SAT-based Proof Search in Intermediate Propositional Logics

Camillo Fiorentini[1] Mauro Ferrari[2]

[1]Università degli Studi di Milano, Italy

[2]Università 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.

  Claessen & 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 $V = \{a, b, a_1, a_2, \dots\}$ (propositional variables)

$$\begin{array}{lcl} \alpha, \beta & := & a \in V \mid \bot \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \alpha \rightarrow \beta \\ \neg\alpha & := & \alpha \rightarrow \bot \end{array}$$

- CPL (Classical Propositional Logic)

  Set of formulas valid in all classical interpretations.

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

$$I \text{ is a (classical) countermodel for } \alpha$$

- 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:

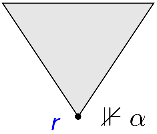
$$w_1 \leq w_2 \quad \Longrightarrow \quad \vartheta(w_1) \subseteq \vartheta(w_2)$$

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

$w \Vdash a$ iff $a \in \vartheta(w)$ $(a \in V)$ and $w \nVdash \bot$
$w \Vdash A \wedge B$ iff $w \Vdash A$ and $w \Vdash B$
$w \Vdash A \vee B$ iff $w \Vdash A$ or $w \Vdash B$
$w \Vdash A \rightarrow B$ iff, for every $w' \in W$ s.t. $w \leq w'$, $w' \nVdash A$ or $w' \Vdash B$

CPL vs. IPL

$\alpha \notin \mathrm{IPL} \quad \Longrightarrow \quad$ $\exists$ K (Kripke model) s.t.
α is not forced at the root of $\mathcal{K}$



$\mathcal{K}$ is a countermodel for α

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

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

- The inclusion is strict

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

$$a \vee \neg a\,, \quad \neg a \vee \neg\neg a\,, \quad (a \to b) \vee (b \to a) \qquad \ldots$$

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

Intermediate Logics

Definition (Intermediate Logic)

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

$$\text{modus ponens} \quad \alpha \to \beta \in L \quad \& \quad \alpha \in L \quad \Longrightarrow \quad \beta \in L$$

$$\text{substitutions} \quad \alpha \in L \quad \Longrightarrow \quad \chi(\alpha) \in L, \quad \forall\, \chi : V \to \mathcal{L}$$

An intermediate logic L can be obtained:

- Semantically
  Impose frame conditions

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

- Syntactically

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

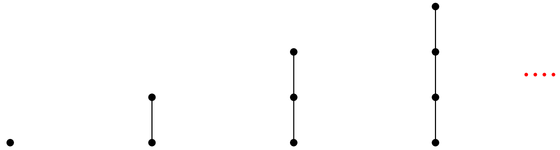
$$\alpha \in L \quad \Longleftrightarrow \quad \Psi \vdash_{\mathrm{ipl}} \alpha$$

Ψ : finite set of instances of the axioms

$\vdash_{\mathrm{ipl}}$: derivability in IPL

Intermediate Logics: GL (Gödel-Dummett Logic)

- Semantical characterization: linear models



- Syntactical characterization:

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

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:

$$\mathrm{GL}_n \;=\; \mathrm{IPL} \;+\; \mathbf{bd}_n \qquad \begin{array}{lcl} \mathbf{bd}_0 & = & a_0 \vee \neg a_0 \\ \mathbf{bd}_{n+1} & = & a_{n+1} \vee (a_{n+1} \to \mathbf{bd}_n) \end{array}$$

We remark that:

- $\mathrm{GL}_0 \;=\; \mathrm{CPL}$
- 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 at 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

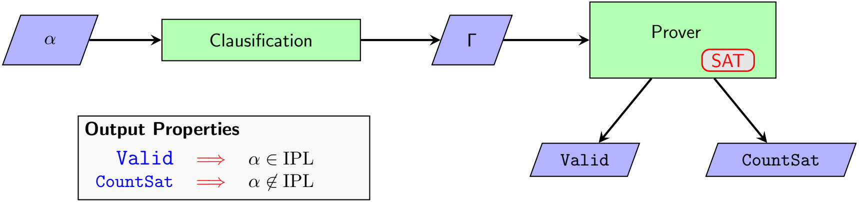
$$\alpha \in L \qquad \Longleftrightarrow \qquad \Psi \vdash_{\mathrm{ipl}} \alpha$$

(2) Decide whether $\Psi \vdash_{\mathrm{ipl}} \alpha$

Steps (1) and (2) are interleaved.

The procedure is designed on the top of `intuitR`

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

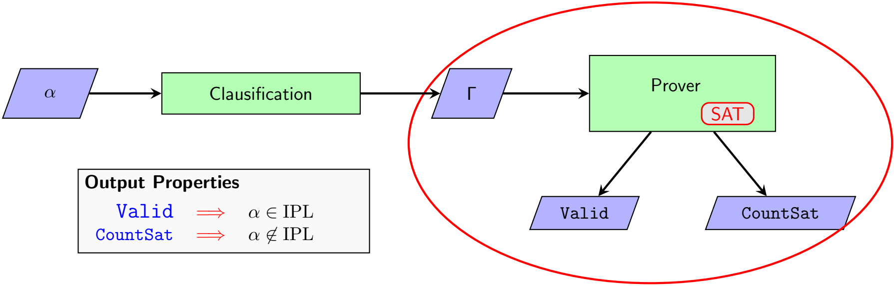
flat clauses $\varphi \quad := \quad \bigwedge A_1 \to \bigvee A_2 \qquad A_1, A_2$: sets of atoms

Clauses added to the SAT-solver

implication clauses $\lambda \quad := \quad (a \to b) \to c \qquad a, b, c$: atoms

Clauses generating new worlds of the countermodel

intuitR: Prover



Output Properties

`Valid`	$\Longrightarrow$	$\alpha \in \mathrm{IPL}$
`CountSat`	$\Longrightarrow$	$\alpha \notin \mathrm{IPL}$

Incremental SAT-solver

Clauses can be added to the solver, but not removed

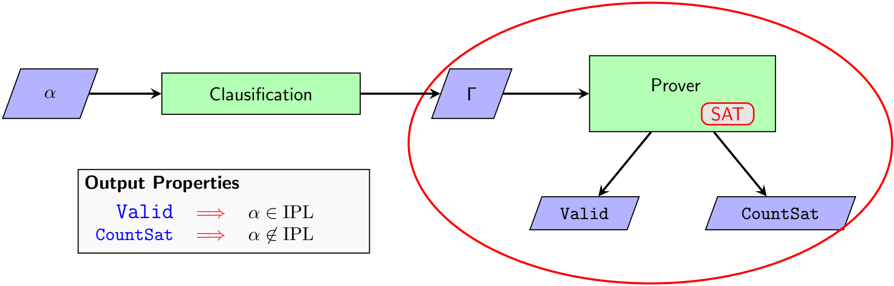
Learning mechanism to generate new clauses to feed the SAT-solver.

- Whenever we add a clause, the SAT-solver performs some internal simplifications, and next queries can be solved more efficiently
- Since the SAT-solver is incremental, backtracking is not allowed.

  Accordingly, clauses must express global and permanent properties.

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

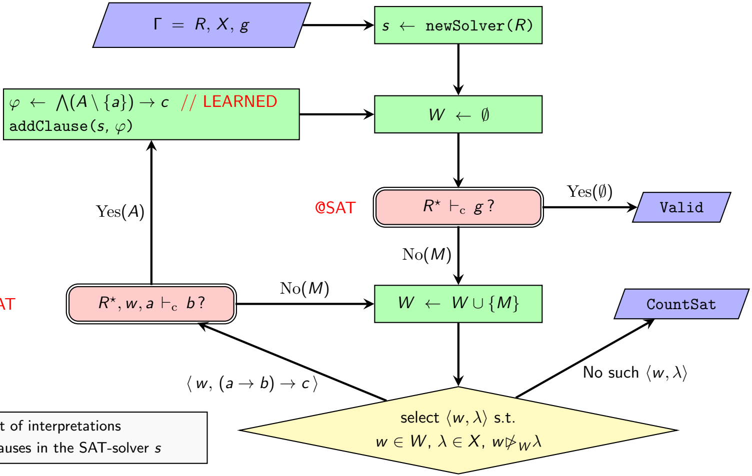
Input Assumptions

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

- R : set of clauses $\bigwedge A_1 \to \bigvee A_2$
- X : set of clauses $(a \to b) \to c$
- g : atom

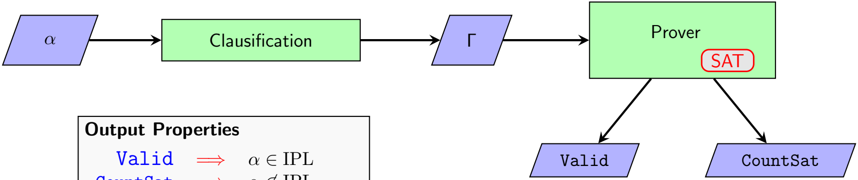
Output Properties

`Valid`	$\Longrightarrow$	$R, X \vdash_{\mathrm{ipl}} g$
`CountSat`	$\Longrightarrow$	$R, X \nvdash_{\mathrm{i}} g$



W:	set of interpretations
$R^\star$:	clauses in the SAT-solver s

intuitR: soundness



Output Properties

Valid	$\Longrightarrow$	$\alpha \in \mathrm{IPL}$
CountSat	$\Longrightarrow$	$\alpha \notin \mathrm{IPL}$

- 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.

$$\Longrightarrow \quad \alpha \notin \mathrm{IPL}$$

- If the construction of a countermodel $\mathcal{K}$ for Γ fails:

  By properties of Clausification, there exists no countermodel for α.

$$\Longrightarrow \quad \alpha \in \mathrm{IPL}$$

intuitRIL

intuitRIL is obtained by extending intuitR to Intermediate Logics.

Given a logic of L, we tweak the countermodel-search procedure:

(1) Whenever a countermodel $\mathcal{K}$ is found,
if $\mathcal{K}$ is not an L-model, then throw a semantic conflict.

(2) Select an instance ψ of the axiom of L falsified in $\mathcal{K}$
(there exists at least one)

(3) Acknowledge ψ as learned axiom

(4) Restart

Main differences with respect to the original procedure:

- In general the learned axiom ψ must be clausified (not a clause)
- Termination must be investigated on a case-by-case analysis.

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

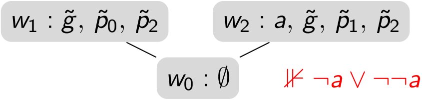
intuitRIL: a learning example

$$\text{Logic} \quad \mathrm{GL} \;=\; \mathrm{IPL} + (\beta \to \gamma) \vee (\gamma \to \beta)$$

$$\text{Input formula} \quad \alpha \;=\; \neg a \vee \neg\neg a \quad \text{(Weak Excluded Middle)}$$

The formula α is valid in GL (α cannot be falsified on linear models).

At some point of the computation of `intuitRIL(`α,GL`)`, we get the following countermodel $\mathcal{K}$ for α



$\tilde{g},\ \tilde{p}_0,\ \tilde{p}_1,\ \tilde{p}_2$: introduced during the clausification of 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 $\mathcal{K}$

$$\psi \;=\; (a \to \neg a) \vee (\neg a \to a) \qquad \text{learned axiom}$$

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 needed to prove a formula α:

- If α is GL-valid, then $\mathrm{Ax}_{\mathrm{GL}}(\alpha) \vdash_{\mathrm{ipl}} \alpha$, where

$$\mathrm{Ax}_{\mathrm{GL}}(\alpha) \quad = \quad \text{instances of the lin. axiom of the form}$$

$$(a \to b) \lor (b \to a)$$
$$(a \to \neg a) \lor (\neg a \to a)$$
$$(a \to (a \to b)) \lor ((a \to b) \to a)$$

where prop. variables a, b occur in α

$\mathrm{Ax}_{\mathrm{GL}}(\alpha)$ 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 Jn

  $$\mathrm{Jn} \;=\; \mathrm{IPL} + \underbrace{\neg\alpha \vee \neg\neg\alpha}_{\text{Weak Excluded Middle}} \qquad \text{models having a maximum world}$$

- Scott Logic ST

  $$\mathrm{ST} \;=\; \mathrm{IPL} \;+\; ((\neg\neg\alpha \to \alpha) \to \neg\alpha \vee \alpha) \;\to\; \neg\neg\alpha \vee \neg\alpha$$

  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 at 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.