  which is  $\lambda$ -provable.  $p$  is  $\bar{\lambda}$ -provable iff every  $p' \in S$  is  $\bar{\lambda}$ -provable.*

*Proof.* We show a more general result. For every  $\lambda$ -micro-position  $p$ , let  $S_p = \{p' | p \xrightarrow{*} p' \text{ and this sequence is maximal}\}$ . Let us show that for every  $\lambda$ -micro-position  $p$ ,  $p$  is  $\lambda$ -provable iff there exists  $p' \in S_p$  which is  $\lambda$ -provable, and  $p$  is  $\bar{\lambda}$ -provable iff every  $p' \in S_p$  is  $\bar{\lambda}$ -provable. By proposition 3.6.9, we may show it by induction on the maximal length  $l_p$  of the sequences  $p \mapsto_{\lambda} p'$  for  $p' \in S_p$ . If  $l_p = 0$ , then  $S_p = \{p\}$  and the result is trivial. Now suppose that  $l_p > 0$  and let  $S' = \{p' | p \mapsto_{\lambda} p'\}$ . By Propositions 3.8.2 and 3.8.3,  $p$  is  $\lambda$ -provable iff there exists  $p' \in S'$  which is  $\lambda$ -provable, and  $p$  is  $\bar{\lambda}$ -provable iff every  $p' \in S'$  is  $\bar{\lambda}$ -provable. For every  $p' \in S'$ ,  $l_{p'} < l_p$  and the induction hypothesis applies. The result follows from the fact that  $S_p = \bigcup_{p' \in S'} S_{p'}$ .  $\square$

**Theorem 3.8.5.** *Let  $p$  be a position and  $\lambda \in \{0, 1\}$ . There is a winning  $\lambda$ -strategy for  $p$  iff  $p$  is  $\lambda$ -provable.*

*Proof.* We prove the result by induction on the maximal length  $l_p$  of the plays starting in  $p$  (see Proposition 3.6.11). If  $l_p = 0$ ,  $p$  is final and there is a winning  $\lambda$ -strategy from  $p$  iff  $p$  is a win, iff  $p$  is  $\lambda$ -provable. Suppose that  $l_p > 0$ . Let  $S = \{p' | p \rightarrow p'\}$ . There are two cases:  $p$  is either a  $\lambda$ -macro-position or a  $\bar{\lambda}$ -macro-position.

If  $p$  is a  $\lambda$ -macro-position, then there is a winning  $\lambda$ -strategy from  $p$  iff there is a winning  $\lambda$ -strategy from some  $p' \in S$ , iff, by induction hypothesis ( $l_{p'} < l_p$ ), there exists  $p' \in S$  which is  $\lambda$ -provable, iff, by lemma 3.8.4,  $p$  is  $\lambda$ -provable.

If  $p$  is a  $\bar{\lambda}$ -macro-position, then there is a winning  $\lambda$ -strategy from  $p$  iff there is a winning  $\lambda$ -strategy from every  $p' \in S$ , iff, by induction hypothesis ( $l_{p'} < l_p$ ), every  $p' \in S$  is  $\lambda$ -provable, iff, by lemma 3.8.4,  $p$  is  $\lambda$ -provable.  $\square$



# A concurrent neutral game for MALL

---

The game presented in the previous chapter captures provability for MALL in a neutral setting. Our next step is to modify the game to capture proofs. This requires a fundamental change of the simplistic positional game model used so far. From a purely sequential game, we shift to a concurrent setting in which players may play simultaneously.

## 4.1 Introduction

The sequential game presented in the previous chapter suffers from major limitations that we discuss here.

### 4.1.1 Non-uniformity

In the sequential game, a position represents two dual slices of derivations, and a move represents a phase. The players base each of their decisions on the whole position instead of just some part of it. As a result, there are significantly more winning strategies than proofs. For example consider a neutral expression of the form  $\Downarrow(\Downarrow E_1 + \Downarrow E_2) \times \Downarrow(\mathbb{1} \times \Downarrow(\Downarrow F_1 + \Downarrow F_2))$ . Any proof of its positive translation  $([E_1]^+ \& [E_2]^+) \otimes (\perp \wp ([F_1]^- \oplus [F_2]^-))$  has the form

$$\frac{\frac{\frac{\vdots}{\vdash; [E_1]^+}; \quad \frac{\vdots}{\vdash; [E_2]^+}}{\vdash; [E_1]^+ \& [E_2]^+} \quad \frac{\frac{\frac{\vdots}{\vdash; [F_i]^-}}{\vdash; [F_1]^- \oplus [F_2]^-}}{\vdash; \perp \wp ([F_1]^- \oplus [F_2]^-)}}{\vdash; ([E_1]^+ \& [E_2]^+) \otimes (\perp \wp ([F_1]^- \oplus [F_2]^-))}$$

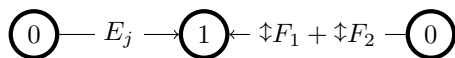
where the details of the phases are omitted. The game on this neutral expression starts with

$$\textcircled{0} \text{ --- } \Downarrow(\Downarrow E_1 + \Downarrow E_2) \times \Downarrow(\mathbb{1} \times \Downarrow(\Downarrow F_1 + \Downarrow F_2)) \text{ --- } \textcircled{1}$$

and the first move, by player 0, necessary leads to

$$\textcircled{0} \text{ --- } \Downarrow E_1 + \Downarrow E_2 \text{ --- } \textcircled{1} \text{ --- } \mathbb{1} \times \Downarrow(\Downarrow F_1 + \Downarrow F_2) \text{ --- } \textcircled{0}$$

At that point, player 0 must be prepared to face all the moves by player 1, including

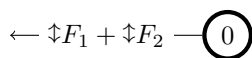


for both choices of  $j \in \{1, 2\}$ . Since those two are distinct positions, a strategy for player 0 may specify choices between  $\Downarrow F_1$  and  $\Downarrow F_2$  that are different in the two positions. In other words, the choice between  $\Downarrow F_1$  and  $\Downarrow F_2$  by player 0 may depend on that between  $\Downarrow E_1$  and  $\Downarrow E_2$  by player 1. We say that such a strategy is not uniform. In contrast, a proof (see above) makes a unique choice between  $[F_1]^-$  and  $[F_2]^-$  and its corresponding winning strategy is uniform. If  $[E_1]^+$ ,  $[E_2]^+$ ,  $[F_1]^-$  and  $[F_2]^-$  are all provable, then player 0 has non-uniform winning strategies which do not correspond to proofs.

Game semantics have come up with a variety of solutions to this problem. They eliminate non-uniform strategies by carefully specifying which information is available to each move. That way, some moves can be made independent of others.

### 4.1.2 Locality and concurrency

The advantage of our formalism is that this dependence is seen in the structure of the neutral graph itself. A sequent precisely represents all the information on which to base choices in proof search. Since a sequent is represented by a node with its surrounding arcs in a neutral graph, choices for a node should be based on that node only. In the example above, the choice between  $\Downarrow F_1$  and  $\Downarrow F_2$  by player 0 should be based on the following part of the neutral graph only:

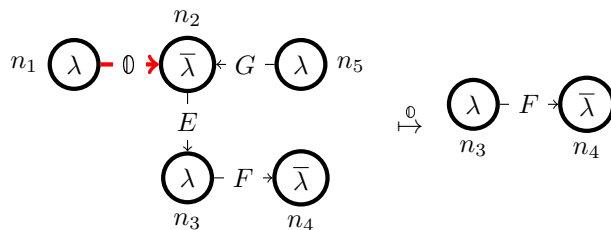


Just like extending a derivation with a rule only affects a branch and leaves the rest of the frontier unchanged, micro-moves only rewrite neutral graphs locally. In this chapter, we make this intuition clearer by expressing micro-moves and macro-moves as transitions rewriting local parts of neutral graphs, in the tradition of interaction nets [Laf90].

The game is turned into a concurrent process: moves rewriting distinct parts of a large structure independently of each other need not be sequentialised. We thus shift towards a different kind of game, in which moves are carried out by both players concurrently.

### 4.1.3 The $\top$ rule

The introduction rule for  $\top$  has a non-local behaviour in our neutral setting. In the following example of a  $\overset{0}{\mapsto}$  micro-move



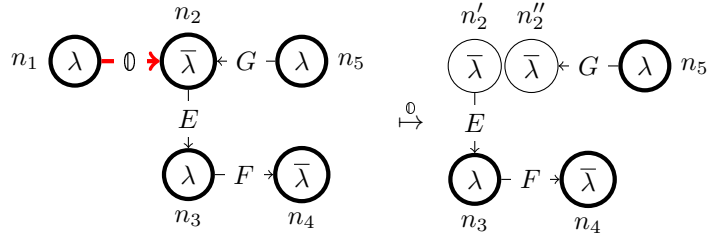
$n_2$  is deleted along with its connected arcs, which affects  $n_1$ ,  $n_3$  and  $n_5$ . For player  $\lambda$ , this amounts to extending three branches of the derivation with  $[\boxtimes]$ .

Player $\lambda$	Player $\bar{\lambda}$
$\frac{}{\vdash; \downarrow 0} [\boxtimes] \quad \frac{\vdash; [F]^+;}{\vdash; [F]^+; [E]^-} [\boxtimes]$ $\frac{}{\vdash; [G]^+} [\boxtimes]$	$\frac{}{\vdash; [E]^+; [G]^- \uparrow \top} [\top] \quad \vdash; [F]^-$

This is not an issue as far as provability is concerned: since player  $\lambda$  faces the unprovable sequent  $\vdash; \downarrow 0$  and has to apply  $[\boxtimes]$  there, she may as well apply it to the two other sequents. On the other hand, the simultaneous extension of distinct branches of a derivation is not a local operation, which prevents a concurrent treatment of those branches. Ideally, only the sequents associated with  $n_1$  and  $n_2$  should be affected:

Player $\lambda$	Player $\bar{\lambda}$
$\frac{}{\vdash; \downarrow 0} [\boxtimes] \quad \vdash; [F]^+; [E]^- \quad \vdash; [G]^+;$	$\frac{}{\vdash; [E]^+; [G]^- \uparrow \top} [\top] \quad \vdash; [F]^-$

That way, interaction on  $E$  and  $G$  can take place concurrently with the one on  $\emptyset$ . This, however, breaks the symmetry between the two derivations, as  $[E]^+$  and  $[G]^-$  disappear from the frontier but  $[E]^-$  and  $[G]^+$  remain. We choose to represent this situation in a neutral graph by keeping the arcs labelled with  $E$  and  $G$ ; when  $n_2$  is deleted, we replace it with nodes of a new kind acting as a “plugs” for the dangling arcs. The micro-move becomes



Those two new nodes  $n'_2$  and  $n''_2$  are called *non-goals* (notice the thinner outline) as they are not associated with any sequent in player  $\bar{\lambda}$ 's derivation. In contrast, the other nodes are *goals*. Non-goals can be used for interaction like goals. For example the players can continue playing on  $E$  and  $G$ . This allows greater modularity, as the micro-move no longer affects  $n_3$  and  $n_5$ . That way, moves on  $n_1$ ,  $n_3$  and  $n_5$  will take place concurrently.

Non-goals play a crucial role in interaction. In the above example, if  $E$  is of the form  $E_1 + E_2$ ,  $n'_2$  allows player  $\bar{\lambda}$  to pick either  $E_1$  or  $E_2$ . This additive choice has an obvious impact on the play from player  $\lambda$ 's point of view. However, it has no impact on the sequents player  $\bar{\lambda}$  has to prove, hence the name “non-goal”.

#### 4.1.4 Failure

The treatment of failure is simpler thanks to the introduction of non-goals. Instead of considering cases for handling failure in the relevant micro-moves



**Additives**

$$\frac{\vdash \Gamma \Downarrow F_i, \Delta}{\vdash \Gamma \Downarrow F_1 \oplus F_2, \Delta} [\oplus_i] \quad \frac{\vdash \Gamma \Uparrow F_1, \Delta \quad \vdash \Gamma \Uparrow F_2, \Delta}{\vdash \Gamma \Uparrow F_1 \& F_2, \Delta} [\&] \quad \frac{}{\vdash \Gamma \Uparrow \top, \Delta} [\top]$$

**Multiplicatives**

$$\frac{\vdash \Gamma_1 \Downarrow F_1, \Delta_1 \quad \vdash \Gamma_2 \Downarrow F_2, \Delta_2}{\vdash \Gamma_1, \Gamma_2 \Downarrow F_1 \otimes F_2, \Delta_1, \Delta_2} [\otimes] \quad \frac{}{\vdash \Downarrow 1} [1]$$

$$\frac{\vdash \Gamma \Uparrow F_1, F_2, \Delta}{\vdash \Gamma \Uparrow F_1 \wp F_2, \Delta} [\wp] \quad \frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \perp, \Delta} [\perp]$$

**Identity**

$$\frac{}{\vdash L^- \Downarrow L^+} [\text{init}]$$

**Daimon**

$$\frac{}{\vdash \Gamma} [\boxtimes]$$

**Phase changes**

$$\frac{\vdash F^-, \Gamma \Downarrow \Delta}{\vdash \Gamma \Downarrow F^-, \Delta} [\text{R}\Downarrow] \quad \frac{\vdash F^+, \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow F^+, \Delta} [\text{R}\Uparrow] \quad \frac{\vdash L^-, \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow L^-, \Delta} [\text{atomic R}\Uparrow]$$

$$\frac{\vdash \Gamma^+, \Lambda^- \Downarrow \Delta^+}{\vdash \Gamma^+, \Delta^+, \Lambda^-} [\text{D}\Downarrow] \quad \frac{\vdash \Gamma \Uparrow F^-}{\vdash \Gamma, F^-} [\text{D}\Uparrow]$$

$\Gamma$  and  $\Delta$  stand for multiset of formulae,  $\Lambda$  for a multiset of literals,  $F$  for a formula, and  $L$  for a literal. In  $[\text{D}\Downarrow]$ ,  $\Delta^+$  is not empty. A proof is a closed derivation which does not use  $[\boxtimes]$ .

Figure 4.1: The proof system used in the concurrent game for MALL

as we did in the sequential game, we will introduce a special move for this. Informally, the daimon move will allow a player to locally give up by replacing one of its goals with non-goals without affecting the rest of the neutral graph. However, the player forfeits the play by doing so.

## 4.2 Proof system

We will use a slightly different proof system in this chapter. Figure 4.1 shows it. Sequents are of the form  $\vdash \Gamma \Downarrow \Delta$  or  $\vdash \Gamma \Uparrow \Delta$ , where  $\Gamma$  and  $\Delta$  are multisets of formulae. The sequents  $\vdash \Delta \Downarrow$  and  $\vdash \Delta \Uparrow$  are identified and also denoted to by  $\vdash \Delta$ .

The main difference with the previous proof system is that sequents have less zones. Another notable difference is that the daimon rule with one premise has been removed, and that the one with no premise is only available outside of phases. Although they are not fundamental, these changes allow for simpler notations and reflect an evolution in the treatment of literals and failure in the

game. Moreover, the  $[D\uparrow]$  rule only allows to pick one formula to focus on, as the game won't need more. This last change is mostly irrelevant, since we use the same notion of proof identity as usual in focused proof systems. That is, we consider proofs up to permutations of individual rules within a phase, but also up to permutations of adjacent asynchronous phases, since our proof system allows it.

### 4.3 Neutral graphs

In this section, we adapt the notion of neutral graph to the new framework.

We will use monomial weights as a way to refer to additive slices.

**Definition 4.3.1** (Monomial weight). *Consider a set  $\mathcal{B}$  of Boolean variables. A literal of  $\mathcal{B}$  is either a propositional variable or a negation of one. A monomial weight (or weight) over  $\mathcal{B}$  is a conjunction of literals. The weights 1 and 0 denote truth and falsehood respectively.*

Two monomial weights are *disjoint* when their conjunction is 0.

Monomial weights are commonly used in proof nets with additives [Gir96, LM08]. In a proof net, a distinct propositional variable is associated with each  $\&$  node, and each formula has a monomial weight. For a  $\&$  node with associated variable  $p$ , if  $w$  denotes the weight of the conclusion of the node, then the two premises have weights  $wp$  and  $w\neg p$ , where we use the product notation for conjunction and  $\neg$  denotes negation. Informally, if one considers  $\&$  as a “fork” of the world into two separate sub-worlds, then the monomial weight of a formula represents the world it lives in. In sequent calculus, those worlds are branches. For example, consider the derivation

$$\frac{\frac{\frac{\frac{\vdash\uparrow A, C \quad \vdash\uparrow A, D}{\vdash\uparrow A, C \& D} [\&]}{\vdash\uparrow A \& B, C \& D} [\&]}{\vdash\uparrow (A \& B) \wp (C \& D)} [\wp]}{\vdash\uparrow (A \& B) \wp (C \& D)} [\&]}$$

There are four branches, because each one of the two  $\&$  is responsible for a fork. If we associated the propositional variables  $p$  and  $q$  to the  $\&$  in  $A\&B$  and  $C\&D$  respectively, and the weight of the conclusion was  $w$ , then the weights of the four premises would be  $wpq$ ,  $wp\neg q$ ,  $w\neg pq$ , and  $w\neg p\neg q$ .

In the game, we will associate a monomial weight to each node of a neutral graph, in order to mark which slice it lives in.

**Definition 4.3.2** (Neutral graph). *A neutral graph is a directed graph structure consisting of labelled nodes and arcs.*

1. Each node has an associated player (0 or 1) and monomial weight; a node with weight 0 is a non-goal, otherwise it is a goal;
2. each arc is labelled with a guarded neutral expression occurrence;
3. the graph is bipartite, i.e. two connected nodes have opposite players;
4. there are no undirected cycles;
5. each non-goal is connected to exactly one other node, which must be a goal.

We use the following convention to represent nodes, where  $\lambda$  stands for the associated player.



Notice that, in contrast to the sequential game, isolated goals are allowed. They will represent instances of the unprovable empty sequent.

As explained informally in Section 4.1.3, both non-goals and goals are used for interaction, but only goals represent sequents that the players have to prove. Non-goals are required to be connected to exactly one node, which must be a goal, because they should be seen as the missing ends of dangling arcs. Non-goals not connected to goals would be completely useless, since none of the interaction they would provide would have any consequence on the victory or loss of the players.

## 4.4 Subgraphs

In proof search, a sequent represents a state in the computation. In sequent calculus, no information flows from a branch of a proof to another. This differs from other formalisms such as natural deduction in which the introduction of an implication triggers the discharge of a hypothesis, which is a non local operation. In our game, goals and their neighbouring arcs represent sequents. As our game aims at modelling proof search, the locality of proof search should be accounted for. That is, the moves of the game will rewrite a neutral graph locally. We need to capture a notion of local part of a neutral graph to serve as a basis for applying moves.

**Definition 4.4.1** (Subgraph). *A subgraph is a part of a neutral graph consisting of some of its nodes and all the arcs connected to those nodes. The missing ends of the arcs are called “holes”. Just like a node, each hole is labelled with a monomial weight.*

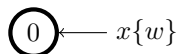
If all the nodes of a subgraph belong to the same player  $\lambda$ , then the subgraph is a  $\lambda$ -subgraph.

**Remark 4.4.2.** *The nodes of a subgraph are distinct. Therefore in*



*the subgraph consisting of the left node is distinct from the one consisting of the right node.*

In graphical representations of subgraphs, we will sometimes use names to distinguish specific holes. In the subgraph



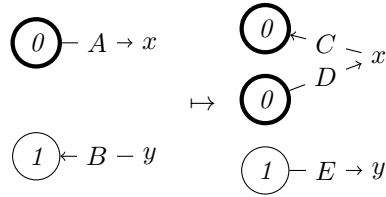
the hole  $x$  has the associated weight  $w$ . When the weight is not written, it is assumed to be 1.

From the definition, it is clear that a subgraph represents a part of a neutral graph. More generally, we can even take a subgraph of a subgraph. The holes of a subgraph are its interface with the rest of this larger structure, which we

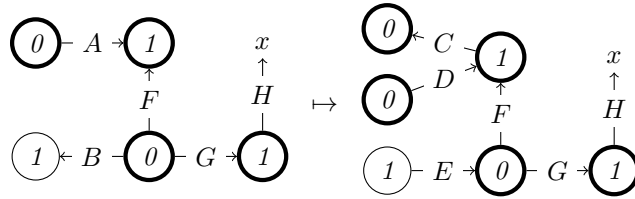
refer to as its “context”. When reconnecting a subgraph to a context, each hole  $h$  is merged with a target  $t$  in the context, which can be a node or a hole itself. The weight of the resulting object is the product of the weights of  $h$  and  $t$ . Most of the time we will consider holes with weight 1, which leave the weight of their target unchanged.

In our game, we will express locality by defining moves as a relation rewriting subgraphs. When doing so, the holes will be the same before and after the transition (although their weights may change), therefore reconnecting the rewritten subgraph to the original context will be unambiguous.

**Example 4.4.3.** *The transition*



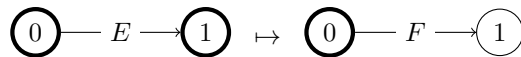
yields, in a context:



A hole with weight 0 turns a goal into a non-goal. For example, the transition



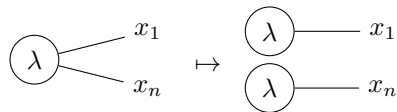
yields:



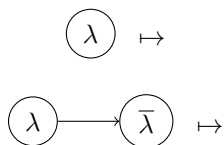
As we will see, weights will be used for additives. The micro-move decomposing the neutral connective  $+$  will apply a weight  $p$  or  $\neg p$  depending on the choice of the operand, which is consistent with its interpretation as the rule  $[\&]$ . The micro-move dealing with the 0-additive  $0$  will apply a null weight like in the above example.

**Reconnection rules** When reconnecting a rewritten subgraph to its context, a few situations may arise.

1. If any non-goal is connected to  $n > 1$  arcs (for example if its weight has just become 0), it is split into  $n$  non-goals. This ensures that no non-goal be connected to more than one node.

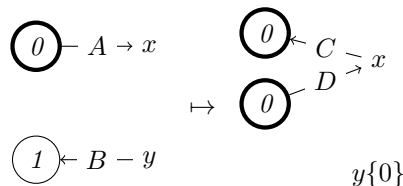


2. Then, isolated non-goals and isolated pairs of connected non-goals are removed. This ensures that each non-goal be connected to exactly one node, and that this node be a goal.

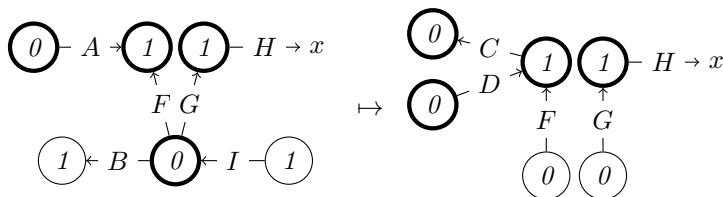


Note that the graph structure resulting from reconnecting a subgraph into a context needs to be bipartite and acyclic to be a subgraph. In practice it will always be the case.

**Example 4.4.4.** *If we modify the transition of Example 4.4.3 in the following way*



we get, in a context:



If  $G$  is a subgraph of  $H$ , we may write  $H$  under the form  $C[G]$  where  $C$  represents the context. Then  $C[G']$  is the result of rewriting  $G$  to  $G'$  inside  $H$ .

In our formalism, the fundamental unit where interaction takes place is not exactly a node, but a node together with its connected arcs. It is also called a subposition.

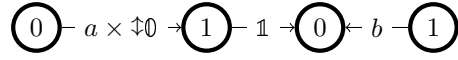
**Definition 4.4.5** (Subposition). *Let  $\lambda \in \{0, 1\}$ . A  $\lambda$ -subposition is a  $\lambda$ -subgraph consisting of a node  $n$  along with its connected arcs.*

From now on, we will mostly use goal (resp. non-goal) to refer to a subposition whose node is a goal (resp. non-goal).

A subposition is a place where the associated player has to move. Among them, goals represent sequents to be proved by the player. The formulae composing the sequent are represented by the arcs connected to the node.

**Definition 4.4.6** (Sequent represented by a goal). *Let  $g$  be a goal in a neutral graph. We collect, for each arc  $a$  of  $g$  labelled with a neutral expression  $E$ , the formula  $[E]^+$  if  $a$  is outgoing, or the formula  $[E]^-$  if  $a$  is incoming. The sequent associated with  $g$  is  $\vdash \Gamma$  where  $\Gamma$  denotes the multiset of those formulae.*

That is, a neutral expression labelling an arc is translated positively by the source and negatively by the target. For example, in the neutral graph



player 0 has the goals  $\vdash A \otimes \top$  and  $\vdash B^\perp, \perp$ , and player 1 has the goals  $\vdash 1, A^\perp \wp 0$  and  $\vdash B$ .

In a neutral graph, a player's subpositions contain all the information on which this player bases her moves. A player will be able to play in subpositions where she is not waiting for any information. They are referred to as playable subpositions.

**Definition 4.4.7** (Playable subposition). *A subposition is playable if the two following conditions are met:*

- the neutral expressions labelling the incoming arcs are atoms,
- if the node is a non-goal, the unique arc connected to it is labelled with a non-atomic neutral expression (and is therefore outgoing by the previous point).

This definition is analogous to that of a source in the sequential game (Definition 3.5.3).

## 4.5 Micro-moves

In this section, we define the micro-moves of the concurrent game. We add the possibility for arcs to be focused.

**Definition 4.5.1** (Micro-subgraph). *Micro-subgraphs extend subgraphs in the sense that*

- each regular arc may be in two states, either normal (like in subgraphs) or focused,
- the neutral expression labelling a focused arc may be not guarded.

We use the following convention in graphical representations of regular arcs.



We define  $\lambda$ -micro-moves as a transitions on  $\lambda$ -micro-subgraphs. Each micro-move is an operation with two readings, one for each player. As explained before, each player “sees” her subpositions only. Player  $\mu$  reads a micro-move in terms of the  $\mu$ -subpositions that are affected. For each  $\lambda$ -micro-move:

- Exactly one  $\lambda$ -subposition  $s$  is directly affected; it is replaced with zero, one or more  $\lambda$ -subpositions  $s_1, \dots, s_n$ . Moreover, if  $s$  is a goal representing the sequent  $\Sigma$ , and  $\Sigma_1, \dots, \Sigma_p$  are the sequents representing those of the  $s_i$  which are goals, then there is a synchronous rule

$$\frac{\Sigma_1, \dots, \Sigma_p}{\Sigma}$$

- A hole  $h$  (standing for a  $\bar{\lambda}$ -subposition) may be affected through the arcs connected to it or changes in its weight; depending on the choices of player  $\lambda$  for this  $\lambda$ -micro-move,  $h$  may become one of  $h_1, \dots, h_n$ . We can associate a generic sequent  $\Sigma$  to  $h$  like we would do for any  $\lambda$ -subposition it stands

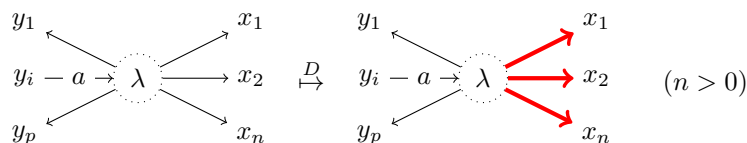
for, by including a generic multiset standing for the arcs that are not part of the subgraph. Then, if  $\Sigma_1, \dots, \Sigma_p$  are the sequents representing those of the  $h_i$  which are goals, there is an asynchronous rule

$$\frac{\Sigma_1, \dots, \Sigma_p}{\Sigma}$$

There might be other subpositions “indirectly” affected. Those are always non-goals which get deleted as a result of the move because of the reconnection rules.

We now describe the micro-moves, each time explicitly giving the reading in terms of a synchronous rule by the player and asynchronous rules by the opponent.

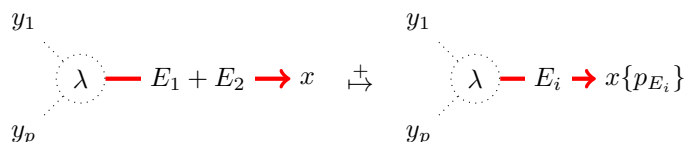
**Decision** Playable  $\lambda$ -subpositions are rewritten by focusing some outgoing arcs (at least one).



This micro-move is read as:

The $\lambda$ -node	Each $x_i$
$\frac{\vdash \Gamma^+, \Lambda^- \Downarrow \Delta^+}{\vdash \Gamma^+, \Delta^+, \Lambda^-} \text{ [D}\Downarrow\text{]}$	$\frac{\vdash \Gamma \Uparrow F^-}{\vdash \Gamma, F^-} \text{ [D}\Uparrow\text{]}$

**Additives** Sums labelling focused arcs are rewritten to one of their operands.



where  $i \in \{1, 2\}$  and  $p_E$  is a propositional variable uniquely associated with the neutral expression occurrence  $E$ .

This micro-move is read as:

The $\lambda$ -node	$x$
$\frac{\vdash \Gamma \Downarrow [E_i]^+, \Delta}{\vdash \Gamma \Downarrow [E_1]^+ \oplus [E_2]^+, \Delta} \text{ [\oplus]}$	$\frac{\vdash \Gamma \Uparrow [E_1]^-, \Delta \quad \vdash \Gamma \Uparrow [E_2]^-, \Delta}{\vdash \Gamma \Uparrow [E_1]^- \& [E_2]^-, \Delta} \text{ [\&]}$

Note that the weight marking  $x$  ensures that the choice between  $E_1$  and  $E_2$  results in two distinct outcomes for  $x$ , corresponding to the two premises of the  $[\&]$  rule.

A non-goal with an outgoing focused arc labelled with an empty sum is removed along with the arc, and its target receives weight 0.

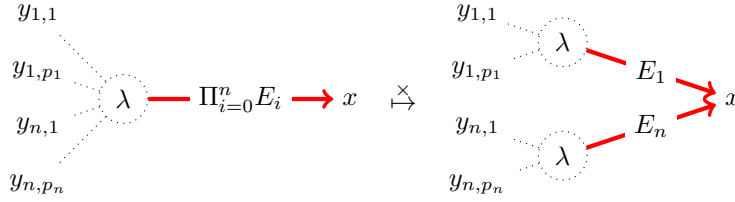
$$\textcircled{\lambda} - 0 \rightarrow x \xrightarrow{0} x\{0\}$$

This micro-move is read as:

$$\frac{}{\vdash \Gamma \uparrow \top, \Delta} [\top]$$

by  $x$ . Indeed,  $x$  receives weight 0 which means that the rule has no premises. Note that there is no interpretation as a synchronous rule by player  $\lambda$  since the  $\lambda$ -subposition is a non-goal.

**Multiplicatives** We cover the binary ( $\times$ ) and nullary ( $\mathbb{1}$ ) multiplicatives simultaneously with a general product  $\Pi$ . A focused arc labelled with a product  $\Pi_{i=1}^n$  is split in  $n$  arcs labelled with each operand. It keeps the same target, but the source is split in  $n$ , partitioning its other connected arcs, if any.



This micro-move is read as:

The $\lambda$ -node	$x$
$\frac{\{\vdash \Gamma_i \downarrow [E_i]^+, \Delta_i   i = 1, \dots, n\}}{\vdash \Gamma_1, \dots, \Gamma_n \downarrow \otimes_1^n [E_i]^+, \Delta_1, \dots, \Delta_n} [\otimes, 1]$	$\frac{\vdash \Gamma \uparrow [E_1]^-, \dots, [E_n]^-, \Delta}{\vdash \Gamma \uparrow \wp_1^n [E_i]^-, \Delta} [\wp, \perp]$

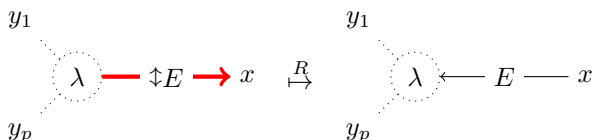
**Axiom** A goal with exactly two connected arcs, both labelled with an atom  $a$ , one of which is normal and incoming, the other one focused and outgoing, is split into two non-goals.

$$y - a \rightarrow \textcircled{\lambda} - a \rightarrow x \xrightarrow{\text{ax}} y - a \rightarrow \textcircled{\lambda} \textcircled{\lambda} - a \rightarrow x$$

This micro-move is read as:

The $\lambda$ -node	$x$
$\frac{}{\vdash [a]^- \downarrow [a]^+} [\text{init}]$	$\frac{\vdash \Gamma, [a]^- \uparrow \Delta}{\vdash \Gamma \uparrow [a]^-, \Delta} [\text{atomic R}\uparrow]$



**Reaction**

This micro-move is read as:

The $\lambda$ -node	$x$
$\frac{\vdash \Gamma, [E]^- \Downarrow \Delta}{\vdash \Gamma \Downarrow [E]^-, \Delta} \text{ [R}\Downarrow\text{]}$	$\frac{\vdash \Gamma, [E]^+ \Uparrow \Delta}{\vdash \Gamma \Uparrow [E]^+, \Delta} \text{ [R}\Uparrow\text{]}$

The  $\lambda$ -micro-move relation  $\mapsto_\lambda$  is the closure under contexts of the above transitions.

As seen above, a micro-move has a proof-theoretic interpretation for each player. However, this interpretation only concerns goals, which is not surprising since goals represent what the players have to prove. Non-goals should not be neglected though, as they take part in the interaction.

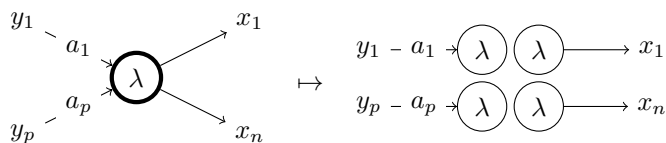
## 4.6 Macro-moves and positions

### 4.6.1 Definition

As we did in the sequential game, we define macro-moves on top of micro-moves. A macro-move abstracts away from micro-moves in the same way that phases abstract away from individual rules in the proof system. We have not mentioned the possibility of failure in the micro-moves yet. Failure is handled on the level of macro-moves.

**Definition 4.6.1** (Macro-move). *Let  $G$  be a playable  $\lambda$ -subposition. A  $\lambda$ -macro-move from  $G$  is a triple  $(G, G', f)$  where*

- $f = \perp$  and  $G \mapsto_\lambda^* G'$  and  $G'$  does not rewrite through a  $\lambda$ -micro-move,
- or  $f = \top$ ,  $G$  is based on a goal and  $G'$  is obtained by turning it into non-goals:



*In the first case the macro-move is regular, in the second case it is a daimon macro-move.*

Daimon macro-moves allow a player to give up on one of her goals. The Boolean flag  $f$  signals the use of the daimon, which prevents a player from winning a play. In other words, daimon is a special “failure” move.

**Definition 4.6.2** (Position). *A position is a triple  $(G, f_0, f_1)$ , where  $G$  is a neutral graph and  $f_0$  and  $f_1$  are Boolean values.*

If  $m = (G, G', f)$  is a  $\lambda$ -macro-move and there is a position  $p$  of the form  $(C[G], f_0, f_1)$ , player  $\lambda$  may play  $m$  in  $p$ :

$$(C[G], f_0, f_1) \xrightarrow{m} (C[G'], f'_0, f'_1)$$

where  $f'_\lambda = f_\lambda \vee f$  and  $f'_\bar{\lambda} = f_{\bar{\lambda}}$ .

The Boolean values  $f_0$  and  $f_1$  have the same meaning as in the sequential game. They are true when the corresponding player has failed.

The following proposition expresses the concurrent nature of the game.

**Proposition 4.6.3.** *Let  $p$  be a position and  $m_1$  and  $m_2$  two macro-moves at distinct playable subpositions  $s_1$  and  $s_2$  of  $p$ . Let  $p \xrightarrow{m_1} p_1$  and  $p \xrightarrow{m_2} p_2$ . Then one of the following applies:*

- $m_1$  and  $m_2$  commute, that is  $p_1 \xrightarrow{m_2} p'$  and  $p_2 \xrightarrow{m_1} p'$  for some  $p'$ ;
- $s_2$  is a non-goal which gets removed by  $m_1$ , and  $p_2 \xrightarrow{m_1} p_1$ ;
- $s_1$  is a non-goal which gets removed by  $m_2$ , and  $p_1 \xrightarrow{m_2} p_2$ ;
- $s_1$  and  $s_2$  are non-goals which get removed by  $m_2$  and  $m_1$  respectively, and  $p_1 = p_2$ .

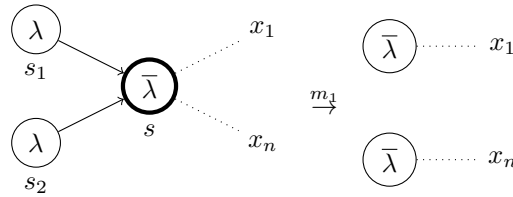
*Proof.* Let  $d$  be the distance between  $s_1$  and  $s_2$  in  $p$  (which may be infinite).

If  $d > 2$ , then  $m_1$  and  $m_2$  affect disjoint parts of  $p$  and commute.

$d$  cannot be 0, as  $s_1$  and  $s_2$  are distinct.

Suppose that  $d = 1$ . There is an arc  $a$  from  $s_1$  to  $s_2$  (for example). Since  $s_1$  and  $s_2$  are playable, this arc is labelled with an atom and  $s_1$  and  $s_2$  are goals.  $m_1$  and  $m_2$  affect disjoint parts of  $p$  and commute.

Suppose that  $d = 2$ . There is a goal  $s$  between  $s_1$  and  $s_2$ , which is where  $m_1$  and  $m_2$  may interfere. From the point of view of  $s$ ,  $m_1$  and  $m_2$  affect distinct arcs and apply independent weights, therefore they commute, except in the case where  $s$  is turned into non-goals (because of a micro-move  $\xrightarrow{0}$ ). In that case the only problem occurs when, for example,  $m_1$  is responsible for this and  $s_2$  is a non-goal itself and gets removed by the reconnection rules as a result of  $m_1$ :



Obviously  $m_2$  cannot follow  $m_1$ . If  $m_2$  also turns  $s$  into non-goals, then  $p_1 = p_2$ . Otherwise  $p_2 \xrightarrow{m_1} p_1$ .  $\square$

## 4.6.2 Macro-moves as phases

As was the case with micro-moves, each player reads a macro-move in the way it affects her subpositions. We show that macro-moves can be seen as phases in the proof system.

**Proposition 4.6.4.** *Let  $s$  be a playable  $\lambda$ -goal and  $\Sigma$  the sequent it represents. Let  $m = (s, G', f)$  be a  $\lambda$ -macro-move at  $s$ . The effect of  $m$  on player  $\lambda$ 's subpositions is to replace  $s$  with the  $\lambda$ -subpositions of  $G'$ :  $s_1, \dots, s_n$  ( $n \geq 0$ ). If*

$\Sigma_1, \dots, \Sigma_p$  denote the sequents representing those of the  $s_i$  which are goals, then there is a derivation in the proof system

$$\frac{\Sigma_1, \dots, \Sigma_p}{\Sigma}$$

which is a synchronous phase if  $f = \perp$  and a  $\boxtimes$  rule otherwise. Conversely, every synchronous phase with conclusion  $\Sigma$  defines a unique regular  $\lambda$ -macro-move at  $s$ . The  $s_i$  are called the successors of  $s$  by  $m$ .

*Proof.* The obvious part is the reading of the daimon macro-move at  $s$  as the rule

$$\overline{\Sigma} \boxtimes$$

Indeed,  $s$  disappears and is replaced with non-goals only, therefore  $p = 0$ .

We have already established the reading of  $\lambda$ -micro-moves as synchronous rules by player  $\lambda$ . It is easy to see that they actually cover all the synchronous rules of the proof system. From the definition of a regular macro-move as a maximal sequence of micro-moves, it is clear that regular macro-moves are indeed read as synchronous phases, and that all synchronous phases are covered by macro-moves. What is less obvious is the unicity part, in other words that a synchronous phase

$$\frac{\Sigma_1, \dots, \Sigma_p}{\Sigma}$$

determines a unique regular move  $m = (s, G', \perp)$  at  $s$ . We show how to determine  $G'$  here. We use the crucial fact that arcs in neutral graphs are labelled with neutral expression *occurrences* and that sequents are made of formula *occurrences*. With that in mind, we can see that the  $\Sigma_i$  determine the  $\lambda$ -goals of  $G'$ . Since  $m$  is regular, the  $\lambda$ -non-goals of  $G'$  are those created by the  $\overset{\text{ax}}{\mapsto}$  micro-move. Those non-goals are easily determined as there is exactly one for each occurrence of a literal that is in  $\Sigma$  and not in the  $\Sigma_i$ . All the  $\lambda$ -subpositions of  $G'$  are determined. Next, the arcs connected to a common hole in  $G'$  are precisely those labelled with a neutral expression with a common ancestor in  $s$ . Lastly, the weights labelling the holes in  $G'$  reflect the information relative to the additive choices in a phase.  $\square$

The last point in the proof of Proposition 4.6.4 hints at a subtlety of the proof system. From the proof-theoretic point of view, there are actually two versions of the synchronous phase

$$\overline{\overline{\vdash 1 \oplus 1}}$$

even if we usually do not explicitly label this synthetic inference rule with the additive choice. In the game, there are two corresponding macro-moves:

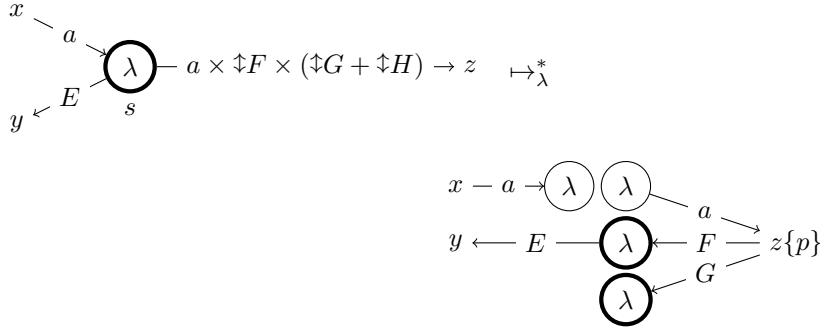
$$\textcircled{0} - \mathbb{1}_a + \mathbb{1}_b \rightarrow x \quad \mapsto_0^* \quad x\{p_i\}$$

where  $i$  is one of the occurrences  $\{a, b\}$ .

**Example 4.6.5.** *The synchronous phase*

$$\frac{\frac{\vdash [E]^+, [F]^- \quad \vdash [G]^-}{\vdash [a]^-, [E]^+, [a]^+ \otimes [F]^- \otimes ([G]^- \oplus [H]^-)}}{\vdash [a]^-, [E]^+, [a]^+ \otimes [F]^- \otimes ([G]^- \oplus [H]^-)}$$

corresponds to the regular macro-move



where  $p$  is the weight associated with the choice of  $\dagger G$  over  $\dagger H$ . We can see that  $s$  has four successors by this macro-move: two goals (corresponding to the premises of the phase) and two non-goals.

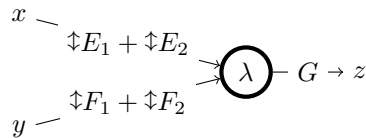
**Proposition 4.6.6.** *Consider a non-playable  $\lambda$ -subposition  $s$  and one of its incoming arcs  $a$  labelled with a non-atomic neutral expression  $E$ . Any  $\bar{\lambda}$ -macro-move focusing  $a$  replaces  $s$  with some  $\lambda$ -subpositions. Consider all those  $\lambda$ -subpositions  $s_1, \dots, s_n$  resulting from such macro-moves. If  $s$  is a goal representing the sequent  $\Sigma$ , and  $\Sigma_1, \dots, \Sigma_p$  are the sequents representing those of the  $s_i$  which are goals, then the asynchronous phase with conclusion  $\Sigma$  focusing on  $[E]^-$  is*

$$\frac{\Sigma_1, \dots, \Sigma_p}{\Sigma}$$

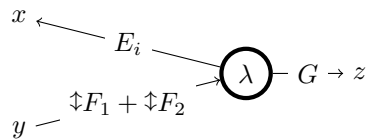
The  $s_i$  are called the successors of  $s$  by an opponent macro-move on  $a$ .

*Proof.* We have already established the reading of  $\bar{\lambda}$ -micro-moves as synchronous rules by player  $\lambda$ . It is easy to see that they actually cover all the asynchronous rules of the proof system. From the definition of a regular macro-move as a maximal sequence of micro-moves, it is clear that the regular  $\bar{\lambda}$ -macro-moves focusing  $a$  are indeed read as the asynchronous phase focusing on  $[E]^-$ .  $\square$

**Example 4.6.7.** *The successors of*



by an opponent macro-move on the arc labelled with  $\dagger E_1 + \dagger E_2$  are the



for  $i \in \{1, 2\}$  where the  $\lambda$ -node gets disjoint weights in the two cases. This corresponds to asynchronous phase

$$\frac{\vdash [E_1]^+, [F_1]^+ \& [F_2]^+, [G]^+ \quad \vdash [E_2]^+, [F_1]^+ \& [F_2]^+, [G]^+}{\vdash [E_1]^+ \& [E_2]^+, [F_1]^+ \& [F_2]^+, [G]^+}$$

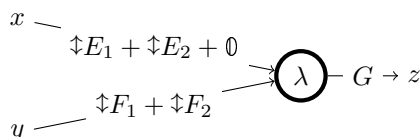
**Proposition 4.6.8.** *Consider a non-playable  $\lambda$ -subposition  $s$  and let  $s_1, \dots, s_n$  be all its goal successors by maximal sequences of opponent macro-moves. Then the  $s_i$  have pairwise disjoint monomial weights. If  $s$  is a goal representing the sequent  $\Sigma$ , and  $\Sigma_1, \dots, \Sigma_n$  are the sequents representing the  $s_i$ , then the maximal layer of asynchronous phases with conclusion  $\Sigma$  is*

$$\frac{\Sigma_1, \dots, \Sigma_n}{\Sigma}$$

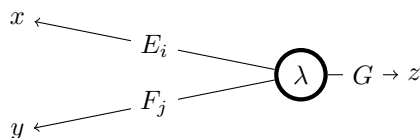
*Proof.* As long as no opponent macro-move assigns a null weight to  $s$  (because of  $\overset{0}{\mapsto}$ ), successive opponent macro-moves affecting  $s$  focus on distinct arcs and assign independent weights to  $s$ . They commute trivially. Maximal sequences of such moves yield successors  $s_i$  that are all goals. Suppose by contradiction that two distinct  $s_i$  have weights that are not disjoint, for example  $s_1$  and  $s_2$ . They cannot have the same weight, since they are distinct. Then one of the weights ( $s_1$ 's for example) must contain a literal  $l$  while the other does not contain  $l$  or  $\neg l$ . But then it means the macro-move responsible for assigning  $l$  or  $\neg l$  has not been applied to  $s_2$ , which contradicts the maximality hypothesis. Therefore the  $s_i$  have pairwise disjoint weights. Lastly, maximal sequences of opponent macro-moves are seen as maximal layers of asynchronous phases thanks to Proposition 4.6.6.

On the other hand, if one of the macro-moves assigns a null weight to  $s$ ,  $s$  is turned into non-goals. Non-goals only have non-goal successors, hence  $n = 0$  and the phase interpreting the whole sequence has no premises.  $\square$

**Example 4.6.9.** *The successors of*



by maximal sequences of opponent macro-moves are some non-goals and the



for  $(i, j) \in \{1, 2\}^2$  where the  $\lambda$ -node gets disjoint weights in the four cases. This corresponds to the maximal layer of asynchronous phases with conclusion  $\vdash [E_1]^+ \& [E_2]^+ \& \top, [F_1]^+ \& [F_2]^+$ :

$$\frac{\left\{ \vdash [E_i]^+, [F_j]^+, [G]^+ \mid (i, j) \in \{1, 2\}^2 \right\}}{\vdash [E_1]^+ \& [E_2]^+ \& \top, [F_1]^+ \& [F_2]^+, [G]^+}$$

## 4.7 Strategies

We defined strategies using a technique for graph arenas inspired by [HS02]. We first define a pre-strategy as an object assigning a macro-move to each of the playable subpositions of a player. Then we restrict it on the actual subpositions reachable from a starting subposition.

**Definition 4.7.1** (Pre-strategy). *A  $\lambda$ -pre-strategy is a function mapping each playable  $\lambda$ -subposition  $s$  to a  $\lambda$ -macro-move at  $s$ .*

**Definition 4.7.2** (Reachable subpositions). *Let  $\alpha$  be a  $\lambda$ -pre-strategy and  $s_0$  a  $\lambda$ -subposition. The set  $\mathcal{R}(\alpha, s_0)$  of the  $\lambda$ -subpositions reachable from  $s_0$  by  $\alpha$  is the smallest set  $R$  such that*

- $s_0 \in R$ ,
- if  $s \in R$  is not playable and  $s'$  is a successor of  $s$  by an opponent macro-move, then  $s' \in R$ ,
- if  $s \in R$  is playable and  $s'$  is a successor of  $s$  by  $\alpha(s)$ , then  $s' \in R$ .

**Definition 4.7.3** (Strategy). *A  $\lambda$ -strategy for a subposition  $s$  is the restriction of a  $\lambda$ -pre-strategy to the playable  $\lambda$ -subpositions it can reach from  $s$ .*

A simple position is a position  $p = (G, \perp, \perp)$  where  $G$  is of the form

$$\textcircled{\mu} - E \rightarrow \textcircled{\bar{\mu}}$$

where  $E$  is a guarded neutral expression. In that case we will often call “ $\lambda$ -strategy for  $p$ ” a  $\lambda$ -strategy for the  $\lambda$ -subposition of  $G$ .

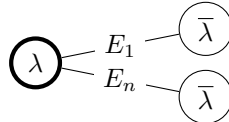
**Definition 4.7.4** (Interaction). *Let  $\sigma_0$  and  $\sigma_1$  be two strategies for a simple position  $p$  for player 0 and player 1 respectively. The interaction of  $\sigma_0$  and  $\sigma_1$  is the relation rewriting positions (starting in  $p$ ) according to the macro-moves specified by the strategies.*

The positions obtained from  $p$  through this relation are called the reachable positions of the interaction.

**Proposition 4.7.5.** *Let  $\lambda \in \{0, 1\}$  and  $\sigma$  be a  $\lambda$ -strategy for a simple position  $p$ . The  $\lambda$ -goals reachable by  $\sigma$  are those occurring in the positions reachable by the interactions involving  $\sigma$ .*

*Proof.* One direction is obvious. From the definition of a strategy, it is clear that all the  $\lambda$ -subpositions occurring in the positions reachable by interactions involving  $\sigma$  are reachable by  $\sigma$ .

Let us show the converse. It suffices to consider the interactions of  $\sigma$  against strategies playing the daimon as soon as possible (either immediately or after one move depending on which player plays first in  $p$ ). Then all subsequent positions will be made of connected components consisting of one  $\lambda$ -goal surrounded by  $\bar{\lambda}$ -non-goals.



Player  $\bar{\lambda}$  can then move independently on each arc. From the point of view of the  $\lambda$ -goal, every opponent macro-move is represented in some  $\bar{\lambda}$ -strategy.  $\square$

**Proposition 4.7.6.** *The interaction of two strategies for a simple position  $p$  is terminating and strongly confluent. It is therefore strongly normalising. The normal form of  $p$  is a position of the form  $(\cdot, f_0, f_1)$ , where at least one of  $f_0$  and  $f_1$  is  $\top$ .*

*Proof.* We first show that interaction is terminating. It suffices to notice that each macro-move strictly decreases the pair  $(s, g)$  of a position for the lexicographic order, where  $s$  is the total number of symbols of the neutral expressions labelling the arcs, and  $g$  is the number of goals. In addition, interaction is strongly confluent because two interacting strategies have disjoint domains and Proposition 4.6.3 applies. Interaction is therefore strongly normalising.

Let us show that the normal form  $n$  of  $p$  has an empty neutral graph. Any playable subposition of a position reachable by the interaction is reachable by its player's strategy, and thus has a macro-move specified for it. Hence  $n$  does not have any playable subposition. Neutral graphs are finite and acyclic, which implies that every non-empty neutral graph has a playable subposition. Therefore  $n$ 's neutral graph is empty.

Let us write  $n = (\cdot, f_0, f_1)$  and show that at least one of  $f_0$  and  $f_1$  is  $\top$ . Assume by contradiction that  $f_0 = f_1 = \perp$ . Consider a rewrite sequence from the starting position of the interaction to  $n$ . None of the moves in this sequence are daimons, otherwise one of the  $f_i$  would be  $\top$ . Consequently, the only ways to remove goals are the  $\overset{\text{ax}}{\rightarrow}$  and 0-ary  $\overset{\times}{\rightarrow}$  micro-moves ( $\overset{0}{\rightarrow}$  requires the presence of a non-goal which can only be introduced by a daimon). At least one of them is used, since the starting position has two goals and  $n$  has none.  $\overset{\text{ax}}{\rightarrow}$  has to be used. Indeed, the 0-ary  $\overset{\times}{\rightarrow}$  micro-move

$$\textcircled{\lambda} - \mathbb{1} \rightarrow x \quad \overset{\times}{\rightarrow} \quad x$$

cannot be the only one to be used. If it were, then there would be no way to introduce non-goals; but when removing the last goal,  $x$  would have to stand for a non-goal. So  $\overset{\text{ax}}{\rightarrow}$

$$y - a \rightarrow \textcircled{\lambda} - a \rightarrow x \quad \overset{\text{ax}}{\rightarrow} \quad y - a \rightarrow \textcircled{\lambda} \textcircled{\lambda} - a \rightarrow x$$

is used at some point. The only non-goals that are introduced have an arc labelled with an atom. Consider an application of this micro-move. At that point  $y$  may stand for a non-goal or a goal. If it is a non-goal, it has become a non-goal due to an earlier application of  $\overset{\text{ax}}{\rightarrow}$ . Otherwise it will do so at some point. By moving from node to node along arcs labelled with the same atom this way, it is possible to create an infinite path in the graph, which contradicts the fact that it is acyclic. Therefore  $f_0$  or  $f_1$  is  $\top$ .  $\square$

An interaction is a win for player  $\lambda$  if its normal form  $(\cdot, f_0, f_1)$  is such that  $f_\lambda = \perp$ ; it is a tie if  $f_0 = f_1 = \top$ .

**Definition 4.7.7** (Winning strategy). *Let  $p$  be a simple position. A  $\lambda$ -strategy for  $p$  is winning iff it wins against all  $\bar{\lambda}$ -strategies for  $p$ .*

**Proposition 4.7.8.** *A  $\lambda$ -strategy for a simple position is winning iff it does not specify any daimon macro-moves.*

*Proof.* Let  $\sigma$  be  $\lambda$ -strategy for  $p$ . If it does not specify any daimon macro-moves, it must win all interactions. Conversely, by Proposition 4.7.5, every  $\lambda$ -goal reachable by  $\sigma$  can be reached by an interaction with an appropriate  $\bar{\lambda}$ -strategy for  $p$ . If  $\sigma$  is winning, it wins all those interactions and therefore never plays daimon.  $\square$

## 4.8 Winning strategies and proofs

### 4.8.1 Strategy classes

A  $\lambda$ -strategy specifies moves for playable  $\lambda$ -subpositions, whether they are goals or not. Strategies must do so because the opponent must be supplied information in all cases. However, only goals are relevant from a proof-theoretic point of view. The information supplied at non-goals is used for purely interactive purposes. We will need to consider strategies modulo that information.

**Proposition 4.8.1.** *The successors of a non-goal by a macro-move are always non-goals. Therefore two  $\lambda$ -pre-strategies specifying the same macro-moves on  $\lambda$ -goals reach the same  $\lambda$ -goals from a  $\lambda$ -subposition.*

**Definition 4.8.2.** *Two  $\lambda$ -pre-strategies are goal-equivalent iff they specify the same moves for goals. A  $\lambda$ -pre-strategy class is a class for this equivalence relation. Two  $\lambda$ -strategies for a  $\lambda$ -subposition  $s$  are goal-equivalent iff they specify the same moves for goals. A  $\lambda$ -strategy class for  $s$  is a class for this equivalence relation.*

Note that if a strategy is winning, then so are its goal-equivalent strategies. The notion of winning strategy class is therefore well defined.

### 4.8.2 Goal trees

The definition of the set of reachable  $\lambda$ -subpositions in Definition 4.7.2 suggests an inductive presentation.

**Definition 4.8.3.** *A  $\lambda$ -goal tree is a tree of  $\lambda$ -goals presented as a derivation in a deductive system given by the two rules described below.*

$$\frac{g_1, \dots, g_n}{g} P, m$$

where  $g$  is playable,  $m$  is a  $\lambda$ -macro-move in  $g$  and the  $g_i$  are the goal successors of  $g$  by  $m$ .

$$\frac{g_1, \dots, g_n}{g} O$$

where  $g$  is not playable and the  $g_i$  are the successors of  $g$  by maximal sequences of opponent macro-moves.

**Definition 4.8.4.** *Let  $\sigma$  be a  $\lambda$ -strategy class for a  $\lambda$ -goal  $g_0$ . The goal tree for  $\sigma$  is the goal tree  $\mathcal{T}(\sigma)$  with root  $g_0$  and whose  $P$  rules are given by the macro-moves specified by  $\sigma$ .*

**Proposition 4.8.5.** *Let  $T$  be a goal tree.*

1. All the branches of  $T$  are finite;



2. no goal appears twice in  $T$ .

*Proof.* If  $s'$  is a successor of  $s$  by a macro-move and  $s$  and  $s'$  are both goals, then the number of symbols of the neutral expressions labelling the arcs is strictly less for  $s'$  than for  $s$ . Therefore all the branches of  $T$  are finite.

We now prove the second property. Suppose by contradiction that a  $\lambda$ -goal appears twice in  $T$ . The two occurrences  $h_1, h_2$  cannot be on the same branch, as that would allow infinite branches. Consider the closest common ancestor  $g$  of those occurrences in the tree.  $g$  cannot be playable, as its successors by the associated macro-move are clearly distinct: they can be part of the same neutral graph, and so can their descendants.  $g$  is therefore not playable. The tree contains

$$\frac{g_1, \dots, g_n}{g} \text{ O}$$

and (for example)  $g_1$  and  $g_2$  are the (distinct) ancestors of  $h_1$  and  $h_2$  respectively. Then  $g_1$  and  $g_2$  must have disjoint monomial weights by Proposition 4.6.8. The only way for  $h_1$  and  $h_2$ , as descendants of  $g_1$  and  $g_2$ , to have the same weight would thus be to have a null weight, which is impossible as the tree only contains goals.  $\square$

**Lemma 4.8.6.** *Let  $g$  be a  $\lambda$ -goal.  $\mathcal{T}$  defines a bijection from the  $\lambda$ -strategy classes for  $g$  to the  $\lambda$ -goal trees with root  $g$ .*

*Proof.* The injectivity part is simple. All the reachable  $\lambda$ -goals of a strategy appear in its goal tree, therefore all the macro-moves it specifies (i.e. all its information) too.

Surjectivity is easily seen too. By Proposition 4.8.5, no playable goal appears twice in a goal tree. Therefore a goal tree specifies exactly one macro-move for each playable goal it contains. It is then easy to see that these moves are precisely those specified by a  $\lambda$ -strategy class for  $g$ , which is naturally the antecedent of the tree by  $\mathcal{T}$ .  $\square$

**Theorem 4.8.7.** *Let  $g$  be a  $\lambda$ -goal. There is a one-to-one correspondence between  $\lambda$ -strategy classes for  $g$  and closed derivations of the sequent represented by  $g$ . Moreover, the strategy class is winning iff the derivation is a proof.*

*Proof.* The previous lemma relates  $\lambda$ -strategy classes to  $\lambda$ -goal trees. Now,  $\lambda$ -goal trees are in one-to-one correspondence with closed derivations.  $\lambda$ -goals in a tree correspond to their associated sequents in a closed derivation, a rule

$$\frac{g_1, \dots, g_n}{g} \text{ P}, m$$

corresponds exactly to a synchronous phase or a daimon rule depending on  $m$  (see Proposition 4.6.4) and a rule

$$\frac{g_1, \dots, g_n}{g} \text{ O}$$

corresponds exactly to a maximal layer of asynchronous phases (see Proposition 4.6.8).

A  $\lambda$ -strategy class is winning iff it does not specify daimon moves, iff its goal tree does not contain a daimon move, iff the corresponding closed derivation does not contain  $\boxtimes$ .  $\square$

# Extensions

In this chapter we develop some extensions of the game model presented in the previous chapters.

## 5.1 Explicit cuts

The approach we developed in the game models dual proof search by an interactive process. In the tradition of computation as proof search, the trace of the computation is seen as a cut-free sequent calculus proof, and sequents are successive states. Our games are slightly different in the sense that two proofs are being built at the same time. Cuts are not part of our proof system, but they can be added and accounted for in the game. This section presents this simple extension.

In focused proof systems, cut is introduced as a rule starting a new phase like the decision rules. The original cut rule in Andreoli's triadic system [And92] becomes, in our system:

$$\frac{\vdash \Gamma_1^+, \Lambda_1^- \Downarrow F^+ \quad \vdash \Gamma_2^+, \Lambda_2^-, F^-}{\vdash \Gamma_1^+, \Lambda_1^-, \Gamma_2^+, \Lambda_2^-}$$

Remark that the only way to extend the right branch is to apply  $[D\uparrow]$  by focusing on  $F^-$ . Including this step in the cut rule would be possible but it would break the clear separation of phases. The cut rule replaces  $[D\Downarrow]$  in the left branch. It seems natural that the player introducing the cut develops the left branch and leaves the asynchronous phase in the right branch to be developed by her opponent.

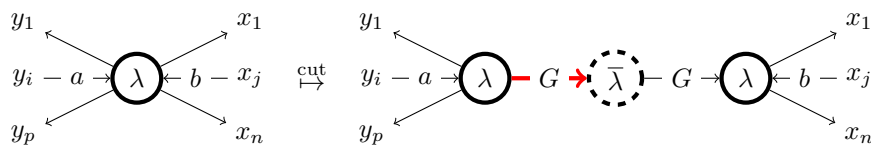
We introduce a new micro-move  $\overset{\text{cut}}{\mapsto}$  matching this rule. This requires a new kind of goal: cut nodes. A cut node will be graphically represented as



Cut nodes are goals and absolutely no distinction is made between them except that they are not taken into account in goal trees.

The micro-move  $\overset{\text{cut}}{\mapsto}$  plays a role similar to that of  $\overset{D}{\mapsto}$  and is presented on Figure 5.1. This micro-move creates a new cut node for the opponent, representing the sequent  $\vdash [G]^+ \uparrow [G]^-$ , which can be proved trivially. The micro-move highlights the fact that the cut and initial rules are the dual of each other.

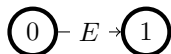
Just like  $\overset{D}{\mapsto}, \overset{\text{cut}}{\mapsto}$  is followed by a maximal sequence of other micro-moves, and together they form a macro-move. Instead of two kinds of macro-moves (regular and daimon), we now have three: regular, cut, and daimon.



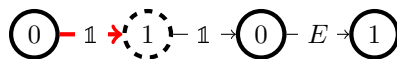
where  $G$  is a guarded neutral expression.

Figure 5.1: The cut micro-move  $\xrightarrow{\text{cut}}$ .

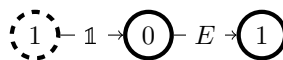
Starting from



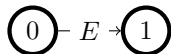
player 0 can cut



and finish the macro-move



At that point player 1 is the only one who can play and her only macro-move leads to



This sequence is then repeated forever, without ever giving a chance to player 1 to prove  $[E]^-$ .

Figure 5.2: An infinite interaction.

### 5.1.1 Infinite interactions

An immediate consequence of the introduction of the cut micro-move is that infinite sequences of macro-moves are possible, see Figure 5.2.

On the other hand, closed derivations are finite objects which contain finitely many instances of the cut rule. We can allow an interaction to be infinite, but playing cut infinitely many times must cause a player to lose the play. The interaction in Figure 5.2 must definitely be lost by player 0. The interesting question is whether it should be won by player 1, since our neutral setting allows ties. The answer is that it should be won by player 1, for a simple reason. If  $[E]^-$  is provable, then we want player 1 to have a winning strategy, therefore to be able to win all interactions with player 0. If  $[E]^-$  is not provable, then player 0 must have some way to force player 1 to fail without resorting to infinitely many cuts.

Cutting infinitely many times is not the only “losing condition” in our games. In contrast to other game models for logic, the neutral approach forces interaction to continue after a player has failed. In the concurrent game, we use a dedicated micro-move (the daimon) to this end. The state of a game, represented by a position  $(G, f_0, f_1)$ , keeps track of the applications of the daimon

in the Boolean flags  $f_0$  and  $f_1$ . As soon as a player plays the daimon, her flag becomes  $\top$  and, although interaction can go on, she cannot win it.

The use of cut macro-moves should also be recorded in positions. That way, it is possible to determine who is responsible for applying cut infinitely often in an infinite interaction. At least one player has to be, but it is possible that both be. We replace the Boolean flags with values in  $\overline{\mathbb{N}}$ , the set of natural numbers with the additional element  $\infty$ . We redefine macro-moves by modifying Definition 4.6.1.

**Definition 5.1.1** (Macro-move). *Let  $G$  be a playable  $\lambda$ -subposition. A  $\lambda$ -macro-move from  $G$  is a triple  $(G, G', n)$  where*

- $G \mapsto_{\lambda}^* G'$ ,  $G'$  does not rewrite through a  $\lambda$ -micro-move, and  $n$  is 0 if the first micro-move is  $\xrightarrow{D}$  or 1 if it is  $\xrightarrow{cut}$ ;
- or  $G'$  is obtained from  $G$  by the daimon transition of Definition 4.6.1, and  $n$  is  $\infty$ .

$n$  can take three values (0, 1 or  $\infty$ ) to identify different kinds of macro-moves. We now proceed to redefine how macro-moves affect positions.

**Definition 5.1.2.** *If  $m = (G, G', n)$  is a  $\lambda$ -macro-move and there is a position  $p$  of the form  $(C[G], n_0, n_1)$ , player  $\lambda$  may play  $m$  in  $p$ :*

$$(C[G], n_0, n_1) \xrightarrow{m} (C[G'], n'_0, n'_1)$$

where  $n'_\lambda = n_\lambda + n$  and  $n'_\bar{\lambda} = n_{\bar{\lambda}}$  with the usual notion of addition in  $\overline{\mathbb{N}}$ .

The idea behind this choice is that a simple counter is all that is necessary to determine if a player loses a finite or infinite interaction. A position records the number of uses of cut by each player. Let us fix a player  $\lambda$ . In a finite or infinite rewrite sequence of an interaction, the sequence of the values of  $n_\lambda$  is non-decreasing. Three situations are possible:

1. the sequence contains finite values only and its limit is finite;
2. the sequence contains finite values only and its limit is  $\infty$ ;
3. all the elements of the sequence equal  $\infty$  after a finite prefix.

The first case occurs when player  $\lambda$  uses finitely many cuts and no daimons, and thus does not fail. The second case occurs when player  $\lambda$  cuts infinitely many times but uses no daimons, and the third when player  $\lambda$  uses at least a daimon. Those two latter cases are the ones in which player  $\lambda$  fails. A way to discriminate between the first and the two last cases is to compare the limit of the sequence to  $\infty$ . The player has failed iff the limit is  $\infty$ . Note that this works for finite or infinite sequences, with the convention that the limit of a finite sequence is its last value.

This whole analysis is based on a sequence of moves. However, our game is concurrent and an interaction is a more complex object. Our objective is to take this fact into account and properly define the winning conditions in the game with cuts.

The concurrent nature of the game is not affected by adding cuts in the sense that Proposition 4.6.3 still holds. The definition of strategies must be slightly modified to take into account that a cut macro-move by the opponent spawns a cut node for the player. Since a player may play cut at any point in a play, choosing any guarded neutral expression as the one being cut, her

opponent must be prepared to play at any cut node possibly spawned by a cut rule. Strategies can take this into account by considering all those cut nodes as reachable subpositions. Note that in a cut micro-move the neutral expression occurrence  $G$  (from Figure 5.1) is not a sub-occurrence of neutral expression occurrences previously in the graph. We choose the cut node and  $G$  to be uniquely associated with this  $\overset{\text{cut}}{\mapsto}$  instance. This distinction allows player  $\bar{\lambda}$ 's strategies to behave independently in the spawned  $\bar{\lambda}$ -cut node for each instance of a cut micro-move.

Interactions of strategies keep some useful properties. Proposition 4.7.5 still holds, and interactions are still strongly confluent. However, interactions are not terminating in general, and therefore not strongly normalising in general. In fact they are not even weakly normalising in general, as a player can force all plays to be infinite by always playing cuts. We cannot define the winner of an interaction by its normal form as we did before. Instead, we observe from the interaction if the number of cuts a player plays is bounded or not, and if a player ever plays daimon or not.

**Definition 5.1.3.** *Let  $I$  be an interaction. For each player  $\lambda$ , consider the set  $\{n_\lambda \in \mathbb{N} \mid (G, n_0, n_1) \text{ is reachable by } I\}$ . This set has a least upper bound  $b_\lambda$ . Player  $\lambda$  wins  $I$  iff  $b_\lambda$  is finite.*

In other words, a player wins an interaction if she never plays daimon and plays only finitely many cuts. With this definition, the infinite interaction presented in Figure 5.2 is won by player 1 as intended.

Let us show here that with this new definition the two players cannot win the same interaction.

**Proposition 5.1.4.** *An interaction cannot be won by the two players.*

*Proof.* Consider, by contradiction, an interaction  $I$  won by both players.

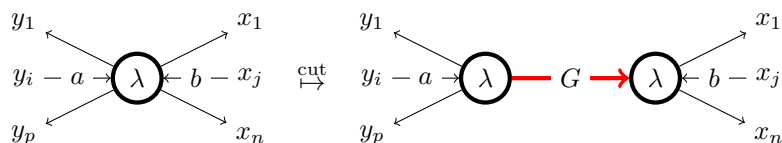
$I$  is terminating. Indeed, in an infinite play, at least one player cuts infinitely often. If there were an infinite play where player  $\lambda$  cut infinitely often, the least upper bound of the set  $\{n_\lambda \in \mathbb{N} \mid (G, n_0, n_1) \text{ is reachable by } I\}$  would be  $\infty$  and player  $\lambda$  would not win the interaction.

$I$  is strongly normalising and has a normal form  $(\cdot, n_0, n_1)$ . By using the same arguments as in the proof of Proposition 4.7.6, it is clear that at least one player has to play daimon. Therefore, at least one of  $n_0$  and  $n_1$  is infinite, which contradicts the hypothesis that  $I$  is won by both players.  $\square$

Winning strategies are still defined as strategies winning all the interactions against other suitable strategies. By analogy with Proposition 4.7.8, a strategy is winning iff it specifies no daimon and only finitely many cuts.

### 5.1.2 An alternate cut micro-move

In [HHH08], the authors model cut with a move that would correspond to the following micro-move in our game:



Starting from

$$\textcircled{0} - E \rightarrow \textcircled{1}$$

player 0 cuts

$$\textcircled{0} - \mathbf{1} + F \rightarrow \textcircled{0} - E \rightarrow \textcircled{1}$$

and finishes the macro-move by choosing  $\mathbf{1}$  over  $F$

$$\textcircled{0} - E \rightarrow \textcircled{1}$$

Figure 5.3: A macro-move based on the alternate cut micro-move.

where  $G$  is a guarded neutral expression. Obviously, we could not introduce such a micro-move in our context, since we require neutral graphs to be bipartite. If we managed to circumvent this issue, we would be able to obtain the equivalence of provability and existence of a winning strategy. However, we could not relate winning strategies to proofs easily. Figure 5.3 shows a macro-move using this alternate cut micro-move. We would like this macro-move to be read by player 0 as the derivation

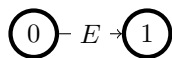
$$\frac{\frac{\overline{\vdash \downarrow \mathbf{1}} \ [1]}{\vdash \downarrow \mathbf{1} \oplus [F]^+} \ [\oplus_1] \quad \frac{\frac{\vdash [E]^+}{\vdash [E]^+ \uparrow \perp} \ [\perp] \quad \vdash [E]^+ \uparrow [F]^- \quad \vdots}{\vdash [E]^+ \uparrow \perp \ \& \ [F]^-} \ [\&]}{\vdash [E]^+} \text{ cut}}$$

But in fact, we only get the slice

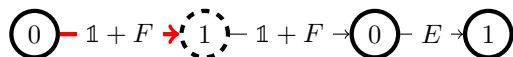
$$\frac{\frac{\overline{\vdash \downarrow \mathbf{1}} \ [1]}{\vdash \downarrow \mathbf{1} \oplus [F]^+} \ [\oplus_1] \quad \frac{\frac{\vdash [E]^+}{\vdash [E]^+ \uparrow \perp} \ [\perp]}{\vdash [E]^+ \uparrow \perp \ \& \ [F]^-} \ [\&_1]}{\vdash [E]^+} \text{ cut}}$$

The issue arises because the arc introduced by the cut connects two 0-goals. Player 0 gets to choose between  $\mathbf{1}$  and  $F$ . From the point of view of the source of the arc, it makes perfect sense since that choice is seen as the application of the  $[\oplus_1]$  rule over the  $[\oplus_2]$  rule. From the point of view of the target of the arc, player 0 should be prepared to face both  $\mathbf{1}$  and  $F$ , and the choice should be made by player 1. Here player 0 only faces  $\mathbf{1}$ . Notice that our definition of strategy is done on a subposition basis, hence a strategy for player 0 would still be required to face both choices. However there would be a discrepancy between the definition of strategy and the dynamics of interaction. Here the part of the strategy dealing with  $F$  would not be reachable by any interaction, making Proposition 4.7.5 false. With our notion of cut micro-move, the situation is different (see Figure 5.4). By inserting a cut node in the middle, we give the opponent the option to “forward” the moves (which is also known as a copycat strategy) or to play differently, thus solving the issue presented above.

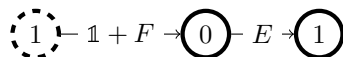
Starting from



player 0 cuts



and finishes the macro-move by choosing  $\mathbb{1}$  over  $F$



Then, player 1 could focus on  $\mathbb{1} + F$  and choose  $\mathbb{1}$ . This copycat strategy is a safe option. Player 1 could also choose  $F$ , after having played daimon on the cut node if needed (if  $F = \emptyset$ , for example).

Figure 5.4: Interaction based on our cut micro-move.

### 5.1.3 Winning strategies and proofs

In the game without cut, strategies contain more information than closed derivations because they must specify how to play on non-goals, while goals are the only nodes with information relevant to the derivation the player builds. By removing this information in a shift to strategy classes, we obtain a one-to-one correspondence. In the game with cuts, cut nodes are also irrelevant. They are introduced by the cut moves of the opponent and are purely here to allow interaction with her. However, cut nodes are goals and a player may lose by playing on them. There is a straightforward and deterministic way to treat them, though, by playing copycat.

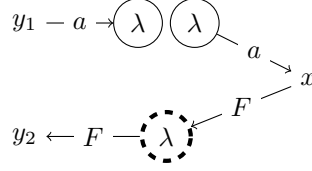
Playing copycat on a cut node consists in copying the opponent's macro-moves, maintaining the invariant that, before each opponent macro-move affecting it, a cut node has one incoming arc and one outgoing arc labelled with the same guarded neutral expression. When units or atoms are reached, the node is removed with the appropriate micro-move. It is exactly the same as deriving the general axiom  $\vdash F^-, F^+$  from the atomic initial rule

$$\frac{}{\vdash L^- \Downarrow L^+} \text{[init]}$$

For example the opponent macro-move

$$\begin{array}{c} \text{--- } (a + E) \times \Downarrow F \text{ ---} \rightarrow \textcircled{\lambda} \text{ --- } (a + E) \times \Downarrow F \text{ ---} \rightarrow x \\ \\ \begin{array}{c} y_1 \text{ --- } a \text{ ---} \textcircled{\lambda} \text{ --- } (a + E) \times \Downarrow F \text{ ---} \rightarrow x \\ \bar{\lambda} \text{ ---} \textcircled{\lambda} \\ y_2 \text{ --- } F \text{ ---} \textcircled{\lambda} \end{array} \end{array} \quad (5.1)$$

is followed by a  $\lambda$ -macro-move to



Copycat macro-moves are of the form  $(G, G', 0)$  and have no impact on the fact that a strategy is winning or not.

**Definition 5.1.5.** *A strategy is cut-copycat if it always plays copycat on cut nodes.*

A cut-copycat strategy is essentially composed of the same kind of information as strategies in the previous game, along with all the (deterministic and generic) information for playing copycat on cut nodes. By taking goal-equivalence classes, we get the abstraction we need.

**Theorem 5.1.6.** *Let  $g$  be a  $\lambda$ -goal. There is a one-to-one correspondence between winning cut-copycat  $\lambda$ -strategy classes for  $g$  and proofs of the sequent represented by  $g$  in the proof system with cut.*

*Proof.* We do not give the details of the proof here, as this result is a slight variation of those in Section 4.8. Goal trees are extended by adding a case for the  $P$  rule corresponding to a cut macro-move in the game and to a cut phase in the proof system. Notice that goal trees are finite because we restrict ourselves to winning strategies here, which cannot play infinitely many cuts.  $\square$

## 5.2 Exponentials

We can add a form of infinite behaviour to the game with exponential connectives. The exponentials of linear logic are not the best candidates. First, our neutral approach forces us to handle connectives symmetrically, which cannot be done easily with  $!$  and  $?$ . Second, the fact that  $!$  and  $?$  are not canonical implies that they cannot be reduced to their behaviour.

We define other connectives that are infinite versions of the MALL connectives. For each subset  $I$  of the set of natural numbers  $\mathbb{N}$ , we add a pair of dual unary connectives  $!_I$  and  $?_I$  to the logic. They should be seen as infinite synthetic connectives:

$$!_I F \equiv \bigoplus_{n \in I} \bigotimes_{i=1}^n F \quad ?_I F \equiv \bigotimes_{n \in I} \bigoplus_{i=1}^n F$$

Our sequent calculus can be extended to those connectives with the rules

$$\frac{\{\vdash \Gamma_i \Downarrow F, \Delta_i \mid 1 \leq i \leq n\}}{\vdash \Gamma_1, \dots, \Gamma_n \Downarrow !_I F, \Delta_1, \dots, \Delta_n} [!_I, n] \quad \frac{\{\vdash \Gamma \Uparrow F^{(1)}, \dots, F^{(n)}, \Delta \mid n \in I\}}{\vdash \Gamma \Uparrow ?_I F, \Delta} [?_I]$$

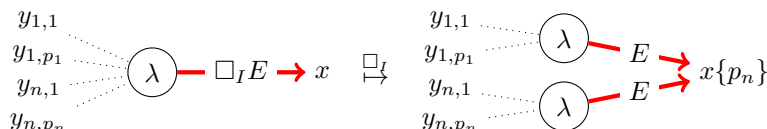
In  $[!_I, n]$ ,  $n$  is an element of  $I$ . In  $[?_I]$ ,  $F^{(1)}, \dots, F^{(n)}$  are  $n$  distinct occurrences of  $F$ . Note that  $[?_I]$  may have infinitely many premises.



We define a corresponding neutral unary connective  $\Box_I$ , and the translations  $[\Box_I F]^+ = !_I[F]^+$  and  $[\Box_I F]^- = ?_I[F]^-$ .

In order to define the associated micro-move, we need to adapt the notion of monomial weight to infinite additives. Instead of considering Boolean variables, we will more generally have  $I$ -variables, where  $I$  is a subset of  $\mathbb{N}$ . If  $p$  is an  $I$ -variable, then there is a literal  $p_n$  for each  $n \in I$ . We have the equations  $p_i p_j = 0$  if  $i \neq j$ . A weight is a product of literals.

We can now define the micro-move:



where  $n \in I$  and  $p$  is an  $I$ -variable uniquely associated with the  $\Box_I E$  occurrence. Here, player  $\lambda$  chooses  $n$  and then splits the  $\lambda$ -node like in the  $n$ -ary  $\overset{\Box_I}{\rightarrow}$  micro-move.

In the absence of cut micro-moves, the interaction of two strategies is still terminating because for every  $I \subset \mathbb{N}$  the  $\overset{\Box_I}{\rightarrow}$  micro-move replaces a neutral expression with finitely many neutral expressions with one less occurrence of the  $\Box_I$  symbol.

The canonicity of those exponential connectives can be discussed. First of all, two ‘‘colours’’ of the exponentials defined by a common set of indices  $I$  are provably equivalent. However, two distinct sets of indices define exponentials that are provably not equivalent. The main characteristic of this parametrisation is that it defines an order on the  $!_I$  that is the same as inclusion for the  $I$ . That is,  $\vdash !_I A \multimap !_J A$  is provable if and only if  $I \subset J$ . A proper proof-theoretic treatment of multiple modalities would allow to reason on them finitely, in contrast to the possible infinity of choices for the  $\overset{\Box_I}{\rightarrow}$  micro-move. Investigations were carried out in similar areas in [DJS93] and with *subexponentials* in [NM09]. In a recent paper [HHH09] the authors further developed the graph game model from [HHH08] to account for exponentials, by internalising the  $?C$  rule into the  $\otimes$  rule.

### 5.3 First-order quantification

We have defined our game in a purely propositional setting. Extending it to first-order logic is possible. Our proof system can be extended with the first-order rules

$$\frac{\vdash \Gamma \Downarrow F\{t/x\}, \Delta}{\vdash \Gamma \Downarrow \exists x F, \Delta} [\exists] \quad \frac{\vdash \Gamma \Uparrow F, \Delta}{\vdash \Gamma \Uparrow \forall x F, \Delta} [\forall]$$

where  $x$  does not appear as a free variable in the conclusion of  $[\forall]$ .

Hintikka’s games also contain first-order quantification and atoms. Éloïse plays at  $\exists x F$  by selecting an instance  $F\{t/x\}$  and Abélard does the same at  $\forall x F$ . In other words, first-order quantifiers are treated like infinite conjunctions and disjunctions. When the plays ends up at a ground atom, an external model tells who wins. This kind of game thus determines the truth of a formula in a given model.

Our approach is different as we aim at modelling provability (and, in the concurrent game, proofs). For example, in the propositional setting we deal with atoms through an axiom micro-move. In a game modelling truth, atoms would have been treated like the units of the logic through the use of an external propositional model.

First-order quantifiers are more than infinite versions of additive connectives. While proving an existential amounts to selecting an instance, proving a universal is more than proving all the ground instances of the formula; it consists in finding a *generic* argument that covers all instances.

In this section, we restrict ourselves to modelling provability in the neutral setting. Modelling proofs is substantially more complex and less elegant.

In our neutral setting, when a player faces  $\exists xF$ , she picks an instance  $F\{t/x\}$ . The opponent, who faces  $\forall xF^\perp$ , is then challenged to go on with  $F^\perp\{t/x\}$ . A winning strategy for the opponent must account for all choices for  $t$ . In general this does not yield a generic argument. It does, however, when the language contains infinitely many constant symbols. We give an informal explanation of this reason here. With an infinite supply of constant symbols we can always pick a fresh one to instantiate a universal quantifier. If  $\vdash \Gamma, F\{t/x\}$  is provable for every ground term  $t$ , then  $\vdash \Gamma, F\{c/x\}$  is provable, where  $c$  is a constant symbol not appearing in  $\Gamma$  or  $F$ . In any proof of that sequent,  $c$  plays the same role as an eigenvariable; therefore  $\vdash \Gamma, \forall xF$  is provable.

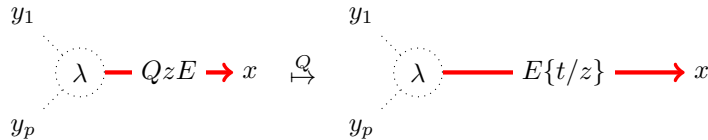
For the rest of the section, we will assume that the term language contains infinitely many constant symbols.

We extend the syntax of neutral expressions to include a neutral quantifier.

$$\begin{aligned} G, G' &::= a(t) \mid E + E' \mid \mathbb{0} \mid E \times E' \mid \mathbb{1} \mid QxE \\ E, E' &::= G \mid \dagger G \end{aligned}$$

We define the translations as  $[QxE]^+ = \exists x[E]^+$  and  $[QxE]^- = \forall x[E]^-$ . This is natural since  $\exists$  is synchronous and  $\forall$  is asynchronous.

We add a new micro-move



where  $t$  is a ground term. This micro-move is read as:

The $\lambda$ -node		$x$
$\frac{\vdash \Gamma \Downarrow [E]^+\{t/z\}, \Delta}{\vdash \Gamma \Downarrow \exists z[E]^+, \Delta} \quad [\exists]$		$\frac{\{\vdash \Gamma \Uparrow [E]^-\{t/z\}, \Delta \mid t \text{ ground term}\}}{\vdash \Gamma \Uparrow \forall z[E]^-, \Delta}$

Note that the restriction to instantiate  $z$  with a ground term in this micro-move is irrelevant since the sequents do not have eigenvariables.

The reading of the micro-move by  $x$  is not the  $[\forall]$  rule, but it is equivalent in terms of provability. In other words, the existence of a winning strategy is still equivalent to provability, but there are more winning strategy classes than proofs.

If we aimed at having a correspondence between winning strategy classes and proofs, we would need to impose a uniformity condition on strategies. For example, let  $c$  be a constant of the language and consider the following proof:

$$\frac{\frac{\frac{\overline{\overline{x; \vdash \top, P^-(x)}}}{x; \vdash P^+(c) \oplus \top, P^-(x)} [D\Downarrow, \oplus_2, R\Downarrow]}{\vdash \forall x(P^+(c) \oplus \top) \wp P^-(x)} [D\Uparrow, \forall, \wp, R\Uparrow, \text{atomic } R\Uparrow]} [D\Uparrow, \top]$$

With the other rule for the universal quantifier, there is a proof of the form

$$\frac{\frac{\frac{\overline{\overline{\vdash \top, P^-(c')}}}{\vdash P^+(c) \oplus \top, P^-(c')} [D\Downarrow, \oplus_2, R\Downarrow]}{\vdash \forall x(P^+(c) \oplus \top) \wp P^-(x)} [D\Uparrow, \top]}{\frac{\frac{\overline{\overline{\vdash P^+(c) \oplus \top, P^-(c)}}}{\vdash P^+(c) \oplus \top, P^-(c)} [D\Downarrow, \oplus_1, \text{init}]}{\vdash \forall x(P^+(c) \oplus \top) \wp P^-(x)} \dots}$$

in which the first branch, instantiated with a fresh constant  $c'$ , plays the role of the generic proof above, and the second branch, instantiated with  $c$ , is a specialised proof. Winning strategy classes allow this kind of behaviour and contain too much information to correspond to proofs. This issue could be tackled by forbidding specialised behaviour. If a playable subposition  $p$  can be obtained from another,  $p_0$ , by uniformly substituting ground terms for constants, a strategy would be required to play in  $p$  the image by the substitution of what it would play in  $p_0$ .

## 5.4 Equality

First-order quantification opens the way to other additions, such as equality. Equality has an interesting treatment in proof theory. First of all, we extend the syntax of formulae with two new constructors,  $t = s$  and  $t \neq s$ , where  $t$  and  $s$  are terms. Those two formulae are De Morgan duals.

When discussing first-order quantification, we presented two approaches to the universal quantifier  $\forall$ . The proof-theoretic, syntactic approach, uses the inference rule  $[\forall]$  and introduces eigenvariables in sequent. The semantic approach uses no eigenvariables at the price of having a rule  $[\forall']$  with infinitely many premises, one for each ground term of the language.

Similarly, there are two approaches to equality. In the syntactic tradition, our proof system would be extended with the three following inference rules:

$$\frac{}{\vdash \Downarrow t = t} [=] \quad \frac{\vdash \Gamma \theta \Uparrow \Delta \theta}{\vdash \Gamma \Uparrow t \neq s, \Delta} [\neq] \quad \frac{}{\vdash \Gamma \Uparrow t \neq s, \Delta} [\neq']$$

where  $t$  and  $s$  stand for terms. In  $[\neq]$ ,  $s$  and  $t$  are unifiable and  $\theta$  is their most general unifier. In  $[\neq']$ ,  $s$  and  $t$  are not unifiable. In a higher-order setting, the  $[\neq]$  rule would have one premise per substitution  $\theta$  in some complete set of unifiers of  $t$  and  $s$ .

This treatment is purely syntactic in that it handles unification explicitly. This complements the fact that the  $[\forall]$  rule requires the explicit introduction of

eigenvariables. For example, in the proof

$$\frac{\frac{\frac{\overline{\overline{z; \vdash \uparrow z = z}} \text{ [R}\uparrow, \text{D}\downarrow, =]}{\vdash \uparrow z = z} \text{ [}\neq\text{]}}{\vdash \uparrow x = y, x \neq y} \text{ [}\neq\text{]}}{\vdash \uparrow \forall x \forall y x = y \wp x \neq y} \text{ [}\forall, \wp\text{]}$$

it is clear that the introduction of the eigenvariables  $x$  and  $y$  is complemented by their unification carried out by the  $[\neq]$  rule.

Our approach to quantification using the  $[\forall']$  rule is more semantic in nature. Due to the absence of eigenvariables,  $t$  and  $s$  are ground terms in formulae of the form  $t = s$  or  $t \neq s$  (assuming that we work with sentences). In that semantic approach, the inference rules for equality simply become

$$\frac{}{\downarrow t = t} \text{ [}=\text{]} \quad \frac{\vdash \Gamma \uparrow \Delta}{\vdash \Gamma \uparrow t \neq t, \Delta} \text{ [}\neq\text{]} \quad \frac{}{\vdash \Gamma \uparrow t \neq s, \Delta} \text{ [}\neq'\text{]}$$

where  $t$  and  $s$  are distinct ground terms. In other words, equality/disequality of two ground terms is treated

- multiplicatively, like  $1/\perp$ , when those terms are the same;
- additively, like  $0/\top$ , when those terms are distinct.

In that approach, the above proof becomes

$$\frac{\dots \frac{}{\vdash \uparrow t = s, t \neq s} \text{ [}\neq'\text{]} \quad \dots \frac{\frac{\overline{\overline{\vdash \uparrow t = t}} \text{ [R}\uparrow, \text{D}\downarrow, =]}{\vdash \uparrow t = t} \text{ [}\neq\text{]}}{\vdash \uparrow t = t, t \neq t} \text{ [}\neq\text{]} \quad \dots}{\vdash \uparrow \forall x \forall y x = y \wp x \neq y} \text{ [}\forall', \wp\text{]}$$

where the  $[\forall', \wp]$  layer has one premise for each pair of ground terms  $(t, s)$  of the language. Each premise is proved in one of two ways, depending on whether  $t$  and  $s$  are the same or not.

Equality can be built in the game by introducing a new constructor  $t \doteq t'$  for guarded neutral expressions, where  $t$  and  $t'$  are terms. We define the translations as  $[t \doteq t']^+ = (t = t')$  and  $[t \doteq t']^- = (t \neq t')$ .

Unsurprisingly, the micro-moves for equality match those for  $\mathbf{1}$  and  $\mathbf{0}$ . In the case of two identical ground terms:

$$\textcircled{\lambda} \text{ --- } t \doteq t \text{ --- } x \quad \overset{\doteq}{\mapsto} \quad x$$

In the case of two distinct ground terms  $t$  and  $s$ :

$$\textcircled{\lambda} \text{ --- } t \doteq s \text{ --- } x \quad \overset{\doteq}{\mapsto} \quad x\{0\}$$

## 5.5 Fixed points

Fixed points can be added to the game. We briefly present how they were introduced in [MS06]. Let  $\iota$  denote the type of terms and  $\epsilon$  the type of neutral expressions. Then consider, for each natural number  $n$ , the neutral fixed point operator  $\text{fix}_n$  of type  $(\alpha_n \rightarrow \alpha_n) \rightarrow \alpha_n$ , where  $\alpha_n = \iota \rightarrow \dots \rightarrow \iota \rightarrow \epsilon$  ( $n$

occurrences of  $\iota$ ). The associated micro-move unfolds the fixed point:

$$\begin{array}{ccc}
 \begin{array}{c} y_1 \\ \vdots \\ \lambda \\ \vdots \\ y_p \end{array} & \xrightarrow{\text{fix}_n B \vec{t}} & x \\
 & & \downarrow \text{fix}_n \\
 \begin{array}{c} y_1 \\ \vdots \\ \lambda \\ \vdots \\ y_p \end{array} & \xrightarrow{B(\text{fix}_n B) \vec{t}} & x
 \end{array}$$

where  $\beta$ -reduction is carried out.

The neutral fixed point operator shall be translated positively as the least fixed point operator  $\mu$  and negatively as the greatest fixed point operator  $\nu$ . As usual, we require the body  $B$  of any fixed point  $\text{fix}_n B$  to be monotonic: if  $B$  is written  $\lambda X.T$ , then  $X$  may only occur in  $T$  under an even number of  $\lambda$ .

Unfolding a fixed point allows plays to inspect a finite portion of a recursive structure. A player infinitely unfolding fixed points shall lose the play. Note that fixed point unfolding alone does not give the ability to reason by induction or coinduction, which requires a much more involved proof-theoretic treatment [Bae08] that cannot be adapted to our neutral setting easily. It is however possible to e.g. decide bisimulation in a similar way to Stirling's game [Sti99]. Informally speaking, if two processes are bisimilar, it is possible to tell them apart after finitely many transitions.

# Related and future work

---

Most of the work on game semantics for linear logic models cut elimination in a setting where the Player’s strategies are the objects of interest, and the Opponent represents an abstract environment. Composition is defined for strategies, which are global objects containing information about the whole game. Composition corresponds to cut elimination: computation is defined on *complete* proofs. In our case, we see computation as an ongoing process exploring an unknown search space. We precisely capture the dynamics of the application of individual inference rules, *gradually building* a neutral object which will eventually turn out to be a proof of a formula  $F$ , a proof of  $F^\perp$ , or a refutation of both. In that sense, our games are more closely related to the dialogue games inspired by the early work of Lorenzen [Lor61] (see also [Her95]).

The closest related work, by Hirshowitz *et al.* [HHH08], investigates a similar graph-based game for atom-free MALL with a topological flavour: moves modify graphs continuously. The game explicitly contains cut moves, the authors provide a cut elimination procedure for strategies, and discuss the composition of strategies. Their focus is somewhat different from our “neutral approach” line of research.

Game-theoretic accounts of concurrency in logic exist in concurrent [AM99] and asynchronous [Mel04] games. It would be interesting to connect our concurrent game to the promising asynchronous line of research. Although our notion of concurrency is naturally dictated by the structure of the neutral graphs, it could be modelled by a homotopy relation on sequences of moves. The asynchronous model is quite flexible. For example Mimram [Mim08] investigated asynchronous games in which the players are not forced to alternate.

Ludics [Gir01] is an account of logic based on interaction. In that framework, interaction is the primitive notion that allows to recover logic. Our work shares some similarities with Ludics. Our comprehensive use of focalisation is natural in our neutral approach, since synchrony corresponds to activity and asynchrony to passivity. The fact that focalisation is primitive in Ludics stems from a similar observation on interaction. Our account of failure motivated the introduction of the daimon, by analogy with Ludics. However, our game were designed with dual proof search in mind with a strong emphasis on syntax and finite behaviour, while Ludics deals with infinite objects which are more primitive than both syntax and semantics.

Our goal here has been rather conservative: we have worked entirely within the proof theory of MALL, attempting to find a symmetric and interactive account of it. Many of the other works, e.g., on Ludics and L-nets [CF05, FM05], are willing to move beyond MALL and to develop new logical principles motivated by interaction.

Another relevant question is the connection between the computation-as-proof-normalisation and computation-as-proof-search traditions. By providing

an interactive process closely following proof search, and at the same time seeing interaction of strategies as proof normalisation, our work has some connections with both paradigms. Recent work aiming at relating the two was developed by Saurin—albeit in an asymmetric setting—in which proof search is guided by normalisation with tests [Sau08a].

# Conclusion

---

In this thesis we developed an original approach to the proof theory of MALL using symmetric two-player games. We describe a setting in which two agents play with the same rules, one of them attempting to prove a formula  $F$  and the other one attempting to prove  $F^\perp$ . The current state of the game is a single object seen as two orthogonal derivations of  $F$  and  $F^\perp$ , and each move is seen as the simultaneous extension of those derivations with dual inference rules. In other words, the game is a process building a syntactic object which will eventually turn out to be a proof of a formula  $F$ , a proof of  $F^\perp$ , or a refutation of both. A single computation accounts for the two simultaneous orthogonal proof searches. We summarise this outline under the slogan “*neutral approach*”. Focalisation is used to structure proof search in alternating phases of “*don’t know*” *nondeterminism* corresponding to player moves, and “*don’t care*” *nondeterminism* corresponding to opponent moves. The interaction between multiplicatives of opposite polarities requires the introduction of graph structures representing a multicut between sequent calculus derivations, or cut links between two dual proof structures. Since the game is indeterminate, we also internalise failure in the plays.

The locality of rewriting brought us to define a concurrent neutral game in which plays are replaced with confluent applications of moves by the players. Proofs correspond to winning strategies modulo some information which is only used to interact with the other player. Our use of concurrency is dictated by the structure of neutral graphs themselves. Each subposition in a neutral graph is treated independently of the others, and the arcs mark dependencies between moves operated on them. We extended this game with explicit cut moves, and several accounts of infinity: first-order structure, exponentials, and fixed points.





# Bibliography

- [Abr03] Samson Abramsky. Sequentiality vs. concurrency in games and logic. *Mathematical Structures in Computer Science*, 13(4):531–565, August 2003.
- [AJ94] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *J. of Symbolic Logic*, 59(2):543–574, 1994.
- [AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, 2000.
- [AM99] Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *14th Symp. on Logic in Computer Science*, pages 431–442. IEEE Computer Society Press, 1999.
- [And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 2(3):297–347, 1992.
- [Bae08] David Baelde. *A linear approach to the proof-theory of least and greatest fixed points*. PhD thesis, École Polytechnique, December 2008.
- [Bla92] Andreas Blass. A game semantics for linear logic. *Annals Pure Appl. Logic*, 56:183–220, 1992. Special Volume dedicated to the memory of John Myhill.
- [CF05] Pierre-Louis Curien and Claudia Faggian. L-nets, strategies and proof-nets. In C.-H. Luke Ong, editor, *CSL 2005: Computer Science Logic*, volume 3634 of *LNCS*, pages 167–183. Springer, 2005.
- [CLN98] Roberto Di Cosmo, Jean-Vincent Loddo, and Stephane Nicolet. A game semantics foundation for logic programming (extended abstract). In C. Palamidessi, H. Glaser, and K. Meinke, editors, *Principles of Declarative Programming: 10th International Symposium PLILP'98*, number 1490, pages 355–373, 1998.
- [CMS08] Kaustuv Chaudhuri, Dale Miller, and Alexis Saurin. Canonical sequent proofs via multi-focusing. In G. Ausiello, J. Karhumäki, G. Mauri, and L. Ong, editors, *Fifth International Conference on Theoretical Computer Science*, volume 273 of *IFIP*, pages 383–396. Springer, September 2008.
- [Cur05] Pierre-Louis Curien. Introduction to linear logic and ludics, Part I. *Advances in Mathematics (China)*, 34(5):513–544, January 2005.
- [Cur06] Pierre-Louis Curien. Introduction to linear logic and ludics, Part II. *Advances in Mathematics (China)*, 35(1):1–44, January 2006.

- [DJS93] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Kurt Gödel Colloquium*, volume 713 of *LNCS*, pages 159–171. Springer, 1993.
- [DM08] Olivier Delandé and Dale Miller. A neutral approach to proof and refutation in MALL. In F. Pfenning, editor, *23th Symp. on Logic in Computer Science*, pages 498–508. IEEE Computer Society Press, 2008.
- [DMS09] Olivier Delandé, Dale Miller, and Alexis Saurin. Proof and refutation in mall as a game. Accepted to a special issue of *Annals of pure and applied logic*, May 2009.
- [DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.
- [FH02] Claudia Faggian and Martin Hyland. Designs, disputes and strategies. In *Computer Science Logic*, volume 2471 of *LNCS*, pages 713–748. Springer, 2002.
- [FM05] Claudia Faggian and François Maurel. Ludics nets, a game model of concurrent interaction. In *20th Symp. on Logic in Computer Science*, pages 376–385. IEEE Computer Society, 2005.
- [Gen69] Gerhard Gentzen. Investigations into logical deductions. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- [Gir96] Jean-Yves Girard. Proof-nets: the parallel syntax for proof-theory. In Aldo Ursini and Paolo Agliano, editors, *Logic and Algebra*, volume 180 of *Lecture Notes In Pure and Applied Mathematics*, pages 97–124, New York, 1996. Marcel Dekker.
- [Gir98] J.-Y. Girard. On the meaning of logical rules II: multiplicatives and additives. Unpublished manuscript, February 1998.
- [Gir99] Jean-Yves Girard. On the meaning of logical rules I: syntax vs. semantics. In Berger and Schwichtenberg, editors, *Computational Logic*, pages 215–272. Springer, 1999.
- [Gir01] Jean-Yves Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, June 2001.
- [GRW08] Chrysidá Galanaki, Panos Rondogiannis, and William W. Wadge. An infinite-game semantics for well-founded negation in logic programming. *Annals of Pure and Applied Logic*, 151(2):70–88, 2008.

- [Her95] Hugo Herbelin. *Séquents qu'on calcule: de l'interprétation du calcul des séquents comme calcul de lambda-termes et comme calcul de stratégies gagnantes*. PhD thesis, Université Paris 7, 1995.
- [HHH08] André Hirschowitz, Michel Hirschowitz, and Tom Hirschowitz. Topological observations on multiplicative additive linear logic. Informal publication, 2008.
- [HHH09] André Hirschowitz, Michel Hirschowitz, and Tom Hirschowitz. Contraction-free proofs and finitary games for linear logic. Informal publication, 2009.
- [HO00] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I. models, observables and the full abstraction problem, II. dialogue games and innocent strategies, III. A fully abstract and universal game model. *Information and Computation*, 163:285–408, 2000.
- [Hod04] Wilfrid Hodges. Logic and games. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, 2004.
- [HS02] Martin Hyland and Andrea Schalk. Games on graphs and sequentially realizable functionals. In *Proceedings of the 17th IEEE Symposium on Logic in Computer Science (LICS-02)*, pages 257–264, Los Alamitos, July 22–25 2002. IEEE Computer Society.
- [Kan94] Max I. Kanovich. Simulating linear logic with 1-linear logic. Preprint 94-02, Laboratoire de Mathématiques Discrètes, University of Marseille, 1994.
- [Laf90] Yves Lafont. Interaction nets. In *17th ACM Symp. on Principles of Programming Languages*, pages 95–108, San Francisco, California, 1990. ACM Press.
- [Lak76] Imre Lakatos. *Proofs and Refutations*. Cambridge University Press, 1976.
- [Lau04a] Olivier Laurent. Polarized games. *Annals of Pure and Applied Logic*, 130(1–3):79–123, December 2004.
- [Lau04b] Olivier Laurent. A proof of the focalization property of linear logic. Unpublished note, May 2004.
- [LC00] Jean-Vincent Loddo and Roberto Di Cosmo. Playing logic programs with the alpha-beta algorithm. In *Logic for Programming and Automated Reasoning (LPAR)*, number 1955, pages 207–224. Springer, 2000.
- [LM07] Chuck Liang and Dale Miller. Focusing and polarization in intuitionistic logic. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646 of *LNCS*, pages 451–465. Springer, 2007. Extended version to appear in TCS.
- [LM08] Olivier Laurent and Roberto Maieli. Cut elimination for monomial MALL proof nets. In *LICS*, pages 486–497. IEEE Computer Society, 2008.

- [LMSS92] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals Pure Applied Logic*, 56:239–311, 1992.
- [Lod02] Jean-Vincent Loddo. *Généralisation des Jeux Combinatoires et Applications aux Langages Logiques*. PhD thesis, Université Paris VII, 2002.
- [Lor61] Paul Lorenzen. Ein dialogisches konstruktivitätskriterium. In *Infinistic Methods: Proceed. Symp. Foundations of Math.*, pages 193–200. PWN, 1961.
- [LS91] Y. Lafont and T. Streicher. Games semantics for linear logic. In Albert R. Meyer, editor, *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science*, pages 43–50, Amsterdam, The Netherlands, July 1991. IEEE Computer Society Press.
- [Mel04] Paul-André Melliès. Asynchronous games 2: The true concurrency of innocence. *Theoretical Computer Science*, 358(2–3):200–228, 2004.
- [Mim08] Samuel Mimram. *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle*. PhD thesis, HAL - CCSD, November 13 2008.
- [MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.
- [MS06] Dale Miller and Alexis Saurin. A game semantics for proof search: Preliminary results. In *Proceedings of the Mathematical Foundations of Programming Semantics (MFPS05)*, number 155 in Electronic Notes in Theoretical Computer Science, pages 543–563, 2006.
- [MS07] Dale Miller and Alexis Saurin. From proofs to focused proofs: a modular proof of focalization in linear logic. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646 of *LNCS*, pages 405–419. Springer, 2007.
- [NM09] Vivek Nigam and Dale Miller. Algorithmic specifications in linear logic with subexponentials. 2009. To appear in PPDP09.
- [PR04] David Pym and Eike Ritter. *Reductive Logic and Proof-search: proof theory, semantics, and control*, volume 45 of *Oxford Logic Guides*. Oxford University Press, 2004.
- [Sau08a] Alexis Saurin. Towards ludics programming: Interactive proof search. In *Logic Programming, 24th International Conference*, volume 5366 of *LNCS*, pages 253–268, December 2008.
- [Sau08b] Alexis Saurin. *Une étude logique du contrôle (appliquée à la programmation fonctionnelle et logique)*. PhD thesis, Ecole Polytechnique, September 2008.
- [Smu95] Raymond M. Smullyan. *First-Order Logic*. Dover Publications, 1995. Second edition.

- [Sti99] Colin Stirling. Bisimulation, modal logic and model checking games. *Logic Journal of the IGPL*, 7(1):103–124, 1999.
- [vE86] Maarten H. van Emden. Quantitative deduction and its fixpoint theory. *J. of Logic Programming*, 3(1):37–53, 1986.