A proof-theoretic perspective on SMT-solving for intuitionistic propositional logic

Camillo Fiorentini, Rajeev Goré and Stéphane Graham-Lengrand

TABLEAUX'19, 4th September 2019

A calculus rediscovered at least 6 times (in various forms)

(Sequent) Calculus for Intuitionistic Propositional Logic

- Vorob'ev in the 50s
- Hudelmaier (88)
- Dyckhoff (90)
- Paulson (91)
- Lincoln-Scedrov-Shankar (91) (with a linear logic approach)

A calculus rediscovered at least 6 times (in various forms)

(Sequent) Calculus for Intuitionistic Propositional Logic

- Vorob'ev in the 50s
- Hudelmaier (88)
- Dyckhoff (90)
- Paulson (91)
- Lincoln-Scedrov-Shankar (91) (with a linear logic approach)

Called

- LJT by Hudelmaier and then Dyckhoff (T for "Terminating", nothing to do with LJT from the linear logic tradition),
- G4ip by Troelstra-Schwichtenberg.
- "Contraction-free sequent calculus"
- "(Hudelmaier's) Depth-bounded sequent calculus"

A calculus rediscovered at least 6 times (in various forms)

(Sequent) Calculus for Intuitionistic Propositional Logic

- Vorob'ev in the 50s
- Hudelmaier (88)
- Dyckhoff (90)
- Paulson (91)
- Lincoln-Scedrov-Shankar (91) (with a linear logic approach)

Called

- LJT by Hudelmaier and then Dyckhoff (T for "Terminating", nothing to do with LJT from the linear logic tradition),
- G4ip by Troelstra-Schwichtenberg.
- "Contraction-free sequent calculus"
- "(Hudelmaier's) Depth-bounded sequent calculus"

Each time, the calculus comes up with slight variations

THE JOURNAL OF SYMBOLIC LOGIC
Volume 57, Number 3, Sept. 1992

CONTRACTION-FREE SEQUENT CALCULI FOR INTUITIONISTIC LOGIC

ROY DYCKHOFF

§0. Prologue. Gentzen's sequent calculus LJ, and its variants such as G3 [21], are (as is well known) convenient as a basis for automating proof search for IPC (intuitionistic propositional calculus). But a problem arises: that of detecting loops, arising from the use (in reverse) of the rule $\supset \Rightarrow$ for implication introduction on the left. We describe below an equivalent calculus, yet another variant on these systems, where the problem no longer arises: this gives a simple but effective decision procedure for IPC.

The underlying method can be traced back forty years to Vorob′ev [33], [34]. It has been rediscovered recently by several authors (the present author in August 1990, Hudelmaier [18], [19], Paulson [27], and Lincoln et al. [23]). Since the main idea is not plainly apparent in Vorob′ev's work, and there are mathematical applications [28], it is desirable to have a simple proof. We present such a proof, exploiting the Dershowitz-Manna theorem [4] on multiset orderings.

§1. Introduction. Consider the task of constructing proofs in Gentzen's sequent calculus LJ of intuitionistic sequents $\Gamma \Rightarrow G$, where Γ is a set of assumption

THE JOURNAL OF SYMBOLIC LOGIC
Volume 57, Number 3, Sept. 1992

CONTRACTION-FREE SEQUENT CALCULI FOR INTUITIONISTIC LOGIC

ROY DYCKHOFF

§0. Prologue. Gentzen's sequent calculus LJ, and its variants such as G3 [21], are (as is well known) convenient as a basis for automating proof search for IPC (intuitionistic propositional calculus). But a problem arises: that of detecting loops, arising from the use (in reverse) of the rule $\supset\Rightarrow$ for implication introduction on the left. We describe below an equivalent calculus, yet another variant on these systems, where the problem no longer arises: this gives a simple but effective decision procedure for IPC.

The underlying method can be traced back forty years to Vorob′ev [33], [34]. It has been rediscovered recently by several authors (the present author in August 1990, Hudelmaier [18], [19], Paulson [27], and Lincoln et al. [23]). Since the main idea is not plainly apparent in Vorob′ev's work, and there are mathematical applications [28], it is desirable to have a simple proof. We present such a proof, exploiting the Dershowitz-Manna theorem [4] on multiset orderings.

§1. Introduction. Consider the task of constructing proofs in Gentzen's sequent calculus LJ of intuitionistic sequents $\Gamma \Rightarrow G$, where Γ is a set of assumption

SAT Modulo Intuitionistic Implications

Koen Claessen(✉) and Dan Rosén(✉)

Chalmers University of Technology, Gothenburg, Sweden
{koen,danr}@chalmers.se

Abstract. We present a new method for solving problems in intuitionistic propositional logic, which involves the use of an incremental SAT-solver. The method scales to very large problems, and fits well into an SMT-based framework for interaction with other theories.

1 Introduction

Let us take a look at *intuitionistic propositional logic*. Its syntax looks just like classical propositional logic:

$$\begin{array}{llr} A ::= & a \mid b \mid c \mid .. \mid q & \text{-- atoms} \\ \mid & A_1 \wedge A_2 & \text{-- conjunction} \\ \mid & A_1 \vee A_2 & \text{-- disjunction} \\ \mid & A_1 \rightarrow A_2 & \text{-- implication} \\ \mid & \bot \mid \top & \text{-- false/true} \end{array}$$

However, its definition of truth is considerably weaker than for classical logic. In Fig. 1, we show a Hilbert-style proof system for intuitionistic propositional

Intuitionistic Sequent Calculus

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{(C' \Rightarrow C), \Gamma \vdash C' \quad C, \Gamma \vdash D}{(C' \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

Intuitionistic Sequent Calculus

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{(C' \Rightarrow C), \Gamma \vdash C' \quad C, \Gamma \vdash D}{(C' \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

Variant: atom a can be generalised as formula A (still sound)

Intuitionistic Sequent Calculus

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{(C' \Rightarrow C), \Gamma \vdash C' \quad C, \Gamma \vdash D}{(C' \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

Variant: atom a can be generalised as formula A (still sound)

In (at least one version of) LK, applying rules bottom-up removes at least one connective (comparing a given premiss to the conclusion)
Makes root-first proof-search terminating.

Intuitionistic Sequent Calculus

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{(C' \Rightarrow C), \Gamma \vdash C' \quad C, \Gamma \vdash D}{(C' \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

Variant: atom a can be generalised as formula A (still sound)

In (at least one version of) LK, applying rules bottom-up removes at least one connective (comparing a given premiss to the conclusion)
Makes root-first proof-search terminating.

In the above version of LJ, this is not true

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{?}{(C' \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \vee B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \vee B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{C, a, \Gamma \vdash D}{(a \Rightarrow C), a, \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \vee B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{C, a, \Gamma \vdash D}{(a \Rightarrow C), a, \Gamma \vdash D} \qquad \frac{A, (B \Rightarrow C), \Gamma \vdash B \quad C, \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \vee B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{C, a, \Gamma \vdash D}{(a \Rightarrow C), a, \Gamma \vdash D} \qquad \frac{A, (B \Rightarrow C), \Gamma \vdash B \quad C, \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

- Obviously sound

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \vee B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{C, a, \Gamma \vdash D}{(a \Rightarrow C), a, \Gamma \vdash D} \qquad \frac{A, (B \Rightarrow C), \Gamma \vdash B \quad C, \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

- Obviously sound
- Complete?

G4ip

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\bot, \Gamma \vdash C} \qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \qquad \frac{A, \Gamma \vdash C \quad B, \Gamma \vdash C}{(A \vee B), \Gamma \vdash C}$$

$$\frac{\Gamma \vdash D}{(\bot \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow (B \Rightarrow C)), \Gamma \vdash D}{((A \wedge B) \Rightarrow C), \Gamma \vdash D} \qquad \frac{(A \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \vee B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{C, a, \Gamma \vdash D}{(a \Rightarrow C), a, \Gamma \vdash D} \qquad \frac{A, (B \Rightarrow C), \Gamma \vdash B \quad C, \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

$$\frac{}{a, \Gamma \vdash a}$$

- Obviously sound
- Complete? rule permutation argument (Dyckhoff'92,'17), cut-elimination (Dyckhoff-Negri'00, Dyckhoff-SGL-Kesner'06)
- Connection with focused seq. calculus LJQ (Dyckhoff-SGL'06)

Good properties

- In each rule, each premiss is "smaller" than the conclusion (for the multiset order on the formulae present in the sequent)
  $\Rightarrow$ The height (aka depth) of proof-trees (for sequent $\Gamma \vdash A$) is bounded: it is a "depth-bounded sequent calculus".
  $\Rightarrow$ "Root-first proof-search" (Roy's preferred terminology) terminates and constitutes decision procedure for provability of IPL (only finitely many trees of height $\leq$ bound)
- Each rule is invertible (if the conclusion is provable then so are the premisses), except (the $\vee$-right rules and)

$$\frac{A, (B \Rightarrow C), \Gamma \vdash B \quad C, \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

  which is *semi-invertible*: if the conclusion is provable, then so is the right premiss (the left premiss can be considered the side-condition of an invertible 1-premiss rule)

A generalised version of the semi-invertible rule

$$\frac{A, (B \Rightarrow C), \Gamma \vdash B \qquad\qquad C \;, \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Still sound.

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad (a_1 \wedge \cdots \wedge a_n {\Rightarrow} C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Still sound. Derivable with a cut:

$$\dfrac{\dfrac{\dfrac{a_1, \ldots, a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad \dfrac{}{C, a_1, \ldots, a_n, \Gamma \vdash C}}{a_1, \ldots, a_n, ((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash C}}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash a_1 \wedge \cdots \wedge a_n {\Rightarrow} C} \qquad (a_1 \wedge \cdots \wedge a_n {\Rightarrow} C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Still sound. Derivable with a cut:

$$\cfrac{\cfrac{\cfrac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad \cfrac{}{C, a_1, \ldots, a_n, \Gamma \vdash C}}{a_1, \ldots, a_n, ((A \Rightarrow B) \Rightarrow C), \Gamma \vdash C}}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash a_1 \wedge \cdots \wedge a_n \Rightarrow C} \qquad (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remarks:

- If $n \neq 0$, rule is not necessarily semi-invertible.

A generalised version of the semi-invertible rule

$$\frac{a_1,\ldots,a_n, A,(B\Rightarrow C),\Gamma\vdash B \qquad ((A\Rightarrow B)\Rightarrow C),(a_1\wedge\cdots\wedge a_n\Rightarrow C),\Gamma\vdash D}{((A\Rightarrow B)\Rightarrow C),\Gamma\vdash D}$$

Still sound. Derivable with a cut:

$$\dfrac{\dfrac{\dfrac{a_1,\ldots,a_n, A,(B\Rightarrow C),\Gamma\vdash B \qquad \dfrac{}{C,a_1,\ldots,a_n,\Gamma\vdash C}}{a_1,\ldots,a_n,((A\Rightarrow B)\Rightarrow C),\Gamma\vdash C}}{((A\Rightarrow B)\Rightarrow C),\Gamma\vdash a_1\wedge\cdots\wedge a_n\Rightarrow C} \qquad ((A\Rightarrow B)\Rightarrow C),(a_1\wedge\cdots\wedge a_n\Rightarrow C),\Gamma\vdash D}{((A\Rightarrow B)\Rightarrow C),\Gamma\vdash D}$$

Remarks:

- If $n\neq 0$, rule is not necessarily semi-invertible. Fixed.

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Still sound. Derivable with a cut:

$$\dfrac{\dfrac{\dfrac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad \overline{C, a_1, \ldots, a_n, \Gamma \vdash C}}{a_1, \ldots, a_n, ((A \Rightarrow B) \Rightarrow C), \Gamma \vdash C}}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash a_1 \wedge \cdots \wedge a_n \Rightarrow C} \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remarks:

- If $n \neq 0$, rule is not necessarily semi-invertible. Fixed.
- More problematic: with or without the fix, weight of premisses not necessarily smaller than weight of conclusion. Depth-boundedness is probably lost.

A generalised version of the semi-invertible rule

$$\frac{a_1,\ldots,a_n, A, (B\Rightarrow C), \Gamma \vdash B \qquad ((A\Rightarrow B)\Rightarrow C), (a_1\wedge\cdots\wedge a_n\Rightarrow C), \Gamma \vdash D}{((A\Rightarrow B)\Rightarrow C), \Gamma \vdash D}$$

Still sound. Derivable with a cut:

$$\dfrac{\dfrac{\dfrac{a_1,\ldots,a_n, A, (B\Rightarrow C), \Gamma \vdash B \qquad \dfrac{}{C, a_1,\ldots,a_n, \Gamma \vdash C}}{a_1,\ldots,a_n, ((A\Rightarrow B)\Rightarrow C), \Gamma \vdash C}}{((A\Rightarrow B)\Rightarrow C), \Gamma \vdash a_1\wedge\cdots\wedge a_n\Rightarrow C} \qquad ((A\Rightarrow B)\Rightarrow C), (a_1\wedge\cdots\wedge a_n\Rightarrow C), \Gamma \vdash D}{((A\Rightarrow B)\Rightarrow C), \Gamma \vdash D}$$

Remarks:

- If $n \neq 0$, rule is not necessarily semi-invertible. Fixed.
- More problematic: with or without the fix, weight of premisses not necessarily smaller than weight of conclusion. Depth-boundedness is probably lost.
- Is it not a bad idea to use the cut-rule in proof-search?
  How do we come up with $a_1,\ldots,a_n$?

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1 \wedge \cdots \wedge a_n {\Rightarrow} C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Still sound. Derivable with a cut:

$$\dfrac{\dfrac{\dfrac{a_1, \ldots, a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad \dfrac{}{C, a_1, \ldots, a_n, \Gamma \vdash C}}{a_1, \ldots, a_n, ((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash C}}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash a_1 \wedge \cdots \wedge a_n {\Rightarrow} C} \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1 \wedge \cdots \wedge a_n {\Rightarrow} C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Remarks:

- If $n \neq 0$, rule is not necessarily semi-invertible. Fixed.
- More problematic: with or without the fix, weight of premisses not necessarily smaller than weight of conclusion. Depth-boundedness is probably lost.
- Is it not a bad idea to use the cut-rule in proof-search? How do we come up with $a_1, \ldots, a_n$?

In conclusion: this generalisation sounds like a terrible idea.

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1 \wedge \cdots \wedge a_n{\Rightarrow}C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Recovering termination:

- Let's impose (1) that $(a_1 \wedge \cdots \wedge a_n{\Rightarrow}C) \notin \Gamma$, otherwise the right premiss is identical/equivalent to the conclusion.

A generalised version of the semi-invertible rule

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Recovering termination:

- Let's impose (1) that $(a_1 \wedge \cdots \wedge a_n \Rightarrow C) \notin \Gamma$, otherwise the right premiss is identical/equivalent to the conclusion.
- Using cuts in root-first proof-search is cumbersome unless we have a magic trick to produce the cut-formula (here: the $a_1, \ldots, a_n$)

A generalised version of the semi-invertible rule

$$\frac{a_1,\ldots,a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1{\wedge}\cdots{\wedge}a_n{\Rightarrow}C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Recovering termination:

- Let's impose (1) that $(a_1{\wedge}\cdots{\wedge}a_n{\Rightarrow}C) \notin \Gamma$, otherwise the right premiss is identical/equivalent to the conclusion.
- Using cuts in root-first proof-search is cumbersome unless we have a magic trick to produce the cut-formula (here: the $a_1,\ldots,a_n$)
- Let's impose (2) that $a_1,\ldots,a_n$ are atoms present in the conclusion.

A generalised version of the semi-invertible rule

$$\frac{a_1,\ldots,a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1\wedge\cdots\wedge a_n{\Rightarrow}C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Recovering termination:

- Let's impose (1) that $(a_1\wedge\cdots\wedge a_n{\Rightarrow}C) \notin \Gamma$, otherwise the right premiss is identical/equivalent to the conclusion.
- Using cuts in root-first proof-search is cumbersome unless we have a magic trick to produce the cut-formula (here: the $a_1,\ldots,a_n$)
- Let's impose (2) that $a_1,\ldots,a_n$ are atoms present in the conclusion.

(1) and (2) recover termination (& preserve completeness).

A generalised version of the semi-invertible rule

$$\frac{a_1,\ldots,a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1{\wedge}\cdots{\wedge}a_n{\Rightarrow}C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Recovering termination:

- Let's impose (1) that $(a_1{\wedge}\cdots{\wedge}a_n{\Rightarrow}C) \notin \Gamma$, otherwise the right premiss is identical/equivalent to the conclusion.
- Using cuts in root-first proof-search is cumbersome unless we have a magic trick to produce the cut-formula (here: the $a_1,\ldots,a_n$)
- Let's impose (2) that $a_1,\ldots,a_n$ are atoms present in the conclusion.

(1) and (2) recover termination (& preserve completeness).
Still, a lot of choices for $\{a_1,\ldots,a_n\}$

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Clearly, this rule is the most complex of the calculus, it branches and is only semi-invertible.

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Clearly, this rule is the most complex of the calculus, it branches and is only semi-invertible.
$\Rightarrow$ We probably want to apply it as a last resort, leaving formulae in Γ of the form $((A \Rightarrow B) \Rightarrow C)$ ignored for as long as we can.

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B{\Rightarrow}C), \Gamma \vdash B \qquad ((A{\Rightarrow}B){\Rightarrow}C), (a_1 \wedge \cdots \wedge a_n{\Rightarrow}C), \Gamma \vdash D}{((A{\Rightarrow}B){\Rightarrow}C), \Gamma \vdash D}$$

Clearly, this rule is the most complex of the calculus, it branches and is only semi-invertible.
$\Rightarrow$ We probably want to apply it as a last resort, leaving formulae in Γ of the form $((A{\Rightarrow}B){\Rightarrow}C)$ ignored for as long as we can.

Apply the other rules eagerly, trying to prove a sequent $\Gamma \vdash d$ while ignoring the formulae in Γ of the form $((A{\Rightarrow}B){\Rightarrow}C)$.

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Clearly, this rule is the most complex of the calculus, it branches and is only semi-invertible.
$\Rightarrow$ We probably want to apply it as a last resort, leaving formulae in Γ of the form $((A \Rightarrow B) \Rightarrow C)$ ignored for as long as we can.

Apply the other rules eagerly, trying to prove a sequent $\Gamma \vdash d$ while ignoring the formulae in Γ of the form $((A \Rightarrow B) \Rightarrow C)$.

Who would do a better job at doing that?

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Clearly, this rule is the most complex of the calculus, it branches and is only semi-invertible.
$\Rightarrow$ We probably want to apply it as a last resort, leaving formulae in Γ of the form $((A \Rightarrow B) \Rightarrow C)$ ignored for as long as we can.

Apply the other rules eagerly, trying to prove a sequent $\Gamma \vdash d$ while ignoring the formulae in Γ of the form $((A \Rightarrow B) \Rightarrow C)$.

Who would do a better job at doing that?
. . . a SAT-solver!

SAT-solvers are perfectly fine intuitionistic provers!

If they conclude that a bunch of clauses $C_1, \ldots, C_n$ is unsat,
they have established intuitionistic provability of $C_1, \ldots, C_n \vdash \bot$.

SAT-solvers are perfectly fine intuitionistic provers!

If they conclude that a bunch of clauses $C_1, \ldots, C_n$ is unsat, they have established intuitionistic provability of $C_1, \ldots, C_n \vdash \bot$.
...well, if each clause $\{\overline{a_1}, \ldots, \overline{a_n}, b_1, \ldots, b_m\}$ is read as

$$(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$$

instead of $\neg a_1 \vee \cdots \vee \neg a_n \vee b_1 \vee \cdots \vee b_m$.

SAT-solvers are perfectly fine intuitionistic provers!

If they conclude that a bunch of clauses $C_1, \ldots, C_n$ is unsat,
they have established intuitionistic provability of $C_1, \ldots, C_n \vdash \bot$.
…well, if each clause $\{\overline{a_1}, \ldots, \overline{a_n}, b_1, \ldots, b_m\}$ is read as

$$(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$$

instead of $\neg a_1 \vee \cdots \vee \neg a_n \vee b_1 \vee \cdots \vee b_m$.
Indeed, SAT-solver would (implicitly or explicitly) produce a resolution proof of $C_1, \ldots, C_n \vdash \bot$

SAT-solvers are perfectly fine intuitionistic provers!

If they conclude that a bunch of clauses $C_1, \ldots, C_n$ is unsat,
they have established intuitionistic provability of $C_1, \ldots, C_n \vdash \bot$.
...well, if each clause $\{\overline{a_1}, \ldots, \overline{a_n}, b_1, \ldots, b_m\}$ is read as

$$(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$$

instead of $\neg a_1 \vee \cdots \vee \neg a_n \vee b_1 \vee \cdots \vee b_m$.
Indeed, SAT-solver would (implicitly or explicitly) produce a resolution proof of $C_1, \ldots, C_n \vdash \bot$
Resolution rule

$$\frac{C \vee x \quad C' \vee \neg x}{C \vee C'} \text{ should be read as } \frac{A \Rightarrow (B \vee x) \quad (A' \wedge x) \Rightarrow B'}{(A \wedge A') \Rightarrow (B \vee B')}$$

which is perfectly sound in intuitionistic logic.

SAT-solvers are perfectly fine intuitionistic provers!

If they conclude that a bunch of clauses $C_1, \dots, C_n$ is unsat,
they have established intuitionistic provability of $C_1, \dots, C_n \vdash \bot$.
…well, if each clause $\{\overline{a_1}, \dots, \overline{a_n}, b_1, \dots, b_m\}$ is read as

$$(a_1 \wedge \dots \wedge a_n) \Rightarrow (b_1 \vee \dots \vee b_m)$$

instead of $\neg a_1 \vee \dots \vee \neg a_n \vee b_1 \vee \dots \vee b_m$.
Indeed, SAT-solver would (implicitly or explicitly) produce a resolution proof of $C_1, \dots, C_n \vdash \bot$
Resolution rule

$$\frac{C \vee x \quad C' \vee \neg x}{C \vee C'} \text{ should be read as } \frac{A \Rightarrow (B \vee x) \quad (A' \wedge x) \Rightarrow B'}{(A \wedge A') \Rightarrow (B \vee B')}$$

which is perfectly sound in intuitionistic logic.
Even better: if they conclude that $C_1, \dots, C_n, \neg d$ is unsat, they have established intuitionistic provability of $C_1, \dots, C_n \vdash d$.

SAT-solvers are perfectly fine intuitionistic provers!

If they conclude that a bunch of clauses $C_1, \ldots, C_n$ is unsat,
they have established intuitionistic provability of $C_1, \ldots, C_n \vdash \bot$.
…well, if each clause $\{\overline{a_1}, \ldots, \overline{a_n}, b_1, \ldots, b_m\}$ is read as

$$(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$$

instead of $\neg a_1 \vee \cdots \vee \neg a_n \vee b_1 \vee \cdots \vee b_m$.
Indeed, SAT-solver would (implicitly or explicitly) produce a resolution proof of $C_1, \ldots, C_n \vdash \bot$
Resolution rule

$$\frac{C \vee x \quad C' \vee \neg x}{C \vee C'} \text{ should be read as } \frac{A \Rightarrow (B \vee x) \quad (A' \wedge x) \Rightarrow B'}{(A \wedge A') \Rightarrow (B \vee B')}$$

which is perfectly sound in intuitionistic logic.
Even better: if they conclude that $C_1, \ldots, C_n, \neg d$ is unsat, they have established intuitionistic provability of $C_1, \ldots, C_n \vdash d$.
Conclusion: they are very good intuitionistic provers
…but are limited to proving sequents of that form.

Preprocessing

It's the preprocess that implements "every formula F can be transformed into an equisatisfiable* CNF $C_1 \wedge \cdots \wedge C_n$"
that uses classical reasoning.

*: $F \vdash \bot$ iff $C_1 \wedge \cdots \wedge C_n \vdash \bot$

Preprocessing

It's the preprocess that implements "every formula F can be transformed into an equisatisfiable* CNF $C_1 \wedge \cdots \wedge C_n$"
that uses classical reasoning. *: $F \vdash \bot$ iff $C_1 \wedge \cdots \wedge C_n \vdash \bot$
In the intuitionistic case, every formula F can be transformed into an (intuitionistically) equiprovable sequent $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$ with

- d an atom
- Γ_{flat} made of *flat clauses*: $(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$
- Γ_{imp} made of *implication clauses*: $((a \Rightarrow b) \Rightarrow c)$

Idea for proof-search:

- flat clauses are treated eagerly,
  to see if, by chance, $\Gamma_{\text{flat}} \vdash d$ is provable,
  using e.g., a SAT-solver.
- implication clauses treated lazily,
  using the (generalised) G4ip rule.

Roy Dyckhoff's 1992 paper

§9. Related work. Vorob′ev [33], [34] described a decision algorithm for IPC based on similar considerations. The present article may be regarded in part as a restatement of this relatively ancient Soviet work: it is offered however as a clarification and simplification, in the knowledge that the technique is now being reinvented and exploited. The sequent calculus lying behind Vorob′ev's algorithm in [34] is concealed by the pre-processing of sequents into a normal form (using the distributive laws); his algorithm also takes advantage of the equivalence (for negated goals) of the intuitionistic decision problem with the classical one. See [25] for a summary of some of the related Soviet work.

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to “name” subformulae of the sequent to prove

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to “name” subformulae of the sequent to prove

- For $A \wedge B$, introduce c with: $c \Rightarrow A, c \Rightarrow B, (A \wedge B) \Rightarrow c$, recursively introduce names for A and for B to get flat clauses

$$c \Rightarrow a, c \Rightarrow b, (a \wedge b) \Rightarrow c$$

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to “name” subformulae of the sequent to prove

- For $A \wedge B$, introduce c with: $c \Rightarrow A, c \Rightarrow B, (A \wedge B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses

$$c \Rightarrow a, c \Rightarrow b, (a \wedge b) \Rightarrow c$$

- For $A \vee B$, introduce c with: $c \Rightarrow (A \vee B), A \Rightarrow c, B \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses

$$c \Rightarrow (a \vee b), a \Rightarrow c, b \Rightarrow c$$

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to "name" subformulae of the sequent to prove

- For $A \wedge B$, introduce c with: $c \Rightarrow A$, $c \Rightarrow B$, $(A \wedge B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses
  $$c \Rightarrow a,\ c \Rightarrow b,\ (a \wedge b) \Rightarrow c$$
- For $A \vee B$, introduce c with: $c \Rightarrow (A \vee B)$, $A \Rightarrow c$, $B \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses
  $$c \Rightarrow (a \vee b),\ a \Rightarrow c,\ b \Rightarrow c$$
- For $A \Rightarrow B$, introduce c with: $(c \wedge A) \Rightarrow B$, $(A \Rightarrow B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clause $(c \wedge a) \Rightarrow b$
  ... and implication clause $(a \Rightarrow b) \Rightarrow c$

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to “name” subformulae of the sequent to prove

- For $A \wedge B$, introduce c with: $c \Rightarrow A, c \Rightarrow B, (A \wedge B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses

  $$c \Rightarrow a, c \Rightarrow b, (a \wedge b) \Rightarrow c$$

- For $A \vee B$, introduce c with: $c \Rightarrow (A \vee B), A \Rightarrow c, B \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses

  $$c \Rightarrow (a \vee b), a \Rightarrow c, b \Rightarrow c$$

- For $A \Rightarrow B$, introduce c with: $(c \wedge A) \Rightarrow B, (A \Rightarrow B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clause $(c \wedge a) \Rightarrow b$
  …and implication clause $(a \Rightarrow b) \Rightarrow c$

Note 1: transformation only preserves satisfiability/provability

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to “name” subformulae of the sequent to prove

- For $A \wedge B$, introduce c with: $c \Rightarrow A$, $c \Rightarrow B$, $(A \wedge B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses

$$c \Rightarrow a,\ c \Rightarrow b,\ (a \wedge b) \Rightarrow c$$

- For $A \vee B$, introduce c with: $c \Rightarrow (A \vee B)$, $A \Rightarrow c$, $B \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses

$$c \Rightarrow (a \vee b),\ a \Rightarrow c,\ b \Rightarrow c$$

- For $A \Rightarrow B$, introduce c with: $(c \wedge A) \Rightarrow B$, $(A \Rightarrow B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clause $(c \wedge a) \Rightarrow b$
  . . . and implication clause $(a \Rightarrow b) \Rightarrow c$

Note 1: transformation only preserves satisfiability/provability
Note 2: Introducing names can be done more sparingly using some of the other G4ip rules for normalisation
(basically, the invertible non-branching ones)

Preprocessing: how?

Like Tseitin transformation to turn any formula into an equisatisfiable CNF.
Introduce atoms to "name" subformulae of the sequent to prove

- For $A \wedge B$, introduce c with: $c \Rightarrow A$, $c \Rightarrow B$, $(A \wedge B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses
  $$c \Rightarrow a, c \Rightarrow b, (a \wedge b) \Rightarrow c$$
- For $A \vee B$, introduce c with: $c \Rightarrow (A \vee B)$, $A \Rightarrow c$, $B \Rightarrow c$,
  recursively introduce names for A and for B to get flat clauses
  $$c \Rightarrow (a \vee b), a \Rightarrow c, b \Rightarrow c$$
- For $A \Rightarrow B$, introduce c with: $(c \wedge A) \Rightarrow B$, $(A \Rightarrow B) \Rightarrow c$,
  recursively introduce names for A and for B to get flat clause $(c \wedge a) \Rightarrow b$
  ...and implication clause $(a \Rightarrow b) \Rightarrow c$

Note 1: transformation only preserves satisfiability/provability
Note 2: Introducing names can be done more sparingly using some of the other G4ip rules for normalisation
(basically, the invertible non-branching ones)
Note 3: some of these rules were already presented by Vorob'ev in the form of pre-processing

From G4ip to SMT solving

With pre-processing the rule becomes

$$\frac{\Gamma_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad ((a \Rightarrow b) \Rightarrow c), \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{((a \Rightarrow b) \Rightarrow c), \Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

with $a, c \notin \{a_1, \ldots, a_n\}$ and $(a_1 \wedge \cdots \wedge a_n \Rightarrow c) \notin \Gamma_{\text{flat}}$

From G4ip to SMT solving

With pre-processing the rule becomes

$$\frac{\Gamma_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad ((a \Rightarrow b) \Rightarrow c), \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{((a \Rightarrow b) \Rightarrow c), \Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

with $a, c \notin \{a_1, \ldots, a_n\}$ and $(a_1 \wedge \cdots \wedge a_n \Rightarrow c) \notin \Gamma_{\text{flat}}$

Remarks:

- added formulae are all flat clauses
  (SAT-solver is good at treating increments)

From G4ip to SMT solving

With pre-processing the rule becomes

$$\frac{\Gamma_{\mathrm{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash b \qquad ((a \Rightarrow b) \Rightarrow c), \Gamma_{\mathrm{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash d}{((a \Rightarrow b) \Rightarrow c), \Gamma_{\mathrm{imp}}, \Gamma_{\mathrm{flat}} \vdash d}$$

with $a, c \notin \{a_1, \ldots, a_n\}$ and $(a_1 \wedge \cdots \wedge a_n \Rightarrow c) \notin \Gamma_{\mathrm{flat}}$

Remarks:

- added formulae are all flat clauses
  (SAT-solver is good at treating increments)
- Γ_{imp} never increases throughout proof-search,
  it actually decreases by 1 in the left branch

From G4ip to SMT solving

With pre-processing the rule becomes

$$\frac{\Gamma_{\mathrm{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash b \qquad ((a \Rightarrow b) \Rightarrow c), \Gamma_{\mathrm{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash d}{((a \Rightarrow b) \Rightarrow c), \Gamma_{\mathrm{imp}}, \Gamma_{\mathrm{flat}} \vdash d}$$

with $a, c \notin \{a_1, \ldots, a_n\}$ and $(a_1 \wedge \cdots \wedge a_n \Rightarrow c) \notin \Gamma_{\mathrm{flat}}$

Remarks:

- added formulae are all flat clauses
(SAT-solver is good at treating increments)
- Γ_{imp} never increases throughout proof-search,
it actually decreases by 1 in the left branch
- proofs have a **spine shape**,
and you cannot persistently climb up the left branches more times than the number of implication clauses

From G4ip to SMT solving

With pre-processing the rule becomes

$$\frac{\Gamma_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad ((a \Rightarrow b) \Rightarrow c), \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{((a \Rightarrow b) \Rightarrow c), \Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

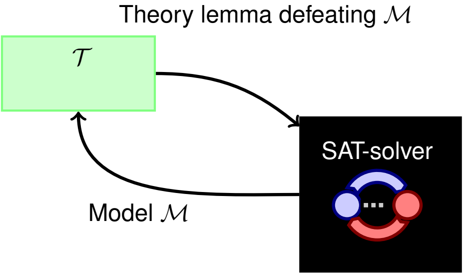
with $a, c \notin \{a_1, \ldots, a_n\}$ and $(a_1 \wedge \cdots \wedge a_n \Rightarrow c) \notin \Gamma_{\text{flat}}$

Remarks:

- added formulae are all flat clauses
  (SAT-solver is good at treating increments)
- Γ_{imp} never increases throughout proof-search,
  it actually decreases by 1 in the left branch
- proofs have a **spine shape**,
  and you cannot persistently climb up the left branches more times than the number of implication clauses
- thinking in terms of root-first proof-search, implemented recursively, the right premiss really corresponds to a tail call (i.e., a while loop)

From G4ip to SMT solving

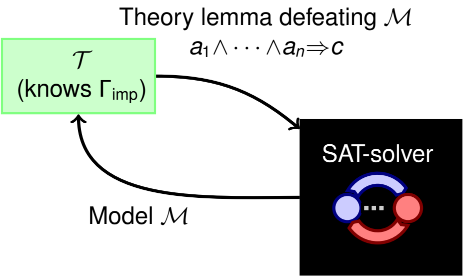
The spine shape describes a traditional SMT algorithm: DPLL($\mathcal{T}$).



From G4ip to SMT solving

The spine shape describes a traditional SMT algorithm: DPLL($\mathcal{T}$).

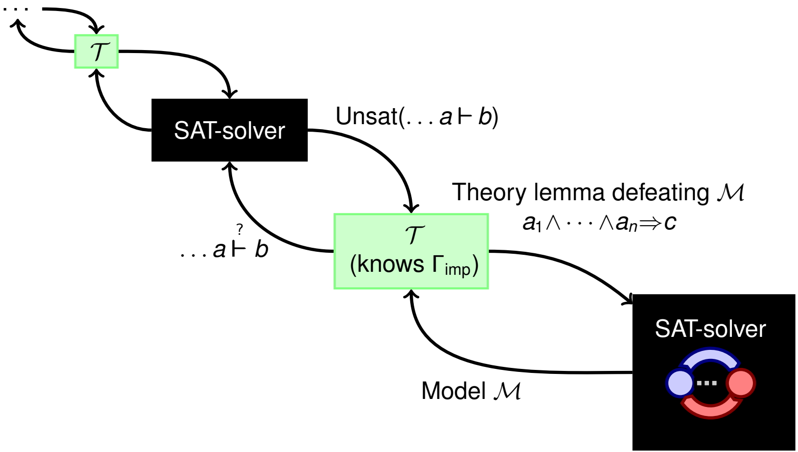
Theory $\mathcal{T}$ is that of "intuitionistic entailment"



From G4ip to SMT solving

The spine shape describes a traditional SMT algorithm: DPLL($\mathcal{T}$).

Theory $\mathcal{T}$ is that of "intuitionistic entailment"

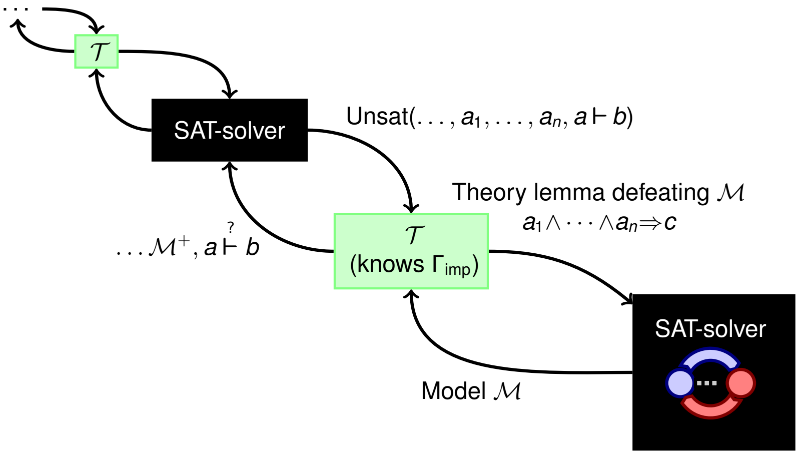


...except the "theory reasoning" that understands Γ_{imp} recursively relies on general provability.

From G4ip to SMT solving

The spine shape describes a traditional SMT algorithm: DPLL($\mathcal{T}$).

Theory $\mathcal{T}$ is that of "intuitionistic entailment"



$\dots$ except the "theory reasoning" that understands Γ_{imp} recursively relies on general provability.

And finally! we have the magic trick to pick $\{a_1, \dots, a_n\}$:

those atoms interpreted as true in $\mathcal{M}$ that were useful to prove $\dots a \vdash b$

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done.

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$.

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$. Recursively try to prove

$$\Gamma'_{\text{imp}}, \mathcal{M}^+, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \quad \text{where } \Gamma'_{\text{imp}} \text{ is } \Gamma_{\text{imp}} \backslash ((a \Rightarrow b) \Rightarrow c)$$

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$. Recursively try to prove
   $$\Gamma'_{\text{imp}}, \mathcal{M}^+, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \quad \text{where } \Gamma'_{\text{imp}} \text{ is } \Gamma_{\text{imp}} \backslash ((a \Rightarrow b) \Rightarrow c)$$
   If failure, find another implication clause to take into account.

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$. Recursively try to prove

   $$\Gamma'_{\text{imp}}, \mathcal{M}^+, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \quad \text{where } \Gamma'_{\text{imp}} \text{ is } \Gamma_{\text{imp}} \backslash ((a \Rightarrow b) \Rightarrow c)$$

   If failure, find another implication clause to take into account.
   If success, extract the $a_1, \ldots, a_n$ in $\mathcal{M}^+$ used in the proof.

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$. Recursively try to prove

   $$\Gamma'_{\text{imp}}, \mathcal{M}^+, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \quad \text{where } \Gamma'_{\text{imp}} \text{ is } \Gamma_{\text{imp}} \backslash ((a \Rightarrow b) \Rightarrow c)$$

   If failure, find another implication clause to take into account.
   If success, extract the $a_1, \ldots, a_n$ in $\mathcal{M}^+$ used in the proof.
   Return "flat theory clause" $a_1 \wedge \cdots \wedge a_n \Rightarrow c$ to SAT-solver so as to "defeat" its classical model $\mathcal{M}$, effectively applying rule

   $$\frac{\Gamma'_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$. Recursively try to prove

   $$\Gamma'_{\text{imp}}, \mathcal{M}^+, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \quad \text{where } \Gamma'_{\text{imp}} \text{ is } \Gamma_{\text{imp}} \backslash ((a \Rightarrow b) \Rightarrow c)$$

   If failure, find another implication clause to take into account.
   If success, extract the $a_1, \ldots, a_n$ in $\mathcal{M}^+$ used in the proof.
   Return "flat theory clause" $a_1 \wedge \cdots \wedge a_n \Rightarrow c$ to SAT-solver so as to "defeat" its classical model $\mathcal{M}$, effectively applying rule

   $$\frac{\Gamma'_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

   Then go back to the SAT-solver (1.)

Note: by construction, the learnt clause could not already be in Γ_{flat} otherwise the SAT solver would not have proposed model $\mathcal{M}$

From G4ip to SMT solving

So in order to prove $\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d$,

1. Run a SAT-solver on $\Gamma_{\text{flat}}, \neg d$ to see if $\Gamma_{\text{flat}} \vdash d$ is provable.
   If it is, we are done. If not, the SAT-solver returns a (classical) model $\mathcal{M}$ such that $\mathcal{M}(\Gamma_{flat}) = 1$ and $\mathcal{M}(d) = 0$. Then:
2. Pick in Γ_{imp} an implication clause $(a \Rightarrow b) \Rightarrow c$ such that $\mathcal{M}(c) = 0$, $\mathcal{M}(a) = 0$. Recursively try to prove

   $$\Gamma'_{\text{imp}}, \mathcal{M}^+, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \quad \text{where } \Gamma'_{\text{imp}} \text{ is } \Gamma_{\text{imp}} \backslash ((a \Rightarrow b) \Rightarrow c)$$

   If failure, find another implication clause to take into account.
   If success, extract the $a_1, \ldots, a_n$ in $\mathcal{M}^+$ used in the proof.
   Return "flat theory clause" $a_1 \wedge \cdots \wedge a_n \Rightarrow c$ to SAT-solver so as to "defeat" its classical model $\mathcal{M}$, effectively applying rule

   $$\frac{\Gamma'_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

   Then go back to the SAT-solver (1.)

Note: by construction, the learnt clause could not already be in Γ_{flat} otherwise the SAT solver would not have proposed model $\mathcal{M}$

If you run out of implication clauses in 2.: your sequent is unprovable.

Recursivity

Is the recursive nature of the general algorithm necessary?
Could we not have one big SMT-solving run?

$$\frac{\Gamma'_{\mathrm{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash b \qquad \Gamma_{\mathrm{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash d}{\Gamma_{\mathrm{imp}}, \Gamma_{\mathrm{flat}} \vdash d}$$

The recursivity is about climbing into the left premiss.
A SAT-solver has an internal learning mechanism.
It would be good if whatever is learnt by the SAT-solver of the recursive call could be shared with the SAT-solver of the caller.

Recursivity

Is the recursive nature of the general algorithm necessary?
Could we not have one big SMT-solving run?

$$\frac{\Gamma'_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

The recursivity is about climbing into the left premiss.
A SAT-solver has an internal learning mechanism.
It would be good if whatever is learnt by the SAT-solver of the recursive call could be shared with the SAT-solver of the caller.
So what really changes between the SAT-solver of the caller and that of the callee?

Recursivity

Is the recursive nature of the general algorithm necessary?
Could we not have one big SMT-solving run?

$$\frac{\Gamma'_{\text{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\text{flat}} \vdash b \qquad \Gamma_{\text{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\text{flat}} \vdash d}{\Gamma_{\text{imp}}, \Gamma_{\text{flat}} \vdash d}$$

The recursivity is about climbing into the left premiss.
A SAT-solver has an internal learning mechanism.
It would be good if whatever is learnt by the SAT-solver of the recursive call could be shared with the SAT-solver of the caller.
So what really changes between the SAT-solver of the caller and that of the callee? Mostly:

- the addition of a
- the addition of $\neg b$
- most most most importantly: the removal of $\neg d$

The SAT-solver of the callee is not allowed to exploit $\neg d$ to get UNSAT

Recursivity

Actually:

- Claessen and Dosén actually reuse the same SAT-solver for the recursive call. They use an incremental SAT-solver, where you can push and pop literals.
  Here: popping $\neg d$, pushing $a, \neg b$, so that what is learnt from each run (by the standard learning mechanisms of SAT-solving) is shared between the different runs.

Recursivity

Actually:

- Claessen and Dosén actually reuse the same SAT-solver for the recursive call. They use an incremental SAT-solver, where you can push and pop literals.
  Here: popping $\neg d$, pushing $a, \neg b$, so that what is learnt from each run (by the standard learning mechanisms of SAT-solving) is shared between the different runs.
- For this, you have to make sure the theory lemma $a_1 \wedge \cdots \wedge a_n \Rightarrow c$ is derivable from the input problem alone (irrespective of the pushed and popped literals).

Recursivity

Actually:

- Claessen and Dosén actually reuse the same SAT-solver for the recursive call. They use an incremental SAT-solver, where you can push and pop literals.
  Here: popping $\neg d$, pushing $a, \neg b$, so that what is learnt from each run (by the standard learning mechanisms of SAT-solving) is shared between the different runs.
- For this, you have to make sure the theory lemma $a_1 \wedge \cdots \wedge a_n \Rightarrow c$ is derivable from the input problem alone (irrespective of the pushed and popped literals).

In any case, popping $\neg d$ is not part of the main algorithm of the SAT-solver (DPLL/CDCL)

Recursivity

Actually:

- Claessen and Dosén actually reuse the same SAT-solver for the recursive call. They use an incremental SAT-solver, where you can push and pop literals.
  Here: popping $\neg d$, pushing $a, \neg b$, so that what is learnt from each run (by the standard learning mechanisms of SAT-solving) is shared between the different runs.
- For this, you have to make sure the theory lemma $a_1 \wedge \cdots \wedge a_n \Rightarrow c$ is derivable from the input problem alone (irrespective of the pushed and popped literals).

In any case, popping $\neg d$ is not part of the main algorithm of the SAT-solver (DPLL/CDCL)

This method provides what is probably the fastest prover for IPL (at least in 2015)

How to account for learned clauses proof-theoretically?

More precisely:
those clauses that are discovered while proving a subgoal, and possibly reused later?

How to account for learned clauses proof-theoretically?

More precisely:
those clauses that are discovered while proving a subgoal, and possibly reused later?

- In [GL13, GL14], I described them in terms of *memoisation* of root-first proof search

How to account for learned clauses proof-theoretically?

More precisely:
those clauses that are discovered while proving a subgoal, and possibly reused later?

- In [GL13, GL14], I described them in terms of *memoisation* of root-first proof search
- In [FGLM13, GL14] as well as this paper, we described them in terms of *cuts*

A proof-producing / model-constructing version

In the paper, we rephrased Claessen and Dosén's approach, formulating a root-first proof search algorithm so that

- in case of success, it builds proofs using 3 rules:

A proof-producing / model-constructing version

In the paper, we rephrased Claessen and Dosén's approach, formulating a root-first proof search algorithm so that

- in case of success, it builds proofs using 3 rules:
  - the generalised $\Rightarrow$-left rule of G4,
  - the cut rule,
  - an axiom rule that captures any resolution proof (valid in inituitionstic logic) returned by the SAT-solver;

A proof-producing / model-constructing version

In the paper, we rephrased Claessen and Dosén's approach, formulating a root-first proof search algorithm so that

- in case of success, it builds proofs using 3 rules:
  - the generalised $\Rightarrow$-left rule of G4,
  - the cut rule,
  - an axiom rule that captures any resolution proof (valid in inituitionstic logic) returned by the SAT-solver;
- in case of failure, it builds a Kripke counter-model of the sequent

A proof-producing / model-constructing version

In the paper, we rephrased Claessen and Dosén's approach, formulating a root-first proof search algorithm so that

- in case of success, it builds proofs using 3 rules:
  - the generalised $\Rightarrow$-left rule of G4,
  - the cut rule,
  - an axiom rule that captures any resolution proof (valid in inituitionstic logic) returned by the SAT-solver;
- in case of failure, it builds a Kripke counter-model of the sequent
  - with explicit worlds: sequences of implication clauses

A proof-producing / model-constructing version

In the paper, we rephrased Claessen and Dosén's approach, formulating a root-first proof search algorithm so that

- in case of success, it builds proofs using 3 rules:
  - the generalised $\Rightarrow$-left rule of G4,
  - the cut rule,
  - an axiom rule that captures any resolution proof (valid in inituitionstic logic) returned by the SAT-solver;
- in case of failure, it builds a Kripke counter-model of the sequent
  - with explicit worlds: sequences of implication clauses
  - when applying the the generalised $\Rightarrow$-left rule:

$$\frac{\Gamma'_{\mathrm{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash b \qquad \Gamma_{\mathrm{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash d}{\Gamma_{\mathrm{imp}}, \Gamma_{\mathrm{flat}} \vdash d}$$

    going into the left premiss means "pick a world above w where a is true but b is false, then try to find a contradiction from there".

A proof-producing / model-constructing version

In the paper, we rephrased Claessen and Dosén's approach, formulating a root-first proof search algorithm so that

- in case of success, it builds proofs using 3 rules:
  - the generalised $\Rightarrow$-left rule of G4,
  - the cut rule,
  - an axiom rule that captures any resolution proof (valid in inituitionstic logic) returned by the SAT-solver;
- in case of failure, it builds a Kripke counter-model of the sequent
  - with explicit worlds: sequences of implication clauses
  - when applying the the generalised $\Rightarrow$-left rule:

$$\frac{\Gamma'_{\mathrm{imp}}, a_1, \ldots, a_n, a, (b \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash b \qquad \Gamma_{\mathrm{imp}}, (a_1 \wedge \cdots \wedge a_n \Rightarrow c), \Gamma_{\mathrm{flat}} \vdash d}{\Gamma_{\mathrm{imp}}, \Gamma_{\mathrm{flat}} \vdash d}$$

    going into the left premiss means "pick a world above w where a is true but b is false, then try to find a contradiction from there".

This also provides a constructive proof of the completeness of the approach.

Questions

1. Could we open up the black box of the SAT-solver and integrate inside it theory reasoning, in our case "intuitionistic entailment", so as to have an intuitionistic version of DPLL?

Questions

1. Could we open up the black box of the SAT-solver and integrate inside it theory reasoning, in our case “intuitionistic entailment”, so as to have an intuitionistic version of DPLL?
   An “intuitionistic DPLL” would have to integrate some mechanism equivalent to making $\neg d$ unusable in (the part of the computation corresponding to) the recursive call.

Questions

1. Could we open up the black box of the SAT-solver and integrate inside it theory reasoning, in our case “intuitionistic entailment”, so as to have an intuitionistic version of DPLL?
   An “intuitionistic DPLL” would have to integrate some mechanism equivalent to making $\neg d$ unusable in (the part of the computation corresponding to) the recursive call.
2. Could we bypass the preprocessing and directly work on the input formulae?

Questions

1. Could we open up the black box of the SAT-solver and integrate inside it theory reasoning, in our case “intuitionistic entailment”, so as to have an intuitionistic version of DPLL?
   An “intuitionistic DPLL” would have to integrate some mechanism equivalent to making $\neg d$ unusable in (the part of the computation corresponding to) the recursive call.
2. Could we bypass the preprocessing and directly work on the input formulae?
3. Could we use SMT-solving’s quantifier instantiation techniques to generalise this to first-order?

Questions?

M. Farooque, S. Graham-Lengrand, and A. Mahboubi.
A bisimulation between DPLL(T) and a proof-search strategy for the focused sequent calculus.
In A. Momigliano, B. Pientka, and R. Pollack, editors, *Proc. of the 2013 Int. Work. on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP 2013)*. ACM Press, 2013.

S. Graham-Lengrand.
Psyche: a proof-search engine based on sequent calculus with an LCF-style architecture.
In D. Galmiche and D. Larchey-Wendling, editors, *Proc. of the 22nd Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux'13)*, volume 8123 of *LNCS*, pages 149–156. Springer-Verlag, 2013.

S. Graham-Lengrand.
Polarities & Focussing: a journey from Realisability to Automated Reasoning.
Habilitation thesis, Université Paris-Sud, 2014.
Available at `http://hal.archives-ouvertes.fr/tel-01094980`

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remark: the more a_i's there are, the weaker the new hypothesis in the right premiss

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remark: the more a_i's there are, the weaker the new hypothesis in the right premiss $\Rightarrow$ Take $\{a_1, \ldots, a_n\}$ as small as possible (if need be: post-process the proof to remove the a_i's that were not used)

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remark: the more a_i's there are, the weaker the new hypothesis in the right premiss $\Rightarrow$ Take $\{a_1, \ldots, a_n\}$ as small as possible (if need be: post-process the proof to remove the a_i's that were not used)

- if C is one of the a_i: uninteresting

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remark: the more a_i's there are, the weaker the new hypothesis in the right premiss $\Rightarrow$ Take $\{a_1, \ldots, a_n\}$ as small as possible (if need be: post-process the proof to remove the a_i's that were not used)

- if C is one of the a_i: uninteresting
- if A is one of the a_i: uninteresting (that a_i can be removed)

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remark: the more a_i's there are, the weaker the new hypothesis in the right premiss $\Rightarrow$ Take $\{a_1, \ldots, a_n\}$ as small as possible (if need be: post-process the proof to remove the a_i's that were not used)

- if C is one of the a_i: uninteresting
- if A is one of the a_i: uninteresting (that a_i can be removed)
- if B is one of the a_i: left premiss trivial to prove, & no other a_j needed. Interesting inasmuch it implements the rule

$$\frac{((A \Rightarrow B) \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D} \ (B \Rightarrow C) \notin \Gamma$$

(which we could have separately)

Restricting $\{a_1, \ldots, a_n\}$

$$\frac{a_1, \ldots, a_n, A, (B \Rightarrow C), \Gamma \vdash B \qquad ((A \Rightarrow B) \Rightarrow C), (a_1 \wedge \cdots \wedge a_n \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D}$$

Remark: the more a_i's there are, the weaker the new hypothesis in the right premiss $\Rightarrow$ Take $\{a_1, \ldots, a_n\}$ as small as possible (if need be: post-process the proof to remove the a_i's that were not used)

- if C is one of the a_i: uninteresting
- if A is one of the a_i: uninteresting (that a_i can be removed)
- if B is one of the a_i: left premiss trivial to prove, & no other a_j needed. Interesting inasmuch it implements the rule

  $$\frac{((A \Rightarrow B) \Rightarrow C), (B \Rightarrow C), \Gamma \vdash D}{((A \Rightarrow B) \Rightarrow C), \Gamma \vdash D} \; (B \Rightarrow C) \notin \Gamma$$

  (which we could have separately)

Then what?