

SAT-based Proof Search in Intermediate Propositional Logics

Camillo Fiorentini¹ Mauro Ferrari²

¹Università degli Studi di Milano, Italy

²Università degli Studi dell'Insubria, Italy

IJCAR 2022, August 8th, Haifa, Israel

Part of FLoC 2022, <https://www.floc2022.org/>

Motivations

- In 2015, Claessen and Rosén introduced `intuit`, an efficient decision procedure for `IPL` (Intuitionistic Propositional Logic) exploiting an incremental SAT-solver.

K. Claessen and D. Rosén. SAT Modulo Intuitionistic Implications, LPAR 2015

- To improve performances, we have re-designed `intuit` by adding a restart operation, thus obtaining `intuitR` (`intuit` with Restart).

C. Fiorentini. Efficient SAT-based Proof Search in Intuitionistic Propositional Logic. CADE 2021

- `intuitR` outperforms `intuit` and other state-of-the-art provers:

`fCube` Ferrari et al. LPAR 2010

`intHistGC` Goré et al., IJCAR 2014

- Here we present `intuitRIL`, an extension of `intuitR` to `Intermediate Logics`.

CPL vs. IPL

- Language \mathcal{L} over $V = \{p, q, p_1, p_2, \dots\}$ (propositional variables)

$$\begin{aligned}\alpha, \beta &:= p \in V \mid \perp \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \alpha \rightarrow \beta \\ \neg\alpha &:= \alpha \rightarrow \perp\end{aligned}$$

- CPL (Classical Propositional Logic)

Set of formulas valid in **all** classical interpretations.

$$\alpha \notin \text{CPL} \quad \implies \quad \exists I \text{ (classical interpretation) s.t. } I \not\models \alpha$$

(classical) countermodel

- IPL (Intuitionistic Propositional Logic)

Set of formulas valid in **all** Kripke models.

- A **frame** $\langle W, \leq, r \rangle$ is a poset (partially ordered set), where r (the **root**) is the minimum element.
- A **Kripke model** over the frame $\langle W, \leq, r \rangle$ is obtained by defining a **valuation** $\vartheta : W \rightarrow 2^V$ on the worlds W which is **persistent**:

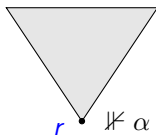
$$w_1 \leq w_2 \quad \implies \quad \vartheta(w_1) \subseteq \vartheta(w_2)$$

- Validity of a formula in a world in a expressed by **forcing** (\Vdash)

$$\begin{aligned}w \Vdash p &\text{ iff } p \in \vartheta(w) \text{ } (p \in V) \text{ and } w \nVdash \perp \\ w \Vdash A \wedge B &\text{ iff } w \Vdash A \text{ and } w \Vdash B \\ w \Vdash A \vee B &\text{ iff } w \Vdash A \text{ or } w \Vdash B \\ w \Vdash A \rightarrow B &\text{ iff, for every } w' \in W \text{ s.t. } w \leq w', w' \nVdash A \text{ or } w' \Vdash B\end{aligned}$$

CPL vs. IPL

$\alpha \notin \text{IPL} \implies$ There is a Kripke model K s.t.
 α is not forced at the root of K



K is a **countermodel** for α

- A classical interpretation can be viewed as a “degenerate” Kripke model only containing the root.

Accordingly $\text{IPL} \subseteq \text{CPL}$.

- The inclusion is **strict**

Examples of formula valid in CPL, but not in IPL.

$$a \vee \neg a, \quad \neg a \vee \neg \neg a, \quad (a \rightarrow b) \vee (b \rightarrow a) \quad \dots$$

Are there logics L such that $\text{IPL} \subset L \subset \text{CPL}$?

Intermediate Logics

An **intermediate logic** L is a set of formulas s.t. $\text{IPL} \subset L \subset \text{CPL}$ and:

(C1) L is closed under **modus ponens**

$$\alpha \rightarrow \beta \in L \quad \& \quad \alpha \in L \quad \implies \quad \beta \in L$$

(C2) L is closed under **substitutions** (maps $\chi : V \rightarrow \mathcal{L}$)

$$\alpha \in L \quad \implies \quad \chi(\alpha) \in L, \quad \forall \chi : V \rightarrow \mathcal{L}$$

An intermediate logic L can be obtained:

- **Semantically**

Impose some frame conditions

$$\alpha \in L \quad \text{iff} \quad \begin{array}{l} \alpha \text{ is valid in all Kripke models} \\ \text{satisfying the frame conditions} \end{array}$$

- **Syntactically**

$$L = \text{IPL} + \text{Axioms}$$

Axioms: set of formulas

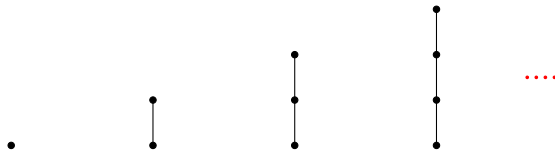
$$\alpha \in L \text{ iff } \Psi \vdash_{\text{ipl}} \alpha$$

Ψ : finite set of instances of the axioms (extra assumptions)

\vdash_{ipl} : derivability in IPL

Intermediate Logics: GL (Gödel-Dummett Logic)

- Semantical characterization: linear models



- Syntactical characterization:

$$\text{GL} = \text{IPL} + \underbrace{(\alpha \rightarrow \beta) \vee (\beta \rightarrow \alpha)}_{\text{linearity axiom}}$$

GL has been deeply investigated in the literature:

- close connections with fuzzy logics;
- Curry-Howard Interpretation of GL [Aschieri et al., LICS 2017]: extension of the λ -calculus so to capture parallel computations and communications between them.

Intermediate Logics: GL_n ($n \geq 0$)

- **Semantical characterization:** linear models having depth at most n
- **Syntactical characterization:**

$$GL_n = IPL + bd_n \quad \begin{array}{lcl} bd_0 & = & a_0 \vee \neg a_0 \\ bd_{n+1} & = & a_{n+1} \vee (a_{n+1} \rightarrow bd_n) \end{array}$$

We remark that:

- $GL_0 = CPL$
- GL_1 : formulas valid in the models



GL_1 is also known as *Here and There Logic (HT)*, well-known for its applications in ASP (Answer Set Programming).

See the characterization of stable model semantics based on HT-models introduced in [Lifschitz et al., TOCL 2021].

Intermediate Logics

- Infinitely many intermediate logics (power of the continuum).
- Ad hoc decision procedures: each logic is treated apart.

We present a **general** approach to decide validity in Intermediate Logics based on reduction to IPL-validity.

Given a logic L and a formula α :

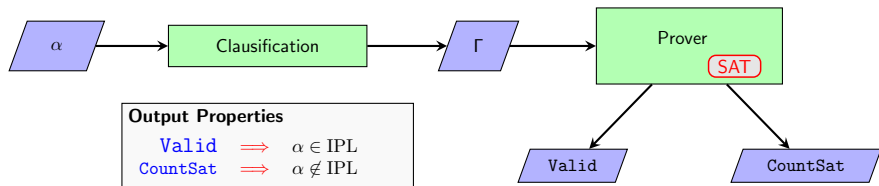
- (1) Single out a finite set Ψ containing instances of the characteristic axiom of L such that

$$\alpha \in L \quad \text{iff} \quad \Psi \vdash_{\text{iPl}} \alpha$$

- (2) Decide whether $\Psi \vdash_{\text{iPl}} \alpha$

Steps (1) and (2) are interleaved.

intuitR: architecture

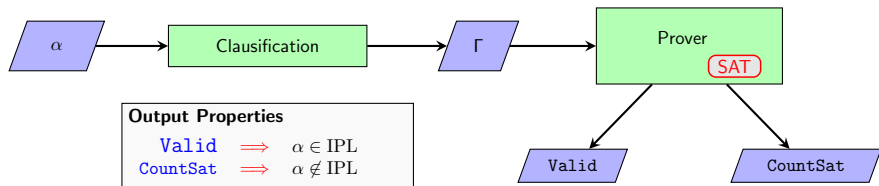


- The prover search for a countermodel for α
- Most of the computation is performed by an **incremental SAT-solver**
- We need a preprocessing phase (**clausification**) to reduce the input formula α to an equivalent set of clauses Γ of the form

flat clauses $\varphi \quad := \quad \bigwedge A_1 \rightarrow \bigvee A_2 \quad A_1, A_2: \text{sets of atoms}$
clauses added to the SAT-solver

implication clauses $\lambda \quad := \quad (a \rightarrow b) \rightarrow c \quad a, b, c: \text{atoms}$
clauses generating new worlds
of the countermodel

intuitR: Prover

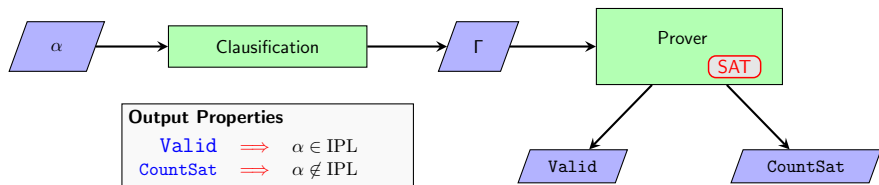


Incremental SAT-solver

clauses can be added to the solver, but not removed

- **Learning mechanism** to generate new clauses to feed the SAT-solver. Whenever we add a clause, the SAT-solver performs some internal simplifications, and next queries can be solved more efficiently
- However, since the SAT-solver is incremental, it is not possible to backtrack!
Accordingly, the decision procedure is quite different from standard strategy based on tableaux/sequent calculi, where backtracking is crucial to get completeness.

intuitR: Prover



Loop

- (1) Try to build a Kripke countermodel \mathcal{K} for Γ .
- (2) Whenever the construction of the countermodel fails:
 - (2.1) Learn a new flat clause (encoding the thrown semantic conflict)
 - (2.2) Add the learned clause to the SAT-solver
 - (2.3) Restart from (1) (new iteration of the loop)

The learned clauses prevent the repetition of the same semantic conflict, and this is crucial to get termination.

intuitR: Prover

Input Assumptions

$\Gamma = R, X, g$ where:

R : set of clauses $\bigwedge A_1 \rightarrow \bigvee A_2$

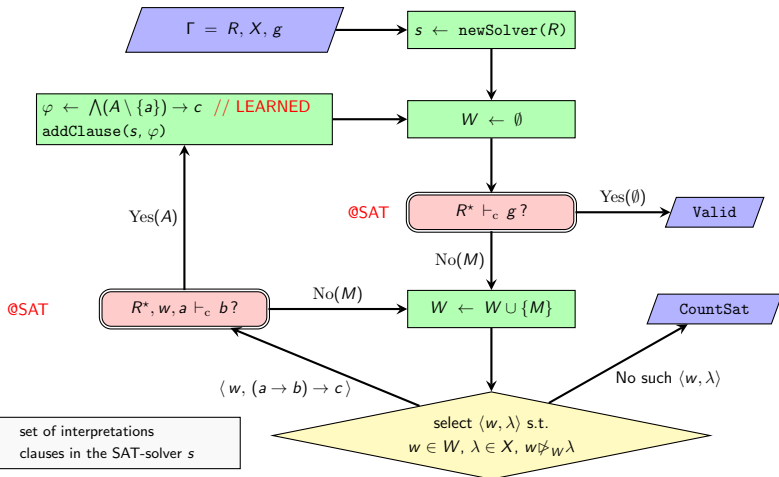
X : set of clauses $(a \rightarrow b) \rightarrow c$

g : atom

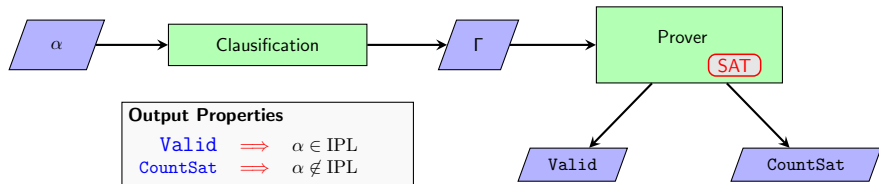
Output Properties

Valid $\Rightarrow R, X \vdash_{\text{ipl}} g$

CountSat $\Rightarrow R, X \not\vdash_i g$



intuitR: soundness



- The procedure is terminating.
- If the construction of a countermodel \mathcal{K} for Γ succeeds:
By properties of Clausification, \mathcal{K} is a countermodel for α .

$$\Rightarrow \alpha \notin \text{IPL}$$

- If the construction of a countermodel \mathcal{K} for Γ fails:
By properties of Clausification, there exists no countermodel for α .

$$\Rightarrow \alpha \in \text{IPL}$$

`intuitRIL` is obtained by extending `intuitR` to Intermediate Logics. Given an intermediate logic of L , we tweak the countermodel-search procedure as follows.

- (1) Whenever a countermodel \mathcal{K} is found, if \mathcal{K} is not an L -model, we throw a semantic conflict.
- (2) We pinpoint an instance ψ of the characteristic axiom of L which is falsified in \mathcal{K} (there exists at least one)
- (3) ψ is acknowledged as **learned axiom**
- (4) Restart

Main differences with respect to the original procedure:

- In general the learned axiom ψ is not a flat clause; thus, ψ must be clausified.
- Termination is not guaranteed and must be investigated on a case-by-case analysis.

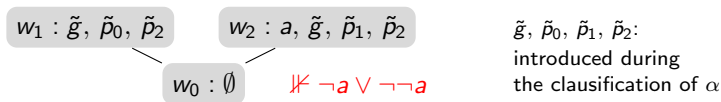
Actually, we can only prove that the learned axioms are pairwise non IPL-equivalent.

intuitRIL: a learning example

Logic $GL = IPL + (\beta \rightarrow \gamma) \vee (\gamma \rightarrow \beta)$

Input formula $\alpha = \neg a \vee \neg \neg a$ (**Weak Excluded Middle**)

The formula α is valid in GL (α cannot be falsified on linear models).
At some point of the computation of $\text{intuitRIL}(\alpha, GL)$, we get the following countermodel \mathcal{K} for α



\mathcal{K} is not a model for GL (actually, GL is not linear)

The following instance ψ of the linearity axiom is falsified in \mathcal{K}

$\psi = (a \rightarrow \neg a) \vee (\neg a \rightarrow a)$ **learned axiom**

We clausify ψ , add the obtained clauses and restart.

intuitRIL: termination

For logic GL, the procedure is terminating.

This follows from the fact that we can bound the instances of the linearity axiom $(\beta \rightarrow \gamma) \vee (\gamma \rightarrow \beta)$ needed to prove a formula

- If α is GL-valid, there is $\Psi_\alpha \subseteq \text{Ax}_{\text{GL}}(\alpha)$ such that $\Psi_\alpha \vdash_{\text{ipl}} \alpha$, where

$$\begin{aligned}\text{Ax}_{\text{GL}}(\alpha) = & \{ (a \rightarrow b) \vee (b \rightarrow a) \mid a, b \in \mathcal{V}_\alpha \} \cup \\ & \{ (a \rightarrow \neg a) \vee (\neg a \rightarrow a) \mid a \in \mathcal{V}_\alpha \} \cup \\ & \{ (a \rightarrow (a \rightarrow b)) \vee ((a \rightarrow b) \rightarrow a) \mid a, b \in \mathcal{V}_\alpha \} \\ & \mathcal{V}_\alpha: \text{ prop. variables occurring in } \alpha\end{aligned}$$

Ψ_α is a **bounding function** for GL.

Here we improve the bounding functions for GL introduced in [Avellone et al., TABLEAUX 1997; Ciabattoni et al., JSL 2021].

intuitRIL: termination

Using similar techniques we can guarantee termination for:

- All the Gödel-Dummett Logic GL_n (bounded depth)
- Jankov logic J_n

$$J_n = IPL + \underbrace{\neg\alpha \vee \neg\neg\alpha}_{\text{Weak Excluded Middle}} \quad \text{models having a maximum world}$$

- Scott Logic ST

This case is peculiar since ST -models are not first-order definable.

This witnesses that our approach is quite robust and general.

Conclusions & future work

- `intuitRIL` is a general prover for Intermediate Logics.
An Haskell implementation is available at
`https://github.com/cfiorentini/intuitRIL`
- The procedure is quite modular; to treat a specific logic L :
 - ✓ Implement a concrete learning mechanism for L
 - ✓ Prove termination
- We have implemented some of the mentioned intermediate logics.
- Future work
 - ✓ Extensions to other non-classical logics and to (fragments of) predicate logics.
 - ✓ [Goré et al, TABLEAUX 2021]: applications to Modal logics.
In this case, it is not possible to use a single SAT-solver, since the forcing relation in modal Kripke models is not persistent.