

Proof-Search and the Logic of Interaction

Alexis Saurin



ASL annual meeting
Special Session on Structural Proof-Theory
02/04/2011, Madison, Wisconsin

Aim of the Presentation

What this talk is about:

- Proof-theoretical foundations of programming languages;
- The central role the *cut rule* plays in the dynamics of proofs;
- Differences and similarities between the logical foundations of *functional programming* and *logic programming*;
- Game-theoretic approach to proofs and computation: proofs as dialogical argumentations;
- Interactive Proof-search: Proof-Search by Cut-Elimination;

The aim of the presentation is:

- to demonstrate the fruitfulness of the proof-theoretical and game-theoretical approaches;
- to challenge the difference between proof-search and proof-normalization;

Organization of the Talk

Outline of the presentation

- *Sequent calculus, proof theory and computation;*
- *Background on linear logic;*
- *Interactive proof search in $MALL^{\times}$;*
- *Abstracting away from sequent proofs: from $MALL^{\times}$ to Ludics;*
- *A uniform framework for computation-as-proof-search.*

Proofs and Programs

Poof-Theory and Programming (1)

The *cut rule*: the corner-stone of mathematical reasoning

In order to establish theorem \mathcal{T} , a typical pattern is:

- first, to find an appropriate lemma \mathcal{L} ;
- second, to prove \mathcal{T} under the assumption that \mathcal{L} holds;
- then, to establish the lemma \mathcal{L} ;
- finally, to deduce theorem \mathcal{T} .

This is reflected in Gentzen's sequent calculi by the *cut* inference rule:

$$\boxed{\frac{\frac{\mathcal{D}_1}{\vdash \mathcal{L}} \quad \frac{\mathcal{D}_2}{\mathcal{L} \vdash \mathcal{T}}}{\vdash \mathcal{T}} \text{ cut}}$$

Proof theory and Programming (2)

The *cut rule*: the corner-stone of the computational interpretation of proofs.

$$\boxed{\frac{\frac{\mathcal{D}_1}{\vdash \mathcal{L}} \quad \frac{\mathcal{D}_2}{\mathcal{L} \vdash \mathcal{T}}{\vdash \mathcal{T}} \text{ cut}}$$

Gentzen's Fundamental Theorem (Hauptsatz):

- ***Cut admissibility:*** One can prove the same statements with or without cuts.
- ***Cut elimination:*** One can transform a proof with cuts into a proof without cuts by an algorithmic process;

⇒ ***Connects Programming languages & Proof theory:***

Functional Programming Proof Normalization (PN) λ -calculus	/	Logic Programming Proof Search (PS) Prolog
--	---	---

Proof Theory and Functional Programming

The well-known *Curry-Howard correspondence*:

Typed λ -calculus	\longleftrightarrow	NJ
Data Types	\longleftrightarrow	Formulas
Typing Judgements	\longleftrightarrow	NJ Sequents
Typed λ -terms	\longleftrightarrow	NJ Deductions
Redex	\longleftrightarrow	Cut
β -reduction	\longleftrightarrow	Cut Reduction
Normal Forms	\longleftrightarrow	Cut-free Proofs

Extends to sequent calculus (Curien-Herbelin)

Proof Theory and Logic Programming

- The program is encoded as a sequent, typically $\mathcal{P} \vdash G$;
- The *operational meaning* of this search lies in constraints that are imposed to the search strategy, for instance a *goal-directed search*. Example:

$$\frac{\mathcal{P}, A \vdash G}{\mathcal{P} \vdash A \Rightarrow G} \text{load/} \Rightarrow$$

Logic Programming	\longleftrightarrow	Sequent calculus
Program	\longleftrightarrow	Sequent
Program Clause	\longleftrightarrow	Formula
Computation	\longleftrightarrow	Proof Search
Successful computation	\longleftrightarrow	Cut-free Proofs

Background on linear logic

Background on Linear Logic

Linear Logic [Girard, 1987]:

- *results from a careful analysis of **structural rules** in sequent calculus;*
- *more connectives than LK (2 conjunctions, 2 disjunctions plus modalities), but the new inference rules are actually derived in a simple way from the usual rules for LK;*
- *built on strong duality principles \Rightarrow one-sided sequents.*

LL and programming:

- PN: more refined datatypes, finer-grained cut-elimination;
- PS: more expressive program clauses.

LL Sequent Calculus

$$F ::= a \mid F \otimes F \mid F \oplus F \mid \mathbf{1} \mid \mathbf{0} \mid !F \\ a^\perp \mid F \wp F \mid F \& F \mid \perp \mid \top \mid ?F$$

$$\frac{}{\vdash a, a^\perp} [ini] \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} [cut]$$

$$\frac{}{\vdash \mathbf{1}} [1] \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} [\otimes] \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} [\wp] \quad \frac{\vdash \Gamma}{\vdash \Gamma, \perp} [\perp]$$

$$\frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus A_2} [\oplus_i] \quad i \in \{1, 2\} \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} [\&] \quad \frac{}{\vdash \Gamma, \top} [\top]$$

$$\frac{\vdash ?\Gamma, B}{\vdash ?\Gamma, !B} [!] \quad \frac{\vdash \Gamma, B}{\vdash \Gamma, ?B} [?d] \quad \frac{\vdash \Gamma}{\vdash \Gamma, ?B} [?w] \quad \frac{\vdash \Gamma, ?B, ?B}{\vdash \Gamma, ?B} [?c]$$

MALL Sequent Calculus

$F ::= a \mid F \otimes F \mid F \oplus F \mid \mathbf{1} \mid \mathbf{0}$ positive
 $a^\perp \mid F \wp F \mid F \& F \mid \perp \mid \top$ negative

$$\frac{}{\vdash a, a^\perp} [ini] \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} [cut]$$

$$\frac{}{\vdash \mathbf{1}} [1] \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} [\otimes] \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} [\wp] \quad \frac{\vdash \Gamma}{\vdash \Gamma, \perp} [\perp]$$

$$\frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus A_2} [\oplus_i] \quad i \in \{1, 2\} \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} [\&] \quad \frac{}{\vdash \Gamma, \top} [\top]$$

Focalization in Linear Logic

- Rules for negative connectives are *reversible*: no choice to make, provability of the conclusion implies provability of the premisses.
- Rules for positive connectives *involve choices*, resulting in possible erroneous choices during proof-search.
- Still, *positive connectives* satisfy *focalization*: in a sequent $\vdash F_0, \dots, F_n$ containing no negative formulas, some formula F_i can be chosen as a *focus* for the search

Γ contains a negative formula	Γ contains no negative formula
choose any negative formula (e.g. the leftmost one) and decompose it using the only possible negative rule	choose some positive formula and decompose it (and its subformulas) hereditarily until we reach atoms or negative subformulas

Focalization in Linear Logic

Synthetic Connectives and Computation

- Focalization precisely describes what clusters of connectives form a *synthetic connective*: arbitrary clusters of connectives of the same polarity: $(A \otimes B) \oplus C$ can be considered as a *ternary connective*: $(_ \otimes _) \oplus _$.
- *Hypersequentialized calculus* by building maximal clusters:
 - Connectives: $\oplus_{i \in I} \otimes_{j \in J_i} N_i^j$ and $\&_{i \in I} \wp_{j \in J_i} P_i^j$
 - Strict *alternation of polarity* (cf *alternation of players in games*.)
 - One of the key ingredients of Ludics, which is built on a *hypersequentialized calculus*.

Focalization and Programming:

- PN: pattern matching constructions;
- PS: structures the search and reduces the size of the search space

Proof Search by Cut-Elimination: Interactive Proof Search

Contrasting Logic and Functional Programming

- *Proof-Search*: the dynamics of the computation comes from the search for a cut-free proof;
- *Cut-Elimination*: the dynamics of the computation lies in the normalization of a proof into a cut-free proof;
- **Cut-admissibility** vs. **Cut-elimination** (*two aspects of the same result*);
- In both cases, results of computations are cut-free proofs;
- Complexity lies in the choice of the cut-formula;
- There seems to be a dichotomy between the two approaches: difficult to relate functional programming and logic programming in a logically satisfying way.

Logic and Logic Programming: a Mismatch?

A fundamental problem: The very objects of proof search are *unfinished/uncompleted proofs* which are not objects of the theory of sequent calculus. For instance, pruning operators prune the “*search space*”. (*search space* \neq *space of proofs*)

For instance:

- How to use a *failed search* for future computations?
- How to use the *past computations* in order to improve the next computations?

We look for a setting which would be both:

- *more uniform*
 - ⇒ the different components of computation as proof search should be grouped in objects of the same sort
- *more flexible*
 - ⇒ it should allow to describe in a more declarative way operations on the search space

Comparing Proof Search and Cut-Elimination

Cut

(*lemma*)

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

Improper Axiom

(*joker*)

$$\overline{\Gamma \vdash \Delta}$$

Comparing Proof Search and Cut-Elimination

$$\begin{array}{l}
 \text{Cut} \quad \quad \quad (\text{lemma}) \quad \quad \quad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta} \\
 \text{Improper Axiom} \quad (\text{joker}) \quad \quad \quad \frac{}{\Gamma \vdash \Delta}
 \end{array}$$

Paradigm	Functional	Logic
Program	Proof without improper axiom but with cuts	Proof without cuts but with improper axioms
Computation	Cut Elimination	Improper Axioms Elimination
Result	Proof <u>without cuts</u> and <u>without improper axiom</u>	

Comparing Proof Search and Cut-Elimination

$$\begin{array}{l}
 \text{Cut} \quad \quad \quad (\text{lemma}) \quad \quad \quad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta} \\
 \text{Improper Axiom} \quad (\text{joker}) \quad \quad \quad \frac{}{\Gamma \vdash \Delta}
 \end{array}$$

Paradigm	Functional	Logic
Program	Proof without improper axiom but with cuts	Proof without cuts but with improper axioms
Computation	Cut Elimination	Improper Axioms Elimination
Result	Proof <u>without cuts</u> and <u>without improper axiom</u>	

Dialogical Games for Proofs and Programs

Games to provide a semantics to proofs and programs:

- Early interactive interpretations of proofs: Gentzen's consistency proof (1936), Gödel Dialectica interpretation (1958), Lorenzen's Dialogue games for intuitionistic provability (1960);
- Modern game-theoretical semantics to give meaning to proofs of LL: Blass games (1992), Hyland and Ong (1993), Abramsky et al's (1994), Girard's Geometry of interaction and lots of recent developments;
- Game semantics for function programming: starting with Hyland-Ond, Abramsky et al's results on fully abstract model for PCF. Many developments capturing other programming constructions. Basic principle of those games: the play describes the interaction between program's strategy and the strategy of its environment.

Games and Logic Programming

Games to model logic programs:

- A very natural approach, first introduced by Van Emden in 1986;
- Yet, much less investigated than the game-semantical approaches to functional programming;
- Basic idea: given a program \mathcal{P} and a goal q , the player believes the goal is a consequence of the program while the opponent doubts it.
- Loddo & di Cosmo were the first to develop van Emden's idea providing a game-semantics for logic programs (Horn clauses) which is sound and complete with respect to SLD-resolutions.
- Later, Pym & Ritter, Miller, Saurin & Delandé, Galanaki, Rondogiannis & Wadge, Tsouanas provided various game semantics for logic programming and proof-search.

Game-theoretical interpretation of logic programs

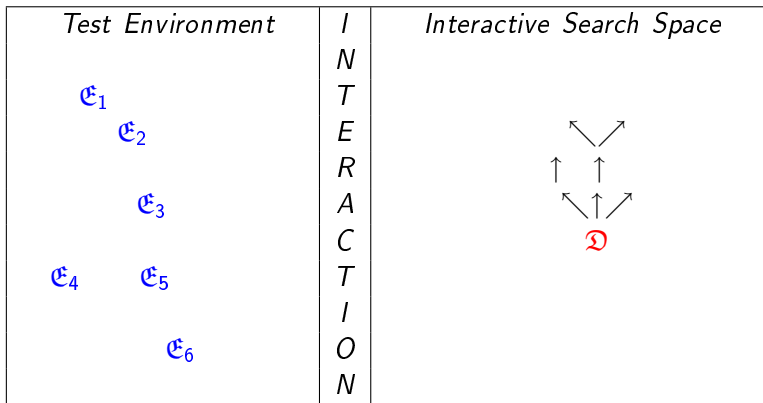
- Given a logic program \mathcal{P} and a goal $q = a_1, \dots, a_n$;
- The player P aims at justifying that q can be deduced from \mathcal{P} while the Opponent O doubts that the goal can be inferred from the program.
- O begins by selecting one of the conjuncts of the goal q , say a_i , challenging P to justify a_i .
- P shall thus select a program clause ($c \in \mathcal{P}$) to support his assertion, such that the head of the clause matches a_i : his argumentation rely on the body of the clause.
- There are three possible situations:
 - at some point, P picks a clause that is a fact: $a \leftarrow$. O is left with an empty conjunct and cannot play. The game is over, P wins.
 - at some point, O picks a body formula that no clause's head matches. P is therefore unable to justify the formula. The game is over, O wins.
 - otherwise, the play goes on forever, in such a case, O wins.

Interactive Proof-Search, in principle

<i>Test Environment</i>	<i>I N T E R A C T I O N</i>	<i>Interactive Search Space</i>
<p data-bbox="251 280 289 317">\mathcal{E}_1</p> <p data-bbox="305 334 344 370">\mathcal{E}_2</p> <p data-bbox="330 436 369 472">\mathcal{E}_3</p> <p data-bbox="209 541 248 578">\mathcal{E}_4</p> <p data-bbox="333 541 371 578">\mathcal{E}_5</p> <p data-bbox="374 643 412 679">\mathcal{E}_6</p>		

Duality test environment/interactive search space.

Interactive Proof-Search, in principle



Duality test environment/interactive search space.

Proof Construction by Consensus: \mathcal{D} must satisfy all tests \mathcal{E}_i :

Either \mathcal{D} has an argumentation to oppose to
(interact with) any \mathcal{E}_i or it shall quit (ie use his joker).

MALL

MALL formulas will be given addresses to distinguish occurrences:

$\mathbf{1}_0 \ \&\langle \rangle \ (\perp_{10} \oplus_1 \perp_{11})$

ξ	$::= \langle \rangle \mid \xi 0 \mid \xi 1$	(locations)
F_ξ	$::= F_{\xi 0} \otimes_\xi F_{\xi 1} \mid F_{\xi 0} \oplus_\xi F_{\xi 1} \mid \mathbf{1}_\xi \mid \mathbf{0}_\xi$	(pos)
	$\mid F_{\xi 0} \wp_\xi F_{\xi 1} \mid F_{\xi 0} \&_\xi F_{\xi 1} \mid \perp_\xi \mid \top_\xi$	(neg)

$$\frac{}{\vdash A_\xi, A_\xi^\perp} \text{ini} \qquad \frac{\vdash \Gamma, A_\xi \quad \vdash \Delta, A_\xi^\perp}{\vdash \Gamma, \Delta} \text{cut}$$

$$\frac{}{\vdash \mathbf{1}_\xi} \mathbf{1} \qquad \frac{\vdash \Gamma, A_{\xi 0} \quad \vdash \Delta, B_{\xi 1}}{\vdash \Gamma, \Delta, A_{\xi 0} \otimes_\xi B_{\xi 1}} \otimes \qquad \frac{\vdash \Gamma, A_{\xi i}}{\vdash \Gamma, A_{\xi 0} \oplus_\xi A_{\xi 1}} \oplus_i, \ i \in \{0, 1\}$$

$$\frac{}{\vdash \Gamma, \top_\xi} \top \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp_\xi} \perp \qquad \frac{\vdash \Gamma, A_{\xi 0}, B_{\xi 1}}{\vdash \Gamma, A_{\xi 0} \wp_\xi B_{\xi 1}} \wp \qquad \frac{\vdash \Gamma, A_{\xi 0} \quad \vdash \Gamma, B_{\xi 1}}{\vdash \Gamma, A_{\xi 0} \&_\xi B_{\xi 1}} \&$$

In $\wp, \&$, Γ contains no negative formula.

MALL \boxtimes and IPS in MALL \boxtimes

MALL formulas will be given addresses to distinguish occurrences:

$\mathbf{1}_0 \ \& \langle \rangle \ (\perp_{10} \oplus_1 \perp_{11})$

ξ	::=	$\langle \rangle \mid \xi_0 \mid \xi_1$	(locations)
F_ξ	::=	$F_{\xi_0} \otimes_\xi F_{\xi_1} \mid F_{\xi_0} \oplus_\xi F_{\xi_1} \mid \mathbf{1}_\xi \mid \mathbf{0}_\xi$	(pos)
		$F_{\xi_0} \wp_\xi F_{\xi_1} \mid F_{\xi_0} \&_\xi F_{\xi_1} \mid \perp_\xi \mid \top_\xi$	(neg)

$$\frac{}{\vdash A_\xi, A_\xi^\perp} \text{ini} \qquad \frac{\vdash \Gamma, A_\xi \quad \vdash \Delta, A_\xi^\perp}{\vdash \Gamma, \Delta} \text{cut}$$

$$\boxed{\vdash \Gamma} \boxtimes$$

$$\frac{}{\vdash \mathbf{1}_\xi} \mathbf{1} \qquad \frac{\vdash \Gamma, A_{\xi_0} \quad \vdash \Delta, B_{\xi_1}}{\vdash \Gamma, \Delta, A_{\xi_0} \otimes_\xi B_{\xi_1}} \otimes \qquad \frac{\vdash \Gamma, A_{\xi_i}}{\vdash \Gamma, A_{\xi_0} \oplus_\xi A_{\xi_1}} \oplus_i, \ i \in \{0, 1\}$$

$$\frac{}{\vdash \Gamma, \top_\xi} \top \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp_\xi} \perp \qquad \frac{\vdash \Gamma, A_{\xi_0}, B_{\xi_1}}{\vdash \Gamma, A_{\xi_0} \wp_\xi B_{\xi_1}} \wp \qquad \frac{\vdash \Gamma, A_{\xi_0} \quad \vdash \Gamma, B_{\xi_1}}{\vdash \Gamma, A_{\xi_0} \&_\xi B_{\xi_1}} \&$$

In \boxtimes , Γ contains no negative formula.

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash} \quad \perp}{\vdash \perp_{10}} \quad \oplus_1 \quad \perp_{11}}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \ \&\langle \rangle \ (\perp_{10} \oplus_1 \perp_{11})} \ \& \ \oplus_0$$

$$\mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash} \quad \perp}{\vdash \perp_{11}} \quad \oplus_1 \quad \perp_{10}}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \ \&\langle \rangle \ (\perp_{10} \oplus_1 \perp_{11})} \ \& \ \oplus_1$$

IPS in MALL \boxtimes

$$\mathfrak{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10}} \quad \boxtimes}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_0 \quad \&$$

$$\mathfrak{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{11}} \quad \boxtimes}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_1 \quad \&$$

Used to build \mathfrak{D} by interaction ($i \in \{0, 1\}$):

$$\frac{\frac{\overline{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \quad \mathfrak{D}_i}{\vdash} \quad \frac{\overline{\vdash \perp_0 \oplus_{\langle \rangle} (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \quad \mathfrak{D}}{\vdash} \quad \text{cut}$$

$\downarrow \text{cut-elim}$

$$\frac{}{\vdash} \quad \boxtimes$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \perp_{10} \oplus_1 \perp_{11}} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \oplus_0}{\&} \quad \mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \perp_{10} \oplus_1 \perp_{11}} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}{\&}$$

$$\mathcal{D} = \overline{\vdash \perp_0 \oplus \langle \rangle (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})}$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \perp_{10} \oplus_1 \perp_{11}} \quad \frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}} \oplus_0}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \&$$

$$\mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \perp_{10} \oplus_1 \perp_{11}} \quad \frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}} \oplus_1}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \&$$

$$\mathcal{D} = \frac{}{\vdash \perp_0 \oplus \langle \rangle (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \oplus_0}{\&} \quad \mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}{\&}$$

$$\mathcal{D} = \frac{\overline{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}}}{\vdash \perp_0 \oplus \langle \rangle (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_0}{\&}$$

$$\mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}{\&}$$

$$\mathcal{D} = \frac{\frac{\overline{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}}}{\vdash \perp_0 \oplus_{\langle \rangle} (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1}{\&}$$

$$\& = \&_0 \cup \&_1$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_0}{\&} \quad \mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}{\&}$$

$$\mathcal{D} = \frac{\frac{\overline{\vdash \mathbf{1}_{10}} \quad \overline{\vdash \mathbf{1}_{11}}}{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}} \&}{\vdash \perp_0 \oplus_{\langle \rangle} (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_0}{\&} \quad \mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}{\&}$$

$$\mathcal{D} = \frac{\frac{\frac{\overline{\vdash \mathbf{1}_{10}} \quad \mathbf{1}}{\vdash \mathbf{1}_{10}} \quad \frac{\overline{\vdash \mathbf{1}_{11}} \quad \mathbf{1}}{\vdash \mathbf{1}_{11}}}{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}} \&}{\vdash \perp_0 \oplus_{\langle \rangle} (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1}$$

IPS in MALL \boxtimes

$$\mathcal{D}_0 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{10}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_0}{\&} \quad \mathcal{D}_1 = \frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \mathbf{1}_0} \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{11}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}}}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}{\&}$$

$$\mathcal{D} = \frac{\frac{\frac{\overline{\vdash \mathbf{1}_{10}} \quad \mathbf{1} \quad \overline{\vdash \mathbf{1}_{11}} \quad \mathbf{1}}{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}} \quad \&}{\vdash \perp_0 \oplus_{\langle \rangle} (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1}{\&}}$$

IPS in MALL \boxtimes

Interacting with

$$\mathcal{D}_i = \frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1} \quad \boxed{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{1i}} \quad \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}} \quad \oplus_i}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \quad \&, i \in \{0, 1\}$$

one can get

$$\mathcal{D} = \frac{\overline{\vdash \mathbf{1}_{10}} \quad \mathbf{1} \quad \overline{\vdash \mathbf{1}_{11}} \quad \mathbf{1}}{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}} \quad \&}{\vdash \perp_0 \oplus \langle \rangle (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \quad \oplus_1$$

IPS in MALL \boxtimes

Interacting with

$$\mathfrak{D}_i = \frac{\boxed{\frac{\overline{\vdash \mathbf{1}_0} \mathbf{1}}{\vdash \mathbf{1}_0}} \quad \frac{\frac{\overline{\vdash} \boxtimes}{\vdash \perp_{1i}} \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}} \oplus_i}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \&, i \in \{0, 1\}}$$

one can get

or

$$\mathfrak{D} = \frac{\frac{\overline{\vdash \mathbf{1}_{10}} \mathbf{1} \quad \overline{\vdash \mathbf{1}_{11}} \mathbf{1}}{\vdash \mathbf{1}_{10} \&_1 \mathbf{1}_{11}} \&}{\vdash \perp_0 \oplus \langle \rangle (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_1 \quad \mathfrak{D}' = \frac{\frac{\overline{\vdash} \boxtimes}{\vdash \perp_0} \perp}{\vdash \perp_0 \oplus \langle \rangle (\mathbf{1}_{10} \&_1 \mathbf{1}_{11})} \oplus_0$$

But \mathfrak{D}' uses \boxtimes : it is a failure.

How to avoid this second interaction for \mathfrak{D}' ?

MALL \boxtimes and IPS in MALL \boxtimes

One could add new tests to the environment:

$$\mathcal{D}_2 = \frac{\overline{\vdash \perp_{10} \oplus_1 \perp_{11}} \boxtimes}{\vdash \mathbf{1}_0 \& \langle \rangle (\perp_{10} \oplus_1 \perp_{11})} \&|_1$$

Adding \mathcal{D}_2 would have **forbidden** the search that leads to a failure by forcing the selection of \oplus_1 .

Slice of the $\&$ rule.

From MALL \boxtimes to designs

Ludics contains appropriate ingredients to represent MALL \boxtimes proofs.

$$\mathfrak{D}_2 = \frac{\overline{\vdash \perp_{10} \oplus_1 \perp_{11}} \boxtimes}{\vdash \mathbf{1}_0 \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \&_{|1}$$

The partial inference rule $\&_{|1}$ is a **negative** rule with **active formula** indexed by $\langle \rangle$ producing one **subformula** located in 1.

This can be summarized in $(\langle \rangle, 1)^-$:

$$\frac{\boxtimes}{(\langle \rangle, 1)^-}$$

From MALL✚ to designs

To obtain designs, several steps away from MALL✚:

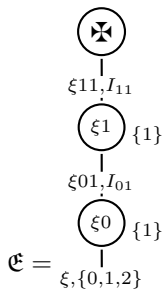
- Retain only the locations, not the formulas
- Only the essence of proof remains: proof rules not the sequents. They are **actions**: $(\xi, l)^-$, $(\tau i, \{1, 2, 3\})^+$, $(\xi 2, \emptyset)^-$
- The actions correspond to maximal clusters of rules of a single polarity to ensure **alternation of polarity**.
- **Daimon** remains: special action ✚
- Work by **slices**: negative rules are collections of **unary rules**.
- **Designs**: correspond to proofs (or strategies). Polarized according to their last rule. May have logical mistakes to be detected interactively (a design can be losing or winning).
- Successful interactions (ie. cut-eliminations) give rise to an **orthogonality relation** between designs of opposite polarity.
- A **Behaviour** is a set of design which is closed by biorthogonality.

From MALL \boxtimes to Designs

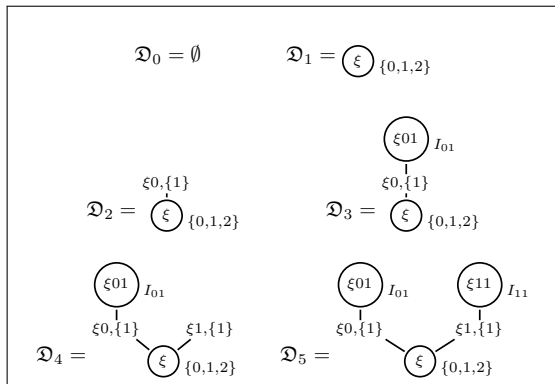
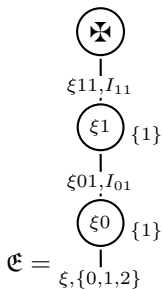
$$\mathfrak{D}_i = \frac{\frac{\frac{\overline{\vdash \mathbf{1}_0} \quad \mathbf{1}}{\vdash \perp_{10}} \oplus_1 \quad \frac{\frac{\overline{\vdash} \quad \boxtimes}{\vdash \perp_{1i}} \perp}{\vdash \perp_{10} \oplus_1 \perp_{11}} \oplus_i}{\vdash \perp_{10} \&_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \&}{\vdash \perp_0 \oplus_{\langle \rangle} (\perp_{10} \oplus_1 \perp_{11})} \oplus_1}, i \in \{0, 1\} \longrightarrow$$

$$\mathfrak{D} = \frac{\frac{\frac{\overline{\vdash \mathbf{1}_{10}} \quad \mathbf{1}}{\vdash \perp_{10}} \oplus_1 \quad \frac{\overline{\vdash \mathbf{1}_{11}} \quad \mathbf{1}}{\vdash \perp_{11}} \oplus_1}{\vdash \perp_{10} \&_1 \perp_{11}} \&}{\vdash \perp_0 \oplus_{\langle \rangle} (\perp_{10} \&_1 \perp_{11})} \oplus_1} \longrightarrow$$

Scheme of IPS on an Example



Scheme of IPS on an Example



Updating the test environment

- The environment is a set of tests isomorphic to the objects we search for;
 - We develop the searched design depending on those tests;
 - By symmetry, we might also consider the possibility to modify the test environment depending on the tests;
- ⇒ We may for instance add new elements to the test environment.

A dynamical change of the “program” to be compared to the ($\Rightarrow R$) which implemented the loading of a module.

Changes the search behaviour (both in the intuitive and formal sense):

$$\text{if } \mathbf{E} \subseteq \mathbf{E}' \text{ then } \mathbf{E}'^\perp \subseteq \mathbf{E}^\perp.$$

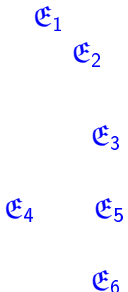
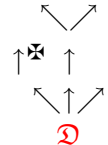
The behaviour in which the search is taking place gets more and more precise... there are more tests to constrain the search.

Backtracking

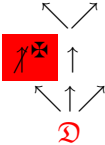
The backtracking behaviour corresponds to undoing the last choice and going for an alternative.

- When an interactive search has finished and the searched design uses a ✘, one may wish to search again.
- We shall allow enlarging the test environment \mathbf{E}_{ENV} with new designs that will contribute to guide the search. Those designs shall be such that the previously computed design is not orthogonal to these designs: add restricted ***negative daimons*** which forbid exploring the branches that previously led to failures.

Backtracking, Interactively

<i>Test Environment</i>		<i>Interactive Search Space</i>
	I N T E R A C T I O N	

Backtracking, Interactively

<i>Test Environment</i>		<i>Interactive Search Space</i>
\mathcal{E}_1 \mathcal{E}_2 \mathcal{E}_3 \mathcal{E}_4 \mathcal{E}_5 \mathcal{E}_6 $\text{Backtrack}(\mathcal{D})$	I N T E R A C T I O N	

Using the interaction paths for \mathcal{D}

Conclusions

- Relate cut-elimination and proof-search: proof-search by cut-elimination. A test environment guiding the search of an unspecified proof \mathcal{D} ;
- Possible to enrich the search environment by adding new tests;
- Uniform approach to proof-search: the test environment gather usual logical search instructions as well as constraints to the search strategy;
- Investigate a general framework where FP and LP could be related? \mathcal{D} could be partially specified, interact with the test environment \mathbf{E} and then be extended as a proof under construction (for instance, add new slices to \mathcal{D} , or change \boxtimes to proper positive actions $(\xi, I)^+ \dots$)

Backtracking, Interactively

Definition ($\mathfrak{Test}(p)$)

- $\mathfrak{Test}(\epsilon) = \emptyset$;
- $\mathfrak{Test}(\kappa) = \{\kappa^+\}$;
- $\mathfrak{Test}(s \cdot \kappa \cdot \kappa') = \{\ulcorner s \cdot \kappa \cdot \kappa' \urcorner^+, \ulcorner s \cdot \kappa \urcorner^-\} \cup \mathfrak{Test}(s)$.

Definition ($\mathfrak{Backtrack}(p)$)

$\mathfrak{Backtrack}(p)$ is the smallest design such that:

- $\mathfrak{Backtrack}(p)$ contains all positive chronicles of $\mathfrak{Test}(p)$ except $\ulcorner p \urcorner^+$;
- if $\chi \in \mathfrak{Backtrack}(p)$ is a positive chronicle ending with a proper action $(\xi, I)^+$, then for any $i \in I$ and $J \in \mathcal{P}_f(\omega)$ such that $\chi \cdot (\xi i, J)^- \notin \mathfrak{Test}(p)$, then $\chi \cdot (\xi i, J)^- \cdot \boxtimes \in \mathfrak{Backtrack}(p)$.

Control Operators in Logic Programming

Two important uses of control in logic programming:

- Speed up the search, save resources (*efficiency*)
- Find new solutions, unreachable in finite time (*expressiveness*)

Control Operators in Logic Programming

Two important uses of control in logic programming:

- Speed up the search, save resources *(efficiency)*
pruning a finite part of the search space
- Find new solutions, unreachable in finite time *(expressiveness)*
pruning an infinite part of the search space

Interactive Control

Control treated interactively by updating the test environment \mathbf{E} thanks to ϕ : Given a test environment \mathbf{E} , a design \mathcal{D} and an interaction path $p_{\mathcal{D}}$, ϕ outputs a set of designs $\{\mathfrak{E}_i^{\phi, \mathcal{D}, p_{\mathcal{D}}}, i \in I\}$ used to enlarge \mathbf{E} :

$$\mathbf{E} \mapsto \mathbf{E} \cup \{\mathfrak{E}_i^{\phi, \mathcal{D}, p_{\mathcal{D}}}, i \in I\}$$

Given $\mathbf{E} = \{\mathfrak{E}_i, i \in I\}$ and successive answers $\langle \mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_n, \dots \rangle$
Applying ϕ after obtaining \mathcal{D}_0 , one may consider the set

$$J = \{i > 0, \mathcal{D}_i \in (\mathbf{E} \cup \{\mathfrak{E}_i^{\phi, \mathcal{D}_0, p_{\mathcal{D}_0}}, i \in I\})^\perp\}$$

- if J is infinite, one skipped some of the searched results
- if J is finite, that will possibly allow to find new solutions (that were not reachable at first).

Use behavioural theory to analyze those control mechanisms

Interactive Control

Consider other usual pruning mechanisms in Prolog and see how “interactive” they can be made:

- `!/0;`
 - `a :- b, c, !, d.`
 - `a :- e, !, f.`
 - `a :- g.`
- soft cut;
- once;
- other backtracking modifiers
- intelligent backtracking
- ...

