Gen2sat: an Automated Tool for Deciding Derivability in Analytic Pure Sequent Calculi

Yoni Zohar – Tel Aviv University

Anna Zamansky – Haifa University

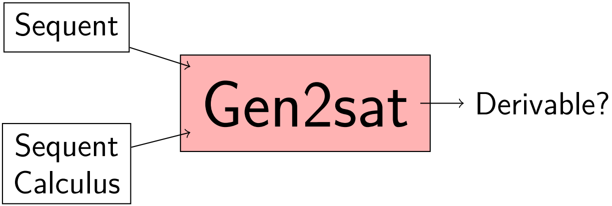
IJCAR 2016

Gen2sat

- Gen2sat is a tool that decides derivability in a wide variety of logics.
- It receives a **Gen**tzen-style calculus and a sequent, and decides whether the sequent is derivable in it.

- Based on a reduction [Lahav, Z. '14] **to sat**.
- Goes through a semantic reading of sequent calculi.
- Written in `Java`, uses `sat4j`.

- Logic engineering: MUltlog, TINC, NESCOND, LoTREC, MetTel, ...

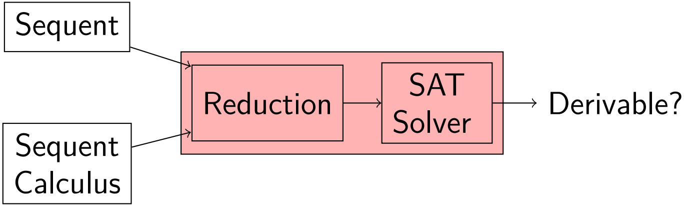


Gen2sat

- Gen2sat is a tool that decides derivability in a wide variety of logics.
- It receives a **Gen**tzen-style calculus and a sequent, and decides whether the sequent is derivable in it.

- Based on a reduction [Lahav, Z. '14] **to sat**.
- Goes through a semantic reading of sequent calculi.
- Written in `Java`, uses `sat4j`.

- Logic engineering: MUltlog, TINC, NESCOND, LoTREC, MetTel, ...

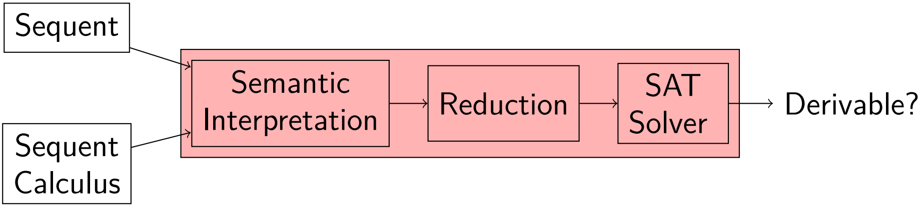


Gen2sat

- Gen2sat is a tool that decides derivability in a wide variety of logics.
- It receives a **Gen**tzen-style calculus and a sequent, and decides whether the sequent is derivable in it.

- Based on a reduction [Lahav, Z. '14] **to sat**.
- Goes through a semantic reading of sequent calculi.
- Written in `Java`, uses `sat4j`.

- Logic engineering: MUltlog, TINC, NESCOND, LoTREC, MetTel, ...

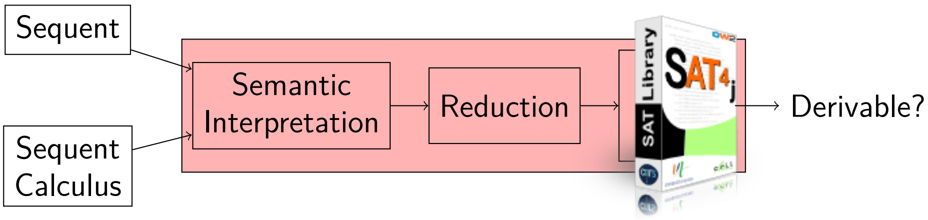


Gen2sat

- Gen2sat is a tool that decides derivability in a wide variety of logics.
- It receives a **Gen**tzen-style calculus and a sequent, and decides whether the sequent is derivable in it.

- Based on a reduction [Lahav, Z. '14] **to sat**.
- Goes through a semantic reading of sequent calculi.
- Written in `Java`, uses `sat4j`.

- Logic engineering: MUltlog, TINC, NESCOND, LoTREC, MetTel, ...



Analytic Pure Sequent Calculi

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

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

- *Pure sequent calculi* are propositional sequent calculi that include all usual structural rules, and a finite set of pure logical rules [Avron '91]:

$$\checkmark \ \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta} \qquad X \ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}$$

Analytic Pure Sequent Calculi

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

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

- *Pure sequent calculi* are propositional sequent calculi that include all usual structural rules, and a finite set of pure logical rules [Avron '91]:

$$\checkmark \ \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta} \qquad X \ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}$$

- A calculus is *analytic* if $\vdash \Gamma \Rightarrow \Delta$ implies that there is a derivation of $\Gamma \Rightarrow \Delta$ using only *sub* $(\Gamma \Rightarrow \Delta)$.

Analytic Pure Sequent Calculi

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

$$A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \quad \leftrightsquigarrow \quad A_1 \wedge \ldots \wedge A_n \supset B_1 \vee \ldots \vee B_m$$

- *Pure sequent calculi* are propositional sequent calculi that include all usual structural rules, and a finite set of pure logical rules [Avron '91]:

$$\checkmark \ \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta} \qquad \text{X} \ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}$$

- A calculus is *⊚-analytic* if $\vdash \Gamma \Rightarrow \Delta$ implies that there is a derivation of $\Gamma \Rightarrow \Delta$ using only $sub^{\circledcirc}(\Gamma \Rightarrow \Delta)$.
- $sub^{\circledcirc}(A) = sub(A) \cup \{\circ B \mid \circ \in \circledcirc, B \in sub(A) \setminus \{A\}\}$.

Analytic Pure Sequent Calculi

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

$$A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \quad \leftrightsquigarrow \quad A_1 \wedge \ldots \wedge A_n \supset B_1 \vee \ldots \vee B_m$$

- *Pure sequent calculi* are propositional sequent calculi that include all usual structural rules, and a finite set of pure logical rules [Avron '91]:

$$\checkmark \ \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \supset B, \Delta} \qquad \text{✗} \ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B}$$

- A calculus is *$\circledcirc$-analytic* if $\vdash \Gamma \Rightarrow \Delta$ implies that there is a derivation of $\Gamma \Rightarrow \Delta$ using only $sub^{\circledcirc}(\Gamma \Rightarrow \Delta)$.
- $sub^{\circledcirc}(A) = sub(A) \cup \{\circ B \mid \circ \in \circledcirc, B \in sub(A) \setminus \{A\}\}$.

- Many logics have pure sequent calculi. Including: classical logic, three-valued logics, four-valued logics, paraconsistent logics, etc.

Examples

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:

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

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

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

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

Examples

Primal Infon Logic [Gurevich,Neeman '09]

- An extremely efficient propositional logic developed in Microsoft.
- One of the main logical engines behind the authorization language DKAL.
- Provides a balance between expressivity and efficiency.

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

$$(\vee \Rightarrow) \quad \textit{none} \qquad (\Rightarrow \vee) \quad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}$$

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

Examples

Łukasiewicz 3-valued Logic [Avron '03]

A $\{\neg\}$-analytic pure calculus for Ł₃ is obtained by augmenting the positive fragment of **LK** with some pure rules. For example:

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

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

Next Operators

- Gen2sat supports analytic pure calculi

Next Operators

- Gen2sat supports $\circledcirc$-analytic pure calculi

Next Operators

- Gen2sat supports $\circledcirc$-analytic pure calculi augmented with impure rules of the form: $\frac{\Gamma \Rightarrow \Delta}{*\Gamma \Rightarrow *\Delta}$

Next Operators

- Gen2sat supports $\circledcirc$-analytic pure calculi augmented with impure rules of the form: $\frac{\Gamma \Rightarrow \Delta}{*\Gamma \Rightarrow *\Delta}$
- Unary modalities that are often employed in temporal logics.

Next Operators

- Gen2sat supports $\circledcirc$-analytic pure calculi augmented with impure rules of the form: $\dfrac{\Gamma \Rightarrow \Delta}{*\Gamma \Rightarrow *\Delta}$
- Unary modalities that are often employed in temporal logics.

$KF/KD!/KDalt1$

X-fragment of $LTL + \Box$ in (multi-)modal logic of functional models

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

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

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

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

$$\dfrac{\Gamma \Rightarrow \Delta}{\Box\Gamma \Rightarrow \Box\Delta}$$

Next Operators

- Gen2sat supports ⊚-analytic pure calculi augmented with impure rules of the form: $\dfrac{\Gamma \Rightarrow \Delta}{*\Gamma \Rightarrow *\Delta}$
- Unary modalities that are often employed in temporal logics.

Primal Infon Logic with Quotations

- "*said*" operators are indispensable for applications.
- Each principle q has an operator "q *said*".

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

$$(\vee \Rightarrow)\quad \textit{none} \qquad (\Rightarrow \vee)\quad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}$$

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

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

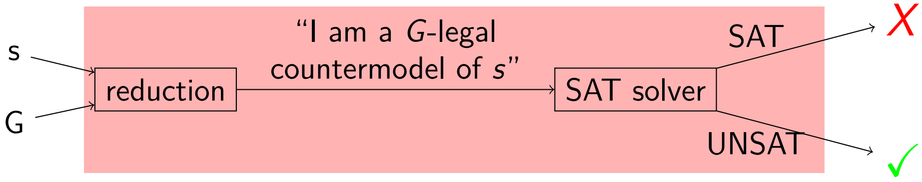
Decision Procedure [Lahav, Z. '14]

Semantics of Pure Sequent Calculi

- Pure calculi induce a 2-valued semantics based on bivaluations.
- Bivaluations satisfying the semantic reading of G are G-legal.
- Example: for Łukasiewicz 3-valued logic, we get a 2-valued semantics!

Theorem

$\vdash_G s \Leftrightarrow \neg\exists$ *G-legal bivaluation with domain* $sub^{\circledcirc}(s)$ *that refutes* s.



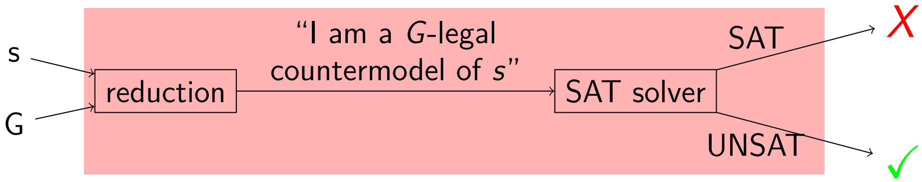
Decision Procedure [Lahav, Z. '14]

Semantics of Pure Sequent Calculi with **Next** Operators

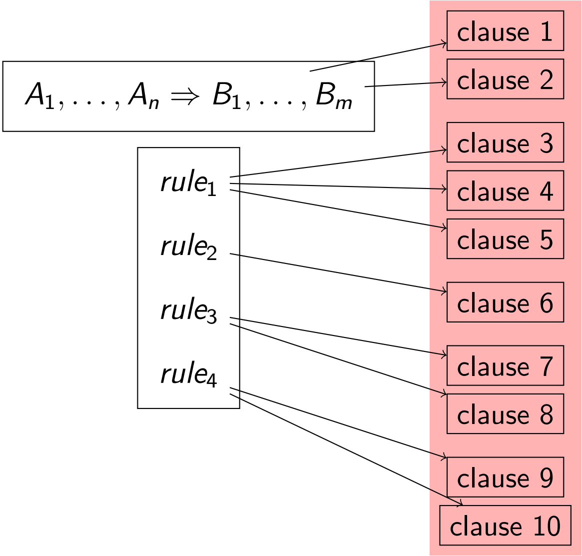
- Pure calculi induce a 2-valued semantics based on functional Kripke models.
- Kripke models satisfying the semantic reading of G are G-legal.
- Example: for Łukasiewicz 3-valued logic, we get a 2-valued semantics!

Theorem

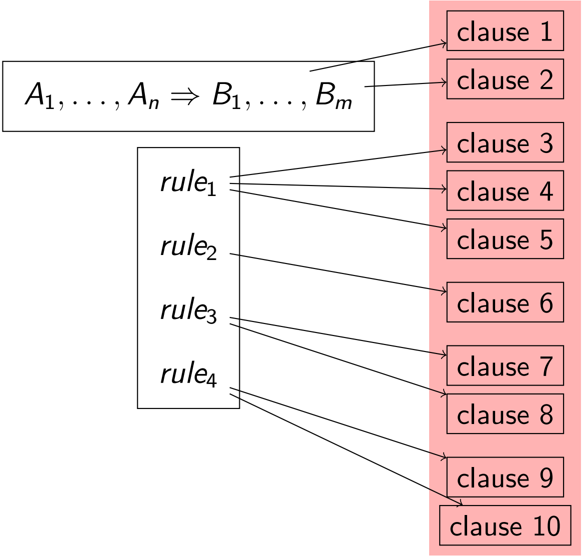
$\vdash_G s \Leftrightarrow \neg\exists$ *G-legal Kripke model with domain* $sub^{\circledcirc}(s)$ *that refutes* s.



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

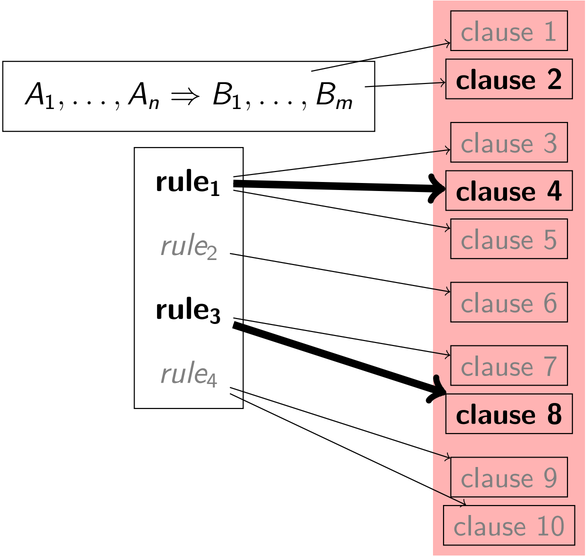


SAT

SAT assignment

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

Reduction

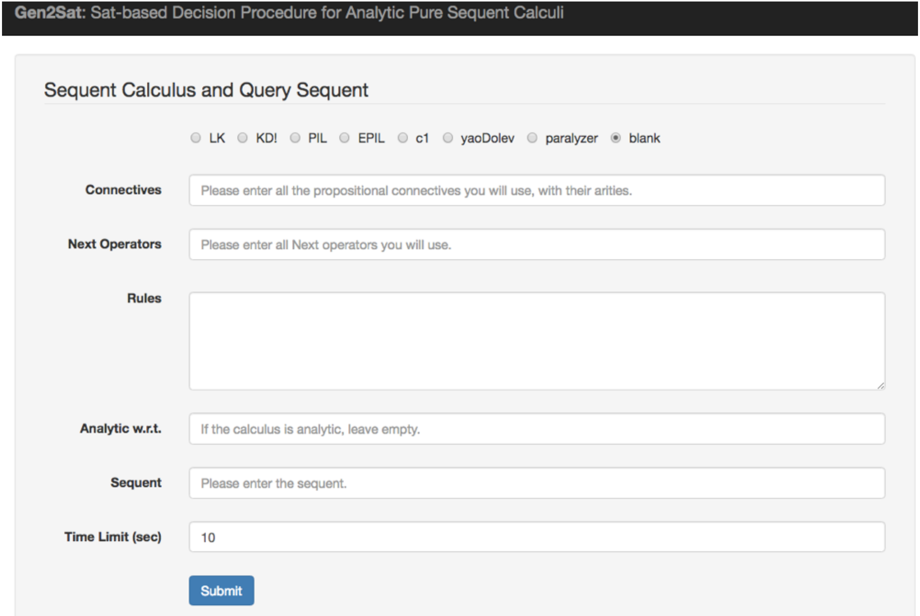


UNSAT

UNSAT core

2, 4, 8

Web Interface



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
```

Performance

- Input: $\{\neg\}$-analytic calculus for the paraconsistent logic C_1 [Avron et al., '12].
- Compared running times with the tableau prover KEMS for this logic.
- Gen2sat$_m$: Does not ask for an UNSAT core.

Derivable Sequents (running times in ms)

size	KEMS	Gen2sat	Gen2sat$_m$	#vars	#clauses
10	133	342	213	137	344
20	675	252	73	277	694
50	13934	747	143	697	1744
80	75578	1393	148	1117	2794
100	175716	2178	235	1397	3494

- Gen2sat performs better than KEMS on larger sequents, but not on the smaller ones.
- Underivable sequents are easier for Gen2sat than derivable ones.
- The opposite holds for KEMS.
- Gen2sat$_m$ performs better than Gen2sat on derivable sequents.

Performance

- Input: $\{\neg\}$-analytic calculus for the paraconsistent logic C_1 [Avron et al., '12].
- Compared running times with the tableau prover KEMS for this logic.
- Gen2sat$_m$: Does not ask for an UNSAT core.

Underivable Sequents (running times in ms)

size	KEMS	Gen2sat	Gen2sat$_m$	#vars	#clauses
10	153	224	215	135	339
20	686	70	70	275	689
50	14247	146	159	695	1739
80	78212	175	203	1115	2789
100	182904	226	284	1395	3489

- Gen2sat performs better than KEMS on larger sequents, but not on the smaller ones.
- Underivable sequents are easier for Gen2sat than derivable ones.
- The opposite holds for KEMS.
- Gen2sat$_m$ performs better than Gen2sat on derivable sequents.

Educational Pilot

Background:

- Purpose: increasing students' engagement and motivation.
- Gen2sat is a natural candidate for such a task, as it leaves all heuristic considerations to the SAT-solver.
- The assignment was to present a minimal test plan with maximal coverage, as well as finding (intensionally planted) bugs.

Preliminary results:

- 13 students participated, all got 70%-85% coverage.
- Some used 0-ary and 3-ary connectives.
- The bugs were found by some of the students.
- Feedback:
  - "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
- Command-line and web interfaces

Future work:

- Automatically detect analyticity (when possible)
- Integrate with a theorem prover
- Develop an educational version

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
- Command-line and web interfaces

Future work:

- Automatically detect analyticity (when possible)
- Integrate with a theorem prover
- Develop an educational version

Thank you!