SuperPower Lorenzen Games

Valeria de Paiva and Harley Eades III

May 2014

Abstract

This note discusses game semantics, in the style of Lorenzen, for Full Intuitionistic Linear Logic, following Blass and Rahman.

Blass on Linear Logic Games

Blass is responsible, to a large extent, for the introduction of game-theoretic ideas related to Linear Logic[1]. The project was enthusiastically embraced by Hyland, Abramsky and many others and there are now several conferences dedicated to the theme, especially to the application of game-theoretical ideas to programming language semantics.

The essence of the idea of a game theoretic semantics of a logic is that a game (or an argument, dialogue or protocol) consists of two players, one (the Proponent or Player) seeking to establish the truth of a formula under consideration (trying to prove it) while the other (the Opponent) disputing it, trying to prove it false. The two players alternate, attacking and defending their positions. As Blass puts it, the heart of the semantics consists of rules of the debate between the players.

The idea of using these dialogues goes back to, at least, Lorenzen in the late 50s. Lorenzen wanted to show that his dialogues were an alternative semantics for *Intuitionistic* Logic, but somehow the games looked much more like Classical Logic. Felscher came up with conditions that guaranteed that only constructive formulas were provable, but the conditions are not very transparent. Similarly Blass first introduced games for Linear Logic, but the composition of strategies in his games was not associative, a problem fixed by Abramsky and Jaghadeesan. But their semantics has problems of its own, as described in [3].

We recall that Blass presented two slightly different semantics for Linear Logic using games in[2, 1]. The semantics in [1] is for affine logic, that is the rule of weakening is validated. Both semantics say that to play the game $A\&B$, the opponent has to choose one of A or B and then the chosen game is played. And to play the game $A \oplus B$ the proponent chooses one of A or B and then the chosen game is played. A play of $A \otimes B$ consists of interleaved runs of the two games A and B, whenever it is the proponent's turn to play in either or both constituents, he must play there. When it's the opponent's turn to move in both components, he must choose one and move in it. Blass also provided an interpretation of the modality "of course" , denoted by "!", similar to a repeated tensor product. Since linear negation is interpreted simply as exchange of players, "par", the multiplicative disjunction, and linear implication were *not* described, as in Classical Linear Logic they can be defined in terms of tensor and linear negation.

Lorenzen Games

Lorenzen games were developed by Lorenz, Felscher and Rahman, who together with co-workers, established a collection of games for specific (non-classical) logics, including Linear Logic. This general framework was named Dialogic. We recap the basics of Lorenzen games, as expounded by the Rahman school ([5]).

The language L is composed of the standard components of first-order logic, with four connectives $\wedge, \vee, \rightarrow, \neg$, with small letters $(a, b, c, \ldots)$ for prime formulas and capital italic letters $(A, B, C, \ldots)$ for complex formulas. We need two labels, **O** and **P** (standing for the players, Opponent and Proponent, respectively). We also need some special force symbols: ?... and !... , where ? stands for attack (or question) and ! stands for defense. The set of rules in dialogic is divided into *particle* rules and *structural* rules. Particle rules describe the way a formula can be attacked and defended, according to its main connective. Structural rules, on the other hand, specify the general organization of the game. We restrict ourselves to propositional logic. In the following we describe three systems, intuitionistic logic, classical logic and classical linear logic, before we embark on the novelty of this note, *full intuitionistic* linear logic.

Games for Intuitionistic and Classical Logic

The table with particle rules for intuitionistic and classical logic is as follows:

Assertion	Attack	Response
$A \wedge B$	$\wedge_L$	A
$A \wedge B$	$\wedge_R$	B
$A \vee B$	?	A or B
$A \rightarrow B$	A	B
$\neg A$	A	–

Disjunction and conjunction differ only by the player who has the choice of the immediate subformula with which the game will proceed: in a conjunction, the challenger may choose, confident that either conjunct can be refuted; in a disjunction the choice lies with the defender. Thus, to defend a conjunction, a player must be able to defend any of the conjuncts, while in the case of a disjunction, it is sufficient to be able to defend one of the disjuncts. Most importantly to challenge an implication essentially amounts to providing a proof of the antecedent and claiming that the other player will fail to build a proof of the consequent from it. The defence against such an attack then consists of a proof of the consequent. And for negation, the only way to attack the assertion $\neg A$ is to assert A, and be prepared to defend this assertion. Thus there is no proper defence against such an attack, but it may be possible to counterattack the assertion of A. Lorenzen games were born in the context of constructivism, so this is not a surprise, the constructive negation of A is $A \to \bot$.

The difference between classical and intuitionistic logic resides on the structural rules. For classical logic the rules are:

S1 **P** may assert an atomic formula only after it has been asserted by **O**.

S4 A **P**-assertion may be attacked at most once.

S5 **O** can react only upon the immediately preceding **P**-statement.

while for intuitionistic logic the structural rules are the ones above plus:

S2 If p is an **P**-position, and if at round $n-1$ there are several open attacks made by **O**, then only the latest of them may be answered at n (and the same with **P** and **O** reversed).

S3 An attack may be answered at most once.

A *dialogue* can be thought of as a set of dialogical games, structured as a tree, the root of which is the thesis of the dialogue. Splits in the dialogue tree are generated by the propositional choices of the Opponent. Any possible attack by the Opponent against a conjunction, any possible defence of a disjunction, and either possible reaction in the case of an attack by the Proponent against an implication he defends (counterattack or defence) will generate a new branch in the dialogue tree. No move of the Proponent will open a new branch. A completed dialogue tree will thus contain all the Opponent's possible choices.

Starting with the thesis produced by the Proponent, the dialogue unravels connectives, until we get to atomic propositions. A dialogical game is said to be closed iff there is some atomic formula which has been played by both players. A dialogical game is finished iff it is closed or the rules do not allow any further move by the player who has to move.

For a set S of dialogue rules governing how the game is to proceed, a formula A is S-valid iff Proponent **P** has a *winning strategy* for A. A strategy for a player in a dialogical game is a function telling him what to do, according to what has previously happened in the game, to win it.

The problem with the structural rules is that it is not clear which modifications can be made to them without changing the set of provable formulas.

Lorenzen Games for Linear Logic

We now recall Rahman's[7] definitions of Lorenzen games for Linear logic, see [6]. In Linear Logic each occurrence of one formula in a proof must be taken as a distinct resource for the inference process. In the dialogical framework, each move is an action. It is therefore natural to consider that two distinct moves are different actions, consuming different resources, even when the two moves have the same propositional content. But we have to introduce some book-keeping to deal with usage of resources.

To keep linearity and show validity of a formula we must use all and each formula that has been asserted throughout the dialogue. And we cannot use one round more than once.

To adapt the Lorenzen games to Linear Logic, we have to provide a new table of particle rules (for the new linear connectives) and we have to adopt some new principles, like:

1. An atomic **O**-formula has been used iff this formula is the propositional content of a **P**-move. Atomic **P**-formulas are used by the very move in which they appear.

2. A complex formula A has been used iff all the possible aggressive and defensive moves related to A have been stated.

Linear dialogues are contextual, the flow of information within the proof is constrained by an explicit structure of *contexts*, ordered by a relation of subordination. The introduction of a new context (which corresponds to a splitting of contexts in the sequent calculus) will always be a consequence of the Opponent's choices. The Proponent will stay in the same context as long as he can.

A sequent $\Gamma \vdash A$ is the statement that from assumptions Γ one can infer conclusion A. From the dialogical point of view, assumptions are the Opponent's conc1essions, while conclusions are the Proponent's claims. Splitting contexts for tensor occurs when it is asserted by the Proponent, so the dialogical particle rule will let the challenger choose the context where the dialogue will proceed. Dually, the multiplicative disjunction par will generate a splitting of contexts when asserted by the Opponent, thus the particle rule will let the defender choose the context. The interpretation of linear implication $\multimap$ remains constant, to attack it, we one must assert the antecedent, hoping that the opponent cannot use it to prove the consequent. Also the interpretation of linear negation stays the same.

Assertion	Attack	Response
$A \otimes B$ in context μ	$\otimes_L$ in subcontext ν of μ	A in ν
$A \otimes B$ in context μ	$\otimes_R$ in subcontext ν' of μ	B in ν'
$A \parr B$ in context μ	$?$ in μ	A in subcontext ν of μ
$A \parr B$ in context μ	$?$ in μ	B in subcontext ν' of μ
$A \multimap B$ in context μ	A in subcontext ν of μ	B in subcontext ν of μ
$\neg A$ in context μ	A in subcontext ν of μ	$-$ in subcontext ν of μ
$A \,\&\, B$	$?\&_L$	A
$A \,\&\, B$	$?\&_R$	B
$A \oplus B$	$?$	A
$A \oplus B$	$?$	B

Rahman also introduces an interpretation for the linear logic exponentials in terms of games, but says that the proof that the games are sound and complete for the exponentials is future work.

It is worth noticing that while Rahman deals exclusively with classical linear logic, he does mention that the intuitionistic structural rule could be used instead. Using the intuitionistic structural rule we should arrive at a semantics for Full Intuitionistic Linear Logic.

Full Intuitionistic Linear Logic

Full Intuitionistic Linear Logic(FILL) is a variant of Linear Logic introduced by de Paiva and Hyland in [8]. FILL is an interesting system, as it has independent multiplicative and additive connectives, just like Intuitionistic Logic has three independent connectives, conjunction, disjunction and implication.

FILL came into existence from work on Dialectica categories, which can be considered a different kind of game, a superpower game where players, instead of playing a proper game, simply face-off their collections of strategies. In one round the superpower game is decided, either Proponent has a winning strategy or Opponent has one or neither has one.

Conclusions

We recalled games for Linear Logic in two traditions, Blass-style games (as discussed by Hyland, Abramsky and Jagadhesan and many others) and Lorenzen-style games (as introduced by Rahman, Keiff and others). It seems clear that Lorenzen games have the ability to model full intuitionistic linear logic easily. Emphasis is given to the interpretation of linear implication, instead of negation.

This note is just a first step in the programme to develop *compositional* versions of Lorenzen Linear logic games. We need still to prove our soundness and completeness result and more importantly we need to make sure that strategies are compositional, to make sure we can produce categories of Lorenzen games.

References

[1] Blass, A. *A Game Semantics for Linear Logic*, Annals of Pure and Applied Logic 56, 183-220, 1992.

[2] *Is game semantics necessary?* In Börger, Gurevich and Meinke, eds. Computer Science Logic, 7th Workshop, 1993, springer-Verlag, 1994.

[3] Blass A. *Some Semantical Aspects of Linear Logic*, Journal of the Interest Group in Pure and Applied Logic 5, 487-503, 1997.

[4] Blass A. *Questions and Answers – a category arising in Linear Logic, Complexity Theory and Set Theory* In Girard, Lafont and Regnier, eds. “Advances in Linear Logic”, vol. 222 of London Mathematical Society Lecture Notes, Cambridge University Press, 1995.

[5] Rahman, S. and Keiff, L. *On how to be a dialogician.* In D. Vanderveken (ed.): Logic, Thought and Action, Dordrecht: Springer, pp. 359-408, 2004.

[6] Keiff, L. *Dialogical Logic.* In Stanford Encyclopedia of Philosophy, 2009.

[7] Rahman, S. *Un desafio para las teorias cognitivas de la competencia logica: los fundamentos pragmaticos de la semantica de la logica linear*, Manuscrito, volume XXV(2), pp. 381-432, October 2002.

[8] Hyland, J. M. E. and de Paiva, V. C.V. *Full Intuitionistic Linear Logic* (extended abstract). Annals of Pure and Applied Logic, 64(3), pp.273-291, 1993.

[9] Felscher, W. *Dialogues as a Foundation for Intuitionistic Logic.* In: Gabbay, D. and Guenthner, F. (eds.), Handbook of Philosophical Logic, Bd. III, Dordrecht, 1985, pp. 341-372.

A The full specification of FILL

$$\begin{array}{lll}
\textit{term\_var},\ x,\ y,\ z & & \\
\textit{index\_var},\ i,\ j,\ k & & \\
\textit{form},\ A,\ B,\ C,\ D,\ E & ::= & \\
 & | & I \\
 & | & \bot \\
 & | & A \multimap B \\
 & | & A \otimes B
\end{array}$$

	$\mid$	$A \parr B$	
	$\mid$	(A)	S
$patterns,\ p$	$::=$		
	$\mid$	$*$	
	$\mid$	x	
	$\mid$	$p_1 \otimes p_2$	
	$\mid$	$p_1 \parr p_2$	
	$\mid$	$-$	
$term,\ t,\ e,\ d,\ f,\ g,\ u$	$::=$		
	$\mid$	x	
	$\mid$	$*$	
	$\mid$	$\circ$	
	$\mid$	$e_1 \otimes e_2$	
	$\mid$	$e_1 \parr e_2$	
	$\mid$	$\lambda x.t$	
	$\mid$	$\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ e$	
	$\mid$	$f\ e$	
	$\mid$	$[t/x]t'$	M
	$\mid$	$[t/x, e/y]t'$	M
	$\mid$	(t)	
	$\mid$	t	
Γ	$::=$		
	$\mid$	$x : A$	
	$\mid$	$\cdot$	
	$\mid$	Γ, Γ'	
$\mathsf{\Delta}$	$::=$		
	$\mid$	$t : A$	
	$\mid$	$\cdot$	
	$\mid$	$\mathsf{\Delta} \mid \mathsf{\Delta}'$	
	$\mid$	$\mathsf{\Delta}$	
$formula$	$::=$		
	$\mid$	$judgement$	
	$\mid$	$formula_1 \quad formula_2$	
	$\mid$	$(formula)$	
	$\mid$	$x \notin \mathsf{FV}(\mathsf{\Delta})$	

$$\begin{array}{lll}
\mathit{InferRules} & ::= & \\
 & | & \Gamma \vdash \Delta \\
 & | & f = e
\end{array}$$

$$\begin{array}{lll}
\mathit{judgement} & ::= & \\
 & | & \mathit{InferRules}
\end{array}$$

$$\begin{array}{lll}
\mathit{user\_syntax} & ::= & \\
 & | & \mathit{term\_var} \\
 & | & \mathit{index\_var} \\
 & | & \mathit{form} \\
 & | & \mathit{patterns} \\
 & | & \mathit{term} \\
 & | & \Gamma \\
 & | & \Delta \\
 & | & \mathit{formula}
\end{array}$$

$\boxed{\Gamma \vdash \Delta}$

$$\frac{}{x : A \vdash x : A}\quad \textsc{Identity}$$

$$\frac{\Gamma \vdash t : A \mid \Delta \quad y : A, \Gamma' \vdash f_i : B_i}{\Gamma, \Gamma' \vdash \Delta \mid [t/y] f_i : B_i}\quad \textsc{Cut}$$

$$\frac{\Gamma \vdash e_i : A_i}{\Gamma, x : I \vdash \mathsf{let}\, x\, \mathsf{be}\, *\, \mathsf{in}\, e_i : A_i}\quad \textsc{Il}$$

$$\frac{}{\cdot \vdash * : I}\quad \textsc{Ir}$$

$$\frac{\Gamma, x : A, y : B \vdash f_i : C_i}{\Gamma, z : A \otimes B \vdash \mathsf{let}\, x\, \mathsf{be}\, x \otimes y\, \mathsf{in}\, f_i : C_i}\quad \textsc{Tl}$$

$$\frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma' \vdash f : B \mid \Delta'}{\Gamma, \Gamma' \vdash e \otimes f : A \otimes B \mid \Delta \mid \Delta'}\quad \textsc{Tr}$$

$$\frac{}{x : \perp \vdash \cdot}\quad \textsc{Pl}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \circ : \perp \mid \Delta}\quad \textsc{Pr}$$

$$\frac{\Gamma, x : A \vdash d_i : C_i \quad \Gamma', y : B \vdash f_j : D_j}{\Gamma, \Gamma', z : A \parr B \vdash \mathsf{let}\, z\, \mathsf{be}\, x \parr - \mathsf{in}\, d_i : C_i \mid \mathsf{let}\, z\, \mathsf{be}\, - \parr y\, \mathsf{in}\, f_j : D_j}\quad \textsc{Parl}$$

$$\frac{\Gamma \vdash \Delta \mid e : A \mid f : B \mid \Delta'}{\Gamma \vdash \Delta \mid e \parr f : A \parr B \mid \Delta'}\quad \textsc{Parr}$$

$$\frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma', x : B \vdash f_i : C_i}{\Gamma, y : A \multimap B, \Gamma' \vdash [y\ e/x] f_i : C_i \mid \Delta}\quad \textsc{Impl}$$

$$\frac{\Gamma, x : A \vdash e : B \mid \Delta \quad x \notin \mathsf{FV}(\Delta)}{\Gamma \vdash \lambda x.e : A \multimap B \mid \Delta}\ \textsc{ImpR}$$

$\boxed{f = e}$

$$\frac{}{(\lambda x.e)\ e' = [e'/x]e}\ \textsc{Eq\_Beta}$$

$$\frac{}{\lambda x.f\ x = f}\ \textsc{Eq\_Eta}$$

$$\frac{}{\mathsf{let}\ *\ \mathsf{be}\ *\ \mathsf{in}\ e = e}\ \textsc{Eq\_I}$$

$$\frac{}{\mathsf{let}\ u\ \mathsf{be}\ *\ \mathsf{in}\ [*/z]f = [u/z]f}\ \textsc{Eq\_Stp}$$

$$\frac{}{\mathsf{let}\ e \otimes t\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ u = [e/x, t/y]u}\ \textsc{Eq\_T1}$$

$$\frac{}{\mathsf{let}\ u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [x \otimes y/z]f = [u/z]f}\ \textsc{Eq\_T2}$$

$$\frac{}{\mathsf{let}\ u \parr t\ \mathsf{be}\ x \parr -\ \mathsf{in}\ e = [u/x]e}\ \textsc{Eq\_P1}$$

$$\frac{}{\mathsf{let}\ u \parr t\ \mathsf{be}\ - \parr y\ \mathsf{in}\ e = [t/y]e}\ \textsc{Eq\_P2}$$

$$\frac{}{(\mathsf{let}\ x\ \mathsf{be}\ x \parr -\ \mathsf{in}\ x) \parr (\mathsf{let}\ u\ \mathsf{be}\ - \parr y\ \mathsf{in}\ y) = u}\ \textsc{Eq\_P3}$$