Article

Set-Blocked Clause and Extended Set-Blocked Clause in First-Order Logic

Xinran Ning *, Yang Xu, Guanfeng Wu and Huimin Fu

National-Local Joint Engineering Laboratory of System Credibility Automatic Verification, Southwest Jiaotong University, Chengdu 610031, China; xuyang@swjtu.edu.cn (Y.X.); wl520gx@gmail.com (G.W.); fhm6688@my.swjtu.edu (H.F.)

* Correspondence: xinxinran@my.swjtu.edu.cn; Tel.: +86-158-2830-6582

Received: 11 October 2018; Accepted: 24 October 2018; Published: 28 October 2018

Abstract: Due to scale and complexity of first-order formulas, simplifications play a very important role in first-order theorem proving, in which removal of clauses and literals identified as redundant is a significant component. In this paper, four types of clauses with the local redundancy property were proposed, separately called a set-blocked clause (SBC), extended set-blocked clause (E-SBC), equality-set-blocked clause (ESBC) and extended equality-set-blocked clause (E-ESBC). The former two are redundant clauses in first-order formulas without equality while the latter two are redundant clauses in first-order formulas with equality. In addition, to prove the correctness of the four proposals, the redundancy of the four kinds of clauses were proved. It was guaranteed eliminating clauses with the four forms has no effect on the satisfiability or the unsatisfiability of original formulas. In the end, effectiveness and confluence properties of corresponding clause elimination methods were analyzed and compared.

Keywords: set-blocked clause; extended set-blocked clause; first-order logic

1. Introduction

Simplifications have been diffusely recognized as an indispensable component of both propositional SAT solving and first-order theorem proving. Clause elimination has always been crucial among simplification techniques. In 2010, blocked clause elimination in propositional logic was proposed for reducing the size of formulas and speeding up SAT solvers as a preprocessing technique [1]. After that, there have been published further research work related to blocked clause. Hidden blocked clause and asymmetric blocked clause were created by combining hidden literal addition and asymmetric literal addition with blocked clause at the same year [2,3]. Then, abcdSAT [4] based on blocked clause decomposition [5] won the SAT-Race 2015 competition. In 2016, the extension of blocked clause was widened further. Super-blocked clause and set-blocked clause in propositional logic were produced, in which super-blocked clause had the most general local redundancy property [6]. After that, Blocked clause was successfully lifted to first-order logic in 2017 [7], which as a preprocessing technique of Vampire boosted the efficiency of Vampire [8].

In the paper, we generalize the conception of blocked clause further in first-order logic. Set-blocked clause (SBC) and extended set-blocked clause (E-SBC) in first-order logic without equality are presented, which are generalizations of blocked clause in first-order logic. The evolution is that SBC is blocked by a subset of its literals while blocked clause (BC) is blocked by one of its literals. Similarly, E-SBC can be considered as a further generalization based on SBC. Informally, if a clause C is an E-SBC, for any assignment β over the external ground atoms of the clause C, there exists a subset $S_\beta$ of literals of C, C is a SBC upon $S_\beta$ in $F|\beta$. Assignments over the external ground atoms of C may transform resolution environment of C into a different one from the original resolution environment, in addition, the subset $S_\beta$ is able to be various along with the diversity of the assignment β, which makes the requirement for
a clause to be an extended set-blocked clause is quite flexible compared with the requirement for an SBC. To guarantee those clauses are capable to be eliminated from formulas without influencing the satisfiability or unsatisfiability, the redundancy of the two categories of clauses are proved. The proof is not given directly under the circumstance of first-order logic but lower it to propositional logic and connect with the variant of Herbrand’s Theorem: if a first-order formula \( F \) is satisfiable if and only if it is satisfiable for all the finite ground instances of \( F \). After that, their abilities to simplify formulas are contrasted with blocked clause elimination (BCE) by comparing their effectiveness. Furthermore, set-blocked clause elimination (SBCE) and extended set-blocked clause elimination (E-SBCE) have the confluence properties just the same as blocked clause elimination.

The paper is not only relevant to first-order formulas without equality but also first-order formulas with equality. Because of the peculiarities of equality, there will be some mistakes if we remove clauses according to the definitions of SBC and E-SBC. The solution is to combine SBC and E-SBC with flattening and flat resolution, developing the new two categories of clauses: equality-set-blocked clause (ESBC) and extend equality-set-blocked clause (E-ESBC). The combination can erase the “confusion jamming” caused by the characteristics of equality that some different literals with distinct items have the same truth values under any assignment. Similarly, redundancy, effectiveness and confluence properties of ESBC and E-ESEB are also demonstrated, analyzed and compared.

The contribution of the paper mainly is: (1) Establish the concepts of SBC and E-SBC in formulas without equality and ESBC and E-ESBC in formulas with equality; (2) Prove all the four types of clauses are redundant; (3) Contrast the four corresponding clause elimination methods’ effectiveness; (4) Illuminate the confluence properties of the four clause elimination methods.

The rest of the paper is organized as follows. After some necessary preliminaries are introduced in Section 2, we propose SBC and E-SBC, and prove the redundancy of the two types of clauses in Section 3. In Section 4, we present ESBC and E-ESBC, show how they can deal with formulas with equality and prove the redundancy of the two types of clauses. Finally, we compare and analyze those clause elimination methods’ effectiveness and confluence properties in Section 5.

2. Preliminaries

In the section, we introduce some necessary notations, definitions and theorems for the paper.

Here we only consider first-order formulas in conjunctive normal form (CNF). A formula is a conjunction of clauses. A clause is a disjunction of literals. A literal is an atom or the negation of an atom. An atom is made up of a predicate symbol and items. Items can be the mixture of variables, constants and functions or just single variable, constant or function. Variables are usually represented by \( x, y, \ldots \), constants are represented by \( a, b, c \ldots \) and functions are represented by \( f, g, h \ldots \).

A propositional assignment is a mapping from ground atoms to the truth values 1 (true) and 0 (false). Accordingly, a set \( V \) of ground clause is propositionally satisfiable if there exists a propositional assignment which can assign every ground clause in \( V \) to the truth value 1. A clause is valid when it is true under any assignment, tautology is one case of valid clause. Let \( F \) be a formula and \( \alpha \) be an assignment, \( F[\alpha] \) is defined as \( \{C|C \in F \text{ and } \alpha \text{ does not satisfy } C\} \). Two formulas \( F \) and \( F' \) are satisfiability equivalent if they are either both satisfiable or unsatisfiable. A clause \( C \) is redundant w.r.t. \( F \) if \( F \) and \( F\setminus\{C\} \) are satisfiability equivalent. A substitution is a mapping from variables to terms. A ground substitution is a mapping of which the range consists only of ground terms. For a literal, clause or formula \( F, \text{atom}(F) \) denotes the atoms in \( F \) and \( \text{Gatom}(F) \) denotes the ground atoms in \( F \).

A clause is a blocked clause upon \( L \in C \) in a first-order formula \( F \) without equality, if all its \( L \)-resolvents are tautologies. \( L \)-resolvent is defined below [7]:

**Definition 1.** Given two clauses \( C = L \lor C_1 \) and \( D = N_1 \lor \ldots \lor N_m \lor D_1 \) such that there exists a substitution \( \sigma \) which can unify \( L, \neg N_1 \ldots \neg N_m \lor D_1 \lor C_1 \lor D_1 ) \sigma \) is called as the \( L \)-resolvent of \( C \) and \( D \).
For a clause \( C \) in a formula \( F \), the resolution environment, \( env_F(C) \), of \( C \) in first-order formula \( F \) without equality is the set of all the clauses in \( F \setminus \{ C \} \) which can be resolved with \( C \): \( env_F(C) = \{ C' \in F \setminus \{ C \} | \exists L' \text{ in } C' \text{ such that } L \in C \text{ and } L' \text{ can be unified} \} \).

And for those atoms in \( env_F(C) \) but not appearing in \( C \) and not able to be unified with \( \text{atom}\{C\} \), they are called as the external atoms of \( C \). The definition is: \( \text{ext}_F(C) = \{ A | A \in \text{atom}\{env_F(C)\}, A \notin \text{atom}\{C\} \} \). The set of external ground atoms is defined as \( \text{ext}_{GF}(C) \).

Since first-order CNF formulas with equality will be discussed in this paper, equality axioms \( \varepsilon_C \) is introduced here [10]: (1) \( x = x \); (2) \( x \neq y \lor y = x \); (3) \( x \neq y \lor y \neq z \lor z = x \); (4) if \( f \) is a \( n \)-ary function symbol, \( x_1 \neq y_1 \lor \ldots \lor x_n \neq y_n \lor f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n) \); (5) if \( P \) is a \( n \)-ary predicate symbol, \( x_1 \neq y_1 \lor \ldots \lor x_n \neq y_n \lor P(x_1, \ldots, x_n) \lor \neg P(y_1, \ldots, y_n) \).

Flattening and flat resolution [11] are the constituent parts of E-SBC and E-ESBC. The definitions are showed below [7].

**Definition 2.** After flattening the literal \( L(s_1, s_2, \ldots, s_n) \) of the clause \( C = L(s_1, s_2, \ldots, s_n) \lor C_1 \), the new flattened literal is \( L' = L(x_1, x_2, \ldots, x_n) \lor C_1 \) and the new flattened clause is \( C' = \lor_{1 \leq i \leq n} x_i \neq s_i \lor L(x_1, x_2, \ldots, x_n) \lor C_1 \) with \( x_i (i = 1, 2, \ldots, n) \) being variables not occurring in \( C \).

**Definition 3.** Suppose \( C = L \lor C_1 \) and \( D = N_1 \lor N_2 \lor \ldots \lor N_m \lor D_1 \) and \( L, -N_1, \ldots, -N_m \) have the same predicate symbol but the opposite polarity. After flattening \( L \) in \( C \) and \( N_1, \ldots, N_m \) in \( D \), the new flattened clauses \( C' = L' \lor C_1 \) and \( D' = N_1' \lor \ldots \lor N_m' \lor D_1' \) are obtained. In addition, the resolvent \( (C' \setminus \{L'\}) \sigma \lor (D' \setminus \{N_1', \ldots, N_m'\}) \sigma \) of \( C' \) and \( D' \) is called a flat \( L \)-resolvent of \( C \) and \( D \).

A clause \( C \) is an equality-blocked clause (EBC) upon \( L \lor C \) in a formula \( F \) with equality, if all its flat \( L \)-resolvents are valid.

The resolution environment of a clause \( C \) in a formula \( F \) with equality is different from the definition in formulas without equality. The existence of equality makes the clauses own literals, which can be resolved with some literal in \( C \), cannot consist of the whole resolution environment of \( C \), but extend to clauses contain literals have the same predicate but the contrary polarity with some literal in \( C \). \( env_{F_E}(C) = \{ C' \in F \setminus \{ C \} | \exists L' \in C' \text{ such that } L \in C \text{ and } L' \text{ have the same predicate symbol} \} \).

Meanwhile, the external atoms in \( env_{F_E}(C) \), of \( C \) in first-order formulas with equality are also distinguished from the definition of external atom in first-order formulas without equality: \( \text{ext}_{F_E}(C) = \{ A | A \in \text{atom}\{env_{F_E}(C)\}, A \notin \text{atom}\{C\} \} \) in which the set of ground external atoms is defined as \( \text{ext}_{GF}(C) \).

In a formula \( F \) with equality, flipping the truth value of a ground literal \( L = P(s_1, \ldots, s_n) \) under a propositional assignment \( \beta \) should flip the truth value of all the ground literals with the form \( L' = P(t_1, \ldots, t_n) \) such that \( \beta(s_i = t_i) = 1 \) for all \( 1 \leq i \leq n \) rather than simply flip the truth value of \( L \) [7].

**Definition 4.** Given a propositional assignment \( \beta \) and a ground literal \( L = P(t_1, \ldots, t_n) \) of which the predicate symbol is not equality symbol. The definition of a propositional assignment \( \beta' \) is obtained from \( \beta \) by equivalence flipping the truth value of \( L = P(t_1, \ldots, t_n) \) is below:

\[
\beta'(L) = \begin{cases} 
1 - \beta(L) & \text{if } L = P(s_1, s_2, \ldots, s_n) \text{ and } \beta(t_i \approx s_i) = 1 \text{ for all } 1 \leq i \leq n \\
\beta(L) & \text{otherwise}
\end{cases}
\]

The following are two variants of Herbrand’s Theorm we adopt in the rest part of the paper. One is suitable for first-order CNF formulas without equality, while the other is suitable for first-order CNF formulas with equality [10].
Theorem 1. For a first-order formula $F$ without equality predicate, it is satisfiable if and only if every finite set of ground instances of clauses in $F$ is propositionally satisfiable.

Theorem 2. For a first-order formula $F$ with equality predicate, it is satisfiable if and only if every finite set of ground instances of clauses in $F \cup \varepsilon_L$ is propositionally satisfiable.

3. Set-Blocked Clause and Extended Set-Blocked Clause in First-Order Logic without Equality

In this section, we demonstrate set-blocked clause and extended set-blocked clause in formulas without equality. First, we give the definition of $L^S$-resolvent as follows, which is different from the definition of $L$-resolvent of a clause $C$ in the paper [7]. The former includes substituted literals from negations of some literals in the clause $C$ while the latter has no such characteristics in it.

Definition 5. Given two clauses, $C = L_1 \lor L_2 \lor \ldots \lor L_n \lor C'$ with the subset $S = \{L_1, L_2, \ldots, L_n\} \subseteq C$ such that $L_i(1 \leq i \leq n)$ and $L_j(1 \leq j \leq n)$ cannot be resolved with each other, and $D = N_1 \lor N_2 \lor \ldots \lor N_m \lor D'$. If one of literals $L_i \in S(1 \leq i \leq n), \neg N_1, \ldots, \neg N_m$ can be unified by an mgu $\sigma$, the clause $C'\sigma \lor D'\sigma \subset (S' \setminus \{L_i\})\sigma$ is called $L^S$-resolvent of $C$ and $D$.

Example 1. Let the clause $C = Q(b) \lor P(b) \lor R(c)$ with $S = \{Q(b), P(b), R(c)\}$ and the clause $D = \neg P(b) \lor \neg P(x) \lor Q(x)$. $P(b), P(b)$ and $P(x)$ can be unified by the substitution $\sigma = \{x \mapsto b\}$. We can know that $C'\sigma = \Phi, D'\sigma = Q(b)$ and $(S' \setminus \{L_1\})\sigma = \neg Q(b) \lor \neg R(c)$. Therefore, $Q(b) \lor \neg Q(b) \lor \neg R(c)$ is the $P(b)\{Q(b), P(b), R(c)\}$-resolvent of $C$ and $D$.

Next is the definition of set-blocked clause, a generalization from blocked clause by extending one blocking literal of clause $C$ to multiple blocking literals of clause $C$.

Definition 6. A clause $C$ is a set-blocked clause (SBC) in the formula $F$ if there exists a set $S = \{L_1, L_2, \ldots, L_n\} \subseteq C$ such that $L_i(1 \leq i \leq n)$ and $L_j(1 \leq j \leq n)$ cannot be resolved with each other and all its $L_i^S$-resolvent $(1 \leq i \leq n)$ are tautologies.

Example 2. Let the clause $C = Q(a) \lor P(b) \lor R(c), S = \{Q(a), P(b)\}$ and the formula $F = \{Q(a) \lor P(b) \lor R(c), \neg P(x) \lor Q(a), \neg Q(a) \lor P(b), P(x) \lor \neg R(x)\}$. We can see that there is only one $P(b)\{Q(a), P(b)\}$-resolvent and one $Q(a)\{Q(a), P(b)\}$-resolvent of $C$ and $D$, separately obtained by resolving $C$ with $\neg P(x) \lor Q(a)$ and resolving $C$ with $\neg Q(a) \lor P(b), P(b)\{Q(a), P(b)\}$-resolvent is $R(c) \lor Q(a) \lor \neg Q(a)$ which is a tautology and $Q(a)\{Q(a), P(b)\}$-resolvent is $R(c) \lor P(b) \lor \neg P(b)$ which is also a tautology. Then, all the $L^S_i$-resolvents $(1 \leq i \leq n)$ are tautologies. Therefore, $C$ is set-blocked upon $S$ w.r.t. $F$.

In Example 2, the clause $C$ is an SBC; however, the clause $C$ is not blocked no matter upon $Q(a), P(b)$ or $R(c)$. To justify it has no impact on the satisfiability or unsatisfiability of formulas by removing SBCs in formulas, the redundancy property of SBC is proved subsequently.

Lemma 1. Given a clause $C$ is set-blocked upon $S = \{L_1, L_2, \ldots, L_n\} \subseteq C$ in a formula $F$. Let $\alpha$ be an assignment that propositionally satisfies all the ground instances of clauses in $F \setminus \{C\}$ but falsifies a ground instance $CA$ of $C$. Then, the assignment $\alpha'$, obtained from $\alpha$ by flipping all the truth values of ground literals $L_1\lambda, L_2\lambda, \ldots, L_n\lambda$ in $SA$, still satisfies all the ground instances in $F \setminus \{C\}$.

Proof. Let $D\tau$ be a ground instance of $D$ in $F \setminus \{C\}$. $D\tau$ may be falsified according to the assignment $\alpha'$ by flipping all the truth values of ground literals $L_1\lambda, L_2\lambda, \ldots, L_n\lambda$ in $SA$, only if $D\tau$ contains a set of literals $\{\neg L_1\lambda, \neg L_2\lambda, \ldots, \neg L_n\lambda\} \subseteq \{\neg L_1\lambda, \neg L_2\lambda, \ldots, \neg L_n\lambda\}$. Without loss of generality, assume that $D\tau$ contains the literal $\neg L_1\lambda$ (if $D\tau$ contains more than one literal from $\{\neg L_1\lambda, \neg L_2\lambda, \ldots, \neg L_n\lambda\}$, the proof process is the same) and let $N_1, \ldots, N_k$ be all the literals in $D$ such that $N_i\tau = \neg L_1\lambda$ for
We can obtain the following equation:

\[ S \]

**Theorem 3.**

If \( \text{value of Symmetry 2018} \) literals of the ground instances of \( F \) is finite, we can satisfy those ground instances of \( S \) according to Lemma 1, but it may falsify some other ground instances of \( S \). Ground literals of ground instance \( \{ F \} \) are not capable to be resolved with ground literals in \( C \). Assume \( C \) falsifies the other ground instance \( C \). Therefore, \( \alpha' \) satisfies \( D \). Hence, flipping the truth values of ground literals \( 1 \lambda_1, 2 \lambda_2, \ldots, n \lambda \) will not falsify any ground instances in \( F \setminus \{ C \} \), so \( \alpha' \) satisfies all the ground instances in \( F \setminus \{ C \} \).

Lemma 1 demonstrates that flipping the truth values of ground literals in \( S \) has no influence on the truth values of ground instances of clauses in \( F \setminus \{ C \} \). However, it may falsify other ground instances of \( C \). Here is an example.

**Example 3.** Assume \( C = \neg P(x) \land P(f(y)) \lor \neg Q(a) \lor Q(f(v)) \) is a SBC upon \( S = \{ P(f(y)), Q(f(v)) \} \) in a formula \( F \). Consider two ground instances of \( C, C_1 = \neg P(a) \lor P(f(a)) \lor \neg Q(a) \lor Q(f(a)) \), and \( C_2 = \neg P(a) \lor P(f(a)) \lor \neg Q(a) \lor Q(f(b)) \). The assignment \( \alpha(P(a)) = P(a)Q(a)Q(f(a))Q(f(b))Q(f(b)) \) falsifies \( C_1 \), but we can satisfy \( C_1 \) by flipping the truth values of ground instances \( P(f(a)) \land Q(f(a)) \) in \( S_1 \subseteq C_1 \) and we can obtain the new assignment \( \alpha(P(a))Q(a)Q(f(a))Q(f(b))Q(f(b)) \), apparently the new assignment satisfies \( C_1 \); however, it falsifies the other ground instance \( C_2 \) of \( C \).

Even though flipping the truth values of ground literals of the set \( S_1 \), one ground instance of \( S \) in a ground instance \( C_1 \) of \( C \), may falsify other ground instances of \( C \), it has no severe consequence. Assume that flipping the truth values of ground literals in \( S_1 \) falsifies another ground instance \( C_2 \) of \( C \), then there are no identical ground literals in \( S_2 \subseteq C_2 \), one ground instance of \( S \) in the ground instance \( C_2 \). Now that \( S_2 \) contains totally different ground literals from \( S_1 \), and ground literals in \( S_2 \) are not capable to be resolved with ground literals in \( S_1 \), according to the definition of SBC, then it will not falsify \( C_1 \) by flipping the truth value of ground literals in \( S_2 \). We can conclude that both \( C_1 \) and \( C_2 \) can keep their truth values as true, by flipping the truth values of ground literals as true in \( S_1 \) and \( S_2 \).

**Theorem 3.** If a clause \( C \) is set-blocked upon \( S = \{ L_1, L_2, \ldots, L_n \} \subseteq C \) in a formula \( F \), the clause \( C \) is redundant w.r.t. \( F \).

**Proof.** Given a clause \( C \) is set-blocked upon \( S = \{ L_1, L_2, \ldots, L_n \} \subseteq C \) in a formula \( F \). Assume that \( F \setminus \{ C \} \) is satisfiable. Let \( F' \) and \( F_C \) be finite sets of ground instances of clauses in \( F \setminus \{ C \} \) and \( \{ C \} \). Since \( F \setminus \{ C \} \) is satisfiable, there exists a propositional assignment \( \alpha \) satisfies \( F' \). Assume that it falsifies some ground instances \( \{ C_1, C_2, \ldots, C_4 \} \) of \( C \) which are contained in \( F_C \). Flipping all the truth value of ground literals of ground instance \( S_i \) of \( S \) in \( C_{i}(1 \leq i \leq k) \), has no influence on ground instances in \( F' \) according to Lemma 1, but it may falsify some other ground instances of \( F_C \). Nevertheless, since \( F_C \) is finite, we can satisfy those ground instances of \( F_C \) by flipping orderly the truth value of ground literals of the ground instances of \( S \) in those falsified ground instances of \( C \). Eventually, all the ground
instances of $F_C$ are satisfied. According to Theorem 1, $F$ is satisfiable, which means $F$ and $F \setminus \{C\}$ are satisfiability equivalent. Therefore, the clause $C$ is redundant w.r.t. $F$. 

Apparently the redundancy property of set-blocked clause is local. A clause can be assessed whether it is an SBC, by only considering its resolution environment rather than the whole formula. If a clause $C$ is an SBC in a formula $F$ and the clause $C$ has the same resolution environment in another formula $F'$, then the clause $C$ is also an SBC in $F'$.

Next, we introduce extended set-blocked clause (E-SBC), a generalization of SBC. With respect to E-SBC, another factor, external ground atoms, is added into consideration. external ground atoms of a clause $C$ are ground atoms in the resolution environment of $C$ but not occurring in $C$ and not able to be unified with atoms in $\text{atom}\{C\}$, which means variations of truth values of external ground axioms and truth values of $C$ are not relevant. In other words, if a clause $C'$ is true because of the truth values of several external ground atoms of $C$ in $C'$, then the clause $C'$ will not be falsified under any change of the truth values of literals in $C$.

**Definition 7.** A clause $C$ is an extended set-blocked clause (E-SBC) in a formula $F$ if, for every assignment $\beta$ over the external ground atoms $\text{ext}G_F(C)$, there exists a subset $S_\beta = \{L_1, L_2, \ldots, L_{|\beta|}\} \subseteq C$ such that $C$ is set-blocked upon $S_\beta = \{L_1, L_2, \ldots, L_{|\beta|}\} \subseteq C$ in $F|\beta$.

**Example 4.** Suppose $C = P(x) \lor Q(x)$ and the resolution environment of $C$: $\text{env}_F(C) = \{\neg P(a) \lor Q(a) \lor T(z) \lor R(a), \neg Q(b) \lor \neg R(a), \neg Q(f(y)) \lor P(f(y))\}$. Apparently, the clause $C$ is not a set-blocked clause, because not all the $L_i^2$-resolvents ($1 \leq i \leq n$) are tautologies no matter which subset of $C$ has been chosen to be. The external ground atoms of $C$ is $\text{ext}G_F(C) = \{R(a)\}$. If an assignment $\alpha$ satisfies $\alpha(R(a)) = 1$ and $\alpha(\neg R(a)) = 0$, then $\alpha(\neg P(a) \lor Q(a) \lor T(z) \lor R(a)) = 1$ and $\neg P(a) \lor Q(a) \lor T(z) \lor R(a)$ will not be included in $F|\alpha$ according to the definition of $F|\alpha$. Therefore, we only need to consider $\neg Q(b) \lor \neg R(a)$ and $\neg Q(f(y)) \lor P(f(y))$ as the resolution environment of $C$ in $F|\alpha$. If we choose $S = \{P(x)\}$, $C$ cannot be resolved with either $\neg Q(b) \lor \neg R(a)$ or $\neg Q(f(y)) \lor P(f(y))$ upon $S$, clause $C$ is trivially set-blocked upon $S = \{P(x)\}$ in $F|\alpha$. If another assignment $\lambda$ satisfies $\lambda(R(a)) = 0$ and $\lambda(\neg R(a)) = 1$, then $\lambda(\neg Q(b) \lor \neg R(a)) = 1$ and it is not covered in $F|\lambda$. We only need to consider $\neg P(a) \lor Q(a) \lor T(z) \lor R(a)$ and $\neg Q(f(y)) \lor P(f(y))$ as the resolution environment of $C$ in $F|\lambda$. If we choose $S = \{P(x), Q(x)\}$, the $P(x)^{\{P(x), Q(x)\}}$-resolvents $Q(a) \lor T(x) \lor R(a) \lor \neg Q(a)$ and $Q(x)^{\{P(x), Q(x)\}}$-resolvent $\neg P(f(y)) \lor P(f(y))$ both are tautologies by resolving $C$ with $\neg P(a) \lor Q(a) \lor T(z) \lor R(a)$ and $\neg Q(f(y)) \lor P(f(y))$. Therefore, clause $C$ is set-blocked upon $S = \{P(x), Q(x)\}$ in $F|\lambda$. Because $\alpha$ and $\lambda$ cover all the possible assignments over $R(a)$, then we can conclude that $C$ is set-blocked in $F|\beta$ for any assignment $\beta$ over the external ground atoms $\text{ext}G_F(C)$. Therefore, the clause $C$ is an E-SBC.

Evaluating whether a clause $C$ is an E-SBC when there are no external ground atoms, it only needs to evaluate if the clause $C$ is an SBC. No external ground atoms means no way to influence the resolution environment of $C$, therefore, $F|\beta$ is equal to $F$ all the time for any assignment $\beta$.

**Lemma 2.** Given a clause $C$ is an E-SBC in a formula $F$. Let $\alpha$ be an assignment that propositionally satisfies all the ground instances of clauses in $F \setminus \{C\}$ but falsifies a ground instance $C\lambda$ of $C$. Then, there exist a subset $S\lambda = \{L_1\lambda, L_2\lambda, \ldots, L_n\lambda\}$ of $C\lambda$ such that the assignment $\alpha'$, obtained from $\alpha$ by flipping all the truth values of ground literals $L_1\lambda, L_2\lambda, \ldots, L_n\lambda$ in $S\lambda$, still satisfies all the ground instances in $F \setminus \{C\}$.

**Proof.** Since flipping the truth values of ground literals of $C$ can only affect the truth values of ground clauses in the resolution environment of $C$, we only consider the ground instances in the resolution environment of $C$ whether they will be falsified by flipping the truth values of ground literals in $C$. It is analyzed in two cases:

**Case 1:** In the resolution environment $\text{env}_F(C)$ of $C$, there are no external ground atoms. Since $C$ is an E-SBC and there is no ground atoms in the resolution environment of $C$, it means that $C$ is
set-blocked in the formula \( F \). Now that \( C \) is set-blocked in the formula \( F \), there exists a subset \( S\lambda = \{L_1\lambda, L_2\lambda, \ldots, L_n\lambda\} \) of \( \mathcal{C}\lambda \) such that the assignment \( \lambda' \), obtained from \( \lambda \) by flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( S\lambda \), still satisfies all the ground instances in \( F \setminus \{C\} \) according to Lemma 1.

**Case 2:** In the resolution environment \( env_F(C) \) of \( C \), there exist external ground atoms \( \{A_1, A_2, \ldots, A_m\} \). Since \( \alpha \) is an assignment which covers all the ground instances of clauses in \( F \setminus \{C\} \), it also assigns the truth values of those external ground atoms \( \{A_1, A_2, \ldots, A_m\} \) to be true or false. Assume that the assignment \( \alpha \) to the external ground atoms \( \{A_1, A_2, \ldots, A_m\} \) make the truth values of clauses \( \{C_1, C_2, \ldots, C_k\} \) in the resolution environment of \( C \) true. Since clause \( C \) is extended set-blocked in a formula \( F \), there exist a subset \( S = \{L_1, L_2, \ldots, L_n\} \) of \( C \), \( C \) is set-blocked upon \( S \) in \( F[\alpha] \), which means \( C \) is set-blocked upon \( S \) in the formula \( F \setminus \{C_1, C_2, \ldots, C_k\} \). Now that \( C \) is set-blocked upon \( S \) in the formula \( F \setminus \{C_1, C_2, \ldots, C_k\} \), flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( S\lambda \) will not falsify any ground instances of \( F \setminus \{C, C_1, C_2, \ldots, C_k\} \). Furthermore, \( \alpha \) already satisfies all the ground instances of \( \{C_1, C_2, \ldots, C_k\} \) by its assignment to those external ground atoms \( \{A_1, A_2, \ldots, A_m\} \) according to the assumption, flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( S\lambda \) will not falsify any ground instances of \( F \setminus \{C, C_1, C_2, \ldots, C_k\} \). Hence, flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( S\lambda \) will not falsify any ground instances of \( F \setminus \{C\} \). Therefore, the assignment \( \lambda' \), obtained from \( \lambda \) by flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( S\lambda \), still satisfies all the ground instances in \( F \setminus \{C\} \).

**Theorem 4.** If a clause \( C \) is an E-SBC in a formula \( F \), it is redundant w.r.t. \( F \).

**Proof.** Given finite ground instances \( F' \) and \( F_C \) are sets of finite ground instances of \( F \setminus \{C\} \) and \( \{C\} \). Assume that \( F \setminus \{C\} \) is satisfiable, then there exists a propositional assignment \( \alpha \) satisfies all the ground instances of \( F' \), but it may falsify some ground instances in \( F_C \). According to Lemma 2, there exists a subset \( S = \{L_1, L_2, \ldots, L_n\} \) of \( C \), it will not affect the truth values of ground instances in \( F' \) by flipping the truth values of ground literals of \( S \) in those falsified ground instances of \( C \). Even though flipping those truth values of ground literals in those falsified ground instances of \( C \) may falsify other ground instances of \( C \) in \( F_C \); however, we can flip the truth values of ground literals in ground instances of \( S \) in those falsified ground instances successively, until all the truth values of falsified ground instances become true. According to Theorem 1, \( F \) is satisfiable. Therefore, the clause \( C \) is redundant w.r.t. \( F \).

4. Equality-Set-Blocked Clause and Extended Equality-Blocked Clause in First-Order Logic Formulas with Equality

In the last section, SBC and E-SBC in formulas without equality are discussed. In fact, the conceptions of SBC and E-SBC can only be adopted in first-order formulas without equality, if they are utilized in first-order formulas with equality, some clauses will be removed mistakenly. A counter-example is given as follows:

**Example 5.** Let \( C = R(a) \lor P(a) \) and the formula \( F = \{R(a) \lor P(a), \neg R(b), \neg P(b), a = b\} \). According to the definition of SBC in Section 3, clause \( C \) is trivially set-blocked upon \( \{R(a), P(a)\} \), because there is no clauses in \( F \setminus \{C\} \) can be directly resolved with clause \( C \) upon \( \{R(a), P(a)\} \), which means clause \( C \) can be removed from formula \( F \) without influencing the satisfiability or unsatisfiability. However, \( F \) is unsatisfiable while \( F \setminus \{C\} \) is satisfiable.

The reason why this situation happens is because the definitions of SBC and E-SBC are not involved with equality. For example, the truth values of \( R(a) \) and \( R(b) \) are always the same even though their forms are diverse in Example 5, nevertheless, this situation is not considered in the definitions of SBC and E-SBC. But if we combine the definitions of SBC and E-SBC in Section 3 with flattening and flat resolution, this problem can be solved.
The definition of flat \( L_d^1 \)-resolvent of clause \( C \) in first-order logic with equality is given below. It deletes a subset of substituted literals from literals in clause \( C \) but adds the subset of substituted literals from the negations of literals in clause \( C \) compared with flat \( L \)-resolvent of clause \( C \).

**Definition 8.** Given two clauses, \( C = L_1 \lor L_2 \lor \ldots \lor L_n \lor C' \) with \( S = \{L_1, L_2, \ldots, L_n\} \) such that \( L_k (1 \leq k \leq n) \) does not contain the predicate symbol \( = \) and \( L_i (1 \leq i \leq n) \) and \( L_j (1 \leq j \leq n) \) do not have the same predicate symbol but the opposite polarity, and \( D = N_1 \lor N_2 \lor \ldots \lor N_m \lor D' \).

If a literal \( L_i \in S(1 \leq i \leq n) \) and \( \neg N_k \) have the same predicate symbol and polarity, by flattening \( L_i, N_1, \ldots, N_m \), new literals \( L'_1, L'_2, \ldots, L'_m \) can be obtained from \( L_i, N_1, \ldots, N_m \) and new flattened clauses \( C' \) and \( D' \) can be obtained from \( C \) and \( D \). If the mgu of \( L'_1, L'_2, \ldots, L'_m \) is \( \sigma \), the clause \( (C' \setminus L'_1)\sigma \cup (D' \setminus L'_1, \ldots, L'_n)\sigma \cup (\{N_1, \ldots, N_m\}\sigma \setminus \{L_i\})\sigma \) is called the flat \( L_d^1 \)-resolvent of \( C \) and \( D \).

**Example 6.** Let \( C = P(b) \lor Q(b) \) and \( D = \neg Q(c) \lor P(b) \).

After flattening \( Q(b) \) and \( \neg Q(c) \), two new flattened clauses \( C' = P(b) \lor Q(x) \lor \neg P(z) \lor Q(z) \) and \( D' = y = c \lor \neg Q(y) \lor P(b) \).

Then, the flat \( Q(b)[P(b),Q(b)] \)-resolvent of \( C \) and \( D \) is \( x \neq b \lor x \neq c \lor P(b) \lor \neg P(b) \) via substitution \( \delta = \{y \mapsto x\} \).

Compared with equality-blocked clause, equality-set-blocked clause is obtained by generalizing one blocking literal in equality-blocked clause to multiple blocking literals.

**Definition 9.** For a clause \( C \) in a formula \( F \), if there exists a subset \( S = \{L_1, L_2, \ldots, L_n\} \subseteq C \) such that \( L_k (1 \leq k \leq n) \) does not contain the predicate symbol \( = \) and \( L_i (1 \leq i \leq n) \) and \( L_j (1 \leq j \leq n) \) do not have the same predicate symbol but the opposite polarity. If all the flat \( L_d^1 \)-resolvents \( 1 \leq i \leq n \) of \( C \) are valid, then the clause \( C \) is called as an equality-set-blocked clause (ESBC) upon \( S \) in the formula \( F \).

**Example 7.** Let \( C = P(b) \lor Q(b), S = \{P(b), Q(b)\} \), and the formula \( F = \{P(b) \lor Q(b), \neg Q(c) \lor P(b), \neg P(z) \lor Q(z)\} \).

About \( Q(b) \), there is only the clause \( \neg Q(c) \lor P(b) \) in \( F \setminus \{C\} \) contains the literal \( \neg Q(c) \) has the same predicate symbol and the opposite polarity with the literal \( Q(b) \) in \( S \). Hence, the only flat \( Q(b)[P(b),Q(b)] \)-resolvent of \( C \) is \( x \neq b \lor x \neq c \lor P(b) \lor \neg P(b) \) which is valid. In addition, with respect to \( P(b) \), there is only the clause \( \neg P(z) \lor Q(z) \) in \( F \setminus \{C\} \) contains the literal \( \neg P(z) \) has the same predicate symbol and the opposite polarity with the literal \( P(b) \) in the set \( S \), then the only one flat \( P(b)[P(b),Q(b)] \)-resolvent is \( y \neq b \lor Q(y) \lor \neg Q(b) \) and it is valid. According to Definition 9, the clause \( C \) is ESBC upon \( S = \{P(b), Q(b)\} \) in the formula \( F \).

ESBC is also redundant in first-order formulas with equality. Before proving its redundancy, a Lemma is introduced first.

**Lemma 3.** Suppose a clause \( C \) is an ESBC upon \( S = \{L_1, \ldots, L_m\} \subseteq C \) in a formula \( F \). \( \beta \) is a propositional assignment satisfying all the ground instances of equality axioms and all the ground instances of clauses in \( F \setminus \{C\} \), but falsifies a ground instance \( CA \) of \( C \). \( \beta' \) is a propositional assignment obtained from \( \beta \) by equivalence flipping all the truth values of ground literals \( L_1, \ldots, L_m \) in the set \( S \), then \( \beta' \) satisfies all the ground instances of clauses in \( F \setminus \{C\} \) and all the ground instances of equality axioms.

**Proof.** Let the clause \( C = L_1(t_{11}, \ldots, t_{1k_1}) \lor \ldots \lor L_m(t_{m1}, \ldots, t_{mk_m}) \lor C' \) and \( S = \{L_1(t_{11}, \ldots, t_{1k_1}), \ldots, L_m(t_{m1}, \ldots, t_{mk_m})\} \).

Assume that equivalence flipping the truth values of ground literals \( L_1(t_{11}, \ldots, t_{1k_1}) \ldots, L_m(t_{m1}, \ldots, t_{mk_m}) \) in the set \( S \) falsifies the truth value of a ground instance \( D \gamma \) in \( F \setminus \{C\} \), the form of the clause must be \( D \) with at least one literal \( \neg L_i(s_{i1}, \ldots, s_{ik}) \) \((1 \leq i \leq m)\) in it and it has \( \beta(t_{ij} \lambda = s_{ij} \gamma) = 1 \) for all \( 1 \leq j \leq k_j \). Without loss of generality, we assume that the clause \( D \) with the literal \( \neg L_i \).

Let the clause \( D \) be a clause with literals \( \neg L_1(s_{11}, \ldots, s_{1k_1}), \ldots, \neg L_i(p_{1j}, \ldots, p_{ik}) \) such that \( \beta(t_{ij} \lambda = s_{ij} \gamma) = 1, \ldots, \beta(t_{ij} \lambda = p_{ij} \gamma) = 1 \) for \( 1 \leq j \leq k_j \). To simplify the presentation and
without loss of generality, we assume that \(\neg L_1(s_{11}, \ldots, s_{1k_1})\) and \(\neg L_1(p_{11}, \ldots, p_{1k_1})\) are all such literals in the clause \(D\), therefore \(D = \neg L_1(s_{11}, \ldots, s_{1k_1}) \lor \neg L_1(p_{11}, \ldots, p_{1k_1}) \lor D'.\)

Since \(C\) is equality-set-blocked upon \(S\), then all the flat \(L_1(t_{11}, \ldots, t_{1k_1})\) \(S\)-resolvents are valid. Therefore, the flat \(L_1(t_{11}, \ldots, t_{1k_1})\) \(S\)-resolvent between \(C\) and \(D\):

\[
R = (C' \lor D' \lor (\neg L_1(t_{11}, \ldots, t_{1k_1})) \lor (\forall 1 \leq j \leq k_1 x_{ij} \neq t_{ij} \lor y_{ij} \neq s_{ij} \lor z_{ij} \neq p_{ij})\sigma
\]

is valid, where \(\sigma\) is an mgu of the literals \(L_1(x_{11}, \ldots, x_{1k_1}), L_1(y_{11}, \ldots, y_{1k_1})\) and \(L_1(z_{11}, \ldots, z_{1k_1})\), obtained by flattening \(L_1(t_{11}, \ldots, t_{1k_1}), L_1(s_{11}, \ldots, s_{1k_1})\) and \(L_1(p_{11}, \ldots, p_{1k_1})\). Without loss of generality, we assume that \(\sigma = \{y_{ij} \mapsto x_{ij}, z_{ij} \mapsto x_{ij} | 1 \leq j \leq k_1\}\), then the flat \(L_1(t_{11}, \ldots, t_{1k_1})\) \(S\)-resolvent can be written as:

\[
R = C' \lor D' \lor (\neg L_1(t_{11}, \ldots, t_{1k_1})) \lor (\forall 1 \leq j \leq k_1 x_{ij} \neq t_{ij} \lor x_{ij} \neq s_{ij} \lor x_{ij} \neq p_{ij})
\]

Because \(R\) is valid, then the assignment \(\beta\) must satisfy all the ground instances of \(R\). A ground instance \(R\delta\) of \(R\) can be created by implementing the following substitution:

\[
\delta(x) = \begin{cases} 
    t_{ij}\lambda & \text{if } x = x_{11}, \ldots, x_{1k_1} \\
    x\lambda & \text{if } x \in C' \\
    x\gamma & \text{if } x \in D' \\
    x\lambda & \text{if } x \in (\bar{S}\backslash L_1(t_{11}, \ldots, t_{1k_1})) 
\end{cases}
\]

Then \(R\delta\) can be written as follows:

\[
R\delta = C' \lor D' \lor (\neg L_1(t_{11}, \ldots, t_{1k_1})) \lor (\forall 1 \leq j \leq k_1 t_{ij}\lambda \neq t_{ij}\lambda \lor t_{ij}\lambda \neq s_{ij}\gamma \lor t_{ij}\lambda \neq p_{ij}\gamma)
\]

which must be satisfied by \(\beta'\) because it is valid. In \(R\delta\), all the \(t_{ij}\lambda \neq t_{ij}\lambda\), all the \(t_{ij}\lambda \neq s_{ij}\gamma\), all the \(t_{ij}\lambda \neq p_{ij}\gamma\) (according to the assumption) and \(C'\lambda\) are falsified by \(\beta\), we can conclude that all the \(t_{ij}\lambda \neq t_{ij}\lambda\), all the \(t_{ij}\lambda \neq s_{ij}\gamma\), all the \(t_{ij}\lambda \neq p_{ij}\gamma\) and \(C'\lambda\) are also falsified by \(\beta'\). In addition, because \(\beta\) satisfies \((\bar{S}\backslash L_1(t_{11}, \ldots, t_{1k_1}))\lambda\) and \(\beta'\) is obtained by equivalence flipping the truth values of \((\bar{S}\backslash L_1(t_{11}, \ldots, t_{1k_1}))\lambda\) from \(\beta\), \(\beta'\) falsifies \((\bar{S}\backslash L_1(t_{11}, \ldots, t_{1k_1}))\lambda\). As a result, \(D'\gamma\lambda\) must contain at least one literal which is satisfied by \(\beta'\) due to \(\beta'(R\delta) = 1\). Hence, \(D'\gamma\lambda\) is satisfied by \(\beta'\). Therefore, \(\beta'\) satisfies not only \(C\lambda\) but also all the ground instances of clauses in \(F\backslash\{C\}\) and all the ground instances of equality axioms. □

**Theorem 5.** If a clause \(C\) is an ESBC upon \(S = \{L_1, \ldots, L_m\} \subseteq C\) in a formula \(F\), it is redundant w.r.t. \(F\).

**Proof.** Given there are some finite ground instances \(F_C, F'\) and \(F_E\) of \(C \cup F\backslash\{C\} \cup \varepsilon_L\) and assume \(\beta\) is a propositional assignment which satisfies \(F'\) and \(F_E\). Suppose the assignment \(\beta\) falsifies some ground instances in \(F_C\), those ground instances can be satisfied by equivalence flipping the truth values of ground literals of ground instances of \(S\) in those ground instances without influencing the satisfiability of \(F'\) and \(F_E\) according to Lemma 3. Even though it may falsify other ground instances in \(F_C\) by equivalence flipping, those ground instances can be satisfied by equivalence flipping the truth values of ground literals of ground instances of \(S\) in those falsified ground instances of \(C\) in \(F_C\) without affecting the satisfiability of the previous ground instances. Because those ground instances are finite in \(F_C\), all the ground instances in \(F_C\) can be satisfied by orderly equivalence flipping the truth values of ground literals of ground instances of \(S\) in those ground instances which have been falsified. Therefore, for any satisfying propositional assignment \(\beta\) of ground instances \(F'\) and \(F_E\) of \(F\backslash\{C\} \cup \varepsilon_L\), there exists a satisfying propositional assignment \(\beta'\), obtained by orderly equivalence flipping the truth values of ground literals of ground instances of \(S\) in those falsified ground instances of \(C\) from \(\beta\).
Hence, \( F \cup \varepsilon_L \) is satisfiable according to Theorem 2. Therefore, \( F \cup \varepsilon_L \) and \( F \setminus \{C\} \cup \varepsilon_L \) are satisfiable equivalent, then the clause \( C \) is redundant w.r.t. \( F \). \( \square \)

Similarly, there is a generalization of ESBC in first-order logic with equality, by adding a new factor, the assignments over the external ground atoms.

**Definition 10.** A clause \( C \) is an extended equality-set-blocked clause (E-ESBC) in a formula \( F \) if, for every assignment \( \beta \) over the external ground atoms \( \text{ext}G_F(C) \), there exists a subset \( S_\beta = \{L_1, L_2, \ldots, L_{n_\beta}\} \subseteq C \), \( C \) is equality-set-blocked upon \( S_\beta = \{L_1, L_2, \ldots, L_{n_\beta}\} \subseteq C \) in \( F|\beta \).

**Example 8.** Suppose \( C = P(a) \lor Q(a) \) and the resolution environment of \( C \): \( \text{env}_{F_\lambda}(C) = \{\neg P(x) \lor x = a \lor R(a), Q(a) \lor \neg P(y), \neg P(a) \lor \neg R(a)\} \). We can see that external ground atoms \( \text{ext}G_F(C) = \{R(a)\} \).

If an assignment \( \alpha \) satisfies \( \lambda(R(a)) = 1 \) and \( \neg \lambda(R(a)) = 0 \), then \( \lambda(\neg P(x) \lor x \neq a \lor R(a)) = 1 \). Therefore, \( Q(a) \lor \neg P(y) \) and \( \neg P(a) \lor \neg R(a) \) are the clauses in the resolution environment in \( C \) in \( F|\alpha \).

Furthermore, \( L \) ground literals \( \{\lambda\} \) is equality-set-blocked according to Lemma 3. 

**Lemma 4.** Suppose a clause \( C \) is an E-ESBC in a formula \( F \). \( \beta \) is a propositional assignment satisfying all the ground instances of equality axioms and all the ground instances of clauses in \( F \setminus \{C\} \), but falsifies a ground instance \( CA \) of \( C \). There exist a subset \( S = \{L_1, \ldots, L_n\} \) such that \( \beta' \), obtained from \( \beta \) by equivalence flipping all the truth values of ground literals \( L_1\lambda, \ldots, L_n\lambda \) in the set \( SL \), satisfies all the ground instances of clauses in \( F \setminus \{C\} \) and all the ground instances of equality axioms.

**Proof.** We only consider whether the ground instances of clauses will be falsified in the resolution environment of \( C \). It is analyzed in two cases.

**Case 1:** In the resolution environment \( \text{env}_{F_\lambda}(C) \) of \( C \), there is no external ground atoms. Since there are no external ground atoms in the resolution environment of \( C \) and \( C \) is an extended equality-set-blocked clause in the formula \( F \), it means that \( C \) is equality-set-blocked in the formula \( F \).

In addition, because \( C \) is equality-set-blocked in the formula \( F \), there must exist a subset \( SL = \{L_1\lambda, L_2\lambda, \ldots, L_n\lambda\} \) of \( CA \) and the assignment \( \beta' \), obtained from \( \beta \) by flipping all the truth values of ground literals \( L_1\lambda, \ldots, L_n\lambda \) in \( SL \), still satisfies all the ground instances in \( F \setminus \{C\} \) and all the ground instances of equality axioms according to Lemma 3.

**Case 2:** In the resolution environment \( \text{env}_{F_\lambda}(C) \) of \( C \), there exist external ground atoms \( \{A_1, A_2, \ldots, A_m\} \). Since \( \beta \) is an assignment which covers all the ground instances of clauses in \( F \setminus \{C\} \), it also assigns the truth values of those external ground atoms \( \{A_1, A_2, \ldots, A_m\} \). Assume that the assignment to the external ground atoms make the truth values of clauses \( \{C_1, C_2, \ldots, C_k\} \) in the resolution environment of \( C \) are true. Therefore, there exist a subset \( S = \{L_1, L_2, \ldots, L_n\} \) of \( C \), \( C \) is equality-set-blocked upon \( S \) in the formula \( F \setminus \{C_1, C_2, \ldots, C_k\} \). Now that \( C \) is equality-set-blocked upon \( S \) in the formula \( F \setminus \{C_1, C_2, \ldots, C_k\} \), flipping all the truth values of ground literals \( L_1\lambda, \ldots, L_n\lambda \) in \( SL \) will not falsify any ground instances of \( F \setminus \{C, C_1, C_2, \ldots, C_k\} \).

Furthermore, \( \lambda \) satisfies all the ground instances of \( \{C_1, C_2, \ldots, C_k\} \) by its assignment to those external ground atoms \( \{A_1, A_2, \ldots, A_m\} \), as a result, equivalence flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( SL \) will not falsify any ground instances of \( \{C, C_1, C_2, \ldots, C_k\} \). Hence, equivalence flipping all the truth values of ground literals \( L_1\lambda, L_2\lambda, \ldots, L_n\lambda \) in \( SL \) will not falsify any ground instances of \( F \setminus \{C\} \). Therefore, the assignment \( \beta' \), obtained from \( \beta \) by flipping all the truth values
of ground literals $L_1 \lambda, L_2 \lambda, \ldots, L_n \lambda$ in $S \lambda$, still satisfies all the ground instances in $F \setminus \{C\}$ and all the ground axioms of equality axioms. □

**Theorem 6.** If a clause $C$ is an E-ESBC in a formula $F$, it is redundant w.r.t. $F$.

**Proof.** For every finite ground instances of $F \setminus \{C\} \cup \{C\} \cup \varepsilon_L$ and any assignment $\beta$ propositionally satisfies finite ground instances of $F \setminus \{C\} \cup \varepsilon_L$, there exists a subset $S \subseteq C$, equivalence flipping the truth values of ground literals of the subset $S$ in some ground instances of $C$ will not affect the truth values of those finite ground instances in $F \setminus \{C\} \cup \varepsilon_L$ according to Lemma 4. Besides, since the ground instances of $C$ are finite, those falsified ground instances of $C$ can become true by equivalence flipping the ground literals in ground instances of $S$ in those falsified ground instances of $C$ successively. Therefore, $F \setminus \{C\} \cup \{C\} \cup \varepsilon_L$ is satisfiable according to Theorem 2. Hence, the clause $C$ is redundant w.r.t. $F$. □

5. Effectiveness and Confluence Property

In this section, we evaluate effectiveness and confluence properties of the four corresponding clause elimination methods set-blocked clause elimination, extended set-blocked clause elimination, equality-set-blocked clause elimination and extended equality-set-blocked clause elimination.

5.1. Comparison of Effectiveness

Effectiveness is a significant evaluation standard for a clause elimination method. It reflects the capability of clause elimination methods to simplify formulas. The more effective a kind of clause elimination method is, the more clauses can be removed from the formulas. Below is the definition of effectiveness [2].

**Definition 11.** Given two clause elimination methods $CE_1$ and $CE_2$, and a CNF formula $F$. After the clause elimination method $CE_1$ or $CE_2$ is implemented, the new formula is $CE_1(F)$ or $CE_2(F)$. If $CE_1(F) \subseteq CE_2(F)$, it is called that $CE_1$ is at least as effective as $CE_2$. In addition, if $CE_1(F) \subseteq CE_2(F)$ exists and there exists a formula $F'$, $CE_1(F') \subset CE_2(F')$, then $CE_1$ is more effective than $CE_2$.

First, we compare the effectiveness between blocked clause elimination (BCE) and set-blocked clause elimination (SBCE).

**Theorem 7.** SBCE is more effective than BCE.

**Proof.** If a clause $C$ is a BC upon $L \in C$ in a formula $F$, it must be a SBC upon $S = \{L\} \subseteq C$. Assume that there is a clause $D = N_1 \lor N_2 \lor \ldots \lor N_m \lor D'$ in the formula and $L, \neg N_1, \ldots, \neg N_m$ can be unified by an mgu $\sigma$, then $L^S$-resolvent of $C$ and $D$ is $C'\sigma \cup D'\sigma \cup (S \setminus \{\neg L\})\sigma = C'\sigma \cup D'\sigma$, which is the same as the $L$-resolvent $C' \cup D'\sigma$ of $C$ and $D$. Therefore, $L^S$-resolvent of $C$ and $D$ is a tautology. This situation can generalize to all the $L^S$-resolvent of $C$, so all the $L^S$-resolvent of $C$ are tautologies. Hence, the clause $C$ is an SBC upon $S = \{L\}$. On the contrary, the situation is not the same vice versa. For example, the clause $C$ is an SBC but it is not a BC in Example 2. Therefore, we can conclude that SBCE is more effective than BCE. □

Next, effectiveness between extended set-blocked clause elimination (E-SBCE) and set-blocked clause (SBCE) is compared.

**Theorem 8.** E-SBCE is more effective than SBCE.

**Proof.** If a clause $C$ is a set-blocked clause upon $S = \{L_1, L_2, \ldots, L_n\} \subseteq C$ in a formula $F$, it must be an extended set-blocked clause. It is analyzed by two cases.
**Case 1:** There is no external ground atoms of the clause $C$. When there are no external ground atoms, the clause $C$ is trivially an E-SBC when it is an SBC.

**Case 2:** Some external ground atoms $\{A_1, A_2, \ldots, A_m\}$ exist in the resolution environment of $C$. For any assignment $\alpha$ over $\{A_1, A_2, \ldots, A_m\}$, it may assign some clauses $\{C_1, C_2, \ldots, C_k\}$ in $env_F(C)$ as true and it only needs to consider $env_F(C) \setminus \{C_1, C_2, \ldots, C_k\}$ as the resolution environment of $C$ in $F|\alpha$. Since $C$ is a SBC upon $S \subseteq C$, all the $L^n$-resolvents ($1 \leq i \leq n$) obtained by resolving $C$ with clauses in $env_F(C)$ are tautologies, then there is no doubt that all the $L^n$-resolvents ($1 \leq i \leq n$) obtained by resolving $C$ with clauses in $env_F(C) \setminus \{C_1, C_2, \ldots, C_k\}$ are tautologies. Therefore, $C$ is an SBC upon $S$ in $F|\alpha$. As a result, $C$ is an E-SBC.

However, if a clause $C$ is an E-SBC, it may not be an SBC. For example, the clause $C$ is an E-SBC but it is not an SBC in Example 4.

Since effectiveness has the property of transitivity, then E-SBCE is more effective than BCE. Similarly, situation is analogous among those clause elimination methods dealing with first-order formulas with equality.

**Theorem 9.** Equality-set-blocked clause elimination (ESBCE) is more effective than equality-blocked clause elimination (EBCE).

**Proof.** If a clause is an EBC upon $L \in C$ in a formula $F$, it must be an ESBC upon $S = \{L\} \subseteq C$ in the formula $F$. The reason is because all the flat $L^n$-resolvents of $C$ are the same as all the flat $L$-resolvents of $C$, then it is no doubt all the flat $L^n$-resolvents of $C$ are also valid. Therefore, the clause $C$ is also an ESBC. However, if a clause is an EBC, it may not be an EBC. For example, in Example 7, the clause $C$ is an E-SBC but it is not an EBC. Hence, EBC is more effective than EBECE. 

**Theorem 10.** Extended equality-set-blocked clause elimination (E-ESBCE) is more effective than equality-set-blocked clause elimination (ESBCE).

**Proof.** If a clause $C$ is an ESBC upon $S = \{L_1, L_2, \ldots, L_n\} \subseteq C$ in a formula $F$, then for all possible assignments over the external ground atoms of $C$, $C$ must be an ESBCE upon $S = \{L_1, L_2, \ldots, L_n\} \subseteq C$. Therefore, we can conclude that $C$ is also an E-ESBCE. Nevertheless, if a clause is an E-ESBCE, it may not be an E-ESBCE. For example, in Example 8, the clause $C$ is an E-ESBCE but it is not an E-ESBCE.

Since effectiveness has the property of transitivity, then E-SBCE is more effective than BCE. Similarly, situation is analogous among those clause elimination methods dealing with first-order formulas with equality.

**Figure 1.** Effectiveness among those clause elimination methods. A arrow from A to B means A is more effective than B.
5.2. Confluence Property

Confluence property is also a crucial evaluation standard for a clause elimination method. It illuminates whether the difference of the sequence of eliminating clauses in a formula will cause the final obtained new formula different. If a clause elimination method has the confluence property, the final obtained new formula is the same no matter what the elimination sequence is. However, the final obtained formula will vary according to the distinction of elimination sequence if the clause elimination method has no confluence property, as a result, making a good strategy for elimination sequence determines how much clauses can be removed. In this subsection, we discuss the four clause elimination methods’ confluence properties. Below is the definition of diamond property [12]:

Definition 12. If a relation R has the diamond property, for \( \forall x, y, z \) with \( xRy \) and \( xRz \), there exists a \( v \) with \( yRv \) and \( zRv \).

If a relation has the diamond property, it also has the confluence property [12]. Clause elimination methods can be seen as relations, which is between original formulas and new formulas after the elimination. Next, we prove all the four clause elimination methods have the confluence properties.

Theorem 11. SBCE is confluent.

Proof. Assume that there are two SBCs \( C_1 \) and \( C_2 \) in a formula \( F \). Here we prove the other clause still can be removed no matter which clause will be removed first. Without loss of generality, assume that \( C_1 \) is removed earlier and \( C_2 \) is not an SBC in \( F \setminus \{C_1\} \) after that. Since \( C_2 \) is an SBC in the formula \( F \), there exists \( S = \{L_1, L_2, \ldots, L_n\} \subseteq C_2 \) such that all the \( L_i^5 \)-resolvents \( (1 \leq i \leq n) \) obtained by resolving \( C_2 \) with clauses in \( F \setminus \{C_2\} \) are tautologies. Now that \( C_2 \) is not an SBC after removing the clause \( C_1 \), then there exists at least one \( L_i^5 \)-resolvents \( (1 \leq j \leq n) \) acquired by resolving \( C_2 \) with clauses in \( F \setminus \{C_1, C_2\} \) is not a tautology which is contradictory with the fact that \( C_2 \) is an SBC in the formula \( F \). Therefore, \( C_2 \) is still a SBC after removing the clause \( C_1 \). As a result, we can conclude that SBCE is confluent. \( \square \)

Theorem 12. E-SBCE is confluent.

Proof. Here we prove E-SBCE is confluent by proving a clause \( C \) is still an E-SBC in a subset \( F' \) of the formula \( F \) if it is an E-SBC in the formula \( F \). Assume that the clause \( C \) is not an E-SBC in the subset \( F' \), then there exists at least one assignment \( \alpha \) over the external ground atoms of \( C \), satisfying that there exists a subset \( S_\alpha = \{L_1, L_2, \ldots, L_n\} \subseteq C \) such that \( C \) is not a set-blocked clause upon \( S_\alpha \) in \( F'\vert\alpha \), but \( C \) is a set-blocked clause upon \( S_\alpha \) in \( F\vert\alpha \). Then there is at least one \( L_i^5 \)-resolvents \( (1 \leq i \leq n) \) is not a tautology by resolving \( C \) with the clauses in \( F'\vert\alpha \setminus \{C\} \), which is a contradiction against the fact \( C \) is a set-blocked clause upon \( S_\alpha \) in \( F\vert\alpha \). E-SBCE is confluent. \( \square \)

Theorem 13. ESBCE is confluent.

Proof. Assume that there are two ESBCs \( C_1 \) and \( C_2 \) in a formula \( F \). We prove that the order of eliminating clauses has no effect on the redundancy of ESBCs. Without loss of generality, we assume that the clause \( C_1 \) is removed first and \( C_2 \) is not an ESBC in \( F \setminus \{C_1\} \) after eliminating \( C_1 \). Since \( C_2 \) is not an ESBC after removing \( C_1 \), then for arbitrary subset \( S = \{L_1, L_2, \ldots, L_n\} \subseteq C_2 \), there exists at least one flat \( L_i^2 \)-resolvents \( (1 \leq i \leq n) \) is not valid obtained by resolving \( C_2 \) with clauses in \( F \setminus \{C_1, C_2\} \), which is contradictory to the fact \( C_2 \) is an ESBC in \( F \). Therefore, \( C_2 \) is also an ESBC in \( F \setminus \{C_1\} \). Hence, equality-set-blocked clause is confluent. \( \square \)

Theorem 14. E-ESBCE is confluent.
Proof. Assume that a clause $C$ is an E-ESBC in a formula $F$, then the clause $C$ must be an E-ESBC in any subset of $F$. Assume that $C$ is not an E-ESBC in a subset $F'$ of $F$, then there exists an assignment $\alpha$ over the external ground atoms such that $C$ is equality-set-blocked upon subset $S_\alpha = \{L_1, L_2, \ldots, L_n\} \subseteq C$ in $F|\alpha$ while $C$ is not equality-set-blocked upon $S$ in $F'|\alpha$, which means there exists a flat $L_i^{S_\alpha}$-resolvents $(1 \leq i \leq n)$ is not valid obtained by resolving $C$ with clauses in $F'|\alpha \{C\}$. Apparently it is a contradiction that $C$ is an equality-set-blocked clause upon $S_\alpha = \{L_1, L_2, \ldots, L_n\} \subseteq C$ in $F|\alpha$. Therefore, $C$ is also an E-ESBC in $F'$. Hence, E-ESBC is confluent.

Table 1 shows that all the novel clause elimination methods have the confluence property.

<table>
<thead>
<tr>
<th>Clause elimination method</th>
<th>Confluence</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBCE</td>
<td>Yes</td>
</tr>
<tr>
<td>E-SBCE</td>
<td>Yes</td>
</tr>
<tr>
<td>ESBCE</td>
<td>Yes</td>
</tr>
<tr>
<td>E-ESBCE</td>
<td>Yes</td>
</tr>
</tbody>
</table>

6. Conclusions

In the paper, we generalized blocked clause in first-order logic further, proposing four types of redundant clauses set-blocked clause, extended set-blocked clause, equality-set-blocked clause and extended equality-set-blocked clause, of which the former two were suitable for formulas without equality while the latter two were suitable for formulas with equality. Besides, we proved the redundancies of the four types of clauses and they could be removed from formulas without influencing the satisfiability or unsatisfiability of the original formulas. Finally, we discussed and analyzed their effectiveness and confluence properties. It shows that the four clause elimination methods are more effective compared with blocked clause elimination and equality-blocked clause elimination, and all the four clause elimination methods have the confluence properties.

The paper is a theoretical work about the properties of the four types of clauses. Even though all the four clause elimination methods are more effective than blocked clause elimination and equality-blocked clause elimination, identification of the four types of clauses will be more complicated and more time-consuming in the specific implementation. In future work, we will implement those clause elimination methods as preprocessing techniques of first-order theorem provers by considering the balance between effectiveness and time consumption, expecting to improve the performance of first-order theorem provers.

Author Contributions: Conceptualization, X.N.; methodology, X.N.; validation, H.F., G.W. and Y.X.; writing—original draft preparation, X.N.; writing—review and editing, X.N.; supervision, Y.X.; funding acquisition, Y.X.

Funding: This research was funded by National Natural Science Foundation of China grant number 61673320 and Fundamental Research Funds for the Central Universities grant number 2682018ZT10.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (Grant No. 61673320), and by the Fundamental Research Funds for the Central Universities (Grant No. 2682018ZT10).

Conflicts of Interest: The authors declare no conflict of interest.

References


