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

Yoni Zohar – Tel Aviv University

Joint work with Ori Lahav and Anna Zamansky

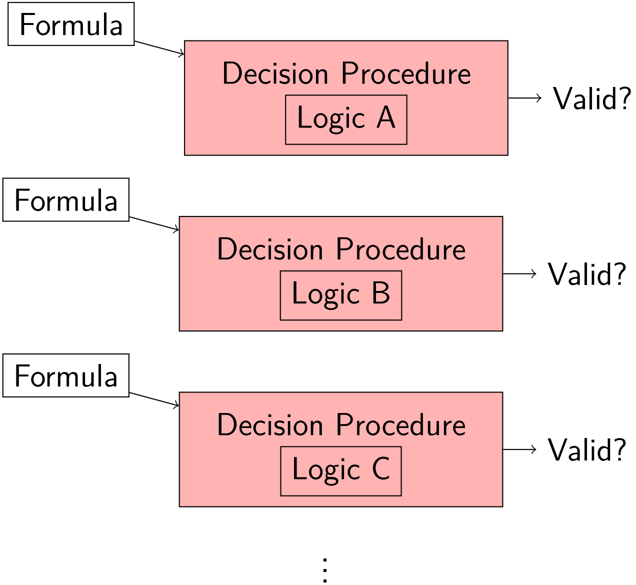
MUGS

April 26, 2017

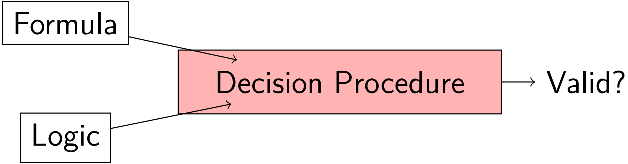
Motivation

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

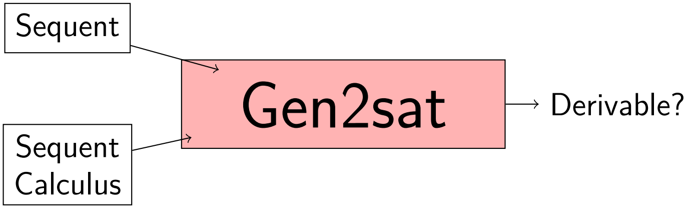
Tools for Non-classical Logics



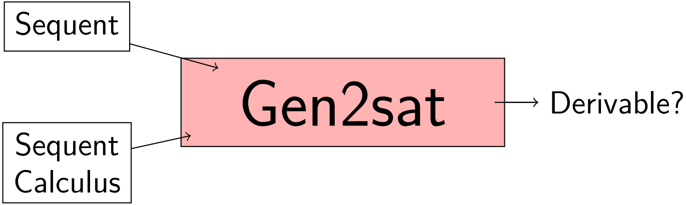
Tool for Non-classical Logics



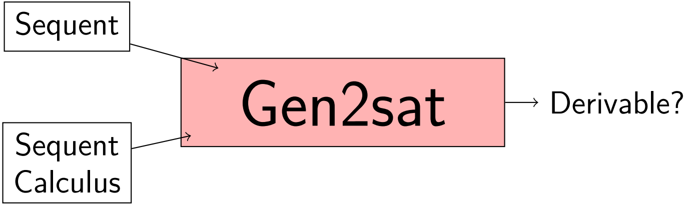
Tool for Non-classical Logics



Tool for Non-classical Logics



Tool for Non-classical Logics

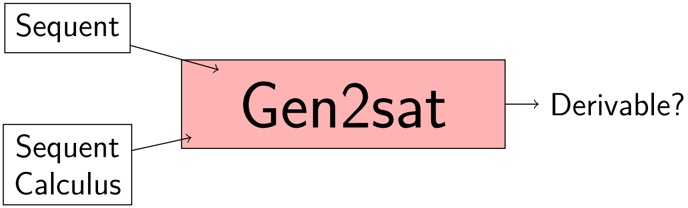


Works for Propositional Pure Analytic Sequent Calculi
with "Next" Operators

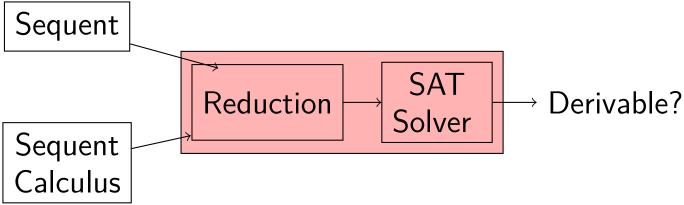
IN:
classical logic
3-valued logics
4-valued logics
paraconsistent logics
...

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

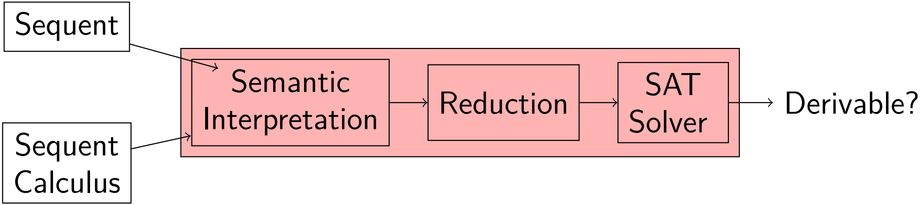
Tool for Non-classical Logics



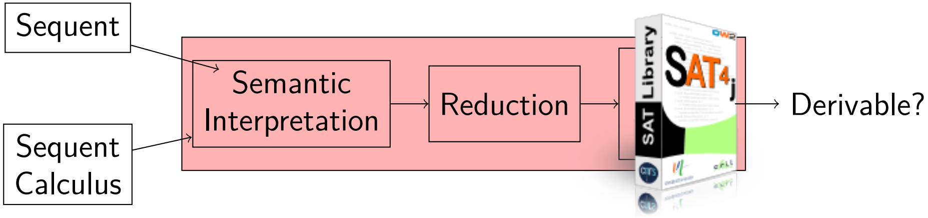
Tool for Non-classical Logics



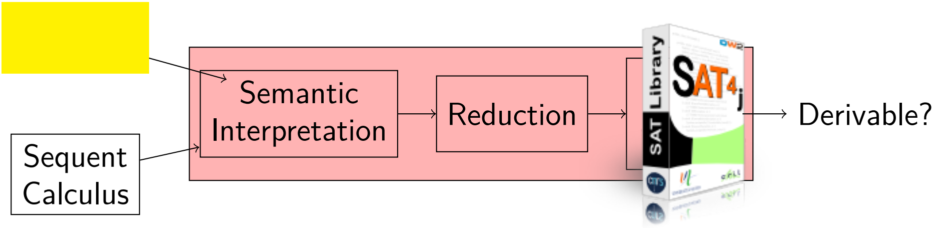
Tool for Non-classical Logics



Tool for Non-classical Logics



Tool for Non-classical Logics



What Are Sequents?

- *Sequents* have the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of formulas.
- Intuition:

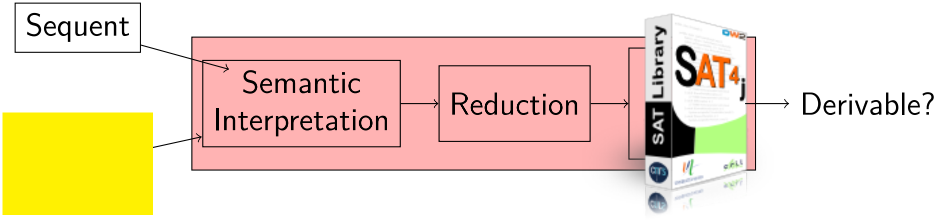
$$A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \quad \rightsquigarrow \quad A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_m$$

- Special instance 1: Δ has one element: $\Gamma \Rightarrow A$
- Special instance 2: Γ is empty: $\Rightarrow A$

Example

- $A, B \Rightarrow A \wedge B$
- $A \Rightarrow A \vee B$
- $\Rightarrow A \vee \neg A$
- $A, \neg A \Rightarrow$
- $\Rightarrow A, \neg A$
- $A \Rightarrow A, B, C$

Tool for Non-classical Logics



Sequent Calculi

Sequent Calculi

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

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

Examples of Sequent Calculi

The Propositional Fragment of **LK** [Gentzen 1934]

Structural Rules:

$$(id)\quad \frac{}{\Gamma, A \Rightarrow A, \Delta} \qquad (cut)\quad \frac{\Gamma, A \Rightarrow \Delta \quad \Gamma \Rightarrow A, \Delta}{\Gamma \Rightarrow \Delta}$$

$$(W \Rightarrow)\quad \frac{\Gamma \Rightarrow \Delta}{\Gamma, A \Rightarrow \Delta} \qquad (\Rightarrow W)\quad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow A, \Delta}$$

Logical Rules:

$$(\neg \Rightarrow)\quad \frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta} \qquad (\Rightarrow \neg)\quad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta}$$

$$(\wedge \Rightarrow)\quad \frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \qquad (\Rightarrow \wedge)\quad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta}$$

$$(\vee \Rightarrow)\quad \frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta} \qquad (\Rightarrow \vee)\quad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}$$

$$(\supset \Rightarrow)\quad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \supset B \Rightarrow \Delta} \qquad (\Rightarrow \supset)\quad \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}$$

A Sequent Proof of Peirce's Law

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow A}}{A \Rightarrow B, A}}{\Rightarrow A \supset B, A} \qquad \dfrac{}{A \Rightarrow A}}{\dfrac{(A \supset B) \supset A \Rightarrow A}{\color{red}{\Rightarrow ((A \supset B) \supset A) \supset A}}}$$

$$(id)\,\dfrac{}{A \Rightarrow A}$$

A Sequent Proof of Peirce's Law

$$\dfrac{\dfrac{\dfrac{\overline{A \Rightarrow A}}{A \Rightarrow B, A}}{\Rightarrow A \supset B, A} \qquad \overline{A \Rightarrow A}}{\dfrac{(A \supset B) \supset A \Rightarrow A}{\Rightarrow ((A \supset B) \supset A) \supset A}}$$

$$(\Rightarrow W)\dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow A, \Delta}$$

A Sequent Proof of Peirce's Law

$$\dfrac{\dfrac{\dfrac{\overline{A \Rightarrow A}}{A \Rightarrow B, A}}{\Rightarrow A \supset B, A} \qquad \overline{A \Rightarrow A}}{\dfrac{(A \supset B) \supset A \Rightarrow A}{\Rightarrow ((A \supset B) \supset A) \supset A}}$$

$$(\Rightarrow\supset)\dfrac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}$$

A Sequent Proof of Peirce's Law

$$\cfrac{\cfrac{\cfrac{\cfrac{}{A \Rightarrow A}}{A \Rightarrow B, A}}{\Rightarrow A \supset B, A} \qquad \cfrac{}{A \Rightarrow A}}{\cfrac{(A \supset B) \supset A \Rightarrow A}{\Rightarrow ((A \supset B) \supset A) \supset A}}$$

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

A Sequent Proof of Peirce's Law

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow A}}{A \Rightarrow B, A}}{\Rightarrow A \supset B, A} \qquad \dfrac{}{A \Rightarrow A}}{\dfrac{(A \supset B) \supset A \Rightarrow A}{\Rightarrow ((A \supset B) \supset A) \supset A}}
$$

$$
(\Rightarrow\supset)\dfrac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}
$$

A Sequent Proof of Peirce's Law

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow A}}{A \Rightarrow B, A}}{\Rightarrow A \supset B, A} \qquad \dfrac{}{A \Rightarrow A}}{\dfrac{(A \supset B) \supset A \Rightarrow A}{\Rightarrow ((A \supset B) \supset A) \supset A}}$$

$$(\Rightarrow\supset)\frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}$$

The Subformula Property

Only subformulas of the proved sequent are used!

Examples of Sequent Calculi

Łukasiewicz 3-valued Logic [Avron '03]

A sequent calculus for $Ł_3$ is obtained by augmenting the positive fragment of **LK** with some pure rules for negation. For example:

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

$$(\Rightarrow\neg\supset)\quad \frac{\Gamma \Rightarrow A, \Delta \qquad \Gamma \Rightarrow \neg B, \Delta}{\Gamma \Rightarrow \neg(A \supset B), \Delta}$$

The $\neg$-Subformula Property

Only subformulas of the proved sequent and their negations are used!

Examples of Sequent Calculi

A Calculus for da-Costa's $\mathbf{C}_1$ [Avron, Konikowska, Zamansky '12]

$$\frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta} \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta} \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma, \neg\neg A \Rightarrow \Delta}$$

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow \neg A, \Delta}{\Gamma, \neg(A \wedge \neg A) \Rightarrow \Delta} \qquad \frac{\Gamma, \neg A \Rightarrow \Delta \quad \Gamma, \neg B \Rightarrow \Delta}{\Gamma, \neg(A \wedge B) \Rightarrow \Delta}$$

$$\frac{\Gamma, \neg A \Rightarrow \Delta \quad \Gamma, B, \neg B \Rightarrow \Delta}{\Gamma, \neg(A \vee B) \Rightarrow \Delta} \qquad \frac{\Gamma, A, \neg A \Rightarrow \Delta \quad \Gamma, \neg B \Rightarrow \Delta}{\Gamma, \neg(A \vee B) \Rightarrow \Delta}$$

$$\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B, \neg B \Rightarrow \Delta}{\Gamma, \neg(A \supset B) \Rightarrow \Delta} \qquad \frac{\Gamma, A, \neg A \Rightarrow \Delta \quad \Gamma, \neg B \Rightarrow \Delta}{\Gamma, \neg(A \supset B) \Rightarrow \Delta}$$

The $\neg$-Subformula Property

Only subformulas of the proved sequent and their negations are used!

Examples of Sequent Calculi

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

$$(\wedge \Rightarrow) \quad \frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \qquad (\Rightarrow \wedge) \quad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta}$$

$$(\vee \Rightarrow) \quad \xcancel{\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta}} \qquad (\Rightarrow \vee) \quad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}$$

$$(\supset \Rightarrow) \quad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \supset B \Rightarrow \Delta} \qquad (\Rightarrow \supset) \quad \frac{\Gamma, \xcancel{A} \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}$$

$$\frac{\Gamma \Rightarrow \Delta}{q \ said \ \Gamma \Rightarrow q \ said \ \Delta}$$ for every principal q

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

The Subformula Property

Only subformulas of the proved sequent are used!

Analytic Pure Sequent Calculi with "Next" Operators

Pure sequent calculi with "Next" Operators

- propositional and structural
- include pure logical rules that allow any Γ and Δ:

✓ $$\frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta}$$

✗ $$\frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}$$

- May include impure rules of the form:

$$\frac{\Gamma \Rightarrow \Delta}{*\Gamma \Rightarrow *\Delta}$$

Analytic sequent calculi

- Admit the subformula property
- Weaker notions are possible (e.g. negations)

Main Result

The Derivability Problem of a Calculus G

Input: A sequent s
Output: Is s derivable in G?

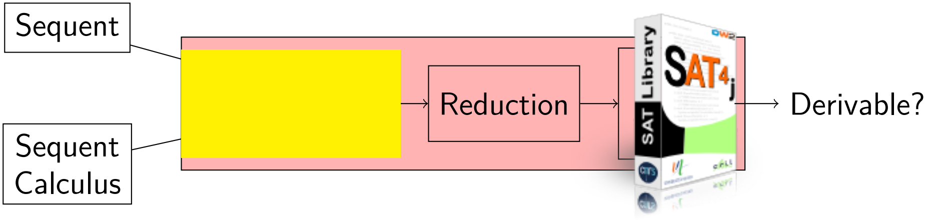
Theorem

There is a polynomial reduction from the derivability problem of any pure analytic sequent calculus with "Next" operators to (the complement) of SAT.

Corollary

For these calculi, the derivability problem is in co-NP.

Tool for Non-classical Logics



Semantics for Pure Calculi

valuations

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

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 the rules of G.

Semantics for Pure Calculi

valuations

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

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 the rules of G.

Example (Classical Conjunction)

$$\frac{\Rightarrow A \qquad \Rightarrow B}{\Rightarrow A \wedge B} \qquad \frac{A, B \Rightarrow}{A \wedge B \Rightarrow}$$

Corresponding semantic conditions:

1. If $v(A) = \mathrm{T}$ and $v(B) = \mathrm{T}$ then $v(A \wedge B) = \mathrm{T}$
2. If $v(A) = \mathrm{F}$ or $v(B) = \mathrm{F}$ then $v(A \wedge B) = \mathrm{F}$

Semantics for Pure Calculi

Example (Sequent Calculus for C_1)

$$\frac{A \Rightarrow}{\Rightarrow \neg A} \qquad \frac{A \Rightarrow}{\neg\neg A \Rightarrow} \qquad \frac{\neg A \Rightarrow \qquad \neg B \Rightarrow}{\neg(A \wedge B) \Rightarrow}$$

Corresponding semantic conditions:

1. If $v(A) = \mathrm{F}$ then $v(\neg A) = \mathrm{T}$
2. If $v(A) = \mathrm{F}$ then $v(\neg\neg A) = \mathrm{F}$
3. If $v(\neg A) = \mathrm{F}$ and $v(\neg B) = \mathrm{F}$ then $v(\neg(A \wedge B)) = \mathrm{F}$

This semantics is non-deterministic.

Semantics for Pure Calculi

Soundness and Completeness [Béziau '01]

s is provable in G

$\Longleftrightarrow$

s is satisfied by every G-legal valuation

Semantics for Pure Calculi

Soundness and Completeness

s is provable in G using $\mathcal{F} \subseteq WFF$

$\Longleftrightarrow$

s is satisfied by every G-legal valuation with domain $\mathcal{F}$

Semantics for Pure Calculi

Soundness and Completeness

s is provable in G using $sub(s)$

$\iff$

s is satisfied by every G-legal valuation with domain $sub(s)$

Semantics for Pure Calculi

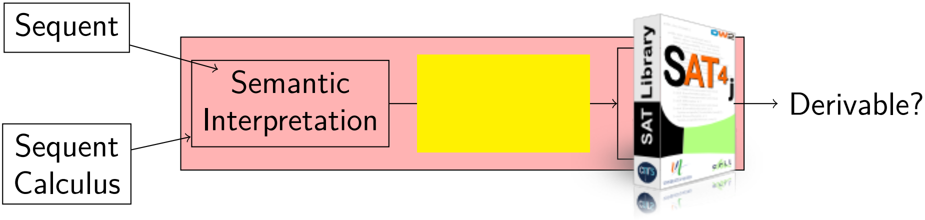
Soundness and Completeness

s is provable in G

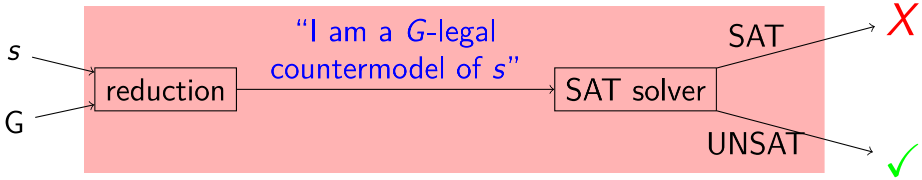
$$\iff$$

s is satisfied by every G-legal valuation with domain $sub(s)$

Tool for Non-classical Logics



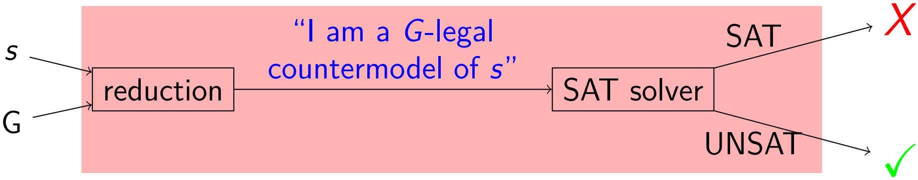
Reduction to SAT



Correctness

s is provable in G iff the generated set of clauses is UNSAT.

Reduction to SAT



Correctness

s is provable in G iff the generated set of clauses is UNSAT.

- In the presence of Next operators, we use Kripke models
- Correctness is more challenging
- Construct a Kripke model from a satisfying assignment

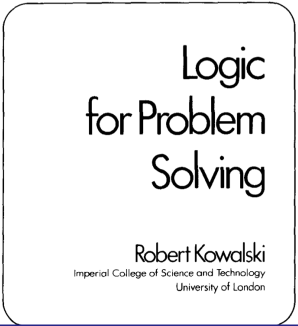
Propositional logic example

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

x & y	names the proposition	x* and y*
x V y		x* or y*
x ⊃ y		if x* then y*
x<=>y		x* if and only if y*
¬ x		it is not the case that x*.

where &, V , ⊃, <=> and ¬ are infix function symbols. Read True(x) as stating that x is true. The following set of clauses cannot be reexpressed as Horn clauses by renaming predicate symbols.

T1	True(x&y) <- True(x), True(y)
T2	True(x) <- True(x&y)
T3	True(y) <- True(x&y)
T4	True(xVy) <- True(x)
T5	True(xVy) <- True(y)
T6	True(x), True(y) <- True(xVy)
T7	True(x⊃y),True(x) <-
T8	True(x⊃y) <- True(y)
T9	True(y) <- True(x), True(x⊃y)
T10	True(x<=>y) <- True(x y), True(y⊃
T11	True(x⊃y) <- True(x<=>y)
T12	True(y⊃x) <- True(x<=>y)
T13	True(¬x), True(x) <-
T14	<- True(¬x), True(x)



Time Complexity

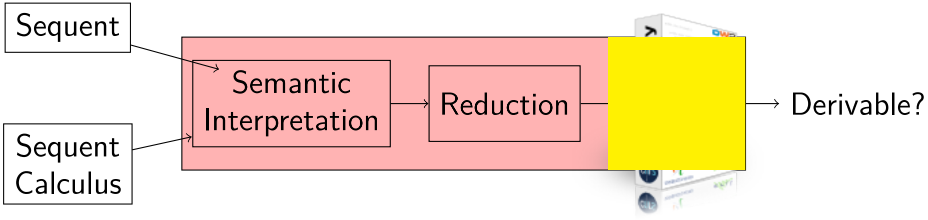
Translating

- Always polynomial
- $O(n^k)$, where:
  - n – formula length
  - k – depends on the calculus
- In all examples: $k = 1$ (linear)

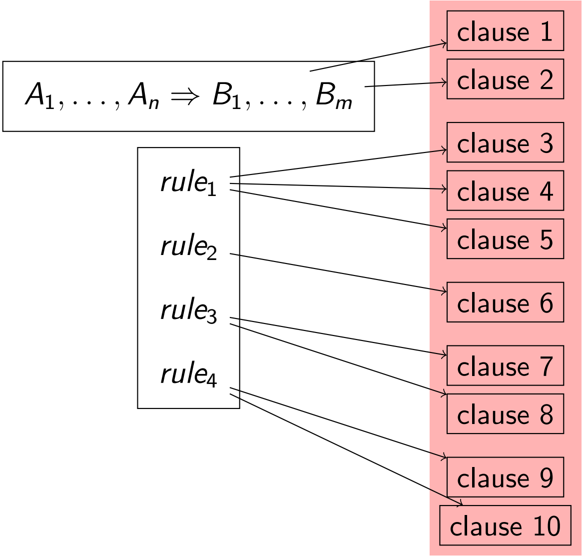
Solving

- Exp in the worst case
- linear with HORNSAT
- "Horn calculi": the generated SAT-instances consist of Horn clauses.
- Example: Primal infon logic

Tool for Non-classical Logics

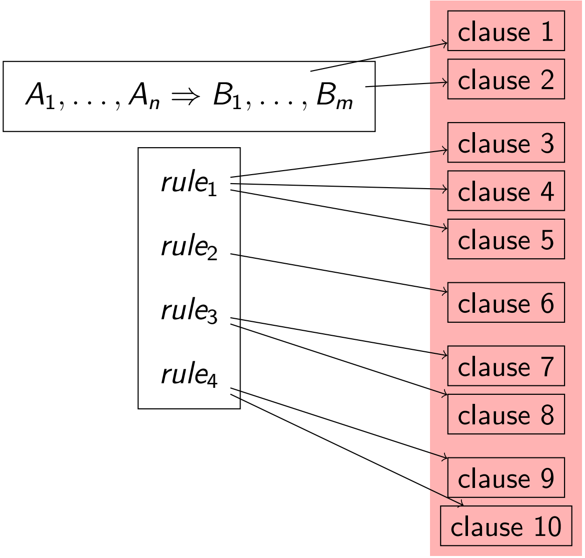


Reduction



$A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m$
$rule_1$
$rule_2$
$rule_3$
$rule_4$
clause 1
clause 2
clause 3
clause 4
clause 5
clause 6
clause 7
clause 8
clause 9
clause 10

Reduction

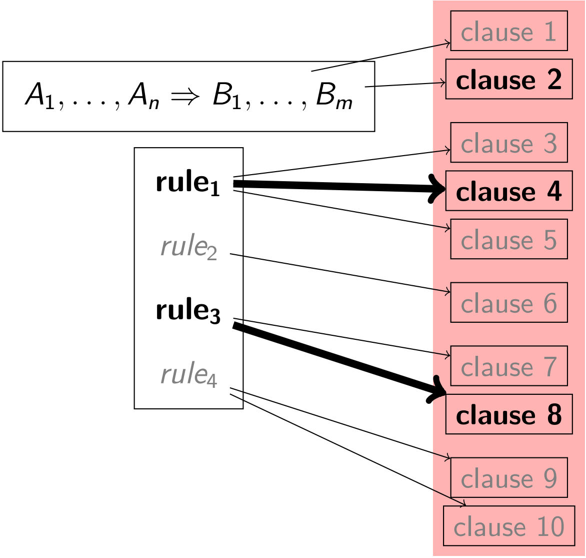


SAT

SAT assignment

$A_1 = \mathit{false}, A_2 = \mathit{true}, \ldots$

Reduction



UNSAT

UNSAT core

$2, 4, 8$

Command-line Interface

```
>cat dolev_yao.txt
```

```
connectives: P:2, E:2
rule: =>a; =>b / =>aPb
rule: a=> / aPb=>
rule: b=> / aPb=>
rule: =>a; =>b / =>aEb
rule: =>b; a=> / aEb=>
analyticity:
inputSequent: (((m1 P m2 ) E k) E k),k=>m1
```

```
>java -jar gen2sat.jar dolev_yao.txt
```

```
provable
There’s a proof that uses only these rules:
[=>b; a=> / a E b=>, a=> / a P b=>]
```

Command-line Interface

```
>cat primal.txt
```

```
connectives: AND:2,IMPLIES:2
nextOperators: q1 said, q2 said, q3 said
rule: =>p1; =>p2 / =>p1 AND p2
rule: p1,p2=> / p1 AND p2=>
rule: =>p2 / =>p1 IMPLIES p2
rule: =>p1; p2=> / p1 IMPLIES p2=>
analyticity:
inputSequent: =>q1said (p IMPLIES p)
```

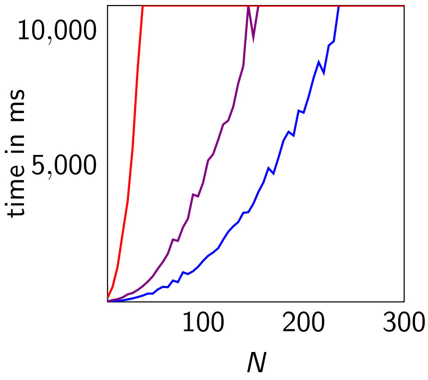
```
>java -jar gen2sat.jar primal.txt
```

```
unprovable
Countermodel:
q1said p=false, q1said(p IMPLIES p)=false
```

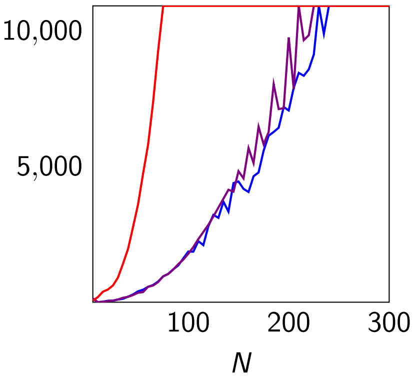
Evaluation: Structured Problems

- Compared running time with another generic prover: MetTeL
- input: $\{\neg\}$-analytic calculus for Łukasiewicz 3-valued logic
- Problems for Łukasiewicz infinite-valued logic [Rothenberg'07]
- Gen2sat$_m$, Gen2sat, MetTeL

Provable

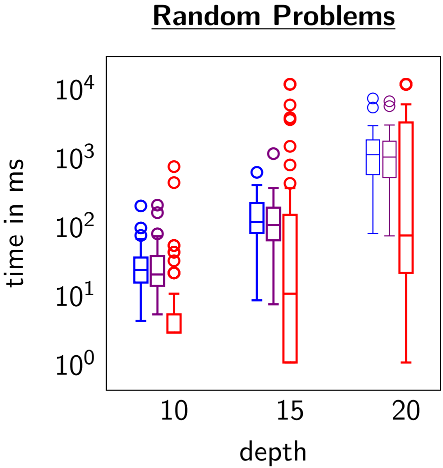


Unprovable



Evaluation: Random Problems

- Compared running time with another generic prover: MetTeL
- input: $\{\neg\}$-analytic calculus for Łukasiewicz 3-valued logic
- Random problems generated by MetTeL
- Gen2sat$_m$, Gen2sat, MetTeL



An Idea: Logic Education

Motivation:

- Gen2sat can be useful for teaching sequent calculi
- The student 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
- 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"
  - ...

Conclusions

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:

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

Conclusions

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:

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

Thank you!