A game semantics for proof search: Preliminary results

Dale Miller & Alexis Saurin
INRIA-Futurs & LIX
16 July 2005

Outline

1. A neutral approach to proof and refutation.
2. The noetherian Horn clause case.
3. Games for simple expressions.
4. Games for non-simple expressions.
5. Additive games and truth
6. Games for recursion.
Review: Horn clauses

The syntactic variable \( A \) denotes \textit{atomic formulas}: that is, a formula with a predicate (a non-logical constant) as its head: the formulas \( \bot \) and \( \top \) and \( t = s \) are \textit{not} atomic formulas.

A \textit{Horn goal} \( G \) is any formula generated by the grammar:

\[
G ::= \top | \bot | t = s | A | G \land G | G \lor G | \exists x \ G.
\]

A \textit{Horn clause for} the predicate \( p \) is a formula

\[
\forall x_1 \ldots \forall x_n[p(x_1, \ldots, x_n) \equiv G]
\]

where \( n \geq 0 \), \( p \) is an \( n \)-ary predicate symbol, and the \( G \), the \textit{body}, is a Horn goal formula whose free variables are in \( \{x_1, \ldots, x_n\} \).

A \textit{Horn program} is a finite set \( \mathcal{P} \) of Horn clauses all for distinct predicates.
**Review: noetherian Horn clauses**

Define \( q \prec p \) to hold for two predicates if \( q \) appears in the body of the Horn clause for \( p \).

\( \mathcal{P} \) is *noetherian* if the transitive closure of \( \prec \) is acyclic.

When \( \mathcal{P} \) is noetherian, it can be rewritten to a logically equivalent logic program \( \mathcal{P}' \) for which the relation \( \prec \) is empty: that is, there are no atomic formulas in the body of clauses in \( \mathcal{P}' \).

Repeatedly replace \( \prec \)-minimal predicates by their equivalent body.

Thus: in noetherian programs, atoms are not necessary.
Prolog and noetherian Horn clauses

Assume that the noetherian Horn clause program $\mathcal{P}$ is loaded into Prolog and we ask the query

?- G.

Prolog will respond by either reporting yes or no.

If yes then Prolog has a proof of $G$. Such a proof can be represented in “usual” sequent calculus (say, of, Gentzen).

If no then there is a proof of $\neg G$ in proof systems extended to deal with the closed world assumption: Clark’s completion or more recent work on definitions and fixed points in proof theory (Schroeder-Heister & Hallnäs, Girard, and McDowell & Miller &Tiu).
Proof and refutation in one computation

This description of Prolog is a challenge to the conventional understanding of logic-as-proof-search paradigm (Miller, et.al., in late 1980’s).

Prolog did one computation which yielded a proof of $G$ or a refutation of $G$ (i.e., a proof of $\neg G$).

Proof search states that you must select first what you plan to prove and then proceed to prove that: i.e.,

\[ \text{start with either } \rightarrow G \text{ or with } \rightarrow \neg G. \]

How can we formalize this neutral approach?

Can this behavior of Prolog be extended to richer logics?
A neutral approach to proof and refutation

Since a “neutral computation” could yield a proof of either \( G_1 \land G_2 \)
or \( \neg G_1 \lor \neg G_2 \); or either \( \exists x.G \) or \( \forall x.\neg G \), we chose to compute with
a new language of neutral expressions.

\[
N ::= 1 \mid N \times N \mid 0 \mid N + N \mid pt_1 \ldots t_n \mid Q x N
\]

Here, 1 and 0 are the units of \( \times \) and \(+\), respectively.
The expression \( pt_1 \ldots t_n \) is will correspond to the literal \( pt_1 \ldots t_n \)
or \( \neg pt_1 \ldots t_n \).
The variable \( x \) in the expression \( Q x . N \) is bound in the usual sense.
First-order models, briefly

Let $\mathcal{M}$ be a first-order model in the usual sense.

- $|\mathcal{M}|$ denotes the domain of quantification of the model.
- For every $c \in |\mathcal{M}|$ there is a parameter $\bar{c}$ in the language of the logic.
- An atomic formula $p(t_1, \ldots, t_n)$ is true if the $n$-tuple $\langle t_1, \ldots, t_n \rangle \in \mathcal{P}$.

Herbrand Models

Given a signature $\Sigma$, the model $\mathcal{H}_\Sigma$ is such that $|\mathcal{H}_\Sigma|$ is the set of closed terms built from $\Sigma$ and in which the sole predicate that is interpreted is equality: $\mathcal{H}_\Sigma \models t = s$ if and only if $t$ and $s$ are identical closed terms.
Rewriting neutral expressions

Given a model $\mathcal{M}$ we describe a nondeterministic rewriting of multisets of neutral expressions.

\[
\begin{align*}
1, \Gamma & \mapsto \Gamma & N \times M, \Gamma & \mapsto N, M, \Gamma \\
N + M, \Gamma & \mapsto N, \Gamma & N + M, \Gamma & \mapsto M, \Gamma \\
p(t_1, \ldots, t_n), \Gamma & \mapsto \Gamma, & \text{if } \mathcal{M} \models p(t_1, \ldots, t_n) \\
Qx.N, \Gamma & \mapsto N[t/x], \Gamma, & \text{where } t \in |\mathcal{M}|
\end{align*}
\]

Let $\mapsto^*$ be the reflective and transitive closure of $\mapsto$.

Since expressions simplify, rewriting always terminates. Since the domain of quantification is infinite (all terms), rewriting can also be infinitely branching.

**Main question:** Given $N$, does $N \mapsto^* \{\}$?
Main proposition for Horn clauses over $\mathcal{H}_\Sigma$

**Proposition.** Let $N$ be a neutral expression. If $N \mapsto^* \emptyset$ then $\vdash [N]^+$. If $N$ cannot be rewritten to $\emptyset$ then $\vdash [N]^-$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$[N]^+$</th>
<th>$[N]^-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>0</td>
<td>$\top$</td>
</tr>
<tr>
<td>$1$</td>
<td>1</td>
<td>$\bot$</td>
</tr>
<tr>
<td>$t = s$</td>
<td>$t = s$</td>
<td>$\neg(t = s)$</td>
</tr>
<tr>
<td>$N_1 + N_2$</td>
<td>$[N_1]^+ \oplus [N_2]^+$</td>
<td>$[N_1]^- &amp; [N_2]^-$</td>
</tr>
<tr>
<td>$N_1 \times N_2$</td>
<td>$[N_1]^+ \otimes [N_2]^+$</td>
<td>$[N_1]^- \not\oplus [N_2]^-$</td>
</tr>
<tr>
<td>$Qx.N$</td>
<td>$\exists x.[N]^+$</td>
<td>$\forall x.[N]^-$</td>
</tr>
</tbody>
</table>

The range of $[\cdot]^+$ is a familiar linearization of *Horn goal* formulas. The range of $[\cdot]^-$ is their negation.
Treatment of Equality

\[ \vdash t = t \]  \[ \vdash \Delta \theta \]  \[ \vdash \neg(t = s), \Delta \] ^†

The proviso † requires that \( t \) and \( s \) are unifiable and \( \theta \) is their most general unifier (\( \Delta \theta \) is the multiset resulting from applying \( \theta \) to all formulas in \( \Delta \)).

The proviso ‡ requires that \( t \) and \( s \) are not unifiable.

The free variables of a sequent are also called *eigenvariables*, which are introduced by the usual rule for \( \forall R \).
Extending this neutral approach

Can we extend this neutral approach to proof and refutation beyond simple Horn goal formulas?

Proof search alternates between two phases.

- **asynchronous** phase where all inference rules are invertible. No choices need to be made.
- **synchronous** phase where inference rules require choices. A path through a proof must be made.

These two phases arise from dual aspects of the same logical connective.

So far, we only have one phase, with no alternation possible.

- **asynchronous** phase: all paths starting at $N$ do not end in $\{\}$.
- **synchronous** phase: there is a path $N \mapsto^* \{\}$. 
Adding the switch operator

Now add the switch operator to the language of neutral expressions.

\[ N ::= \ldots | \downarrow N. \]

Rewriting leaves switched expressions untouched.

Main question: Given \( N \), does

\[ N \rightarrow^* \{ \downarrow N_1, \ldots, \downarrow N_m \} = \downarrow \{ N_1, \ldots, N_m \} ? \]

The motivation here:

1. One player starts with her instructions \( N \).
2. She works on \( N \) in order to finish her “work”, if possible.
3. If she finishes successfully, she gives to the other player \( m \) instructions \( N_1, \ldots, N_m \).

A class of simple expressions can be defined for which \( m \leq 1 \).
Games: Arenas, strategies, winning strategies

The pair $\langle P, \rho \rangle$ is an arena: $P$ is a set of positions and $\rho$ be a binary relationship on $P$ that describes moves.

A play is a sequence $P_1.P_2.\cdots.P_n$ of $\rho$-related moves.

If $\sigma$ is a set of plays then the set $\sigma/N = \{ S \mid N.S \in \sigma \}$.

A $\forall\exists$-strategy for $N$ is a prefixed closed set $\sigma$ of plays such that $N \in \sigma$ and for all $M$ such that $N \rho M$, the set $\sigma/N$ is a $\exists\forall$-strategy for $M$.

A $\exists\forall$-strategy for $N$ is a prefixed closed set $\sigma$ of plays such that $N \in \sigma$ and for at most one position $M$ such that $N \rho M$, the set $\sigma/N$ is a $\forall\exists$-strategy for $M$.

A winning $\forall\exists$-strategy is a $\forall\exists$-strategy such that all its maximal sequences are of odd length. A winning $\exists\forall$-strategy $\sigma$ is a $\forall\exists$-strategy such that all maximal sequences are of even length.
Games for simple expressions

Define \([\downarrow N]^- = [N]^+\) and \([\downarrow N]^+ = [N]^-\).

Let \(P\) be the set of neutral expressions. The move relation is defined as: \(N \not\rho \emptyset\) if \(N \leftrightarrow^* \{\}\) and \(N \rho M\) if \(N \leftrightarrow^* \{\downarrow M\}\).

Conjecture. Let \(N\) be a simple expression.
There is a winning \(\forall \exists\)-strategy for \(N\) if and only if \(\vdash [N]^-.\)
There is a winning \(\exists \forall\)-strategy for \(N\) if and only if \(\vdash [N]^+.\)

We have a number of examples supporting this Conjecture.
The Conjecture holds in the proposition case (when the model \(M\) is not relevant).
**Example: finite sets**

Encode 0, 1, 2, ... as terms $z, s(z), s(s(z)), \ldots$.

Let finite set $A = \{n_1, \ldots, n_k\}$ of natural numbers can be encoded as $A(x) = x = n_1 + \cdots + x = n_k$.

The expression $A(n)$ has a winning $\exists\forall$-strategy if and only if $n \in A$. In that case, $(n = n_1) \oplus \cdots \oplus (n = n_k)$ is provable.

The expression $A(n)$ has a winning $\forall\exists$-strategy if and only if $n \notin A$. In that case, $\neg(n = n_1) \land \cdots \land \neg(n = n_k)$ is provable.

If $A(x)$ and $B(x)$ encode two finite sets $A$ and $B$, then the expressions $A(x) + B(x)$ and $A(x) \times B(x)$ encode in the intersection and union, respectively, of $A$ and $B$. 
Example: subset

The expression $Q_x.(A(x) \times \uparrow B(x))$ encodes $A \subseteq B$.

Let $P$ be the set $\{0, 2\}$ and let $Q$ be the set $\{0, 1, 2\}$. The expression labeled $P \subseteq Q$, namely,

$$Q_x.([(x \neq 0) + (x \neq 2)] \times \uparrow [(x \neq 0) + (x \neq 1) + (x \neq 2)])$$

has a winning $\forall \exists$-strategy. Thus the following are provable.

$$\forall x.([\neg (x = 0) \& \neg (x = 2)] \$ (x = 0) \oplus (x = 1) \lor (x = 2))).$$

$$\forall x.([(x = 0) \oplus (x = 2)] \rightarrow [(x = 0) \oplus (x = 1) \lor (x = 2)]).$$

The expression labeled $Q \subseteq P$, namely,

$$Q_x.([(x \neq 0) + (x \neq 1) + (x \neq 2)] \times \uparrow [(x \neq 0) + (x \neq 2)])$$

has a winning $\exists \forall$-strategy. Thus the following is provable:

$$\exists x.([(x = 0) \oplus (x = 1) \oplus (x = 2)] \otimes [\neg (x = 0) \& \neg (x = 2)]).$$
Games for non-simple expressions

We do not know yet how to define games for general expressions. Nor do we have any “computer science motivated” examples that indicate the need for non-simple expressions.

It is clear that such games cannot be determinate: that is, not all games will have either a winning $\forall\exists$-strategy or a winning $\exists\forall$-strategy.

For example, $\uparrow \top \times \uparrow \top$ should yield a game with stuck states since neither $\top \otimes \top$ nor $\bot \otimes \bot$ are provable.
Additive Games and Truth

Hintikka showed that games can characterize truth in first-order logic.

Two players $P$ and $O$ play on the same formula:

- if that formula is a conjunction, then player $P$ would choose one of the conjuncts;
- if is a universal quantifier, then player $P$ would pick an instance;
- if the formulas is a disjunction, then player $O$ picks a disjunct; and
- if the formula is an existential quantifier, play $O$ picks an instance.

In our setting, such a game is purely additive: that is, the neutral expressions for such games contain no occurrences of $\times$ and $1$. 
## Additive Games and Truth

Define two mappings, $f(\cdot)$ and $h(\cdot)$, from classical formulas in negation normal form (formulas where negations have only atomic scope) into additive neutral expressions.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Mapping $f$</th>
<th>Mapping $h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B \land C$</td>
<td>$f(B \land C) = f(B) + f(C)$</td>
<td>$h(B \land C) = \uparrow f(B \land C)$</td>
</tr>
<tr>
<td>$B \lor C$</td>
<td>$f(B \lor C) = \uparrow h(B \lor C)$</td>
<td>$h(B \lor C) = h(B) + h(C)$</td>
</tr>
<tr>
<td>$\top$</td>
<td>$f(\top) = 0$</td>
<td>$h(\top) = \uparrow f(\top)$</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$f(\bot) = \uparrow h(\bot)$</td>
<td>$h(\bot) = 0$</td>
</tr>
<tr>
<td>$\forall x.B$</td>
<td>$f(\forall x.B) = Q_x.f(B)$</td>
<td>$h(\forall x.B) = \uparrow f(\forall x.B)$</td>
</tr>
<tr>
<td>$\exists x.B$</td>
<td>$f(\exists x.B) = \uparrow h(\exists x.B)$</td>
<td>$h(\exists x.B) = Q_x.h(B)$</td>
</tr>
<tr>
<td>$\neg(p(t_1, \ldots, t_n))$</td>
<td>$f(\neg(p(t_1, \ldots, t_n))) = \dot{p}(t_1, \ldots, t_n)$</td>
<td>$h(\neg A) = \uparrow f(A)$</td>
</tr>
<tr>
<td>$A$</td>
<td>$f(A) = \uparrow h(A)$</td>
<td>$h(p(t_1, \ldots, t_n)) = \dot{p}(t_1, \ldots, t_n)$</td>
</tr>
</tbody>
</table>
Correctness of additive games with validity

**Proposition.** Let $\mathcal{M}$ be a model and let $f(E) = N$, where $E$ is a closed first-order formula. The formula $E$ is true in $\mathcal{M}$ if and only if there is a $\forall\exists$-win for $N$.

**Proof.** By simple induction over the structure of formulas.
Extending for recursion

Extend expressions with the fixed point constructors \( \{\text{fix}_n\}_{n \geq 0} \). In

\[
(f\text{ix}_n \lambda P \lambda x_1 \ldots \lambda x_n. M)
\]

the bound variable \( P \) is an \( n \)-ary recursive function. Extend \( \mapsto \):

\[
(f\text{ix}_n F t_1 \ldots t_n), \Gamma \mapsto (F(f\text{ix}_n F)t_1 \ldots t_n), \Gamma,
\]

Extend the notions of winning strategies to infinite plays.

An infinite play is a lose for in a \( \exists \forall \)-strategy while it is win for an \( \forall \exists \)-strategy.

The positive translation of fix is the least fixed point operation \( \mu \); negative translation of fix is the greatest fixed point operation \( \nu \).
**Example: less-than-or-equal**

The logic program

\[
\text{leq}(z, N).
\text{leq}(s(P), s(Q)) :- \text{leq}(P, Q).
\]

can be written rather directly (using the Clark completion) as the expression

\[
(\text{fix}_{2} \lambda \text{leq} \lambda n \lambda m [(n \doteq z) + Q_p Q_q (n \doteq s(p) \times m \doteq s(q) \times \text{leq}(p, q))])
\]

This expression, named $L$, has no $\uparrow$ operator (it is just a Horn clause program).

$L(n, m)$ has a winning $\exists \forall$-strategy if and only if $n \leq m$.

$L(n, m)$ has a winning $\forall \exists$-strategy if and only if $n > m$. 
Example: maximum

We can now define the maximum of a set of numbers. Let $A$ be a non-empty set of numbers and let $A(n)$ be the expression encoding this set.

Let $\text{max}A(n)$ be the following expression:

$$A(n) \times \uparrow Q m(A(m) \times \uparrow L(m, n))$$

The expression $\text{max}A(n)$ as a winning $\forall \exists$-strategy if and only if $n$ is not in $A$ or it is not the largest member of $A$. Similarly, $\text{max}A(n)$ as a winning $\exists \forall$-strategy if and only if $n$ is the largest member of $A$. 
Example: bisimulation

Let $\delta \subseteq S \times \Lambda \times S$ be a finite transition on states $S$ and labels $\Lambda$. Encoded this as the expression $\delta(x, y, z)$ given by

$$\sum_{(p,a,q) \in \delta} (x \doteq p \times y \doteq a \times z \doteq q).$$

Bisimulation between two states can be defined using the following recursive expression

$$(\text{fix}_2 \lambda \text{bisim} \lambda p \lambda q. \ [QaQp'.\delta(p, a, p') \times \downarrow Qq'((\delta(q, a, q') \times \downarrow \text{bisim}(p', q'))))]$$

$$+ [QaQq'.\delta(q, a, q') \times \downarrow Qp'((\delta(p, a, p') \times \downarrow \text{bisim}(p', q')))])$$

If $\text{Bisim}$ names the above expression and if $p$ and $q$ are two states (members of $S$), then the game for the expression $\text{Bisim}(p, q)$ is exactly the game usually used to describe bisimulation, eg., by C. Sterling.
Conclusions and Questions

• We have described a neutral approach to proof and refutation for an interesting and useful subset of logic (from the computer science point-of-view).

• Games and winning strategies provide a new way to look at proofs. This is not an approach to “full abstraction” for sequent proofs. We are hopeful for better “proof objects” than those.

• What is really going on with the multiplicatives?

• Can we extend this development to the modals (!, ?) of linear logic? To higher-order quantification?

• How does one implement the search for winning strategies using, say, unification?