Clausal Completion

Francois Bronsard*
Andersen Consulting,
Northbrook, IL 60062, USA.

Uday S. Reddy
Dept. of Computer Science
The University of Illinois at Urbana-Champaign

January 11, 1996

Abstract

We develop a new deduction method for clausal deduction inspired by the Knuth-Bendix completion method for equational reasoning. Our *clausal completion* method combines a powerful simplification method based on the notion of *reductive proofs* and a restricted forward deduction technique based on the notion of *critical consequences*. Reductive proofs are the clausal counterpart of *rewrite proofs* for equational deduction. They offer a decidable and powerful, albeit incomplete, deduction technique well-suited for simplification. Critical consequences are the clausal counterpart of *critical pairs*. Given a set of axioms and some theorems, the completion method *completes* the set of axioms by adding its critical consequences until the theorems have reductive proofs. This method is sound, and with some constraints, complete. We illustrate the method with non-trivial examples and give an overview of its completeness proof.
Area: MECHANISMS - resolution, rewriting

1 Introduction

Our objective in this work is to develop a new deduction technique for first-order clausal reasoning. We take inspiration from a successful method for equational reasoning: the Knuth-Bendix *completion method* [1970]. This method addresses problems of the following form: Given a set of equations $\mathcal{E}$—the *theory*—and a ground equation $a = b$—the *goal*—is $a = b$ a consequence of $\mathcal{E}$, *i.e.*, does $\mathcal{E} \models a = b$ hold? This is called "the word problem for the theory $\mathcal{E}$." The Knuth-Bendix completion method proceeds as follows:

- First, one selects a well-founded order $\succ$ over terms.
- Then, the set $\mathcal{E}$ is *completed* by adding its *critical consequences*.
- In the process, the original equations of $\mathcal{E}$, or the critical consequences being added to $\mathcal{E}$, can be deleted or simplified if they can be proved from other equations of $\mathcal{E}$ by a *rewrite proof*.

*Work supported by Ministère des Affaires Etrangères, France. Part of this work was done when visiting INRIA-Lorraine, Nancy, France; and part of this work was done when working at The University of Illinois at Urbana-Champaign, USA.

- Finally, let $\mathcal{E}_C$ be the resulting *complete*, or *canonical*, set. $a = b$ is a consequence of $\mathcal{E}$ if and only if there is a rewrite proof of $a = b$ using the equations of $\mathcal{E}_C$.

This method is sound, and with some additional constraints, it is complete. It has proved to be very successful for solving word problems in equational theories (see [Dershowitz and Jouannaud, 1990] for a survey of completion and of term orders). This method can also be used to test the inconsistency of a set of equations and inequations [Bachmair *et al.*, 1989]. Finally, one can show that for a given goal, a finite set of critical consequences of the given theory suffices to obtain a rewrite proof of the goal [Bachmair *et al.*, 1986]. Thus, the search for a rewrite proof of the goal can be interleaved with the generation of $\mathcal{E}_C$ to obtain a semi-decision procedure.

The main practical benefit of the completion approach is that it divides the deduction process into a "cheap" mechanism, viz., rewriting, and an "expensive" mechanism, viz., critical consequence generation. Rewriting is "cheap" in that it terminates finitely, involves no search, and uses pattern matching instead of unification. The completion process uses the cheap mechanism as much as possible to reduce the work of the expensive mechanism. A second benefit of the completion approach is that it allows "theory building." In many applications of automated deduction, e.g., program verification, several theorems are routinely proved using the same basic theory. Since the completion process works solely on the the theory, independent of the goal, the results of the completion process can be reused for proving other goals. Thus, completion often thought of as a "compilation" phase, which makes the given theory more efficient for problem solving.

Our method, called the *clausal completion method*, closely follows the Knuth-Bendix completion method outlined above. The deduction problems we want to address have the following form: Given a set of clauses $\mathcal{T}$—the *theory*—and a ground clause Γ—the *goal*—is Γ a consequence of $\mathcal{T}$, *i.e.*, does $\mathcal{T} \models \Gamma$ hold? We call this the "clausal word problem for $\mathcal{T}$." The clausal completion method proceeds as follows:

- First, one selects a well-founded order $\succ$ over literals.
- Then, the set $\mathcal{T}$ is *completed* by adding its *critical consequences*.
- In the process, the original clauses of $\mathcal{T}$, or the critical consequences being added to $\mathcal{T}$, can be deleted or simplified if they can be proved from other clauses of $\mathcal{T}$ by a *reductive proof*, where *reductive proofs* are the clausal counterparts of rewrite proofs.
- Finally, let $\mathcal{T}_C$ be the resulting *complete* set. Γ is a consequence of $\mathcal{T}$ if and only if there is a reductive proof of Γ using the clauses of $\mathcal{T}_C$.

As with the Knuth-Bendix completion method, our clausal completion method is sound, and with some additional constraints, it is complete. Further, the method can be used to test the inconsistency of the original theory $\mathcal{T}$. Thus, it is refutationally complete. Finally, for any given goal, a finite set of critical consequences of the given theory suffices to obtain a reductive proof of that goal. Hence, the search for a reductive proof of the goal can be interleaved with the generation of $\mathcal{T}_C$ to obtain a semi-decision procedure.

¿From a practical point of view, this method brings the benefits of the completion method into the realm of clausal theorem proving. The deduction process is divided into a "cheap" mechanism, viz., reductive proof, and "expensive" mechanism, viz., critical consequence generation. Moreover, the benefits of "compilation" and "theory building" become available.

The main contributions of this paper are the formal definitions of reductive proofs, critical consequences and the completion procedure. The main difference between our proposal and previous adaptations of the Knuth-Bendix completion method rests on our notion of reductive deduction. We propose a powerful, yet practical, notion of reductive deduction without sacrificing completeness. Our notion of complete sets is also original. Previous works were mostly concerned with establishing the refutational completeness of the inferences used to generate critical consequences. Although refutational completeness is all that is necessary to prove a single theorem, if one wants to prove multiple theorems, then the construction of a complete, or partially complete, set may be more profitable. We should, however, point out that since few mathematical theories are decidable, most complete sets are infinite. Hence, in general, we can only generate a "partially complete" set. Finally, another significant technical contribution is the technique we have used to prove the completeness of the completion method. By relying on a graph representation of proofs, we have obtained a simple proof-theoretic completeness proof very similar to Bachmair, Dershowitz, and Hsiang's [1986] completeness proof for the equational completion method. This technique is dicussed in detail in Bronsard [1995].

This paper is organized as follows: We discuss related works below. In Section 2 we recall the classical inference rules of clausal deduction and the completeness results for those rules. Section 3 defines reductive proofs while Section 4 defines critical consequences and the completion procedure. Section 5 gives two non-trivial examples to illustrate the method. Finally, Section 6 gives an overview of the completeness proof of the method.

1.1 Related Works

Ours is not the first attempt to adapt the equational completion procedure for clausal deduction. We take inspiration from Hsiang and Rusinowitch [1986] who proposed *ordered resolution* and *ordered paramodulation* to generate critical consequences and proved the refutational completeness of these inferences. That work formed the basis for many subsequent works on clausal completion, but it did not address the issue of simplification.

Zhang and Kapur [1988] developed a practical clausal completion-like method using *superposition inferences* to generate critical consequences, and a *contextual rewriting* technique for simplification. This work offers both a restricted method to generate critical consequences and a powerful simplification mechanism. However, this approach does not guarantee completeness. Superposition with tautology deletion is incomplete [Bachmair and Ganzinger, 1994] and it is unclear what effect simplification by contextual rewriting would have on the completeness of a deduction method.

Independently, Nieuwenhuis and Orejas [1991] developed another practical clausal completion method using ordered inferences and a technique called *clausal rewriting* for simplification. This method is refutationally complete for clauses with at most one equational literal. With some additional constraints on clauses, it is also complete.

The clausal rewriting technique defined by Nieuwenhuis and Orejas relies on *nondeterministic* instantiation of clauses. In [Bronsard and Reddy, 1992], we improved upon their method by relying on hyper-resolution rather than nondeterministic instantiation. However, this extension did not address equational reasoning.

The clausal completion procedure reported here improves on these earlier works by guaranteeing completeness and proposing a stronger simplification mechanism.

Bachmair and Ganzinger [1994] presented a thorough study of the superposition inference in the context of conditional equations. Their approach, like ours, is inspired by the Knuth-Bendix completion procedure. However, they do not attempt to develop a clausal completion procedure, but rather concentrate on the refutational capability of deduction using ordered inferences. Their main result shows that *ordered superposition* and *merging paramodulation* form a refutationally complete deduction system. Further, this completeness property holds even if clauses are deleted or simplified using the following rule:

$$\text{Simplify} \quad \frac{(\mathcal{T} \cup \{\Gamma\})}{(\mathcal{T} \cup \mathcal{T}')} \quad \text{if } \mathcal{T} \cup \Gamma \models \mathcal{T}' \text{ and } \mathcal{T}'_{<\Gamma} \models \Gamma \ .$$

where $\mathcal{T}'_{<\Gamma} = \{\Delta\theta \mid \Delta \in \mathcal{T}' \ \wedge \ \Delta\theta < \Gamma\}$.

The notion of critical consequences used in this work is weaker than Bachmair and Ganzinger's restriction to ordered superposition and merging paramodulation. However, this relaxation is necessary to obtain complete sets. Our notion of reductive proof also offers a practical, rather than theoretical, simplification method. It satisfies the condition of Bachmair and Ganzinger's Simplify rule. Thus, for a given theory, if one wishes to prove only one theorem, the optimal approach would be to use Bachmair and Ganzinger's inference rules together with our powerful simplification method. Conversely, if multiple theorems need to be proved, using our method to build a complete, or partially complete, set would probably be preferable.

2 Clausal inferences

In this section we briefly review background on clausal deduction techniques. Consider the following inference rules:

$$\begin{array}{ccc} \text{Cut} & \text{Contraction} & \text{Weakening} \\ \dfrac{\neg A\,\Gamma \quad A\,\Delta}{\Gamma\,\Delta} & \dfrac{A\,A\,\Gamma}{A\,\Gamma} & \dfrac{\Gamma}{A\,\Gamma} \end{array}$$
$$\begin{array}{ccc} \text{Resolution} & \text{Factoring} & \\ \dfrac{\neg A'\,\Gamma \quad A\,\Delta}{\Gamma\sigma\,\Delta\sigma}\,(\text{Res}) & \dfrac{A\,A'\,\Gamma}{A\sigma\,\Gamma\sigma}\,(\text{Fac}) & \text{where } \sigma = \text{mgu}(A, A') \end{array} \tag{1}$$

For each rule we identify a *principal literal*. For the Cut, Contraction, and Weakening inferences, the principal literal is A. For the Resolution and Factoring inferences, the principal literal is $A\sigma$.

The soundness and completeness of these inference rules are available from classical work [Robinson, 1965]

Previous works on adapting the Knuth-Bendix completion method to first-order logic have relied on the notion of *ordered inferences* introduced by Hsiang and Rusinowitch [1986]:

Definition 2.1 [Hsiang and Rusinowitch, 1986] Let $\succ$ be a well-founded order over literals. The following terminology applies to inferences (1):

- A Cut inference is *ordered* if $\neg A$ and A are maximal (w. r. to the order $\succ$) in their respective premises.
- A Contraction inference is *ordered* if A is maximal in the premise.
- A Resolution inference is *ordered* if $\neg A'\sigma$ is maximal in $\neg A'\sigma\,\Gamma\sigma$, and $A\sigma$ is maximal in $A\sigma\,\Delta\sigma$.
- A Factoring inference is *ordered* if $A\sigma$ is maximal in $A\sigma\,\Gamma\sigma$.

Theorem 2.1 [Hsiang and Rusinowitch, 1986] A set of clauses is unsatisfiable if and only if the empty clause $\bot$ is derivable using ordered Resolution and ordered Factoring inferences.

Reasoning with equations

To reason with equations, the following inference rules are added:

$$\begin{array}{cc} \text{Paramodulation} & \text{Reflection} \\ \dfrac{A[s']\,\Gamma \quad s = t\,,\Delta}{A\sigma[t\sigma]\,\Gamma\sigma\,\Delta\sigma}\,(\text{Par}) & \dfrac{\neg(s = s')\,\Delta}{\Delta\sigma}\,(\text{Ref}) \quad \text{where } \sigma = \text{mgu}(s, s') \end{array} \tag{2}$$

The principal literal of the Paramodulation inference is $A[s']\sigma$. The equational literal of the Paramodulation inference is $s\sigma = t\sigma$ and the principal literal of the Reflection inference is $s\sigma = s'\sigma$. The literal $A\sigma[t\sigma]$ occurring in the conclusion of the Paramodulation inference is called a *derived* literal.

The soundness and refutational completeness of these inference rules are again available from classical work [Robinson and Wos, 1969]

The notion of ordered inferences can be adapted to Paramodulation and Reflection inferences. However, for Paramodulation inferences there are two variants of the definition.

Definition 2.2 [Hsiang and Rusinowitch, 1986] Let $\succ$ be a well-founded order over terms and literals. The following terminology applies to inferences (2):

- A paramodulation step is *oriented* if $s\sigma \not\preceq t\sigma$.
- A paramodulation step is *ordered* if (i) it is oriented, (ii) s' is not a variable, (iii) $A[s']\sigma$ is maximal in $A[s']\sigma\,\Gamma\sigma$, and (iv) $(s\sigma = t\sigma)$ is maximal in $(s\sigma = t\sigma)\,\Delta\sigma$.
- A paramodulation step is called a *superposition* step if it is ordered, and whenever $A[s']$ is an equation of the form $l[s'] = r$, $l[s']\sigma \not\preceq r\sigma$.
- A reflection step is *ordered* if $\neg(s = t)\sigma$ is maximal in $\neg(s = t)\sigma\,\Delta\sigma$.

Theorem 2.2 [Hsiang and Rusinowitch, 1986] A set of clauses is unsatisfiable if and only if the empty clause is derivable using ordered Resolution, ordered Factoring, ordered Paramodulation, and ordered Reflection inferences.

3 Reductive proofs

The key part of our clausal completion method is the concept of *reductive proofs* which forms the clausal equivalent of rewrite proofs for equational deduction. Taking inspiration from the definition of rewrite proofs, we propose to define *reductive proofs* as proofs where each successive inference involve a smaller principal literal. For example, consider the following clauses:

$$S(x) \geq S(y) \quad \Leftarrow \quad x \geq y \tag{3}$$

$$S(x) \geq 0 \tag{4}$$

We use as order over literals the multiset path ordering $\succ_{mpo}$ based on the precedence

$$\text{“}\geq\text{”} \;>_p\; S \;>_p\; 0 \;>_p\; \neg \;>_p\; \bot$$

To prove the proposition $S(S(S(0))) \geq S(0)$, we construct the following refutation.

$$\dfrac{\dfrac{\neg(S(S(S(0))) \geq S(0))}{\neg(S(S(0)) \geq 0)}\ \text{Resolution with (3)}}{\bot}\ \text{Resolution with (4)}$$

We say that this proof is reductive since the successive principal literals are decreasing:

$$\neg(S(S(S(0))) \geq S(0)) \succ_{mpo} \neg(S(S(0)) \geq 0)$$

Formally, we define reductive proofs as follows:

Definition 3.1 Let T be a proof tree and $\succ$ a well-founded order over literals.

- A path in T is a *reductive path* if the principal literals of the non-factorization inference steps in the path are decreasing w. r. to $\succ$.
- T is a *reductive proof tree* if all the paths in T from the leaves to the root are reductive paths.
- T is a *reductive refutation* if it is a reductive proof whose conclusion is the empty clause.

Definition 3.2 Let T be a proof tree built using a theory $\mathcal{T}$ and a set of goal clauses $\mathcal{G}$.

- T is *flat* if no equational premise of any paramodulation step is a derived literal.
- T is *goal-oriented* if it is flat and all innermost non-factoring inferences involve a goal clause.

Definition 3.3 (Reductive Proofs) Let $\mathcal{T}$ be a set of clauses and $\succ$ a well-founded order over literals. A clause Γ is said to be *reductively provable*, or to have a *reductive proof*, from $\mathcal{T}$, written $\mathcal{T} \vdash^{\downarrow} \Gamma$, if there exists a goal-oriented, flat, reductive refutation built using the theory $\mathcal{T}$ and the set of goal clauses $\bar{\Gamma}$ where $\bar{\Gamma}$ denotes the skolemized negation of Γ.

Example 1 Consider the theory

$$x \in l \quad \Rightarrow \quad max(l) \geq x \tag{5}$$
$$x \geq y \wedge y \geq z \quad \Rightarrow \quad x \geq z \tag{6}$$
$$x \geq y \quad \vee \quad y \geq x \tag{7}$$

We select as order over literals the lexicographic path ordering, $\succ_{lpo}$, defined using the precedence $\in \, >_p \, \geq \, >_p \, max$. We can prove

$$\neg(max(l) \geq a) \Rightarrow (x \in l \Rightarrow a \geq x)$$

The skolemized negation of the goal is the set of clauses $\{\{\neg(max(L) \geq A)\}, \{X \in L\}, \{\neg(A \geq X)\}\}$ where L, A, X are the skolem constants replacing the variables l, a, x. The order is extended over the skolem constants by $L \; >_p \; A \; >_p \; X$. This gives us the reductive refutation

$$\dfrac{\dfrac{\dfrac{\dfrac{X \in L \quad (5)}{max(L) \geq X}\,\text{Res} \qquad (6)}{x \geq max(L) \Rightarrow x \geq X} \qquad \dfrac{\neg(max(L) \geq A) \quad (7)}{A \geq max(L)}\,\text{Res}}{A \geq X}\,\text{Res} \qquad \neg(A \geq X)}{\bot}\,\text{Res}$$

This is a reductive refutation since

$$X \in L \;\succ_{lpo}\; max(L) \geq X \;\succ_{lpo}\; A \geq max(L) \;\succ_{lpo}\; A \geq X \;, \quad \text{and}$$
$$max(L) \geq A \;\succ_{lpo}\; A \geq max(L) \;\succ_{lpo}\; A \geq X$$

Theorem 3.1 Let $\mathcal{T}$ be a set of clauses, Γ a ground clause, and $\succ$ a stable well-founded order over literals. Whether Γ has a reductive proof is decidable.

Clausal Church-Rosser property

The Church-Rosser property for a set of equations is the property that all consequences of that set have rewrite proofs. Since rewriting is decidable (assuming given a finite set of equations and a well-founded order), the Church-Rosser property, when it holds, ensures that equational deduction is decidable. Using the notion of reductive proofs, we can define a similar property for sets of clauses.

Definition 3.4 A set of clauses $\mathcal{T}$ is *Church-Rosser* if all ground clausal consequences of $\mathcal{T}$ have reductive proofs.

A set of clauses with the Church-Rosser property is called a *complete* set.

Theorem 3.2 Let $\mathcal{T}$ be a finite complete set of clauses. The clausal word problem for $\mathcal{T}$ is decidable.

4 The Completion procedure

The second part of the completion method is the *completion procedure*. This procedure transforms an arbitrary set of clauses into a set for which the Church-Rosser property holds. It does so by augmenting the set of clauses with their *critical consequences*.

Definition 4.1 Let $\succ$ be a well-founded order over literals.

- A Resolution inference is *critical* if $\neg A'$ and A are maximal in their respective premises.
- A Factoring inference is *critical* if A or A' is maximal in the premise.
- A Paramodulation inference is *critical* if (i) it is oriented, (ii) s' is not a variable, and (iii) $A[s']$ is maximal in both clauses.
- A Reflection inference is *critical* if $\neg(s = t)$ is maximal in $\neg(s = t), \Delta$.

Definition 4.2 (Clausal Completion Procedure) Let $\succ$ be a well-founded order over literals. The *clausal completion procedure* is defined by three state transition rules acting on two sets of clauses $\mathcal{T}$ and $\mathcal{R}$. The clauses in $\mathcal{T}$ can be simplified and deleted, while the clauses in $\mathcal{R}$ are *accepted clauses* that cannot be further simplified but can be freely used to simplify the clauses of $\mathcal{T}$. The state transition rules are

$$\text{Accept} \quad \frac{(\mathcal{T} \cup \{\Gamma\}, \mathcal{R})}{(\mathcal{T}, \mathcal{R} \cup \{\Gamma\})}$$

$$\text{Deduce} \quad \frac{(\mathcal{T}, \mathcal{R})}{(\mathcal{T} \cup \{\Gamma\}, \mathcal{R})} \quad \text{if } \Gamma \text{ is a critical consequence of } \mathcal{R} \cup \mathcal{T}$$

$$\text{Simplify} \quad \frac{(\mathcal{T} \cup \{\Gamma\}, \mathcal{R})}{(\mathcal{T} \cup \mathcal{T}', \mathcal{R})} \quad \text{if } \mathcal{R} \cup \mathcal{T} \cup \Gamma \models \mathcal{T}' \text{ and } \mathcal{R} \cup \mathcal{T}'_{\succ\Gamma} \vdash^{\downarrow} \Gamma \ .$$

A *completion derivation* is a sequence $(\mathcal{T}, \emptyset), (\mathcal{T}_1, \mathcal{R}_1), (\mathcal{T}_2, \mathcal{R}_2), \ldots$ such that for each i, $(\mathcal{T}_i, \mathcal{R}_i) \vdash (\mathcal{T}_{i+1}, \mathcal{R}_{i+1})$ by one of the above state transition rules. The set of *persisting* clauses is defined by $\mathcal{T}^\infty = \bigcup_i \bigcap_{j \geq i} \mathcal{T}_j$ and $\mathcal{R}^\infty = \bigcup_i \mathcal{R}_i$. A derivation is said to be *fair* if $\mathcal{T}^\infty = \emptyset$ and $\bigcup_i \mathcal{T}_i$ includes all the critical consequences of $\mathcal{R}^\infty$. If the derivation is fair, $\mathcal{R}^\infty$ is a complete axiomatization for the theory $\mathcal{T}$. The derivation is *successful* if $\mathcal{R}^\infty$ is finite. If a derivation is fair and successful, then $\mathcal{R}^\infty = \mathcal{R}_i$ for some i.

The clausal completion procedure is *sound*. That is, for any i, $\mathcal{T} \models \mathcal{T}_i \cup \mathcal{R}_i$. The procedure is also *complete* for ground clauses not containing inequations[1]. That is, if $\mathcal{T} \models \Gamma$, for some ground clause Γ not containing inequations, then, for some i, Γ is reductively provable using $\mathcal{R}^i$. This implies refutational completeness since, if a set of clauses is inconsistent, the empty clause, which is ground and does not contain inequations, must be deducible from it.

Theorem 4.1 The clausal completion procedure is sound.

Theorem 4.2 The clausal completion procedure is complete for ground clauses not containing inequations.

The proof of this theorem is the topic of Section 6.

Corollary 4.3 Critical inferences are refutationally complete.

The Simplify rule given above is not practical since we did not present a method to construct $\mathcal{T}'$. The following are three simple possible values for $\mathcal{T}'$:

1. $\mathcal{T}$: In this case, the Simplify rule deletes the clause Γ.
2. $\mathcal{T} \cup \{\Gamma[b/a]\}$: if $\mathcal{R} \cup \mathcal{T}'_{\succ\Gamma}$ can rewrite some subterm a in Γ to b.
3. $\mathcal{T} \cup \{\Gamma - \{A\}\}$: if $\mathcal{R} \cup \mathcal{T}'_{\succ\Gamma}$ can disprove the literal A in Γ.

Example 2 [Nieuwenhuis, 1990] Suppose we are given the set of clauses

1. $x \leq x$
2. $\max(x, y) = x \ \vee \ x \leq y$
3. $\max(x, y) = y \ \vee \ \neg(x \leq y)$
4. $x \leq \max(y, x)$
5. $x \leq \max(x, y)$

and the order $\succ_{mpo}$ with the precedence $\max \ >_p \ \geq$.

Applying the completion procedure produces the clause

6. $x \leq y \ \vee \ y \leq x$

as a result of a critical paramodulation inference between clauses 2 and 4.

Suppose we accept clauses 1, 2, 3, and 6. Then, clauses 4 and 5 can be deleted since they can be proved by reductive refutation using the accepted clauses. For example, a reductive proof of clause 5 is (X and Y are the skolem constants replacing x and y).

$$\dfrac{\dfrac{\dfrac{\neg(X \leq \max(X,Y)) \quad (2)}{\neg(X \leq X), X \leq Y}\,\text{Para} \qquad \dfrac{\dfrac{\neg(X \leq \max(X,Y)) \quad (3)}{\neg(X \leq Y), \neg(X \leq Y)}\,\text{Para}}{\neg(X \leq Y)}\,\text{Fac}}{\neg(X \leq X)}\,\text{Cut} \qquad (1)}{\bot}\,\text{Res}$$

In the above proof, we assumed $Y >_p X$. A reductive proof with $X >_p Y$ is obtained by interchanging the last two steps of this proof.

The set $\mathcal{T}$ is now empty and any critical consequences of the accepted clauses can be deleted. Thus, the clauses 1, 2, 3 and 6 forms a complete set.

[1]To obtain completeness for clauses containing inequations, one would need an inference rule of the form $A(a), \neg A(b) \vdash a \neq b$. Such a rule would be too inefficient for any practical purpose.

5 Examples

Example 5.1 (Bumcroft's identity in Lattice theory) Bumcroft [1965] proposed the following problem in Lattice Theory:

> Let a and b be two elements of a modular lattice with unique complements, and $\bar{a}$ and $\bar{b}$ their unique complements. Assuming the join and the meet of a and b exist, does the join of $\bar{a}$ and $\bar{b}$ exist and is it the unique complement of the meet of a and b?

Guard et al. [1969] showed this identity to be true with the help of their semi-automated proof checker SAM. Essential to their proof was a lemma , thereafter known as SAM's lemma, stating a non-trivial identity for modular lattices. Using this lemma, Guard and his group were able to give a simple proof of Bumcroft's open question. Other recent approaches to this problem include [Zhang, 1988, McCune, 1988, Vigneron, 1994].

We show here how the clausal completion method could prove this identity with minimal user interaction. With the completion method we do not require an advance knowledge of SAM's lemma. This lemma, along with Bumcroft's identity, becomes reductively provable once sufficient critical consequences of the axioms of modular lattices have been generated.

The axiomatization of modular lattices with unique complements is given in Figure 1. It is taken from McCune [1988]. We use the symbols $\wedge$ and $\vee$ in infix notation to denote the join and meet relations in a lattice, and the predicates Comp and $\mathsf{Unicomp}$ to denote the complement and unique complement relations.

The theorem to be proved is

$$\mathsf{Unicomp}(a,\bar{a})\ \&\ \mathsf{Unicomp}(b,\bar{b})\ \&\ \mathsf{Comp}(x,a\vee b)\ \&\ \mathsf{Comp}(y,a\wedge b)\quad\Longrightarrow\quad \mathsf{Unicomp}(a\wedge b,\bar{a}\vee\bar{b})$$

We use, as order relation, the AC recursive path ordering of Dershowitz and Mitra [1993] with the precedence

$$\mathsf{Unicomp}\ >_p\ \mathsf{Comp}\ >_p\ \neq\ >_p\ \wedge\ >_p\ \vee\ >_p\ 0\ >_p\ 1\ >_p\ \bar{a}\ >_p\ \bar{b}\ >_p\ a\ >_p\ b\ >_p\ t$$

Initially, no reductive proof of the theorem is possible. Thus, we apply the completion procedure. Although the procedure fails to find a complete theory for modular lattices, it generates the following critical consequences:

21. $z = y \quad \Longleftarrow x \vee y = 1\ \&\ x \wedge y = 0\ \&\ \mathsf{Unicomp}(x, z)$ from a critical resolution between the axioms 16 and 18;
22. $x \wedge (y \vee z) = x \wedge (y \vee ((x \vee v) \wedge z)) \quad \Longleftarrow (x \vee v) \wedge y = y$ from a critical AC-superposition inference between axioms 13 and 3;
23. $x = y \vee (x \wedge z) \quad \Longleftarrow x \wedge y = y\ \&\ \mathsf{Comp}(y, z)$ from a critical AC-superposition between axioms 13 and 15 followed by a simplification using axiom 11;
24. $x \wedge (y \vee z) = x \wedge y \quad \Longleftarrow (x \vee v) \wedge y = y\ \&\ \mathsf{Comp}(x \vee v, z)$ from a critical AC-superposition between axioms 22 and 14 followed by a simplification using axiom 10;
25. $x \vee y = y \vee (x \wedge z) \quad \Longleftarrow x \wedge (y \wedge v) = y \wedge v\ \&\ \mathsf{Comp}(y \wedge v, z)$ from a critical AC-superposition between axioms 23 and 4;
26. $x \wedge ((y \wedge v) \vee z) = x \wedge y \wedge v \quad \Longleftarrow \mathsf{Comp}(x \vee v, z)$ from a critical AC-superposition between axioms 24 and 3; ;
27. $x \vee y = y \vee (x \wedge z) \quad \Longleftarrow \mathsf{Comp}(y \wedge x, z)$ from a critical AC-superposition between axioms 25 and 1;

The following 8 equations characterize lattices:

1. $x \wedge x = x$ (idempotence)
2. $x \vee x = x$
3. $x \wedge (x \vee y) = x$ (absorption)
4. $x \vee (x \wedge y) = x$
5. $x \wedge y = y \wedge x$ (commutativity)
6. $x \vee y = y \vee x$
7. $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ (associativity)
8. $x \vee (y \vee z) = (x \vee y) \vee z$

0 and 1 denote respectively the top and bottom elements of the lattice:

9. $x \wedge 0 = 0$
10. $x \vee 0 = x$
11. $x \wedge 1 = x$
12. $x \vee 1 = 1$

The lattice is modular:

13. $x \wedge (y \vee z) = y \vee (x \wedge z) \quad \Longleftarrow x \wedge y = y$

Complements and unique complements are defined as follows:

14. $x \wedge y = 0 \quad \Longleftarrow \mathsf{Comp}(x, y)$
15. $x \vee y = 1 \quad \Longleftarrow \mathsf{Comp}(x, y)$
16. $\mathsf{Comp}(x, y) \quad \Longleftarrow x \vee y = 1 \;\&\; x \wedge y = 0$
17. $\mathsf{Comp}(x, y) \quad \Longleftarrow \mathsf{Unicomp}(x, y)$
18. $z = y \quad \Longleftarrow \mathsf{Comp}(x, y) \;\&\; \mathsf{Unicomp}(x, z)$
19. $\mathsf{Unicomp}(x, y) \quad \Longleftarrow \mathsf{Comp}(x, y) \;\&\; \neg(\mathsf{Comp}(x, f(x, y)))$
20. $\mathsf{Unicomp}(x, y) \quad \Longleftarrow \mathsf{Comp}(x, y) \;\&\; f(x, y) = y$

Figure 1: Axiomatization of modular lattices with unique complements

1. $x \leq z \Longleftarrow y \leq z \ \wedge \ x \leq y$
2. $\mathsf{min}(x, y) \leq x$
3. $x + y \leq z \Longleftarrow x \leq \mathsf{half}(z) \ \wedge \ y \leq \mathsf{half}(z)$
4. $|x + y| \leq z \Longleftarrow |x| + |y| \leq z$
5. $0 \leq \mathsf{half}(z) \Longleftarrow 0 \leq z$
6. $0 \leq \mathsf{min}(x, y) \Longleftarrow 0 \leq x \ \wedge \ 0 \leq y$
7. $(x + w) - (y + z) = (x - y) + (w - z)$
8. $+$ and $\mathsf{min}(,)$ are associative and commutative.

Figure 2: Calculus Axioms

The hypotheses:
9. $0 \leq \delta_1(dy) \Longleftarrow 0 \leq dy$
10. $|f(x) - L_1| \leq dy \Longleftarrow |x - A| \leq \delta_1(dy) \ \wedge \ 0 \leq dy$
11. $0 \leq \delta_2(dy) \Longleftarrow 0 \leq dy$
12. $|g(x) - L_2| \leq dy \Longleftarrow |x - A| \leq \delta_2(dy) \ \wedge \ 0 \leq dy$

The negation of the conclusion:
13. $0 \leq B$
14. $|\mathsf{sol}(dy) - A| \leq dy \Longleftarrow 0 \leq dy$
15. $\bot \Longleftarrow |(f(\mathsf{sol}(dy)) + g(\mathsf{sol}(dy))) - (L_1 + L_2)| \leq B \ \wedge \ 0 \leq dy$

Figure 3: Goal clauses for the theorem

Augmenting the theory in Figure 1 with the above propositions suffices to obtain reductive proofs for both Bumcroft's identity (see Appendix A) and SAM's lemma. Thus, we have obtained a reductive proof of Bumcroft's identify after seven Deduce steps using AC-unification. We note that the critical consequences generated by these Deduce steps can be divided into three *generations* of critical consequences. The propositions 21, 22, and 23 belong to the first generation; the propositions 24 and 25 belong to the second; and the propositions 26 and 27 belong to the third. The final proof requires no auxiliary lemmas; this is an improvement over previous approaches to this problem. On the other hand, seven Deduce steps, particularly if they use AC-unification, is a significant requirement.

Example 5.2 (The sum of two continuous functions is continuous) McCharen, Overbeek, and Wos [1976] proposed a series of problems for automated theorem provers based on the classical theorem from analysis that the sum of two continuous functions is continuous. The relevant part of the theory used by these problems is given in Figure 2.

We want to prove the theorem

> Let f and g be two continuous functions. The sum $f + g$ is a continuous function.

The clausal representation of the theorem is given in Figure 3.

We will prove this theorem with the completion procedure. We base the order relation on the generalized recursive path order [Dershowitz and Hoot, 1993] with the precedence

$$- \ >_p \ + \ >_p \ | \ | \ >_p \ \mathsf{half} \ >_p \ \leq \ >_p \ \mathsf{min} \ >_p \ f \ >_p \ g \ >_p \ \delta_1 \ >_p \ \delta_2 \ >_p \ B \ >_p \ \mathsf{sol}$$

Further, comparisons with $\leq$ are lexicographic and, to account for unification, in comparisons involving the skolem function sol we discard its argument.

Two critical consequences of the initial set of axioms and one critical consequences of the goal clauses are needed before a reductive refutation is possible. These are

16. $|x + y| \leq z \Longleftarrow |x| \leq \mathsf{half}(z) \ \wedge \ |y| \leq \mathsf{half}(z)$ from a critical resolution between 3 and 4;
17. $x \leq y \Longleftarrow x \leq \mathsf{min}(y, z)$ from a critical resolution between 2 and 1;
18. $|\mathsf{sol}(\mathsf{min}(y, z)) - A| \leq y \Longleftarrow 0 \leq \mathsf{min}(y, z)$ from a critical resolution between 17 and 14.

Now, we can easily build a reductive refutation starting with the goal clause 15. This reductive refutation is a linear sequence of inference steps starting with a paramodulation step between clauses 15 and 7 and continuing with resolution or paramodulation steps involving, in order, the clauses 16, 10, 12, 18, 18, 6, 9, 11, 5, and 13. Thus, the theorem is true.

6 Completeness

6.1 Completeness of equational completion

To prove the completeness of the clausal completion procedure, we take inspiration from Bachmair, Dershowitz, and Hsiang's completeness proof of the equational completion procedure [1986]. This proof is based on Huet's proof [1980] that the *local confluence property* is a sufficient condition for the Church-Rosser property.

The *local confluence* property is the property that for any *local peak proof* of the form $t \leftarrow z \rightarrow r$, there is a rewrite proof of the form $t \rightarrow^* u \leftarrow^* r$. This property implies the Church-Rosser property as can be seen from the following proof adapted from Huet [1980]:

Assume the order $>$ is well-founded and consider a proof of the form

$$y_0 \leftarrow^+ x_1 \rightarrow^+ y_1 \leftarrow^+ x_2 \rightarrow^+ \ldots \leftarrow^+ x_n \rightarrow^+ y_n \tag{8}$$

where the x_i's denote the *peaks* in the proof. We measure the *complexity* of a proof by the multiset of its peaks. A proof of minimal complexity has no peak, so it is a rewrite proof. The goal is to show that using the local confluence property, we can transform the above proof and obtain a *smaller* proof of $y_0 \leftrightarrow^* y_n$. By induction, we will then be able to conclude that a rewrite proof of $y_0 \leftrightarrow^* y_n$ exists.

Consider the local peak rooted at x_1. It has the form $y' \leftarrow x_1 \rightarrow z'$. By the local confluence property, there must be an equivalent proof of the form $y' \rightarrow^* u \leftarrow^* z'$. Thus, proof (8) above can be replaced by

$$y_0 \leftarrow^* y' \rightarrow^* u \leftarrow^* z' \rightarrow^* y_1 \leftarrow^+ x_2 \rightarrow^+ \ldots \leftarrow^+ x_n \rightarrow^+ y_n$$

The multiset of peaks is now $\{y', z', x_2, \ldots, x_n\}$. This is smaller than the original multiset of peaks, since $x_1 \rightarrow y'$ and $x_1 \rightarrow z'$. We have a smaller proof of $y_0 \leftrightarrow^* y_n$, so, by induction, $\exists u, y_0 \rightarrow^* u \leftarrow^* y_n$.
□

Bachmair, Hsiang, and Dershowitz's proof of the completeness of the Knuth-Bendix completion procedure [1986] can be seen as an extension of Huet's proof. Their proof proceeds by showing that the completion procedure can reduce the complexity of a proof by either eliminating local peaks through Deduce steps or simplifying axioms through Simplify steps.

$$\dfrac{\neg A \qquad \dfrac{A \vee D \qquad \dfrac{\neg D \vee C \qquad \neg C \vee B}{D \vee B}\,Cut}{A \vee B}\,Cut}{B}\,Cut \qquad\qquad \dfrac{\dfrac{\neg A \qquad A \vee D}{D}\,Cut \qquad \dfrac{\neg D \vee C \qquad \neg C \vee B}{B \vee \neg D}\,Cut}{B}\,Cut$$

Figure 4: Two proof trees denoting the same proof

6.2 Completeness of Clausal Completion

Huet's proof uses the simple proof-theoretic mechanism of proof transformations to show that non-reductive proofs can be transformed into reductive proofs. However, the attempts of Bachmair [1989], Nieuwenhuis and Orejas [1991], and Bronsard and Reddy [1992] to use a similar proof theoretic approach to prove the completeness of clausal deduction mechanisms led to excessively complex and error-prone proofs.

We claim that when the proof transformation approach is successful, the success rests essentially on the proper representation of proofs. The crux of the proof transformation approach is to show that all proofs can be transformed into proofs of a certain kind. Clearly, how proofs are represented is the crucial factor that determines the ease, or the complexity, of defining the appropriate transformation relation. We believe, therefore, that the chief impediment to applying the proof transformation approach to clausal deduction is the classical representation of proofs as proof trees. For example, the complexity of the three above-mentioned attempts to use proof transformation techniques resulted in part from the need for many low-level tree manipulation operations.

The classical representation of proofs as proof trees encompasses *inessential informations*, namely the order of execution of the inferences. We take inspiration from Girard's *proof nets* for linear logic [1987] and from the *resolution graphs* of Yates, Raphael, and Hart [1970] to develop a graph-based representation of proofs that abstracts such control information to reveal the *essence* of the proof, that is, the axioms needed by the proof and the inferences using these axioms. We call these graphs *proof nets*. They provide an almost *canonical* representation of proofs over which proof transformation relations can be defined easily.

To illustrate the difference between proof trees and proof nets consider the following example: Let $\mathcal{A}$ be the set of axioms $\{\neg A,\, B \vee \neg C,\, C \vee \neg D,\, D \vee A\}$. The proposition $\{B\}$ is a consequence of $\mathcal{A}$. Figure 4 presents two derivations of B from $\mathcal{A}$. These two proof trees are significantly different, although they both denote essentially the same proof: the same axioms are used, the same conclusion is reached, both have exactly three cut inferences, and the same literals are cut in these inferences. However, the order in which the cut inferences are executed is different and this leads to two different proof trees. In contrast, such control information is absent in the proof net representation, so there is only one proof net showing that B is a consequence of $\mathcal{A}$; it is

$$\boxed{\neg A} \sim \boxed{A \quad D} \sim \boxed{\neg D \quad C} \sim \boxed{\neg C \quad B} \tag{9}$$

where the *boxes* represent the axioms of $\mathcal{A}$, and $\sim$, the *connection* relation, represents cut inferences. Contraction inferences are represented by allowing multiple connections to the same literal.

The use of proof nets suggests a simple adaptation of Huet's proof. Suppose given a well-founded order over literals. We define as *local peak proofs*, proofs of the form

$$\ulcorner \Gamma \quad A \lrcorner \sim \ulcorner \neg A \quad \Delta \lrcorner$$

where A is greater than all literals in Γ or Δ. A proof free of local peaks is a reductive proof. We define the *local confluence* property as the property that for any local peak proof as above, there exists a proof of $\Gamma\,\Delta$ where all literals are smaller than A.

Consider an arbitrary proof net with conclusion Γ. We define the complexity of the proof as the set of the literals occuring in local peak subproofs. First, we transform the proof net to *isolate* the local peaks, i.e., to ensure that they are not part of some contraction inferences. Then, using the local confluence property, we replace all the local peaks of maximal complexity by new subproofs with smaller literals. Thus, the complexity of the proof reduces. By induction, we can conclude that Γ must have a proof free of local peaks, i.e., a reductive proof. □

The above informal proof is the counterpart for clausal proofs of Huet's proof. Like Huet's proof, it can be extended to prove the completeness of the completion method. (see [Bronsard, 1995] for details)

7 Conclusion

In conclusion, the key parts of our method are the concepts of critical consequences and reductive proofs. These concepts were defined in order to maximize the efficiency of the completion method without sacrificing completeness or expressiveness. Critical consequences are defined using the notion of critical inferences. These inferences are less constrained than the ordered inferences previously used when investigating adaptation of the Knuth-Bendix method to first-order deduction. However, we think that this relaxation is necessary to obtain completeness. Conversely, our notion of reductive proofs provides a powerful, yet practical, simplification method stronger than previous methods. Finally, by using a graph-based representation of proofs we have obtained a simpler and intuitive completeness proof.

References

[Bachmair, 1989] L. Bachmair. Proof normalization for resolution and paramodulation. In N. Dershowitz, editor, *Rewriting Techniques and Applications*, pages 15–28, Springer-Verlag, 1989.

[Bachmair and Ganzinger, 1994] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *J. Logic Computation*, 4(3):217–247, 1994.

[Bachmair *et al.*, 1986] L. Bachmair, N. Dershowitz, and J. Hsiang. Orderings for equational proofs. In *LICS*, pages 346–357, 1986.

[Bachmair *et al.*, 1989] L. Bachmair, N. Dershowitz, and D. A. Plaisted. Completion without failure. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures*, chapter 1, pages 1–30, Academic Press, 1989.

[Bronsard, 1995] F. Bronsard. *Using Term Ordering to Control Clausal Deductions.* PhD thesis, University of Illinois, Urbana, 1995. (Tech. Report UIUCDCS-R-95-1910).

[Bronsard and Reddy, 1992] F. Bronsard and U. S. Reddy. Reduction techniques for first-order reasoning. In M. Rusinowitch and J. L. Rémy, editors, *3rd CTRS Workshop*, pages 242–256, Springer-Verlag, 1992.

[Bumcroft, 1965] R. Bumcroft. In *Proc. Glasgow Math. Assoc. 7*, pages 22–23, 1965.

[Dershowitz and Hoot, 1993] N. Dershowitz and C. Hoot. Natural termination. In C. Kirchner, editor, *5th RTA Conf.*, pages 405–420, Springer-Verlag, Montreal (Canada), 1993.

[Dershowitz and Jouannaud, 1990] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science B: Formal Methods and Semantics*, chapter 6, pages 243–320, North-Holland, Amsterdam, 1990.

[Dershowitz and Mitra, 1993] N. Dershowitz and S. Mitra. Path orderings for termination of associative-commutative rewriting. In M. Rusinowitch and J. L. Rémy, editors, *3rd CTRS Workshop*, pages 168–174, Springer-Verlag, 1993.

[Girard, 1987] J.-Y. Girard. Linear logic. *Theoretical Comp. Science*, 50:1–102, 1987.

[Guard *et al.*, 1969] J. R. Guard, F. C. Oglesby, J. H. Bennett, and Settle. Semi-automated mathematics. *J. ACM*, 16:49–62, 1969.

[Hsiang and Rusinowitch, 1986] J. Hsiang and M. Rusinowitch. A new method for establishing refutational completeness in theorem proving. In J. H. Siekmann, editor, *8th CADE*, pages 141–152, Springer-Verlag, 1986.

[Huet, 1980] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. ACM*, 27(4):797–821, October 1980.

[Knuth and Bendix, 1970] D. Knuth and P. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297, Pergamon Press, Oxford, 1970.

[McCharen *et al.*, 1976] J.D. McCharen, R.A. Overbeek, and L.A. Wos. Experiments for and with automated theorem proving programs. *IEEE Transactions on Computers*, C-25(8):773–782, 1976.

[McCune, 1988] W. McCune. Challenge equality problems in lattice theory. In E. Lusk and R. Overbeek, editors, *9th CADE Conf.*, pages 704–709, Springer-Verlag, 1988.

[Nieuwenhuis, 1990] R. Nieuwenhuis. *Theorem proving in first order logic with equality by clausal rewriting and completion*. PhD thesis, UPC Barcelone, 1990.

[Nieuwenhuis and Orejas, 1991] R. Nieuwenhuis and F. Orejas. Clausal rewriting. In S. Kaplan and M. Okada, editors, *2nd CTRS Workshop*, pages 246–258, Springer-Verlag, 1991.

[Robinson and Wos, 1969] G. Robinson and L. Wos. Paramodulation and theorem-proving in first order theories with equality. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 135–150, Edinburgh University Press, Edinburgh, Scotland, 1969.

[Robinson, 1965] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12:23–41, 1965.

[Vigneron, 1994] L. Vigneron. Associative-commutative deduction with constraints. In A. Bundy, editor, *12th CADE Conf.*, pages 530–544, Springer-Verlag, 1994.

[Yates *et al.*, 1970] R. A. Yates, B. Raphael, and T. Hart. Resolution graphs. *Artificial Intelligence*, 1:257–289, 1970.

[Zhang, 1988] H. Zhang. *Reduction, Superposition and Induction: Automated Reasoning in an Equational Logic*. PhD thesis, Rensselaer Polytechnic Institute, 1988.

[Zhang and Kapur, 1988] H. Zhang and D. Kapur. First-order theorem proving using conditional rewrite rules. In E. Lusk and R. Overbeek, editors, *9th CADE Conf.*, pages 1–20, Springer-Verlag, 1988.

A Proof of Bumcroft's identity

We start with the following simple subproofs

$$\alpha_1 \equiv \frac{(\mathsf{Comp}(y, a \wedge b)) \quad (14)}{y \wedge (a \wedge b) = 0}\,\text{Res} \qquad \alpha_2 \equiv \frac{(\mathsf{Unicomp}(a, \bar{a})) \quad (21)}{\bar{a} = u,\ a \vee u \neq 1,\ a \wedge u \neq 0}\,\text{Res}$$

$$\alpha_3 \equiv \frac{(\mathsf{Comp}(y, a \wedge b)) \quad (27)}{a \vee (b \wedge y) = a \vee b}\,\text{Res} \qquad \alpha_4 \equiv \frac{(\mathsf{Comp}(x, a \vee b)) \quad (15)}{x \vee (a \vee b) = 1}\,\text{Res}$$

$$\alpha_5 \equiv \frac{(\mathsf{Unicomp}(b, \bar{b})) \quad (21)}{\bar{b} = u,\ b \vee u \neq 1,\ b \wedge u \neq 0}\,\text{Res} \qquad \alpha_6 \equiv \frac{(\mathsf{Comp}(x, a \vee b)) \quad (26)}{b \wedge ((u \wedge a) \vee x) = b \wedge (u \wedge a)}\,\text{Res}$$

$$\alpha_7 \equiv \frac{(\mathsf{Comp}(y, a \wedge b)) \quad (14)}{a \wedge (y \wedge b) \to 0}\,\text{Res}$$

Let β_1 be the subproof

$$\cfrac{(\alpha_4) \quad \cfrac{(\alpha_3) \quad \cfrac{(\alpha_2) \quad \cfrac{(\alpha_1) \quad \cfrac{(A) \quad (26)}{a \wedge ((u \wedge b) \vee x) = a \wedge (u \wedge b)}\,\text{Res}}{a \wedge ((y \wedge b) \vee x) = 0}\,\text{AC-Para}}{\bar{a} = (y \wedge b) \vee x,\ a \vee ((y \wedge b) \vee x) \neq 1}\,\text{Res}}{\bar{a} = (y \wedge b) \vee x,\ a \vee (b \vee x) \neq 1}\,\text{AC-Para}}{\bar{a} = (y \wedge b) \vee x}\,\text{Res}$$

where A is the goal clause $\mathsf{Comp}(x, a \vee b)$.

The next subproof, β_2, is similar to β_1 except that a and b are interchanged and α_2 is replaced by α_5. Thus, β_2 has the conclusion $\bar{b} = (y \wedge a) \vee x$.

Let β_3 be the subproof

$$\cfrac{(A) \quad \cfrac{(10) \quad \cfrac{(\alpha_7) \quad \cfrac{(\alpha_6) \quad \cfrac{(13) \quad \cfrac{(\beta_2) \quad \cfrac{(\beta_1) \quad \cfrac{(19) \quad \neg\mathsf{Unicomp}(a \vee b, \bar{a} \wedge \bar{b})}{\mathsf{Comp}(a \vee b, \bar{a} \wedge \bar{b}),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{Res}}{\mathsf{Comp}(a \vee b, ((y \wedge b) \vee x) \wedge \bar{b}),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{Para}}{\mathsf{Comp}(a \vee b, ((y \wedge b) \vee x) \wedge ((y \wedge a) \vee x)),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{Para}}{\mathsf{Comp}(a \vee b, x \vee ((y \wedge b) \wedge ((y \wedge a) \vee x))),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{AC-Para}}{\mathsf{Comp}(a \vee b, x \vee (y \wedge (b \wedge a))),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{AC-Para}}{\mathsf{Comp}(a \vee b, x \vee 0),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{AC-Para}}{\mathsf{Comp}(a \vee b, x),\ \neg\mathsf{Comp}(t, a \vee b)}\,\text{Para}}{\neg\mathsf{Comp}(t, a \vee b)}\,\text{Res}$$

where t denotes $f(a \vee b, a \wedge b)$.

Replacing the goal clause A by the subproof β_3 in the subproofs β_1 and β_2 produces two subproofs, which we denote by β_4 and β_5, with conclusions

$$\bar{a} = (y \wedge a) \vee t$$
$$\bar{b} = (y \wedge a) \vee t$$

The final proof is similar to the subproof β_3. The first inference is

$$\frac{(20) \quad \neg\mathsf{Unicomp}(a \vee b, \bar{a} \wedge \bar{b})}{\mathsf{Comp}(a \vee b, \bar{a} \wedge \bar{b}),\ t \neq \bar{a} \wedge \bar{b}}\ \text{Res}$$

The literal $\mathsf{Comp}(a \vee b, \bar{a} \wedge \bar{b})$ can be eliminated as was done in the subproof β_3, leaving us with the literal $t \neq \bar{a} \wedge \bar{b}$. The term $\bar{a} \wedge \bar{b}$ is rewritten to $t \vee 0$ by a subproof similar to β_3 with β_1 and β_2 replaced by the subproofs β_4 and β_5. The resulting literal $t \neq t \vee 0$ can be resolved with the axiom 10. We obtain the empty clause. Thus, Bumcroft's identity holds.