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) based on a Satisfiability Modulo Theories (SMT) approach and 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 extend the procedure to Intermediate Logics; we get

  `intuitRIL` (`intuit` with Restart for Intermediate Logics).

CPL vs. IPL

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

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

- CPL (Classical Propositional Logic) is the set of formulas valid in all classical interpretations.

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

(classical) countermodel

- IPL is the 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 $\langle W, \leq, r\rangle$ is obtained by defining a valuation $\vartheta : W \to 2^V$ on the worlds of the frame which is persistent:

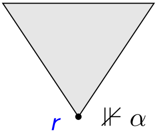
$$w_1 \leq w_2 \quad \Longrightarrow \quad \vartheta(w_1) \subseteq \vartheta(w_2)$$

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

$w \Vdash p$ iff $p \in \vartheta(w)$ ($p \in V$) 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 $w \Vdash B$
$w \Vdash A \to B$ iff, for every $w' \in W$ s.t. $w \leq w'$, $w' \nVdash A$ or $w' \Vdash B$

CPL vs. IPL

$\alpha \notin \mathrm{IPL} \implies$ There is a Kripke model K 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 ($\mathrm{IPL} \subsetneq \mathrm{CPL}$).

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

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

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

Intermediate Logics

- An intermediate logic L is a set of formulas such that $\mathrm{IPL} \subset L \subset \mathrm{CPL}$ and:
  - (C1) L is closed under modus ponens

$$\alpha \to \beta \in L \quad \& \quad \alpha \in L \quad \Longrightarrow \quad \beta \in L$$

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

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

- An intermediate logic L can be obtained:
  - Semantically
    Impose some frame conditions

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

  - Syntactically

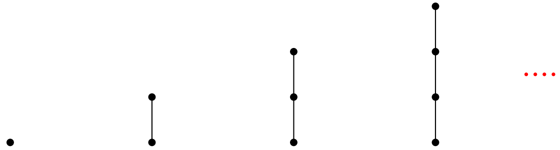
$$L \;=\; \mathrm{IPL} \;+\text{Axioms} \qquad\qquad \text{Axioms: set of formulas}$$

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

Ψ	:	finite set of instances of the axioms (extra assumptions)
$\vdash_{\mathrm{ipl}}$	:	derivability in IPL

Intermediate Logics: GL (Gödel-Dummet 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 [Aschier 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:

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

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 nice 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

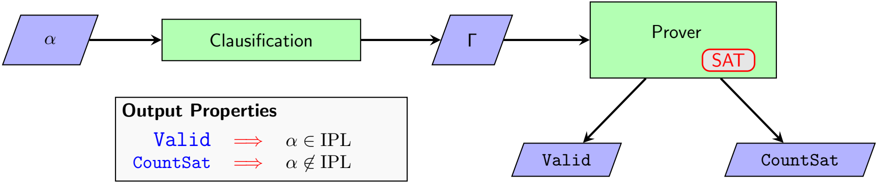
$$\alpha \in L \qquad \text{iff} \qquad \Psi \vdash_{\mathrm{ipl}} \alpha$$

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

Steps (1) and (2) are interleaved.

We define a variant of `intuitR` (`intuit` with Restart).

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

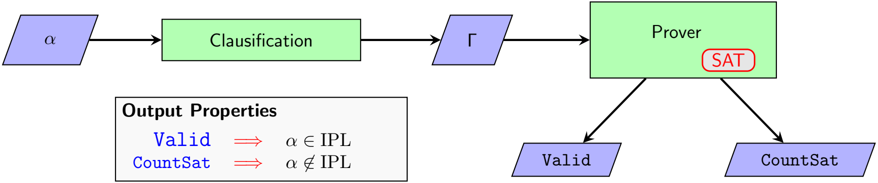
flat clauses φ $:=$ $\bigwedge A_1 \to \bigvee A_2$ A_1, A_2: sets of atoms

clauses added to the SAT-solver

implication clauses λ $:=$ $(a \to b) \to c$ a, b, c: atoms

clauses used to generate new worlds

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

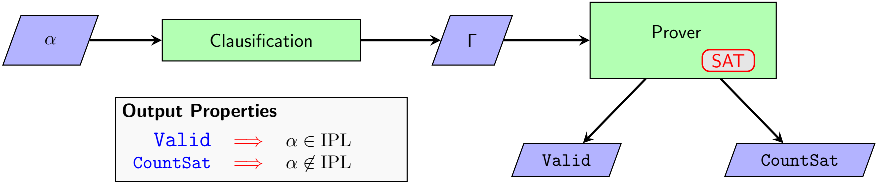
- Learning mechanism to generate new clauses to feed the SAT-solver.

  Whenever we add a clause, the solver internally performs some simplifications and next queries can be solved more efficiently

- However, since the 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 obtained semantic conflict)
(2.2) Add the learned clause to the SAT-solver
(2.3) Restart from (1) (new iteration of the main loop)

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

intuitR: Prover

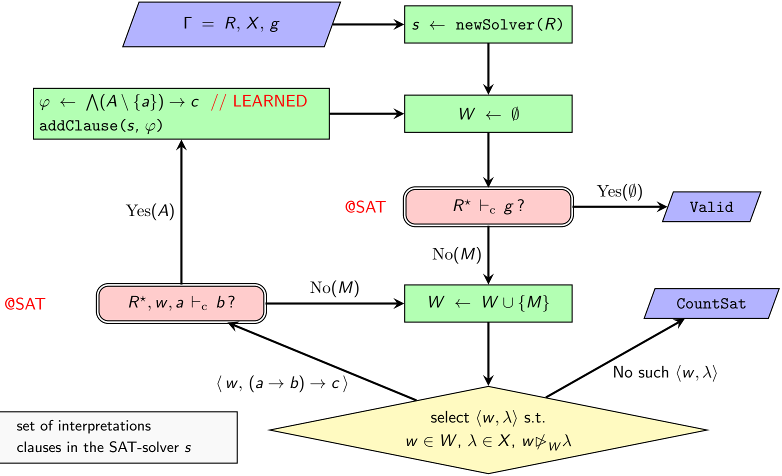
Input Assumptions

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

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

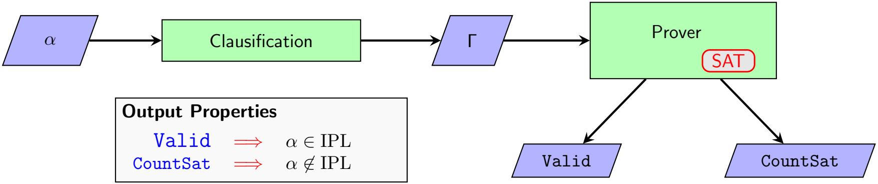
Output Properties

`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



- 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 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 not forced in $\mathcal{K}$ (there exists at least one)

(3) We take ψ as learned axiom and restart

Remark

- In general ψ is not a flat clause; thus, ψ must be clausified.
- We can guarantee that the learned axioms are pairwise non IPL-equivalent.
  However, this is not sufficient to guarantee termination, which must be investigated on a case-by-case analysis.

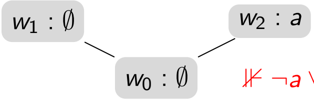
intuitRIL: a learning example

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

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

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



$\nVdash \neg a \lor \neg\neg a$

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

The following instance ψ of the GL-axiom is falsified in $\mathcal{K}$

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

We clausify ψ, add the obtained clauses and restart.

$\texttt{intuitRIL}$: Termination

For logic GL, the procedure is terminating.

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

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

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

Ψ_a is a bounding function for GL.

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-Dummet Logic GL_n (bounded depth)
- Jankov logic Jn

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

- 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 and 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 specific learning mechanism for L
  - √ Prove termination
- We have implemented some of the mentioned intermediate logics.
- Future work
  - √ Application of the method to other non-classical logics or to fragments of predicate logics.
  - √ [Goré at al, TABLEAUX 2021]: applications to Modal logics.

    However, it is not possible to use a single SAT-solver, since the forcing relation in modal Kripke models is not persistent.