Contradiction Separation Based Dynamic Multi-Clause Synergized Automated Deduction

Yang XU[1, 3], Jun LIU[†2, 3], Shuwei CHEN[1, 3], Xiaomei ZHONG[†1, 3] and Xingxing HE[1, 3]

1. School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China

2. School of Computing, Ulster University, Northern Ireland, UK

3. National-Local Joint Engineering Laboratory of System Credibility Automatic Verification, Southwest Jiaotong University, Chengdu 610031, China

Abstract

Resolution as a famous rule of inference has played a key role in automated reasoning for over five decades. A number of variants and refinements of resolution have been also studied, essentially, they are all based on binary resolution, that is, the cutting rule of the complementary pair while every deduction involves only two clauses. In the present work, we consider an extension of binary resolution rule, which is proposed as a novel contradiction separation based inference rule for automated deduction, targeted for dynamic and multiple (two or more) clauses handling in a synergized way, while binary resolution is its special case. This contradiction separation based dynamic multi-clause synergized automated deduction theory is then proved to be sound and complete. The development of this new extension is motivated not only by our view to show that such a new rule of inference can be generic, but also by our wish that this inference rule could provide a basis for more efficient automated deduction algorithms and systems.

Keywords: Propositional logic, first-order logic, resolution, automated deduction, theorem proving, contradiction separation, dynamic multi-clause synergized deduction

1. Introduction

Resolution [34] as a famous rule of inference is particularly suitable for automation so has played a key role in automated reasoning for over five decades [12, 31, 37]. In developing

† The corresponding authors: email: j.liu@ulster.ac.uk; zhongxm2013@home.swjtu.edu.cn

resolution based automated deduction, dozens of variants and refinements of resolution have been studied from both the empirical and analytical sides aimed at improving the efficiency of the deduction process, for the detailed review and collection of different variations or strategies please refer to [6, 8, 12, 13, 15, 37, 40]. It is worthy note that those methods are all essentially based on binary resolution inference rule and indeed have improved the efficiency and capability of resolution based ATP systems in different ways.

In its simplest form, binary resolution may be viewed as a procedure for deducting a new clause (from the two 'source' clauses) which is a result of eliminating the occurrences of a complementary pair while leaving a disjunction of everything else. There are two key characteristics in the binary resolution: 1) it is based on the cutting rule of the complementary pair from two clauses respectively; 2) in the whole resolution deduction process every deduction involves only two clauses, so can be regarded as a static deduction process in term of the clause involvement.

This simple and elegant binary resolution inference scheme has been very successful, however there are still a lot of real problems unsolved or not solved efficiently as illustrated in TPTP (Thousands of Problems for Theorem Provers) [44]. From the latest release of the TPTP benchmark library up to version 6.4.0 in 2016, there are 4,982 easy, 12,368 difficult, and 3,547 unsolved problems among 20897 problems for theorem provers, where as indicated in [20], more than a third of these problems have more than 100 axioms, more than 10% have more than 1000 axioms, and more than 5% have more than 10000 axioms. The efficiency and versatility of contemporary automated deduction depend on inference rule and techniques that may go beyond the pure resolution calculus, especially go beyond binary resolution [11, 31].

This present work aims at addressing the following questions: although the simple and elegant binary resolution inference scheme has been successful, has it been too restrictive? Instead of treating a contradiction as a complementary pair based on two clauses, can we extend it into a contradiction consists of more than two clauses? Accordingly, can we make a flexible or dynamic selection of the number of clauses involved in each deduction to get better efficiency and capability?

There have been previous earlier attempts to use chains of rules (e.g., generalized resolution [13, 36], hyper-resolution [35], and unit-resulting resolution [26]) which were not very successful. There has been up to now (to the best of our knowledge) no exist of theory and algorithm capable of handling multiple clauses dynamically in a synergized way, i.e., dynamic multi-clause synergized deduction.

Motivated by the above questions, plus our previous research work on resolution-based automated deduction based on many-valued logic [48], this paper proposes a new inference principle and its sound and complete automated deduction theory framework to extend from the existing static (i.e., fixed) binary resolution into a contradiction separation based dynamic multi-clause (two or more clauses) synergized inference rule. A key idea behind this new method is the extension of the concept of contradiction from a complementary pair based on two clauses to a typical kind of unsatisfiable clause set consists of more than two clauses. This typical kind of unsatisfiable clause set does not imply only one complementary pair among the clause set, therefore, the computation/searching is synergized among multiple clauses in terms of a contradiction.

The present work aims at establishing a new automated deduction theory with the following distinctive features in order to address the above questions: 1) *multi-clause deduction*: multiple clauses from a clause set (or even the whole clause set) are involved in each deduction process; 2) *dynamic and flexible deduction*: the number of clauses involved in each deduction can be varied from each other in the whole deduction process, so it is regarded as a dynamic and flexible deduction process; 3) *synergized deduction*: cooperative interaction among multiple clauses that creates a combined effect of all the clauses (two or more) on the deduction result, which can reflect better the overall logical relationship among multiple clauses than only considering two clauses several times; 5) *robust deduction*: deleting or adding some literals in the contradiction following certain strategies will not affect its contradictoriness as well as the corresponding deduction result; 6) *generic deduction*: it is generic, can be applied into a rich set of automated deduction systems, where the existing binary resolution rules and their variations are its special cases.

Note that an extended abstract of this paper was presented in [49]. The present paper provides a comprehensive introduction of all concepts and results with detailed proofs along with a good number of example illustrations. The focus of this paper is mainly on the new concept introduction and the corresponding automated deduction theory set up (i.e., soundness and completeness) to serve as a theoretical ground for the development of new provers. Therefore, the automated deduction theory is presented in a generic way, so that future work may easily build on it and explore various proof search strategies and implementation techniques. Actually, some specific algorithms and strategies to support this theory and achieve the implementation for automation with detailed experiments and case studies have been also established by the same author team, but are beyond the scope of this paper, so not covered here.

The remainder of this paper is structured as follows. In Section 2, we briefly review some related works. Followed some preliminaries about the notations and terminologies, the key concept of contradiction separation based deduction in propositional logic is provided in Section 3, along with soundness and completeness proved. Section 4 extends it into first-order logic. A graphical illustration of the key technical ideas is given in Section 5. The paper is concluded in Section 6.

2. Related Work

As indicated in the Introduction section, a lot of variants of resolution or strategies have been studied from both the empirical and analytical sides. For example, strategies by restricting or specifying the resolution path [36, 37], including set-of-support strategy as one of the most powerful strategies of this kind [47], semantic resolution [24, 36], block resolution [24], linear resolution [32] and lock-resolution [4]; strategies like hyper-resolution to reduce the number of intermediate resolvents by combining several resolution steps into a single inference step [36]; strategies to specify the selection of clauses or literals [1, 15, 37, 39]; resolution supplemented by heuristic strategies in the deduction process [7, 10, 16, 38, 40]; reducing the search space [9, 25, 29, 33]; splitting the clause set [16]; reducing the function terms by using equality [28] etc., among others. These methods indeed improved the efficiency and capability of resolution deduction in different ways and in different extents, but they are all essentially based on binary resolution inference rule.

Some resolution deductions did consider to handle several clauses, such as Robinson (1968) [36] and Harrison and Rubin (1978) [13] independently proposed generalized resolution principle respectively, although both generalizations seem to handle several clauses, both share the key concept and have two essential features: 1) the clauses involved in the resolution are all binary clauses; 2) there must exist a special clause which includes a negation of literal appeared in those binary clauses. It is not easy to find that special clause in practical implementation. In addition, its soundness is still on the basis of binary resolution. Therefore, their works were not further developed and followed up since then. Another interesting one is hyper-resolution [36], which is a multi-step binary resolution process where intermediate clauses are discarded. The clauses to be resolved are divided into two types: clauses with only positive literals are referred to as electrons and a selected clause containing one or more negative literals which is referred to as the nucleus. The nucleus is resolved with a series of electrons until the final resultant clause itself is an electron (contains no negative literals) and this is the output of the hyper-resolution step. Hyper-resolution is complete and will reduce the number of generated clauses as only one clause is generated for several resolution steps but the proof found may require more steps overall, negating some of the advantage. The theorem prover Otter and its successor Prover9 [27] use hyper-resolution. Hyper-resolution can be viewed as a sequence of binary resolution steps ending with a positive clause. In addition, although some simplification technique in propositional logic called blocked clause elimination [14, 18, 23] or super-blocked clauses [21] consider handling several clauses together, they are based on redundancy property, are only a kind of simplification process, and are quite different from the proposed work which actually is an inference rule, i.e., the result from the logical inference, not an equivalent result in terms of satisfiability in simplification process. Some most recent works have shown different visions to advanced automated reasoning from different points of view, such as s new paradigm called explainable Artificial Intelligence is proposed which explores the relationship between automated reasoning and machine learning [2]; a semantically-guided goal-sensitive reasoning was proposed in [3], but not trying to expand the resolution; [5] introduces an extension of the resolution calculus called conflict resolution calculus, where the resolution inference rule is restricted to (first-order) unit propagation and the calculus is extended with a mechanism for assuming decision literals and with a new inference rule for clause learning, which is a first-order generalization of the propositional conflict-driven clause learning procedure. In addition, a theorem prover based on conflict resolution called Scavenger 0.1 is proposed in [17]; A unifying principle for clause elimination in first-order logic as one of a preprocessing techniques for formulas in CNF was proposed in [22]; abstract interpretation is discussed in [42] to explore how it can be linked with algorithmic deduction applied in automated deduction; the new method and tool is proposed in [41] to explore how automated reasoning can be applied to detecting inconsistencies in large first-order knowledge bases; Superposition is further highlighted and reviewed in [45] and saturation with redundancy as preprocessing techniques is further checked in [46], all these work aimed at advancing the current automated reasoning.

From the literature review, we noticed that most of the advancements in the area of resolution-based first-order automated deduction since Robinson's seminal paper were in the direction of addressing the problem of restricting resolution search space from either syntactic or semantic point of view while preserving its completeness. They are all focused on binary resolution. As a question raised earlier, has binary resolution been too restrictive? Can we go beyond binary resolution, e.g., a dynamic multi-clause synergized deduction, to provide the basis for the more efficient automated deduction? This is the main motivation of the present work.

3. Contradiction Separation Based Deduction in Propositional Logic

We need some preliminaries first. We consider the propositional formula in *conjunctive normal forms* (CNF) which are defined as follows.

A *literal* is either a propositional logic variable p or its negation $\sim p$. Two literals are said to be *complements* or *a complementary pair* if one is the negation of the other (e.g., $\sim p$ is taken to be the complement to p).

A *clause C*, is an expression formed from a finite collection of literals, is a disjunction of literals usually written as $C= p_1\vee\cdots\vee p_k$, where p_i (i = 1, …, k) is a literal. In the subsequent section, for the notation simplicity, we also use C to denote the set of literals in C. The readers can easily distinguish it from the context, regarded as either a disjunction of literals or a set of literals.

A clause can be empty (defined from an empty set of literals), denoted by $\emptyset$. The truth assignment of an empty clause is always *false*.

A *propositional formula* is a conjunction of clauses, i.e., *a conjunctive normal form* (CNF). A formula in CNF, $S = C_1\wedge\cdots\wedge C_m$, is usually regarded a set of clauses, written as $S =\{C_1,\dots, C_m\}$. So in the subsequent section, $S = C_1\wedge\cdots\wedge C_m$ is an equivalent expression to $S =\{C_1,\dots, C_m\}$ as a clause set.

A formula is said to be *satisfiable* if it can be made TRUE by assigning appropriate logical values (i.e. TRUE, FALSE) to its variables. We refer the reader to, for instance, [34], for more details about logical notations and resolution concept.

Definition 3.1 Let S_1 and S_2 be two propositional formulae. If for any true assignment I, $I(S_1) \le I(S_2)$, then it is denoted as $S_1 \le S_2$.

Definition 3.2 Let $S =\{C_1, C_2,\dots, C_m\}$ be a clause set. The Cartesian product of C_1, C_2, …, C_m, denoted as $\prod_{i=1}^{m} C_i$, is the set of all ordered tuples $(p_1,\dots, p_m)$ such that $p_i\in C_i$ (i=1,…, m), where p_i is a literal, and C_i is also regarded as a set of literals (i=1,…, m).

Central to the present discussion is the notion of a *contradiction*. We have the following contradiction definition, which expands the normal way of defining a contradiction as a complementary pair, considers the contradictory normal form as a whole instead.

Definition 3.3 (Contradiction) Let $S =\{C_1, C_2,\dots, C_m\}$ be a clause set. If $\forall(p_1,\dots, p_m)\in \prod_{i=1}^{m} C_i$, there exists at least one complementary pair among $\{p_1,\dots, p_m\}$, then $S=\bigwedge_{i=1}^{m} C_i$ is called a *standard contradiction* (in short, SC). If $\bigwedge_{i=1}^{m} C_i$ is unsatisfiable, then $S=\bigwedge_{i=1}^{m} C_i$ is called a *quasi-contradiction* (in short, QC).

Remark 3.1: from Definition 3.3, a contradiction does not simply contain only two clauses with one complementary pair (which certainly is a special case of contradiction). It can contain more than two clauses, and can be regarded a collection (or group) of those contradictory clauses. This concept plays a critical role in the subsequent novel inference rule and automated deduction.

Lemma 3.1 Assume a clause set $S=\{C_1, C_2, \ldots, C_m\}$ in propositional logic. Then S is a standard contradiction if and only if S is a quasi-contradiction.

Proof. See the detailed proof in Appendix.

Remark 3.2: according to Lemma 3.1, in propositional logic, a standard contradiction is equivalent to a quasi-contradiction, so we just call them *contradiction* in short. However, as discussed in Section 4, in general, this conclusion does not hold for first-order logic, so they are discussed separately in Section 4.

Remark 3.3: it follows also from Lemma 3.1 that whether $\wedge_{i=1}^{m} C_i$ is a contradiction or not is regardless of the ordering of $C_1, C_2, \ldots, C_m$.

Lemma 3.2 In propositional logic, a clause set $S=C_1 \wedge C_2$ is unsatisfiable if and only if C_1 and C_2 are single-literal clauses and $C_1=\sim C_2$.

Proof. See the detailed proof in Appendix.

Based on the above definitions and lemmas, we introduce the following new concept.

Definition 3.4 (**Contradiction Separation Rule in Propositional Logic**) Assume a clause set $S=\{C_1, C_2, \ldots, C_m\}$. The following inference rule that produces a new clause from S is called *a contradiction separation rule*, in short, a CS rule:

For each C_i (i=1,…, m), separate it into two sub-clauses C_i^- and C_i^+ such that

(1) $C_i=C_i^- \vee C_i^+$, where C_i^- and C_i^+ have no common literals;

(2) C_i^+ can be an empty clause itself, but C_i^- cannot be an empty clause;

(3) $\wedge_{i=1}^{m} C_i^-$ is a standard contradiction.

The resulting clause $\vee_{i=1}^{m} C_i^+$, denoted as $\mathcal{C}_m(C_1, C_2, \ldots, C_m)$, is called a *contradiction separation clause* (CSC) of $C_1, C_2, \ldots, C_m$, and $\wedge_{i=1}^{m} C_i^-$ is called a *separated contradiction* (SC).

Remark 3.4: note that in the above CS rule, some clauses in $\{C_1, C_2, \ldots, C_m\}$ can be repeated, and the CSC $\mathcal{C}_m(C_1, C_2, \ldots, C_m)$ is regardless of the ordering of $C_1, C_2, \ldots, C_m$. In addition, since contradiction is actually a group of the contradictory clauses, $\mathscr{C}_m(C_1, C_2,\ldots, C_m)$ represents the resulting clause of the CS rule, especially resulted from $C_1, C_2,\ldots, C_m$ through multiple contradictory clauses separation. Actually, as justified in the subsequent section that $\mathscr{C}_m(C_1, C_2,\ldots, C_m)$ is the logical consequence of $C_1, C_2,\ldots, C_m$, is not used as an operator or function.

Remark 3.5: binary resolution rule is actually a special case of the CS rule when only two clauses are involved in the contradiction separation process. Different from binary resolution where only the complementary pair is excluded, the CS rule means the contradiction as a group of multiple contradictory clauses itself (a set of sub-clauses which is unsatisfiable) is jointly eliminated regardless of how many other literals involved in each clause. A CS step will then create a new clause by combining together the leftover literals in those clauses. This allows much bigger steps of deduction. This reflects the key motivation of the proposed CS rule: these bigger deduction steps are expected to allow the CS-based automated deduction to solve problems faster or to solve more problems.

The above facts are illustrated by the following examples.

Example 3.1 Let C_1=~p_3∨~p_7, C_2=p_2∨p_3∨p_5∨~p_6, C_3=p_1∨~p_2∨p_5∨~p_7, C_4=p_1∨p_3∨~p_5, C_5= p_3∨p_4∨p_6, C_6=~p_1∨p_3, and C_7=p_7. Then it follows from the CS rule (Definition 3.4 and Table 3.1) that one CSC involving 7 clauses is $\mathscr{C}_7$ (C_1, C_2, C_3, C_4, C_5, C_6, C_7)= p_4, while the corresponding SC is

(~p_3∨~p_7)∧(p_2∨p_3∨p_5∨~p_6)∧(p_1∨~p_2∨p_5∨~p_7)∧(p_1∨p_3∨~p_5)∧(p_3∨p_6)∧(~p_1∨p_3)∧(p_7).

Table 3.1 The sub-clauses C_i^- and C_i^+ for C_1, C_2, C_3, C_4, C_5, C_6, C_7

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_i^+					p_4		
C_i^-	~p_3∨~p_7	p_2∨p_3∨p_5∨~p_6	p_1∨~p_2∨p_5∨~p_7	p_1∨p_3∨~p_5	p_3∨p_6	~p_1∨p_3	p_7

However, it will need several steps of binary resolutions on (C_1, C_2, C_3, C_4, C_5, C_6, C_7) to obtain p_4 as illustrated below:

C_8= R(C_1, C_7)= ~p_3; C_9= R (C_3, C_7)= p_1∨~p_2∨p_5

C_{10}= R (C_2, C_8)= p_2∨p_5∨~p_6; C_{11}= R (C_4, C_8)= p_1∨~p_5

C_{12}= R (C_5, C_8)= p_4∨p_6; C_{13}= R (C_6, C_8)= ~p_1

$C_{14}= R\ (C_9, C_{13})= \sim p_2 \vee p_5$; $C_{15}= R\ (C_{11}, C_{13})= \sim p_5$

$C_{16}= R\ (C_{10}, C_{15})= p_2 \vee \sim p_6$; $C_{17}= R\ (C_{14}, C_{15})= \sim p_2$

$C_{18}= R\ (C_{16}, C_{17})= \sim p_6$; $C_{19}= R\ (C_{12}, C_{18})= p_4$

This example shows that the CS rule could go beyond the binary resolution in terms of efficiency, depending on the strategy making the suitable CS step.

Remark 3.6: suppose two sequences of clauses C_i and D_i, i=1,…, n, such that $C_i \leq D_i$ respectively. It does not normally follow that $\mathcal{C}_n(C_1, C_2,\ldots, C_n) \leq \mathcal{C}_n\ (D_1, D_2,\ldots, D_n)$. This also implies that the CS rule reflects the synergized effects of all the clauses involved in the CS deduction process**.** It can be illustrated in the following example.

Example 3.2 Let two clause sets

S_C: $C_1=p_1$, $C_2=p_2 \vee p_4$, $C_3=\sim p_1 \vee p_3$, $C_4=\sim p_3$;

S_D: $D_1=p_1$, $D_2=\sim p_1 \vee p_2 \vee p_4$, $D_3=\sim p_1 \vee \sim p_2 \vee p_3$, $D_4=\sim p_3$.

It is easy to note that $C_i \leq D_i$, i=1,…, 4.

1) For the clause set S_C, we have

Table 3.2 The sub-clauses C_i^- and C_i^+ for C_1, C_2, C_3, C_4

	C_4	C_3	C_2	C_1
C_i^+			p_2	
C_i^-	$\sim p_3$	$\sim p_1 \vee p_3$	p_4	p_1

It follows that $\mathcal{C}_4\ (C_1, C_2, C_3, C_4)=p_2$ (the corresponding SC is $\{p_1\} \wedge \{p_4\} \wedge \{\sim p_1, p_3\} \wedge \{\sim p_3\}$).

2) For the clause set S_D, we have

Table 3.3 The sub-clauses D_i^- and D_i^+ for D_1, D_2, D_3, D_4

	D_4	D_3	D_2	D_1
D_i^+			p_4	
D_i^-	$\sim p_3$	$\sim p_1 \vee \sim p_2 \vee p_3$	$\sim p_1 \vee p_2$	p_1

It follows that $\mathcal{C}_4\ (D_1, D_2, D_3, D_4)=p_4$ (the corresponding SC is $\{p_1\} \wedge \{\sim p_1, p_2\} \wedge \{\sim p_1, \sim p_2, p_3\} \wedge \{\sim p_3\}$). However, $\mathcal{C}_4\ (C_1, C_2, C_3, C_4) \nleq \mathcal{C}_4(D_1, D_2, D_3, D_4)$.

Definition 3.5 Suppose a clause set $S = \{C_1, C_2,\ldots, C_m\}$ in propositional logic. $\Phi_1, \Phi_2,\ldots, \Phi_t$ is called a *contradiction separation based dynamic deduction sequence* (*or a CS based dynamic deduction sequence) from S to a clause* Φ_t, denoted as $\mathcal{D}$, if

(1) $\Phi_i \in S$, i =1, 2, …, t; or

(2) there exist $r_1, r_2, \ldots, r_{k_i} < i$, $\Phi_i = \mathscr{C}_{k_i}(\Phi_{r_1}, \Phi_{r_2}, \ldots, \Phi_{r_{k_i}})$.

Remark 3.8: the k_i in (2) varies with the deduction process, which means that the number of clauses involved in the contradiction separation in each deduction process could be different from each other, i.e., not fixed. This reflects the meaning of "dynamic deduction". This is another key motivation of the proposed automated deduction. Dynamic selection of different numbers of clauses during the deduction process provides much flexibility and enhances the adaptive behaviour of the automated deduction. This is quite similar to the local search in an optimization problem: due to the restriction into two clauses in binary resolution, it may be easy to get stuck somewhere so stop the proof search. The dynamic nature, i.e., the flexibility in selecting the number of clauses in the proposed CS-based dynamic deduction actually provides an effective way to overcome the two-clause restriction and continue the proof search using multiple paths.

Both Remarks 3.4 and 3.8 clarify the key motivations of the present research. More specifically, the key contribution of the present paper is to introduce and justify theoretically that this CS-based dynamic deduction is sound and complete, so will play an role of theoretical foundation for the present research.

Two examples below are provided to illustrate the key features of this CS-based dynamic deduction.

Example 3.3 Suppose a clause set $S = \{C_1, C_2, \ldots, C_{13}\}$ in propositional logic with

C_1: $\sim p_4 \vee p_6$, C_2: $p_6 \vee \sim p_7$, C_3: $\sim p_6 \vee p_7$, C_4: $\sim p_6 \vee \sim p_7$, C_5: $p_1 \vee p_2 \vee p_3$, C_6: $p_1 \vee p_2 \vee \sim p_3$

C_7: $\sim p_1 \vee p_2 \vee p_3$, C_8: $\sim p_1 \vee \sim p_2 \vee p_3$, C_9: $\sim p_1 \vee \sim p_2 \vee \sim p_3$, C_{10}: $p_4 \vee \sim p_5 \vee p_7$

C_{11}: $p_1 \vee \sim p_2 \vee p_3 \vee p_4$, C_{12}: $p_1 \vee \sim p_2 \vee \sim p_3 \vee p_5$, C_{13}: $\sim p_1 \vee p_2 \vee \sim p_3 \vee p_6$.

Using the CS rule for the clauses $C_5, C_6, C_7, C_8, C_9, C_{11}, C_{12}, C_{13}$, we obtain a CSC involving 8 clauses:

$$C_{14} = \mathscr{C}_8 (C_5, C_6, C_7, C_8, C_9, C_{11}, C_{12}, C_{13}) = p_4 \vee p_5 \vee p_6.$$

The corresponding SC is:

$$(p_1 \vee p_2 \vee p_3) \wedge (p_1 \vee p_2 \vee \sim p_3) \wedge (\sim p_1 \vee p_2 \vee p_3) \wedge (\sim p_1 \vee \sim p_2 \vee p_3) \wedge (\sim p_1 \vee \sim p_2 \vee \sim p_3) \wedge (p_1 \vee \sim p_2 \vee p_3) \wedge (p_1 \vee \sim p_2 \vee \sim p_3) \wedge (\sim p_1 \vee p_2 \vee \sim p_3).$$

Furthermore, using the CS rule for 3 clauses C_1, C_{10}, and C_{14}, we obtain another CSC involving 3 clauses:

$C_{15}= \mathscr{C}_3\ (C_1, C_{10}, C_{14}) = p_6 \vee p_7$.

The corresponding SC is: $(\sim p_4) \wedge (p_4 \vee \sim p_5) \wedge (p_4 \vee p_5)$.

Finally, we have

$C_{16}=\mathscr{C}_4\ (C_2, C_3, C_4, C_{15}) =\varnothing$

The corresponding SC: $(p_6 \vee \sim p_7) \wedge (\sim p_6 \vee p_7) \wedge (\sim p_6 \vee \sim p_7) \wedge (p_6 \vee p_7)$.

The above process illustrates a CS based dynamic deduction from S to an empty clause $\varnothing$ using 3 steps of CS deduction.

Below shows binary resolution in multiple steps:

$C_{14}= R\ (C_2, C_4) = \sim p_7$; $C_{15}= R\ (C_3, C_{14}) = \sim p_6$; $C_{16}= R\ (C_{10}, C_{14}) = p_4 \vee \sim p_5$

$C_{17}= R\ (C_1, C_{15}) = \sim p_4$; $C_{18}= R\ (C_{13}, C_{15}) = \sim p_1 \vee p_2 \vee \sim p_3$

$C_{19}= R\ (C_{11}, C_{17}) = p_1 \vee \sim p_2 \vee p_3$; $C_{20}= R\ (C_{16}, C_{17}) = \sim p_5$

$C_{21}= R\ (C_5, C_6) = p_1 \vee p_2$; $C_{22}= R\ (C_5, C_{19}) = p_1 \vee p_3$

$C_{23}= R\ (C_7, C_8) = \sim p_1 \vee p_3$; $C_{24}= R\ (C_9, C_{12}) = \sim p_2 \vee \sim p_3$

$C_{25}= R\ (C_9, C_{18}) = \sim p_1 \vee \sim p_3$; $C_{26}= R\ (C_{21}, C_{23}) = p_2 \vee p_3$

$C_{27}= R\ (C_{21}, C_{25}) = p_2 \vee \sim p_3$; $C_{28}= R\ (C_{26}, C_{27}) = p_2$

$C_{29}= R\ (C_{24}, C_{28}) = \sim p_3$; $C_{30}= R\ (C_{22}, C_{29}) = p_1$

$C_{31}= R\ (C_{23}, C_{29}) = \sim p_1$; $C_{32}= R\ (C_{30}, C_{31}) = \varnothing$

Remark 3.9: from the above example, compared with binary resolution, the CS rule has the following distinctive features: 1) the number of clauses involved in each CS process can be more than two, while the number of literals deleted through the CS process is much more than one binary resolution process. For example, the first CS process to obtain C_{14}, 32 literals were deleted in one step; however, only two literals are deleted in each binary resolution; it also follows that the size of a CSC is normally much small than the size of binary resolvent; 2) the number of clauses involved in the CS deduction is not fixed (e.g., 8, 3 and 4 for those 3 CS processes respectively in Example 3.3), which reflects the dynamic feature; 3) each deduction reflects the synergized effects of all the clauses involved and reduce the deduction steps. For example, it is not easy or straightforward to obtain the CSC $C_{14} = p_4 \vee p_5 \vee p_6$ from C_5, C_6, C_7, C_8,

C_9, C_{11}, C_{12}, C_{13} using multiple steps of binary resolution; however, as illustrated above, C_{14} play an important role in obtaining the empty clause $C_{16}=\mathscr{C}_4$ (C_2, C_3, C_4, C_{15}) $=\emptyset$.

Lemma 3.3 (Soundness Lemma of the CS-Based Dynamic Deduction in Propositional Logic) Suppose D_k, D_{k-1},..., D_1 are k clauses in propositional logic, where $D_i=D_i^+\vee D_i^-$, $i=1,\ldots,k$. If $\bigwedge_{i=1}^{k} D_i^-$ is unsatisfiable, then $D_k\wedge D_{k-1}\wedge\ldots\wedge D_1 \le D_k^+\vee D_{k-1}^+\vee\ldots\vee D_1^+$, that is,

$$D_k\wedge D_{k-1}\wedge\ldots\wedge D_1 \le \mathscr{C}_k(D_k, D_{k-1},\ldots, D_1).$$

Proof. See the detailed proof in Appendix.

Theorem 3.1 (**Soundness Theorem of the CS-Based Dynamic Deduction in Propositional Logic**) Suppose a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in propositional logic. Φ_1, Φ_2,…, Φ_t is a CS based dynamic deduction sequence from S to a clause Φ_t. If Φ_t is an empty clause, then S is unsatisfiable.

Proof. It follows from the definition of a CS-based dynamic deduction sequence (Definition 3.5) and also the soundness lemma (Lemma 3.3) that

$$C_1\wedge C_2\wedge\ldots\wedge C_m\le\Phi_1\wedge\Phi_2\wedge\ldots\wedge\Phi_t\le\Phi_t .$$

This concludes the proof.

Theorem 3.2 (**Completeness Theorem of the CS-Based Dynamic Deduction in Propositional Logic**) Suppose a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in propositional logic. If S is unsatisfiable, then there exists a CS based dynamic deduction sequence from S to an empty clause.

Proof. One way to prove the completeness is based on the fact that standard binary resolution is a special case of the CS deduction, i.e., any binary resolution derivation can be represented as a single application of the CS rule involving two clauses. However, for the readers' interest and also for the integrity of the proposed work, the detailed proof by induction is given in Appendix.

4. Contradiction Separation Based Deduction in First-Order Logic

First-order logic in the present paper follows the standard way of definition. Some basic concepts and notations are given briefly here.

In the first-order logic, a *literal* is either an atom or a negated atom, where an *atom* is an n-ary predicate (denoted P or Q) applied to n terms. A *term* is either a constant (denoted a or b), a variable (denoted x, y, v or z) or an n-ary function (denoted f or g) applied to n terms. A *clause* is simply a disjunction of literals where all variables are universally quantified.

Substitutions (denoted by σ, possibly superscripted) is a mapping from variables to terms. Considering a clause C, we write C^{σ} to denote the result of substituting each assigned variable with the assigned term in C. The empty (i.e. identity) substitution is denoted ε. If none of the terms in a substitution contains a variable, i.e., all the terms in the substitution are *ground terms*, we have a so-called *ground substitution*. If σ is a (ground) substitution, then C^{σ} is called an (*ground*) *instance* of C.

A substitution σ is a *unifier* of terms $e_1,\ldots, e_n$ if and only if $\sigma(e_1)=\ldots=\sigma(e_n)$ where "=" denotes syntactic identity. A unifier σ is a *most general unifier* (*mgu*) of $e_1,\ldots, e_n$ if and only if for every unifier σ' of $e_1,\ldots, e_n$ there exists a substitution σ'' such that $\sigma'(e_i) = \sigma''(\sigma(e_i))$ for all $e_i \in \{e_1,\ldots, e_n\}$. If a set of terms of first-order logic can be unified, there exists a *mgu*.

Definition 4.1 (Standard Contradiction Separation Rule in First-Order Logic) Suppose a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in first-order logic. Without loss of generality, assume that there does not exist the same variables among $C_1, C_2,\ldots, C_m$ (if there exists the same variables, there exists a rename substitution which makes them different). The following inference rule that produces a new clause from S is called *a standard contradiction separation rule*, in short, an S-CS rule:

For each C_i (i=1, 2,…, m), firstly apply a substitution σ_i to C_i (σ_i could be an empty substitution but not necessary the most general unifier), denoted as $C_i^{\sigma_i}$; then separate $C_i^{\sigma_i}$ into two sub-clauses $C_i^{\sigma_i -}$ and $C_i^{\sigma_i +}$ such that

i) $C_i^{\sigma_i} = C_i^{\sigma_i -} \vee C_i^{\sigma_i +}$, where $C_i^{\sigma_i -}$ and $C_i^{\sigma_i +}$ have no common literals;

ii) $C_i^{\sigma_i +}$ can be an empty clause itself, but $C_i^{\sigma_i -}$ cannot be an empty clause;

iii) $\bigwedge_{i=1}^{m} C_i^{\sigma_i -}$ is a standard contradiction, that is $\forall(x_1,\ldots, x_m)\in \prod_{i=1}^{m} C_i^{\sigma_i -}$, there exists at least one complementary pair among $\{x_1,\ldots, x_m\}$.

The resulting clause $\bigvee_{i=1}^{m} C_i^{\sigma_i +}$, denoted as $\mathcal{C}_m^{s\sigma}(C_1,\ldots, C_m)$ (here "s" means "standard", $\sigma = \bigcup_{i=1}^{m} \sigma_i$, σ_i is a substitution to C_i, i=1,…, m), is called a *standard contradiction separation clause* (S-CSC) of $C_1,\ldots, C_m$, and $\bigwedge_{i=1}^{m} C_i^{\sigma_i -}$ is called a *separated standard contradiction* (S-SC).

Remark 4.1: it is apparent that whether $\bigwedge_{i=1}^{m} C_i^{\sigma_i -}$ is a standard separated contradiction or not is regardless of the ordering of $C_1, C_2,\ldots, C_m$. The S-CSC $\mathcal{C}_m^{s\sigma}(C_1,\ldots, C_m)$ is also regardless of the ordering of $C_1, C_2,\ldots, C_m$. Similar to Remark 3.4, some clauses in $C_1, C_2,\ldots, C_m$ can be repeated. In the first-order logic case, $\mathcal{C}_m^{s\sigma}(C_1,\ldots, C_m)$ is also resulted from $C_1, C_2,\ldots, C_m$ through a multiple clause synergized deduction!

Definition 4.2 (Quasi-Contradiction Separation Rule in First-Order Logic) Suppose a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in first-order logic. Without loss of generality, assume that there does not exist the same variables among $C_1, C_2,\ldots, C_m$ (if there exists the same variables, there exists a rename substitution which makes them different). The following inference rule that produces a new clause from S is called *a quasi-contradiction separation rule*, in short, a Q-CS rule:

For each C_i (i=1, 2,…, m), firstly use a substitution σ_i to C_i (σ_i could be an empty substitution but not necessary the most general unifier), denoted as $C_i^{\sigma_i}$; then separate $C_i^{\sigma_i}$ into two sub-clauses $C_i^{\sigma_i -}$ and $C_i^{\sigma_i +}$ such that

i) $C_i^{\sigma_i} = C_i^{\sigma_i -} \vee C_i^{\sigma_i +}$, where $C_i^{\sigma_i -}$ and $C_i^{\sigma_i +}$ have no common literals;

ii) $C_i^{\sigma_i +}$ can be an empty clause itself, but $C_i^{\sigma_i -}$ cannot be an empty clause;

iii) $\bigwedge_{i=1}^{m} C_i^{\sigma_i -}$ is unsatisfiable.

The resulting clause $\bigvee_{i=1}^{m} C_i^{\sigma_i +}$, denoted as $\mathcal{C}_m^{q\sigma}(C_1, C_2,\ldots, C_m)$ (here "q" means "quasi", $\sigma = \bigcup_{i=1}^{m} \sigma_i$, σ_i is a substitution to C_i, i=1,…, m), is called a *quasi-contradiction separation clause* (Q-CSC) of $C_1, C_2,\ldots, C_m$, and $\bigwedge_{i=1}^{m} C_i^{\sigma_i -}$ is called a *separated quasi-contradiction* (S-QC).

Remark 4.2: it is apparent that if $\bigwedge_{i=1}^{m} C_i^{\sigma_i -}$ is a quasi-contradiction that is regardless of the ordering of $C_1, C_2, \ldots, C_m$, the Q-CSC $\mathscr{C}_m^{q\sigma}(C_1, C_2, \ldots, C_m)$ is also regardless of the ordering of $C_1, C_2, \ldots, C_m$, and some clauses in $C_1, C_2, \ldots, C_m$ can be repeated.

Remark 4.3: the reason that the variation Q-CS in the first-order case is introduced and discussed is due to the following facts: if $S=\{C_1, C_2, \ldots, C_m\}$ is unsatisfiable, it does not mean that S is a standard contradiction. For example, considering $S=\{P(x), \sim P(f(y))\}$, it is obviously that S is unsatisfiable, however, $P(x)$ and $\sim P(f(y))$ is not a complementary pair. Therefore, in first-order logic, quasi-contradiction may not be a standard contradiction, but the following relationship holds.

Lemma 4.1 Suppose a clause set $S=\{C_1, C_2, \ldots, C_m\}$ in first-order logic. If $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ is a standard contradiction, then S is a quasi-contradiction (i.e., S is unsatisfiable).

Proof. See the detailed proof in Appendix.

From the proof of Lemma 4.1, we have the following corollary:

Corollary 4.1 (Invariance of Standard Contradiction in terms of Variable Substitution) Suppose $S=\{C_1, C_2, \ldots, C_m\}$, where $C_1, C_2, \ldots, C_m$ are clauses in the first-order logic. If $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ is a standard contradiction, then for any variable substitution σ of S, $(C_1 \wedge C_2 \wedge \ldots \wedge C_m)^{\sigma}$ is also a standard contradiction.

Remark 4.4: a quasi-contradiction obtained after applying some substitution may not be a quasi-contradiction any more. For example, considering $S=\{P(x), \sim P(f(a))\}$, here a is a constant, obviously S is an unsatisfiable clause set. Suppose a substitution $\sigma=\{a/x\}$, then $S^{\sigma} = \{P(a), \sim P(f(a))\}$. Obviously, S^{σ} is satisfiable.

Definition 4.3 Suppose a clause set $S=\{C_1, C_2, \ldots, C_m\}$ in first-order logic. $\Phi_1, \Phi_2, \ldots, \Phi_t$ is called *a standard contradiction separation based dynamic deduction sequence* (*or a S-CS based dynamic deduction sequence* from S to a clause Φ_t, denoted as $\mathscr{D}^s$, if

(1) $\Phi_i \in S$, $i=1, 2, \ldots, t$; or

(2) there exist $r_1, r_2, \ldots, r_{k_i} < i$, $\Phi_i = \mathscr{C}_{r_{k_i}}^{s\theta_i}(\Phi_{r_1}, \Phi_{r_2}, \ldots, \Phi_{r_{k_i}})$, where $\theta_i = \bigcup_{j=1}^{k_i} \sigma_j$, σ_j is a substitution to Φ_{r_j}, $j=1, \ldots, k_i$.

Two examples below are provided to illustrate the key features of this S-CS-based dynamic deduction.

Example 4.1 Let $S=\{C_1, C_2,\ldots, C_{11}\}$ be a clause set in first-order logic, where

C_1: $P_1(a_1)\vee\sim P_2(x_{11})$ C_2: $\sim P_1(x_{21})\vee P_2(a_2)$ C_3: $\sim P_1(a_1)\vee\sim P_2(x_{31})$

C_4: $P_3(x_{41}, a_1)\vee P_4(x_{42})\vee P_5(f_2(a_2, x_{43}), x_{43})$ C_5: $P_3(x_{51}, x_{52})\vee P_4(f_1(x_{53}))\vee\sim P_5(x_{54}, x_{55})$

C_6: $\sim P_3(a_2, x_{61})\vee P_4(x_{62})\vee P_5(f_2(x_{63}, x_{64}), a_2)$ C_7: $\sim P_3(x_{71}, x_{72})\vee\sim P_4(f_1(a_1))\vee P_5(f_2(x_{73},a_2), x_{73})$

C_8: $\sim P_3(a_2, x_{81})\vee\sim P_4(f_1(x_{81}))\vee\sim P_5(x_{82}, x_{83})$ C_9: $P_3(a_2, a_1)\vee\sim P_4(x_{91})\vee P_5(x_{92}, a_2)$

C_{10}: $P_3(x_{101}, a_1)\vee\sim P_4(f_1(x_{102}))\vee\sim P_5(f_2(a_2, a_2), x_{103})\vee P_1(x_{102})$

C_{11}: $\sim P_3(a_2, x_{111})\vee P_4(x_{112})\vee\sim P_5(x_{113}, x_{114})\vee P_2(a_2)$

Here a_1, a_2 are constants; x_{11}, x_{21}, x_{31}, x_{41},x_{42}, x_{43}, x_{51}, x_{52}, x_{53}, x_{54}, x_{55}, x_{61}, x_{62}, x_{63}, x_{64}, x_{71}, x_{72}, x_{73}, x_{81}, x_{82}, x_{83}, x_{91}, x_{92}, x_{101}, x_{102}, x_{103}, x_{111}, x_{112}, x_{113}, x_{114} are variables; $P_1,\ldots, P_5$ are predicate symbols; f_1, f_2 are function symbols.

Now using the S-CS rule for 8 clauses $C_4 - C_{11}$, we obtain an S-CSC involving 8 clauses: $C_{12}= \mathcal{C}_8^{s\theta_{12}}(C_4, C_5,\ldots, C_{11})= P_1(a_1)\vee P_2(a_2)$, where the corresponding S-SC is:

$(P_3(a_2, a_1)\vee P_4(f_1(a_1))\vee P_5(f_2(a_2, a_2), a_2)) \wedge(P_3(a_2, a_1)\vee P_4(f_1(a_1))\vee\sim P_5(f_2(a_2, a_2), a_2))\wedge$
$(\sim P_3(a_2, a_1)\vee P_4(f_1(a_1))\vee P_5(f_2(a_2, a_2), a_2)) \wedge (\sim P_3(a_2, a_1)\vee\sim P_4(f_1(a_1))\vee P_5(f_2(a_2,a_2), a_2)) \wedge$
$(\sim P_3(a_2, a_1)\vee\sim P_4(f_1(a_1))\vee\sim P_5(f_2(a_2, a_2), a_2)) \wedge(P_3(a_2, a_1)\vee\sim P_4(f_1(a_1))\vee P_5(f_2(a_2, a_2), a_2)) \wedge$
$(P_3(a_2,a_1)\vee\sim P_4(f_1(a_1))\vee\sim P_5(f_2(a_2,a_2), a_2)) \wedge(\sim P_3(a_2, a_1)\vee P_4(f_1(a_1))\vee\sim P_5(f_2(a_2, a_2), a_2))$

Furthermore, we obtain another S-CSC involving 4 clauses:

$C_{13}= \mathcal{C}_4^{s\theta_{13}}(C_1, C_2, C_2, C_{12})= \varnothing$,

where the corresponding S-SC is:

$(P_1(a_1)\vee\sim P_2(a_2))\wedge(\sim P_1(a_1)\vee P_2(a_2))\wedge(\sim P_1(a_1)\vee\sim P_2(a_2))\wedge(P_1(a_1)\vee P_2(a_2))$.

This illustrates an S-CS-based dynamic deduction sequence from S to an empty clause.

Below shows binary resolution in multiple steps from S to an empty clause:

C_{12}: $\sim P(x_{11})$ (from C_1 and C_3)

C_{13}: $\sim P_1(x_{21})$ (from C_{12} and C_2)

C_{14}: $P_3(x_{41}, a_1) \vee P_4(x_{42}) \vee P_3(x_{51}, x_{52}) \vee P_4(f_1(x_{53}))$ (from C_4 and C_5)

C_{15}: $\sim P_3(a_2, x_{61}) \vee\sim P_3(x_{71}, x_{72}) \vee P_5(f_2(x_{63}, x_{64}), a_2) \vee P_5(f_2(x_{73}, a_2), x_{73})$ (from C_6 and C_7)

C_{16}: $P_3(x_{41}, a_1) \vee P_4(f_1(x_{53}))$ (simplification of C_{14})

C_{17}: $\sim P_3(a_2, x_{61}) \vee P_5(f_2(a_2, a_2), a_2)$ (simplification of C_{15})

C_{18}: $P_4(f_1(x_{53})) \vee \sim P_5(x_{54}, x_{55}) \vee P_4(x_{112}) \vee \sim P_5(x_{113}, x_{114}) \vee \sim P_5(x_{113}, x_{114}) \vee P_2(a_2)$ (from C_5 and C_{11})

C_{19}: $P_4(f_1(x_{53})) \vee \sim P_5(x_{54}, x_{55}) \vee P_4(x_{112}) \vee \sim P_5(x_{113}, x_{114})$ (from C_{18} and C_{12})

C_{20}: $P_4(f_1(x_{53})) \vee \sim P_5(x_{54}, x_{55})$ (simplification of C_{19})

C_{21}: $P_4(f_1(x_{53})) \vee P_5(f_1(a_2, a_2), a_2)$ (from C_{17} and C_{16})

C_{22}: $P_4(f_1(x_{53}))$ (from C_{21} and C_{20})

C_{23}: $\sim P_3(x_{71}, x_{72}) \vee P_4(f_1(a_1)) \vee \sim P_3(a_2, x_{81}) \vee \sim P_4(f_1(x_{81}))$ (from C_7 and C_8)

C_{24}: $\sim P_3(a_2, a_1) \vee \sim P_4(f_1(a_1))$ (simplification of C_{23})

C_{25}: $\sim P_3(a_2, a_1)$ (from C_{23} and C_{24})

C_{26}: $\sim P_4(f_1(a_1)) \vee \sim P_5(x_{82}, x_{83})$ (from C_{25} and C_8)

C_{27}: $\sim P_5(x_{82}, x_{83})$ (from C_{26} and C_{22})

C_{28}: $\sim P_4(f_1(a_1)) \vee P_5(f_2(x_{73}, a_2), x_{73})$ (from C_{25} and C_7)

C_{29}: $P_5(f_2(x_{73}, a_2), x_{73})$ (from C_{28} and C_{22})

C_{30}: $\emptyset$ (from C_{29} and C_{27})

Remark 4.5: similar to Remark 3.9, from the above example, compared with binary resolution, the S-CS rule has the following features: 1) the number of clauses involved in each deduction can be more than two; 2) the number of clauses involved in the deduction is not fixed, which reflects the dynamic feature; 3) each deduction reflects the synergized effects of all the clauses involved and the S-CS rule reduces the multiple steps binary resolution significantly, so the S-CS rule could go beyond the binary resolution in terms of efficiency.

Schubert [43] presented the following problem (which came to be known as Schubert's Steamroller) as a classic puzzle to automated-deduction systems, this is a typical problem that is naturally range-restricted and includes a considerable amount of facts. This problem has been solved by different theorem provers. Here we also use this typical example to illustrate how the S-CS deduction can be used and successful to solve this problem.

Example 4.2 (Famous Steamroller Example in Automated Reasoning [43]**):**

"*Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not shads. Caterpillars and snails like to eat some plants. Therefore, there is an animal that likes to eat a grain-eating animal*".

Assume a_1, a_2, a_3, a_4, a_5, and a_6 are used to represent an individual *wolf, fox, bird, caterpillar, snail* and *grain* respectively.

Firstly use the following predicates:

$A(x)$: x is an animal	$B(x)$: x is a bird
$C(x)$: x is a caterpillar	$E(x, y)$: x likes to eat y
$F(x)$: x is a fox	$G(x)$: x is a grain
$M(x, y)$: x is much smaller than y	$P(x)$: x is a plant
$S(x)$: x is a snail	$W(x)$: x is a wolf

Now allows the premises to be expressed as below:

C_1: $W(a_1)$ C_2: $F(a_2)$ C_3: $B(a_3)$ C_4: $C(a_4)$ C_5: $S(a_5)$ C_6: $G(a_6)$

C_7: $\sim W(x) \vee A(x)$ C_8: $\sim F(x) \vee A(x)$ C_9: $\sim B(x) \vee A(x)$

C_{10}: $\sim C(x) \vee A(x)$ C_{11}: $\sim S(x) \vee A(x)$ C_{12}: $\sim G(x) \vee P(x)$

C_{13} $\sim A(x) \vee \sim P(y) \vee \sim A(z) \vee \sim P(v) \vee E(x, y) \vee \sim M(z, x) \vee \sim E(z, v) \vee E(x, z)$

C_{14}: $\sim C(x) \vee \sim B(y) \vee M(x, y)$ C_{15}: $\sim S(x) \vee \sim B(y) \vee M(x, y)$

C_{16}: $\sim B(x) \vee \sim F(y) \vee M(x, y)$ C_{17}: $\sim F(x) \vee \sim W(y) \vee M(x, y)$

C_{18}: $\sim W(x) \vee \sim F(y) \vee \sim E(x, y)$ C_{19}: $\sim W(x) \vee \sim G(y) \vee \sim E(x, y)$

C_{20}: $\sim B(x) \vee \sim C(y) \vee E(x, y)$ C_{21}: $\sim B(x) \vee \sim S(y) \vee \sim E(x, y)$

C_{22}: $\sim C(x) \vee P(h(x))$ C_{23}: $\sim C(x) \vee E(x, h(x))$

C_{24}: $\sim S(x) \vee P(i(x))$ C_{25}: $\sim S(x) \vee E(x, i(x))$

The phrase "grain-eating animal" may mean an animal that eats *some* grain. That interpretation is assumed by Pelletier [30] (also by Stickel [43]), so that the conclusion is formally interpreted as:

$\exists x \exists y[A(x) \wedge A(y) \wedge [E(x,y) \wedge \exists z[G(z) \wedge E(y,z)]]]$,

with negated clause form

C_{26}: $\sim A(x) \vee \sim A(y) \vee \sim G(z) \vee \sim E(x, y) \vee \sim E(y, z)$.

where x, y, z, and v are variables, a_1, a_2, a_3, a_4, a_5, and a_6 are Skolem constants, and h and i are Skolem functions, here the Skolem standard form follows the standard definition.

The S-CS based dynamic deduction sequence is given and illustrated as follows:

C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{11}, C_{12}, C_{13},

C_{15}, C_{16}, C_{17}, C_{18}, C_{19}, C_{21}, C_{22}, C_{24}, C_{25}, C_{26}

C_{27}: $\mathcal{C}_2^{s\theta_{27}}(C_6, C_{12})=P(a_6)$, the SC is: $G(a_6) \wedge \sim G(a_6)$

C_{28}: $\mathcal{C}_2^{s\theta_{28}}(C_{13}, C_{27})=E(x, a_6) \vee \sim M(y, x) \vee \sim E(y, z) \vee E(x, y) \vee \sim A(x) \vee \sim A(y) \vee \sim P(z)$, the SC is: $P(a_6) \wedge \sim P(a_6)$

C_{29}: $\mathcal{C}_3^{s\theta_{29}}(C_1, C_2, C_{17})=M(a_2, a_1)$, the SC is: $W(a_1) \wedge F(a_2) \wedge (\sim W(a_1) \vee \sim F(a_2))$

C_{30}: $\mathcal{C}_3^{s\theta_{30}}(C_2, C_3, C_{16})=M(a_3, a_2)$, the SC is: $(\sim F(a_2) \vee \sim B(a_3)) \wedge F(a_2) \wedge B(a_3)$

C_{31}: $\mathcal{C}_3^{s\theta_{31}}(C_3, C_5, C_{15})=M(a_5, a_3)$, the SC is: $S(a_5) \wedge (\sim B(a_3) \vee \sim S(a_5)) \wedge B(a_3)$

C_{32}: $\mathcal{C}_3^{s\theta_{32}}(C_1, C_6, C_{19})=\sim E(a_1, a_6)$, the SC is: $(\sim W(a_1) \vee \sim G(a_6)) \wedge W(a_1) \wedge G(a_6)$

C_{33}: $\mathcal{C}_3^{s\theta_{33}}(C_3, C_5, C_{21})=\sim E(a_3, a_5)$, the SC is: $B(a_3) \wedge (\sim B(a_3) \vee \sim S(a_5)) \wedge S(a_5)$

C_{34}: $\mathcal{C}_3^{s\theta_{34}}(C_{28}, C_{29}, C_{32})=E(a_1, a_2) \vee \sim A(a_1) \vee \sim E(a_1, x) \vee \sim A(a_2) \vee \sim P(x)$, the SC is: $(\sim M(a_2, a_1) \vee E(a_1, a_6)) \wedge M(f, a_1) \wedge \sim E(a_1, a_6)$

C_{35}: $\mathcal{C}_2^{s\theta_{35}}(C_1, C_7)=A(a_1)$, the SC is: $W(a_1) \wedge \sim W(a_1)$

C_{36}: $\mathcal{C}_2^{s\theta_{36}}(C_3, C_9)=A(a_3)$, the SC is: $B(a_3) \wedge \sim B(a_3)$

C_{37}: $\mathcal{C}_3^{s\theta_{37}}(C_{13}, C_{30}, C_{36})=\sim E(a_3, x) \vee E(a_2, a_3) \vee \sim A(a_2) \vee E(a_2, y) \vee \sim P(y) \vee \sim P(x)$, the SC is: $A(a_3) \wedge (\sim M(a_3, a_2) \vee \sim A(a_3)) \wedge M(a_3, a_2)$

C_{38}: $\mathcal{C}_3^{s\theta_{38}}(C_{13}, C_{31}, C_{36})=\sim E(a_5, x) \vee E(a_3, a_5) \vee E(a_3, y) \vee \sim A(a_5) \vee \sim P(y) \vee \sim P(x)$, the SC is: $(\sim M(a_5, a_3) \vee \sim A(a_3)) \wedge A(a_3) \wedge M(a_5, a_3)$

C_{39}: $\mathcal{C}_3^{s\theta_{39}}(C_1, C_2, C_{18})=\sim E(a_1, a_2)$, the SC is: $(\sim F(a_2) \vee \sim W(a_1)) \wedge F(a_2) \wedge W(a_1)$

C_{40}: $\mathcal{C}_3^{s\theta_{40}}(C_{34}, C_{35}, C_{39})$=~$A(a_2)$∨~$E(a_2, x)$∨~$P(x)$, the SC is: (~$A(a_1)$∨$E(a_1, a_2)$)∧~$E(a_1, a_2)$∧$A(a_1$

C_{41}: $\mathcal{C}_5^{s\theta_{41}}(C_4, C_{22}, C_{24}, C_{37}, C_{40})$=~$A(a_2)$ ∨$E(a_2, a_3)$ ∨~$E(a_3, i(x))$ ∨~$S(x)$, the SC is: (~$P(h(a_4))$ ∨~$E(a_2, h(a_4))$)∧(~$C(a_4)$ ∨$P(h(a_4))$)∧($E(a_2, h(a_4))$ ∨~$P(i(x))$)∧$P(i(x))$∧$C(a_4)$

C_{42}: $\mathcal{C}_2^{s\theta_{42}}(C_5, C_{11})$=$A(a_5)$, the SC is: $S(a_5)$∧~$S(a_5)$

C_{43}: $\mathcal{C}_3^{s\theta_{43}}(C_{28}, C_{31}, C_{42})$=$E(a_3, a_5)$ ∨$E(a_3, a_6)$ ∨~$P(x)$ ∨~$E(a_5, x)$ ∨~$A(a_3)$, the SC is: (~$A(a_5)$ ∨~$M(a_5, a_3)$)∧$A(a_5)$∧$M(a_5, a_3)$

C_{44}: $\mathcal{C}_4^{s\theta_{44}}(C_5, C_{25}, C_{33}, C_{43})$=~$P(i(a_5))$ ∨$E(a_3, a_6)$ ∨~$A(a_3)$, the SC is: (~$E(a_5, i(a_5))$ ∨$E(a_3, a_5)$)∧($E(a_5, i(a_5))$ ∨~$S(a_5)$)∧~$E(a_3, a_5)$∧$S(a_5)$

C_{45}: $\mathcal{C}_2^{s\theta_{45}}(C_5, C_{21})$=~$B(x)$ ∨~$E(x, a_5)$, the SC is: ~$S(a_5)$∧$S(a_5)$

C_{46}: $\mathcal{C}_3^{s\theta_{46}}(C_3, C_{38}, C_{45})$=~$E(a_5, x)$ ∨$E(a_3, y)$ ∨~$A(a_5)$ ∨~$P(y)$ ∨~$P(x)$, the SC is: (~$E(a_3, a_5)$ ∨~$B(a_3)$)∧$B(a_3)$∧$E(a_3, a_5)$

C_{47}: $\mathcal{C}_4^{s\theta_{47}}(C_5, C_6, C_{26}, C_{41})$=~$A(a_2)$ ∨~$E(a_3, i(a_5))$ ∨~$A(a_3)$ ∨~$E(a_3, a_6)$, the SC is: (~$E(a_2, a_3)$ ∨~$G(a_6)$)∧($E(a_2, a_3)$ ∨~$S(a_5)$)∧$S(a_5)$∧$G(a_6)$

C_{48}: $\mathcal{C}_3^{s\theta_{48}}(C_{24}, C_{36}, C_{44})$=~$S(a_5)$ ∨$E(a_3, a_6)$, the SC is: $A(a_3)$∧(~$P(i(a_5))$ ∨~$A(a_3)$) ∧$P(i(a_5))$

C_{49}: $\mathcal{C}_4^{s\theta_{49}}(C_5, C_{36}, C_{47}, C_{48})$=~$A(a_2)$ ∨~$E(a_3, i(a_5))$, the SC is: (~$E(a_3, a_6)$ ∨~$A(b)$)∧($E(a_3, a_6)$ ∨~$S(a_5)$)∧$A(a_3)$∧$S(a_5)$

C_{50}: $\mathcal{C}_6^{s\theta_{50}}(C_2, C_5, C_8, C_{24}, C_{46}, C_{49})$=~$P(x)$ ∨~$A(a_5)$ ∨~$E(a_5, x)$, the SC is: ($E(a_3, i(a_5))$ ∨~$P(i(a_5))$∧(~$A(a_2)$ ∨~$E(a_3, i(a_5))$)∧(~$S(a_5)$ ∨$P(i(a_5))$)∧(~$F(a_2)$ ∨$A(a_2)$)∧$S(a_5)$∧$F(a_2)$

C_{51}: $\mathcal{C}_4^{s\theta_{51}}(C_5, C_{24}, C_{42}, C_{50})$=~$E(a_5, i(a_5))$, the SC is: (~$A(a_5)$ ∨~$P(i(a_5))$)∧($P(i(a_5))$ ∨~$S(a_5)$)∧$A(a_5)$∧$S(a_5)$

C_{52}: $\mathcal{C}_3^{s\theta_{52}}(C_5, C_{25}, C_{51})$=∅, the SC is: ($E(a_5, i(a_5))$ ∨~$S(a_5)$)∧$S(a_5)$∧~$E(a_5, i(a_5))$.

Remark 4.6: Note that the above given deduction sequence is only for illustration purpose as one possible deduction. Due to the dynamic nature or flexibility in selecting the number of clauses involved in each S-CS based deduction step, there could be various deduction sequences with much less steps to reach to the solution, but will not be addressed in this paper.

Now that standard contradiction and quasi-contradiction are not equivalent concepts in first-order logic case, below we will introduce some results based on quasi-contradiction.

Definition 4.4 Suppose a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in first-order logic. $\Phi_1, \Phi_2,\ldots, \Phi_t$ is called *a quasi-contradiction separation based dynamic deduction sequence* (*or an Q-CS based dynamic deduction sequence* from S to a clause Φ_t, denoted as $\mathscr{D}^q$, if

(1) $\Phi_i \in S$, $i=1, 2, \ldots, t$; or

(2) there exist $r_1, r_2, \ldots, r_{k_i} < i$, $\Phi_i = \mathscr{C}_{r_{k_i}}^{q\theta_i}(\Phi_{r_1}, \Phi_{r_2}, \ldots, \Phi_{r_{k_i}})$, where $\theta_i = \bigcup_{j=1}^{k_i} \sigma_j$, σ_j is a substitution to Φ_{r_j}, $j=1,\ldots, k_i$.

Definition 4.5 Suppose a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in first-order logic. If there are two clauses C_{i_0} and C_{j_0} in S such that $C_{i_0} = P \vee C_{i_0}^*$, $C_{j_0} = \sim P \vee C_{j_0}^*$, where P and $\sim P$ is a complementary pair of literals, $C_{i_0}^* \neq \emptyset$, and $C_{j_0}^* \neq \emptyset$. Then it is said that S *satisfies the complementary condition*.

Lemma 4.2 Suppose D_k, $D_{k-1},\ldots, D_1$ are k clauses in first-order logic. Assume that a substitution σ_i is applied to D_i (σ_i could be an empty substitution) for $i=k, k-1,\ldots, 1$, and the same literals merged after substitution, such that $\bigwedge_{i=k}^{1} D_i^{\sigma_i}$ is a standard contradiction.

The following statements hold:

(1) If there exists some complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, then there exist $k-2$ clauses among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$ which are all redundant clauses.

(2) If there does not exist any complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, then $\{D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}\}$ satisfies the complementary condition, i.e., there exists a complementary pair of literals P and $\sim P$, and there also exist $D_i^{\sigma_i}$ and $D_j^{\sigma_j}$ such that: $D_i^{\sigma_i} = D_{i_0}^{\sigma_i} \vee P$, $D_{i_0}^{\sigma_i} \neq \emptyset$, and $D_j^{\sigma_j} = D_{j_0}^{\sigma_j} \vee \sim P$, $D_{j_0}^{\sigma_j} \neq \emptyset$.

Proof. See the detailed proof in Appendix.

Lemma 4.3 (Soundness Lemma of the S-CS Based Dynamic Deduction in First-Order Logic) Let D_k, D_{k-1},..., D_1 be k clauses in first order logic. Assume that a substitution σ_i is applied to D_i (σ_i could be an empty substitution) for i=k, k–1…, 1, and the same literals merged after substitution. Suppose $D_i^{\sigma_i}$ is partitioned into two sub-clauses $D_i^{\sigma_i-}$ and $D_i^{\sigma_i+}$ such that

i) $D_i^{\sigma_i}=D_i^{\sigma_i-}\vee D_i^{\sigma_i+}$, where $D_i^{\sigma_i-}$ and $D_i^{\sigma_i+}$ have no common literals;

ii) $D_i^{\sigma_i+}$ can be an empty clause, but $D_i^{\sigma_i-}$ cannot be an empty clause;

ii) $\wedge_{i=k}^{1} D_i^{\sigma_i-}$ is a standard contradiction.

Then we have

$$D_k\wedge D_{k-1}\wedge\ldots\wedge D_1 \le D_k^{\sigma_k+}\vee D_{k-1}^{\sigma_{k-1}+}\vee\ldots\vee D_1^{\sigma_1+}, \text{ i.e., } D_k\wedge D_{k-1}\wedge\ldots\wedge D_1\le \mathcal{C}_k^{s\sigma}(D_k, D_{k-1},\ldots, D_1),$$

where $\sigma=\cup_{i=k}^{1}\sigma_i$, σ_i is a substitution to D_i, i= k, k–1…, 1.

Proof. See the detailed proof in Appendix.

Theorem 4.1 (**Soundness Theorem of the S-CS Based Dynamic Deduction in First-Order Logic**) Suppose a clause set S ={C_1, C_2,…, C_m} in first-order logic. Φ_1, Φ_2,…, Φ_t is an S-CS based dynamic deduction from S to a clause Φ_t. If Φ_t is an empty clause, then S is unsatisfiable.

Proof. It follows from the definition of a S-CS based dynamic deduction (Definition 4.3) and also the Soundness Lemma of an S-CS based dynamic deduction (Lemma 4.3) that

$$C_1\wedge C_2\wedge\ldots\wedge C_m\le\Phi_1\wedge\Phi\wedge\ldots\wedge\Phi_t\le\Phi_t.$$

This concludes the proof.

Remark 4.7: in general, soundness of the Q-CS based dynamic deduction in first-order logic does not hold. For example,

Example 4.3 Let C_1=$P(x)\vee Q(x)$, C_2=~$P(f(x))\vee R(x)$. If we obtain the deduction result $Q(x)\vee R(x)$ by directly deleting the quasi-contradiction $P(x)\wedge$~$P(f(x))$, then $Q(x)\vee R(x)$ is not a logical consequence of C_1 and C_2. Actually, suppose that the Herbrand field (denoted by H) [8] is given as {a, $f(a)$, $f(f(a))$, $f(f(f(a)))$,…}, and an interpretation I_0 is given as follows (↦ means the true-value assignment, 0 mean *false* and 1 means *true*):

P: $f(a)$ ↦ 0, where all the other elements in H: ↦ 1;

~P: $f(a) \mapsto 1$, where all the other elements in H: $\mapsto 0$;

Q: $f(a) \mapsto 1$, where all the other elements in H: $\mapsto 0$;

R: $a \mapsto 0$, where all the other elements in H: $\mapsto 1$.

Then $I_0(C_1)=1$, $I_0(C_2)=1$, but $I_0(Q(x)\vee R(x))=0$. Therefore, the Q-CS based deduction does not hold for soundness in general, due to the fact that the Q-CS based deduction cannot guarantee the result from each deduction is the logical consequents from all the clauses used.

In the following, reference to the construction procedure of the Lifting Lemma about binary resolution deduction, we establish the following lemma.

Lemma 4.4 (**Lifting Lemma of the S-CS Based Dynamic Deduction in First-Order Logic**) In first-order logic, let $C_1, C_2, \ldots, C_m$ be clauses without common variables, C_i^0 an instance of C_i, $i = 1, 2, \ldots, m$. If Ω_0 ($=\mathcal{C}_m^{q\mu}(C_1^0, C_2^0, \ldots, C_m^0)$) is an S-CSC of $C_1^0, C_2^0, \ldots, C_m^0$, then there exists an S-CSC Ω($=\mathcal{C}_m^{q\theta}(C_1, C_2, \ldots, C_m)$) of $C_1, C_2, \ldots, C_m$ such that Ω_0 is an instance of Ω, i.e., the following transformation diagram Fig. 4.1 holds, where μ and θ are the substitutions applied to $C_1^0, \ldots, C_m^0$ and $C_1, \ldots, C_m$ respectively when constructing the S-CSC.

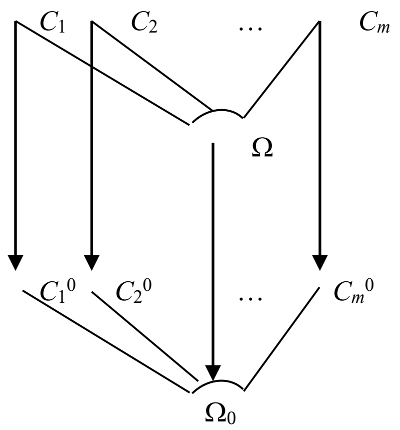

Fig. 4.1 The transformation diagram

*Proof. S*ee the detailed proof in Appendix.

Based on the Herbrand Theorem II [8] and the above Lifting Lemma, we have the following completeness theorem:

Theorem 4.2 (**Completeness of the S-CS Based Dynamic Deduction in First-Order Logic**) Suppose a clause set $S=\{C_1, C_2, \ldots, C_m\}$ in first-order logic. If S is unsatisfiable, then there exists an S-CS based dynamic deduction from S to an empty clause.

Proof. In fact, according to Herbrand Theorem II [8], if S is unsatisfiable, then there exists at least a ground instance S^{σ} of S such that S^{σ} is unsatisfiable. According to the completeness theorem of the CS-based dynamic deduction in propositional logic (Theorem 3.2), there exists a CS based dynamic deduction sequence $\mathscr{D}_0$ from S^{σ} to an empty clause. Moreover, we can lift $\mathscr{D}_0$ to an S-CS based dynamic deduction sequence $\mathscr{D}$ from S to an empty clause by using the above Lifting Lemma (Lemma 4.4).

Theorem 4.3 (**Completeness of the Q-CS Based Dynamic Deduction in First-Order Logic**) Suppose a clause set $S = \{C_1, C_2, \ldots, C_m\}$ in first-order logic. If S is unsatisfiable, then there exists a Q-CS based dynamic deduction from S to an empty clause.

Proof. It only needs to note that a standard contradiction is a quasi-contradiction (Lemma 4.1). Then it follows from Theorem 4.2 that the conclusion holds.

Remark 4.8: if the S-CS rule only involves two clauses in Propositional Logic or in First-Order Logic, then the S-CS rule is reduced to binary resolution rule in Propositional Logic or in First-Order Logic respectively. Therefore, alternatively, the above completeness in the first-order case simply follows from the fact that the S-CS rule simulates (the complete) binary resolution.

5. Graphical Illustration of the Key Ideas

This section provides a graphical and intuitive illustration on the essential features of the CS-based dynamic deduction, as well as the essential difference from binary resolution deduction. We use the funnel as an intuitive figure to show the automated deduction process from the input clause set. The one coming out from the exit of the funnel is the final output. Fig. 5.1 and Fig. 5.2 below show a graphical funnel view comparison between the binary resolution deduction process and the CS-based dynamic deduction process.

Fig. 5.1 actually also illustrates some insights why the pre-processing and simplification steps are essential in the binary resolution deduction, even take the majority of the steps and time, and also why lots of work have been focused on splitting and simplifying the clause set into the simpler ones just because the exit is too narrow. Fig. 5.2 illustrates the dynamic and flexible nature of the CS-based dynamic deduction, which essentially opens multiple paths by which the outcome may be discovered.

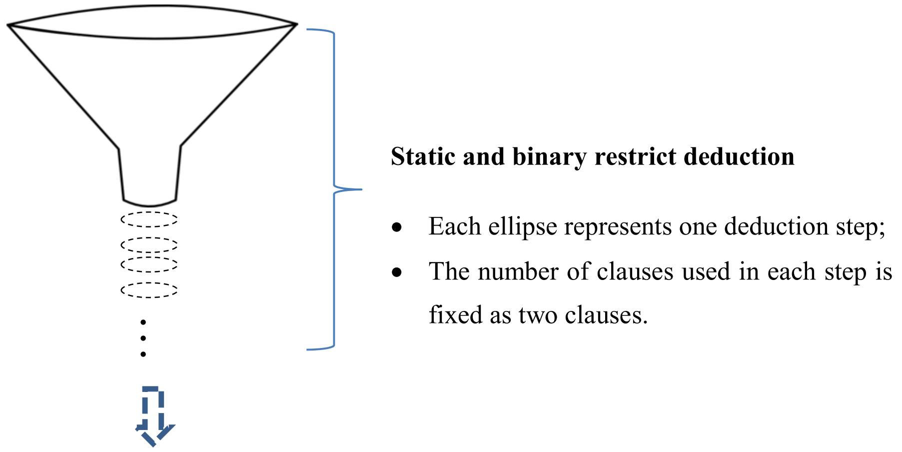

Fig. 5.1 The graphical funnel view of binary resolution deduction process

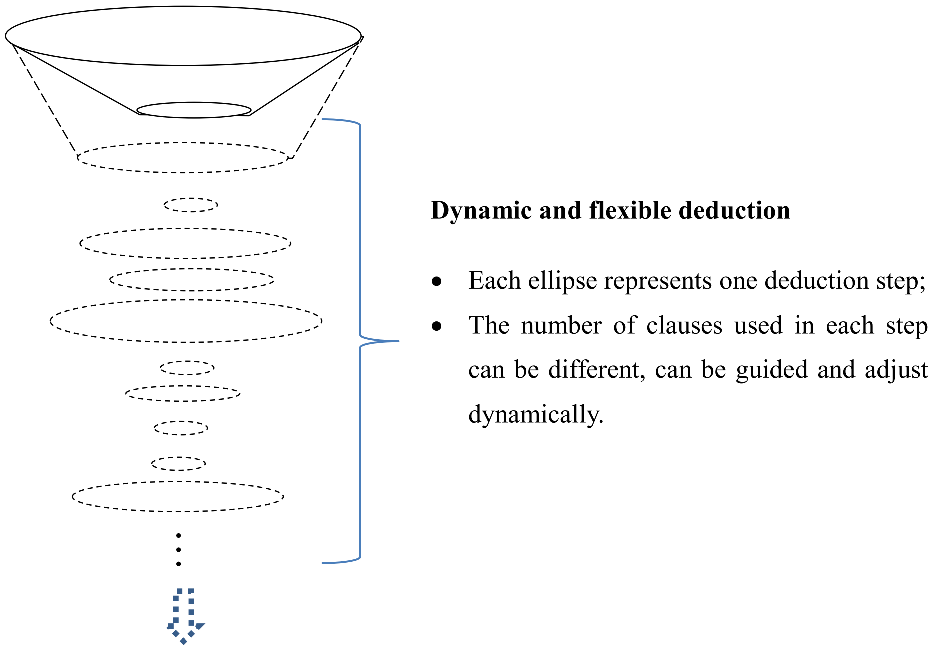

Fig. 5.2 The graphical funnel view of CS-based dynamic deduction process

6. Conclusions and Future Works

After extending the term "contradiction" from the traditionally defined a complementary pair based on two clauses into a typical unsatisfiable clause set (i.e., a standard contradiction consists of more than two clauses), this paper proposed a new inference principle and set up its sound and complete theory framework to extend the existing static (i.e., fixed) binary resolution into a dynamic multi-clause synergized contradiction separation based inference rule. Three terms "dynamic", "multi-clauses" and "synergized" reflected the key motivations of the present work. The key contribution of the present paper was then focused on justifying theoretically that this CS-based dynamic deduction is sound and complete, along with some example illustrations of those key features.

From the computational point of view, the term "dynamic" reflects non-determinism in terms of which clauses and how many of them involved in each deduction (binary resolution is a special case in terms of which two clauses involved). This non-determinism is different from its more familiar deterministic counterpart in its ability to arrive at outcomes using various routes. Nondeterminism is especially beneficial for those problems when there is a single outcome with multiple paths by which the outcome may be discovered, this is the common case in mathematical theorem proving. Compared with binary resolution, the proposed CS deduction offers more chances or new windows of algorithms and implementations development for proof search.

This established CS-based automated deduction theory is just a first step towards the development of a proof search procedure that could be implemented as an effective CS-based theorem prover. Practical implementation of the CS-based automated deduction further hinges on specific algorithms and strategies making the "right" single CS step including the "suitable selection" of the number of clauses to be involved in each deduction process useful for proof search. These algorithms and strategies (including indexing techniques) could vary in quality and efficiency. This kind of CS-based proof search algorithms or strategies and implementation will be still one of the challenging problems in this topic.

Although it is challenging, it does not mean it is impossible. The good news however is that some concrete search algorithms and strategies (such as so-called *standard triangle-type contradiction separation based deduction algorithm*) to support the theory and achieve the implementation for automation with detailed experiments and case studies have been also established by the author team, along with their corresponding automated reasoning systems (called MC-SCS), which is reported in a subsequent work [50]. It has been shown from a big amount of experimental testing using the benchmark problem in TPTP [44] or Mizar [20], that it is possible and feasible to dynamically, automatically, and efficiently select multiple clauses to be involved in deduction according to the deduction process, this escapes from the two clause restriction, enhances the deduction flexibility, increases the concurrent behaviour and synergistic effect among the clauses involved, therefore, could improves the overall capability and efficiency of automated deduction.

This present work mainly placed the theoretical foundation of CS-based automated deduction. The CS-based proof search algorithms or strategies is beyond the scope of this paper, but is a crucial direction for future work.

Another essential direction for further development would be the extension of CS-based automated deduction with ability to handle the equalities, since equality is a very common and important relation for applications. This, certainly, is the next step work for our research team working toward to incorporate the superposition calculus into the CS-based deduction.

Finally, it is worth noting that: it is known that the resolution principle extended the MP rule in the classical logic. In this proposed work, the contradiction separation inference rule has generalized the resolution principle. Therefore, if the MP rule in the classical logical system is replaced and generalized by the contradiction separation inference rule, it is expected that classical logical system can be generalized into a new generic logical system.

Future plans include extensive and deeper experimental studies and comparative analysis with the state of art based on the benchmark problem; new and better CS-based search algorithms and strategies, forward and backward deduction, complexity analysis as well as the real application etc.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (Grant No.61673320) and the Fundamental Research Funds for the Central Universities (Grant No.2682017ZT12, 2682016 CX119).

REFERENCES

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

[2] M.P. Bonacina, Automated reasoning for explainable artificial intelligence, The First International ARCADE (Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements) Workshop (in association with CADE-26), Gothenburg, Sweden, August 6, 2017.

[3] M.P. Bonacina and D. A. Plaisted, Semantically-guided goal-sensitive reasoning: inference system and completeness. Journal of Automated Reasoning, 59(2)(2017) 165-218.

[4] R.S. Boyer. Locking: A Restriction of Resolution. Doctoral Dissertation, University of Texas at Austin, 1971.

[5] J.S. Bruno and W. Paleo, Conflict resolution: a first-order resolution calculus with decision literals and conflict-driven clause learning. Journal of Automated Reasoning, 58 (1)(2017) 1–24.

[6] J. Buresh-Oppenheim and T. Pitassi, The complexity of resolution refinements. Proc. of the 18th IEEE Sym. on LICS, pp. 138–147, 2003.

[7] Z.Y. Cai, Q. Huang, Q.Y. Huang, and D.R. Huang, Set-support strategy based resolution inference method: implementation and optimization. Modern Computer, 5(2005) 74-76 (in Chinese).

[8] C.L. Chang and R.C.T. Lee. Symbolic Logic and Mechanical Theorem Proving. 1973, Academic Press.

[9] A. Degtyarev, R. Nieuwenhuis, and A. Voronkovc, Stratified resolution, Journal of Symbolic Computation, 36(2003) 79–99.

[10] M.R. Fellows, S. Szeider, G. Wrightson. On finding short resolution refutations and small unsatisfiable subsets. Theoretical Computer Science, 351(3)(2006) 351-359.

[11] J. Gozny and B.W. Paleo, Towards the compression of first-order resolution proofs by lowering unit clauses, Automated Deduction – CADE-25, Berlin, Germany, August 1-7, 2015, pp. 356-366.

[12] J. Harrison, Handbook of Practical Logic and Automated Reasoning, Cambridge University Press, 2009.

[13] M.C. Harrison and N. Rubin, Another generalization of resolution, Journal of the Association for Computing Machinery, 25(3)(1978) 341-351.

[14] M. Heule, M. Jarvisalo, F. Lonsing, M. Seidl, A. Biere, Clause elimination for SAT and QSAT, Journal of Artificial Intelligence Research, 53(2015) 127-168.

[15] K. Hoder, G. Reger, M. Suda, and A. Voronkov, Selecting the selection. The 8th International Joint Conference on Automated Reasoning – IJCAR 2016, LNAI 9706, Coimbra, Portugal, June 27-July 2, 2016, pp. 313-329.

[16] K. Hoder, A. Voronkov, The 481 ways to split a clause and deal with propositional variables. In: MP. Bonacina (ed.) CADE 2013. LNCS, vol. 7898, pp. 450–464. Springer, Heidelberg (2013).

[17] D. Itegulov, J. S. Bruno, and W. Paleo, Scavenger 0.1: a theorem prover based on conflict resolution. International Conference on Automated Deduction CADE 2017: Automated Deduction – CADE 26, pp. 344-356, 2017.

[18] M. Jaarvisalo, A. Biere, and M. Heule. Blocked clause elimination. In Javier Esparza and Rupak Majumdar, editors, Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010), volume 6015 of Lecture Notes in Computer Science, pages 129–144. Springer, 2010.

[19] C. Kaliszyk and J. Urban, MizAR 40 for Mizar 40. CoRR, abs/1310.2805 (2013).

[20] C. Kaliszyk, S. Schulz, J. Urban, and J. Vyskočil, System Description: E.T. 0.1, Proceedings of the 25th International Conference on Automated Deduction (CADE-25), Lecture Notes in Artificial Intelligence 9195, Berlin, Germany, August 1-7, 2015, pp. 389 – 398.

[21] B. Kiesl, M. Seidl, H.Tompits, and A. Biere, Super-blocked clauses, Proceeding of the 8th International Joint Conference on Automated Reasoning – IJCAR 2016, LNAI 9706, June 27-July 2, 2016, pp. 45-61.

[22] B. Kiesl and M. Suda, A unifying principle for clause elimination in first-order logic. International Conference on Automated Deduction CADE 2017: Automated Deduction – CADE 26, pp. 274-290, 2017.

[23] O. Kullmann, On a generalization of extended resolution, Discrete Applied Mathematics, 96–97(1999) 149–176.

[24] X.H. Liu, A new semantic resolution principle, Journal of Jilin University, 2(1978) 112-117 (in Chinese).

[25] X.H. Liu, Deletion strategies about resolution, Journal of Jilin University, 2(1982) 97-106 (in Chinese).

[26] J. McCharen, R. Overbeek, and L. Wos. Complexity and related enhancements for automated theorem-proving programs. Computers and Mathematics with Applications, 2(1976) 1–16.

[27] W.W. McCune, Prove9 and Mae4 (2005-2010). http://www.cs.unm.edu/mccune/prover9/. Accessed 27 April 2016.

[28] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving, in Book: Handbook of Automated Reasoning, pp. 371-443, 2001.

[29] J. Nordström, A simplified way of proving trade-off results for resolution, Information Processing Letters 109 (2009) 1030–1035.

[30] F.J. Pelletier, Completely non-clausal, completely heuristically driven automatic theorem proving, MSc Thesis, University of Alberta, Edmonton, Alberta, Canada, 1982.

[31] D. Plaisted, History and prospects for first-order automated deduction, Automated Deduction – CADE-25, Berlin, Germany, August 1-7, 2015, pp. 3-28.

[32] D. Plaisted, The search efficiency of theorem proving strategies. Automated Deduction - CADE-12. LNCS, Vol. 814. Springer, Heidelberg (1994).

[33] A. Riazanov and A. Voronkov, Limited resource strategy in resolution theorem, Journal of Symbolic Computation 36 (2003) 101–115.

[34] J.A. Robinson, A machine oriented logic based on the resolution principle, J. ACM, 12(1)(1965) 23-41.

[35] J.A. Robinson, Automatic deduction with hyper-resolution. International Journal of Computer Mathematics, 1(1965) 227-234.

[36] J.A. Robinson, The generalized resolution principle, Journal of Symbolic Computation, 3(1968) 135-151.

[37] J.A. Robinson and A. Voronkov, Handbook of Automated Reasoning, Vols. 1 and 2, the MIT Press and North Holland, 2001.

[38] S. Schulz, Learning search control knowledge for equational theorem proving, Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (KI-2001), volume 2174 of LNAI, 2001, pp. 320-334.

[39] S. Schulz. System description: E 1.8. In Proc. of the 19th LPAR, 2013, Stellenbosch, volume 8312 of LNCS. Springer.

[40] S. Schulz and M. Mohrmann, Performance of clause selection heuristics for saturation-based theorem proving, The 8th International Joint Conference on Automated Reasoning – IJCAR 2016, LNAI 9706, Coimbra, Portugal, June 27-July 2, 2016, pp. 330-345.

[41] S. Schulz, G. Sutcliffe, J. Urban, and A. Peas, Detecting inconsistencies in large first-order knowledge bases. International Conference on Automated Deduction CADE 2017: Automated Deduction – CADE 26, pp. 310-325, 2017.

[42] V.D. Silva and C. Urban, Abstract interpretation as automated deduction. Journal of Automated Reasoning. 58 (3)(2017) 363–390.

[43] M.E. Stickel, Schubert's Steamroller problem: formulations and solutions, Journal of Automated Reasoning, 2(1986) 89-101.

[44] G. Sutcliffe. The TPTP problem library and associated infrastructure: the FOF and CNF parts, v3.5.0. Journal of AR, 43(4)(2009) 337–362.

[45] U. Waldmann, Superposition. In: Paleo, B.W. (eds.) Towards an Encyclopaedia of Proof Systems, pp. 29. College Publications, London, UK, 2017.

[46] U. Waldmann, Saturation with redundancy. In: Paleo, B.W. (eds.) Towards an Encyclopaedia of Proof Systems, pp. 30. College Publications, London, UK, 2017.

[47] L. Wos, G.A. Robinson, and D.F. Cason, Efficiency and completeness of the set of support strategy in theorem proving, Journal of the ACM, 12(4)(1965) 536-541.

[48] Y. Xu, J. Liu, X.M. Zhong, and S.W. Chen, Multi-ary α-resolution principle for a lattice-valued logic, IEEE Trans. on Fuzzy Systems, 21(5) (2013) 898-912.

[49] Y. Xu, J. Liu, S.W. Chen, and X.M. Zhong, A novel generalization of resolution principle for automated deduction, The 12th International FLINS Conference on Uncertainty Modelling in Knowledge Engineering and Decision Making (FLINS2016), August 24-26, 2016, ENSAIT, Roubaix, France.

[50] J. Zhong, Y. Xu, J. Liu, X.X. He, Q.H. Liu, S.W. Chen, X.M. Zhong, G.F. Wu, X.R. Ning, X.D. Guan, Q.S. Chen, and Z.M. Song, Multi-clause synergized contradiction separation based first-order theorem prover – MC-SCS, Proceeding of the 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE2017), Nanjing, China, November 24-26, 2017, pp. 453-458.

Appendix:

This appendix section provides the detailed proofs for the relevant lemmas and theorems in the paper.

Lemma 3.1 Assume a clause set $S=\{C_1, C_2,\ldots, C_m\}$ in propositional logic. Then S is a standard contradiction if and only if S is a quasi-contradiction.

Proof. It needs to prove that S is unsatisfiable if and only if $\forall\ (x_1,\ldots, x_m)\in \prod_{i=1}^{m} C_i$, there exists at least one complementary pair among $\{x_1,\ldots, x_m\}$.

It follows from the distributive law between the disjunction and conjunction that

$$S = \wedge_{i=1}^{m} C_i = \vee_{(x_{1,\ldots,}x_m)\in\prod_{i=1}^{m} C_i}(x_1\wedge \ldots \wedge x_m).$$

Hence, S is unsatisfiable if and only if $\forall(x_1,\ldots, x_m)\in\prod_{i=1}^{m} C_i$, $x_1\wedge\ldots\wedge x_m$ is unsatisfiable if and only if $\forall(x_1,\ldots, x_m)\in\prod_{i=1}^{m} C_i$, there exists at least one complementary pair among $\{x_1,\ldots, x_m\}$.

Lemma 3.2 In propositional logic, a clause set $S=C_1\wedge C_2$ is unsatisfiable if and only if C_1 and C_2 are single-literal clauses and $C_1={\sim}C_2$.

Proof. ($\Rightarrow$) If C_1 or C_2 has more than one literal, we can assume $x_1, x_2\in C_1$, $x_1\neq x_2$, and $y\in C_2$, then it follows from the commutativity of the conjunction that no complementary pair exists in either $\{x_1, y\}$ or $\{x_2, y\}$. By Lemma 3.1, it is a contradiction to the fact that $S=C_1\wedge C_2$ is unsatisfiable. If C_1 and C_2 are all single-literal clauses but $C_1\neq{\sim}C_2$, it is also a contradiction to the fact that $S=C_1\wedge C_2$ is unsatisfiable.

($\Leftarrow$) Obviously.

Lemma 3.3 (Soundness Lemma of the CS-Based Dynamic Deduction in Propositional Logic) Suppose $D_k, D_{k-1},\ldots, D_1$ are k clauses in propositional logic, where $D_i=D_i^{+}\vee D_i^{-}$, $i=1,\ldots, k$.

If $\wedge_{i=1}^{k} D_i^{-}$ is unsatisfiable, then $D_k\wedge D_{k-1}\wedge\ldots\wedge D_1 \leq D_k^{+}\vee D_{k-1}^{+}\vee\ldots\vee D_1^{+}$, that is,

$$D_k\wedge D_{k-1}\wedge\ldots\wedge D_1 \leq \mathcal{C}_k\ (D_k, D_{k-1},\ldots, D_1).$$

Proof. For arbitrary a valuation I in propositional logic, suppose

$$I([D_k^{+}\vee D_k^{-}]\wedge[\ D_{k-1}^{+}\vee D_{k-1}^{-}]\wedge\ldots\wedge[\ D_1^{+}\vee D_1^{-}])=I(D_k\wedge D_{k-1}\wedge\ldots\wedge D_1)=1.$$

If $I(D_k^{+}\vee D_{k-1}^{+}\vee\ldots\vee D_1^{+})=0$, now that $I(D_i^{+})\leq I(D_k^{+}\vee D_{k-1}^{+}\vee\ldots\vee D_1^{+})$, then

$$I(D_i^{+})=0,\ i=1,\ldots, k-1, k.$$

Therefore, we have

$1=I(D_k \wedge D_{k-1} \wedge \ldots \wedge D_1)$

$=I([D_k^+ \vee D_k^-] \wedge [D_{k-1}^+ \vee D_{k-1}^-] \wedge \ldots \wedge [D_1^+ \vee D_1^-])$

$=[I(D_k^+) \vee I(D_k^-)] \wedge [I(D_{k-1}^+) \vee I(D_{k-1}^-)] \wedge \ldots \wedge [I(D_1^+) \vee I(D_1^-)]$

$=I(D_k^-) \wedge I(D_{k-1}^-) \wedge \ldots \wedge I(D_1^-)$

$=I(D_k^- \wedge D_{k-1}^- \wedge \ldots \wedge D_1^-)$.

This, however, is contradictory to the assumption that $\bigwedge_{i=1}^{k} D_i^-$ is unsatisfiable. Hence, for arbitrary a valuation I in propositional logic,

if $I(D_k \wedge D_{k-1} \wedge \ldots \wedge D_1)=1$, then $I(D_k^+ \vee D_{k-1}^+ \vee \ldots \vee D_1^+)=1$,

that is, $D_k \wedge D_{k-1} \wedge \ldots \wedge D_1 \leq D_k^+ \vee D_{k-1}^+ \vee \ldots \vee D_1^+$. This concludes the proof.

Theorem 3.2 (**Completeness Theorem of the CS-Based Dynamic Deduction in Propositional Logic**) Suppose a clause set $S=\{C_1, C_2, \ldots, C_m\}$ in propositional logic. If S is unsatisfiable, then there exists a CS based deduction sequence from S to an empty clause.

Proof. (1) If S includes only one clause C. Now that S is unsatisfiable, the result holds obviously because C will be an empty clause.

(2) If S includes more than one clause. For any $i=1, 2, \ldots, m$, let $|C_i|$ be the number of all literals occurring in C_i. Suppose $K(S)$ represents the difference between the total number of literals occurring in S and the number of clauses in S, i.e., $K(S)=\sum_{i=1}^{m}|C_i|-m$. Hence, we have the following two cases:

Case 1: If $K(S)=0$, then S is composed of unit clauses, i.e., each clause in S includes only one literal. Since S is unsatisfiable, it follows that the unit clause set $\{C_1, C_2, \ldots, C_m\}$ includes some complementary pairs. Without loss of generality, suppose a complementary pair is $C_1=p$, $C_2=\sim p$, therefore, $\{C_1=C_1^-, C_2=C_2^-\}$ forms a contradiction, and $\mathcal{C}_2(C_1, C_2)=\varnothing$. It means the result holds.

Case 2: Suppose the result holds for $K(S)<n$ $(n>0)$. Now we need to prove the result also holds for $K(S)=n$.

Let $K(S)=n$. Then S has at least one non-unit clause. Suppose g is a literal occurring in a non-unit clause of S. Let $C_i=C_i^* \vee g$, where C_i^* is not an empty clause. Now we have

$S=C_1 \wedge \ldots \wedge C_{i-1} \wedge C_i \wedge C_{i+1} \wedge \ldots \wedge C_m$

$=(C_1 \wedge \ldots \wedge C_{i-1} \wedge C_i^* \wedge C_{i+1} \wedge \ldots \wedge C_m) \vee (C_1 \wedge \ldots \wedge C_{i-1} \wedge g \wedge C_{i+1} \wedge \ldots \wedge C_m)$.

Suppose

$S_1 = C_1 \wedge\ldots\wedge C_{i-1} \wedge C_i^* \wedge C_{i+1} \wedge\ldots\wedge C_m$; and

$S_2 = C_1 \wedge\ldots\wedge C_{i-1} \wedge g \wedge C_{i+1} \wedge\ldots\wedge C_m$.

Obviously, now that $S_1 \leq S$ and S is unsatisfiable, we have S_1 is unsatisfiable and $K(S_1) < n$. According to the induction hypothesis, there exists a CS based deduction sequence $\mathscr{D}_1^*$ from S to an empty clause.

Replacing all C_i^* occurring in $\mathscr{D}_1^*$ with C_i and modifying the corresponding contradiction separation clauses, we can obtain a deduction sequence $\mathscr{D}_1$. In fact, $\mathscr{D}_1$ is a CS based deduction sequence from S to an empty clause or g. Note that for the C_i^- applied in order to get the contradiction separation clauses $\mathscr{C}_m(\ldots, C_i,\ldots)$ involving C_i as well as the C_i^{*-} applied in order to get the contradiction separation clauses $\mathscr{C}_m(\ldots, C_i^*,\ldots)$ involving C_i^*, there will be two cases: 1) $C_i^{*-} = C_i^-$; or 2) $C_i^- = C_i^{*-} \vee g$ (that means g may appear in some contradiction involving C_i when replacing all C_i^* occurring in $\mathscr{D}_1^*$ with C_i).

If $\mathscr{D}_1$ is a CS based deduction sequence from S to an empty clause, then it completes the proof according to the induction proof.

If $\mathscr{D}_1$ is a CS based deduction sequence from S to g, now that $S_2 \leq S$ and S is unsatisfiable, we have S_2 is unsatisfiable and $K(S_2) < n$. According to the induction hypothesis, there exists a CS based deduction sequence $\mathscr{D}_2$ from S_2 to an empty clause. Connecting $\mathscr{D}_1$ and $\mathscr{D}_2$, we can obtain a CS deduction sequence from S to an empty clause. Hence, this completes the proof according to the induction proof.

Lemma 4.1 Suppose a clause set $S = \{C_1, C_2,\ldots, C_m\}$ in first-order logic. If $C_1 \wedge C_2 \wedge\ldots\wedge C_m$ is a standard contradiction, then S is a quasi-contradiction (i.e., S is unsatisfiable).

Proof. Suppose arbitrary a substitution σ, for $(x_1,\ldots, x_m) \in \prod_{i=1}^{m} C_i$, consider $(x_1^\sigma, \ldots, x_m^\sigma) \in \prod_{i=1}^{m} C_i^\sigma$, from the assumption, there is some complementary pair among $(x_1,\ldots, x_m)$.

Without loss of generality, we assume x_1 and x_2 is a complementary pair, this implies that x_1 and x_2 share the same terms, the same number of terms, and the same location of each term, except for the predicate symbols which are negative each other. Therefore, x_1^σ and x_2^σ is also a complementary pair for the substitution σ, it follows that there is some complementary pair among $(x_1^\sigma, \ldots, x_m^\sigma)$.

Assume that all the variables in S are $y_1,\ldots, y_n$, and β is a ground substitution applied to $y_1,\ldots, y_n$. Then, $\prod_{i=1}^{m} C_i^\beta$ is a standard contradiction in propositional logic. Therefore, the clause set $S^\beta=\{C_1^\beta, \ldots, C_m^\beta\}$ corresponding to $\prod_{i=1}^{m} C_i^\beta$ is unsatisfiable. Moreover, notice that $S \leq S^\beta$, it implies that S is unsatisfiable.

Lemma 4.2 Suppose D_k, $D_{k-1}, \ldots, D_1$ are k clauses in first order logic. Assume that a substitution σ_i is applied to D_i (σ_i could be an empty substitution) for $i=k,\ldots, 1$, and the same literals merged after substitution, such that $\bigwedge_{i=k}^{1} D_i^{\sigma_i}$ is a standard contradiction.

The following statements hold:

(1) If there exists some complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, then there exist k-2 clauses among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$ which are all redundant clauses.

(2) If there does not exist any complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, then $\{D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}\}$ satisfies the complementary condition, i.e., there exists a complementary pair of literals P and ~P, and there also exist $D_i^{\sigma_i}$ and $D_j^{\sigma_j}$ such that: $D_i^{\sigma_i}=D_{i_0}^{\sigma_i} \vee P$, $D_{i_0}^{\sigma_i} \neq \emptyset$, and $D_j^{\sigma_j}=D_{j_0}^{\sigma_j} \vee \sim P$, $D_{j_0}^{\sigma_j} \neq \emptyset$.

Proof. (1) Since there exists some complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, without loss of generality, assume $D_1^{\sigma_1}$ and $D_2^{\sigma_2}$ is a complementary pair, it follows that $\bigwedge_{i=k}^{1} D_i^{\sigma_i}=D_2^{\sigma_2} \wedge D_1^{\sigma_1}$. Therefore, $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_3^{\sigma_3}$ are all redundant clauses.

(2) Since there does not exist any complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, so among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, there exists some clause which includes more than one literal.

Apparently, among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$, there exists non-redundant clause which includes more than one literal (otherwise, if the clauses which include more than one literal are all redundant clauses, then the non-redundant clause among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \ldots, D_1^{\sigma_1}$ are only unit clauses. In addition, note that $\wedge_{i=k}^{1} D_i^{\sigma_i}$ is a standard contradiction, it follows that there exists some complementary pair among $D_k^{\sigma_k}, D_{k-1}^{\sigma_{k-1}}, \dots, D_1^{\sigma_1}$, this contradicts to the assumption).

Without loss of generality, assume $D_k^{\sigma_k}$=$D_{k_0}^{\sigma_k}$∨P, $D_{k_0}^{\sigma_k} \neq \varnothing$, and $D_k^{\sigma_k}$ is a non-redundant clause which includes more than one literal. Then among $D_{k-1}^{\sigma_{k-1}}, \dots, D_1^{\sigma_1}$, there exists some clause which includes ~P (because there exists some complementary pair among any (x_k,…, x_1)∈ $D_k^{\sigma_k} \times D_{k-1}^{\sigma_{k-1}} \times \dots \times D_1^{\sigma_1}$, if there is no ~$P$ among $D_{k-1}^{\sigma_{k-1}}, \dots, D_1^{\sigma_1}$, then for any ($x_{k-1}$,…, x_1)∈ $D_{k-1}^{\sigma_{k-1}} \times \dots \times D_1^{\sigma_1}$, there exists some complementary pair among P, x_{k-1},…, x_1 if and only if there exists some complementary pair among x_{k-1},…, x_1. Therefore, there exists some complementary pair among any (x_{k-1},…, x_1)∈ $D_{k-1}^{\sigma_{k-1}} \times \dots \times D_1^{\sigma_1}$, that is, $\wedge_{i=k-1}^{1} D_i^{\sigma_i}$ is a standard contradiction. It follows that $D_k^{\sigma_k}$ is a redundant clause. This again contradicts to the assumption.

Without loss of generality, we assume $D_{k-1}^{\sigma_{k-1}}$=$D_{k-1_0}^{\sigma_{k-1}}$∨~P.

(1) If $D_{k-1_0}^{\sigma_{k-1}} \neq \varnothing$, then the conclusion holds;

(2) Suppose $D_{k-1_0}^{\sigma_{k-1}} = \varnothing$. In this case, considering a literal Q in $D_k^{\sigma_k}$ (because $D_k^{\sigma_k}$ is a non-redundant clause and $\varnothing \neq D_{k_0}^{\sigma_k}(\subset D_k^{\sigma_k})$, there always exists such literal Q), without loss of generality, assume there exist more than one literal in the clause $D_{k-2}^{\sigma_{k-2}}$ which includes ~Q, then the conclusion holds (now that $D_k^{\sigma_k} = D_{k_1}^{\sigma_k}$∨$Q$, $D_{k-2}^{\sigma_{k-2}}$=$D_{k-2_0}^{\sigma_{k-2}}$∨~Q, where $P \in D_{k_1}^{\sigma_k} \neq \varnothing$, $D_{k-2_0}^{\sigma_{k-2}} \neq \varnothing$); If there exist only one literal in the clause $D_{k-2}^{\sigma_{k-2}}$ which includes ~Q (i.e., $D_{k-2_0}^{\sigma_{k-2}} = \varnothing$), then $D_{k-1}^{\sigma_{k-1}}$=~P, $D_{k-2}^{\sigma_{k-2}}$=~Q.

Note that $D_k^{\sigma_k}$ includes P, so $D_k^{\sigma_k}$ does not include ~P. In addition, $D_{k_0}^{\sigma_k}$ includes p and $D_{k_0}^{\sigma_k} \subset D_k^{\sigma_k}$, it follows that Q is not ~P (certainly p is not P too).

Therefore $D_{k-1}^{\sigma_{k-1}}$ can be reconstructed as Q∨~P, denoted as ${D_{k-1}^{\sigma_{k-1}}}^*$=$Q$∨~$P$.

Notice that it follows from the binary resolution that $(Q\vee{\sim}P)\wedge{\sim}Q\leq{\sim}P$, and $(Q\vee{\sim}P)\wedge{\sim}Q\leq{\sim}Q$. Therefore, ${\sim}P\wedge{\sim}Q\leq(Q\vee{\sim}P)\wedge{\sim}Q\leq{\sim}P\wedge{\sim}Q$, and ${\sim}P\wedge{\sim}Q=(Q\vee{\sim}P)\wedge{\sim}Q$. Accordingly, we have

$$D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}}\wedge D_{k-2}^{\sigma_{k-2}}\wedge D_{k-3}^{\sigma_{k-3}}\wedge\ldots\wedge D_1^{\sigma_1}$$

$$=D_k^{\sigma_k}\wedge{\sim}P\wedge{\sim}Q\wedge D_{k-3}^{\sigma_{k-3}}\wedge\ldots\wedge D_1^{\sigma_1}$$

$$=D_k^{\sigma_k}\wedge(Q\vee{\sim}P)\wedge{\sim}Q\wedge D_{k-3}^{\sigma_{k-3}}\wedge\ldots\wedge D_1^{\sigma_1}$$

$$=D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}*}\wedge{\sim}Q\wedge D_{k-3}^{\sigma_{k-3}}\wedge\ldots\wedge D_1^{\sigma_1}$$

$$=D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}*}\wedge D_{k-2}^{\sigma_{k-2}}\wedge D_{k-3}^{\sigma_{k-3}}\wedge\ldots\wedge D_1^{\sigma_1} \qquad (*)$$

Considering the equation (*), we have $D_k^{\sigma_k}=D_{k_0}^{\sigma_k}\vee P$, $D_{k_0}^{\sigma_k}\neq\emptyset$ and $D_{k-1}^{\sigma_{k-1}*}=Q\vee{\sim}P$, which coincides with the conclusion.

Finally, we can always assume $D_{k-1_0}^{\sigma_{k-1}}\neq\emptyset$, this concludes the proof.

Lemma 4.3 (Soundness Lemma of the S-CS Based Dynamic Deduction in First-Order Logic) Let D_k, D_{k-1},..., D_1 be k clauses in first order logic. Assume that a substitution σ_i is applied to D_i (σ_i could be an empty substitution) for $i=k,\ldots,1$, and the same literals merged after substitution. Suppose $D_i^{\sigma_i}$ is partitioned into two sub-clauses $D_i^{\sigma_i-}$ and $D_i^{\sigma_i+}$ such that

i) $D_i^{\sigma_i}=D_i^{\sigma_i-}\vee D_i^{\sigma_i+}$, where $D_i^{\sigma_i-}$ and $D_i^{\sigma_i+}$ have no common literals;

ii) $D_i^{\sigma_i+}$ can be an empty clause, but $D_i^{\sigma_i-}$ cannot be an empty clause;

iii) $\bigwedge_{i=k}^{1} D_i^{\sigma_i-}$ is a standard contradiction.

Then we have $D_k\wedge D_{k-1}\wedge\ldots\wedge D_1\leq D_k^{\sigma_k+}\vee D_{k-1}^{\sigma_{k-1}+}\vee\ldots\vee D_1^{\sigma_1+}$, i.e.,

$$D_k\wedge D_{k-1}\wedge\ldots\wedge D_1\leq\mathfrak{C}_k^{s\sigma}(D_k, D_{k-1},\ldots, D_1),$$

where $\sigma=\bigcup_{i=k}^{1}\sigma_i$, σ_i is a substitution to D_i, $i=k, k-1\ldots,1$.

Proof. Note that $D_k\wedge D_{k-1}\wedge\ldots\wedge D_1\leq D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}}\wedge\ldots\wedge D_1^{\sigma_1}$, it implies that we only need to prove the following relationship holds:

$$D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}}\wedge\ldots\wedge D_1^{\sigma_1}\leq D_k^{\sigma_k+}\vee D_{k-1}^{\sigma_{k-1}+}\vee\ldots\vee D_1^{\sigma_1+} \qquad (A.1)$$

Set $N(D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-},...,\ D_1^{\sigma_1 -})= \sum_{i=k}^{1}\left|D_i^{\sigma_i -}\right| - k.$

The above (4.1) can be proved using induction proof on $N(D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-}, ...,\ D_1^{\sigma_1 -})$.

(1) If $N(D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-},...,\ D_1^{\sigma_1 -})=0$, it follows that $D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-},...,\ D_1^{\sigma_1 -}$ are all unit clauses. Hence, there exists some complementary pair among $D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-}, ...,\ D_1^{\sigma_1 -}$. Without loss of generality, assume $D_k^{\sigma_k -}$ and $D_{k-1}^{\sigma_{k-1}-}$ is a complementary pair. Then based on the soundness lemma of binary resolution, we have

$$D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}}\wedge...\wedge D_1^{\sigma_1}\leq D_k^{\sigma_k}\wedge D_{k-1}^{\sigma_{k-1}}=(D_k^{\sigma_k +}\vee D_k^{\sigma_k -})\wedge(D_{k-1}^{\sigma_{k-1}+}\vee D_{k-1}^{\sigma_{k-1}-})$$

$$\leq D_k^{\sigma_k +}\vee D_{k-1}^{\sigma_{k-1}+}\leq D_k^{\sigma_k +}\vee D_{k-1}^{\sigma_{k-1}+}\vee...\vee D_1^{\sigma_1 +}.$$

This mean the above (A.1) holds.

(2) Suppose the above (A.1) holds for $N(D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-}, ...,\ D_1^{\sigma_1 -})<m$, we need to prove the above (1) also holds for $N(D_k^{\sigma_k -},\ D_{k-1}^{\sigma_{k-1}-},...,D_1^{\sigma_1 -})=m$ (>0).

Note that $\wedge_{i=k}^{1} D_i^{\sigma_i -}$ is a standard contradiction. Without loss of generality, assume there is no complementary pair among $D_k^{\sigma_k -}, D_{k-1}^{\sigma_{k-1}-},..., D_1^{\sigma_1 -}$ (otherwise, if there is some complementary pair among $D_k^{\sigma_k -},D_{k-1}^{\sigma_{k-1}-},...,D_1^{\sigma_1 -}$, following the similar step in the above proof case (1), we can prove that (A.1) holds). According to Lemma 4.2, without loss of generality, assume $D_k^{\sigma_k -} = D_{k_0}^{\sigma_k -}\vee P$, $D_{k-1}^{\sigma_{k-1}-} = D_{k-1_0}^{\sigma_{k-1}-}\vee{\sim}P$, where P and $\sim P$ is a complementary pair of literals, $D_{k_0}^{\sigma_k -}\neq\varnothing$, and $D_{k-1_0}^{\sigma_{k-1}-}\neq\varnothing$.

Also note that

$$(D_{k_0}^{\sigma_k -}\vee P)\wedge(D_{k-1_0}^{\sigma_{k-1}-}\vee{\sim}P)\wedge D_{k-2}^{\sigma_{k-2}-}\wedge...\wedge D_1^{\sigma_1 -} = D_k^{\sigma_k -}\wedge D_{k-1}^{\sigma_{k-1}-}\wedge D_{k-2}^{\sigma_{k-2}-}\wedge...\wedge D_1^{\sigma_1 -}$$

$$\equiv 0 \text{ (i.e., unsatisfiable),}$$

Therefore, we have

$$D_{k_0}^{\sigma_k -}\wedge D_{k-1}^{\sigma_{k-1}-}\wedge D_{k-2}^{\sigma_{k-2}-}\wedge...\wedge D_1^{\sigma_1 -}\equiv 0 \tag{A.2}$$

$$D_k^{\sigma_k -}\wedge D_{k-1_0}^{\sigma_{k-1}-}\wedge D_{k-2}^{\sigma_{k-2}-}\wedge...\wedge D_1^{\sigma_1 -}\equiv 0. \tag{A.3}$$

i) Considering Eq. (4.2), we have

$N(D_{k_0}^{\sigma_k -}, D_{k-1}^{\sigma_{k-1}-}, D_{k-2}^{\sigma_{k-2}-}, \ldots, D_1^{\sigma_1 -}) = N(D_k^{\sigma_k -}, D_{k-1}^{\sigma_{k-1}-}, D_{k-2}^{\sigma_{k-2}-}, \ldots, D_1^{\sigma_1 -}) - 1 = m - 1 < m.$

Moreover, note that $D_{k_0}^{\sigma_k -} \wedge D_{k-1}^{\sigma_{k-1}-} \wedge D_{k-2}^{\sigma_{k-2}-} \wedge \ldots \wedge D_1^{\sigma_1 -}$ is still a standard contradiction, therefore, we have,

$$D_k^{\sigma_k} \wedge D_{k-1}^{\sigma_{k-1}} \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1}$$

$$= [D_k^{\sigma_k +} \vee D_k^{\sigma_k -}] \wedge D_{k-1}^{\sigma_{k-1}} \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1}$$

$$= [D_k^{\sigma_k +} \vee (P \vee D_{k_0}^{\sigma_k -})] \wedge D_{k-1}^{\sigma_{k-1}} \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1}$$

$$= [(D_k^{\sigma_k +} \vee P) \vee D_{k_0}^{\sigma_k -}] \wedge D_{k-1}^{\sigma_{k-1}} \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1} \text{ (by induction assumption)}$$

$$\leq (D_k^{\sigma_k +} \vee P) \vee D_{k-1}^{\sigma_{k-1}+} \vee D_{k-2}^{\sigma_{k-2}-} \vee \ldots \vee D_1^{\sigma_1 +} \quad \text{(A.4)}$$

ii) Similarly, considering Eq. (A.2), we have

$N(D_k^{\sigma_k -}, D_{k-1_0}^{\sigma_{k-1}-}, D_{k-2}^{\sigma_{k-2}-}, \ldots, D_1^{\sigma_1 -}) = N(D_k^{\sigma_k -}, D_{k-1}^{\sigma_{k-1}-}, D_{k-2}^{\sigma_{k-2}-}, \ldots, D_1^{\sigma_1 -}) - 1 = m - 1 < m.$

Moreover, note that $D_k^{\sigma_k -} \wedge D_{k-1_0}^{\sigma_{k-1}-} \wedge D_{k-2}^{\sigma_{k-2}-} \wedge \ldots \wedge D_1^{\sigma_1 -}$ is still a standard contradiction, therefore, we have,

$$D_k^{\sigma_k} \wedge D_{k-1}^{\sigma_{k-1}} \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1}$$

$$= D_k^{\sigma_k} \wedge [D_{k-1}^{\sigma_{k-1}+} \vee D_{k-1}^{\sigma_{k-1}-}] \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1}$$

$$= D_k^{\sigma_k} \wedge [D_{k-1}^{\sigma_{k-1}+} \vee (D_{k-1_0}^{\sigma_{k-1}-} \vee {\sim}P)] \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1}$$

$$= D_k^{\sigma_k} \wedge [(D_{k-1}^{\sigma_{k-1}+} \vee {\sim}P) \vee D_{k-1_0}^{\sigma_{k-1}-}] \wedge D_{k-2}^{\sigma_{k-2}} \wedge \ldots \wedge D_1^{\sigma_1} \text{ (induction assumption)}$$

$$\leq D_k^{\sigma_k +} \vee (D_{k-1}^{\sigma_{k-1}+} \vee {\sim}P) \vee D_{k-2}^{\sigma_{k-2}+} \vee \ldots \vee D_1^{\sigma_1 +} \quad \text{(A.5)}$$

It follows from (A.4) and (A.5) that

$$D_k^{\sigma_k} \wedge D_{k-1}^{\sigma_{k-1}} \wedge D_{k-2}^{\sigma_{k-2}} \wedge \cdots \wedge D_1^{\sigma_1}$$

$$\leq [(D_k^{\sigma_k +} \vee P) \vee D_{k-1}^{\sigma_{k-1}+} \vee D_{k-2}^{\sigma_{k-2}+} \vee \cdots \vee D_1^{\sigma_1 +}] \wedge [D_k^{\sigma_k +} \vee (D_{k-1}^{\sigma_{k-1}+} \vee {\sim}P) \vee D_{k-2}^{\sigma_{k-2}+} \vee \cdots \vee D_1^{\sigma_1 +}]$$

(applying binary resolution to P and $\sim P$)

$$\leq D_k^{\sigma_k +} \vee D_{k-1}^{\sigma_{k-1}+} \vee D_{k-2}^{\sigma_{k-2}+} \vee \cdots \vee D_1^{\sigma_1 +}.$$

Therefore, (A.1) holds. This concludes the proof.

Lemma 4.4 (**Lifting Lemma of the S-CS Based Dynamic Deduction in First-Order Logic**) In first-order logic, let $C_1, C_2,\ldots, C_m$ be clauses without common variables, C_i^0 an instance of C_i, $i = 1, 2,\ldots, m$. If Ω_0 (=$\mathscr{C}_m^{q\mu}(C_1^0, C_2^0,\ldots, C_m^0)$) is an S-CSC of $C_1^0, C_2^0,\ldots, C_m^0$, then there exists an S-CSC Ω(=$\mathscr{C}_m^{q\theta}(C_1, C_2,\ldots, C_m)$) of $C_1, C_2,\ldots, C_m$ such that Ω_0 is an instance of Ω, i.e., the following transformation diagram Fig. 4.1 holds, where μ and θ are the substitutions applied to $C_1^0, C_2^0,\ldots, C_m^0$ and $C_1, C_2,\ldots, C_m$ respectively while constructing the S-CSC.

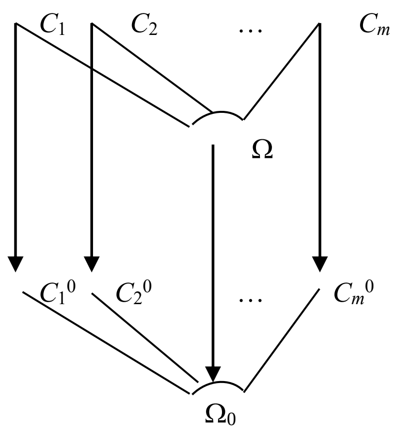

Fig. 4.1 The transformation diagram

Proof. Since C_i^0 is an instance of clause C_i, $i = 1, 2,\ldots, m$, so there exists a substitution ε_i such that $C_i^0 = C_i^{\varepsilon_i}$. Since Ω_0 is a standard contradiction separation clause of $C_1^0, C_2^0,\ldots, C_m^0$, it follows that there exists a substitution γ such that $(C_1^0)^{\gamma-}\wedge(C_2^0)^{\gamma-}\wedge\ldots\wedge(C_m^0)^{\gamma-}$ is a standard contradiction, and $\Omega_0 = \mathscr{C}_m^q(C_1^0, C_2^0,\ldots, C_m^0)= (C_1^0)^{\gamma+}\vee(C_2^0)^{\gamma+}\vee\ldots\vee(C_m^0)^{\gamma+}$, where $C_i^{\varepsilon_i\gamma}=(C_i^0)^{\gamma}= (C_i^0)^{\gamma+}(C_i^0)^{\gamma-}\vee$, $(C_i^0)^{\gamma+}$ and $(C_i^0)^{\gamma-}$ have no common literals, $(C_i^0)^{\gamma+}$ can be an empty clause, but $(C_i^0)^{\gamma-}$ cannot be an empty clause, $i = 1, 2,\ldots, m$.

Let $C_i = C_i^* \vee Y_i \vee X_i \vee Z_i$, $i = 1, 2,\ldots, m$ (here C_i^*, Y_i, X_i , Z_i are clauses which are composed of literals appearing in C_i) satisfy

(1) $(C_i^0)^{\gamma+}=(C_i^*)^{\varepsilon_i\gamma}=(C_i^* \vee Y_i)^{\varepsilon_i\gamma}$ (it implies that Y_i is unified into $(C_i^*)^{\varepsilon_i\gamma}$ after applied the substitution $\varepsilon_i\gamma$);

(2) $(C_i^0)^{\gamma-}={X_i}^{\varepsilon_i\gamma}=(X_i \vee Z_i)^{\varepsilon_i\gamma}$ (it implies that Z_i is unified into ${X_i}^{\varepsilon_i\gamma}$ after applied the substitution $\varepsilon_i\gamma$).

Assume there is no common variables among C_1, C_2,…, C_m (otherwise, rename substitution can be applied). So there is no common variables to be substituted while using the substitutions ε_1, ε_2,…, ε_m (i.e., there is no common variable in the denominators of ε_1, ε_2,…, ε_m), therefore, set $\varepsilon = \bigcup_{i=1}^{m} \varepsilon_i$, we have

$$(\wedge_{i=1}^{m} X_i)^{\varepsilon\gamma} = (\wedge_{i=1}^{m} X_i^{\varepsilon_i})^{\gamma} = \wedge_{i=1}^{m} X_i^{\varepsilon_i\gamma} = \wedge_{i=1}^{m} (X_i \vee Z_i)^{\varepsilon_i\gamma} = (\wedge_{i=1}^{m} (X_i \vee Z_i))^{\varepsilon\gamma}, \quad \text{(A.6)}$$

and

$$\begin{aligned}\Omega_0 &= (C_1^0)^{\gamma+} \vee (C_2^0)^{\gamma+} \vee \ldots \vee (C_m^0)^{\gamma+}\\ &= (C_1^* \vee Y_1)^{\varepsilon_1\gamma} \vee (C_2^* \vee Y_2)^{\varepsilon_2\gamma} \vee \ldots \vee (C_m^* \vee Y_m)^{\varepsilon_m\gamma}\\ &= (C_1^*)^{\varepsilon_1\gamma} \vee (C_2^*)^{\varepsilon_2\gamma} \vee \ldots \vee (C_m^*)^{\varepsilon_m\gamma}\\ &= (C_1^*)^{\varepsilon\gamma} \vee (C_2^*)^{\varepsilon\gamma} \vee \ldots \vee (C_m^*)^{\varepsilon\gamma}\\ &= (C_1^* \vee C_2^* \vee \ldots \vee C_m^*)^{\varepsilon\gamma}.\end{aligned}$$

Let $V_i = \{y_{i1}, y_{i2}, \ldots, \}$ be the set of all variables occurring in C_i ($i = 1, 2, \ldots, m$). Since C_1, C_2,…, C_m are clauses without common variables, so $V_1 \cap V_2 \cap \ldots \cap V_m = \varnothing$.

Let λ_i be the most general unifier of $\{X_i, X_i \vee Z_i\}$, $V_i = \{y_{i1}, y_{i2}, \ldots, y_{i\varepsilon_i}\}$ the set of all variables occurring in C_i ($i = 1, 2, \ldots, m$), $\lambda_i = \{t_{i1}/y_{i1}, t_{i2}/y_{i2}, \ldots, t_{is_i}/y_{i\varepsilon_i}\}$(without loss of generality, assume for any $i, j \in \{1, 2, \ldots, m\}$, if $i \neq j$, then any of $y_{i1}, y_{i2}, \ldots, y_{i\varepsilon_i}$ does not include any variable appearing in C_j, that is because C_1, C_2,…, C_m are clauses without common variables, and λ_i can be chosen as the regular substitution of C_i.

Let σ be a most general unifier of $\wedge_{i=1}^{m} X_i^{\lambda_i}$ and $\wedge_{i=1}^{m} (X_i \vee Z_i)^{\lambda_i}$ (here obviously $\wedge_{i=1}^{m} X_i^{\lambda_i}$ and $\wedge_{i=1}^{m} (X_i \vee Z_i)^{\lambda_i}$ can be unified). Then σ is also the most general unifier of $(\wedge_{i=1}^{m} X_i)^{\lambda}$ and $(\wedge_{i=1}^{m} (X_i \vee Z_i))^{\lambda}$, here $(\wedge_{i=1}^{m} X_i)^{\lambda\sigma} = (\wedge_{i=1}^{m} (X_i \vee Z_i))^{\lambda\sigma}$, $\lambda = \bigcup_{i=1}^{m} \lambda_i$.

It only needs to prove that $\lambda\sigma$ is a most general unifier $\wedge_{i=1}^{m} X_i$ and $\wedge_{i=1}^{m} (X_i \vee Z_i)$.

It is easy to see that $\lambda\sigma$ is a unifier of $\wedge_{i=1}^{m} X_i$ and $\wedge_{i=1}^{m} (X_i \vee Z_i)$.

Let ρ be a unifier of $\wedge_{i=1}^{m} X_i$ and $\wedge_{i=1}^{m} (X_i \vee Z_i)$. Then ρ is also a unifier of X_i and $X_i \vee Z_i$ (because C_1, C_2,…, C_m are clauses without common variables), i=1, 2,…, m.

Suppose the denominator part of ρ only have the variables $y_{11}, \ldots, y_{1\varepsilon_1}, \ldots, y_{i1}, \ldots, y_{i\varepsilon_i}, \ldots, y_{m1}, \ldots, y_{m\varepsilon_m}$(that is, ρ is a regular substitution of C_1, C_2,…, C_m), and let

$$\rho_i = \{u \mid u \in \rho, \text{ the denominator of } u \text{ occurs in } \{y_{i1}, y_{i2}, \ldots, y_{i\varepsilon_i}\}\}.$$

Then ρ_i is a unifier of X_i and $X_i \vee Z_i$, so there exists a substitution δ_i such that $\rho_i = \lambda_i\delta_i$, i = 1, 2,…, m.

Since the denominator of ρ_i only has variables $\{y_{i1}, y_{i2},\ldots, y_{i\varepsilon_i}\}$, so the denominator of δ_i also only has variables $\{y_{i1}, y_{i2},\ldots, y_{i\varepsilon_i}\}$. Let $\delta = \delta_1\cup\delta_2\cup\ldots\cup\delta_m$. We then have

$$
\begin{aligned}
&(\lambda_1\cup\lambda_2\cup\ldots\cup\lambda_m)\cdot(\delta_1\cup\delta_2\cup\ldots\cup\delta_m)\\
&=D_{k_0}^{\sigma_k}\ \{t_{11}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{11},\ t_{12}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{12},\ldots,\ t_{1s_1}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{1s_1},\ldots,\\
&\qquad t_{i1}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{i1},\ t_{i2}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{i2},\ldots,\ t_{is_i}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{is_i},\ldots,\\
&\qquad t_{m1}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{m1},\ t_{m2}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{m2},\ldots,\ t_{ms_m}^{(\delta_1\cup\ldots\cup\delta_m)}/y_{ms_m},\ldots,\\
&\qquad \delta_1\cup\delta_2\cup\ldots\cup\delta_m\}\\
&= \{t_{11}^{\delta_1}/y_{11},\ t_{12}^{\delta_1}/y_{12},\ \ldots,\ t_{1s_1}^{\delta_1}/y_{1\varepsilon_1},\ \ldots,\\
&\qquad t_{i1}^{\delta_i}/y_{i1},\ t_{i2}^{\delta_i}/y_{i2},\ \ldots,\ t_{is_i}^{\delta_i}/y_{i\varepsilon_i},\ \ldots,\\
&\qquad t_{m_1}^{\delta_m}/y_{m_1},\ t_{m_2}^{\delta_m}/y_{m_2},\ldots,\ t_{ms_m}^{\delta_m}/y_{ms_m},\ \delta_1\cup\delta_2\cup\ldots\cup\ \delta_m\}\\
&= \{t_{11}^{\delta_1}/y_{11},\ t_{12}^{\delta_1}/y_{12},\ldots,\ t_{1s_1}^{\delta_1}/y_{1\varepsilon_1},\ \delta_1,\ldots,\\
&\qquad t_{i1}^{\delta_i}/y_{i1},\ t_{i2}^{\delta_i}/y_{i2},\ldots,\ t_{is_i}^{\delta_i}/y_{i\varepsilon_i},\ \delta_i,\ldots,\\
&\qquad t_{m1}^{\delta_m}/y_{m1},\ t_{m2}^{\delta_m}/y_{m2},\ldots,\ t_{ms_m}^{\delta_m}/y_{ms_m},\ \delta_m\}\\
&= (\lambda_1\delta_1)\cup(\lambda_2\delta_2)\cup\ldots\cup(\lambda_m\delta_m).
\end{aligned}
$$

Denote $\lambda = \lambda_1\cup\lambda_2\cup\ldots\cup\lambda_m$ and $\delta = \delta_1\cup\delta_2\cup\ldots\cup\delta_m$, we then have

$$\rho = (\lambda_1\delta_1)\cup(\lambda_2\delta_2)\cup\ldots\cup(\lambda_m\delta_m) = \lambda\delta.$$

Because ρ is a unifier of $\bigwedge_{i=1}^{m} X_i$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)$, it follows that $\lambda\delta$ is also a unifier of $\bigwedge_{i=1}^{m} X_i$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)$, that implies that $(\bigwedge_{i=1}^{m} X_i)^{\lambda\delta} = (\bigwedge_{i=1}^{m}(X_i \vee Z_i))^{\lambda\delta}$, so δ is a unifier of $(\bigwedge_{i=1}^{m} X_i)^{\lambda}$ and $(\bigwedge_{i=1}^{m}(X_i \vee Z_i))^{\lambda}$.

Accordingly, there exists a substitution ξ such that $\delta=\sigma\xi$. It follows that $\rho = \lambda\sigma\xi$. Hence, $\lambda\sigma$ is the most general unifier of $\bigwedge_{i=1}^{m} X_i$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)$.

It follows from the above Eq. (A.6) that $\varepsilon\gamma$ is a unifier of $\bigwedge_{i=1}^{m} X_i$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)$, that means there exists a substitution η such that $\varepsilon\gamma=\lambda\sigma\eta$.

Let

$$\mathcal{C}_m^{q\theta}(C_1, C_2,\ldots, C_m)= (C_1^* \vee Y_1)^{\lambda_1\sigma}\vee(C_2^* \vee Y_2)^{\lambda_2\sigma}\vee\ldots\vee(C_m^* \vee Y_m)^{\lambda_m\sigma}.$$

Since $(\bigwedge_{i=1}^{m} X_i)^{\varepsilon\gamma}$=$\bigwedge_{i=1}^{m} X_i{}^{\varepsilon_i\gamma}$=$(C_1^0)^{\gamma-}\wedge(C_2^0)^{\gamma-}\wedge\ldots\wedge(C_m^0)^{\gamma-}$ is a standard contradiction, for any $\left(x_1^{\lambda_1\sigma}, x_2^{\lambda_2\sigma}, \ldots, x_m^{\lambda_m\sigma}\right) \in \prod_{i=1}^{m} X_i{}^{\lambda_i\sigma}$, it follows that

$$\left(x_1^{\lambda_1\sigma\eta}, x_2^{\lambda_2\sigma\eta}, \ldots, x_m^{\lambda_m\sigma\eta}\right) = \left(x_1^{\varepsilon_1\gamma}, x_2^{\varepsilon_2\gamma}, \ldots, x_m^{\varepsilon_m\gamma}\right) \in \prod_{i=1}^{m} X_i^{\varepsilon_i\gamma} .$$

Without loss of generality, we can set $x_1{}^{\varepsilon_1\gamma} = \sim(x_2{}^{\varepsilon_2\gamma})$, hence $x_1^{\lambda_1\sigma\eta}$ = $x_1{}^{\varepsilon_1\gamma} = \sim(x_2{}^{\varepsilon_2\gamma}) = \sim(x_2^{\lambda_2\sigma\eta})$. In addition, for arbitrary a unifier ρ of $\bigwedge_{i=1}^{m} X_i$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)$, there always exists a unifier δ of $(\bigwedge_{i=1}^{m} X_i)^{\lambda}$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)^{\lambda}$, and $\varepsilon\gamma$ is a unifier of $\bigwedge_{i=1}^{m} X_i$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)$. It follows that there exists a unifier $\delta_{\varepsilon\gamma}$ of $(\bigwedge_{i=1}^{m} X_i)^{\lambda}$ and $\bigwedge_{i=1}^{m}(X_i \vee Z_i)^{\lambda}$ such that $x_1^{\lambda_1\delta_{\varepsilon\gamma}} = \sim(x_2^{\lambda_2\delta_{\varepsilon\gamma}}) = \sim(\ x_2{}^{\lambda_2})^{\delta_{\varepsilon\gamma}}$, that is $x_1^{\lambda_1}$ and $\sim(x_2^{\lambda_2})$ can be unified with σ being the unifier. It means $x_1^{\lambda_1\sigma}$ =~$(\ x_2{}^{\lambda_2})^{\sigma}$, that is, there exists a complementary pair among $(x_1^{\lambda_1\sigma}, x_2^{\lambda_2\sigma}, \ldots, x_m^{\lambda_m\sigma})$. Therefore, $\bigwedge_{i=1}^{m} X_i{}^{\lambda_i\sigma}$ is a standard contradiction. Note that $\bigwedge_{i=1}^{m} X_i{}^{\lambda_i\sigma}$=$\bigwedge_{i=1}^{m}(X_i \vee Z_i)^{\lambda_i\sigma}$, so $\bigwedge_{i=1}^{m}(X_i \vee Z_i)^{\lambda_i\sigma}$ is also a standard contradiction. Therefore, we have

$$\mathfrak{C}_m^{q\theta}(C_1, C_2,\ldots, C_m)= (C_1^* \vee Y_1)^{\lambda_1\sigma}\vee(C_2^* \vee Y_2)^{\lambda_2\sigma}\vee\ldots\vee(C_m^* \vee Y_m)^{\lambda_m\sigma}$$

is a standard contradiction separation clause of $C_1, C_2,\ldots, C_m$ (here $D_i^+ = (C_i^* \vee Y_i)^{\lambda_i\sigma}$, $D_i^- = (X_i \vee Z_i)^{\lambda_i\sigma}$, i=1, 2,…, m, $\bigwedge_{i=1}^{m} D_i^-$ is a standard contradiction, and $\mathfrak{C}_m^{q\theta}(C_1, C_2,\ldots, C_m)$=$\bigvee_{i=1}^{m} D_i^+$ is standard contradiction separation clause of $C_1, C_2,\ldots, C_m$)

In addition, note that

$$(C_1^* \vee Y_1)^{\lambda_1\sigma}\vee(C_2^* \vee Y_2)^{\lambda_2\sigma}\vee\ldots\vee(C_m^* \vee Y_m)^{\lambda_m\sigma}$$
$$=[(C_1^* \vee Y_1)\vee(C_2^* \vee Y_2)\vee\ldots\vee(C_m^* \vee Y_m)]^{\lambda\sigma}.$$

Therefore, we have

$$(\mathfrak{C}_m^{q\theta}(C_1, C_2,\ldots, C_m))^{\eta}$$
$$=[(C_1^* \vee Y_1)\vee(C_2^* \vee Y_2)\vee\ldots\vee(C_m^* \vee Y_m)]^{\lambda\sigma\eta}$$
$$=[(C_1^* \vee Y_1)\vee(C_2^* \vee Y_2)\vee\ldots\vee(C_m^* \vee Y_m)]^{\varepsilon\gamma}$$
$$=(C_1^* \vee Y_1)^{\varepsilon\gamma}\vee(C_2^* \vee Y_2)^{\varepsilon\gamma}\vee\ldots\vee(C_m^* \vee Y_m)^{\varepsilon\gamma}$$
$$=(C_1^* \vee Y_1)^{\varepsilon_1\gamma}\vee(C_2^* \vee Y_2)^{\varepsilon_2\gamma}\vee\ldots\vee(C_m^* \vee Y_m)^{\varepsilon_m\gamma}$$
$$=(C_1^*)^{\varepsilon_1\gamma}\vee(C_2^*)^{\varepsilon_2\gamma}\vee\ldots\vee(C_m^*)^{\varepsilon_m\gamma}$$

$=(C_1^*)^{\varepsilon\gamma} \vee (C_2^*)^{\varepsilon\gamma} \vee \ldots \vee (C_m^*)^{\varepsilon\gamma}$

$=(C_1^* \vee C_2^* \vee \ldots \vee C_m^*)^{\varepsilon\gamma}$

$=\Omega_0$.

This completes the proof.