Gen2sat:
A Generic Tool for Reasoning with Non-classical Logics

Yoni Zohar – Tel Aviv University

Joint work with Ori Lahav and Anna Zamansky

AIC @ SRI
August 24, 2017
Motivation

- Propositional classical logic:
  - A lot of research
  - Used in applications

- Propositional Non-classical logics:
  - Many-valued and fuzzy logics: \( \nu(A) = 0.5 \)
  - Paraconsistent logics: \( A, \neg A \not\vdash B \)
  - Intuitionistic logic: \( \not\vdash A \lor \neg A \)
  - Modal logics: \( \Box A \)
  - A lot of research
  - Few are used in applications

- A possible explanation:
  - Lack of available tools for reasoning with non-classical logics
  - One has to develop a reasoning tool from scratch for each logic
Tool for Non-classical Logics

Formula

Logic

Decision Procedure

Valid?

Yoni Zohar

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
Tool for Non-classical Logics

Gen2sat

Derivable?

Sequent

Sequent Calculus

Yoni Zohar

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
Tool for Non-classical Logics

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics

Sequent

Sequent Calculus

Derivable?
Tool for Non-classical Logics

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics

Sequent
Sequent
Calculus

Derivable?

Works for Propositional Pure Analytic Sequent Calculi with “Next” Operators

IN:
classical logic
3-valued logics
4-valued logics
paraconsistent logics
...

OUT:
intuitionistic logic
relevance logics
fuzzy logics
first-order logics
...

Yoni Zohar
Tool for Non-classical Logics

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics

Yoni Zohar
Tool for Non-classical Logics

Sequent

Reduction

SAT Solver

Derivable?

Sequent Calculus
Tool for Non-classical Logics

Yoni Zohar

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
Tool for Non-classical Logics

Sequent

Semantic Interpretation → Reduction

Derivable?

Sequent Calculus

Yoni Zohar

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
Tool for Non-classical Logics

Sequent Calculus → Semantic Interpretation → Reduction → SAT4j Library → Derivable?
What Are Sequents?

- **Sequents** have the form \( \Gamma \Rightarrow \Delta \), where \( \Gamma, \Delta \) are finite sets of formulas.
- Intuition:

\[
A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \iff A_1 \land \ldots \land A_n \rightarrow B_1 \lor \ldots \lor B_m
\]

Example

- \( A, B \Rightarrow A \land B \)
- \( A \Rightarrow A \lor B \)
- \( \Rightarrow A \lor \neg A \)
- \( A, \neg A \Rightarrow \)
- \( \Rightarrow A, \neg A \)
- \( A \Rightarrow A, B, C \)
Tool for Non-classical Logics

Sequent

Semantic Interpretation → Reduction

Derivable?

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
Sequent Calculi

- Proof systems that manipulate sequents
- Sequent Calculus = finite set of sequent derivation rules

\[
\frac{\Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_n \Rightarrow \Delta_n}{\Gamma_0 \Rightarrow \Delta_0}
\]

Example (Thinking of a Sequent Rule)

\[
\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A \land B} \quad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A \land B} \quad \frac{A \Rightarrow \Delta}{A \land B \Rightarrow \Delta} \quad \frac{B \Rightarrow \Delta}{A \land B \Rightarrow \Delta}
\]
Sequent Calculi

- Proof systems that manipulate sequents
- Sequent Calculus = finite set of sequent derivation rules

\[ \Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_n \Rightarrow \Delta_n \]
\[ \Gamma_0 \Rightarrow \Delta_0 \]

Example (Thinking of a Sequent Rule)

\[ \Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta \]
\[ \frac{\Gamma \Rightarrow A \land B, \Delta}{\Gamma \Rightarrow A \land B, \Delta} \]
\[ \Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta \]
\[ \frac{\Gamma, A \Rightarrow \Delta}{\Gamma, A \land B \Rightarrow \Delta} \quad \frac{\Gamma, B \Rightarrow \Delta}{\Gamma, A \land B \Rightarrow \Delta} \]
### The Propositional Fragment of LK [Gentzen 1934]

#### Structural Rules:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(id)</td>
<td>$\Gamma, A \Rightarrow A, \Delta$</td>
</tr>
<tr>
<td>(cut)</td>
<td>$\Gamma \Rightarrow A, \Delta, \Gamma, A \Rightarrow \Delta \vdash \Gamma \Rightarrow \Delta$</td>
</tr>
<tr>
<td>(weak)</td>
<td>$\Gamma \Rightarrow \Delta, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$</td>
</tr>
</tbody>
</table>

#### Logical Rules:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>($\neg \Rightarrow$)</td>
<td>$\Gamma \Rightarrow A, \Delta \vdash \Gamma, \neg A \Rightarrow \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \neg$)</td>
<td>$\Gamma, A \Rightarrow \Delta \vdash \Gamma \Rightarrow \neg A, \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \wedge$)</td>
<td>$\Gamma, A \wedge B \Rightarrow \Delta \vdash \Gamma \Rightarrow A, \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \wedge$)</td>
<td>$\Gamma \Rightarrow A, \Delta \vdash \Gamma \Rightarrow A \wedge B, \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \vee$)</td>
<td>$\Gamma, A \Rightarrow \Delta \vdash \Gamma \Rightarrow A \vee B, \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \vee$)</td>
<td>$\Gamma, B \Rightarrow \Delta \vdash \Gamma \Rightarrow A \vee B, \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \supset$)</td>
<td>$\Gamma \Rightarrow A, \Delta \vdash \Gamma \Rightarrow B, \Delta$</td>
</tr>
<tr>
<td>($\Rightarrow \supset$)</td>
<td>$\Gamma, A \Rightarrow B, \Delta \vdash \Gamma \Rightarrow A \supset B, \Delta$</td>
</tr>
</tbody>
</table>

### Analyticity (“subformula property”)

Only subformulas of the proved sequent are used.
Examples of Sequent Calculi

Łukasiewicz 3-valued Logic [Avron ’03]

A sequent calculus for Ł₃ is obtained by augmenting the positive fragment of LK with some pure rules for negation. For example:

\[(\neg \supset \Rightarrow) \quad \frac{\Gamma, A, \neg B \Rightarrow \Delta}{\Gamma, \neg (A \supset B) \Rightarrow \Delta}\]

\[(\Rightarrow \neg \supset) \quad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow \neg B, \Delta}{\Gamma \Rightarrow \neg (A \supset B), \Delta}\]

¬-Analyticity

Only subformulas of the proved sequent and their negations are used.
Examples of Sequent Calculi

Calculus for Primal Infon Logic [Gurevich, Neeman ’09]

\[(\wedge \Rightarrow)\]
\[
\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \land B \Rightarrow \Delta}
\]
\[
\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \lor B \Rightarrow \Delta}
\]
\[
\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}
\]

\[(\Rightarrow \wedge)\]
\[
\frac{\Gamma \Rightarrow A, \Delta}{\Gamma \Rightarrow A \land B, \Delta}
\]
\[
\frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \lor B, \Delta}
\]
\[
\frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}
\]

\[(\Rightarrow \supset)\]
\[
\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}
\]

\[(\Rightarrow \lor)\]
\[
\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \lor B, \Delta}
\]

\[(\Rightarrow \land)\]
\[
\frac{\Gamma \Rightarrow A, \Delta}{\Gamma \Rightarrow A \land B, \Delta}
\]
\[
\frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \lor B, \Delta}
\]
\[
\frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}
\]

\[q \ said \ \Gamma \Rightarrow q \ said \ \Delta \]

for every principal q

- An extremely efficient propositional logic.
- One of the main logical engines behind MSR DKAL

Analyticity

Only subformulas of the proved sequent are used
Pure Analytic sequent calculus with “Next” Operators

- Propositional and structural
- Include pure logical rules that allow any $\Gamma$ and $\Delta$:

  \[
  \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}
  \]

  \[
  \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}
  \]

- May include impure rules of the form:

  \[
  \frac{\Gamma \Rightarrow \Delta}{*\Gamma \Rightarrow *\Delta}
  \]

- Analytic (subformula property and its variants)

Theorem

There is a polynomial reduction from the derivability problem of any such calculus to (the complement of) SAT.

corollary

The derivability problem for every such calculus is in co-NP.
Tool for Non-classical Logics

Yoni Zohar

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
A valuation is a function \( v : WFF \rightarrow \{ T, F \} \)

**Warning**

Valuations are defined over all formulas, not only the atomic ones!

**G-legal valuations**

A valuation is *G-legal* if it respects the “semantic reading” of \( G \).

**Example (Classical Conjunction)**

\[
\begin{align*}
\Rightarrow A & \quad \Rightarrow B \\
\Rightarrow A \land B & \\
A \Rightarrow & \\
A \land B \Rightarrow & \\
B \Rightarrow & \\
A \land B \Rightarrow &
\end{align*}
\]

1. If \( v(A) = T \) and \( v(B) = T \) then \( v(A \land B) = T \)
2. If \( v(A) = F \) then \( v(A \land B) = F \)
3. If \( v(B) = F \) then \( v(A \land B) = F \)
Semantics for Pure Calculi

valuations

A valuation is a function $v : WFF \rightarrow \{T, F\}$

Warning

Valuations are defined over all formulas, not only the atomic ones!

$G$-legal valuations

A valuation is $G$-legal if it respects the “semantic reading” of $G$.

Example (Sequent Calculus for $C_1$)

\[
\frac{A \Rightarrow}{\Rightarrow \neg A} \quad \frac{A \Rightarrow}{\neg \neg A \Rightarrow} \quad \frac{\neg A \Rightarrow \neg B \Rightarrow}{\neg (A \land B) \Rightarrow}
\]

1. If $v(A) = F$ then $v(\neg A) = T$
2. If $v(A) = F$ then $v(\neg \neg A) = F$
3. If $v(\neg A) = F$ and $v(\neg B) = F$ then $v(\neg (A \land B)) = F$

This semantics is non-deterministic.
Semantics for Pure Calculi

Soundness and Completeness [Béziau ‘01]

\[ s \text{ is provable in } G \iff s \text{ is satisfied by every } G\text{-legal valuation} \]
Semantics for Pure Calculi

Soundness and Completeness

\[ s \text{ is provable in } G \text{ using } \mathcal{F} \subseteq WFF \]

\[ \iff \]

\[ s \text{ is satisfied by every } G\text{-legal valuation with domain } \mathcal{F} \]
Semantics for Pure Calculi

Soundness and Completeness

\[ s \text{ is provable in } G \text{ using } \text{sub}(s) \iff s \text{ is satisfied by every } G\text{-legal valuation with domain } \text{sub}(s) \]
Soundness and Completeness

\[ s \text{ is provable in } G \iff s \text{ is satisfied by every } G\text{-legal valuation with domain } \text{sub}(s) \]
Tool for Non-classical Logics

Yoni Zohar

Gen2sat: A Generic Tool for Reasoning with Non-classical Logics
Reduction to SAT

Theorem

- Correctness: \( s \) is provable in \( G \) iff the generated instance is UNSAT.
- Complexity:
  - Translating: \( O(n^k) \); \( k \) depends on \( G \) (“usually” \( k = 1 \))
  - Solving: Exp in the worse case. Linear for “Horn calculi”

- In the presence of Next operators, we use Kripke models
- Correctness is more challenging
Propositional logic example

The clauses which define the semantics of propositional logic provide instructive examples of the resolution rule. Here if $x$ and $y$ name propositions $x^*$ and $y^*$ respectively then

$$
\begin{align*}
x \land y & \quad \text{names the proposition } x^* \text{ and } y^* \\
x \lor y & \quad \text{ } x^* \text{ or } y^* \\
x \supset y & \quad \text{if } x^* \text{ then } y^* \\
x \leftrightarrow y & \quad x^* \text{ if and only if } y^* \\
\neg x & \quad \text{it is not the case that } x^*.
\end{align*}
$$

where $\land, \lor, \supset, \leftrightarrow$ and $\neg$ are infix function symbols. Read $\text{True}(x)$ as stating that $x$ is true. The following set of clauses cannot be reexpressed as Horn clauses by renaming predicate symbols.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>$\text{True}(x&amp;y) \leftarrow \text{True}(x), \text{True}(y)$</td>
</tr>
<tr>
<td>T2</td>
<td>$\text{True}(x) \leftarrow \text{True}(x&amp;y)$</td>
</tr>
<tr>
<td>T3</td>
<td>$\text{True}(y) \leftarrow \text{True}(x&amp;y)$</td>
</tr>
<tr>
<td>T4</td>
<td>$\text{True}(x\lor y) \leftarrow \text{True}(x)$</td>
</tr>
<tr>
<td>T5</td>
<td>$\text{True}(x\lor y) \leftarrow \text{True}(y)$</td>
</tr>
<tr>
<td>T6</td>
<td>$\text{True}(x), \text{True}(y) \leftarrow \text{True}(x\lor y)$</td>
</tr>
<tr>
<td>T7</td>
<td>$\text{True}(x\supset y), \text{True}(x) \leftarrow$</td>
</tr>
<tr>
<td>T8</td>
<td>$\text{True}(x\supset y) \leftarrow \text{True}(y)$</td>
</tr>
<tr>
<td>T9</td>
<td>$\text{True}(y) \leftarrow \text{True}(x), \text{True}(x\supset y)$</td>
</tr>
<tr>
<td>T10</td>
<td>$\text{True}(x\leftrightarrow y) \leftarrow \text{True}(x \ y), \text{True}(y\rightarrow x)$</td>
</tr>
<tr>
<td>T11</td>
<td>$\text{True}(x\leftrightarrow y) \leftarrow \text{True}(x\leftrightarrow y)$</td>
</tr>
<tr>
<td>T12</td>
<td>$\text{True}(y\rightarrow x) \leftarrow \text{True}(x\leftrightarrow y)$</td>
</tr>
<tr>
<td>T13</td>
<td>$\text{True}(\neg x), \text{True}(x) \leftarrow$</td>
</tr>
<tr>
<td>T14</td>
<td>$\leftarrow \text{True}(\neg x), \text{True}(x)$</td>
</tr>
</tbody>
</table>
Gen2sat: A Generic Tool for Reasoning with Non-classical Logics

Tool for Non-classical Logics

- Sequent
- Sequent Calculus
- Semantic Interpretation
- Reduction
- Derivable?
Reduction

\[ A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \]

\[ \text{rule}_1 \]
\[ \text{rule}_2 \]
\[ \text{rule}_3 \]
\[ \text{rule}_4 \]

\[ \text{clause 1} \]
\[ \text{clause 2} \]
\[ \text{clause 3} \]
\[ \text{clause 4} \]
\[ \text{clause 5} \]
\[ \text{clause 6} \]
\[ \text{clause 7} \]
\[ \text{clause 8} \]
\[ \text{clause 9} \]
\[ \text{clause 10} \]
\[ A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \]

**Rule 1**

**Rule 2**

**Rule 3**

**Rule 4**

**SAT**

SAT assignment

\[ A_1 = false, A_2 = true, \ldots \]
Reduction

\[ A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \]

\text{rule}_1 \quad \text{rule}_2 \quad \text{rule}_3 \quad \text{rule}_4

\begin{align*}
\text{clause 1} \\
\text{clause 2} \\
\text{clause 3} \\
\text{clause 4} \\
\text{clause 5} \\
\text{clause 6} \\
\text{clause 7} \\
\text{clause 8} \\
\text{clause 9} \\
\text{clause 10}
\end{align*}

\text{UNSAT}

\text{UNSAT core}

2, 4, 8
Evaluation

- **Gen2sat vs. MetTeL**
- **input:** The calculus for Łukasiewicz 3-valued logic
- **Structured Problems** for Łukasiewicz infinite-valued logic [Rothenberg’07]
- **Random Problems** generated by MetTeL

**Structured Problems**

<table>
<thead>
<tr>
<th>N</th>
<th>time in ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>5,000</td>
</tr>
<tr>
<td>200</td>
<td>10,000</td>
</tr>
<tr>
<td>300</td>
<td>10,000</td>
</tr>
</tbody>
</table>

**Random Problems**

<table>
<thead>
<tr>
<th>depth</th>
<th>time in ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>10^3</td>
</tr>
<tr>
<td>20</td>
<td>10^4</td>
</tr>
</tbody>
</table>
An Idea: Logic Education

Motivation:
- Gen2sat can be useful for teaching sequent calculi
- Students can focus solely on the logical aspects
- Heuristics and search are left for the SAT solver

Preliminary Pilot:
- 13 logic students were given a bonus assignment: *present a minimal test plan with maximal coverage for Gen2sat*
- They all got 70%-85% coverage
- Some used 0-ary and 3-ary connectives.
- Some found (intentionally planted) bugs
- Feedback from students was encouraging
  - “it helped me see the variety of different connectives and rules”
  - “for me thinking of the extreme cases was really illuminating”
  - “I wish all of the course assignments were more of this type”
Conclusion

We have seen:

- A generic tool for deciding derivability in analytic pure (and some impure) sequent calculi
- The actual search is done by a SAT-solver
- Based on a semantic interpretation

Future work:

- Applications
- Support more logics and calculi
- Automatically detect analyticity (when possible)
- Integrate with a theorem prover
Conclusion

We have seen:

- A generic tool for deciding derivability in analytic pure (and some impure) sequent calculi
- The actual search is done by a SAT-solver
- Based on a semantic interpretation

Future work:

- Applications
- Support more logics and calculi
- Automatically detect analyticity (when possible)
- Integrate with a theorem prover

Thank you!