Forward proof-search and Countermodel Construction in Intuitionistic Propositional Logic*

Camillo Fiorentini[1], Mauro Ferrari[2]

[1] DI, Univ. degli Studi di Milano, Via Celoria, 18, 20133 Milano, Italy
[2] DiSTA, Univ. degli Studi dell'Insubria, Via J.H. Dunant, 3, 21100, Varese, Italy

Abstract. In this extended abstract we review some recent work about the application of the inverse method to refute formulas in Intuitionistic Propositional Logic.

The inverse method, introduced in the 1960s by Maslov [21], is a saturation based theorem proving technique closely related to (hyper)resolution [7]; it relies on a forward proof-search strategy and can be applied to cut-free calculi enjoying the subformula property. Given a goal, a set of instances of the rules of the calculus at hand is selected; such specialized rules are repeatedly applied in the forward direction, starting from the axioms (i.e., the rules without premises). Proof-search terminates if either the goal is obtained or the database of proved facts saturates (no new fact can be added). As pointed out by Vladimir Lifschitz [20], *"the role of the inverse method in the Soviet work on proof procedures for predicate logic can be compared to the role of resolution method in theorem proving projects in the West"*. But, he regrets, *"for a number of reasons, this work has not been duly appreciated outside a small circle of Maslov's associates"*. The method has been popularized by Degtyarev and Voronkov [7], who provide the general recipe to design forward calculi, with applications to Classical Predicate Logic and some non-classical logics. Further extensions can be found in [2,8,19]. A significant investigation is presented in [4,5], where focused calculi and polarization of formulas are exploited to reduce the search space in forward proof-search for Intuitionistic Logic. These techniques are at the heart of the design of the prover Imogen [22].

In all the mentioned papers, the inverse method has been exploited to prove the validity of a goal in a specific logic. In [15,16] we follow the dual approach, namely: we design a forward calculus to derive *refutations* asserting the unprovability of a goal formula in Intuitionistic Propositional Logic (IPL). Our motivation is twofold. Firstly, we aim to define a refutation calculus which constructively ascertains the unprovability of a formula by providing a concise coun-

 This extended abstract is a summary of [16].

termodel for it[3]. The second motivation is to clarify the role of the saturated database obtained when the *search for a refutation* (*refutation-search*) fails. In the case of the usual forward calculi for Intuitionistic provability, if proof-search fails, a saturated database is generated which *"may be considered a kind of countermodel for the goal sequent"* [22]. However, as far as we know, no method has been proposed to effectively extract it. Actually, the main problem comes from the high level of non-determinism involved in the construction of countermodels. Here, assuming the dual approach, the saturated database generated by a failed refutation-search can be considered as *"a kind of proof of the goal"*; we give evidence of this by showing how to extract from such a database a derivation witnessing the Intuitionistic validity of the goal.

The formula to be proved (the goal formula) determines the instances of the rules of the forward calculus. The calculus we define is parametrized by the goal formula G (where the goal is to prove that G is not valid in IPL). We call the related calculus $\mathbf{FRJ}(G)$ (Forward Refutation calculus for IPL parametrized by G); formulas occurring in the sequents of $\mathbf{FRJ}(G)$ are suitable subformulas of G. In [16] we define a forward refutation-search procedure to build an $\mathbf{FRJ}(G)$-refutation of a goal formula G, namely an $\mathbf{FRJ}(G)$-refutation of a sequent of the form $\Gamma \not\Rightarrow G$, meaning that G is not derivable in IPL from assumptions Γ. This is a standard saturation procedure where the derivable sequents of $\mathbf{FRJ}(G)$ are collected step-by-step in a database $\mathtt{D}_G$. To avoid redundancies and maintain $\mathtt{D}_G$ compact, we introduce a *subsumption relation* between sequents; for instance, if at some step σ is proved and σ is subsumed by a sequent already in $\mathtt{D}_G$, then σ is discarded and not added to $\mathtt{D}_G$ (*forward subsumption*).

If the formula G is valid in IPL, refutation-search for G fails (indeed, no $\mathbf{FRJ}(G)$-refutation of G can be built) and we eventually get a *saturated database* $\mathtt{D}_G$ for G. This means that for every sequent σ derivable in $\mathbf{FRJ}(G)$, $\mathtt{D}_G$ contains a sequent σ' which subsumes σ; thus $\mathtt{D}_G$ is in some sense representative of all the sequents derivable in $\mathbf{FRJ}(G)$. We can exploit $\mathtt{D}_G$ to build a derivation of G in a sequent calculus for IPL. To this aim, we introduce the sequent calculus $\mathbf{Gbu}(G)$, a variant of the well-known sequent calculus **G3i** [25]. From a $\mathbf{Gbu}(G)$-derivation of G, we can immediately obtain a **G3i**-derivation of G. Differently from **G3i**, backward proof-search in $\mathbf{Gbu}(G)$ always terminates; indeed, we can define a weight function on sequents such that, after the backward application of a rule of $\mathbf{Gbu}(G)$ to a sequent, the weight of the sequents decreases. Nonetheless, backward-proof search in $\mathbf{Gbu}(G)$ might present several backtrack points, in correspondence of the applications of rules for left implication and right disjunction. The crucial point is that we can remove backtracking by exploiting the database $\mathtt{D}_G$: in presence of multiple non-deterministic choices, we query $\mathtt{D}_G$ as an oracle to select the right way so to successfully continue proof-search. Thus, we can consider $\mathtt{D}_G$ as a *proof-certificate* of the validity of G, in the sense that it contains enough information to reconstruct a derivation of G in the sequent calculus **G3i**. In general $\mathtt{D}_G$ is not unique; however, if we eliminate all the re-

[3] Note that our use of the term *refutation* is different from the one in the context of resolution, where it is about establishing that False is entailed in all models.

dundancies from $\mathtt{D}_G$ (if σ belongs to $\mathtt{D}_G$, then remove from $\mathtt{D}_G$ all the sequents subsumed by σ), then we get a saturated database $\mathtt{D}^*_G$ which is the *minimum* among the saturated databases of G, hence we can consider $\mathtt{D}^*_G$ as the canonical proof-certificate of the validity of G. To get the minimum saturated database, we have to enhance the refutation-search procedure by implementing *backward subsumption*. We have experimented that the backtracking-free proof-search in $\mathbf{Gbu}(G)$ driven by a saturated database can be more efficient than the usual backward proof-search procedure in **G3i**.

The rules of $\mathbf{FRJ}(G)$ are inspired by Kripke semantics. We show that, from a refutation of G, we can extract a countermodel for G, namely a Kripke model such that, at its root, the formula G is not forced, witnessing that G is not valid in IPL [3]. Actually, there is a close correspondence between a refutation and the related Kripke model. Thus, our forward refutation-search procedure can be understood as a top-down method to build a countermodel for G, starting from the final worlds down to the root. This original approach is dual to the standard one, where countermodels are built bottom-up, mimicking the backward application of rules (see, e.g., [1,6,9,10,11,17,18,23,24]). This different viewpoint has a significant impact on the outcome. Indeed, the countermodels generated by a backward procedure are always trees, which might contain some redundancies. Instead, forward methods re-use sequents and do not replicate them; thus the generated models are DAGs (Direct Acyclic Graphs) not containing duplications and are in general very concise. As a significant example, let us consider the one-variable formulas N_i of the Nishimura family [3], which are not valid in IPL:

$$N_1 = p \qquad N_{2n+3} = N_{2n+1} \vee N_{2n+2}$$
$$N_2 = \neg p \qquad N_{2n+4} = N_{2n+3} \supset N_{2n+1} \qquad n \geq 0$$

For formulas N_j our approach generates the standard "tower-like" minimum countermodel [3]; e.g. in Fig. 1 we display the countermodel for N_{17}.

We also investigate the relationship between a non-valid formula G and the height of the countermodel extracted from an $\mathbf{FRJ}(G)$-refutation of G. We show that, given a countermodel for G of height h, we can build an $\mathbf{FRJ}(G)$-refutation of G having height at most h. By this fact, we conclude that, if G is not valid in IPL, we can build an $\mathbf{FRJ}(G)$-refutation of G such that the height h of the extracted countermodel is minimal (namely, there exists no countermodel for G having height less than h). Actually, we can tweak the refutation-search procedure so that, if G is not valid, it yields an $\mathbf{FRJ}(G)$-refutation of G such that the extracted countermodel has minimal height. However, in general the obtained models are not minimal in the number of worlds, and the definition of a calculus devoted to the construction of minimal models (in the number of worlds) seems to be challenging. A different approach to generate minimal models, exploiting Answer Set Programming, is presented in [14].

To evaluate the potential of our approach we have implemented `frj`, a Java prototype of our refutation-search procedure based on the JTabWb framework [12][4]. `frj` implements term-indexing, forward and backward subsumption

[4] `frj` is available at `http://github.com/ferram/jtabwb_provers/`.

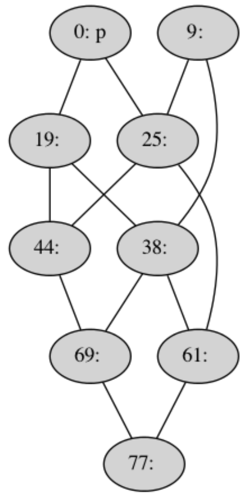

Fig. 1. Countermodel for N_{17}

and it allows the user to generate the rendering of proofs and of the extracted countermodels. We point out that the minimal countermodel in Fig. 1 has been generated by `frj`; the other provers we have tested fail to get such a concise countermodel.

As a future work we plan to investigate the applicability of our method to other logics, in particular to modal logics (see [13] for a preliminary work) and intermediate logics such as Gödel-Dummett logic characterized by linear Kripke models.

Acknowledgments

This work has been partially funded by the INdAM-GNCS project 2019 "METALLIC #2: METodi di prova per il ragionamento Automatico per Logiche nonLassICHe #2".

References

1. A. Avellone, C. Fiorentini, and A. Momigliano. A semantical analysis of focusing and contraction in intuitionistic logic. *Fundamenta Informaticae*, 140(3-4):247–262, 2015.
2. T. Brock-Nannestad and K. Chaudhuri. Disproving using the inverse method by iterative refinement of finite approximations. In H. De Nivelle, editor, *TABLEAUX 2015*, volume 9323 of *LNCS*, pages 153–168. Springer, 2015.
3. A. Chagrov and M. Zakharyaschev. *Modal Logic*. Oxford University Press, 1997.
4. K. Chaudhuri and F. Pfenning. A focusing inverse method theorem prover for first-order linear logic. In R. Nieuwenhuis, editor, *CADE-20*, volume 3632 of *LNCS*, pages 69–83. Springer, 2005.

5. K. Chaudhuri, F. Pfenning, and G. Price. A logical characterization of forward and backward chaining in the inverse method. In U. Furbach et al., editor, *IJCAR 2006*, volume 4130 of *LNCS*, pages 97–111. Springer, 2006.
6. K. Claessen and D. Rosén. SAT modulo intuitionistic implications. In M. Davis, A. Fehnker, A. McIver, and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings*, volume 9450, pages 622–637. Springer, 2015.
7. A. Degtyarev and A. Voronkov. The inverse method. In J.A. Robinson et al., editor, *Handbook of Automated Reasoning*, pages 179–272. Elsevier and MIT Press, 2001.
8. K. Donnelly, T. Gibson, N. Krishnaswami, S. Magill, and S. Park. The inverse method for the logic of bunched implications. In F. Baader et al., editor, *LPAR 2004*, volume 3452 of *LNCS*, pages 466–480. Springer, 2004.
9. M. Ferrari, C. Fiorentini, and G. Fiorino. FCube: An efficient prover for intuitionistic propositional logic. In C. G. Fermüller et al., editor, *LPAR 2010*, volume 6397 of *LNCS*, pages 294–301. Springer, 2010.
10. M. Ferrari, C. Fiorentini, and G. Fiorino. Contraction-free linear depth sequent calculi for intuitionistic propositional logic with the subformula property and minimal depth counter-models. *Journal of Automated Reasoning*, 51(2):129–149, 2013.
11. M. Ferrari, C. Fiorentini, and G. Fiorino. An evaluation-driven decision procedure for G3i. *ACM Transactions on Computational Logic (TOCL)*, 16(1):8:1–8:37, 2015.
12. M. Ferrari, C. Fiorentini, and G. Fiorino. JTabWb: a Java framework for implementing terminating sequent and tableau calculi. *Fundamenta Informaticae*, 150:119–142, 2017.
13. M. Ferrari, C. Fiorentini, and Fiorino G. Forward countermodel construction in modal logic K. In P. Felli and Montali M, editors, *Proceedings of the 33rd Italian Conference on Computational Logic, Bolzano, Italy, September 20-22, 2018*, volume 2214 of *CEUR Workshop Proceedings*, pages 75–81. CEUR-WS.org, 2018.
14. C. Fiorentini. An ASP approach to generate minimal countermodels in intuitionistic propositional logic. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1675–1681. ijcai.org, 2019.
15. C. Fiorentini and M. Ferrari. A forward unprovability calculus for intuitionistic propositional logic. In R. A. Schmidt and C. Nalon, editors, *TABLEAUX 2017*, volume 10501 of *LNCS*, pages 114–130. Springer, 2017.
16. C. Fiorentini and M. Ferrari. Duality between unprovability and provability in forward refutation-search for intuitionistic propositional logic. *ACM Trans. Comput. Logic*, 21(3), March 2020.
17. C. Fiorentini, R. Goré, and S. Graham-Lengrand. A proof-theoretic perspective on SMT-solving for intuitionistic propositional logic. In S. Cerrito and A. Popescu, editors, *TABLEAUX 2019*, volume 11714 of *LNCS*, pages 111–129. Springer, 2019.
18. R. Goré and L. Postniece. Combining derivations and refutations for cut-free completeness in bi-intuitionistic logic. *Journal of Logic and Computation*, 20(1):233–260, 2010.
19. L. Kovács, A. Mantsivoda, and A. Voronkov. The inverse method for many-valued logics. In F. Castro-Espinoza et al., editor, *MICAI 2013*, volume 8265 of *LNCS*, pages 12–23. Springer, 2013.
20. V. Lifschitz. What is the inverse method? *J. Automat. Reason.*, 5(1):1–23, 1989.
21. S. Ju. Maslov. An invertible sequential version of the constructive predicate calculus. *Zap. Naučn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 4:96–111, 1967.

22. S. McLaughlin and F. Pfenning. Imogen: Focusing the polarized inverse method for intuitionistic propositional logic. In I. Cervesato et al., editor, *LPAR 2008*, volume 5330 of *LNCS*, pages 174–181. Springer, 2008.
23. S. Negri. Proofs and countermodels in non-classical logics. *Logica Universalis*, 8(1):25–60, 2014.
24. L. Pinto and R. Dyckhoff. Loop-free construction of counter-models for intuitionistic propositional logic. In Behara et al., editor, *Symposia Gaussiana, Conference A*, pages 225–232. Walter de Gruyter, Berlin, 1995.
25. A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Camb. Univ. Press, 2ed edition, 2000.