axioms

MDPI

Article

Decomposition-Based Checking and Local Certification for Propositional Circumscription via Minimal Reducts

Zhongtao Xie [1,*], Xin Zhou [1], Hongbo Hu [2] and Xiang Du [1]

[1] State Key Laboratory of Public Big Data, Institute of Artificial Intelligence, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
[2] Multi-Dimensional Data Perception and Intelligent Recognition Chongqing Engineering Research Center, Chongqing University of Arts and Sciences, Chongqing 402160, China
* Correspondence: francisol.net@gmail.com

Abstract

Propositional circumscription selects models that are minimal with respect to designated atoms while permitting another set of atoms to vary. For a clause theory φ, minimized atoms P, varied atoms Z, and a candidate interpretation M, the minimal-reduct characterization reduces candidate-model checking to the entailment $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge(M \cap P)$. We establish a structural decomposition of this entailment using the collapsed negative dependency graph of the reduct. Each selected source component induces a scoped entailment, and the global entailment is equivalent to the finite sequence of local obligations generated by successive contraction. The proof combines graph-based source selection, constructive extension of scoped countermodels, and preservation under contraction. These results yield a sound and complete checker that contracts certified minimized atoms and records origin-preserving certificate fragments over the original clauses. We instantiate the framework by direct SAT-based entailment checking and MUS-based support extraction. Experiments on 5445 random 3CNF instances and 462 industrial CNF instances show complete agreement with the global reduct criterion, and every generated certificate is successfully replayed. MUS-based extraction reduces the mean accumulated support size by 97.4% and 99.85% on the two benchmark collections, respectively, while incurring additional running time. The results provide a formal basis for local, replayable certification of propositional circumscription models.

Keywords: circumscription; non-monotonic reasoning; propositional logic; minimal reduct; decomposition; local certification; dependency graph; model checking

MSC: 68T30

Academic Editor: Churn Jung Liau

Received: 15 May 2026
Revised: 7 July 2026
Accepted: 9 July 2026
Published: 10 July 2026

1. Introduction

Circumscription is a foundational formalism of non-monotonic reasoning. It was introduced to represent closed-world and common-sense assumptions by minimizing designated predicates subject to a background theory [1–4]. In the propositional setting, the atoms are divided into minimized, varied, and fixed classes. A selected model admits no strictly preferable competing model that preserves the fixed part of the interpretation while decreasing the minimized part; the varied atoms remain unconstrained by the comparison. This semantics provides a concise account of defeasible conclusions, but it also makes model checking a global task over alternative interpretations.

Most computational work on propositional circumscription has therefore concentrated on deciding whether a candidate is minimal, generating circumscribed models, or compiling circumscription into another reasoning formalism. Representative approaches employ predicate elimination, translations to logic programming, SAT- and MaxSAT-based encodings, or structural restrictions on the input theory [5–10]. The complexity of the underlying inference and candidate-model checking tasks has been analyzed separately for propositional circumscription and closely related closed-world formalisms [11–13]. These methods provide powerful decision procedures. A Boolean answer alone, however, does not show which clauses force the minimized atoms of the candidate, how the relevant constraints interact, or how the result can be checked independently of the solver that produced it.

Recent research in mathematical logic has examined formal relationships between axiomatic and rule-based presentations, cut-free proof systems, and algorithmic model checking. Borrego-Díaz et al. [14] establish conservativity results for theories extended by conditional axioms and their associated inference rules. Lin and Ma [15] develop cut-free Gentzen sequent calculi for tense logics. In a different setting, Yu et al. [16] define a fuzzy computation-tree temporal logic with quality constraints and provide a model-checking algorithm together with its complexity analysis. These studies concern logical formalisms distinct from circumscription and are cited as recent examples of mathematical investigations into proof systems and logical decision procedures. The present work focuses specifically on candidate-relative certification for propositional circumscription, where a global minimality judgment is decomposed into local entailment obligations that can be independently reconstructed and checked.

This limitation is significant in verification-oriented applications. A result used in formal verification, knowledge-base maintenance, or solver validation should preferably be accompanied by a finite object whose correctness can be checked by a separate and comparatively simple procedure. For circumscription, such an object must retain the connection between the global minimality condition and the original clauses. Merely returning a collection of clauses is insufficient: the certificate must identify the obligation proved by those clauses and specify how a sequence of local proofs establishes the global result. We refer to a certificate with this property as replayable.

The starting point of this paper is the minimal-reduct characterization of propositional circumscription [17]. Let φ be a clause theory, let P and Z denote the minimized and varied atoms, respectively, and let M be a candidate interpretation. The characterization reduces circumscription model checking to a classical entailment:

$$M \models \mathrm{CIRC}[\varphi; P; Z] \iff \mathrm{Red}[\varphi; P; Z, M] \models \bigwedge(M \cap P). \tag{1}$$

The reduct discards clauses and literals that cannot participate in an admissible counter-model below the candidate. The remaining entailment is still global, but it has a purely propositional form and is therefore a suitable object for structural analysis.

Our aim is to decompose this entailment into a sequence of smaller obligations without weakening the equivalence above. This cannot be achieved by an arbitrary partition of the residual clauses. A correct decomposition must identify the part of the reduct relevant to a current block of minimized atoms, establish that clauses outside that part cannot affect the local entailment, and preserve the remaining global obligation after the verified atoms have been contracted. The construction must also account for source components containing only varied literals and must maintain a precise correspondence between residual clauses and their origins in the input theory. These requirements specify the mathematical conditions under which the decomposition is valid.

https://doi.org/10.3390/axioms15070521

We organize the residual dependencies through a collapsed negative dependency graph. Its strongly connected components represent groups of literals tied by cyclic negative dependencies. At each stage, source components whose labels are disjoint from the current verification target are pruned from the working graph. A remaining source component with non-empty target intersection then determines the next local obligation together with the residual clauses in its dependency scope. After that obligation has been certified, the verified minimized atoms are contracted, the reduct and graph are recomputed, and the construction continues. The main theorem proves that this process terminates and that the resulting local obligations are jointly equivalent to the original reduct entailment.

The proof separates four issues that are easily conflated in an algorithmic description. First, pruning must expose a source with a non-empty minimized target whenever such targets remain. Second, the selected scope must be semantically closed for the local obligation. Third, contraction must preserve exactly the unresolved part of the global entailment. Fourth, clause origins must remain well defined when residual clauses are simplified, deleted, or merged. Formalizing these points yields a decomposition theorem, a sound and complete checking procedure, and an implementation-independent replay relation.

A successful run records the selected components and clause supports needed for their local entailments. During replay, the checker reconstructs the current reduct, graph, scope, and target from the candidate and the previously verified steps. Consequently, the persistent certificate need not store implementation-specific graph identifiers or intermediate residual theories. A failed local obligation produces a local countermodel, which supplies a checkable explanation of why the corresponding global test fails.

We consider two certification strategies. The direct SAT-based strategy checks each scoped entailment and retains the complete scoped support. The MUS-based strategy extracts an inclusion-minimal unsatisfiable subtheory before mapping the selected residual clauses back to their original clauses. Both strategies implement the same semantics and return the same checking result. Their difference concerns the proof object: MUS extraction usually provides a much smaller certificate at the cost of additional computation. Accordingly, the proposed framework is intended for local and replayable certification. It is not claimed to improve the running time of a global reduct checker when only the Boolean decision is required. The term *local certification* refers to the construction and independent validation of local proof obligations; the candidate interpretation itself is not modified.

The principal contributions are as follows.

(i) We give a graph-theoretic decomposition of the minimal-reduct entailment used in propositional circumscription checking. The formal development includes source exposure after pruning, semantic separation of local scopes, and preservation under successive contraction.

(ii) We derive a sound and complete source-component checker and define an origin-preserving certificate format with an independent replay procedure. The construction specifies a canonical encoding of stage-local dependency blocks and a representative origin map that remains well defined under residual simplification and duplicate-clause merging.

(iii) We instantiate the framework with SAT-based and MUS-based certification and compare both variants with the global reduct criterion on random and industrial CNF benchmarks. The evaluation measures decision agreement, locality, certificate size, replayability, and computational overhead.

The remainder of the paper is organized as follows. Section 2 reviews circumscription computation, minimal reducts, graph-based minimal-model construction, structural decomposition, and certificate-oriented reasoning. Section 3 introduces the formal setting. Section 4 develops the graph construction and proves the decomposition results. Section 5

presents the checker, certificate format, and certification procedures. Section 6 reports the experimental evaluation. Section 7 discusses the scope and limitations of the framework, and Section 8 concludes the paper.

2. Related Work

Circumscription was introduced by McCarthy as a second-order formalization of predicate minimization and subsequently developed as a general mechanism for non-monotonic knowledge representation [2,3]. Its computational study has included predicate elimination, reductions to propositional satisfiability, complexity analyses, candidate-model checking, and translations into logic programming [5–9,18–21]. Eiter and Gottlob characterize the complexity of propositional circumscription and extended closed-world reasoning [11], while Cadoli studies the complexity of checking models of circumscriptive formulae [12]. A broader account of complexity results for non-monotonic logics is given by Cadoli and Schaerf [13]. Related closed-world formalisms include the weak generalized closed-world assumption of Rajasekar et al. [22]. These works principally address the computation or complexity of circumscribed and closed-world consequences, rather than the construction of replayable local certificates for a fixed candidate model.

Reduct constructions provide a closely related semantic technique. They are fundamental to stable-model semantics, and their relationship with circumscription has been investigated in both propositional and first-order settings [23,24]. Dependency graphs and loop formulas further expose cyclic structure and reduce non-monotonic conditions to classical constraints [25]. For propositional circumscription, the minimal reduct of [17] is defined relative to a candidate interpretation and isolates the residual constraints relevant to the existence of a strictly preferred model. This characterization converts candidate minimality into a classical entailment problem.

Structural decomposition has also been studied extensively in logic programming and minimal-model reasoning. Dependency-based methods divide a reasoning task according to strongly connected components, program modules, or problem-specific structural partitions [26–28]. In answer set programming, such methods are generally formulated with respect to stable-model semantics and the rule dependency structure of the input program. Their purpose ranges from modular evaluation to search-space reduction and distributed solving.

A particularly relevant contribution is the graph-based construction of minimal models proposed by Angiulli et al. [29]. Their framework considers positive propositional theories represented by rules and forms a super-dependency graph from the strongly connected components of the associated atom–clause dependency graph. Given a source component S, the projected theory T^S is solved first; if X is a minimal model of T^S, the assignment on S is propagated by Reduce$(T, X, S \setminus X)$. Their composition theorem shows that X can be combined with a minimal model of the reduced theory to obtain a minimal model of T. Iteration yields the algorithm ModuMin. Its worst-case bound is expressed in terms of the largest component on which a complete minimal-model procedure is required, and source theories belonging to the HCF or HEF classes can be handled in polynomial time by the incomplete subprocedure used in the algorithm. Distinct non-trivial sources may be processed in parallel, and the residual theory is independent of the order in which the corresponding reductions are applied. The same paper derives an ordinary minimal-model checking procedure by restricting a positive theory to subsets of a candidate model and invoking ModuMin. It also observes that, although ModuMin always returns a minimal model, it need not generate every minimal model; sufficient conditions for generative completeness are formulated through the modular and one-source-head properties.

Proof-producing reasoning provides a complementary line of work. From a proof-complexity perspective, Beyersdorff and Chew [30] compare sequent-style and tableau calculi for propositional circumscription and establish strict separations in proof strength and exponential lower bounds for selected systems. Their analysis concerns the size of formal derivations, whereas the present work constructs candidate-relative local supports that can be reconstructed and replayed from the original theory.

SAT and MaxSAT cores, proof logging, and inconsistency certificates supply evidence that can be checked independently of the original solver [31–35]. At the level of mathematical proof systems, conservativity between axiomatic and rule-based presentations and cut-free sequent calculi provide related accounts of how semantic validity can be represented by explicit derivations [14,15]. These results do not yield circumscription certificates, but they clarify the distinction between a semantic consequence and a structured proof object witnessing that consequence. Justifications, explanations, and witness structures have likewise been developed for answer-set semantics [36–41]. These approaches establish the broader methodological basis for replacing an unverifiable Boolean answer with a compact semantic or solver-level object.

The closest connection with [29] concerns recursive graph-guided simplification. Both frameworks use a condensation DAG, select structurally exposed components, and simplify a residual theory after processing the selected component. The invariant maintained by the recursion is different. In the construction of Angiulli et al., the selected source is assigned a locally computed minimal model, and the composition theorem preserves the existence and minimality of the model assembled from the local assignment and a residual minimal model. In the present setting, the candidate interpretation is fixed before decomposition. A selected component C is accepted only after the scoped entailment establishes that every model of the corresponding local reduct satisfies every atom in Δ_C. Contraction therefore records a proved consequence of the reduct; it does not select a local assignment from among alternative minimal models.

This difference prevents a direct application of the minimal-model construction theorem to the minimal reduct. The framework of Angiulli et al. uses positive theories and ordinary set-inclusion minimality over all atoms. A minimal reduct in the present setting may contain constraint clauses and signed occurrences of varied atoms, while the circumscription order minimizes only the atoms in P and leaves the atoms in Z free to change. Treating the reduct as an ordinary minimal-model problem would consequently minimize the varied atoms as well, unless an additional semantic encoding were introduced. More importantly, the target condition is the skeptical entailment $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge(M \cap P)$. Computing one minimal model of the reduct cannot decide this condition, since another minimal model may omit an atom of $M \cap P$. The auxiliary checking construction of [29] addresses ordinary minimality of a fixed model by restricting the search to its subsets; it does not provide a decomposition of this candidate-relative entailment with varied atoms.

The graph-theoretic locality required by the two methods also differs. In [29], a source is an SCC of the atom–clause dependency graph, and the local theory is the projection whose clauses use only atoms of that source. In the collapsed negative dependency graph used here, the vertex set is $R \cup Z \cup \neg Z$, where R is the current verification target. After empty sources are pruned, a component containing minimized atoms may remain dependent on ancestors consisting solely of varied literals. Its local obligation must therefore be evaluated over an atom-completed ancestor scope rather than over the component projection alone. The scoped-countermodel extension lemma supplies the formal condition under which clauses outside that scope are irrelevant to the selected entailment. After a successful obligation, Δ_C is removed from R, and both the reduct and its graph are recomputed;

correctness is derived from the two model-transfer propositions, without assuming the commutativity property of the Reduce steps established for parallel sources in [29].

These distinctions also delimit the algorithmic claims. The component-sensitive bound and the HCF/HEF tractability results of MODUMIN concern the construction of a minimal model of a positive theory and do not transfer to the local entailment tests considered here, which retain the complexity of propositional entailment and may incur further cost from MUS extraction. The purpose of the present decomposition is therefore structural localization and independently replayable certification. It yields a sound and complete decision procedure for the fixed circumscription candidate, together with an origin-preserving sequence of local supports or a scoped countermodel. This contrasts with the generative objective of MODUMIN, whose output is a minimal model and whose ability to generate all minimal models requires additional structural conditions.

3. Preliminaries

Let $\mathcal{L}$ be a propositional language over a finite set $\mathcal{A}$ of atoms. The constants $\bot$ and $\top$ are also allowed. A literal is an atom $p \in \mathcal{A}$ or its negation $\neg p$. A clause is a finite set of literals, identified with the disjunction of its elements. The empty clause is identified with $\bot$. A clause theory is a finite set of clauses, identified with their conjunction. For a clause α, we write

$$\alpha^+ = \mathcal{A} \cap \alpha, \qquad \alpha^- = \{p \in \mathcal{A} \mid \neg p \in \alpha\}.$$

For $S \subseteq \mathcal{A}$, let $\overline{S} = \mathcal{A} \setminus S$, $\neg S = \{\neg p \mid p \in S\}$, $\bigvee S = \bigvee_{p \in S} p$, and $\bigwedge S = \bigwedge_{p \in S} p$. An interpretation is identified with the set of atoms assigned true. Classical satisfaction is denoted by $\models$.

As usual in propositional circumscription, quantification over propositional variables is shorthand for expansion over the truth constants. Thus, if $A(z, p_1, \ldots, p_k)$ is a propositional formula, $\exists z A(z, p_1, \ldots, p_k)$ denotes $A(\top, p_1, \ldots, p_k) \vee A(\bot, p_1, \ldots, p_k)$, and $\forall z A(z, p_1, \ldots, p_k)$ is defined dually. The notation extends componentwise to tuples.

In the following, tuples of atoms are used for simultaneous replacement. We identify a tuple with the corresponding set when this causes no confusion. For instance, if $P = (p_1, \ldots, p_m)$ and $Z = (z_1, \ldots, z_n)$, then expressions such as $p \in P$, $P \cap M$, and $\mathcal{A} \setminus (P \cup Z)$ refer to the underlying sets of atoms. If $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ are tuples of propositional variables of the same lengths as P and Z, respectively, then $\varphi(X, Y)$ denotes the result of simultaneously replacing every p_i in $\varphi(P, Z)$ by x_i and every z_j by y_j. Atoms outside $P \cup Z$ are not replaced.

For tuples $P = (p_1, \ldots, p_m)$ and $X = (x_1, \ldots, x_m)$, define

$$X \leq P := \bigwedge_{i=1}^{m} (x_i \to p_i), \tag{2}$$

$$X = P := \bigwedge_{i=1}^{m} (x_i \leftrightarrow p_i), \tag{3}$$

$$X < P := (X \leq P) \wedge \neg(X = P). \tag{4}$$

Thus $X < P$ is a formula expressing that the comparison tuple is componentwise no larger than P and strictly smaller in at least one component.

Example 1. *Let* $P = \{p_1, p_2, p_3\}$, $Z = \{z_1\}$, *and clause theory*

$$\varphi = \{\alpha_1 : \neg p_1 \vee p_2, \quad \alpha_2 : p_1 \vee \neg p_2, \quad \alpha_3 : p_1 \vee \neg p_3, \quad \alpha_4 : p_3 \vee z_1\}.$$

When simultaneous replacement is used, take the displayed orders $P = (p_1, p_2, p_3)$ and $Z = (z_1)$. If $X = (x_1, x_2, x_3)$ and $Y = (y_1)$, then $\varphi(X, Y)$ is the clause theory

$$\{\neg x_1 \vee x_2, \quad x_1 \vee \neg x_2, \quad x_1 \vee \neg x_3, \quad x_3 \vee y_1\}.$$

No atom outside $P \cup Z$ occurs in this example; in general, such atoms are not replaced.

Definition 1 (Parallel circumscription)**.** *Let P and Z be disjoint tuples of atoms, and $\varphi(P, Z)$ be a propositional formula. The parallel circumscription of P in φ with Z allowed to vary is*

$$\mathrm{CIRC}[\varphi(P, Z); P; Z] := \varphi(P, Z) \wedge \neg\exists X \exists Y \big(\varphi(X, Y) \wedge X < P\big). \tag{5}$$

When $Z = \emptyset$, we write $\mathrm{CIRC}[\varphi; P]$.

The atoms in P are minimized, the atoms in Z are varied, and the atoms in $\mathcal{A} \setminus (P \cup Z)$ are fixed. The following model-theoretic order is the semantic counterpart of the syntactic comparison $X < P$.

Definition 2 (Interpretation ordering)**.** *Let $P, Z \subseteq \mathcal{A}$ be disjoint, and $M, N \subseteq \mathcal{A}$ be interpretations. We write $N \leq^{P;Z} M$ if*

$$N \cap P \subseteq M \cap P \quad \textit{and} \quad N \setminus (P \cup Z) = M \setminus (P \cup Z).$$

We write $N <^{P;Z} M$ if $N \leq^{P;Z} M$ and $M \not\leq^{P;Z} N$.

The relation $\leq^{P;Z}$ separates the two semantic requirements of circumscription: containment on the minimized atoms and agreement on the fixed atoms. Its strict part therefore identifies exactly the interpretations that can refute the minimality of a candidate. The next definition uses this order to state the model condition independently of the second-order presentation in Definition 1.

Definition 3 (Circumscription model)**.** *An interpretation M is a model of* $\mathrm{CIRC}[\varphi; P; Z]$*, written $M \models \mathrm{CIRC}[\varphi; P; Z]$, if $M \models \varphi$ and there is no model N of φ such that $N <^{P;Z} M$.*

Definitions 1–3 give syntactic and model-theoretic presentations of the same minimization principle. The model-theoretic form will be used in the reduct proofs, because it makes the construction of a smaller countermodel explicit. The following example isolates the role of the varied atoms in that comparison.

Example 2 (Effect of varied atoms)**.** *Let $P = \{p_1, p_2, p_3\}$, $Z = \{z_1\}$, and clause theory*

$$\varphi = \{\alpha_1 : \neg p_1 \vee p_2, \quad \alpha_2 : p_1 \vee \neg p_2, \quad \alpha_3 : p_1 \vee \neg p_3, \quad \alpha_4 : p_3 \vee z_1\}.$$

The interpretation $M = \{p_1, p_2, p_3\}$ satisfies φ. However, $N = \{z_1\}$ also satisfies φ, and

$$N \cap P = \emptyset \subsetneq P = M \cap P, \qquad N \setminus (P \cup Z) = M \setminus (P \cup Z) = \emptyset.$$

Thus $N <^{P;Z} M$. The atom z_1 is allowed to change because it is varied, while the minimized atoms in P become strictly smaller. Hence M is not a model of $\mathrm{CIRC}[\varphi; P; Z]$.

The syntactic formula $X < P$ and the semantic relation $<^{P;Z}$ will be used in different contexts: the former occurs in the second-order definition of circumscription, while the latter compares interpretations. In the decomposition and certification sections, $R \subseteq M \cap P$

denotes the current verification target. Initially $R = M \cap P$; after a certified block $\Delta \subseteq R$ is contracted, the target is updated to $R \setminus \Delta$. When a statement concerns an arbitrary set $R \subseteq P$, this is specified explicitly. In $\mathrm{Red}[\varphi; R; Z, M]$, R denotes the set of minimized atoms, and atoms in $P \setminus R$ are treated as fixed.

3.1. Minimal Reducts

The minimal reduct used below is the reduct for propositional circumscription introduced in [17]. It is used here as a model-checking device: relative to a candidate interpretation, it keeps precisely the residual clauses that may still rule out interpretations smaller under $<^{P;Z}$.

Definition 4 (Minimal reduct). *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ which are all disjoint, and $M \subseteq \mathcal{A}$ be an interpretation. For each clause $\alpha \in \varphi$, define*

$$\mathrm{Red}[\alpha; P; Z, M] = \big(\alpha^+ \cap ((P \cap M) \cup Z)\big) \cup \neg\big(\alpha^- \cap ((P \cap M) \cup Z)\big) \tag{6}$$

provided that the following conditions hold:

(i) $\alpha^- \cap P \cap \overline{M} = \emptyset$;
(ii) $\alpha^+ \cap \overline{P \cup Z} \cap M = \emptyset$;
(iii) $\alpha^- \cap \overline{P \cup Z} \cap \overline{M} = \emptyset$.

If one of these conditions fails, set $\mathrm{Red}[\alpha; P; Z, M] = \top$, *meaning that the clause is omitted from the reduct. The minimal reduct of φ with respect to P, Z, M is*

$$\mathrm{Red}[\varphi; P; Z, M] = \{\mathrm{Red}[\alpha; P; Z, M] \mid \alpha \in \varphi, \mathrm{Red}[\alpha; P; Z, M] \neq \top\}.$$

The three side conditions distinguish the literals that can be evaluated from those that must remain in the residual theory. Condition (i) removes clauses already satisfied by a minimized atom that is false in the candidate; conditions (ii) and (iii) evaluate the fixed atoms according to M. The retained clause therefore contains only literals whose values may still vary in a comparison below M.

Every atom occurring in the reduct belongs to $(P \cap M) \cup Z$. Fixed atoms are evaluated according to M, and minimized atoms false in M cannot become true in an interpretation smaller than M on P.

Example 3. *Let $P = \{p_1, p_2, p_3\}$, $Z = \{z_1\}$, and clause theory*

$$\varphi = \{\alpha_1 : \neg p_1 \vee p_2, \quad \alpha_2 : p_1 \vee \neg p_2, \quad \alpha_3 : p_1 \vee \neg p_3, \quad \alpha_4 : p_3 \vee z_1\}.$$

For the candidate interpretation $M = \{p_1, p_2, p_3\}$, we have $P \cap M = P$, and there are no fixed atoms in this example. Hence each clause satisfies the side conditions of Definition 4, and

$$\mathrm{Red}[\varphi; P; Z, M] = \varphi.$$

For comparison, if $N = \{z_1\}$, then $P \cap N = \emptyset$. The clauses α_1, α_2, and α_3 are omitted by condition (i), while $\alpha_4 = p_3 \vee z_1$ reduces to z_1. Thus

$$\mathrm{Red}[\varphi; P; Z, N] = \{z_1\}.$$

The reduct is therefore determined jointly by the candidate interpretation and by the roles of minimized and varied atoms.

Propositions 1 and 2 establish the two model-transfer directions used in the reduct characterization.

Proposition 1 (Preservation of smaller models)**.** *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M].$$

If $N \models \varphi$ and $N <^{P;Z} M$, then $N \models \psi$.

Proof. Let $\gamma \in \mathrm{Red}[\varphi; P; Z, M]$ be derived from a clause $\alpha \in \varphi$. Thus

$$\gamma = \big(\alpha^+ \cap ((P \cap M) \cup Z)\big) \cup \neg\big(\alpha^- \cap ((P \cap M) \cup Z)\big),$$

and the three side conditions in Definition 4 hold for α. Suppose, for a contradiction, that $N \not\models \gamma$. Then

$$N \cap \big(\alpha^+ \cap ((P \cap M) \cup Z)\big) = \emptyset \quad \text{and} \quad \alpha^- \cap ((P \cap M) \cup Z) \subseteq N.$$

We show that all omitted literals of α are false in N as well. Since $N <^{P;Z} M$, we have $N \cap P \subseteq M \cap P$ and $N \setminus (P \cup Z) = M \setminus (P \cup Z)$. For positive atoms omitted from γ,

$$\alpha^+ \setminus ((P \cap M) \cup Z) \subseteq (P \setminus M) \cup \overline{P \cup Z}.$$

The part in $P \setminus M$ is false in N because $N \cap P \subseteq M \cap P$, and the fixed part $\alpha^+ \cap \overline{P \cup Z}$ is false in N by agreement with M together with condition (ii), namely $\alpha^+ \cap \overline{P \cup Z} \cap M = \emptyset$. Hence no positive literal of α is true in N.

For negative literals omitted from γ, condition (i) gives $\alpha^- \cap P \cap \overline{M} = \emptyset$; hence no negative minimized atom false in M is omitted. The remaining omitted negative atoms are fixed atoms outside $P \cup Z$. By the fixed-atom agreement between N and M, condition (iii), $\alpha^- \cap \overline{P \cup Z} \cap \overline{M} = \emptyset$, implies that each such atom is true in N. Therefore every omitted negative literal is false in N. Together with $N \not\models \gamma$, this yields $N \not\models \alpha$, contradicting $N \models \varphi$. Thus $N \models \gamma$, and since γ was arbitrary, $N \models \mathrm{Red}[\varphi; P; Z, M]$. □

The converse transfer is stated next for an arbitrary set $R \subseteq P$ of minimized atoms.

Proposition 2 (Lifting reduct models)**.** *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$R \subseteq P, \qquad \psi_R = \mathrm{Red}[\varphi; R; Z, M].$$

If $N \leq^{R;Z} M$ and $N \models \psi_R$, then $N \models \varphi$.

Proof. Let $\alpha \in \varphi$. We prove that $N \models \alpha$.

If $\mathrm{Red}[\alpha; R; Z, M] \neq \top$, then $N \models \mathrm{Red}[\alpha; R; Z, M]$. Since $\mathrm{Red}[\alpha; R; Z, M]$ is obtained from α by deleting literals and leaving the retained literals unchanged, any retained literal satisfying the reduct clause is also a literal of α. Hence $N \models \alpha$.

It remains to consider the case $\mathrm{Red}[\alpha; R; Z, M] = \top$. Then at least one side condition in Definition 4 fails. If condition (i) fails, there is an atom $p \in \alpha^- \cap R \cap \overline{M}$. Since $N \cap R \subseteq M \cap R$, we have $p \notin N$, and therefore the literal $\neg p$ satisfies α. If condition (ii) fails, there is an atom $p \in \alpha^+ \cap \overline{R \cup Z} \cap M$. Because p is fixed with respect to R, Z, the equality $N \setminus (R \cup Z) = M \setminus (R \cup Z)$ gives $p \in N$, so p satisfies α. If condition (iii) fails, there is an atom $p \in \alpha^- \cap \overline{R \cup Z} \cap \overline{M}$. Again p is fixed, whence $p \notin N$, and $\neg p$ satisfies α. In all cases $N \models \alpha$. Since α was arbitrary, $N \models \varphi$. □

The preceding propositions establish the semantic correspondence needed for model checking. A strictly smaller model of the original theory satisfies the reduct, while an appropriately bounded model of the reduct lifts to a model of the original theory. The next proposition records a separate stability property: once atoms have been removed from the minimized set, recomputing the reduct does not depend on whether the removal is performed in one step or in several steps.

The reduct is compositional under successive restriction of the set of minimized atoms.

Proposition 3 (Iterated reduct consistency)**.** *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R \subseteq P \cap M.$$

Then

$$\mathrm{Red}[\psi; R; Z, M] \equiv \mathrm{Red}[\varphi; R; Z, M],$$

where equivalence is understood after removing clauses reduced to $\top$ and ignoring duplicate clauses. More generally, if $R_2 \subseteq R_1 \subseteq P \cap M$, then

$$\mathrm{Red}[\mathrm{Red}[\varphi; R_1; Z, M]; R_2; Z, M] \equiv \mathrm{Red}[\varphi; R_2; Z, M].$$

Proof. It is enough to prove the general statement. Fix a clause $\alpha \in \varphi$. The one-step reduct $\mathrm{Red}[\alpha; R_2; Z, M]$ retains exactly the literals of α whose atoms belong to $(R_2 \cap M) \cup Z = R_2 \cup Z$, provided that no omitted literal is already forced to make α true or false by the side conditions of Definition 4.

Consider instead the two-step reduction through R_1. The first reduction retains only literals over $R_1 \cup Z$. The second reduction then evaluates every atom in $R_1 \setminus R_2$ according to M and retains only literals over $R_2 \cup Z$. Since $R_2 \subseteq R_1 \subseteq M$, a positive occurrence of an atom in $R_1 \setminus R_2$ is treated exactly as a fixed positive atom true in M, and a negative occurrence of such an atom is treated exactly as a fixed negative literal false in M. Therefore the side conditions that omit a clause in the two-step procedure are triggered precisely in the cases in which the corresponding side condition is triggered by the one-step reduction to R_2. If the clause is not omitted, the retained literals after both procedures are precisely

$$\big(\alpha^+ \cap (R_2 \cup Z)\big) \cup \neg\big(\alpha^- \cap (R_2 \cup Z)\big).$$

Thus the two reductions produce the same residual clause, or both omit it. Applying this clause-wise argument to every $\alpha \in \varphi$ proves the claim. □

Combining Propositions 1 and 2 yields the following characterization.

Theorem 1 (Reduct characterization)**.** *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P.$$

The following statements are equivalent:

(i) $M \models \mathrm{CIRC}[\varphi; P; Z]$;
(ii) $\psi \models \bigwedge R$;
(iii) $R = \{p \in P \mid \psi \models p\}$.

Proof. Assume first that $M \models \mathrm{CIRC}[\varphi; P; Z]$ and that (ii) fails. Then there is an interpretation I satisfying $\mathrm{Red}[\varphi; P; Z, M]$ and falsifying some $p \in M \cap P$. Define

$$N = (I \cap ((P \cap M) \cup Z)) \cup (M \setminus (P \cup Z)).$$

The reduct contains only atoms from $(P \cap M) \cup Z$, so $N \models \mathrm{Red}[\varphi; P; Z, M]$. Moreover, $N \leq^{P;Z} M$, and the containment on P is strict because $p \notin N$. By Proposition 2, $N \models \varphi$. Hence $N <^{P;Z} M$ is a smaller model of the original theory, contradicting $M \models \mathrm{CIRC}[\varphi; P; Z]$.

Conversely, assume (ii), and suppose that M is not a circumscription model. Then there exists $N \models \varphi$ such that $N <^{P;Z} M$. By Proposition 1, $N \models \mathrm{Red}[\varphi; P; Z, M]$. Since $N \cap P \subsetneq M \cap P$, some atom in $M \cap P$ is false in N, contradicting (ii). Thus (i) and (ii) are equivalent. Finally, (ii) implies (iii) because every atom of $R = M \cap P$ is entailed by ψ, while for each $p \in P \setminus R$ the interpretation M is a model of ψ and falsifies p; hence $\psi \not\models p$. The converse implication is immediate. This proves the theorem. □

Theorem 1 is the semantic point of departure for the remainder of the paper. It replaces comparison with all $<^{P;Z}$-smaller models by a single classical entailment problem. The decomposition developed below therefore has to preserve the models of the minimal reduct and the entailment of the atoms in $M \cap P$.

Example 4 (Cont. Example 3)**.** *Let* $M = \{p_1, p_2, p_3\}$. *Since* $\mathrm{Red}[\varphi; P; Z, M] = \varphi$, *the interpretation* $N = \{z_1\}$ *is a model of the reduct. In particular,*

$$N \not\models p_1, \qquad N \not\models p_2, \qquad N \not\models p_3.$$

Therefore $\mathrm{Red}[\varphi; P; Z, M] \not\models \bigwedge P \cap M$. *By Theorem 1, M is not a circumscription model. This is the entailment-form version of the comparison* $\{z_1\} <^{P;Z} \{p_1, p_2, p_3\}$ *from Example 2.*

By Theorem 1, model checking is reduced to a classical entailment over the minimal reduct. The dependency structure used below is defined on the literals represented by the vertex set $R \cup Z \cup \neg Z$.

3.2. Dependency Graphs

We use the negative dependency convention of [25,26].

In the graph below, minimized atoms occur as positive vertices, whereas varied atoms occur as both signed vertices. Membership in induced subtheories is determined by underlying atoms.

Definition 5 ((Negative) dependency graph)**.** *Let* φ *be a clause theory and* P *and* Z *be two disjoint sets of atoms. The negative dependency graph of* φ *on* P *with* Z *is the directed graph*

$$\mathbb{G}_{\varphi[P;Z]} = (V, E),$$

where

$$V = P \cup Z \cup \neg Z$$

and

$$E = \{(l_1, l_2) \mid \exists \alpha \in \varphi \text{ such that } \{\neg l_1, l_2\} \subseteq \alpha\}.$$

Here $\neg(\neg z)$ *is identified with* z *for* $z \in Z$. *Thus, for example, a clause containing* z *and* p *induces the edge* $\neg z \to p$.

An edge $l_1 \to l_2$ records that a clause can transmit the failure of l_1 to the requirement represented by l_2. Strongly connected components collect mutually dependent signed vertices, and the collapsed DAG orders the resulting dependency blocks. This order will later determine which sets of minimized atoms may be verified and contracted first.

Given a circumscription CIRC$[\varphi; P; Z]$ and a non-empty set $L \subseteq P$, L is called a *loop* of CIRC$[\varphi; P; Z]$ if, for any $p, q \in L$, there exists a path of non-zero length from p to q in $\mathbb{G}_{\varphi[P;Z]}$ such that all vertices in the path belong to

$$L \cup Z \cup \neg Z.$$

Given a directed graph $G = (V, E)$, the *collapsed dependency graph* $\mathbb{S}_G = (\mathbb{V}, \mathbb{E})$ of G is the directed acyclic graph (DAG) defined as follows:

- $\mathbb{V}$ is the set of strongly connected components (SCCs) of G. That is, every $C \in \mathbb{V}$ is a maximal subgraph $G' = (V', E')$ of G such that G' has a path from any vertex $v' \in V'$ to every other vertex in V'. When no confusion arises, we also denote such a component G' by its vertex set V'.
- $\mathbb{E}$ consists of all edges (C, C') such that $C, C' \in \mathbb{V}$, $C \neq C'$, and there exist vertices $u \in C$ and $v \in C'$ with $(u, v) \in E$.

For a clause theory φ and two disjoint sets of atoms P, Z, we write

$$\mathbb{S}_{\varphi[P;Z]}$$

instead of $\mathbb{S}_{\mathbb{G}_{\varphi[P;Z]}}$. When a strongly connected component C is used as a vertex of a collapsed dependency graph, lab(C) denotes the set of original graph vertices represented by C.

A source S of $\mathbb{S}_{\varphi[P;Z]}$ is called *empty* if $S \cap P = \emptyset$. For a component S of $\mathbb{S}_{\varphi[P;Z]}$, let φ_S denote the set of clauses $\alpha \in \varphi$ such that every vertex of $\mathbb{G}_{\varphi[P;Z]}$ whose underlying atom occurs in α belongs to S. Here the polarity of the occurrence in α is ignored: if an atom p appears in α either as p or as $\neg p$, then the graph vertices whose underlying atom is p are treated as occurring in α.

Example 5. *Let* $P = \{p_1, p_2, p_3\}$, $Z = \{z_1\}$, *and*

$$\varphi = \{\alpha_1 : \neg p_1 \vee p_2, \quad \alpha_2 : p_1 \vee \neg p_2, \quad \alpha_3 : p_1 \vee \neg p_3, \quad \alpha_4 : p_3 \vee z_1\}.$$

Take $M = \{p_1, p_2, p_3\}$. *Then* $M \models \varphi$,

$$\psi = \mathrm{Red}[\varphi; P; Z, M] = \varphi, \qquad R = M \cap P = P.$$

Thus $\mathbb{G}_{\psi[R;Z]}$ *has vertex set*

$$V = \{p_1, p_2, p_3, z_1, \neg z_1\}.$$

The clauses α_1 *and* α_2 *induce the edges* $p_1 \to p_2$ *and* $p_2 \to p_1$, *respectively. The clause* α_3 *induces the edge* $p_3 \to p_1$. *The clause* α_4 *contains* z_1, *and hence induces edges from* $\neg z_1$ *to the literals occurring with* z_1, *in particular* $\neg z_1 \to p_3$.

The solid part of Figure 1 shows the negative dependency graph $\mathbb{G}_{\psi[R;Z]}$. *The dashed boxes denote the vertices of the collapsed dependency graph* $\mathbb{S}_{\psi[R;Z]}$, *obtained by contracting strongly connected components.*

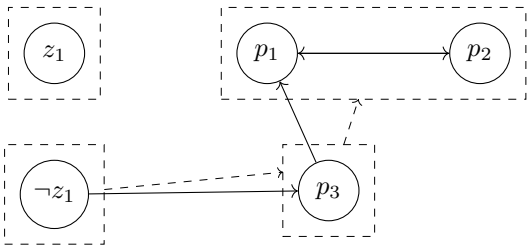

Figure 1. The negative dependency graph and the collapsed dependency graph for Example 5: solid arrows denote edges of $\mathbb{G}_{\psi[R;Z]}$, dashed boxes denote vertices of the collapsed dependency graph $\mathbb{S}_{\psi[R;Z]}$, and dashed arrows denote the dependency relation of $\mathbb{S}_{\psi[R;Z]}$.

For a component C of a collapsed dependency graph $\mathbb{S}$, let $\mathrm{Anc}_{\mathbb{S}}(C)$ be the set of its ancestor components. First form the raw ancestor set

$$A_{\mathbb{S}}(C) = \bigcup_{D \in \mathrm{Anc}_{\mathbb{S}}(C) \cup \{C\}} \mathrm{lab}(D).$$

The scope used in local obligations is the atom-completed ancestor scope

$$\mathrm{Scope}_{\mathbb{S}}(C) = A_{\mathbb{S}}(C) \cup \{z, \neg z \mid z \in Z,\ \{z, \neg z\} \cap A_{\mathbb{S}}(C) \neq \emptyset\}.$$

Thus, whenever one signed vertex of a varied atom is needed in a scope, the other signed vertex is included as well. The set $\mathrm{Scope}_{\mathbb{S}}(C)$ is a set of graph vertices, not a tuple and not necessarily a set of atoms. To speak about a local obligation for a component, we also need to restrict the reduct to the clauses whose graph vertices lie within a chosen scope.

Definition 6 (Induced subtheory)**.** *Let φ be a clause theory, $P, Z \subseteq \mathcal{A}$ be disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

and $\mathbb{G}_{\psi[R;Z]} = (V, E)$. Let $U \subseteq V$. The induced subtheory of ψ on U is

$$\psi_U = \{\alpha \in \psi \mid \textit{every vertex of } \mathbb{G}_{\psi[R;Z]} \textit{ whose underlying atom occurs in } \alpha \textit{ belongs to } U\}.$$

In this definition, occurrence is tested at the atom level rather than at the signed-literal level. Thus, if an atom p occurs in α either as p or as $\neg p$, then the graph vertices whose underlying atom is p are all relevant to the membership test for ψ_U.

Example 6 (Cont. Example 5)**.** *Let $\psi = \varphi$ and $R = P$. The clauses α_1 and α_2 involve only the atoms p_1 and p_2, while α_3 involves p_1 and p_3. The clause α_4 involves the atoms p_3 and z_1. Since z_1 is varied, both graph vertices z_1 and $\neg z_1$ are relevant whenever the atom z_1 occurs, regardless of the polarity of its occurrence in the clause. Consequently, if*

$$U = \{p_1, p_2, p_3\},$$

then

$$\psi_U = \{\alpha_1, \alpha_2, \alpha_3\},$$

because α_4 also depends on the varied-atom vertices z_1 and $\neg z_1$. If

$$U' = \{p_3, z_1, \neg z_1\},$$

then $\psi_{U'} = \{\alpha_4\}$.

4. Decomposition of Reduct Entailment

By Theorem 1, model checking for M is equivalent to the residual entailment $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge(M \cap P)$. We decompose this entailment by the collapsed dependency graph of the reduct. The decomposition is candidate-dependent, since the graph is computed after the minimal reduct has been formed [17,25,26].

The argument is developed in three stages. First, the collapsed graph induces an ordered partition of the minimized atoms occurring in the candidate. Second, a constructive countermodel-extension lemma identifies when the entailment for one graph component can be checked on its ancestor scope. Third, two reduct-transfer propositions justify contracting a verified block and continuing with the remaining blocks. This separation keeps the graph argument and the contraction argument logically distinct.

Throughout this section, put

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P.$$

A dependency partition of R must respect the topological order of $\mathbb{S}_{\psi[R;Z]}$; this is the condition under which verified blocks may be contracted without changing the residual test for later blocks.

Definition 7 (Dependency partition). *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

and $\mathbb{S}_{\psi[R;Z]} = (\mathbb{V}, \mathbb{E})$ be the collapsed dependency graph. Choose any topological ordering $C_1, \ldots, C_t$ of the components in $\mathbb{V}$. Remove all empty intersections $\mathrm{lab}(C_j) \cap R$, and write the remaining non-empty sets, in the induced order, as

$$L_1, \ldots, L_n.$$

The sequence $L_1, \ldots, L_n$ is called a dependency partition of R with Z for the reduct ψ. Each block L_i is the intersection of R with one strongly connected component of $\mathbb{G}_{\psi[R;Z]}$, and the order of the blocks respects the topological order of $\mathbb{S}_{\psi[R;Z]}$.

The partition is determined by dependency, not by an arbitrary grouping of R. Atoms in the same block are mutually dependent in the negative dependency graph, while the order between blocks is inherited from the condensation DAG. The prefix preceding a block therefore consists exactly of the minimized atoms whose dependency position permits them to be contracted before that block is considered.

For $i \geq 1$, let

$$X_i = L_1 \cup \cdots \cup L_{i-1}, \qquad X_1 = \emptyset.$$

The set X_i contains the target atoms verified before the block L_i. For each component, the checker must justify only the minimized atoms that occur in that component. The next definition names this target and the corresponding local entailment condition.

Definition 8 (Local verification obligation). *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

and $\mathbb{S} = \mathbb{S}_{\psi[R;Z]}$. Let C be a vertex of the collapsed dependency graph $\mathbb{S}$. Define

$$\Delta_C = \mathrm{lab}(C) \cap R.$$

If $\Delta_C \neq \emptyset$, the local verification obligation associated with C is

$$\psi_{\mathrm{Scope}_{\mathbb{S}}(C)} \models \bigwedge \Delta_C. \tag{7}$$

If $\Delta_C = \emptyset$, then C carries no local verification obligation.

For component selection, the algorithm works on a copy of the collapsed DAG. It repeatedly deletes current source components whose labels are disjoint from the current verification target, and then selects a source component whose label intersects that target. The scope of the selected component is computed in the original collapsed graph.

The decomposition results below use several closely related objects derived from the current reduct. To make their roles explicit before the main locality lemma and decomposition theorems, Table 1 summarizes the notation used throughout this section. Every entry has already been defined above or in Section 3; the table introduces no additional mathematical objects.

Table 1. Notation used in the decomposition results of Section 4.

Symbol	Meaning
ψ	minimal reduct $\mathrm{Red}[\varphi; P; Z, M]$ associated with the candidate M
R	minimized atoms true in the candidate, $M \cap P$
$\mathbb{S}$	collapsed negative dependency graph $\mathbb{S}_{\psi[R;Z]}$
C	a strongly connected component represented as a vertex of $\mathbb{S}$
$A_{\mathbb{S}}(C)$	union of the labels of C and its ancestor components
$\mathrm{Scope}_{\mathbb{S}}(C)$	atom-completed ancestor scope of C
Δ_C	local minimized-atom target $\mathrm{lab}(C) \cap R$
L_i	the i-th block of a dependency partition of R
X_i	previously verified prefix $L_1 \cup \cdots \cup L_{i-1}$
ψ_U	subtheory of ψ induced by the vertex set U

Lemma 1 (Lifting a scoped countermodel)**.** *Let φ be a clause theory over $\mathcal{A}$, let $R, Z \subseteq \mathcal{A}$ be disjoint sets of atoms, and let $M \models \varphi$ with $R \subseteq M$. Put*

$$\psi = \mathrm{Red}[\varphi; R; Z, M], \qquad \mathbb{S} = \mathbb{S}_{\psi[R;Z]}.$$

For a component C of $\mathbb{S}$, let

$$A = A_{\mathbb{S}}(C), \qquad U = \mathrm{Scope}_{\mathbb{S}}(C), \qquad \Delta = \mathrm{lab}(C) \cap R,$$

and let $\mathrm{At}(U)$ be the set of underlying atoms represented in U. If

$$\psi_U \not\models \bigwedge \Delta,$$

then there exist an atom $p \in \Delta$ and an interpretation I such that

$$I \models \psi_U, \qquad p \notin I,$$

and the interpretation

$$J = \big(I \cap \mathrm{At}(U)\big) \cup \big(M \setminus \mathrm{At}(U)\big) \tag{8}$$

satisfies ψ. In particular, $p \notin J$.

Proof. Choose $p \in \Delta$ for which $\psi_U \not\models p$. Among the models I of ψ_U with $p \notin I$, choose one minimizing, under set inclusion, the set

$$K(I) = \{l \in U \setminus A \mid M \models l \text{ and } I \not\models l\}.$$

Every vertex in $U \setminus A$ is the completion-added signed vertex of a varied atom; its complementary signed vertex belongs to A.

We first show that $K(I) = \emptyset$. Suppose otherwise, and let I' be obtained from I by restoring, to their values in M, all varied atoms whose M-true signed vertices belong to $K(I)$. The atom p is unchanged. Assume that some clause $\alpha \in \psi_U$ is falsified by I'. Since $I \models \alpha$, the clause α contains a literal $\neg l$, for some $l \in K(I)$, that is true under I and false under I'. On the other hand, $M \models \alpha$; hence α contains a literal l' that is true under M. The literal l' is false under I'. Its atom was therefore not restored, and the corresponding signed vertex does not belong to $K(I)$. Since $\alpha \in \psi_U$, this vertex belongs to U. It cannot lie in $U \setminus A$, because then it would belong to $K(I)$. Thus $l' \in A$.

The clause α contains both $\neg l$ and l'. By Definition 5, the graph contains the edge $l \to l'$. Since $l' \in A$, the component containing l is an ancestor of C, and hence $l \in A$, contradicting $l \in U \setminus A$. Therefore $I' \models \psi_U$. Moreover, $p \notin I'$ and $K(I') \subsetneq K(I)$, contradicting the choice of I. Hence $K(I) = \emptyset$.

Define J by (8). Suppose that $J \not\models \gamma$ for some $\gamma \in \psi$. The interpretation M satisfies every clause of ψ: for each retained reduct clause, the side conditions in Definition 4 ensure that every deleted literal is false under M. Thus γ contains a literal l' such that $M \models l'$. Since J agrees with M outside $\mathrm{At}(U)$, the underlying atom of l' belongs to $\mathrm{At}(U)$. The corresponding signed vertex is false under I. Because $K(I) = \emptyset$, this vertex belongs to A.

The clause γ cannot belong to ψ_U, because $I \models \psi_U$ and I and J agree on $\mathrm{At}(U)$. Hence γ contains a literal $\neg l$ whose underlying atom lies outside $\mathrm{At}(U)$. Since $J \not\models \gamma$ and J agrees with M outside $\mathrm{At}(U)$, we have $M \models l$. The signed vertex l lies outside U. However, $\{\neg l, l'\} \subseteq \gamma$, so Definition 5 yields the edge $l \to l'$. As $l' \in A$, this makes l an ancestor of C, contradicting $l \notin U$. Therefore $J \models \psi$. Finally, $p \in \Delta \subseteq \mathrm{At}(U)$, so $p \notin J$. □

The construction in Lemma 1 is the nontrivial direction of locality. It shows that a failure of the scoped obligation is a genuine failure of the full residual entailment, with the falsified minimized atom preserved. The converse direction requires only the set inclusion $\psi_U \subseteq \psi$, and the two directions are summarized in the following corollary.

Corollary 1 (Scoped entailment)**.** *Under the assumptions of Lemma 1,*

$$\psi \models \bigwedge \Delta \quad \Longleftrightarrow \quad \psi_U \models \bigwedge \Delta. \tag{9}$$

Proof. The implication from right to left follows from $\psi_U \subseteq \psi$. For the converse, assume $\psi_U \not\models \bigwedge \Delta$. By Lemma 1, there exists a model J of ψ that falsifies an atom of Δ. Hence $\psi \not\models \bigwedge \Delta$. □

Example 7 (Cont. Example 5)**.** *Let*

$$\psi = \mathrm{Red}[\varphi; P; Z, M] = \varphi, \qquad R = M \cap P = P.$$

In the collapsed dependency graph $\mathbb{S}_{\psi[R;Z]}$*, let* $C_{\neg z_1}$*,* C_{z_1}*,* C_3*, and* C_{12} *be the vertices satisfying*

$$\mathrm{lab}(C_{\neg z_1}) = \{\neg z_1\}, \qquad \mathrm{lab}(C_{z_1}) = \{z_1\}, \qquad \mathrm{lab}(C_3) = \{p_3\}, \qquad \mathrm{lab}(C_{12}) = \{p_1, p_2\}.$$

The component C_{12} corresponds to the strongly connected component formed by p_1 and p_2, while C_3 is the singleton component containing p_3. The edge $p_3 \to p_1$ in $\mathbb{G}_{\psi[R;Z]}$ induces an edge $C_3 \to C_{12}$ in $\mathbb{S}_{\psi[R;Z]}$. The edge $\neg z_1 \to p_3$ induces an edge $C_{\neg z_1} \to C_3$.

Hence the non-empty minimized components, ordered according to the collapsed dependency graph, give the dependency partition

$$L_1 = \mathrm{lab}(C_3) \cap R = \{p_3\}, \qquad L_2 = \mathrm{lab}(C_{12}) \cap R = \{p_1, p_2\}.$$

For the first component C_3, the local target is

$$\Delta_{C_3} = \mathrm{lab}(C_3) \cap R = \{p_3\}.$$

The scope of C_3 contains the vertices needed to express the clauses relevant to C_3. In this example,

$$\mathrm{Scope}_{\mathbb{S}}(C_3) = \{p_3, z_1, \neg z_1\}.$$

Therefore the induced local theory is

$$\psi_{\mathrm{Scope}_{\mathbb{S}}(C_3)} = \{\alpha_4 : p_3 \vee z_1\}.$$

The corresponding local verification obligation is

$$\psi_{\mathrm{Scope}_{\mathbb{S}}(C_3)} \models \bigwedge \Delta_{C_3},$$

that is,

$$\{p_3 \vee z_1\} \models p_3.$$

The interpretation $\{z_1\}$ satisfies $p_3 \vee z_1$ and falsifies p_3, so the obligation fails. The graph determines the component, and the local verification consists of the entailment over its induced subtheory.

The next two propositions establish model transfer between an uncontracted reduct and the reduct obtained after removing the verified blocks from the set of minimized atoms.

Proposition 4. *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

and $L_1, \ldots, L_n$ be a dependency partition of R with Z for ψ. For $1 \leq i \leq n$, let

$$X_i = L_1 \cup \cdots \cup L_{i-1}, \qquad X_1 = \emptyset.$$

If $N \models \mathrm{Red}[\varphi; P \setminus X_i; Z, M]$, then $N \cup X_i \models \mathrm{Red}[\varphi; P; Z, M]$.

Proof. Let $\beta \in \mathrm{Red}[\varphi; P; Z, M]$, and $\alpha \in \varphi$ be a clause from which β is derived. Put

$$\beta_i = \big(\alpha^+ \cap (((P \setminus X_i) \cap M) \cup Z)\big) \cup \neg\big(\alpha^- \cap (((P \setminus X_i) \cap M) \cup Z)\big)$$

when this clause is not omitted in $\mathrm{Red}[\varphi; P \setminus X_i; Z, M]$. Since $X_i \subseteq R = M \cap P$, moving from P to $P \setminus X_i$ treats the atoms of X_i as fixed true atoms.

If β_i belongs to $\mathrm{Red}[\varphi; P \setminus X_i; Z, M]$, then $N \models \beta_i$. The clause β is obtained from β_i by possibly adding literals whose atoms are in X_i. Hence $N \cup X_i \models \beta$ unless all satisfying literals of β were removed by the contraction. This cannot happen: any positive atom from X_i is true in $N \cup X_i$, while any negative literal over X_i is false and therefore irrelevant to satisfaction. Thus $N \cup X_i \models \beta$.

If $\beta_i \notin \mathrm{Red}[\varphi; P \setminus X_i; Z, M]$, then some side condition for $\mathrm{Red}[\varphi; P \setminus X_i; Z, M]$ fails. Since β is present in $\mathrm{Red}[\varphi; P; Z, M]$, conditions involving fixed atoms outside $P \cup Z$ cannot be responsible for the failure. Nor can condition (i) fail because all atoms of X_i are true in M. The only possible new failure is condition (ii), caused by a positive occurrence of an atom in X_i. Such an atom occurs positively in β and is true in $N \cup X_i$. Hence $N \cup X_i \models \beta$ in this case as well. Since β was arbitrary, $N \cup X_i \models \mathrm{Red}[\varphi; P; Z, M]$. □

The converse transfer holds for models containing the previously verified atoms.

Proposition 5. *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ with disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

and $L_1, \ldots, L_n$ be a dependency partition of R with Z for ψ. For some $i \in \{1, \ldots, n\}$, let

$$X_i = L_1 \cup \cdots \cup L_{i-1}, \qquad X_1 = \emptyset.$$

If $N \models \mathrm{Red}[\varphi; P; Z, M]$ and $X_i \subseteq N$, then $N \models \mathrm{Red}[\varphi; P \setminus X_i; Z, M]$.

Proof. Let $\beta_i \in \mathrm{Red}[\varphi; P \setminus X_i; Z, M]$, and $\alpha \in \varphi$ be the original clause from which it is derived. We compare β_i with the clause β obtained from α in $\mathrm{Red}[\varphi; P; Z, M]$.

First, β cannot be omitted. If condition (i) failed for the set P of minimized atoms, then it would also fail for $P \setminus X_i$, except possibly through an atom of X_i; but $X_i \subseteq M$, so no such atom is in $\overline{M}$. If condition (ii) or (iii) failed because of a fixed atom outside $P \cup Z$, the same failure would occur for $P \setminus X_i$. If condition (ii) failed because of a positive atom in X_i, then β_i would be omitted by the contracted reduct, contrary to the choice of β_i. Thus $\beta \in \mathrm{Red}[\varphi; P; Z, M]$.

Since $N \models \mathrm{Red}[\varphi; P; Z, M]$, we have $N \models \beta$. The only literals present in β but absent from β_i have atoms in X_i. Positive literals over X_i cannot occur, because β_i would then have been omitted by condition (ii) for the contracted reduct. Negative literals over X_i, if present in β, are false in N because $X_i \subseteq N$. Therefore $N \models \beta$ must be witnessed by a literal already present in β_i. Hence $N \models \beta_i$. Since β_i was arbitrary, $N \models \mathrm{Red}[\varphi; P \setminus X_i; Z, M]$. □

Propositions 4 and 5 yield the layer decomposition.

The blocks in the following theorem are not an arbitrary set partition. By Definition 7, they are obtained by intersecting R with the SCCs of the negative dependency graph and retaining the order induced by a topological ordering of the collapsed graph. Thus the partition appearing in the statement is graph-derived. Once this ordered partition is fixed, the semantic equivalence below follows from the two model-transfer propositions. The additional graph-theoretic argument required for locality is given afterwards, when a full residual entailment is replaced by an entailment over the ancestor scope of a selected source component.

Theorem 2 (Decomposition of reduct entailment)**.** *Let φ be a clause theory over $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ withs disjoint atom sets, $M \models \varphi$,*

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

and $L_1, \ldots, L_n$ be a dependency partition of R with Z for ψ. For $1 \leq i \leq n$, let

$$X_i = L_1 \cup \cdots \cup L_{i-1}, \qquad X_1 = \emptyset.$$

Then

$$M \models \mathrm{CIRC}[\varphi; P; Z]$$

if and only if, for every $i \in \{1, \ldots, n\}$,

$$\mathrm{Red}[\varphi; P \setminus X_i; Z, M] \models \bigwedge L_i. \tag{10}$$

Proof. By Theorem 1, it is enough to prove that $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge R$ is equivalent to the family of obligations (10). By Definition 7,

$$R = L_1 \dot{\cup} \cdots \dot{\cup} L_n,$$

and the order of these blocks is inherited from the collapsed dependency graph. The proof below uses this ordered partition together with the extension and restriction properties of the reduct.

Assume first that $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge R$. Suppose, for a contradiction, that (10) fails for some i. Then there exists an interpretation N such that

$$N \models \mathrm{Red}[\varphi; P \setminus X_i; Z, M] \quad \text{and} \quad N \not\models \bigwedge L_i.$$

By Proposition 4, $N \cup X_i \models \mathrm{Red}[\varphi; P; Z, M]$. Since $L_i \cap X_i = \emptyset$, the interpretation $N \cup X_i$ still falsifies some atom of L_i. This contradicts $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge R$, because $L_i \subseteq R$.

Conversely, assume that all obligations (10) hold. Let $N \models \mathrm{Red}[\varphi; P; Z, M]$. We prove by induction on i that

$$L_1 \cup \cdots \cup L_i \subseteq N.$$

For $i = 1$, we have $X_1 = \emptyset$, so the first obligation says $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge L_1$, and hence $L_1 \subseteq N$. For the induction step, assume $X_i = L_1 \cup \cdots \cup L_{i-1} \subseteq N$. By Proposition 5,

$$N \models \mathrm{Red}[\varphi; P \setminus X_i; Z, M].$$

The i-th obligation then gives $L_i \subseteq N$. Thus $X_{i+1} \subseteq N$. By induction, $R = L_1 \cup \cdots \cup L_n \subseteq N$. Since N was an arbitrary model of the original reduct, $\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge R$, as required. □

Theorem 2 treats all blocks of a dependency partition simultaneously. For the algorithm, the relevant statement is the corresponding one-step decomposition. Given the atoms already verified, the next result identifies a source block with a non-empty target intersection, replaces its full residual entailment by the scoped obligation, and separates the remaining target into the next reduct.

Theorem 3 (Source-component decomposition)**.** *Let* φ *be a clause theory over* $\mathcal{A}$*, let* $P, Z \subseteq \mathcal{A}$ *be disjoint sets of atoms, and let* $M \models \varphi$*. Let*

$$X \subseteq M \cap P, \qquad R = (M \cap P) \setminus X,$$

and put

$$\psi = \mathrm{Red}[\varphi; P \setminus X; Z, M].$$

Assume that $R \neq \emptyset$*, and let*

$$\mathbb{S} = \mathbb{S}_{\psi[R;Z]}.$$

Starting from a working copy of $\mathbb{S}$*, at each round delete all current source components* D *satisfying* $\mathrm{lab}(D) \cap R = \emptyset$*. Let* C *be a source component of the resulting graph such that*

$$\Delta = \mathrm{lab}(C) \cap R \neq \emptyset.$$

Define

$$U = \mathrm{Scope}_{\mathbb{S}}(C), \qquad \psi' = \mathrm{Red}[\varphi; P \setminus (X \cup \Delta); Z, M].$$

Then

$$\psi \models \bigwedge R \quad \Longleftrightarrow \quad \Big(\psi_U \models \bigwedge \Delta \text{ and } \psi' \models \bigwedge (R \setminus \Delta)\Big). \tag{11}$$

Proof. We first verify that the selection of C is well defined. The collapsed component graph $\mathbb{S}$ is a finite DAG, and every deleted source component has label disjoint from R. Hence no component containing an atom of R is removed. Since $R \neq \emptyset$, a target-bearing component remains. Repeated deletion must therefore expose a source component C satisfying $\Delta = \mathrm{lab}(C) \cap R \neq \emptyset$.

Let $Q = P \setminus X$. Since $X \subseteq M \cap P$,

$$R = (M \cap P) \setminus X = Q \cap M.$$

By Definition 4, atoms of $Q \setminus M$ produce the same reduction whether they are treated as minimized atoms false in M or as fixed atoms false in M. Consequently,

$$\psi = \mathrm{Red}[\varphi; Q; Z, M] = \mathrm{Red}[\varphi; R; Z, M]. \tag{12}$$

Thus Corollary 1 applies to the current residual theory.

Every deleted component was a source at the stage at which it was removed. Hence the deleted components can be placed before C in a topological ordering of $\mathbb{S}$. After blocks with empty intersection with R are omitted, Definition 7 yields a dependency partition whose first block is Δ. Thus the two model-transfer propositions apply to the theory φ, with minimized set $P \setminus X$ and first contracted block Δ.

Assume first that $\psi \models \bigwedge R$. Corollary 1 gives

$$\psi_U \models \bigwedge \Delta.$$

Let $N \models \psi'$. Proposition 4, applied to the first block Δ, yields

$$N \cup \Delta \models \psi.$$

Therefore $R \subseteq N \cup \Delta$. Since $(R \setminus \Delta) \cap \Delta = \emptyset$, it follows that $R \setminus \Delta \subseteq N$. Hence

$$\psi' \models \bigwedge (R \setminus \Delta).$$

Conversely, assume that

$$\psi_U \models \bigwedge \Delta \quad \text{and} \quad \psi' \models \bigwedge (R \setminus \Delta).$$

Let $N \models \psi$. By Corollary 1, $\psi \models \bigwedge \Delta$, and therefore $\Delta \subseteq N$. Proposition 5 then gives $N \models \psi'$. The second entailment implies $R \setminus \Delta \subseteq N$, whence $R \subseteq N$. Since N was arbitrary, $\psi \models \bigwedge R$. This proves (11). □

Example 8 (Source-component checking). *Let*

$$P = \{p_1, p_2, p_3\}, \qquad Z = \{z_1, z_2\},$$

and

$$\varphi = \{\alpha_1 : \neg p_1 \vee p_2,\ \alpha_2 : p_1 \vee \neg p_2,\ \alpha_3 : p_1 \vee \neg p_3,\ \alpha_4 : p_3 \vee z_1,\ \alpha_5 : q_1 \vee \neg z_1,\ \alpha_6 : p_2 \vee p_3 \vee z_2\}.$$

Consider

$$M_1 = \{p_1, p_2, p_3\}.$$

Then $M_1 \models \varphi$. *The minimal reduct of* φ *with respect to* M_1 *is*

$$\psi = \mathrm{Red}[\varphi; P; Z, M_1] = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5' : \neg z_1, \alpha_6\}.$$

Since

$$R = M_1 \cap P = P,$$

the collapsed dependency graph $\mathbb{S}_{\psi[R;Z]}$ *contains two non-empty minimized components. Let* C_3 *and* C_{12} *be the corresponding vertices of* $\mathbb{S}_{\psi[R;Z]}$, *with*

$$\mathrm{lab}(C_3) = \{p_3\}, \qquad \mathrm{lab}(C_{12}) = \{p_1, p_2\}.$$

Thus a source-component order gives the dependency partition

$$L_1 = \mathrm{lab}(C_3) \cap R = \{p_3\}, \qquad L_2 = \mathrm{lab}(C_{12}) \cap R = \{p_1, p_2\}.$$

A corresponding negative dependency graph is shown in Figure 2.

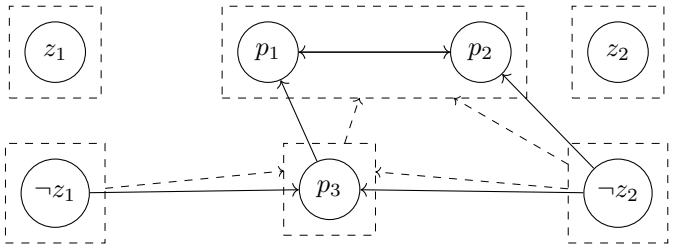

Figure 2. The negative dependency graph $\mathbb{G}_{\psi[R;Z]}$ for Example 8: solid arrows denote edges of $\mathbb{G}_{\psi[R;Z]}$, dashed boxes denote vertices of the collapsed dependency graph $\mathbb{S}_{\psi[R;Z]}$, and dashed arrows denote the dependency relation of $\mathbb{S}_{\psi[R;Z]}$.

For the first component C_3, *the local target is*

$$\Delta_{C_3} = \mathrm{lab}(C_3) \cap R = \{p_3\}.$$

Its scope in the collapsed dependency graph is

$$\mathrm{Scope}_{\mathbb{S}_{\psi[R;Z]}}(C_3) = \{p_3, z_1, \neg z_1, z_2, \neg z_2\}.$$

Indeed, $\alpha_6 = p_2 \vee p_3 \vee z_2$ *induces the edge* $\neg z_2 \to p_3$, *so both signed vertices of the varied atom* z_2 *enter the atom-completed ancestor scope. Nevertheless,* α_6 *is not a clause of the induced subtheory because it also contains the atom* p_2, *which lies outside this scope. Hence the induced local theory is*

$$\psi_{\mathrm{Scope}_{\mathbb{S}_{\psi[R;Z]}}(C_3)} = \{\alpha_4 : p_3 \vee z_1,\ \alpha_5' : \neg z_1\}.$$

The first local verification obligation is therefore

$$\psi_{\mathrm{Scope}_{\mathbb{S}_{\psi[R;Z]}}(C_3)} \models \bigwedge \Delta_{C_3},$$

that is,

$$\{p_3 \vee z_1, \neg z_1\} \models p_3.$$

After this local obligation is certified, the verified atom p_3 is contracted. The next residual reduct is

$$\mathrm{Red}[\varphi; P \setminus \{p_3\}; Z, M_1] = \{\alpha_1, \alpha_2, \alpha_3' : p_1, \alpha_5' : \neg z_1\}.$$

This residual reduct entails

$$p_1 \wedge p_2.$$

Thus the global entailment

$$\mathrm{Red}[\varphi; P; Z, M_1] \models p_1 \wedge p_2 \wedge p_3$$

is checked in two source-component steps:

$$\{p_3\} \quad \textit{first, then} \quad \{p_1, p_2\}.$$

Thus the source-component form verifies the global reduct entailment through successive local obligations and residual-reduct recomputation.

Section 5 augments these obligations with origin-preserving supports and scoped countermodels.

5. Checking Algorithm and Origin-Based Local Certification

The checker repeatedly applies the one-step equivalence of Theorem 3. At each successful step, it certifies the current scoped obligation, contracts the corresponding block, and recomputes the residual reduct. The procedure LocalCertify returns an origin-preserving support or a scoped countermodel.

A certificate fragment is recorded over the original theory. For a local residual theory λ, the returned support is a set $\kappa \subseteq \varphi$ such that replaying κ through the current reduct and scope yields an unsatisfiable target-negation test. Hence κ is a primitive-clause support for the local entailment.

5.1. Origin-Preserving Certificates and Failure Packages

During the iterative checking process, every residual clause is represented by at least one clause of the original theory. The certificate records one such representative because the checker constructs one sufficient certificate and does not enumerate all possible supports.

Let C_i be the source component selected at iteration i, and let

$$\Delta_i = \mathrm{lab}(C_i) \cap R_i.$$

The certificate stores Δ_i, rather than an SCC identifier. During replay, the verifier reconstructs ψ_i, R_i, and the collapsed graph. Since $\Delta_i \neq \emptyset$ and strongly connected components partition the graph vertices, any atom of Δ_i identifies a unique current component. The verifier then checks that its intersection with R_i is exactly Δ_i, and that it is selectable after the repeated removal of source components disjoint from R_i. Thus the encoding is independent of transient SCC numbers and of any ordering on atom names.

The component field and the clause field serve different purposes. The block Δ_i reconstructs the graph position of the local obligation, whereas the origin map reconstructs

a sufficient set of original clauses. Since the checker seeks one replayable support, one representative origin for each residual clause is sufficient.

For every residual theory ψ_i, let

$$\mathrm{Orig}_i : \psi_i \longrightarrow \varphi$$

be a representative origin map. For a set $\Lambda \subseteq \psi_i$, write

$$\mathrm{Orig}_i(\Lambda) = \{\mathrm{Orig}_i(\gamma) \mid \gamma \in \Lambda\}.$$

Initially, for every $\gamma \in \psi_0$, choose an arbitrary original clause $\alpha \in \varphi$ whose reduction is γ, and set $\mathrm{Orig}_0(\gamma) = \alpha$. Let $\rho_i(\gamma)$ denote the clause-level result of applying the next reduct step to γ. For every $\gamma' \in \psi_{i+1}$, choose an arbitrary clause $\gamma \in \psi_i$ such that $\rho_i(\gamma) = \gamma'$, and define

$$\mathrm{Orig}_{i+1}(\gamma') = \mathrm{Orig}_i(\gamma).$$

If several residual clauses produce the same clause γ', one predecessor is selected as its representative. If a clause reduces to $\top$, it is omitted and produces no entry in Orig_{i+1}. The representative choice may be made deterministic, for example by selecting the first predecessor in the fixed input-clause order.

This representative choice is sufficient because the checker seeks one original-clause support for each local entailment. It does not enumerate all supports or all provenance alternatives. For every selected residual clause, reducing its representative origin at the same stage reproduces that residual clause; consequently, replay soundness is independent of which representative was chosen.

Proposition 3, together with the same clause-wise observation used in (12), gives

$$\psi_i \equiv \mathrm{Red}[\varphi; R_i; Z, M],$$

where duplicate clauses are ignored. Hence each residual theory has the form required by Corollary 1.

Definition 9 (Origin-preserving decomposition certificate)**.** *Let φ be a clause theory over $\mathcal{A}$, let $P, Z \subseteq \mathcal{A}$ be disjoint atom sets, and let $M \models \varphi$. Put*

$$\psi_0 = \mathrm{Red}[\varphi; P; Z, M], \qquad R_0 = M \cap P.$$

An origin-preserving decomposition certificate *for M with respect to* $\mathrm{CIRC}[\varphi; P; Z]$ *is a finite sequence*

$$\mathcal{C} = \langle \chi_0, \ldots, \chi_{m-1} \rangle, \qquad \chi_i = (\Delta_i, \kappa_i),$$

where every Δ_i is a finite non-empty set of atoms and every κ_i is a set of clauses of φ. The sequence is valid if there exist residual theories $\psi_0, \ldots, \psi_m$, verification targets $R_0, \ldots, R_m$, and representative origin maps Orig_i *satisfying the following conditions for every $0 \leq i < m$.*

Let

$$\mathbb{S}_i = \mathbb{S}_{\psi_i[R_i;Z]}.$$

The non-empty set Δ_i identifies a component C_i of $\mathbb{S}_i$ through

$$\Delta_i = \mathrm{lab}(C_i) \cap R_i.$$

Starting from a working copy of $\mathbb{S}_i$, at each round delete all current source components whose labels are disjoint from R_i; validity requires that C_i be a source of the resulting graph. Define

$$U_i = \mathrm{Scope}_{\mathbb{S}_i}(C_i), \qquad \lambda_i = (\psi_i)_{U_i}, \qquad \beta_i = \bigvee \neg\Delta_i.$$

The support field satisfies

$$\kappa_i \subseteq \mathrm{Orig}_i(\lambda_i) \qquad \textit{and} \qquad \big(\mathrm{Red}[\kappa_i; R_i; Z, M]\big)_{U_i} \cup \{\beta_i\} \models \bot.$$

Finally,

$$R_{i+1} = R_i \setminus \Delta_i, \qquad \psi_{i+1} = \mathrm{Red}[\psi_i; R_{i+1}; Z, M],$$

the representative origin map is updated by the predecessor rule above, and

$$R_m = \emptyset.$$

The certificate therefore stores neither SCC identifiers nor scopes. The prefix $\Delta_0, \ldots, \Delta_{i-1}$ reconstructs the residual state at stage i; the current non-empty set Δ_i identifies C_i; and U_i and λ_i are derived from the recomputed graph and theory. The validity test also checks that the reconstructed component is eligible for source selection at that stage.

A successful fragment establishes an entailment by an unsatisfiable support. When the local test is satisfiable, a counterexample is given by a model of the scoped theory together with the negation of the local target. The following definition specifies the data required to replay such a failure.

Definition 10 (Failure package)**.** *Suppose the checker has successfully produced the prefix*

$$\mathcal{C}_{<i} = \langle(\Delta_0, \kappa_0), \ldots, (\Delta_{i-1}, \kappa_{i-1})\rangle$$

and has thereby reconstructed the current state $(\psi_i, R_i, \mathrm{Orig}_i)$. Let $\mathbb{S}_i = \mathbb{S}_{\psi_i[R_i;Z]}$. A failure package at stage i is a triple

$$\mathcal{F} = (\mathcal{C}_{<i}, \Delta_i, I_i),$$

where $\Delta_i \neq \emptyset$ identifies the current component C_i by $\Delta_i = \mathrm{lab}(C_i) \cap R_i$. Starting from a working copy of $\mathbb{S}_i$, at each round delete all source components whose labels are disjoint from R_i; validity requires that C_i be a source of the resulting graph. Put

$$U_i = \mathrm{Scope}_{\mathbb{S}_i}(C_i), \qquad \lambda_i = (\psi_i)_{U_i}, \qquad \beta_i = \bigvee \neg\Delta_i.$$

The package is valid if the prefix is replay-valid and

$$I_i \models \lambda_i \cup \{\beta_i\}.$$

Thus the prefix reconstructs the residual stage, Δ_i replaces a transient SCC identifier, and I_i witnesses failure of the reconstructed local obligation.

Corollary 2 (Soundness of a local failure package)**.** *Let $\mathcal{F} = (\mathcal{C}_{<i}, \Delta_i, I_i)$. be a valid failure package at stage i, with reconstructed state $(\psi_i, R_i, \mathrm{Orig}_i)$. Then*

$$\psi_i \not\models \bigwedge \Delta_i.$$

Moreover, there exists an interpretation J_i satisfying ψ_i and falsifying an atom of Δ_i. Writing the targets stored in the prefix as $\Delta_0, \ldots, \Delta_{i-1}$, define

$$J^{(i)} = J_i, \qquad J^{(j)} = J^{(j+1)} \cup \Delta_j \quad (j = i-1, \ldots, 0).$$

Then

$$J^{(0)} \models \mathrm{Red}[\varphi; P; Z, M] \quad \textit{and} \quad J^{(0)} \not\models \bigwedge(M \cap P).$$

Consequently, $M \not\models \mathrm{CIRC}[\varphi; P; Z]$.

Proof. Validity gives $I_i \models (\psi_i)_{U_i}$ and $I_i \not\models \bigwedge \Delta_i$. Corollary 1 therefore yields $\psi_i \not\models \bigwedge \Delta_i$. Lemma 1 constructs a model J_i of ψ_i that falsifies an atom of Δ_i. For each preceding stage $j < i$, Proposition 4 gives $J^{(j)} \models \psi_j$ from $J^{(j+1)} \models \psi_{j+1}$. The targets in the prefix are pairwise disjoint from Δ_i, because each successful stage removes its target from the next verification set. Hence adding $\Delta_{i-1}, \ldots, \Delta_0$ does not change the atom of Δ_i falsified by J_i. It follows that $J^{(0)} \models \psi_0$ and $J^{(0)} \not\models \bigwedge R_0$. The conclusion follows from Theorem 1. □

Successful and failed local checks therefore have dual replay objects. A successful fragment (Δ_i, κ_i) records an original-clause unsatisfiability support. A failure package $(\mathcal{C}_{<i}, \Delta_i, I_i)$ records the successful prefix needed to reconstruct the residual state together with a satisfying assignment for the local counterexample theory.

5.2. The Local Certification Procedure

Let λ be a local residual clause theory, Δ be a non-empty set of atoms, and ω be the restriction of the current origin map to clauses of λ. The local verification problem is

$$\lambda \models \bigwedge \Delta.$$

Equivalently, with

$$\beta_\Delta = \bigvee \neg\Delta, \qquad \Gamma_{\lambda,\Delta} = \lambda \cup \{\beta_\Delta\},$$

the local obligation holds exactly when $\Gamma_{\lambda,\Delta} \models \bot$. A model of $\Gamma_{\lambda,\Delta}$ is a local counterexample.

The procedure LocalCertify$(\lambda, \Delta, t, \omega)$, where $t \in \{\mathrm{SAT}, \mathrm{MUS}\}$, checks the satisfiability of $\Gamma_{\lambda,\Delta}$. If it is satisfiable, the procedure returns a countermodel. If it is unsatisfiable, the procedure first obtains a residual support $H \subseteq \lambda$ and then maps that support back to the original theory by setting

$$\kappa = \omega(H) = \{\omega(\gamma) \mid \gamma \in H\} \subseteq \varphi.$$

In SAT, take $H = \lambda$. In MUS, choose an inclusion-minimal residual subtheory $H \subseteq \lambda$ with $H \cup \{\beta_\Delta\} \models \bot$. An optional second minimization may be applied to $\omega(H)$ at the original-clause level; soundness requires only the displayed unsatisfiability condition.

Example 9. *Using Example 8, consider the first block $L_1 = \{p_3\}$ for the interpretation M_1. The local target is $\Delta = \{p_3\}$, and the induced local residual theory is*

$$\lambda = \{\alpha_4 : p_3 \vee z_1,\ \alpha_5' : \neg z_1\}.$$

The residual clause $\alpha_5' : \neg z_1$ is obtained from the original clause

$$\alpha_5 : q_1 \vee \neg z_1$$

by evaluating the fixed atom q_1 under the candidate interpretation. Hence the origin map sends α_4 to the original clause α_4, and sends α_5' to the original clause α_5. Since

$$\{p_3 \vee z_1,\ \neg z_1,\ \neg p_3\} \models \bot,$$

the local obligation holds. The origin-preserving certificate fragment stores the following original clause support:

$$\kappa = \{\alpha_4 : p_3 \vee z_1,\ \alpha_5 : q_1 \vee \neg z_1\} \subseteq \varphi.$$

During replay, κ reduces on the same scope to the residual support $\{p_3 \vee z_1, \neg z_1\}$; adjoining $\neg p_3$ yields an unsatisfiable theory.

Proposition 6. *Let λ be a local residual theory at a stage with verification target R, scope U, and origin map ω. Let $\Delta \neq \emptyset$, and let*

$$\beta_\Delta = \bigvee \neg\Delta.$$

If Algorithm 1 returns $(\textbf{proved}, \kappa)$, then

$$\kappa \subseteq \varphi \qquad \textit{and} \qquad \big(\mathrm{Red}[\kappa; R; Z, M]\big)_U \cup \{\beta_\Delta\} \models \bot.$$

Consequently,

$$\lambda \models \bigwedge \Delta.$$

If it returns $(\textbf{refuted}, I)$, then

$$I \models \lambda \qquad \textit{and} \qquad I \not\models \bigwedge \Delta.$$

Algorithm 1: LocalCertify$(\lambda, \Delta, t, \omega)$

Input: A local residual theory λ, a non-empty target Δ, a strategy $t \in \{\mathrm{SAT}, \mathrm{MUS}\}$, and an origin map ω for clauses of λ

Output: $(\textbf{proved}, \kappa)$ with $\kappa \subseteq \varphi$, or $(\textbf{refuted}, I)$

$\beta \leftarrow \bigvee \neg\Delta$ `// target-negation clause`
$\Gamma \leftarrow \lambda \cup \{\beta\}$ `// counterexample theory`
test the satisfiability of Γ
if Γ *has a model* I **then**
  return $(\textbf{refuted}, I)$ `//` $I \models \lambda$ `and` $I \not\models \bigwedge \Delta$
else
  if $t = \mathrm{MUS}$ **then**
    choose $H \subseteq \lambda$ such that $H \cup \{\beta\} \models \bot$ and H is inclusion-minimal among residual supports
  else
    $H \leftarrow \lambda$ `// full local residual theory as support`
  $\kappa \leftarrow \omega(H)$ `// map residual support back to original clauses`
  optionally remove original clauses from κ while preserving replay unsatisfiability
  return $(\textbf{proved}, \kappa)$

Proof. If the algorithm returns $(\textbf{refuted}, I)$, then $I \models \lambda \cup \{\beta_\Delta\}$, which immediately gives the stated local countermodel property. If the algorithm returns $(\textbf{proved}, \kappa)$, it has selected a residual support $H \subseteq \lambda$ such that $H \cup \{\beta_\Delta\} \models \bot$ and then mapped H back to original clauses through ω. By the defining invariant of the representative origin map, reducing

each clause in $\kappa = \omega(H)$ at the same stage and restricting to the same scope reproduces the corresponding clause of H. Hence H is contained in the replayed residual support, and

$$\big(\mathrm{Red}[\kappa; R; Z, M]\big)_U \cup \{\beta_\Delta\} \models \bot.$$

This implies $\lambda \cup \{\beta_\Delta\} \models \bot$, and therefore $\lambda \models \bigwedge \Delta$. □

5.3. Main Checker

Algorithm 1 certifies a single source-component obligation. We now combine these local checks into a complete checker for the fixed candidate interpretation M. At the beginning of each iteration, the checker maintains the set R of minimized atoms that remain to be certified, the current residual reduct ψ, a representative origin map from clauses of ψ to clauses of φ, and the certificate prefix $\mathcal{C}$ already obtained. It exposes a source component whose label intersects R, constructs the corresponding ancestor scope, and invokes Algorithm 1 on the induced local theory. A successful local check appends the fragment (Δ, κ) and contracts Δ; a failed check returns the replay-valid prefix together with the current local countermodel, as specified in Definition 10. Algorithm 2 gives the resulting iterative procedure and implements the source-component decomposition established in Theorem 3.

Algorithm 2: CHECKMINMRCIRC(φ, P, Z, M, t)

Input: A clause theory φ, disjoint atom sets P, Z, a candidate interpretation M such that $M \models \varphi$, and a strategy $t \in \{\mathrm{SAT}, \mathrm{MUS}\}$
Output: $(\textbf{true}, \mathcal{C})$ or $(\textbf{false}, \mathcal{F})$

$R \leftarrow M \cap P$ // current verification target
$\psi \leftarrow \mathrm{Red}[\varphi; P; Z, M]$ // initial residual reduct
initialize a representative origin map $\mathrm{Orig} : \psi \to \varphi$, choosing one original clause for each residual clause
$\mathcal{C} \leftarrow \langle\rangle$ // origin-preserving decomposition certificate
while $R \neq \emptyset$ **do**
  $\mathbb{S} \leftarrow \mathbb{S}_{\psi[R;Z]}$ // collapsed dependency graph of the current residual theory
  $H \leftarrow \mathbb{S}$ // working copy of the collapsed graph
  while such vertices exist, delete from H all source vertices D with $\mathrm{lab}(D) \cap R = \emptyset$
  select a source vertex C of H with $\mathrm{lab}(C) \cap R \neq \emptyset$ $U \leftarrow \mathrm{Scope}_{\mathbb{S}}(C)$ // scope computed in the current collapsed graph
  $\lambda \leftarrow \psi_U$ // induced local residual theory
  $\Delta \leftarrow \mathrm{lab}(C) \cap R$ // local target
  $\omega \leftarrow \mathrm{Orig}|_\lambda$ // origin map restricted to local clauses
  $(s, \eta) \leftarrow$ LOCALCERTIFY$(\lambda, \Delta, t, \omega)$ // check $\lambda \models \bigwedge \Delta$
  if $s =$ **proved then**
    $\kappa \leftarrow \eta$ // $\kappa \subseteq \varphi$ replays the local proof
    append (Δ, κ) to $\mathcal{C}$ // serialized component block and support
    $R \leftarrow R \setminus \Delta$ // contract verified minimized atoms
    $\psi \leftarrow \mathrm{Red}[\psi; R; Z, M]$ // residual reduct after contraction
    for each new residual clause, choose one predecessor and inherit its representative origin
  else
    $I \leftarrow \eta$ // local countermodel
    $\mathcal{F} \leftarrow (\mathcal{C}, \Delta, I)$ // replayable prefix, stage-local block, and countermodel
    return $(\textbf{false}, \mathcal{F})$
return $(\textbf{true}, \mathcal{C})$

After deleting Δ from R, the update $\psi \leftarrow \mathrm{Red}[\psi; R; Z, M]$ is correct by Proposition 3. The representative origin map records one provenance choice for each residual clause and does not alter the entailment test.

Example 10. *Run Algorithm 2 on Example 8 with* $M_1 = \{p_1, p_2, p_3\}$ *and* $t = \mathrm{MUS}$. *The first selected source component has target* $\Delta_0 = \{p_3\}$ *and local residual theory*

$$\lambda_0 = \{\alpha_4 : p_3 \vee z_1,\ \alpha_5' : \neg z_1\}.$$

As shown in Example 9, this local obligation is certified by the original support

$$\kappa_0 = \{\alpha_4 : p_3 \vee z_1,\ \alpha_5 : q_1 \vee \neg z_1\} \subseteq \varphi.$$

The checker records $(\{p_3\}, \kappa_0)$, *contracts* p_3, *and recomputes the residual reduct. The next non-empty source component has target* $\Delta_1 = \{p_1, p_2\}$, *scope* $U_1 = \{p_1, p_2\}$, *and local residual theory*

$$\lambda_1 = \{\alpha_1 : \neg p_1 \vee p_2,\ \alpha_2 : p_1 \vee \neg p_2,\ \alpha_3' : p_1\}.$$

An inclusion-minimal original support is

$$\kappa_1 = \{\alpha_1 : \neg p_1 \vee p_2,\ \alpha_3 : p_1 \vee \neg p_3\}.$$

At the second stage,

$$\left(\mathrm{Red}[\kappa_1; \{p_1, p_2\}; Z, M_1]\right)_{U_1} = \{\neg p_1 \vee p_2,\ p_1\},$$

and

$$\{\neg p_1 \vee p_2,\ p_1,\ \neg p_1 \vee \neg p_2\} \models \bot.$$

After contracting $\{p_1, p_2\}$, *the verification target is empty and the checker returns* $(\mathbf{true}, \mathcal{C})$, *where*

$$\mathcal{C} = \langle(\{p_3\}, \kappa_0), (\{p_1, p_2\}, \kappa_1)\rangle.$$

Replay first reconstructs the stage-zero scope and verifies

$$\{p_3 \vee z_1, \neg z_1, \neg p_3\} \models \bot.$$

It then contracts $\{p_3\}$, *reconstructs* U_1 *and* λ_1, *and verifies the second unsatisfiability condition displayed above. Thus the example traces the complete sequence from the initial reduct through component selection, support generation, contraction, and certificate replay.*

The parameter t affects only support extraction. The SAT variant uses $H = \lambda$; the MUS variant chooses an inclusion-minimal $H \subseteq \lambda$ with $H \cup \{\bigvee \neg\Delta\} \models \bot$. In both cases, the stored support is $\kappa = \omega(H) \subseteq \varphi$.

Theorem 4 (Soundness and completeness)**.** *Let* φ *be a clause theory over* $\mathcal{A}$, $P, Z \subseteq \mathcal{A}$ *be disjoint atom sets,* $M \models \varphi$, *and* $t \in \{\mathrm{SAT}, \mathrm{MUS}\}$. *Then Algorithm 2 on input* (φ, P, Z, M, t) *returns*

$$(\mathbf{true}, \mathcal{C})$$

if and only if

$$M \models \mathrm{CIRC}[\varphi; P; Z].$$

Equivalently, it returns

$$(\mathbf{false}, \mathcal{F})$$

only if

$$M \not\models \mathrm{CIRC}[\varphi; P; Z].$$

Moreover, every successful certificate fragment κ_i in $\mathcal{C}$ is a subset of the original clause theory φ whose replayed reduct certifies the corresponding local target.

Proof. By Theorem 1,

$$M \models \mathrm{CIRC}[\varphi; P; Z] \iff \mathrm{Red}[\varphi; P; Z, M] \models \bigwedge(M \cap P).$$

At each iteration, Theorem 3 gives the one-step equivalence between the current residual entailment, the scoped obligation, and the entailment for the next reduct. Since every successful step removes a non-empty block from the finite target, repeated application of the theorem reduces the initial entailment to the conjunction of the scoped obligations generated by Algorithm 2. Proposition 6 shows that a return value $(\textbf{proved}, \kappa_i)$ establishes the corresponding obligation and that $(\textbf{refuted}, I)$ supplies a model of its negation. Therefore the algorithm returns **true** exactly when every generated obligation holds. The inclusion $\kappa_i \subseteq \varphi$ and the replay condition follow from Definition 9 and Proposition 6. □

The correctness theorem separates the semantic decision from the form of the recorded support. SAT and MUS certification establish the same local entailments and therefore induce the same Boolean result; they differ only in the residual support selected for replay.

Remark 1. *The semantics is independent of the certification mode. A certificate stores $\kappa \subseteq \varphi$; its validity is the stage-specific replay condition in Definition 9.*

6. Experimental Evaluation

The experimental evaluation addresses Boolean agreement with the global reduct criterion, the relative size of scoped obligations, the size of origin-preserving supports, and certificate replayability.

6.1. Evaluation Questions

Q1. Do the decomposition variants agree with the global entailment test?
Q2. What fraction of the current residual reduct occurs in a scoped obligation?
Q3. How do SAT and MUS certification compare in accumulated support size?
Q4. Does every returned certificate satisfy the replay condition?

6.2. Methods

The following methods are compared.

(i) **GlobalReduct** tests

$$\mathrm{Red}[\varphi; P; Z, M] \models \bigwedge(M \cap P).$$

(ii) **Decomp-SAT** executes Algorithm 2 with $t = \mathrm{SAT}$ and retains the complete local residual theory as the residual support.

(iii) **Decomp-MUS** executes the same algorithm with $t = \mathrm{MUS}$ and extracts an inclusion-minimal residual support before applying the origin map.

The two decomposition methods generate identical scoped obligations; they differ only in support extraction.

6.3. Instances and Implementation

The benchmark set comprises 5445 solved random 3CNF instances and 462 solved industrial CNF instances derived from the data used in [17]. Each input is a quadruple

(φ, P, Z, M) with $M \models \varphi$. The experiments therefore isolate the model-checking and certificate-generation phases.

All runs were performed on an Intel(R) Core(TM) i7-10700F CPU at 2.90 GHz with 8 physical cores, 16 logical cores, and 32 GB of memory under Ubuntu 26.04. PicoSAT 960 was used for both the global entailment test and every local SAT call, while the PicoMUS utility distributed with PicoSAT 960 was used for MUS extraction. The three methods were run sequentially on the same inputs and machine; no portfolio execution or parallel solver calls were used. At each iteration, all current source components disjoint from the target were removed round by round from a working copy of the collapsed graph. The checker recorded the selected component through the set $\Delta = \mathrm{lab}(C) \cap R$; replay reconstructed the current graph and verified that the recorded component was selectable by the same procedure. When several predecessor clauses produced the same residual clause, the implementation selected the first predecessor in the fixed input-clause order. The selected target block was stored in the certificate, so replay did not depend on an external ordering of source components; representative-origin propagation was deterministic.

Reported runtimes are arithmetic means over solved instances and include the complete execution of each compared method. Certificate-size and replay statistics are computed over runs that returned a successful certificate; a replay success is counted only when every fragment of that certificate passes the stage-specific unsatisfiability test. The benchmark collections form fixed corpora rather than samples drawn for statistical inference. Accordingly, the reported means, maxima, factors, and percentage reductions are interpreted as descriptive summaries of these corpora; no population-level confidence claim is made.

6.4. Measurements

For each instance, we record the Boolean result and runtime of each method. For

$$\psi = \mathrm{Red}[\varphi; P; Z, M], \qquad R = M \cap P,$$

we record $|\varphi|$, $|\psi|$, $|P|$, $|Z|$, the number of SCCs, and the largest SCC. At iteration i, we record

$$|U_i|, \quad |\Delta_i|, \quad |\lambda_i|, \quad \frac{|\lambda_i|}{|\psi_i|}.$$

For a successful certificate $\mathcal{C} = \langle(\Delta_i, \kappa_i)\rangle_{i<m}$, we record

$$m, \qquad \sum_{i<m} |\kappa_i|, \qquad \frac{\sum_{i<m} |\kappa_i|}{|\varphi|}.$$

The summation counts a clause once for each certificate fragment in which it occurs. Replay verifies, for every $i < m$,

$$\big(\mathrm{Red}[\kappa_i; R_i; Z, M]\big)_{U_i} \cup \Big\{\bigvee \neg\Delta_i\Big\} \models \bot.$$

6.5. Results

Table 2 reports agreement with the global criterion and mean runtime.

Table 2. Correctness and runtime comparison.

Family	#Inst.	Global Solved	SAT Correct	MUS Correct	Global Time ms	SAT Time ms	MUS Time ms
Random 3CNF	5445	5445	5445	5445	0.046	16.271	157.274
Industrial CNF	462	462	462	462	29.563	34,435.429	539,509.732

Table 3 reports the initial reduct and graph statistics.

Table 3. Reduct and dependency-graph structure.

Family	Avg. $\|\varphi\|$	Avg. $\|\psi\|$	Avg. $\|P\|$	Avg. $\|Z\|$	Avg. #SCC	Avg. Max SCC	Avg. Reduct Ratio
Random 3CNF	1827.893	555.589	156.545	156.545	119.750	255.379	0.302
Industrial CNF	156,871.643	37,323.433	6263.712	6273.506	3263.699	10,145.747	0.255

Table 4 gives the scoped-obligation statistics.

Table 4. Local obligation sizes.

Family	Mode	Avg. $\|U_i\|$	Avg. $\|\Delta_i\|$	Avg. $\|\lambda_i\|$	Avg. $\|\lambda_i\|/\|\psi_i\|$	Max. $\|\lambda_i\|$
Random 3CNF	SAT	79.747	2.486	157.029	0.407	2483
Random 3CNF	MUS	79.747	2.486	157.029	0.407	2483
Industrial CNF	SAT	3312.655	8.334	10,244.507	0.272	385,375
Industrial CNF	MUS	3312.655	8.334	10,244.507	0.272	385,375

Table 5 reports support size and replay.

Table 5. Origin-preserving certificate statistics over successful certificate-producing runs.

Family	Mode	Avg. $\|\mathcal{C}\|$	Avg. $\sum_i \|\kappa_i\|$	Avg. $\sum_i \|\kappa_i\|/\|\varphi\|$	Reduction vs. SAT (%)	Replay Success (%)	Avg. Replay Time ms
Random 3CNF	SAT	24.757	3887.578	1.723	–	100.000	6.354
Random 3CNF	MUS	24.757	100.541	0.055	97.414	100.000	1.924
Industrial CNF	SAT	175.900	1,802,013.264	6.838	–	100.000	8152.697
Industrial CNF	MUS	175.900	2760.537	0.030	99.847	100.000	1835.944

6.6. Analysis

All 5907 instances satisfy

$$\text{DecompSAT}(\varphi, P, Z, M) = \text{DecompMUS}(\varphi, P, Z, M) = \text{GlobalReduct}(\varphi, P, Z, M).$$

Hence all three methods return identical Boolean values on the complete test set.

The mean initial reduct ratios are 0.302 for random 3CNF and 0.255 for industrial CNF. The corresponding mean scoped ratios $|\lambda_i|/|\psi_i|$ are 0.407 and 0.272. The mean target sizes are 2.486 and 8.334. These ratios quantify the mean fraction of the current residual theory retained by a source-component obligation. The maximum industrial scoped theory contains 385,375 clauses, indicating that a large SCC or ancestor scope can eliminate most of the potential localization.

For random 3CNF, MUS extraction decreases the mean accumulated support from 3887.578 to 100.541; the mean SAT support is therefore 38.67 times the mean MUS support, corresponding to a reduction of approximately 97.4%. For industrial CNF, the mean decreases from 1802013.264 to 2760.537; the mean SAT support is 652.78 times the mean MUS support, and the relative reduction is approximately 99.85%. Every returned certificate satisfies the replay condition.

The runtime data distinguish Boolean decision from certificate construction. GlobalReduct performs a single entailment test. The decomposition methods also construct scopes, recompute reducts, maintain origin maps, and replay local supports; Decomp-MUS additionally performs MUS extraction. The measurements therefore identify the intended trade-off: the global criterion remains the faster procedure when only the Boolean decision is required, whereas the decomposition methods incur additional cost to produce local, replayable proof objects.

7. Discussion

The extent of localization is governed by the collapsed graph $\mathbb{S}_{\psi[R;Z]}$, where $\psi = \mathrm{Red}[\varphi; P; Z, M]$ and $R = M \cap P$, together with the corresponding residual graphs at later stages. Small selected components with small ancestor scopes induce local theories substantially smaller than the current residual reduct. Conversely, a dominant SCC or a large ancestor scope yields an obligation close in size to the current residual theory. This limitation follows directly from the scope definition and is reflected in the maximum industrial scoped-theory size.

The certificate is independent of transient SCC and clause identifiers. Each fragment stores a stage-local dependency block Δ_i and a set $\kappa_i \subseteq \varphi$. Replay reconstructs the residual state from the preceding blocks, identifies the current component C_i, derives its scope U_i and local theory, and then verifies the stage-specific unsatisfiability condition. A single representative origin per residual clause is sufficient because the checker constructs one support and does not enumerate all provenance alternatives.

The current implementation recomputes the residual reduct and collapsed graph after every successful obligation. It also performs MUS extraction independently at each stage. Incremental reduct maintenance, incremental SCC updates, and proof-producing SAT/MUS interfaces are the principal directions for reducing this overhead.

8. Conclusions

We established a source-component decomposition of the minimal-reduct entailment characterizing propositional circumscription models. The repeated deletion of target-free source components guarantees that a target-bearing source is eventually exposed whenever the current verification target is non-empty. The scoped-countermodel extension lemma establishes semantic locality over ancestor scopes, and the contraction results yield equivalence between the sequence of scoped obligations and the global entailment.

The resulting algorithm is sound and complete. A successful run returns

$$\mathcal{C} = \langle (\Delta_i, \kappa_i) \rangle_i, \qquad \kappa_i \subseteq \varphi,$$

with a formally specified replay condition. A failed run returns the replayable successful prefix, the stage-local target at which checking fails, and a scoped countermodel.

On the evaluated instances, the SAT and MUS variants agree with the global criterion, and every returned certificate satisfies the replay condition. MUS extraction yields substantially smaller original-clause supports at the cost of higher runtime.

The following directions warrant further investigation.

(i) *Incremental maintenance.* Develop incremental algorithms for maintaining the residual reduct and collapsed dependency graph. A corresponding complexity analysis should bound the update cost in terms of the clauses and strongly connected components affected by each contraction.

(ii) *Parallel certification.* Investigate parallel certification of incomparable source components. Correctness requires sufficient conditions under which local contractions commute and independently generated certificate fragments admit deterministic composition.

(iii) *Extensions of the semantic setting.* Extend the decomposition to prioritized or nested circumscription and, over finite or bounded domains, to first-order circumscription. Such extensions require suitable reduct characterizations and scope-preservation theorems.

Author Contributions: Methodology, Z.X.; software, X.Z.; validation, X.Z.; formal analysis, H.H.; writing—original draft preparation, Z.X.; writing—review and editing, X.D.; funding acquisition, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Tower Base Foundation Project of Chongqing University of Arts and Sciences, China (Grant No. R2025KJ14).

Data Availability Statement: The benchmark data used in the experiments are derived from the computational results reported for the minimal-reduct approach to propositional circumscription [17]. Additional generated logs and processed tables can be made available upon request.

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

References

1. McDermott, D.V.; Doyle, J. An Introduction to Non-Monotonic Logic. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence, IJCAI 79, Tokyo, Japan, 20–23 August 1979*; Buchanan, B.G., Ed.; William Kaufmann: Los Altos, CA, USA, 1979; Volume 2, pp. 562–567.
2. McCarthy, J. Circumscription—A Form of Non-Monotonic Reasoning. *Artif. Intell.* **1980**, *13*, 27–39. [CrossRef]
3. McCarthy, J. Applications of Circumscription to Formalizing Common-Sense Knowledge. *Artif. Intell.* **1986**, *28*, 89–116. [CrossRef]
4. Reiter, R. Chapter 12—Nonmonotonic Reasoning. In *Exploring Artificial Intelligence*; Shrobe, H.E., the American Association for Artificial Intelligence, Eds.; Morgan Kaufmann: San Mateo, CA, USA, 1988; pp. 439–481. [CrossRef]
5. Lifschitz, V. Computing Circumscription. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, USA, 18–23 August 1985*; Joshi, A.K., Ed.; Morgan Kaufmann Publishers Inc.: Los Altos, CA, USA, 1985; Volume 1, pp. 121–127.
6. Przymusinski, T.C. An Algorithm to Compute Circumscription. *Artif. Intell.* **1989**, *38*, 49–73. [CrossRef]
7. de Kleer, J.; Konolige, K. Eliminating the Fixed Predicates from a Circumscription. *Artif. Intell.* **1989**, *39*, 391–398. [CrossRef]
8. Doherty, P.; Lukaszewicz, W.; Szalas, A. Computing Circumscription Revisited: A Reduction Algorithm. *J. Autom. Reason.* **1997**, *18*, 297–336. [CrossRef]
9. Janhunen, T.; Oikarinen, E. Capturing Parallel Circumscription with Disjunctive Logic Programs. In Proceedings of the 9th European Conference on Logics in Artificial Intelligence, Lisbon, Portugal, 27–30 September 2004; pp. 134–146.
10. Wan, H.; Xiao, Z.; Yuan, Z.; Zhang, H.; Zhang, Y. Computing General First-Order Parallel and Prioritized Circumscription. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014*; Brodley, C.E., Stone, P., Eds.; AAAI Press: Palo Alto, CA, USA, 2014; pp. 1105–1111.
11. Eiter, T.; Gottlob, G. Propositional Circumscription and Extended Closed World Reasoning Are Π_2^p-Complete. *Theor. Comput. Sci.* **1993**, *114*, 231–245. [CrossRef]
12. Cadoli, M. The Complexity of Model Checking for Circumscriptive Formulae. *Inf. Process. Lett.* **1992**, *44*, 113–118. [CrossRef]
13. Cadoli, M.; Schaerf, M. A Survey of Complexity Results for Non-Monotonic Logics. *J. Log. Program.* **1993**, *17*, 127–160. [CrossRef]
14. Borrego-Díaz, J.; Cordón-Franco, A.; Lara-Martín, F.F. On Conditional Axioms and Associated Inference Rules. *Axioms* **2024**, *13*, 306. [CrossRef]
15. Lin, Z.; Ma, M. Cut-Free Gentzen Sequent Calculi for Tense Logics. *Axioms* **2023**, *12*, 620. [CrossRef]
16. Yu, X.; Li, Y.; Geng, S.; Li, H. Fuzzy Computation Tree Temporal Logic with Quality Constraints and Its Model Checking. *Axioms* **2024**, *13*, 832. [CrossRef]
17. Xie, Z.; Wang, Y.; Yang, L.; Feng, R. Minimal reduct for propositional circumscription. *Front. Artif. Intell.* **2025**, *8*, 1614894. [CrossRef] [PubMed]
18. Cadoli, M.; Eiter, T.; Gottlob, G. An Efficient Method for Eliminating Varying Predicates from a Circumscription. *Artif. Intell.* **1992**, *54*, 397–410. [CrossRef]
19. Cadoli, M.; Lenzerini, M. The Complexity of Propositional Closed World Reasoning and Circumscription. *J. Comput. Syst. Sci.* **1994**, *48*, 255–310. [CrossRef]
20. Sakama, C.; Inoue, K. Embedding Circumscriptive Theories in General Disjunctive Programs. In Proceedings of the LPNMR, Lexington, KY, USA, 26–28 June 1995; pp. 344–357.
21. Zhang, H.; Zhang, Y.; Ying, M.; Zhou, Y. Translating First-Order Theories into Logic Programs. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*; Walsh, T., Ed.; AAAI Press: Palo Alto, CA, USA, 2011; pp. 1126–1131. [CrossRef]
22. Rajasekar, A.; Lobo, J.; Minker, J. Weak Generalized Closed World Assumption. *J. Autom. Reason.* **1989**, *5*, 293–307. [CrossRef]
23. Gelfond, M.; Lifschitz, V. Classical Negation in Logic Programs and Disjunctive Databases. *New Gener. Comput.* **1991**, *9*, 365–385. [CrossRef]
24. Ferraris, P.; Lee, J.; Lifschitz, V. Stable Models and Circumscription. *Artif. Intell.* **2011**, *175*, 236–263. [CrossRef]
25. Lee, J.; Lin, F. Loop formulas for circumscription. *Artif. Intell.* **2006**, *170*, 160–185. [CrossRef]

26. Ben-Eliyahu-Zohary, R.; Angiulli, F.; Fassetti, F.; Palopoli, L. Decomposing Minimal Models. In *Proceedings of the Workshop on Knowledge-Based Techniques for Problem Solving and Reasoning Co-Located with 25th International Joint Conference on Artificial Intelligence*; Barták, R., McCluskey, T.L., Pontelli, E., Eds.; CEUR Workshop Proceedings: Aachen, Germany, 2016; Volume 1648.
27. Calimeri, F.; Perri, S.; Zangari, J. Optimizing Answer Set Computation via Heuristic-Based Decomposition. *Theory Pract. Log. Program.* **2019**, *19*, 603–628. [CrossRef]
28. Comploi-Taupe, R.; Friedrich, G.; Schekotihin, K.; Weinzierl, A. Domain-Specific Heuristics in Answer Set Programming: A Declarative Non-Monotonic Approach. *J. Artif. Intell. Res.* **2023**, *76*, 59–114. [CrossRef]
29. Angiulli, F.; Ben-Eliyahu-Zohary, R.; Fassetti, F.; Palopoli, L. Graph-based construction of minimal models. *Artif. Intell.* **2022**, *313*, 103754. [CrossRef]
30. Beyersdorff, O.; Chew, L. The Complexity of Theorem Proving in Circumscription and Minimal Entailment. In *Proceedings of the Automated Reasoning—7th International Joint Conference, IJCAR 2014*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8562, pp. 403–417. [CrossRef]
31. Ansótegui, C.; Bonet, M.L.; Levy, J. A New Algorithm for Weighted Partial MaxSAT. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, GA, USA, 11–15 July 2010*; Fox, M., Poole, D., Eds.; AAAI Press: Palo Alto, CA, USA, 2010; pp. 3–8. [CrossRef]
32. Ansótegui, C.; Bonet, M.L.; Levy, J. SAT-based MaxSAT algorithms. *Artif. Intell.* **2013**, *196*, 77–105. [CrossRef]
33. Alviano, M.; Dodaro, C. Unsatisfiable Core Analysis and Aggregates for Optimum Stable Model Search. *Fundam. Informaticae* **2020**, *176*, 271–297. [CrossRef]
34. Py, M.; Cherif, M.S.; Habet, D. Proofs and Certificates for Max-SAT. *J. Artif. Intell. Res.* **2022**, *75*, 1373–1400. [CrossRef]
35. Alviano, M.; Dodaro, C.; Fichte, J.K.; Hecher, M.; Philipp, T.; Rath, J. Inconsistency Proofs for ASP: The ASP-DRUPE Format. *Theory Pract. Log. Program.* **2019**, *19*, 891–907. [CrossRef]
36. Pontelli, E.; Son, T.C.; El-Khatib, O. Justifications for Logic Programs Under Answer Set Semantics. *Theory Pract. Log. Program.* **2009**, *9*, 1–56. [CrossRef]
37. Cabalar, P.; Fandinno, J.; Fink, M. Causal Graph Justifications of Logic Programs. *Theory Pract. Log. Program.* **2014**, *14*, 603–618. [CrossRef]
38. Cabalar, P.; Fandinno, J. Justifications for Programs with Disjunctive and Causal-choice Rules. *Theory Pract. Log. Program.* **2016**, *16*, 587–603. [CrossRef]
39. Schulz, C.; Toni, F. Justifying Answer Sets Using Argumentation. *Theory Pract. Log. Program.* **2016**, *16*, 59–110. [CrossRef]
40. Denecker, M.; Brewka, G.; Strass, H. A Formal Theory of Justifications. In *Proceedings of the Logic Programming and Nonmonotonic Reasoning*; Calimeri, F., Ianni, G., Truszczynski, M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 250–264. [CrossRef]
41. Wang, Y.; Eiter, T.; Zhang, Y.; Lin, F. Witnesses for Answer Sets of Logic Programs. *ACM Trans. Comput. Log.* **2023**, *24*, 1–46. [CrossRef]