



# Read-Once Resolutions in Horn Formulas

Hans Kleine Büning<sup>1</sup>, P. Wojciechowski<sup>2</sup>, and K. Subramani<sup>2</sup>(✉)

<sup>1</sup> Computer Science Institute,  
University of Paderborn, Paderborn, Germany  
kbcs1@uni-paderborn.de

<sup>2</sup> LDCSEE, West Virginia University,  
Morgantown, WV, USA  
pwojciec@mix.wvu.edu, k.subramani@mail.wvu.edu

**Abstract.** In this paper, we discuss the computational complexity of Read-once resolution (ROR) with respect to Horn formulas. Recall that a Horn formula is a boolean formula in conjunctive normal form (CNF), such that each clause has at most one positive literal. Horn formulas find applications in a number of domains such as program verification and logic programming. It is well-known that deduction in ProLog is based on *unification*, which in turn is based on resolution and *instantiation*. Resolution is a sound and complete procedure to check whether a boolean formula in CNF is satisfiable. Although inefficient in general, resolution has been used widely in theorem provers, on account of its simplicity and ease of implementation. This paper focuses on two variants of resolution, viz., Read-once resolution and Unit Read-once resolution (UROR). Both these variants are **sound**, but **incomplete**. In this paper, the goal is to check for the existence of proofs (refutations) of Horn formulas under these variants. We also discuss the computational complexity of determining optimal length proofs where appropriate.

## 1 Introduction

This paper is concerned with analyzing Horn formulas from the perspectives of Read-once resolution (ROR) and Unit Read-once resolution (UROR). A Horn clause is a clause with at most one positive literal. A Horn formula is a conjunction of Horn clauses. These formulas find applications in a number of domains, including but not limited to program verification [8, 11], logic programming [4], abstract interpretation [3] and econometrics [1]. Resolution is a sound and complete proof system introduced by Robinson [10] for checking the satisfiability of boolean formulas in conjunctive normal form (CNF). As an algorithmic strategy, resolution is not particularly efficient. However, it continues to be used

---

The first author was supported in part by the National Science Foundation through Award CCF-1305054.

This research was made possible by NASA WV EPSCoR Grant # NNX15AK74A.

The second author was supported in part by the Air Force Research Laboratory under US Air Force contract 88ABW-2017-1077.

© Springer Nature Switzerland AG 2019

Y. Chen et al. (Eds.): FAW 2019, LNCS 11458, pp. 100–110, 2019.

[https://doi.org/10.1007/978-3-030-18126-0\\_9](https://doi.org/10.1007/978-3-030-18126-0_9)

extensively in theorem provers on account of its simplicity and ease of implementation. A resolution proof system typically declares a CNF system to be feasible, or (alternatively) finds a sequence of resolution steps which lead to the derivation of a contradiction. Such a sequence of resolution steps is called a *refutation* since it acts as a certificate, which certifies the infeasibility of the given clausal system. There are a number of variants of resolution. Each variant puts limitations on the type of resolution step that is permitted or how the resolution steps are to be counted. Three variants which are particularly important from the program verification perspective are tree-like resolution, dag-like resolution and read-once resolution. In tree-like resolution refutations input clauses can be used in multiple resolution steps, however each derived clause can only be used once. If a resolution step requires the reuse of a derived clause then that clause needs to be re-derived. Note that this re-derivation increases the length of the refutation. In dag-like resolution refutations both input and derived clauses can be used in multiple resolution steps. Finally, in read-once resolution refutations each input and derived clause can be used by only one resolution step. Note that clauses can be re-derived, and thus reused, if they can be derived from a *different* set of input clauses. A complete exposition of different types of resolutions can be found in [9] and [5].

Although the resolution proof system is both sound and complete (in general), several of its variants are not. For instance, read-once resolution (ROR) is **incomplete**, in that there exist unsatisfiable CNF formulas, which do not have read-once resolutions.

*Example 1.* Consider the following 2CNF formula:

$$\begin{array}{l} (x_1 \vee x_2) \quad (x_3 \vee x_4) \quad (\neg x_1 \vee \neg x_3) \\ (\neg x_1 \vee \neg x_4) \quad (\neg x_2 \vee \neg x_3) \quad (\neg x_2 \vee \neg x_4) \end{array}$$

We now show that this formula does not have a read-once refutation.

To derive  $(x_1)$  we need to derive  $(\neg x_2)$ . Similarly, to derive  $(x_2)$  we need to derive  $(\neg x_1)$ . However, the derivations of both  $(\neg x_1)$  and  $(\neg x_2)$  require the use of the clause  $(x_3 \vee x_4)$ .

To derive  $(x_3)$  we need to derive  $(\neg x_4)$ . Similarly, to derive  $(x_4)$  we need to derive  $(\neg x_3)$ . However, the derivations of both  $(\neg x_3)$  and  $(\neg x_4)$  require the use of the clause  $(x_1 \vee x_2)$ .

Another variant of resolution is *unit resolution*. In unit resolution, for each resolution step one of the two parent clauses must be a unit clause, i.e. a clause of the form  $(x_i)$  or  $(\neg x_i)$ . Although unit resolution is incomplete in general, it has been shown to be complete for Horn formulas [4]. In this paper, we study the effect of combining read-once resolution with unit resolution with respect to Horn formulas. We refer to this resolution system as Unit read-once resolution (UROR).

A natural question to ask is: How many times should a given clause be copied so that a read-once refutation can be extracted? This question was first investigated by Iwama and Miyano in [6] and leads naturally to the notion of

*copy complexity* (see Sect. 2). This paper investigates the copy complexity of Horn formulas with respect to unit resolution.

The principal contributions of this paper are as follows:

1. A proof that the problem of finding the shortest read-once refutation for a Horn formula is **NP-hard** (Sect. 3).
2. An algorithm that can determine if a Horn formula with  $m$  clauses and at most two literals per clause has a read-once unit resolution refutation in  $O(m^2)$  time (Sect. 4).
3. Showing that the copy complexity of Horn formulas with respect to read-once unit resolution is  $2^{n-1}$  where  $n$  is the number of variables (Sect. 5).

The rest of the paper is organized as follows. Section 2 details the problems under consideration. In Sect. 3 we cover the optimal length ROR problem for Horn formulas. Section 4 describes our work on the UROR problem for 2-Horn formulas. In Sect. 5, we examine the copy complexity of Horn formulas with respect to read-once unit resolution. Finally, Sect. 6 summarizes our results.

## 2 Statement of Problems

In this section, we briefly discuss the terms used in this paper. We assume that the reader is familiar with elementary propositional logic.

**Definition 1.** A **literal** is a variable  $x$  or its complement  $\neg x$ .  $x$  is called a *positive literal* and  $\neg x$  is called a *negative literal*.

**Definition 2.** A **CNF clause** is a disjunction of literals. The empty clause, which is always false, is denoted as  $\sqcup$ .

**Definition 3.** A  **$k$ -CNF clause** is a CNF clause with at most  $k$  literals.

**Definition 4.** A **Horn clause** is a CNF clause which contains at most one positive literal.

For a single resolution step with parent clauses  $(\alpha \vee x)$  and  $(\neg x \vee \beta)$  and with resolvent  $(\alpha \vee \beta)$ , we write

$$(\alpha \vee x), (\neg x \vee \beta) \stackrel{1}{\text{RES}} (\alpha \vee \beta).$$

The variable  $x$  is called the matching or resolution variable. If for initial clauses  $\alpha_1, \dots, \alpha_n$ , a clause  $\pi$  can be generated by a sequence of resolution steps we write

$$\alpha_1, \dots, \alpha_n \stackrel{\cdot}{\text{RES}} \pi.$$

If a resolution step involves a unit clause, a clause of the form  $(x)$  or  $(\neg x)$ , then it is called a unit resolution step. If a resolution refutation consists of only unit resolution steps, then it is called a unit resolution refutation.

We now formally define the types of resolution refutation discussed in this paper.

**Definition 5.** A **Dag-like** resolution refutation is a refutation in which each clause,  $\pi$ , can be used in any number of resolution steps. This applies to clauses present in the original formula and those derived as a result of previous resolution steps.

Note that Dag-like refutations are unrestricted and thus are equivalent to general resolution. We now introduce several restricted forms of resolution.

**Definition 6.** A **Tree-like** resolution refutation is a refutation in which each derived clause,  $\pi$ , can be used in only one resolution step. However, clauses in the original formula can be reused and clauses can be re-derived if necessary.

**Definition 7.** A **Read-Once** resolution refutation is a refutation in which each clause,  $\pi$ , can be used in only one resolution step. This applies to clauses present in the original formula and those derived as a result of previous resolution steps.

In a read-once refutation, a clause can be reused if can be re-derived from a set of unused input clauses.

More formally, a resolution derivation  $\Phi \mid_{RES} \pi$  is a read-once resolution derivation, if for all resolution steps  $\pi_1 \wedge \pi_2 \mid_{RES} \pi$ , we delete one instance of the clauses  $\pi_1$  and  $\pi_2$  from, and add a copy of the resolvent  $\pi$  to, the current multi-set of clauses. In other words, if  $U$  is the current multi-set of clauses, we obtain  $U = (U \setminus \{\pi_1, \pi_2\}) \cup \{\pi\}$ .

It is important to note Read-Once resolution is an **incomplete** refutation procedure.

We can similarly define read-once unit resolution.

**Definition 8.** A **Read-Once Unit** resolution refutation is a unit resolution refutation in which each clause,  $\pi$ , can be used in only one unit resolution step. This applies to clauses present in the original formula and those derived as a result of previous unit resolution steps.

This lets us define the concept of copy complexity with respect to read-once unit resolution.

**Definition 9.** A CNF formula  $\Phi$  has **copy complexity** at most  $k$ , with respect to unit resolution, if there exists a multi-set of CNF clauses,  $\Phi'$  such that:

1. Every clause in  $\Phi$  appears at most  $k$  times in  $\Phi'$ .
2. Every clause in  $\Phi'$  appears in  $\Phi$ .
3.  $\Phi'$  has a read-once unit resolution refutation.

Let  $k$ -UROR denote the set of CNF formulas with copy complexity  $k$  with respect to unit resolution.

For any type of resolution refutation, we can define the length of that refutation in terms of the number of resolution steps.

**Definition 10.** The length of a resolution refutation  $R$ , is the number of resolution steps in  $R$ .

We now define the problems under consideration.

**The Read-once Refutation (ROR) problem:** Given a CNF formula  $\Phi$ , determine whether or not  $\Phi$  has a read-once refutation.

**The Optimal Length Read-once Refutation (OLROR) problem:** Given a CNF formula  $\Phi$ , produce the read-once refutation of  $\Phi$  that has minimum length.

**The Read-once Unit Resolution Refutation (UROR) problem:** Given a CNF formula  $\Phi$ , determine whether or not  $\Phi$  has a read-once unit resolution refutation.

**The Copy Complexity problem:** Given a CNF formula  $\Phi$ , determine the copy complexity of  $\Phi$  with respect to read-once unit resolution refutation.

### 3 The OLROR Problem for Horn Formulas

In this section, we discuss the problem of finding the optimal read-once refutation of a Horn formula.

Let  $\Phi$  be an unsatisfiable Horn formula. We know that  $\Phi$  has a read-once refutation [12]. The question whether  $\Phi$  has a read-once resolution of length less than  $k$  is equivalent to the question whether  $\Phi$  contains a minimal unsatisfiable formula consisting of at most  $k$  clauses.

**Theorem 1.** *Let  $R$  denote an OLROR of an unsatisfiable Horn formula  $\Phi$  and let  $\Phi_R \subseteq \Phi$  be the set of clauses used by  $R$ .  $R$  is also an optimal tree-like refutation of  $\Phi$  and an optimal dag-like refutation of  $\Phi$ . Additionally,  $\Phi_R$  is a minimum unsatisfiable subset of  $\Phi$ .*

*Proof.* Since  $R$  is a read-once refutation of  $\Phi$ ,  $R$  is also a tree-like and a dag-like refutation of  $\Phi$ .

Assume that  $T$  is a tree like refutation of  $\Phi$  such that  $|T| < |R|$ . Let  $\Phi_T \subseteq \Phi$  be the set of clauses used by  $T$ . It follows that  $|\Phi_T| \leq |T| + 1 < |R| + 1$ .

$\Phi_T$  must have a read-once refutation,  $R_T$  [12]. However,  $|R_T| \leq |\Phi_T| - 1 < |R|$ . This contradicts the fact that  $R$  is the *optimal* read-once refutation of  $\Phi$ . Thus,  $R$  is also an optimal tree-like refutation of  $\Phi$ . Similarly,  $R$  is an optimal dag-like refutation of  $\Phi$ .

Let  $\Phi' \subseteq \Phi$  be an unsatisfiable Horn formula such that  $|\Phi'| < |\Phi_R|$ .  $\Phi'$  must have a read-once refutation,  $R'$ . However,  $|R'| \leq |\Phi'| - 1 < |\Phi_R| - 1 = |R|$ . This contradicts the fact that  $R$  is the *optimal* read-once refutation of  $\Phi$ . Thus,  $\Phi_R$  is a minimum unsatisfiable subset of  $\Phi$ .  $\square$

We conclude that, for unsatisfiable Horn formulas, the length of the shortest resolution refutation equals the length of the shortest read-once refutation and the shortest tree-like resolution. Moreover, if  $k$  is the number of clauses of a minimal unsatisfiable subformula, then the shortest resolution refutation uses  $(k - 1)$  resolution steps.

The following result is a corollary of Theorem 5.2 in [7]. However, it is included here for completeness.

**Theorem 2.** *The problem of deciding whether a Horn formula contains an unsatisfiable sub-formula with at most  $k$  clauses is **NP-complete**.*

*Proof.* We show this by a reduction from Vertex Cover. Let  $G = (V, E)$  be an undirected graph where  $V = \{v_1, \dots, v_n\}$  and  $E = \{e_1, \dots, e_m\}$ . We associate with  $G$  the Horn formula:

$$(v_1) \wedge \dots \wedge (v_n) \wedge (\neg e_1 \vee \dots \vee \neg e_m) \wedge \bigwedge_{1 \leq i \leq m, e_i = (v_{i_1}, v_{i_2})} ((\neg v_{i_1} \vee e_i) \wedge (\neg v_{i_2} \vee e_i))$$

Then there exists a subset  $V' \subseteq V$  such that  $|V'| \leq r$  and  $V' \cap e_i$  is non-empty for every  $1 \leq i \leq m$  if and only if the associated formula contains an unsatisfiable sub-formula with at most  $(1 + m + r)$  clauses. (the negative clause, the clause  $(\neg v \vee e_i)$  for each  $1 \leq i \leq m$ , and  $r$  unit clauses).

Since the problem of finding a vertex cover of size at most  $r$  is **NP-complete**, the problem of finding a read-once resolution refutation of length at most  $k = (1 + m + r)$  is **NP-complete** for Horn formulas.  $\square$

## 4 The UROR Problem for 2-Horn Formulas

In this section, we look at the UROR problem for 2-Horn formulas. In particular, we show that checking whether a 2-Horn formula has a unit read-once refutation is in **P**.

As in the case of Horn formulas, not every unsatisfiable 2-Horn formula has a read-once unit resolution refutation.

*Example 2.* Consider the formula

$$(x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2).$$

This formula is an unsatisfiable Horn formula. However, the clause  $(x_1)$  needs to be used twice in any unit resolution refutation. Thus, this formula does not have a read-once unit resolution refutation.

Let  $\Phi$  be a 2-Horn formula with  $m$  clauses over  $n$  variables. We can divide  $\Phi$  into a set of positive unit clauses ( $PU$ ), a set of implications ( $\phi$ ), a set of negative unit clauses ( $NU$ ), and a set of clauses with two negative literals ( $N$ ).

We write  $\Phi = PU \cup \phi \cup NU \cup N$ . Note that any part ( $PU$ ,  $\phi$ ,  $NU$ , or  $N$ ) can be empty.

For resolution or restricted resolution calculi we say a formula  $\Phi$  is *minimal* with respect to the calculus, if there exists a refutation for  $\Phi$  and no proper sub-formula of  $\Phi$  has a refutation.

Let  $\mathbf{R}$  be a read-once unit resolution of  $\Phi$ . Let  $\Phi_k = PU_k \cup \phi_k \cup NU_k \cup N_k$  be the set of clauses derived from  $\Phi$  after  $k$  steps of  $\mathbf{R}$ . Let us consider the  $(k+1)^{th}$  resolution step of  $\mathbf{R}$ .

Note that this resolution step must resolve one of the following clause pairs:

1. A clause of the form  $(x_i) \in PU_k$  and a clause of the form  $(\neg x_i) \in NU_k$ . In this case, this resolution step produces the empty cause. Note that this is the last step of the refutation.
2. A clause of the form  $(x_i) \in PU_k$  and a clause of the form  $(\neg x_i \vee \neg x_j) \in N_k$ . In this case, the resolution produces the negative unit clause  $(\neg x_j)$ . Thus,  $PU_{k+1} = PU_k \setminus \{(x_i)\}$ ,  $N_{k+1} = N_k \setminus \{(\neg x_i \vee \neg x_j)\}$ , and  $NU_{k+1} = NU_k \cup \{(\neg x_j)\}$ . Note that  $|PU_{k+1} \cup NU_{k+1}| = |PU_k \cup NU_k|$  and  $|NU_{k+1} \cup N_{k+1}| = |NU_k \cup N_k|$ .
3. A clause of the form  $(x_i) \in PU_k$  and a clause of the form  $(\neg x_i \vee x_j) \in \phi_k$ . In this case, the resolution produces the positive unit clause  $(x_j)$ . Thus,  $PU_{k+1} = (PU_k \setminus \{(x_i)\}) \cup \{(x_j)\}$ ,  $N_{k+1} = N_k$ , and  $NU_{k+1} = NU_k$ . Note that  $|PU_{k+1} \cup NU_{k+1}| = |PU_k \cup NU_k|$  and  $|NU_{k+1} \cup N_{k+1}| = |NU_k \cup N_k|$ .
4. A clause of the form  $(\neg x_i) \in NU_k$  and a clause of the form  $(x_i \vee \neg x_j) \in \phi_k$ . In this case, the resolution produces the negative unit clause  $(\neg x_j)$ . Thus,  $PU_{k+1} = PU_k$ ,  $N_{k+1} = N_k$ , and  $NU_{k+1} = (NU_k \setminus \{(\neg x_i)\}) \cup \{(\neg x_j)\}$ . Note that  $|PU_{k+1} \cup NU_{k+1}| = |PU_k \cup NU_k|$  and  $|NU_{k+1} \cup N_{k+1}| = |NU_k \cup N_k|$ .

Let  $\Phi = PU \cup \phi \cup NU \cup N$  be minimal with respect to read-once unit resolution. Let  $\mathbf{R}$  be a read-once unit resolution refutation of  $\Phi$  and let  $(x_1) \wedge (\neg x_1) \vdash_{RES}^1 \perp$ , be the last resolution step of  $\mathbf{R}$ . Based on the preceding arguments,  $\Phi$  must have the following properties:

1.  $|NU \cup N| = 1$ . Note that every prior resolution step of  $\mathbf{R}$  preserves the size of  $NU \cup N$ . Before the last resolution step of  $\mathbf{R}$ ,  $NU \cup N = \{(\neg x_i)\}$ . Thus, before the last resolution step  $|NU \cup N| = 1$ . This means that this must have been true for the original formula.
2.  $|PU \cup NU| = 2$ . Note that every prior resolution step of  $\mathbf{R}$  preserves the size of  $PU \cup NU$ . Before the last resolution step of  $\mathbf{R}$ ,  $PU \cup NU = \{(x_i), (\neg x_i)\}$ . Thus, before the last resolution step  $|PU \cup NU| = 2$ . This means that this must have been true for the original formula.

Thus,  $\Phi$  must satisfy one of the following conditions:

1.  $|PU| = 1$ ,  $|NU| = 1$ , and  $|N| = 0$ .
2.  $|PU| = 2$ ,  $|NU| = 0$ , and  $|N| = 1$ .

This means that we only need to consider two types of read-once unit resolution refutations.

1. Refutations which use a single positive unit clause from  $PU$  and a single negative unit clause from  $NU$ . We refer to this as a Type 1 refutation.
2. Refutations which use two positive unit clauses from  $PU$  and a single negative clause from  $N$ . We refer to this as a Type 2 refutation.

From an arbitrary 2-Horn formula  $\Phi = PU \cup \phi \cup NU \cup N$ , we can create a directed graph  $\mathbf{G} = \langle \mathbf{V}, \mathbf{E} \rangle$  as follows:

1. For each variable  $x_i$  in  $\Phi$ , add the vertex  $v_i$  to  $\mathbf{V}$ .
2. For each clause  $(x_i \vee \neg x_j)$ , add the edge  $(v_j, v_i)$  to  $\mathbf{E}$ .
3. Let  $S = \{v_i | (x_i) \in PU\}$  and  $T = \{v_j | (\neg x_j) \in NU\}$ .

We utilize  $\mathbf{G}$  to find read-once unit resolution refutations.

**Lemma 1.** *The clause  $(x_i)$  is derivable from  $\Phi$  by a read-once unit resolution if and only if  $v_i$  is reachable in  $\mathbf{G}$  from a vertex in  $S$ .*

*Proof.* Assume that  $(x_i)$  is derivable from  $\Phi$  by a read-once unit resolution  $\mathbf{R}$ . We will show that  $v_i$  is reachable from a vertex in  $S$  by induction based on the number of resolution steps in  $\mathbf{R}$ .

If  $\mathbf{R}$  has no resolution steps then  $(x_i) \in \Phi$ . Thus,  $(x_i) \in PU$  and, by construction,  $v_i \in S$ . Thus,  $v_i$  is trivially reachable from a vertex in  $S$ .

Now suppose that this holds true of all resolutions with  $k$  steps. If  $\mathbf{R}$  has  $(k+1)$  steps, then the last resolution step of  $\mathbf{R}$  must be  $(x_j) \wedge (\neg x_j \vee x_i) \stackrel{1}{\text{RES}} (x_i)$  for some clause  $(x_j)$ . This means that  $(x_j)$  must be derivable from  $\Phi$  by a read-once unit resolution with  $k$  steps. Thus, by the induction hypothesis,  $v_j$  is reachable from a vertex in  $S$ . Thus,  $(\neg x_j \vee x_i) \in \Phi$  since non-unit clauses are not derivable from 2-Horn clauses by unit resolution. This means that  $(\neg x_j \vee x_i) \in \phi$  and, by construction, the edge  $(v_j, v_i)$  is in  $\mathbf{E}$ . Thus,  $v_i$  is also reachable from a vertex in  $S$ .

Now assume that  $v_i$  is reachable from a vertex  $v_j \in S$ . Thus, there must be a simple path  $p$  in  $\mathbf{G}$  from  $v_j$  to  $v_i$ . Let the edges in  $p$  be  $(v_j, v_{p_1}), (v_{p_1}, v_{p_2}), \dots, (v_{p_k}, v_i)$ . By construction of  $\mathbf{G}$ , the clauses  $(x_j), (\neg x_j \vee x_{p_1}), (\neg x_{p_1} \vee x_{p_2}), \dots, (\neg x_{p_k} \vee x_i)$  are all in  $\Phi$ .

It is easy to see that these clauses can be used to derive  $(x_i)$  through unit resolution. Since  $p$  was simple no clause is used more than once. Thus, this is a read-once unit resolution of  $(x_i)$ .  $\square$

**Theorem 3.** *Whether the formula  $\Phi$  has a Type 1 refutation can be determined in linear time.*

*Proof.*  $\Phi$  has a Type 1 refutation if and only if for some clause  $(\neg x_t) \in \Phi$ , we can derive the clause  $(x_t)$  from  $\Phi$  by a read-once unit resolution  $\mathbf{R}$ . Thus, by Lemma 1, some vertex  $v_t \in T$  must be reachable from a vertex in  $S$ . This can be checked by doing a depth-first search in  $\mathbf{G}$ .  $\square$

**Theorem 4.** *Whether the formula  $\Phi$  has a Type 2 refutation can be determined in  $O(m^2)$  time.*



*Proof.*  $\Phi$  has a Type 2 refutation if and only if for some clause  $(\neg x_{t_1} \vee \neg x_{t_2}) \in \Phi$ , we can derive the clauses  $(x_{t_1})$  and  $(x_{t_2})$  from  $\Phi$  by read-once unit resolution  $\mathbf{R}$ . Thus, by Lemma 1, both  $v_{t_1}$  and  $v_{t_2}$  must be reachable from vertices in  $S$ . Thus, there is a path  $p_1$  from a vertex in  $S$  to  $v_{t_1}$  and there is a path  $p_2$  from a vertex in  $S$  to  $v_{t_2}$ . Since  $\mathbf{R}$  is a read-once unit resolution,  $p_1$  and  $p_2$  cannot share any edges and cannot have the same starting vertex.

If we are given vertices  $v_{t_1}$  and  $v_{t_2}$ , then we need to check if  $\mathbf{G}$  has such a pair of edge-disjoint paths.

From  $\mathbf{G}$ ,  $S$ ,  $v_{t_1}$ , and  $v_{t_2}$ , we can construct flow network  $\mathbf{G}'$  as follows:

1. For each vertex  $v_i$  in  $\mathbf{G}$ , add the node  $v_i$  to  $\mathbf{G}'$ .
2. For each edge  $(v_i, v_j)$  in  $\mathbf{G}$  add the edge  $(v_i, v_j)$  to  $\mathbf{G}'$  with capacity 1.
3. Add a source  $s$  and the edges  $(s, v_i)$  for  $v_i \in S$  to  $\mathbf{G}'$ .
4. Add a sink  $t$  and the edges  $(v_{t_1}, t)$ , and  $(v_{t_2}, t)$  to  $\mathbf{G}'$ .

The desired paths exist if and only if a flow of 2 can be pushed from  $s$  to  $t$  in  $\mathbf{G}'$ .

Thus,  $\Phi$  has a Type 2 refutation if and only if, for some clause  $(\neg x_{t_1} \vee \neg x_{t_2}) \in \Phi$ , the graph  $\mathbf{G}'$  has a max flow of 2.

Note that all edges in  $\mathbf{G}'$  have capacity 1, and that  $t$  has only two incoming edges. This means that the maximum flow is at most 2. Thus, we can find the max flow from  $s$  to  $t$  in  $\mathbf{G}'$  in  $O(m)$  time where  $m$  is the number of clauses in  $\Phi$  [2]. Since we need to find this flow for each clause of the form  $(\neg x_{t_1} \vee \neg x_{t_2})$ , determining if  $\Phi$  has a Type 2 refutation can be accomplished in  $O(m^2)$  time.  $\square$

Thus, checking if  $\Phi$  has a read-once unit resolution refutation can be accomplished by checking to see if it has either a Type 1 refutation or a Type 2 refutation. This can be accomplished in  $O(m^2)$  time since checking for Type 2 refutations is the more time consuming process.

## 5 UROR Copy Complexity of Horn Formulas

We now examine the copy complexity of Horn formulas with respect to unit resolution.

**Theorem 5.** *The copy complexity of Horn formulas with respect to unit resolution is at most  $2^{n-1}$  where  $n$  is the number of variables.*

*Proof.* Suppose  $\Phi$  is an unsatisfiable Horn formula. Note that adding clauses to a system cannot increase the copy complexity. Thus, we can assume without loss of generality that  $\Phi$  is minimal unsatisfiable. Let  $CC(n)$  denote the copy complexity of a minimal unsatisfiable Horn formula with  $n$  variables. We will show that  $CC(n) \leq 2^{n-1}$ . For a clause  $\phi_j$  of  $\Phi$  let  $N_c(\phi_j)$  be the number of copies of  $\phi_j$  needed for a read-once unit resolution refutation.

Let  $\Phi$  be a minimal unsatisfiable Horn formula with  $n$  variables. Thus,  $\Phi$  has  $(n + 1)$  clauses. If  $n = 1$ , then  $\Phi$  has the form  $(x) \wedge (\neg x)$ . This formula has

a read-once unit resolution refutation. Thus  $CC(1) = 1 \leq 2^0$ . Also note that  $\sum_{\phi_j \in \Phi} N_c(\phi_j) = 2 \leq 2^1$ .

Now assume that  $CC(k) \leq 2^{k-1}$ , and that for each minimal unsatisfiable formula  $\Phi'$  with  $k$  variables,  $\sum_{\phi'_j \in \Phi'} N_c(\phi'_j) \leq 2^k$ . If  $n = k + 1$ , then  $\Phi$  has the form  $(x) \wedge (\neg x \vee \alpha_1) \wedge \dots \wedge (\neg x \vee \alpha_t) \wedge \sigma_{t+1} \wedge \dots \wedge \sigma_{k+1}$ . A read-once unit resolution refutation needs to use the clause  $(x)$  to eliminate each instance of  $\neg x$ . Thus, we need a copy of the clause  $(x)$  for each copy of  $(\neg x \vee \alpha_i)$  for  $i = 1 \dots t$ . Let  $\Phi'$  be the formula  $\alpha_1 \wedge \dots \wedge \alpha_t \wedge \sigma_{t+1} \wedge \dots \wedge \sigma_{k+1}$ . Note that  $\Phi'$  is a minimal unsatisfiable formula with  $n - 1 = k$  variables. Thus,

$$\sum_{j=1}^t N_c(\alpha_i) \leq \sum_{\phi'_j \in \Phi'} N_c(\phi'_j) \leq 2^k.$$

This means that we need at most  $2^k$  copies of the clause  $(x)$ . Thus,  $CC(k+1) \leq 2^k$  and  $\sum_{\phi_j \in \Phi} N_c(\phi_j) \leq 2^k + 2^k = 2^{k+1}$ .  $\square$

**Theorem 6.** *There exists a Horn formula with copy complexity  $2^{n-1}$  where  $n$  is the number of variables.*

*Proof.* Consider the following clauses:

$$\begin{array}{ccc} (\neg x_1 \vee \neg x_2 \vee \dots \vee \neg x_n) & (x_1 \vee \neg x_2 \vee \dots \vee \neg x_n) & (x_2 \vee \neg x_3 \vee \dots \vee \neg x_n) \\ \dots & (x_{n-1} \vee \neg x_n) & (x_n) \end{array}$$

Let us consider the clause  $(\neg x_1 \vee \neg x_2 \vee \dots \vee \neg x_n)$ . To eliminate this clause through unit resolution, we need to derive the clauses  $(x_1), (x_2), \dots, (x_n)$ . We will prove by induction, that for each  $i = 1 \dots n$ ,  $2^{i-1}$  copies of the clause  $(x_i \vee \neg x_{i+1} \vee \dots \vee \neg x_n)$  are required.

To eliminate  $\neg x_1$  from the clause we only need one copy of the clause  $(x_1)$ , thus we only need  $1 = 2^0$  copies of the clause  $(x_1 \vee \neg x_2 \vee \dots \vee \neg x_n)$ .

Now assume that for each  $i < k$ , we need to use  $2^{i-1}$  copies of the clause  $(x_i \vee \neg x_{i+1} \vee \dots \vee \neg x_n)$ . Each of these clauses uses the literal  $\neg x_k$ , as does the clause  $(\neg x_1 \vee \neg x_2 \vee \dots \vee \neg x_n)$ . Thus, we need a total of  $1 + \sum_{i=1}^{k-1} 2^{i-1} = 2^{k-1}$  copies of the clause  $(x_k)$  to cancel every instance of  $\neg x_k$ . The only clause with the literal  $x_k$  is the clause  $(x_k \vee \neg x_{k+1} \vee \dots \vee \neg x_n)$ . Thus, we need to use  $2^{k-1}$  copies of this clause, as desired.

This means that we need a total of  $2^{n-1}$  copies of the clause  $(x_n)$ . Thus, the copy complexity of this formula is  $2^{n-1}$  with respect to unit resolution.  $\square$

From these two results, it is easy to see that the copy complexity of Horn formulas with respect to read-once unit resolution refutation is  $2^{n-1}$ .

## 6 Conclusion

In this paper, we studied two proof systems, viz., Read-once resolution (ROR) and Unit read-once resolution (UROR) from the perspective of Horn clause systems. Our work is motivated by two important factors, viz., the ubiquitousness

of the resolution proof system in SMT solvers and the wide applicability of Horn formulas. As discussed before, the ROR and UROR proof systems are incomplete for general CNF formulas, although ROR is complete for Horn formulas. Note that if an unsatisfiable formula has a read-once refutation or unit read-once refutation, then this refutation is necessarily short. This is in contrast to general resolution proofs, which could be exponentially large with respect to input size. Our investigations established that the problem of checking whether a Horn clause system has a read-once refutation of length at most  $k$  is **NP-hard**. We also showed that the problem of finding a UROR for 2-Horn formulas is in **P**. Finally, we discussed the copy complexity of Horn formulas with respect to unit resolution and obtained an exponential lower bound.

## References

1. Chandrasekaran, R., Subramani, K.: A combinatorial algorithm for Horn programs. *Discrete Optim.* **10**, 85–101 (2013)
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. MIT Press, Cambridge (2001)
3. Cousot, P., Cousot, R.: Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: *POPL*, pp. 238–252 (1977)
4. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Program.* **1**(3), 267–284 (1984)
5. Gallier, J.H.: *Logic for Computer Science: Foundations for Automatic Theorem Proving*. Dover Books on Computer Science, 1st edn. Dover Publications (2015)
6. Iwama, K., Miyano, E.: Intractability of read-once resolution. In: *Proceedings of the 10th Annual Conference on Structure in Complexity Theory (SCTC 1995)*, Los Alamitos, CA, USA, June 1995, pp. 29–36. IEEE Computer Society Press (1995)
7. Liberatore, P.: Redundancy in logic ii: 2CNF and horn propositional formulae. *Artif. Intell.* **172**(2), 265–299 (2008)
8. Owre, S., Shankar, N.: *The formal semantics of PVS*, March 1999
9. Robinson, J.A., Voronkov, A.: *Handbook of Automated Reasoning* (2 volume set). MIT Press, Cambridge (2001)
10. Robinson, J.A.: A machine-oriented logic based on the resolution principle. *J. ACM* **12**(1), 23–41 (1965)
11. Rushby, J.M., Owre, S., Shankar, N.: Subtypes for specifications: predicate subtyping in PVS. *IEEE Trans. Softw. Eng.* **24**(9), 709–720 (1998)
12. Szeider, S.: NP-completeness of refutability by literal-once resolution. In: Goré, R., Leitsch, A., Nipkow, T. (eds.) *IJCAR 2001*. LNCS, vol. 2083, pp. 168–181. Springer, Heidelberg (2001). <https://doi.org/10.1007/3-540-45744-5-13>