Linking Focusing and Resolution with Selection

Guillaume Burel
ENSIIE and Samovar, Télécom SudParis and CNRS, Université Paris-Saclay, Évry, France
Inria and LSV, CNRS and ENS Paris-Saclay, Université Paris-Saclay, Cachan, France
guillaume.burel@ensiie.fr

Abstract

Focusing and selection are techniques that shrink the proof search space for respectively sequent calculi and resolution. To bring out a link between them, we generalize them both: we introduce a sequent calculus where each *occurrence* of an atom can have a positive or a negative polarity; and a resolution method where each literal, whatever its sign, can be selected in input clauses. We prove the equivalence between cut-free proofs in this sequent calculus and derivations of the empty clause in that resolution method. Such a generalization is not semi-complete in general, which allows us to consider complete instances that correspond to theories of any logical strength. We present three complete instances: first, our framework allows us to show that ordinary focusing corresponds to hyperresolution and semantic resolution; the second instance is deduction modulo theory; and a new setting, not captured by any existing framework, extends deduction modulo theory with rewriting rules having several left-hand sides, which restricts even more the proof search space.

2012 ACM Subject Classification Theory of computation → Proof theory, Theory of computation → Automated reasoning

Keywords and phrases logic in computer science, automated deduction, proof theory, sequent calculus, refinements of resolution, deduction modulo theory, polarization

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.9

Related Version A full version of the paper can be found at `https://hal.inria.fr/hal-01670476`.

1 Introduction

In addition to clever implementation techniques and data structures, a key point that explains the success of state-of-the-art automated theorem provers is the use of calculi that dramatically reduce proof search space. In the last decades, the independent developments of two families of techniques can be highlighted. First, in the kind of methods based on resolution, proof search space can be shrunk using ordering and selection techniques. The intuition is to restrict the application of the resolution rule to only some literals in a clause. If equality is considered, this leads to the superposition calculus [2] which is the base calculus of the currently most efficient automated provers for first-order classical logic. Second, in sequent calculi, Andreoli [1] introduced a technique called focusing to reduce non-determinism in the application of sequent-calculus rules. It works by first applying all invertible rules (those whose conclusion is logically equivalent to their premises) and second by chaining the application of non-invertible rules. Originally developed for linear logic, focusing has been extended to intuitionistic and classical first-order logic [26]. Focusing is mostly used in fields where sequent calculi, and related inverse and tableaux methods, are the most accurate proving method. For instance, there exists tools for first-order linear logic [12], for intuitionistic logic [27] and for modal logic [28]. Focusing is also the key ingredient in Miller's ProofCert project aiming at building a universal framework for proof certification [15].

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 9; pp. 9:1–9:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Despite their apparent lack of relation, we show in this paper that selection in refinements of the resolution calculus and focusing in sequent calculus are in fact strongly related, so that ordinary focusing in classical first-order logic corresponds actually to hyperresolution, where all negative literals are selected in a clause and are resolved at once. This connection is obtained by relaxing both techniques: concerning resolution, we allow any literal of the input clauses to be selected, whatever its sign; for the focusing part, we allow polarization not only of connectives, but also of all occurrences of literals. The main theorem of this paper, Theorem 3, shows that the sets of clauses whose insatisfiability can be proved by the resolution method with arbitrary input selection are exactly the sequents that have a cut-free proof in the generalized focusing setting.

This generalization allows us to cover a wider spectrum of proof systems. In particular, this permits to consider systems that search for proofs modulo some theory. Indeed, in real world applications, proof obligations are often verified within one or several theories. This explains the interest in and the success of Satisfiability Modulo Theory tools in recent years. Embedding a theory in our framework amounts to giving an axiomatic presentation of it where some literals are selected.

By relaxing the conditions for selecting literals, our framework is not always refutationaly complete. However, this should not be considered as a drawback, but as an essential point to be able to represent efficiently all kinds of theories. Indeed, let us consider a proof search method $\mathbb{P}(\mathcal{T})$ parameterized by a theory $\mathcal{T}$. Ideally, $\mathbb{P}(\mathcal{T})$ should be as efficient as a generic proof search method if it is fed with a formula that is not related to the theory $\mathcal{T}$. In particular, if it tries to refute the true formula $\top$, it should terminate, and with the answer "NO". Let us say that $\mathbb{P}(\mathcal{T})$ is relatively consistent if it is the case. As we pointed out with Dowek [8], we cannot have a generic proof of the completeness of a relatively consistent method $\mathbb{P}(\mathcal{T})$ that would work for all $\mathcal{T}$. Indeed, such a proof would imply the consistency of the theory $\mathcal{T}$, and, according to Gödel, this cannot be performed in $\mathcal{T}$ itself. So either the completeness of the proof system is proved once and for all, but it cannot represent theories that are logically at least as strong as that proof of completeness; or it is not complete in general but it can be proved to be complete for particular theories of some arbitrary logical strength. What is interesting therefore is to give proofs of completeness of $\mathbb{P}(\mathcal{T})$ for particular theories $\mathcal{T}$.

Therefore, we give three instances of our framework, where we can have proofs of completeness. First, as stated above, we link ordinary focusing with hyperresolution, and, in the ground case, with semantic resolution. Second, we show that Deduction Modulo Theory [20] is also a particular instance of this framework, knowing that there exists numerous proof techniques to prove the completeness of Deduction Modulo a particular theory, for instance [24, 21, 18, 7]. Third, we show how completeness in our framework can be reduced to completeness of several instances of Deduction Modulo Theory. To give an intuition about this last part, and to illustrate how much the proof search space can be constrained without losing completeness, let us consider for example the theory defining the powerset:

$$\forall X,\ \forall Y,\ (X \in \mathcal{P}(Y)) \Leftrightarrow (\forall Z,\ (Z \in X) \Rightarrow (Z \in Y))$$

This theory can be put in clausal normal form, using d as a Skolem symbol, and we select (by underlining them) some literals in these clauses[1]:

[1] We use the associative-commutative-idempotent symbol $\curlyvee$ in clauses to distinguish it from the symbol $\vee$ that is used in formulas.

$$\underline{\neg X \in \mathcal{P}(Y)} \curlyvee \underline{\neg Z \in X} \curlyvee Z \in Y \tag{1}$$
$$\underline{X \in \mathcal{P}(Y)} \curlyvee d(X,Y) \in X \tag{2}$$
$$\underline{X \in \mathcal{P}(Y)} \curlyvee \underline{\neg d(X,Y) \in Y} \tag{3}$$

Using focusing in general, and in our framework in particular, the decomposition of connectives is so restricted that, given an axiom, a proof derivation decomposing this axiom would necessarily have certain shapes. Thus, the axiom can be replaced by new inference rules, called synthetic rules, that are used instead of the derivation of those shapes. See end of Section 2, page 6, for more details. In our framework, this would lead to the following three synthetic rules, that can be used in place of the axioms (the explanation how these rules are obtained is given in Section 5.3):

$$(1)\vdash \frac{\Delta, u \in \mathcal{P}(v), t \in u, t \in v \vdash}{\Delta, u \in \mathcal{P}(v), t \in u \vdash} \qquad (2)\vdash \frac{\Delta, \neg u \in \mathcal{P}(v), d(u,v) \in u \vdash}{\Delta, \neg u \in \mathcal{P}(v) \vdash}$$

$$(3)\vdash \frac{}{\Delta, \neg u \in \mathcal{P}(v), d(u,v) \in v \vdash}$$

The only proof of transitivity of the membership in the powerset is then

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a \in \mathcal{P}(b), b \in \mathcal{P}(c), \underset{\sim}{\neg a \in \mathcal{P}(c)}, d(a,c) \in a, d(a,c) \in b, \underset{\sim}{d(a,c) \in c} \vdash}(3)\vdash}{a \in \mathcal{P}(b), \underset{\sim}{b \in \mathcal{P}(c)}, \neg a \in \mathcal{P}(c), d(a,c) \in a, \underset{\sim}{d(a,c) \in b} \vdash}(1)\vdash}{\underset{\sim}{a \in \mathcal{P}(b)}, b \in \mathcal{P}(c), \neg a \in \mathcal{P}(c), \underset{\sim}{d(a,c) \in a} \vdash}(1)\vdash}{a \in \mathcal{P}(b), b \in \mathcal{P}(c), \underset{\sim}{\neg a \in \mathcal{P}(c)} \vdash}(2)\vdash}{\underset{\sim}{a \in \mathcal{P}(b) \wedge b \in \mathcal{P}(c) \wedge \neg a \in \mathcal{P}(c)} \vdash}\wedge\vdash}{\underset{\sim}{\exists A.\ \exists B.\ \exists C.\ A \in \mathcal{P}(B) \wedge B \in \mathcal{P}(C) \wedge \neg A \in \mathcal{P}(C)} \vdash}\exists\vdash$$

where the active formulas in a sequent are underwaved, and double lines indicate potentially several applications of an inference rule.

On the resolution side, clauses (1) to (3) lead to the following ground derived rules (see also Section 5.3):

$$(1)\ \frac{u \in \mathcal{P}(v) \curlyvee C \qquad t \in u \curlyvee D}{t \in v \curlyvee C \curlyvee D} \qquad (2)\ \frac{\neg u \in \mathcal{P}(v) \curlyvee C}{d(u,v) \in u \curlyvee C}$$

$$(3)\ \frac{\neg u \in \mathcal{P}(v) \curlyvee C \qquad d(u,v) \in v \curlyvee D}{C \curlyvee D}$$

Once again, there is only one proof of transitivity, i.e. starting from the set of clauses $\{a \in \mathcal{P}(b); b \in \mathcal{P}(c); \neg a \in \mathcal{P}(c)\}$:

$$(3)\ \dfrac{\neg a \in \mathcal{P}(c) \qquad (1)\ \dfrac{b \in \mathcal{P}(c) \qquad (1)\ \dfrac{a \in \mathcal{P}(b) \qquad (2)\ \dfrac{\neg a \in \mathcal{P}(c)}{d(a,c) \in a}}{d(a,c) \in b}}{d(a,c) \in c}}{\Box}$$

and we cannot even infer other clauses than those. We let the reader compare with what happens if we used clauses (1) to (3) in resolution, even using the ordered resolution with selection refinement.

Related work. Chaudhuri et al. [13] show that hyperresolution for Horn clauses can be explained as an instance of a sequent calculus for intuitionistic linear logic with focusing where atoms are given a negative polarity.

Farooque et al. [23] developed a sequent calculus, based on focusing, that is able to simulate DPLL($\mathcal{T}$), the most common calculus used in SMT provers. The main difference with our framework is that in [23], the theory is considered as a black box which is called as an oracle. Here, the theory is considered as a first-class citizen.

Within the ProofCert project, resolution proofs can be checked by a kernel built upon a sequent calculus with focusing [15]. Based on this, the tool `Checkers` [14] is able to verify proofs coming from automated theorem provers based on resolution such as E-prover. Different from here, they translate resolution derivations using cuts to get smaller proofs.

Hermant [25] proves the correspondance between the cut-free fragment of a sequent calculus and a resolution method, in the setting of Deduction Modulo Theory. Since Deduction Modulo Theory is subsumed by our framework, Theorem 3 is a generalization of Hermant's work. Proving it is simpler in our setting because focusing restrains the shape of possible sequent calculus proofs, whereas Hermant had to prove technical lemmas to give proofs a canonical shape.

Notations and conventions. We use standard definitions for terms, predicates, formulas (with connectives $\bot, \top, \neg, \wedge, \vee$ and quantifiers $\forall, \exists$), sequents and substitutions. A literal is an atom or its negation. A clause is a set of literals. We will identify a literal with the unit clause containing it. Unless stated otherwise, letters $P, Q, R, P', P_1, \ldots$ denote atoms, $L, K, L', L_1, \ldots$ denote literals, $A, B, A', A_1, \ldots$ denote formulas, $C, D, C', C_1, \ldots$ denote clauses, Γ, Δ denote set of clauses or set of formulas (depending on the context). $A^\perp$ denotes the negation normal form of $\neg A$.

2 Focusing with Polarized Occurrences of Atoms

Focusing was introduced by Andreoli [1] to restrict the non-determinism in some sequent calculus for linear logic. It relies on the alternation of two phases: During the asynchronous phase (sequents with $\Uparrow$), all invertible rules are applied on the formulas of the sequent. Recall that a rule is said invertible if its conclusion implies the conjunction of its premises. During the synchronous phase (sequents with $\Downarrow$), a particular formula is selected – the focus is on it – and all possible non-invertible rules are successively applied on it. This idea has been extended to intuitionistic and classical first-order logic [26]. In these, connectives may have invertible and non-invertible versions of their sequent calculus rules. Therefore, one considers in that case two versions of a connective, one called positive when the right introduction rule is non-invertible, and one called negative when it is invertible. Some connectives, i.e. $\exists$ in classical logic, only have a positive version, and dually, others, such as $\forall$ in classical logic, only have a negative version. Given a usual formula, one can decide which version of a connective one wants to use at a particular occurrence, which is called a polarization of the formula.[2] Note that the polarity of a connective does not affect its semantics, it only alters the shape of the sequent calculus proofs. Similarly, one can decide the polarity of each literal. If a literal with negative polarity L is focused on in a branch, then this branch

[2] Let us note that this notion of polarity is a standard denomination when dealing with focusing, and should not be confused with the more usual but unrelated notion defined by the parity of the negation-depth of a position in a formula.

Asynchronous phase: $\widehat{\Uparrow}\vdash\ \dfrac{}{\Gamma, L, L^\perp \Uparrow\ \vdash}$ $\quad \Uparrow\exists\vdash\ \dfrac{\Gamma \Uparrow \Delta, A \vdash}{\Gamma \Uparrow \Delta, \exists x.\ A \vdash}$ x not free in Γ, Δ

$$\Uparrow\vee\vdash\ \frac{\Gamma \Uparrow \Delta, A \vdash \qquad \Gamma \Uparrow \Delta, B \vdash}{\Gamma \Uparrow \Delta, A \vee^+ B \vdash} \qquad \Uparrow\wedge\vdash\ \frac{\Gamma \Uparrow \Delta, A, B \vdash}{\Gamma \Uparrow \Delta, A \wedge^+ B \vdash} \qquad \Uparrow\top\vdash\ \frac{\Gamma \Uparrow \Delta \vdash}{\Gamma \Uparrow \Delta, \top \vdash}$$

Synchronous phase: $\widehat{\Downarrow}\vdash\ \dfrac{}{\Gamma, L^\perp \Downarrow \underline{L} \vdash}$ $\quad \Downarrow\forall\vdash\ \dfrac{\Gamma \Downarrow \{t/x\}A \vdash}{\Gamma \Downarrow \forall x.\ A \vdash}$ $\quad \Downarrow\bot\vdash\ \dfrac{}{\Gamma \Downarrow \bot \vdash}$

$$\Downarrow\vee\vdash\ \frac{\Gamma \Downarrow A \vdash \qquad \Gamma \Downarrow B \vdash}{\Gamma \Downarrow A \vee^- B \vdash} \qquad \Downarrow\wedge_1\vdash\ \frac{\Gamma \Downarrow A \vdash}{\Gamma \Downarrow A \wedge^- B \vdash} \qquad \Downarrow\wedge_2\vdash\ \frac{\Gamma \Downarrow B \vdash}{\Gamma \Downarrow A \wedge^- B \vdash}$$

Focus $\dfrac{\Gamma, A \Downarrow A \vdash}{\Gamma, A \Uparrow\ \vdash}$ A negative

Release $\dfrac{\Gamma \Uparrow A \vdash}{\Gamma \Downarrow A \vdash}$ A positive

Store $\dfrac{\Gamma, A \Uparrow \Delta \vdash}{\Gamma \Uparrow A, \Delta \vdash}$ A negative or literal

Figure 1 The sequent calculus LKF$^\perp$.

must necessarily be closed, with $L^\perp$ in the same context. (See rule $\widehat{\Downarrow}\vdash$ in Figure 1.) In the ordinary presentation of focusing, this polarity is chosen globally for all occurrences of each atom, and the polarity of $\neg P$ is defined as the inverse of that of P. In our setting, the polarity is attached to the position of the literal in the formula. In particular, if a substitution is applied to the formula, the polarities of the resulting literals do not change. The polarity of a formula is defined as the polarity of its top connective. Besides, note that to switch the polarity of a formula, e.g. to impose a change of phase, one can prefix it by so-called delays: $\delta^- A$ is negative whatever the polarity of A. Delays can be defined for instance by $\delta^- A = \forall x.\ A$ where x is not free in A, so we do not need them in the syntax and the rules.

Liang and Miller [26] introduce the sequent calculus LKF, and prove it to be complete for classical first-order logic. In Figure 1, we present the calculus LKF$^\perp$, which is almost the same with the following differences:

- All formulas are put on the left-hand side of the sequent, instead of the right-hand side. Therefore, one does not try to prove a disjunction of formulas, but one tries to refute a conjunction of formulas. This is the same thanks to the dual nature of classical first-order logic, and this helps to be closer to the resolution derivations. Note that, consequently, the focus is on negative formulas, and invertible rules are applied on positive formulas.
- The polarity of atoms is not chosen globally, but each *occurrence* of a literal can have a positive or a negative polarity. In particular, we can have two literals L and $L^\perp$ which are both negative, or both positive. We denote by $\underline{L}$ the fact that the literal L has a negative polarity. To be able to close branches on which we have two positive opposed literals, we add a rule $\widehat{\Uparrow}\vdash$.

We denote by $\Gamma \Uparrow \Delta \vdash$ (with Γ or Δ, possibly empty, containing polarized formulas) the fact that there exists a proof of the sequent $\Gamma \Uparrow \Delta \vdash$ in LKF$^\perp$, that is, a derivation starting from this sequent and whose branches are all closed (by $\widehat{\Downarrow}\vdash$, $\widehat{\Uparrow}\vdash$ or $\Downarrow\bot\vdash$). Thanks to focusing, such a proof has the following shape :

- Since one starts in an asynchronous ($\Uparrow$) phase, invertible rules are successively applied to the positive formulas of Δ, until one obtains negative formulas or literals that are put on the left of $\Uparrow$ using Store.
- When no formula appears on the right of $\Uparrow$, then either the branch is closed by $\widehat{\Uparrow}\vdash$; or the focus is put on a negative formula using Focus.
- In the latter case, one is now in synchronous ($\Downarrow$) phase where non-invertible rules are

successively applied to the formula upon which the focus is, until either the branch is closed using $\overset{\frown}{\Downarrow}\vdash$ or $\Downarrow\bot\vdash$; or one obtains a positive formula and the synchronous phase ends using Release.

- In the latter case, one starts again in the asynchronous phase.

Focusing therefore strongly constraints the shape of possible proofs, and therefore reduces the proof search space. The $\overset{\frown}{\Downarrow}\vdash$ in particular imposes to close branches immediately when the focus is on a negative literal, and thus rules out many derivations.

Note that proofs can be closed when the polarities of an atom and its negation are both positive (rule $\overset{\frown}{\Uparrow}\vdash$), or when one is positive and the other negative (rule $\overset{\frown}{\Downarrow}\vdash$), but not when they are both negative. Therefore, this restricts how formulas that contains literals with negative polarities can interact one with the others, and this is the main point of LKF$^\bot$ to reduce the proof search space.

Restricting proof search using focusing leads to what are called synthetic rules (see for instance [13, pp.148–150] where they are called derived rules). The idea is to replace some formula A in the context of the sequent by new inference rules. Instead of proving the sequent $A, \Delta \vdash$ in LKF$^\bot$, one proves $\Delta \vdash$ in (LKF$^\bot$+ the synthetic rules obtained from A). Indeed, a proof focusing on A can only have certain shapes, and thus instead of having A in the context, it can be replaced by new rules synthesizing those shapes. For instance, the formula $\underline{P} \vee^- (Q \wedge^+ \underline{R})$ in a context Γ can only lead to the following derivations when the focus is put on it:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \Downarrow P \vdash}\,\overset{\frown}{\Downarrow}\vdash \qquad \dfrac{\dfrac{\dfrac{\Gamma, Q \Uparrow \vdash}{\Gamma \Uparrow Q \vdash}\,\text{Store}}{\Gamma \Downarrow Q \vdash}\,\text{Release}}{\Gamma \Downarrow Q \wedge^- \underline{R} \vdash}\,\Downarrow\wedge_1\vdash}{\Gamma \Downarrow P \vee^- (Q \wedge^- \underline{R}) \vdash}\,\Downarrow\vee\vdash}{\Gamma \Uparrow \vdash}\,\text{Focus} \quad \text{and} \quad \dfrac{\dfrac{\dfrac{}{\Gamma \Downarrow P \vdash}\,\overset{\frown}{\Downarrow}\vdash \qquad \dfrac{\dfrac{}{\Gamma \Downarrow \underline{R} \vdash}\,\overset{\frown}{\Downarrow}\vdash}{\Gamma \Downarrow Q \wedge^- \underline{R} \vdash}\,\Downarrow\wedge_2\vdash}{\Gamma \Downarrow P \vee^- (Q \wedge^- \underline{R}) \vdash}\,\Downarrow\vee\vdash}{\Gamma \Uparrow \vdash}\,\text{Focus}\ .$$

In the left derivation, $P^\bot$ must be in Γ to be able to close the left branch, so Γ is in fact of the form $\underline{P} \vee^- (Q \wedge^+ \underline{R}), \Delta, P^\bot$. In the right one, Γ must be of the form $\underline{P} \vee^- (Q \wedge^+ \underline{R}), \Delta, P^\bot, R^\bot$. Instead of searching for a proof with $\underline{P} \vee^- (Q \wedge^+ \underline{R})$ in the context, the following two synthetic rules can therefore be used:

$$\text{Syn1}\ \dfrac{\Delta, P^\bot, Q \Uparrow \vdash}{\Delta, P^\bot \Uparrow \vdash} \qquad\qquad \text{Syn2}\ \dfrac{}{\Delta, P^\bot, R^\bot \Uparrow \vdash}$$

Provability is the same because each application of a synthetic rule can be replaced by applying Focus on $\underline{P} \vee^- (Q \wedge^+ \underline{R})$ and following the derivation leading the synthetic rule, and vice versa. This is used for instance in provers based on the inverse method and focusing [27].

The sequent calculus LKF$^\bot$ is not complete in general. One of the simplest examples of incompleteness is the sequent $\underline{P} \vee^- Q, \underline{\neg P} \vee^- Q, \neg Q \Uparrow\vdash$ which has no proof although $P \vee Q, \neg P \vee Q, \neg Q$ is not satisfiable.

3 Resolution with Input Selection

Two approaches can be used to reduce the proof search space of the resolution calculus: first, one can restrict on which pairs of clauses the resolution rule can be applied; this leads for instance to the set-of-support strategy [32], in which clauses are split into two sets, called the theory and the set of support; at least one of the clauses involved in a resolution step must be in the set of support. Second, one can restrict which literals in the clauses can be resolved

Resolution $\dfrac{L \curlyvee C \qquad L'^{\perp} \curlyvee D}{\sigma(C \curlyvee D)}$
- $\mathcal{S}(L \curlyvee C) = \emptyset$
- $\mathcal{S}(L'^{\perp} \curlyvee D) = \emptyset$
- σ is the most general unifier of $L =^? L'$

Factoring $\dfrac{L \curlyvee L' \curlyvee C}{\sigma(L \curlyvee C)}$
- $\mathcal{S}(L \curlyvee L' \curlyvee C) = \emptyset$
- σ is the most general unifier of $L =^? L'$

Resolution with Selection $\dfrac{\underline{K_1} \curlyvee \ldots \curlyvee \underline{K_n} \curlyvee C \qquad K_1'^{\perp} \curlyvee D_1 \qquad \ldots \qquad K_n'^{\perp} \curlyvee D_n}{\sigma(C \curlyvee D_1 \curlyvee \ldots \curlyvee D_n)}$
- $\mathcal{S}(\underline{K_1} \curlyvee \ldots \curlyvee \underline{K_n} \curlyvee C) = \{K_1; \ldots; K_n\}$
- $\mathcal{S}(K_i'^{\perp} \curlyvee D_i) = \emptyset$
- σ is the mgu of the simultaneous unification problem $K_1 =^? K_1', \ldots, K_n =^? K_n'$

Figure 2 Resolution with Input Selection.

upon; those literals are said to be selected in the clause. Resolution with free selection is complete for Horn clauses, but incomplete in general. Selecting a subset of the negative literals (if no literal is selected, then any literal of the clause can be used in resolution) is however complete, and combining this with an ordering restriction on clauses with no selected literals leads to Ordered Resolution with Selection, which was introduced by Bachmair and Ganzinger [2] (see also [3]) as a complete refinement of resolution.

Resolution with Input Selection combines these two approaches. It is parameterized by a selection function $\mathcal{S}$ that associate to each input clause a subset of its literals. If the selection function selects at least one literal, only those can be used in Resolution. Otherwise, any of them can be used. Note that for generated clauses, we impose that $\mathcal{S}(C) = \emptyset$. We also allow to have the same input clause several times with different selections. (That is, we actually work with couples composed of a clause and its selected literals.) The inference rules of Resolution with Input Selection are presented in Fig. 2. Literals that are selected in a clause are underlined. We will see that they indeed correspond to the literals that have a negative polarization in LKF$^{\perp}$. As usual, variables are renamed in the clauses to avoid that premises of the inference rules share variables. We have two flavors of the resolution rule: the usual binary resolution, that is applied on two premises that do not select any literal; and Resolution with Selection that is applied on a clause in which n literals are selected and n clauses is which no literal is selected. Consequently, clauses with a non-empty selection cannot be resolved one with the others. By considering them as the theory part, and the clauses with an empty selection as the set of support, it is easy to see that Resolution with Input Selection is a generalization of the set-of-support strategy. Notwithstanding, note that neither Resolution with Input Selection is a generalization of Ordered Resolution with Selection nor the converse.

▶ **Definition 1** (Resolution derivation)**.** We write $\Gamma \rightsquigarrow C$ if C can be derived from some clauses in Γ using the inference rules Resolution with Selection, Resolution, or Factoring presented in Figure 2. We write $\Gamma \rightsquigarrow^* C$ if
- $C \in \Gamma$ or if
- there exists D such that $\Gamma \rightsquigarrow D$ and $\Gamma, D \rightsquigarrow^* C$.

As usual in resolution methods, the goal is to produce the empty clause $\Box$ starting from a set of clauses Γ to show, since all rules are sound, that Γ is unsatisfiable. Here again, the calculus is not complete in general: from the set of clauses $\underline{P} \curlyvee Q, \underline{\neg P} \curlyvee Q, \neg Q$, no inference rule can be applied: to apply Resolution with Selection, we would need a clause where P, or $\neg P$, is not selected, and Resolution needs two clauses without selection.

4 LKF$^\perp$ is a Conservative Extension of Resolution with Input Selection

To link LKF$^\perp$ with Resolution with Input Selection, we need to indicate how clauses are related to polarized formulas.

▶ **Definition 2.** Given a clause $C = \underline{L_1} \curlyvee \cdots \curlyvee \underline{L_n} \curlyvee K_1 \curlyvee \ldots \curlyvee K_m$ whose free variables are $x_1, \ldots, x_l$ and such that $\mathcal{S}(C) = \{L_1; \ldots; L_n\}$, we define the associated formula $\ulcorner C \urcorner = \forall x_1, \ldots, x_l.\ \underline{L_1} \vee^- \cdots \vee^- \underline{L_n} \vee^- \delta^-(K_1 \vee^+ \cdots \vee^+ K_m)$. $\ulcorner C \urcorner$ is said to be in clausal form. By extension, $\ulcorner \Gamma \urcorner$ is the set of the formulas associated to the clauses of the set Γ.

The main theorem of this article relates LKF$^\perp$ with Resolution with Input Selection:

▶ **Theorem 3.** *Let Γ be a set of clauses. We have $\ulcorner \Gamma \urcorner \Uparrow\vdash$ iff $\Gamma \rightsquigarrow^* \Box$.*

The proof can be found in the long version of the paper (`https://hal.inria.fr/hal-01670476`). To prove the right-to-left direction, we prove that all inference rules of Resolution with Input Selection are admissible in LKF$^\perp$, in the sense that if $\Gamma \rightsquigarrow C$ then LKF$^\perp$ proofs of $\ulcorner \Gamma \urcorner, \ulcorner C \urcorner \Uparrow\vdash$ can be turned into proofs of $\ulcorner \Gamma \urcorner \Uparrow\vdash$. Note that they are admissible, but they are not derivable. In particular, the size of the proof in LKF$^\perp$ can be much larger than the resolution derivation, as expected in a cut-free sequent calculus. Using cuts would lead to a closer correspondence between resolution derivations and sequent-calculus proofs, as in [15]. However, we chose to stay in the cut-free fragment to prove that, even in the incomplete case, resolution coincides with cut-free proofs, as in [25].

5 Complete Instances

5.1 Ordinary Focusing and Semantic Hyperresolution

As said earlier, in standard LKF, not all occurrences of literals can have an arbitrary polarity. Instead, each atom P is given globally a polarity, and $P^\perp$ has the opposite polarity.

Let us first look at the simple case where atoms are given a positive polarity. We recall the completeness proof of LKF:

▶ **Theorem 4** (Corollary of [26, Theorem 17])**.** *If the literals with a positive polarity are exactly the atoms, LKF$^\perp$ is (sound and) complete.*

If we look at the corresponding resolution calculus, Resolution with Selection for this particular instance becomes:

$$\text{R.w.S.}\ \frac{\neg P_1 \curlyvee \ldots \curlyvee \neg P_n \curlyvee C \qquad P_1' \curlyvee D_1 \quad \ldots \quad P_n' \curlyvee D_n}{\sigma(C \curlyvee D_1 \curlyvee \ldots \curlyvee D_n)}$$

where C and D_i for all i contain only positive literals, and σ is the most general unifier of $P_1 =^? P_1', \ldots, P_n =^? P_n'$. Note that the clause $\sigma(C \curlyvee D_1 \curlyvee \ldots \curlyvee D_n)$ contains only positive literals, so no literal would be selected in it even if it was an input clause. Besides, Resolution cannot be applied, since there exists no clause $\neg P \curlyvee C$ with $\mathcal{S}(\neg P \curlyvee C) = \emptyset$.

This corresponding resolution calculus is therefore exactly hyperresolution of [29]: premises of an inference contains all only positive literals, except one clause whose all negative literals are resolved at once. Theorem 3 therefore links ordinary focusing with hyperresolution. Consequently, Theorem 4 implies the completeness of hyperresolution.

Chaudhuri et al. [13, Theorem 16] prove a similar result by establishing a correspondence between hyperresolution derivations and proofs in a focused sequent calculus for intuitionistic linear logic, but only considering Horn clauses. In their setting, choosing a negative polarity for atoms leads to SLD resolution, which is the reasoning mechanism of Prolog.

Let us now look at the general case, where atoms are given an arbitrary polarity. Let us stick to the ground case. We first recall a refinement of resolution called Semantic hyperresolution [31][11, Sect. 1.3.5.3]. Let I be an arbitrary Herbrand interpretation, i.e. a model whose domain is the set of terms interpreted as themselves. Note that I is not assumed to be a model of the input set of clauses (which is fortunate, since one is trying to show that it is unsatisfiable). Given a clause C, the idea of semantic hyperresolution is to resolve all literals of C that are valid in I at once, with clauses whose literals are all not valid in I. This gives the rule:

$$\mathsf{SHR}\ \frac{K_1 \curlyvee \ldots \curlyvee K_n \curlyvee C \qquad {K_1}^{\perp} \curlyvee D_1 \quad \ldots \quad {K_n}^{\perp} \curlyvee D_n}{C \curlyvee D_1 \curlyvee \ldots \curlyvee D_n}$$

where for all i, $I \models K_i$ (and thus $I \not\models K_i^{\perp}$), $I \not\models C$ and $I \not\models D_i$. Note that $I \not\models C \curlyvee D_1 \curlyvee \ldots \curlyvee D_n$.

Semantic hyperresolution for a Herbrand interpretation I can be seen as an instance of Resolution with Input Selection by using the following polarization of atoms: a literal L has a negative polarity iff $I \models L$. In that case, SHR corresponds exactly to Resolution with Selection, and Resolution cannot be applied since we cannot have clauses $P \curlyvee C$ and $\neg P \curlyvee D$ where both P and $\neg P$ are not valid in I.

This particular instance of polarization is in fact the ordinary version of focusing. Indeed, once a global polarity is assigned to each atom, the set of literals whose polarity is negative defines an Herbrand interpretation, and we saw reciprocally how to design a global polarization from the Herbrand interpretation. Theorem 3 therefore links ordinary focusing in the ground case with semantic hyperresolution. They are both complete, thanks to this theorem:

▶ **Theorem 5** (Corollary of [26, Theorem 17]). *Given a global polarization of atoms, where the polarity of $P^{\perp}$ is the opposite of that of P, LKF$^{\perp}$ is (sound and) complete.*

5.2 Deduction Modulo Theory

Deduction Modulo Theory [20] is a framework that consists in applying the inference rules of an existing proof system modulo some congruence over formulas. This congruence represents the theory, and it is in general defined by means of rewriting rules. To be expressive enough, these rules are defined not only at the term level, but also for formulas. To get simpler presentations of theories, we distinguish between rewrite rules that can be applied at positive and at negative positions by giving them a polarity[3], where by negative position we mean under an odd number of $\neg$. We therefore have positive rules $P \rightarrow^{+} A$ and negative rules $P \rightarrow^{-} A$ where P is an atom and A an arbitrary formula whose free variables appears in P. Given a rule $P \rightarrow^{+} A$, the rewrite relation $B_1 \xrightarrow{+} B_2$ is defined as usual by saying that there exists a position $\mathfrak{p}$ and a substitution σ such that the subformula of B_1 at position $\mathfrak{p}$ is σP and B_2 equals B_1 where the subformula at position $\mathfrak{p}$ is replaced by σA. $\xrightarrow{-}$ is defined similarly. In Polarized Sequent Calculus Modulo theory [17], the inference rules of the sequent calculus are applied modulo such a polarized rewriting system, as in for instance in $\vdash\wedge\ \dfrac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash C, \Delta}\ C \xrightarrow{+}{}^{*} A \wedge B$. Note that the implicit semantics of a negative

[3] This polarity must not be confused with the other notions of polarity mentioned in the paper.

$$\frac{}{\Gamma, L, L^\perp \vdash}\ \frown\!\vdash \qquad \frac{\Gamma \vdash}{\Gamma, \top \vdash}\ \top\vdash \qquad \frac{}{\Gamma, \bot \vdash}\ \bot\vdash \qquad \frac{\Gamma, A \vdash \quad \Gamma, B \vdash}{\Gamma, A \vee B \vdash}\ \vee\vdash$$

$$\frac{\Gamma, A, B \vdash}{\Gamma, A \wedge B \vdash}\ \wedge\vdash \qquad \frac{\Gamma, A \vdash}{\Gamma, \exists x.\ A \vdash}\ \exists\vdash\ x \text{ not free in } \Gamma \qquad \frac{\Gamma, \forall x.\ A, \{t/x\}A \vdash}{\Gamma, \forall x.\ A \vdash}\ \forall\vdash$$

$$\frac{\Gamma, P, A \vdash}{\Gamma, P \vdash}\ \uparrow^-\vdash\ P \xrightarrow{-} A \qquad \frac{\Gamma, \neg P, A^\perp \vdash}{\Gamma, \neg P \vdash}\ \uparrow^+\vdash\ P \xrightarrow{+} A$$

Figure 3 The sequent calculus PUSC$^\perp$.

rule $P \to^- A$ is therefore $\overline{\forall x}.\ (P \Rightarrow A)$, whereas the semantics of $P \to^+ A$ is $\overline{\forall x}.\ (A \Rightarrow P)$, where $\overline{x}$ are the free variables of P.

With Kirchner [9], we proved the equivalence of Polarized Sequent Calculus Modulo theory to a sequent calculus where polarized rewriting rules are applied only on literals, using explicit rules. This calculus, Polarized Unfolding Sequent Calculus, is almost the calculus PUSC$^\perp$ presented in Figure 3. The only difference is that all formulas are put on the left of the sequent in PUSC$^\perp$. We denote by $\Gamma \vdash_{\mathcal{R}}$ the fact that $\Gamma \vdash$ can be proved in PUSC$^\perp$ using the polarized rewriting system $\mathcal{R}$. Note that the rule for the universal quantifier $\forall\vdash$ as well as the unfolding rules $\uparrow^-\vdash$ and $\uparrow^+\vdash$ contain an implicit contraction rule, as in the sequent calculus G4 of Kleene, in order to ensure that all rules of PUSC$^\perp$ are invertible.

We can translate polarized rewriting rules as formulas with selection, and see PUSC$^\perp$ as an instance of LKF$^\perp$. We first consider how to translate formulas of the right-hand side of polarized rewriting rules. We polarize them by choosing positive connectives for $\vee$ and $\wedge$ and, to unchain the introduction of the universal quantifier, we introduce delays. (Let us recall that a delay δ^+ allows to force a formula to be positive, and it can be encoded using an existantial quantifier.) This gives the translation:

$$\begin{array}{lll} |L| = L \quad \text{when } L \text{ is } \top, \bot \text{ or a literal} & & |A \wedge B| = |A| \wedge^+ |B| \\ |A \vee B| = |A| \vee^+ |B| & |\exists x.\ A| = \exists x.\ |A| & |\forall x.\ A| = \forall x.\ \delta^+|A| \end{array}$$

▶ **Definition 6.** Given a negative rewriting rule $P \to^- A$ where the free variables of P are $x_1, \dots, x_n$, its translation as a formula with selection is $\llbracket P \to^- A \rrbracket = \forall x_1 \dots x_n.\ \underline{\neg P} \vee^- \delta^+|A|$.

Given a positive rewriting rule $P \to^+ A$ where the free variables of P are $x_1, \dots, x_n$, its translation as a formula with selection is $\llbracket P \to^+ A \rrbracket = \forall x_1.\ \dots \forall x_n.\ \underline{P} \vee^- \delta^+|A^\perp|$.

The translation $\llbracket \mathcal{R} \rrbracket$ of a polarized rewriting system $\mathcal{R}$ is the multiset of the translation of its rules.

▶ **Definition 7.** Let $N_1, \dots, N_n$ be a multiset of formulas whose top connective is $\forall$ or $\bot$ or that are literals, and let $P_1, \dots, P_m$ be a multiset of non-literal formulas whose top connective is neither $\forall$ nor $\bot$, then the translation of the PUSC$^\perp$ sequent $N_1, \dots, N_n, P_1, \dots, P_m \vdash$ modulo the rewriting system $\mathcal{R}$ is the LKF$^\perp$ sequent $\llbracket \mathcal{R} \rrbracket, |N_1|, \dots, |N_n| \Uparrow |P_1|, \dots, |P_m| \vdash$.

▶ **Theorem 8.** *With the same assumptions as previous definition, $N_1, \dots, N_n, P_1, \dots, P_m \vdash_{\mathcal{R}}$ in PUSC$^\perp$ iff $\llbracket \mathcal{R} \rrbracket, |N_1|, \dots, |N_n| \Uparrow |P_1|, \dots, |P_m| \vdash$ in LKF$^\perp$.*

The proof can be found in the long version of the paper.

Let us now consider the subcase where the rewriting rules are clausal, according to the terminology of [19], e.g. they are of the form $P \to^- C$ or $P \to^+ \neg C$ for some formula C in clausal normal form. In that case, the resolution method based on Deduction Modulo Theory [20] can be refined into what is called Polarized Resolution Modulo theory [19], whose rules are given in Fig. 4. (A refinement of) Polarized Resolution Modulo theory is actually implemented in the automated theorem prover iProverModulo [5].

$$\text{Resolution}\ \frac{P \curlyvee C \qquad \neg Q \curlyvee D}{\sigma(C \curlyvee D)}\ ^{a} \qquad \text{Factoring}\ \frac{L \curlyvee K \curlyvee C}{\sigma(L \curlyvee C)}\ \sigma = mgu(L, K)$$

$$\text{Ext. Narr.}^{-}\ \frac{P \curlyvee C}{\sigma(D \curlyvee C)}\ ^{a},\ Q \to^{-} D \qquad \text{Ext. Narr.}^{+}\ \frac{\neg Q \curlyvee D}{\sigma(C \curlyvee D)}\ ^{a},\ P \to^{+} \neg C$$

[a] $\sigma = mgu(P, Q)$

Figure 4 Inference rules of Polarized Resolution Modulo theory.

By noting that the translation of the rule $Q \to^{-} D$ is $[\![Q \to^{-} D]\!] = \forall x_1.\ \ldots \forall x_n.\ \underline{\neg Q} \vee^{-} \delta^{+}|D|$ whereas $\ulcorner \underline{\neg Q} \curlyvee D \urcorner = \forall x_1.\ \ldots \forall x_n.\ \underline{\neg Q} \vee^{-} \delta^{-}|D|$, we can relate the rule $Q \to^{-} D$ with the clause with selection $\underline{\neg Q} \curlyvee D$, which is called a one-way clause by Dowek [19]. Indeed, the change of phase is always needed in that particular case, so that the delays are in fact useless. Ext. Narr.$^{-}$ can therefore be seen as an instance of the Resolution with Selection rule:

$$\text{Resolution with Selection}\ \frac{\underline{\neg Q} \curlyvee D \qquad P \curlyvee C}{\sigma(D \curlyvee C)}\ \sigma = mgu(P, Q)\ .$$

Similarly, $P \to^{+} \neg C$ is related to $\underline{P} \curlyvee C$.

Consequently, since PUSC$^{\perp}$ corresponds to LKF$^{\perp}$, and Resolution with Input Selection corresponds to Polarized Resolution Modulo theory, Theorem 3 leads to a new and more generic proof of the correspondence between PUSC$^{\perp}$ and Polarized Resolution Modulo theory.

Deduction Modulo Theory is not always complete. This is the case only if the cut rule is admissible in Polarized Sequent Calculus Modulo theory. It holds for some particular theories, e.g. Simple Type Theory [20] and arithmetic [22]. There are more or less powerful techniques that ensures this property [24, 21, 18, 7]. We even proved that any consistent first-order theory can be presented by a rewriting system admitting the cut rule [6]. As presented with Dowek [8] and discussed in the introduction, the fact that completeness is not proved once for all, but needs to be proved for each particular theory, is essential. Indeed, if a theory is presented entirely by rewriting rules, completeness implies the consistency of the theory, since no rule can be applied on the empty set of clauses. Consequently, the proof of the completeness cannot be easier than the proof of consistency of the theory, and, according to Gödel, cannot be proven in the theory itself.

5.3 Beyond Deduction Modulo Theory

We now consider the general case where several literals are selected in a clause, and show how proving completeness in LKF$^{\perp}$ can be reduced to proving completeness of several systems in Deduction Modulo Theory.

▶ **Example 9.** Let us recall the set of clauses from the Introduction:

$$\underline{\neg X \in \mathcal{P}(Y)} \curlyvee \underline{\neg Z \in X} \curlyvee Z \in Y \quad (1) \qquad \underline{X \in \mathcal{P}(Y)} \curlyvee d(X, Y) \in X \quad (2)$$

$$\underline{X \in \mathcal{P}(Y)} \curlyvee \underline{\neg d(X, Y) \in Y} \quad (3)$$

Note that this example is not covered by Ordered Resolution with Selection, at least not if a simplification ordering is used, because we cannot have $X \in \mathcal{P}(Y) \succ \delta(X, Y) \in X$ since with $\theta = \{X \mapsto \mathcal{P}(Z); Y \mapsto Z\}$ their instances are ordered in the wrong direction: $\mathcal{P}(Z) \in \mathcal{P}(Z) \prec \delta(\mathcal{P}(Z), Z) \in \mathcal{P}(Z)$.

The synthetic rules of the example from the Introduction correspond to the derivations when one of the clauses is focused. For instance, if we consider the clause (1), in a context

Γ containing this clause, a proof putting the focus on $\ulcorner(1)\urcorner$ necessarily is of the following shape:

$$
\dfrac{\dfrac{\dfrac{}{\Gamma \Downarrow \underline{\neg u \in \mathcal{P}(v)} \vdash}\ \widehat{\Downarrow}\vdash \qquad \dfrac{}{\Gamma \Downarrow \underline{\neg t \in u} \vdash}\ \widehat{\Downarrow}\vdash \qquad \dfrac{\dfrac{\Gamma, t \in v \Uparrow \vdash}{\Gamma \Uparrow t \in v \vdash}\ \textsf{Store}}{\Gamma \Downarrow t \in v \vdash}\ \textsf{Release}}{\dfrac{\Gamma \Downarrow \underline{\neg u \in \mathcal{P}(v)} \vee^- \underline{\neg t \in u} \vee^- t \in v \vdash}{\Gamma \Downarrow \forall X\ Y\ Z.\ \underline{\neg X \in \mathcal{P}(Y)} \vee^- \underline{\neg Z \in X} \vee^- Z \in Y \vdash}\ \Downarrow\forall\vdash}\ \Downarrow\vee\vdash}{\Gamma \Uparrow \vdash}\ \textsf{Focus}
$$

where t, u, v are arbitrary terms, and where, to be able to close the left and middle branches, $u \in \mathcal{P}(v)$ and $t \in u$ must belong to Γ. So Γ is in fact of the form $\forall X\ Y\ Z.\ \underline{\neg X \in \mathcal{P}(Y)} \vee^+ \underline{\neg Z \in X} \vee^+ Z \in Y, \Delta, u \in \mathcal{P}(v), t \in u$ for some Δ, and the axiom $\forall X\ Y\ Z.\ \underline{\neg X \in \mathcal{P}(Y)} \vee^+ \underline{\neg Z \in X} \vee^+ Z \in Y$ can be replaced by the synthetic rule:

$$
(1)\vdash\ \dfrac{\Delta, u \in \mathcal{P}(v), t \in u, t \in v \Uparrow \vdash}{\Delta, u \in \mathcal{P}(v), t \in u \Uparrow \vdash}\ .
$$

The computation of the other synthetic rules is left as an exercise for the reader.

The resolution rules given in the Introduction corresponds to the ground instances of Resolution with Selection with our three input clauses.

The question that remains is how we can prove the completeness of such a selection. We can in fact consider only subselections.

▶ **Definition 10** (Singleton subselection)**.** Given a selection function $\mathcal{S}$, the selection function $\mathcal{S}_1$ is a singleton subselection of $\mathcal{S}$ if

- $\mathcal{S}_1(C) \subseteq \mathcal{S}(C)$ for all C
- if $\mathcal{S}(C) \neq \emptyset$ then $\mathrm{card}(\mathcal{S}_1(C)) = 1$

▶ **Example 11.** A singleton subselection of Example 9 can be

$$\neg X \in \mathcal{P}(Y) \curlyvee \underline{\neg Z \in X} \curlyvee Z \in Y \qquad \underline{X \in \mathcal{P}(Y)} \curlyvee d(X,Y) \in X \qquad \underline{X \in \mathcal{P}(Y)} \curlyvee \neg d(X,Y) \in Y$$

▶ **Theorem 12.** *Resolution with input selection $\mathcal{S}$ is complete iff for all singleton subselections $\mathcal{S}_1$ of $\mathcal{S}$, Resolution with input selection $\mathcal{S}_1$ is complete.*

The proof can be found in the long version of the paper.

Since singleton subselections can be linked with rewriting systems in Deduction Modulo Theory according to last subsection, we can reduce the problem of completeness in our framework to several problems of completeness in Deduction Modulo Theory.

Conclusion and Further Work

We generalized focusing and resolution with selection, proved that they correspond, and showed how known calculi are instances of this framework, namely ordinary focusing, hyperresolution and Deduction Modulo Theory. In the long version of the paper, other frameworks, such as Superdeduction [4] or Schroeder-Heister's Definitional reflection [30], are also considered. Furthermore, we presented how to reduce completeness of this framework to several completeness proofs in Deduction Modulo Theory. We can therefore reuse the various techniques for proving completeness in Deduction Modulo Theory [24, 21, 18, 7] in our framework. As Deduction Modulo Theory already gives significant results in industrial applications when the theory is a variant of set theory (more precisely, set theory of the B method) [10], we can expect our framework to lead to even better outcomes. The notable results presented here raise the following new areas of investigations.

First, we need to study how to apply selection also in the generated clauses. This should allow us to cover the cases of Ordered Resolution with Selection and of Semantic Resolution in the first-order case. Dually, in the sequent calculus part, this would correspond to the possibility to dynamically add selection in formulas of subderivations. This could probably be linked with the work of Deplagne [16] where rewrite rules corresponding to induction hypotheses are dynamically added in the rewriting system of a sequent calculus for Deduction Modulo Theory. Note that we already have one direction, namely from Resolution with Input Selection to $\mathrm{LKF}^{\perp}$, since the proof for this direction (see the long version) does not assume anything on the generated clauses; except, for Factoring, that it selects only instances of literals that were already selected. The converse direction would require a meta-theorem of completeness, since obviously it is not complete for all possible dynamic choices of selection.

Since focusing is defined not only for classical first-order logic but also for linear, intuitionistic, modal logics, the work in this paper could serve as a starting point to study how to get automated proof search methods for these logics with a selection mechanism.

Another worthwhile point is how equality should be handled in our framework. In particular, it would be interesting to see how paramodulation calculi, in particular superposition, can be embedded into a sequent calculus.

Finally, it would be worth investigating whether completeness proofs based on model construction, such as semantic completeness proofs of tableaux (related to sequent calculus), and completeness proof of superposition [2], can be linked in our framework.

References

1 Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

2 L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):1–31, 1994.

3 Leo Bachmair and Harald Ganzinger. Resolution theorem proving. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 19–99. Elsevier and MIT Press, 2001.

4 Paul Brauner, Clément Houtmann, and Claude Kirchner. Principle of superdeduction. In Luke Ong, editor, *LICS*, pages 41–50, 2007.

5 Guillaume Burel. Experimenting with deduction modulo. In Viorica Sofronie-Stokkermans and Nikolaj Bjørner, editors, *CADE*, volume 6803 of *LNCS*, pages 162–176. Springer, 2011.

6 Guillaume Burel. From axioms to rewriting rules. Available on author's web page, 2013.

7 Guillaume Burel. Cut admissibility by saturation. In Gilles Dowek, editor, *RTA-TLCA*, volume 8560 of *LNCS*, pages 124–138. Springer, 2014.

8 Guillaume Burel and Gilles Dowek. How can we prove that a proof search method is not an instance of another? In *LFMTP*, ACM International Conference Proceeding Series, pages 84–87. ACM, 2009.

9 Guillaume Burel and Claude Kirchner. Regaining cut admissibility in deduction modulo using abstract completion. *Information and Computation*, 208(2):140–164, 2010.

10 Guillaume Bury, David Delahaye, Damien Doligez, Pierre Halmagrand, and Olivier Hermant. Automated deduction in the B set theory using typed proof search and deduction modulo. In Ansgar Fehnker, Annabelle McIver, Geoff Sutcliffe, and Andrei Voronkov, editors, *LPAR*, volume 35 of *EPiC Series in Computing*, pages 42–58. EasyChair, 2015.

11 Samuel R. Buss, editor. *Handbook of proof theory*. Studies in logic and the foundations of mathematics. Elsevier, Amsterdam, 1998.

12 Kaustuv Chaudhuri and Frank Pfenning. A focusing inverse method theorem prover for first-order linear logic. In Robert Nieuwenhuis, editor, *CADE*, volume 3632 of *LNCS*, pages 69–83. Springer, 2005.

13 Kaustuv Chaudhuri, Frank Pfenning, and Greg Price. A logical characterization of forward and backward chaining in the inverse method. *Journal of Automated Reasoning*, 40(2-3):133–177, 2008.

14 Zakaria Chihani, Tomer Libal, and Giselle Reis. The proof certifier checkers. In *TABLEAUX*, volume 9323 of *LNCS*, pages 201–210, Wroclaw, Poland, 2015. Springer.

15 Zakaria Chihani, Dale Miller, and Fabien Renaud. Foundational proof certificates in first-order logic. In Maria Paola Bonacina, editor, *CADE*, volume 7898 of *LNCS*, pages 162–177. Springer, 2013.

16 Eric Deplagne and Claude Kirchner. Induction as deduction modulo. Rapport de recherche, LORIA, Nov 2004. URL: `https://hal.inria.fr/LORIA/inria-00099871`.

17 Gilles Dowek. What is a theory? In Helmut Alt and Afonso Ferreira, editors, *STACS*, volume 2285 of *LNCS*, pages 50–64. Springer, 2002.

18 Gilles Dowek. Truth values algebras and proof normalization. In Thorsten Altenkirch and Conor McBride, editors, *TYPES*, volume 4502 of *LNCS*, pages 110–124. Springer, 2006.

19 Gilles Dowek. Polarized resolution modulo. In Cristian S. Calude and Vladimiro Sassone, editors, *IFIP TCS*, volume 323 of *IFIP AICT*, pages 182–196. Springer, 2010.

20 Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1):33–72, 2003.

21 Gilles Dowek and Benjamin Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.

22 Gilles Dowek and Benjamin Werner. Arithmetic as a theory modulo. In Jürgen Giesl, editor, *RTA*, volume 3467 of *LNCS*, pages 423–437. Springer, 2005.

23 Mahfuza Farooque, Stéphane Graham-Lengrand, and Assia Mahboubi. A bisimulation between DPLL(T) and a proof-search strategy for the focused sequent calculus. In Alberto Momigliano, Brigitte Pientka, and Randy Pollack, editors, *LFMTP*, pages 3–14. ACM, 2013.

24 Olivier Hermant. *Méthodes Sémantiques en Déduction Modulo*. PhD thesis, École Polytechnique, 2005.

25 Olivier Hermant. Resolution is cut-free. *Journal of Automated Reasoning*, 44(3):245–276, 2009.

26 Chuck Liang and Dale Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoretical Computer Science*, 410(46):4747–4768, 2009. Abstract Interpretation and Logic Programming: In honor of professor Giorgio Levi.

27 Sean McLaughlin and Frank Pfenning. Imogen: Focusing the polarized inverse method for intuitionistic propositional logic. In Iliano Cervesato, Helmut Veith, and Andrei Voronkov, editors, *LPAR*, volume 5330 of *Lecture Notes in Computer Science*, pages 174–181. Springer, 2008.

28 Dale Miller and Marco Volpe. Focused labeled proof systems for modal logic. In Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov, editors, *LPAR*, volume 9450 of *LNCS*, pages 266–280. Springer, 2015.

29 J. A. Robinson. Automatic deduction with hyper-resolution. *International Journal of Computer Mathematics*, 1:227–234, 1965.

30 Peter Schroeder-Heister. Cut elimination for logics with definitional reflection. In *Nonclassical Logics and Information Processing*, volume 619 of *LNCS*, pages 146–171. Springer, 1990.

31 James R. Slagle. Automatic theorem proving with renamable and semantic resolution. *J. ACM*, 14(4):687–697, 1967.

32 Larry Wos, George A. Robinson, and Daniel F. Carson. Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM*, 12(4):536–541, 1965.