LPAR-26 · SPETSES

Towards an Operational Proof-theoretic Semantics

Alexander V. Gheorghiu
University of Southampton · University College London

WHAT IS PROOF-THEORETIC SEMANTICS?

Two ways to give meaning

MEANING AS STRUCTURE

A **model**: states and their relations.

$$\mathcal{M} \models \varphi$$

Does the structure make φ true?

MEANING AS BEHAVIOUR

A **base**: the rules a system follows.

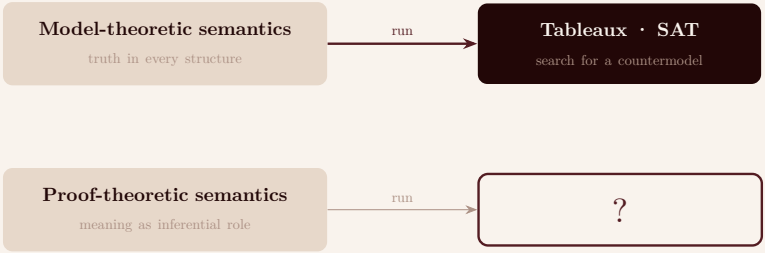
$$\Vdash_{\mathcal{B}} \varphi$$

Do the rules commit us to φ?

Roughly: structures correspond to complete, consistent sets of atomic rules (Lindenbaum & Tarski 1935; Makinson 2014).

A MISSING MACHINE

Model theory comes with a machine



WHAT IS PROOF-THEORETIC SEMANTICS?

The right language for the right logic

Uniform proofs (Miller, Nadathur, Pfenning & Scedrov 1991): the goal drives the search, and the program is consulted only at atoms.

BASES ARE PROGRAMS IN	THE SUPPORT CLAUSES THEN GIVE	
Horn clauses (Prolog)	classical logic	Sandqvist 2009
hereditary Harrop (λProlog)	**intuitionistic logic**	Sandqvist 2015

- Hereditary Harrop adds hypothetical goals: to prove $D \supset G$, add D to the program and prove G.
- A second-order quantifier over *atoms* keeps every base finite.

WHAT IS PROOF-THEORETIC SEMANTICS?

Support in a base

Sandqvist's clauses. Bases $\mathcal{B}$, $\mathcal{C}$ are hereditary Harrop programs; A ranges over atoms.

$\Vdash_{\mathcal{B}} A$	iff	$\mathcal{B} \vdash A$
$\Vdash_{\mathcal{B}} \varphi \to \psi$	iff	$\varphi \Vdash_{\mathcal{B}} \psi$
$\Vdash_{\mathcal{B}} \varphi \wedge \psi$	iff	$\Vdash_{\mathcal{B}} \varphi$ and $\Vdash_{\mathcal{B}} \psi$
$\Vdash_{\mathcal{B}} \varphi \vee \psi$	iff	for every $\mathcal{C} \supseteq \mathcal{B}$ and atom A: if $\varphi \Vdash_{\mathcal{C}} A$ and $\psi \Vdash_{\mathcal{C}} A$, then $\Vdash_{\mathcal{C}} A$
$\Vdash_{\mathcal{B}} \bot$	iff	$\Vdash_{\mathcal{B}} A$ for every atom A
$\Gamma \Vdash_{\mathcal{B}} \varphi$	iff	for every $\mathcal{C} \supseteq \mathcal{B}$: if $\Vdash_{\mathcal{C}} \psi$ for all $\psi \in \Gamma$, then $\Vdash_{\mathcal{C}} \varphi$

φ is valid iff $\Vdash_{\emptyset} \varphi$. With hereditary Harrop bases, the valid formulas are exactly the intuitionistic theorems (Sandqvist 2015).

A MISSING MACHINE

The question

What does for base-extension semantics what tableaux do for model theory?

TRANSPOSE TABLEAUX?

Tableaux search for a structure.

Base-extension semantics has no structures to find, only rules.

PROPOSAL

Search through the rules: **logic programming**.

Itself a mature piece of good old-fashioned AI.

PART 1

Support is Search

Bases are logic programs.

SUPPORT IS SEARCH

The claim

> Logic programming is to proof-theoretic semantics what tableaux and SAT are to model-theoretic semantics.

THE WEAK CLAIM

LP can implement the semantics.

Any mathematics can be implemented.

THE CLAIM

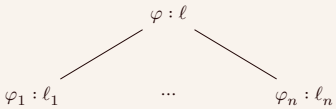
Each step of the search **is** a clause of the semantics.

As each tableau rule is a clause of satisfaction.

SUPPORT IS SEARCH

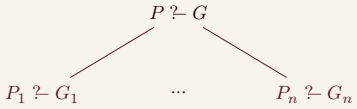
Goal-directed search has the same shape as a tableau

TABLEAUX



Suppose φ false; its main connective picks the rule; a branch closes when it is contradictory.

LOGIC PROGRAMMING



Query G; its main connective picks the rule; a branch closes when the program supplies the goal.

SUPPORT IS SEARCH

Main theorem

$$\Vdash_{\mathcal{B}} \varphi \qquad \text{iff} \qquad \mathcal{B} \vdash_{O} [\![\varphi]\!]$$

for every base $\mathcal{B}$ and every formula φ of intuitionistic propositional logic

BASES

A base is a program, used as it stands.

Only the formula is encoded.

BASE BY BASE

The equivalence holds at every base.

Not only at the level of validity.

VALIDITY

Validity is **one query**:

$\emptyset \vdash [\![\varphi]\!]$?

SUPPORT IS SEARCH

The encoding is read off the support clauses

THE SUPPORT CLAUSE

$\Vdash_{\mathcal{B}} \varphi \vee \psi$ iff
for every $\mathcal{C} \supseteq \mathcal{B}$ and every atom A:
if $\varphi \Vdash_{\mathcal{C}} A$ and $\psi \Vdash_{\mathcal{C}} A$, then $\Vdash_{\mathcal{C}} A$

THE GOAL

$$[\![\varphi \vee \psi]\!] = \forall X\,(((([\![\varphi]\!] \supset X) \cap ([\![\psi]\!] \supset X)) \supset X)$$

X stands for an arbitrary conclusion.

every atom A $\longmapsto$ $\forall X$ search tries one fresh atom

every extension $\mathcal{C} \supseteq \mathcal{B}$ $\longmapsto$ $\supset$ search adds the hypothesis to the program

Russell and Prawitz defined each connective from $\supset$ and $\forall$; here $\forall$ ranges over atoms only.

SUPPORT IS SEARCH

The full encoding

$\cap$ and $\supset$ are the program's conjunction and implication; X is a variable over atoms.

$$
\begin{aligned}
[\![A]\!] &= A && \text{an atom is its own goal} \\
[\![\bot]\!] &= \forall X\, X && \text{every conclusion follows} \\
[\![\varphi \wedge \psi]\!] &= [\![\varphi]\!] \cap [\![\psi]\!] && \text{both parts} \\
[\![\varphi \vee \psi]\!] &= \forall X\, (([\![\varphi]\!] \supset X) \cap ([\![\psi]\!] \supset X)) \supset X) && \text{whatever each disjunct yields} \\
[\![\varphi \to \psi]\!] &= \forall X\, ([\![\varphi]\!] \cap ([\![\psi]\!] \supset X)) \supset X) && \varphi\text{, handed the continuation for } \psi
\end{aligned}
$$

- Only atoms, $\cap$, $\supset$, $\forall$ occur and every head is atomic, so an encoded formula can be queried *and* installed as a hypothesis.
- Hence implication in elimination form: $[\![\varphi]\!] \supset [\![\psi]\!]$ would have a compound head.

SUPPORT IS SEARCH

Running a query: the rules

$[P]$, the elaboration of P, splits conjunctions and instantiates $\forall$ at every atom.

FACT	$P \vdash_O A$ if $A \in [P]$	find it in the program
BACKCHAIN	$P \vdash_O A$ if $G \supset A \in [P]$ and $P \vdash_O G$	find a clause with that head; prove its body
AND	$P \vdash_O G_1 \cap G_2$ if $P \vdash_O G_1$ and $P \vdash_O G_2$	prove both
OR	$P \vdash_O G_1 \cup G_2$ if $P \vdash_O G_1$ or $P \vdash_O G_2$	choose one
AUGMENT	$P \vdash_O D \supset G$ if $P \cup \{D\} \vdash_O G$	assume D and carry on
GENERIC	$P \vdash_O \forall X\, G$ if $P \vdash_O G[A/X]$, A fresh	one atom nothing is known about

Only Fact and Backchain look at the program; the goal's main connective picks every other rule.

SUPPORT IS SEARCH

Running a query

QUERY AGAINST $\emptyset$

$[\![(p \wedge q) \to p]\!]$	✓	succeeds in four steps
$[\![((p \to q) \to p) \to p]\!]$	×	Peirce: every branch comes back to the goal $[\![p \to q]\!]$

A failed search shows *why* the formula fails.

Installed as a hypothesis, $(p \to q) \to p$ yields p only given $p \to q$, and nothing yields q.

Peirce's formula is not supported at the **empty base**.

The search builds no countermodel to show it.

PART 2

Consequences

What each side gets, and what is still open.

CONSEQUENCES

Where the analogy with tableaux breaks

POLARITY

Tableau nodes are labelled `t` or `f`. Queries carry no label.

Nothing is supposed false.

BRANCHES

A tableau must close every branch. A query needs one success.

A failed query returns no countermodel.

GROWTH

Tableaux grow by subformulas. Programs grow by hypotheses.

The program consulted at the leaves is not fixed in advance.

A tableau handles the quantifier over structures by building one structure.
Here the quantifier over extensions is already inside the support clauses.

CONSEQUENCES

What each side gets

PROOF-THEORETIC SEMANTICS

A way to compute support.

To find out whether a base supports φ, run the query.

The encoding is fixed by the support clauses; nobody had to invent it.

LOGIC PROGRAMMING

A semantics for its programs.

A hereditary Harrop program gives meaning to the formulas it supports; it is more than a database of facts.

Uniformity, the restriction that makes search goal-directed, turns out to be a semantic condition.

CONSEQUENCES

A programme

For each logic with a base-extension semantics, find the LP language whose search steps match its support clauses.

SEMANTICS		LOGIC PROGRAMMING	
intuitionistic	⟷	hereditary Harrop	✓ this paper
first-order	⟷	first-order hereditary Harrop	the natural first test
IMLL	⟷	Lolli (linear hereditary Harrop)	expected
classical	⟷	**Horn**	**open**

CONSEQUENCES

Open: classical logic

- Sandqvist fixes the target: over Horn bases, support gives classical logic.
- The encoding needs hypothetical goals, and Horn clauses have none.
- Glivenko's theorem lets an intuitionistic query decide classical validity. That reduces classical logic to intuitionistic logic; it does not run a classical base.

A SUGGESTION

In typed λ-calculus, continuations at classical type are **control operators**.

So perhaps: Horn programs plus a restart rule.

SUMMARY

Takeaway

Logic programming is to proof-theoretic semantics
what tableaux and SAT are to model-theoretic semantics.

- Bases are logic programs, and support is proof search.
- Intuitionistic validity is one query against the empty program.
- For other logics, the task is to find the matching language. Classical logic is open.

REFERENCES & THANKS

Sources

- *Towards an Operational Proof-theoretic Semantics.* LPAR-26 (this talk).
- *Support is Search.* arXiv:2603.13018. The extended version, with full proofs.
- Sandqvist, *Base-extension Semantics for Intuitionistic Sentential Logic.* Logic J. IGPL **23** (2015).
- Sandqvist, *Classical Logic without Bivalence.* Analysis **69** (2009).
- Miller, Nadathur, Pfenning & Scedrov, *Uniform Proofs as a Foundation for Logic Programming.* APAL **51** (1991).

Alexander V. Gheorghiu · University of Southampton & UCL

Thank you.