On the Tractability of Un/Satisfiability

Latif Salum

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 21, 2019
On the Tractability of Un/Satisfiability

Latif Salum
Department of Industrial Engineering, Dokuz Eylul University, Izmir, Turkey
latif.salum@deu.edu.tr & latif.salum@gmail.com

Abstract

This paper shows $P = NP$ via exactly-1 3SAT (X3SAT). $C_k = (r_i \circ r_j \circ r_u)$ denotes a clause, an exactly-1 disjunction $\circ$ of literals, such that $\phi = \bigwedge C_k$, an X3SAT formula. $\phi(r_j) := r_j \land \phi$ denotes that the literal $r_j$ is true, $r_j \in \{x_i, \overline{x_i}\}$. This truth assignment leads to reductions due to $\circ$ of any $C_k = (r_j \circ \overline{x_i} \circ x_u)$ into $C_k = r_j \land \overline{x_i} \land x_u$, and $C_k = (r_j \circ \overline{x_u} \circ x_u)$ into $C_k = (r_u \circ r_u)$. As a result, $\phi(r_j) := r_j \land \phi$ transforms into $\phi(r_j) = \psi(r_j) \land \phi'(r_j)$, unless $\psi(r_j)$ involves $x_i \land \overline{x_u}$, that is, unless $\lnot \psi(r_j)$. Then, $\psi(r_j) = \bigwedge (e_1 \land C_k)$ such that $C_{k'} = r_i$, and $\phi'(r_j) = \bigwedge (C_k \land C_{k'})$. Thus, $\psi(r_j)$ and $\phi'(r_j)$ are disjoint. It is trivial to check if $\lnot \psi(r_j)$, and redundant to check if $\lnot \phi'(r_j)$, in order to verify $\lnot \phi'(r_j)$. Proof of this redundancy is sketched as follows. $\psi(r_j)$ is true, $\psi(r_i) = \psi(r_i | r_j)$ holds, hence $\psi(r_i | r_j)$ is true for every $r_i$, because each $r_j$ such that $\lnot \psi(r_j)$ is removed from $\phi$. Then, any $\overline{x_i}$ consists in $\psi$ so that $\phi$ transforms into $\psi \land \phi'$. If $\psi$ involves $x_j \land \overline{x_i}$, then $\lnot \phi$. Otherwise, $\phi$ is satisfied, since any $\psi(\cdot)$ is disjoint and true, and $\psi(r_1 | r_1), \psi(r_1 | r_2), \ldots, \psi(r_u | r_u)$ compose $\phi$. Thus, $\phi'(r_j)$ is satisfied, since $(r_j \land \phi') \equiv (\psi(r_j) \land \phi'(r_j))$. The time complexity is $O(mn^3)$, hence $P = NP$.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic

Keywords and phrases $P \lor NP$, NP-complete, 3SAT, one-in-three SAT, exactly-1 3SAT, X3SAT

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Acknowledgements

Introduction: Effectiveness of X3SAT in proving $P = NP$

$P = NP$ is the most notorious problem in theoretical computer science. It is well known that $P = NP$, if there exists a polynomial time algorithm for any one of NP-complete problems, since algorithmic efficiency of these problems is equivalent. Nevertheless, some NP-complete problem features algorithmic effectiveness, if it incorporates an effective tool to develop an efficient algorithm. That is, a particular problem can be more effective to prove $P = NP$.

This paper shows that one-in-three SAT, which is NP-complete [2], features algorithmic effectiveness to prove $P = NP$. This problem is also known as exactly-1 3SAT (X3SAT). X3SAT incorporates “exactly-1 disjunction $\circ$”, the tool used to develop a polynomial time algorithm. It facilitates checking incompatibility of a literal $r_j$ for satisfying some formula $\phi$. When every $r_j$ incompatible is removed, $\phi$ becomes un/satisfiable. Thus, each $r_j$ becomes compatible to participate in some satisfiable assignment. Then, an assignment is constructed.

The truth assignment $r_j = T$ (or $r_j$) is incompatible if $\phi(r_j)$ is unsatisfiable, denoted by $\lnot \phi(r_j)$, where $\phi(r_j) := r_j \land \phi$, and $r_j \in \{x_i, \overline{x_i}\}$. Then, the $\phi$ scan algorithm, introduced below, “scans” $\phi$ by checking incompatibility of every $r_i$, and removing each $r_j$ incompatible.

Let $\phi = C_1 \land \cdots \land C_m$ be any X3SAT formula such that a clause $C_k = (r_i \circ r_j \circ r_u)$ is an exactly-1 disjunction $\circ$ of literals $r_i$, hence satisfied iff exactly one of $\{r_i, r_j, r_u\}$ is true.

Note that a clause $\{r_i \lor r_j \lor r_u\}$ in a 3SAT formula is satisfied iff at least one of them is true.
Incompatibility of $r_j$ is checked by a deterministic chain of reductions of any $C_k$ in $\phi(r_j)$, which is constructed via $\circ$. This chain is initiated by $r_j = T$, and followed by $\neg r_j$, because $r_j \Rightarrow \neg r_j$. That is, each $(r_j \circ x_u \circ \neg x_u)$ collapses to $(r_j \land x_u \land \neg x_u)$ due to $r_j \Rightarrow r_j \land x_u \land \neg x_u$, since there exists exactly one true literal in any clause $C_k$ by the definition of 3ISAT. Also, each $(\neg x_u \circ x_u \circ \neg x_u)$ shrinks to $\neg x_u$ due to $\neg r_j$. Thus, $r_j$ transforms $\phi(r_j) := r_j \land \phi$ into $\phi(r_j) := r_j \land x_u \land \neg x_u \land \phi^*$, and $x_u \land \neg x_u$ proceeds the reductions in $\phi^*$, which involves $(\neg x_u \circ x_u)$. The reductions over $\phi_u(r_j)$ terminate iff $r_j \land \phi_u$ transforms into $\psi_u(r_j) \land \phi_u(r_j)$ such that $\psi_u(r_j)$ and $\phi_u(r_j)$ are disjoint, where $s$ denotes the current scan, and $\psi_u(r_j)$ is a conjunction of literals that are true. They are interrupted iff $\psi_u(r_j)$ involves $x_i \land \neg x_i$, thus $\phi \neq \phi_u(r_j)$, that is, $r_j$ is incompatible. By assumption, $\phi \neq \phi_u(r_j)$ is verified solely via $\neq \psi_u(r_j)$ (see Figure 1).

The reductions over $\phi$ terminate iff $\phi$ transforms into $\psi \land \phi'$ such that $\psi$ and $\phi'$ are disjoint, where $\psi = \neg x_i \land x_u \land \cdots \land \neg x_u$ (see Figure 1). Then, $\phi$ is updated, that is, $\phi \leftarrow \phi'$. The $\phi_s$ scan is interrupted if $\psi_s$ involves $x_i \land \neg x_i$ for some $i$, hence $\phi \neq \phi_s$, that is, $\phi$ is unsatisfiable. Hence, $\neq \psi_u(r_j)$ is verified solely by $\neq \psi_u(r_j)$, and whether $\neq \phi_u(r_j)$ is ignored.

Claim 1. It is redundant to check if $\neq \phi_u(r_j)$, thus $\neq \phi(r_j)$ iff $\neq \phi_u(r_j)$ iff $\neq \psi_s(r_j)$ for some $s$. As a result, $\phi(r_j)$ reduces to $\phi_u(r_j) = \psi_u(r_j) \land \phi_u(r_j)$, hence $\psi_u(r_j) = \psi_u(r_j)$ holds. Therefore, $\phi$ is satisfiable iff any truth assignment $\psi(r_j)$ holds (the scan terminates).

Sketch of proof. $\psi(r_j) / \psi(r_j)$ is constructed over $\phi / \phi'(r_j)$, thus $\psi(r_j)$ covers $\psi(r_j)$, hence $\psi(r_j) \models \psi(r_j)$. Because $\psi(r_j)$ and $\phi'(r_j)$ are disjoint, $\psi(r_j)$ and $\psi(r_j)$ are disjoint (see Figure 2). Therefore, $\psi(r_j)$, $\psi(r_j)$, $\psi(r_j)$, $\psi(r_j)$, $\psi(r_j)$, and $\psi(r_j)$ form disjoint minterms $\psi(r_j)$ over $\phi$ such that $\psi(r_j)$, $\psi(r_j)$, $\psi(r_j)$, $\psi(r_j)$, and $\psi(r_j)$ are true, because $\psi(r_j)$ is true for every $r_i$ (the $\phi$ scan terminates), and $\psi(r_j)$ holds. Thus, $\phi$ is composed of $\psi(r_j)$ that are disjoint and true (see Figure 3), hence $\phi$ is satisfied.

A satisfiable assignment $\alpha$ is constructed by composing $\psi(r_j)$ that are disjoint and true. For example, $\alpha = \{ \psi(r_j), \psi(r_j), \psi(r_j), \psi(r_j), \psi(r_j) \}$ (see Figure 3).
2 Basic Definitions

A literal \( r_i \) is a variable \( x_i \) or its negation \( \bar{x}_i \), i.e., \( r_i \in \{ x_i, \bar{x}_i \} \). A clause \( C_k = (r_i \lor r_j \lor r_u) \), or \( C_k = (r_{ik} \lor r_{jk} \lor r_{ku}) \), is an exactly-1 disjunction \( \lor \) of literals that are assumed to be true.

- **Definition 2** (Minterm). \( c_k = \bigwedge r_i \) is a conjunction of literals that are true, hence \( c_k \) is true.

- **Definition 3** (X3SAT formula). \( \phi = \psi \land \phi \) such that \( \psi = \bigwedge c_k \) and \( \phi = \bigwedge C_k \).

Any \( r_i \) in \( \psi \) denotes a *conjunct*, which is necessary (\( r_i = T \)) for satisfying \( \phi \), since \( c_k = T \) by definition. If \( r_i \) is necessary, then \( \bar{r}_i \) is incompatible/removed from \( \phi \), i.e., \( r_i \Rightarrow \neg \bar{r}_i \), while \( r_i \) is incompatible/removed if the assumption \( r_i = T \) cannot hold. That is, if \( r_i \Rightarrow x_j \land \bar{x}_j \), hence \( \neg x_j \lor \neg \bar{x}_j \Rightarrow \neg r_i \), then \( r_i \) is removed from \( \phi \) and \( \bar{r}_i \) is necessary (\( \bar{r}_i = T \)), i.e., \( \neg r_i \Rightarrow \bar{r}_i \).

Where appropriate, \( C_k \), as well as \( \psi \), is denoted by a set. Thus, \( \psi = \psi \land \phi \) the formula, that is, \( \phi = \psi \land C_1 \land C_2 \land \cdots \land C_m \), is denoted by \( \psi = \{ \psi, C_1, C_2, \ldots, C_m \} \) the family of sets.

\( \mathcal{L} = \{ 1, 2, \ldots, n \} \) denotes the index set of the literals \( r_i \) in \( \psi \), and \( \mathcal{C} = \{ 1, 2, \ldots, m \} \) is an index set of the clauses \( C_k \) in \( \phi \), while \( \mathcal{C} = \{ k \in \mathcal{C} \mid r_i \in C_k \} \) denotes \( C_k \) that contain \( r_i \).

- **Example 4.** Let \( \hat{\phi} = (x_{11} \land \bar{x}_{31}) \land (x_{12} \land \bar{x}_{22} \land x_{12}) \land (x_{23} \land \bar{x}_{13} \land \bar{x}_{43}) \land \bar{x}_4 \). Note that \( C_4 = (x_1 \land \bar{x}_4 \land \bar{x}_3) \land (x_2 \land \bar{x}_4) \land (x_3 \land \bar{x}_3) \land (x_4 \land \bar{x}_4) \). Then, \( \hat{\phi}_4 = \emptyset \), and \( \mathcal{C} = \{ x_1, x_3 \} \), \( \mathcal{C}_2 = \{ x_1, x_2 \} \), and \( \mathcal{C}_3 = \{ x_2, x_3 \} \), while \( \psi = \{ x_4 \} \).

- **Definition 5** (Collapse). A clause \( C_k = (r_i \land x_j \land \bar{x}_u) \) is said to collapse to the minterm \( C_k = (r_i \land \bar{x}_j \land x_u) \), thus \( r_i \notin C_k \) if \( r_i \) is necessary, denoted by \( (r_i \land x_j \land \bar{x}_u) \Rightarrow (r_i \land \bar{x}_j \land x_u) \).

- **Definition 6** (Shrinkage). A clause \( C_k = (r_i \lor r_j \lor r_u) \) is said to shrink to another clause \( C_k = (r_j \lor r_u) \), if \( \neg r_i \) (\( r_i \) the incompatible is removed), denoted by \( (r_i \lor r_j \lor r_u) \Rightarrow (r_j \lor r_u) \).

- **Definition 7** (Truth assignment \( r_i = T \) over \( \phi \)). \( \phi(r_i) = r_i \lor \phi \) for any \( r_i \in C_k \) and \( C_k \in \phi \).

- **Note 8.** \( r_i \) is necessary for \( \phi(r_i) \), hence \( r_i \) is removed, i.e., \( r_i \Rightarrow \neg \bar{r}_i \). Then, by the definition of X3SAT, \( r_i \Rightarrow r_i \land \neg x_j \land \neg \bar{x}_u \) to satisfy a clause \( (r_i \lor x_j \lor \bar{x}_u) \). As a result, \( \neg r_i \Rightarrow \bar{x}_u \) and \( \neg \bar{x}_u \Rightarrow x_u \), thus \( \bar{x}_u \) and \( x_u \) become necessary. Therefore, the truth assignment \( \phi(r_i) \) results in \( (r_i \land x_j \land \bar{x}_u) \Rightarrow (r_i \land \bar{x}_j \land x_u) \) and \( (r_i \land \bar{x}_j \land x_u) \Rightarrow (r_u \lor x_u) \) due to Definition 5 and 6.

- **Remark (Reduction).** The collapse or shrinkage of any clause \( C_k \) denotes its reduction, which in turn reduces \( \hat{\phi}_r \), denoted by \( \hat{\phi}_r \Rightarrow \hat{\phi}_{r+1} \). Then, the number of \( C_k \in \hat{\phi}_{r+1} \) is less than the number of \( C_k \in \hat{\phi}_r \), or the number of literals in some \( C_k \in \hat{\phi}_{r+1} \) is less than that in some \( C_k \in \hat{\phi}_r \). Also, a collapse reduces nondeterminism to construct a satisfiable assignment.

- **Definition 9.** \( \phi \) denotes a general formula if \( \{ x_i, \bar{x}_i \} \notin C_k \) for any \( i \in \mathcal{L} \) and \( k \in \mathcal{C} \), hence \( \mathcal{C} \land \mathcal{C}^T = \emptyset \). \( \phi \) denotes a special formula if \( \{ x_i, \bar{x}_i \} \subseteq C_k \) for some \( k \), hence \( \mathcal{C} \land \mathcal{C}^T = \{ k \} \).

The \( \varphi \) scan algorithm accepts a general formula \( \phi \). Recall that \( \varphi = \psi \land \phi \).

- **Lemma 10** (Conversion of a special formula). Each clause \( C_k = (r_j \lor x_i \lor \bar{x}_i) \) is replaced by the conjunct \( \bar{x}_i \) so that \( \mathcal{C} \land \mathcal{C}^T = \emptyset \) for any \( i \in \mathcal{L} \), if \( \phi = \bigwedge C_k \) is a special formula.

**Proof.** \( \phi \) is unsatisfiable due to \( r_j \Rightarrow \bar{x}_j \land x_j \). Then, \( x_j \lor \bar{x}_j \Rightarrow \bar{x}_i \). That is, \( \bar{x}_i \) is necessary for satisfying \( C_k = (r_j \lor x_i \lor \bar{x}_i) \), which is sufficient also, thus \( \bar{x}_i \) is equivalent to \( C_k \). Therefore, each clause \( C_k = (r_j \lor x_i \lor \bar{x}_i) \) is replaced by the conjunct \( \bar{x}_i \) so that \( \mathcal{C} \land \mathcal{C}^T = \emptyset \).

- **Example 11.** \( \phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_3 \lor x_4) \land (x_2 \lor x_3) \) denotes a special formula due to \( C_1 = \{ x_1, x_2, x_3 \} \). Note that \( \mathcal{C} \land \mathcal{C}^T = \{ x_1 \} \). As a result, \( \phi \) is converted by replacing the clause \( C_1 \) with the conjunct \( \bar{x}_1 \). Therefore, \( \phi \Rightarrow \bar{x}_1 \land (x_1 \lor x_3 \lor x_4) \land (x_2 \lor x_3) \). Likewise, \( \phi = (x_1 \lor x_3 \lor x_4) \land (x_1 \lor x_3 \lor x_4) \land (x_2 \lor x_3) \) then \( \phi \Rightarrow \bar{x}_1 \land x_1 \land (x_2 \lor x_3) \). On the other hand, if \( \phi \) involves \( (x_u \lor x_i \lor x_j) \land (x_u \lor x_j \lor \bar{x}_j) \), then \( \phi \) is unsatisfiable due to \( x_u \land x_u \).
This section addresses the \( \varphi \) scan. Section 3.2 introduces the core algorithms. Section 3.3
tackles satisfiability of \( \varphi \), and Section 3.4 tackles construction of a satisfiable assignment.

\( \varphi_s \) denotes the current formula at the \( s \)th scan/step, if \( \neg r_j \) (an incompatible \( r_j \) is removed).

Note that \( \varphi := \varphi_1 \) and \( \varphi \equiv \varphi \). Then, \( \varphi_s = (r_{i_k_1} \odot r_{u_k_1} \odot r_{v_k_1}) \land \cdots \land (r_{i_k_n} \odot r_{u_k_n} \odot r_{v_k_n}) \)
denotes the formula over clauses \( C_k \ni r_i \in \varphi_s \), where \( r_i \in \{x_i, \overline{x_i}\} \). Hence, \( \varphi_s = \{k_1, \ldots, k_n\} \).

\( \models \alpha \varphi \) denotes that the assignment \( \alpha = \{r_1, r_2, \ldots, r_n\} \) satisfies \( \varphi \), and \( \not\models \varphi \) denotes \( \varphi \) is unsatisfiable, while \( \models \psi \cdot \models \psi' \) denotes \( \psi \leq \psi' \) as the logical consequence of \( \psi \) as \( \psi = \mathbb{T}, \psi' = \mathbb{T} \).

\( \hat{\varphi}_s(r_i) \) is called the local effect of \( r_i \) and \( \hat{\varphi}_s(-r_i) \) is the effect of \( \neg r_i \). \( \hat{\varphi}_s(r_i) \) denotes its overall effect such that \( \hat{\varphi}_s(r_i) = \hat{\varphi}_s(r_i) \land \hat{\varphi}_s(-r_i) \), specified below. Also, \( \models \varphi_s(r_i) = \bigwedge (C_k \land C_k) \)
such that \( |C_k| = 1 \). Moreover, \( \models \varphi_s(-r_i) = \bigwedge C_k \) such that \( |C_k| > 1 \), or \( \models \varphi_s(-r_i) \) is empty.

### 3.1 Introduction: Incompatibility and Reductions

Example 12 and 13 introduces incompatibility and reductions, which drive the \( \varphi \) scan.

**Example 12.** Consider \( \varphi(x_1) \) over \( \varphi = \varphi = (x_1 \odot \varphi_3) \land (x_1 \odot \varphi_2 \odot x_3) \land (x_2 \odot \varphi_4) \). Thus, \( x_1 \)
is necessary (see Note 8), hence \( x_1 \models \psi(x_1) \) such that \( \psi(x_1) = (x_1 \land \varphi_3) \land (x_1 \land x_2 \land \varphi_2) \). That is, \( x_1 \Rightarrow \varphi_3 \) holds for \( C_1 = \{x_1 \odot \varphi_3\} \), hence \( \neg \varphi_3 \Rightarrow x_1 \). Likewise, \( x_1 \Rightarrow \neg \varphi_3 \land \neg \varphi_2 \) holds for \( C_2 = \{x_2 \odot \varphi_2 \odot x_3\} \), hence \( \neg \varphi_2 \Rightarrow x_1 \) and \( \neg \varphi_2 \Rightarrow x_2 \). Thus, \( \hat{\varphi}_1(x_1) = \hat{\varphi}_1(x_1) \land \hat{\varphi}_1(-x_2) \) becomes the overall effect, where \( \hat{\varphi}_1(-x_2) \) is empty. Then, the reductions initiated by \( x_1 \) are to proceed due to \( x_2 \). Nevertheless, they are interrupted by \( x_3 \land \varphi_3 \) due to \( \hat{\varphi}_1(x_1) \), hence \( \not\models \hat{\varphi}_1(x_1) \), where \( \models \hat{\varphi}_1(x_1) = \models \varphi_1(x_1) \land (x_2 \odot \varphi_2) \). Therefore, \( x_1 \) is incompatible and removed from \( \varphi \), thus \( \not\models x_1 \Rightarrow \varphi_1 \).

**Example 13.** \( \varphi_1 \) initiates reductions over \( \varphi \) (see Note 8). Then, \( \hat{\varphi}(\varphi_1) = \varphi_1 \land \varphi_2 \), \( \hat{\varphi}(-x_1) = (\varphi_2 \land x_1), \) and \( \hat{\varphi}(\{\varphi_2\}) = \hat{\varphi}(\varphi_1) \land \hat{\varphi}(-x_1) \) such that \( \hat{\varphi}_2 = \hat{\varphi}_2(x_1 \land x_2 \land \varphi_2) \). Note that \((x_2 \land \varphi_2) \) is beyond \( \hat{\varphi}(\varphi_1) \) the overall effect. Note also that \( \varphi_2 \not\models \varphi_2(-x_1) \), since \( \varphi_3 \in \hat{\varphi}(\varphi_1) \), because \( \varphi_1 \Rightarrow \varphi_3 \), since \( \models \varphi_3 \) contains no singleton. Then, \( \varphi_2 \) is the current formula due to the first reduction by \( \varphi_1 \) over \( \varphi \). Thus, \( \varphi \Rightarrow \varphi_2 \) due to \((x_1 \land \varphi_3) \Rightarrow (\varphi_2) \) and \((x_1 \land \varphi_2 \land x_3) \Rightarrow (\varphi_2 \land \varphi_3) \).

As a result, \( \varphi_2 = \varphi_1 \land \varphi_3 \land (\varphi_2 \land x_3) \land (\varphi_2 \land \varphi_3) \), in which \( \varphi_2 = \varphi_1, \varphi_3 \) denotes the conjuncts, and \( C_1 = \{\varphi_2, \varphi_3\} \) and \( C_2 = \{\varphi_2, \varphi_3\} \) denote the clauses. Note that \( \varphi_2 \neq \{\varphi_1, \varphi_3\} \) and \( \varphi_2 \neq \{\varphi_1, \varphi_3\} \).

Then, \( \varphi_3 \) leads to the next reduction over \( \varphi_2: \varphi_3 = \varphi_2 \land \varphi_3 \land \varphi_3 \land \varphi_3 \).

Therefore, \( \varphi_3 = \varphi_3 \land \varphi_3 \land \varphi_3 \land \varphi_3 \), which denotes the cumulative effects of \( \varphi_1 \) and \( \varphi_3 \).

### 3.2 The Core Algorithms: Scope and Scan

This section specifies Scope and Scan, which incorporate the overall effect \( \hat{\varphi}_s(r_j) \), defined below. Recall that \( \varphi_j \) is removed, if \( r_j \) is necessary for satisfying some formula, i.e., \( r_j \Rightarrow \neg \varphi_j \).

Note that \( \hat{\varphi}_r = (r_{i_k_1} \odot r_{u_k_1} \odot r_{v_k_1}) \land \cdots \land (r_{i_k_n} \odot r_{u_k_n} \odot r_{v_k_n}) \) for Lemma 14 and 15 below.

**Lemma 14.** \( r_j \models \hat{\varphi}_s(r_j) \) such that \( \hat{\varphi}_s(r_j) = r_j \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \), unless \( \not\models \hat{\varphi}_s(r_j) \).

**Proof.** Follows from Definition 5. That is, \( r_j \Rightarrow \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \). Hence, \( r_j \Rightarrow r_j \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \land \varphi_s \).

**Lemma 15.** If \( \not\models r_j \), then \( \models \varphi_s(-r_j) \) holds such that \( \models \varphi_s(-r_j) = (r_i \land r_s) \land \cdots \land (r_i \land r_s) \).

**Proof.** Follows from Definition 6. \( \models \varphi_s(-r_j) = \{\{\}\} \), or \( |C_k| > 1 \) for any \( C_k \) in \( \varphi_s(-r_j) \).

**Lemma 16 (Overall effect of \( r_j \)).** \( r_j \models \hat{\varphi}_s(r_j) \) such that \( \hat{\varphi}_s(r_j) = \hat{\varphi}_s(r_j) \land \hat{\varphi}_s(-r_j) \).

**Proof.** Follows from \( r_j \models r_j \land \neg \varphi_j \), as well as from Lemma 14, and Lemma 15 via \( \varphi_s \).
The algorithm $\text{OvrLef}t(r_j, \phi_s)$ below constructs the overall effect $\tilde{\phi}_s(r_j)$ by means of the local effect $\psi_s(r_j)$ (see Lines 1-6, or L:1-6), as well as of the local effect $\phi_s(\neg r_j)$ (L:7-10).

Algorithm 1 $\text{OvrLef}t(r_j, \phi_s)$ ▶ Construction of the overall effect $\tilde{\phi}_s(r_j)$ due to $r_j$ over $\phi_s$

1: for all $k \in \mathcal{C}^1_s$ over $\phi_s$ do ▶ Construction of the local effect $\psi_s(r_j)$ due to $r_j$ (Lemma 14)
2: for all $r_j \in (\mathcal{C}_k - \{r_j\})$ do ▶ $\psi_s(r_j)$ gets $r_j$ via $\mathcal{R}$ (see Scope L:4), or via $\mathcal{F}$ (Remove L:2)
3: $c_k \leftarrow c_k \cup \{r_j\};$ ▶ $(r_j \lor r_{i1k} \lor r_{i2k}) \land \mathcal{R}_{i1k} \land \mathcal{R}_{i2k}$.
4: end for
5: $\tilde{\psi}_s(r_j) \leftarrow \tilde{\psi}_s(r_j) \cup c_k$; ▶ $c_k$ consists in $\psi_s(r_j)$ (see Scope L:4), or in $\psi_s$ (see Remove L:2)
6: end for
7: for all $k \in \mathcal{C}^2_s$ over $\phi_s$ do ▶ Construction of the local effect $\phi_s(\neg r_j)$ due to $\mathcal{F}$ (Lemma 15)
8: $C_k \leftarrow C_k - \{r_j\};$ ▶ $(r_{j1k} \lor r_{i2k} \lor r_{i2k}) \lor (r_{j1k} \lor r_{i2k}),$ or $(r_{j1k} \lor r_{i2k}) \lor (r_{j1k})$ (Definition 6)
9: if $|C_k| = 1$ then $\tilde{\phi}_s(r_j) \leftarrow \tilde{\phi}_s(r_j) \cup C_k$; $C_k \leftarrow \emptyset;$ $\tilde{\phi}_s(\neg r_j)$ contains no singleton, $C_k \rightarrow c_k$
10: end for
11: return $\tilde{\psi}_s(r_j)$ & $\phi_s(\neg r_j)$ − $\phi_s^{*}$; ▶ $\phi_s(\neg r_j) = \bigcup C_k$ such that $|C_k| > 1,$ or $\phi_s(\neg r_j) = \{\}$

Definition 17. $\not\exists \phi_s(r_j)$ iff $r_j$ is incompatible, that is, the assumption $r_j = T$ cannot hold.

Note. If $\not\exists \phi_s(r_j)$, $r_j$ is incompatible, it is removed from $\phi_s$, that is, $\mathcal{F}$ holds over $\phi_s$.

Note 18. $\phi_s(r_j) = \phi_s \wedge r_j \wedge \phi_s$ by Definition 3/7, hence $\not\exists \phi_s(r_j)$ if $\not\exists (\phi_s \wedge r_j)$ or $\not\exists \phi_s(r_j)$.

Note 19 (Assumption). $\not\exists \phi_s(r_j)$ is verified through solely $\psi_s(r_j)$, called the scope of $r_j$.

Lemma 20 (Scope construction). $r_j \vdash \psi_s(r_j)$ such that $\psi_s(r_j) = \bigland C_k.$ unless $\not\exists \psi_s(r_j)$.

Proof. $\phi_s(r_j) = r_j \wedge \phi_s$ by Definition 7, as $r_j = T$. Then, a deterministic chain of reductions is initiated (Note 8). That is, $r_j \Rightarrow r_j \wedge x_i \wedge \mathcal{F}_a$ due to any clause $(r_j \lor x_i \lor \mathcal{F}_a)$ containing $r_j$, as well as $\neg \mathcal{F}_a \Rightarrow \mathcal{F}_a \wedge x_i$ due to any clause $(\mathcal{F}_a \lor x_i \lor \mathcal{F}_a)$ containing $r_j$. These reductions proceed, as long as new conjuncts $r_e$ emerge in $\phi_s(r_j)$ (see Scope L:2-4). If the reductions are interrupted, then $r_j$ is incompatible (L:5). If they terminate, then the scope $\psi_s(r_j)$ and beyond the scope $\phi_s(r_j)$ are constructed (L:9), where $\psi_s(r_j) = \bigland C_k$ and $\phi_s(r_j) = \bigland C_k$.

Algorithm 2 $\text{Scope}(r_j, \phi_s)$ ▶ Construction of $\psi_s(r_j)$ and $\phi_s(r_j)$ due to $r_j$ over $\phi_s$; $\varphi_s = \psi_s \wedge \phi_s$

1: $\psi_s(r_j) \leftarrow \{r_j\};$ $\phi_s \leftarrow \phi_s;$$\varphi_s = \psi_s \wedge \phi_s; \psi_s$ and $\phi_s$ are disjoint due to $\mathcal{F}$ L:3
2: for all $r_e \in (\psi_s(r_j) - \mathcal{R})$ do ▶ $\text{Scan L:1-3}$
3: $\text{OvrLef}t(r_e, \phi_s);$ ▶ Reductions of $C_k$ initiated by $r_j$ over $\phi_s$ start off
4: $\psi_s(r_e) \leftarrow \psi_s(r_e) \cup \{r_e\} \cup \tilde{\psi}_s(r_e);$ ▶ $\text{OvrLef}t L:5,9$ consists in the scope $\psi_s(r_j)$
5: if $\psi_s(r_{j1k}) \supseteq \{x_i, \mathcal{F}_a\}$ then return NULL; ▶ $r_j \Rightarrow x_i \lor \mathcal{F}_a, i \in L^e.$ $\psi_s(r_j)$, thus $\not\exists \phi_s(r_j)$
6: $\phi_s(\neg r_j) \leftarrow \phi_s(\neg r_j) \wedge \hat{\phi}_s(\neg r_j);$ ▶ $\hat{\phi}_s(\neg r_j) = \{\}$ or $\hat{\phi}_s(\neg r_j) = \bigcup C_k, |C_k| > 1$ (OvrLef L:8-11)
7: $\phi_s \leftarrow \phi_s(\neg r_j) \wedge \phi'_s; R \leftarrow R \cup \{r_e\};$ ▶ $\phi'_s(\neg r_j)$ and $\phi'_s$ consist in beyond the scope $\phi'_s(r_j)$
8: end for
9: return $\psi_s(r_j)$ & $\phi'_s(r_j)$ − $\phi_s$; ▶ $\phi'_s(r_j) = \psi_s(r_j) \lor \phi'_s(r_j); \phi'_s(r_j) \lor \psi_s(r_j) = \bigland C_k,$ $\hat{\phi}_s(r_j) = \bigland C_k$

Note 21. $L_s(r_j)$ being an index set of $\psi_s(r_j)$, $L_s(r_j) \cap L_s'(r_j) = \emptyset$ and $L_s(r_j) \cup L_s'(r_j) = L^e$, if Scope $(r_j, \phi_s)$ terminates. As a result, $\psi_s(r_j)$ and $\phi'_s(r_j)$ are disjoint, and compose $\phi_s(r_j)$.

Note 22. If Scan $(\varphi_s)$ terminates, then $\psi_s$ and $\phi_s$ are disjoint, and compose $\varphi_s$ such that $\varphi_s = \bigland C_k$ (see Definition 2), and that $\phi_s = \bigland C_k$, in which $|C_k| > 1$, because each $C_k = \{r_j\}$ in $\phi_s$ for any $s$ transforms into $r_j$ in $\psi_s$. That is, $C_k = (r_j \lor r_j)$ or $C_k = (r_j \lor r_j \lor r_a)$ in $\phi_s$. 

CVIT 2016
Consider $\psi(x_1)$, Scope $(x_1, \phi)$, for $\phi = (x_1 \odot \mathcal{F}_3) \land (x_1 \odot \mathcal{T}_2 \odot x_3) \land (x_2 \odot \mathcal{F}_3)$. Let $\psi(x_1) \subseteq \{x_1\}$ and $\phi_s \leftarrow \phi$ (L:1). Then, $\phi_s^o$ is empty, and $\phi_s^o = (x_1 \odot \mathcal{F}_3) \land (x_1 \odot \mathcal{T}_2 \odot x_3)$ due to Ovr1Eft $(x_1, \phi_s)$. Also, $\mathcal{C}_s^t = \{1, 2\}$, thus $c_1 \leftarrow \{x_3\}$ and $\psi(x_1) \leftarrow \psi(x_1) \cup c_1$, as well as $c_2 \leftarrow \{x_3, \mathcal{F}_3\}$ and $\psi(x_1) \leftarrow \psi(x_1) \cup c_2$ (see Ovr1Eft L:1-6). Then, $\psi(x_1) = \{x_3, x_2, \mathcal{T}_3\}$. Also, $\psi(x_1) = \{x_2, \mathcal{T}_3\}$ due to Ovr1Eft L:11. As a result, $\psi(x_1) \subseteq \psi(x_1) \cup \{x_1\} \cup \psi(x_1)$ (Scope L:4), and $\psi(x_1) \supseteq \{x_3, \mathcal{T}_3\}$. That is, $x_1 \Rightarrow x_3 \land \mathcal{F}_3$, hence $x_1$ is incompatible in the first scan.

**Definition 24.** $\mathcal{L}^+ = \{i \in \mathcal{L} | r_i \in \psi_s\}$ and $\mathcal{L}^- = \{i \in \mathcal{L} | r_i \in \mathcal{C}_k \in \phi_s\}$ due to $\varphi_s = \psi_s \land \phi_s$.

Figure 4 illustrates $\text{Scan}(\varphi_s)$. It decomposes $\varphi_s = \bigwedge \mathcal{C}_k$ into $\psi_s(x_1), \psi_s(\mathcal{T}_1), \ldots, \psi_s(x_n)$, $\psi_s(\mathcal{T}_n)$, such checks if $\varphi_s(x_1)$ and $\varphi_s(\mathcal{T}_i)$, where $\psi_s(.) = \bigwedge \mathcal{C}_k$ is true by Definition 2. $\psi_s$ transforms into $\hat{\phi} \equiv \bigwedge (\psi(x_1) \oplus \psi(\mathcal{T}_i))$, if $\text{Scan}(\varphi_s)$ terminates.

Figure 4: $\text{Scan}$ decomposes $\phi_s$ into $\psi_s(x_1), \psi_s(\mathcal{T}_i), \ldots, \psi_s(\mathcal{T}_n)$, and transforms $\psi \land \phi$ into $\hat{\phi}$.

**Algorithm 3 $\text{Scan}(\varphi_s)$**

1. for all $i \in \mathcal{L}^+$ and $\mathcal{T}_i \subseteq \psi_s$ do \>$\varphi_s(r_i) = \psi_s \land r_i \land \phi_s$, thus $\varphi_s(r_i)$, that is, $r_i \Rightarrow x_1 \land \mathcal{T}_i$.
2. Remove $(r_i, \phi_s)$; \>$\mathcal{T}_i$ is necessary, thus $r_i$ is incompatible trivially, hence $\Gamma \Rightarrow \neg r_i$.
3. for all $i \in \mathcal{L}^-$ do \>$\mathcal{L}^+ \land \mathcal{L}^- = \emptyset$ due to L:1-3. Hence, if $r_i = x_i$ is fixed or $r_i = \mathcal{T}_i$ is fixed.
4. for all $r_i \in \{x_i, \mathcal{T}_i\}$ do \>$\bigwedge r_i \subseteq \{x_i, \mathcal{T}_i\}$ assumed to be true is to be verified.
5. if Scope $(r_i, \phi_s)$ is NULL then Remove $(r_i, \phi_s)$; \>$\text{Incompatible} \text{nontrivially if } \varphi_s(r_i)$.
6. end for; \>$\text{If } r_i \Rightarrow x_i \land x_i$, hence $\neg x_i \lor \neg \mathcal{T}_i \Rightarrow \neg r_i$, then $\neg r_i \Rightarrow \mathcal{T}_i$, where $i \neq j$ due to L:1-3.
7. end for; \>$\neg r_i$ iff $\mathcal{T}_i$, since $\neg r_i \Rightarrow \mathcal{T}_i$ due to nontrivial, and $\neg r_i \Rightarrow \mathcal{T}_i$ due to trivial incompatibility.
8. return $\hat{\phi} = \psi \land \hat{\phi}$, and $\psi(r_i)$, $\phi_s(r_i)$ for all $i \in \mathcal{L}^+$; \>$\hat{\phi} \leftarrow \psi_s$ and $\phi_s \leftarrow \phi_s$. See also Note 22.

**Note 25.** $\mathcal{L}^+$ and $\mathcal{L}^-$ form a partition of $\mathcal{L}$ due to Definition 24 and Scan L:1-3.

**Algorithm 4 $\text{Remove}(r_j, \phi_s)$**

1. Ovr1Eft $(\mathcal{T}_j, \phi_s)$; \>$\text{Ovr1Eft is defined over } \phi_s = \bigwedge \mathcal{C}_k$, $|\mathcal{C}_k| > 1$, and returns $\hat{\psi}_s(\mathcal{T}_j)$ & $\hat{\phi}_s(-\mathcal{T}_j)$.
2. for all $i \in \psi_s + 1$ do \>$\bigwedge s_i = \bigwedge \mathcal{C}_k$ is true by Definition 2, unless $\psi_s + 1$ involves $x_i$ and $\mathcal{T}_j$.
3. if $\psi_s + 1 \supseteq \{x_i, \mathcal{T}_i\}$ for some $i$ then return $\varphi$ is unsatisfiable; \>$\varphi_s = \psi_s \land \phi_s$.
4. $\mathcal{L}^+ \leftarrow \mathcal{L}^+ - \{j\}$; $\mathcal{L}^- \leftarrow \mathcal{L}^- - \{j\}$; \>$\phi_s \subseteq \bigwedge C_k$ for $k \in \mathcal{C}_s$, where $C_i = \mathcal{C} - \{C_i \cup C_i'\}$, and $C_i \cap C_i' = \emptyset$ due to Lemma 10.
5. $\phi_s + 1 \leftarrow \hat{\phi}_s(-\mathcal{T}_j) \land \phi_s'$; Update $\mathcal{C}_s$ over $\phi_s + 1$; \>$\phi_s'$ denotes clauses beyond the entire $\psi_s$ effect.
6. $\text{Scan}(\varphi_s + 1); \text{if } r_j$ verified compatible for $\hat{s} \leq s$ can be incompatible for $\hat{s} > s$ due to $\neg r_j$, in $\phi_s$.
3.3 Unsatisfiability of $\phi(r_j)$ vs Unsatisfiability of $\psi_s(r_j)$ for some $s$

This section tackles satisfiability of $\varphi$ through unsatisfiability of a truth assignment $\phi(r_j)$.

**Proposition 26** (Nontrivial incompatibility). $\not\models \phi(r_j)$ iff $\not\models \psi_s(r_j)$ or $\not\models \phi'(r_j)$ for some $s$.

**Proof.** Proof is obvious due to $\phi_s(r_j) = \psi_s(r_j) \land \phi'_s(r_j)$, transformed from $\phi_s(r_j) := r_j \land \phi_s$ through $\text{Scope}(r_j, \phi_s)$. Moreover, $\not\models \phi(r_j)$ iff $\not\models \phi_s(r_j)$ for some $s$ due to Theorem 36. ◀

**Remark.** It is trivial to verify $\not\models \psi_s(r_j)$ (see Scope L.5). It is redundant to check if $\not\models \phi'_s(r_j)$, since $\not\models \phi_s(r_j)$ is verified solely via $\not\models \psi_s(r_j)$ by assumption (Note 19). Thus, it is easy to verify $\not\models \phi_s(r_j)$ for Scan L.6. The following introduces the tools to justify this assumption.

$\Sigma_s(r_i) = \Sigma(\psi_s(r_i))$ denotes the index set of the scope $\psi_s(r_i)$. Likewise, $\Sigma'_s(r_i) = \Sigma(\phi'_s(r_i))$.

Also, we define the conditional scope $\psi_s(r_j|r_j)$ and beyond the scope $\phi'_s(r_j|r_j)$ over $\phi'_s(r_j)$ for any $j \neq i$, which are constructed by $\text{Scope}(r_i, \phi'_s(r_j))$. Thus, $\Sigma_s(r_j|r_j) = \Sigma(\psi_s(r_j|r_j))$.

**Lemma 27** (No conjunct exists in beyond the scope). $\Sigma_s(r_j) \cap \Sigma'_s(r_j) = \emptyset$ for any $j \in \mathcal{L}$.

**Proof.** $\phi'_s(r_j) = \mathcal{L}$ by $\text{Scope}(r_j, \phi_s)$. Let $r_i$ the conjunct be in $C_k$, $i \in (\Sigma_s(r_j) \cap \Sigma'_s(r_j))$. Then, for any $C_k \ni r_i \cap x_j \cap \pi_k \cap (r_j \land \pi_j \land x_k)$, thus $r_j \not\in C_k$. Moreover, for any $C_k \ni r_i$, $(r_i \cap x_j \cap \pi_k) \not\ni (r_i \cap x_j \cap \pi_k)$, thus $r_i \not\in C_k$. See Definition 5/6. Hence, $i \not\in (\Sigma_s(r_j) \cap \Sigma'_s(r_j))$. ▶

**Note.** No conjunct exists in any clause $C_k$ due to Note 25, which states $\mathcal{L} \cap \mathcal{L} = \emptyset$.

**Lemma 28.** $\mathcal{L}$ is partitioned into $\Sigma_s(r_j), \Sigma_s(r_j_1|r_j), \ldots, \Sigma_s(r_j_m|r_j_m)$ by means of $\text{Scope}$.

**Lemma 29.** $\phi_s(r_j)$ is decomposed into disjoint $\psi_s(r_j), \psi_s(r_j_1), \ldots, \psi_s(r_j_m)$.

**Proof.** $\text{Scope}(r_j, \phi_s)$ partitions $\mathcal{L}$ into $\Sigma_s(r_j) \land \Sigma'_s(r_j)$ for any $j \in \mathcal{L}$ (see Lemma 27). Thus, $\phi_s(r_j)$ is decomposed into disjoint $\psi_s(r_j)$ and $\phi'_s(r_j)$. $\text{Scope}(r_j, \phi'_s(r_j))$ partitions $\Sigma'_s(r_j)$ into $\Sigma_s(r_j_1|r_j) \land \Sigma'_s(r_j_1|r_j)$ for any $j_1 \in \mathcal{L}'(r_j)$. Thus, $\phi'_s(r_j)$ is decomposed into disjoint $\psi_s(r_j_1)$ and $\phi'_s(r_j_1)$. Finally, $\phi'_s(r_j_m|r_j_m)$ is decomposed into disjoint $\psi_s(r_j_m)$ and $\phi'_s(r_j_m)$ for any $j_m \in \mathcal{L}'(r_j_m|r_j_m)$ such that $\mathcal{L}'(r_j_m|r_j_m) = \emptyset$ (see also Note 21). ◀

Let the scan terminate (see Scan L.9), thus $\psi \land \phi$ transforms into $\psi \land \hat{\phi}$. Let $\phi \leftarrow \hat{\phi}$, thus $\mathcal{L} \leftarrow \mathcal{L}$. Also, $\psi(r_i) = T$ for every $i \in \mathcal{L}$ and $r_i \ni \{x_i, \pi_i\}$. Then, Lemma 29 leads to the fact (Theorem 34) that it is redundant to check if $\not\models \phi'_s(r_j)$ to verify $\not\models \phi_s(r_j)$ for any $s$.

**Lemma 30.** $\phi'(r_j)$ is decomposed into disjoint $\psi(r_j_1), \psi(r_j_2), \ldots, \psi(r_j_m)$.

**Proof.** Follows from Lemma 29, and from $\phi(r_j) = \psi(r_j) \land \phi'(r_j)$ due to $\text{Scope}(r_j, \phi)$. ◀

**Lemma 31.** $\phi \equiv \phi'(r_j) \equiv \phi'(r_j) \equiv \phi'(r_j) \equiv \phi'(r_j)$, since it terminates.

**Proof.** Some $C_k$ in $\phi$ collapse to some $C_k$ in $\psi(r_j)$ due to $\text{Scope}(r_j, \phi)$ (see Lemma 20). As a result, the number of $C_k$ in $\phi$ is greater than or equal to that of $C_k$ in $\phi'(r_j)$, thus $|\mathcal{L}| \geq |\mathcal{L}'|$, where $\mathcal{L}$ denotes an index set of $C_k$ in $\phi$. Also, some $C_k$ in $\phi$ shrink to some $C_k \in \phi'(r_j)$, thus $\forall \mathcal{L} \ni \mathcal{L}' \ni \mathcal{L} \ni \mathcal{L}' \ni \mathcal{L}'$. Hence, $\phi \equiv \phi'(r_j)$. Likewise, $\phi'(r_j) \equiv \phi'(r_j)$, since $\phi'(r_j)$ is decomposed into $\psi(r_j_1)$ and $\phi'(r_j_1)$ via $\text{Scope}(r_j_1, \phi'(r_j))$. Therefore, $\phi \equiv \phi'(r_j) \equiv \phi'(r_j) \equiv \phi'(r_j)$, where $\phi'(r_j) = \phi'(r_j_1)$, $\psi(r_j) = \phi'(r_j)$ (see Figure 2). ◀

**Lemma 32** (Any scope entails its conditional scope). $\psi(r_i) \equiv \psi(r_i)$, since it terminates.

**Proof.** $\phi \equiv \phi'(r_j)$ due to Lemma 31. $\text{Scope}(r_i, \phi)$ constructs the scope $\psi(r_i)$ over $\phi$, while $\text{Scope}(r_i, \phi'(r_j))$ constructs the conditional scope $\psi(r_i|r_i)$ over $\phi'(r_j)$, thus $\psi(r_i) \equiv \psi(r_i|r_i)$, where $\psi(r_i) = \bigwedge C_k$ by Definition 2 and Lemma 20. Since $\psi(r_i) \equiv \psi(r_i|r_i)$ and $\psi(r_i)$ is true for all $r_i$ in $\phi$, $\psi(r_i|r_i)$ is true for all $r_i$ in $\phi'(r_j)$. Hence, $\psi(r_i) \equiv \psi(r_i|r_i)$ (see Figure 2). ◀
Lemma 33. \( \psi(r_j | r_j) \), \( \psi(r_j | r_j, r_j) \), \( \ldots \), \( \psi(r_j | r_j, r_j, \ldots, r_j) \) is true for every \( j \in \Sigma \), and for every \( i \in \mathcal{L}(r_j) \), \( i \in \mathcal{L}(r_j | r_j) \), \( \ldots \), \( i \in \mathcal{L}(r_j | r_j, r_j, \ldots, r_j) \), because the scan terminates.

Proof. Recall that the scan terminates. Thus, \( \phi = \hat{\phi} \land \dot{\phi} \) and \( \phi := \hat{\phi} \land \dot{\phi} \subseteq \mathcal{L} \) (see also Note 22). Hence, a truth assignment \( \psi(r_j) \) holds for every \( i \in \mathcal{L} \) and \( r_j \in \{ x_1, \mathcal{T}_j \} \). Moreover, \( \phi \supseteq \phi(r_j) \supseteq \phi(r_j, r_j) \supseteq \phi(r_j, r_j, r_j) \supseteq \ldots \supseteq \phi(r_j, r_j, \ldots, r_j) \) due to Lemma 31 for any \( j \in \Sigma \), and \( j_1 \in \mathcal{L}(r_j) \), \( j_n \in \mathcal{L}(r_j | r_j) \). Then, \( \psi(r_j) \supseteq \psi(r_j, r_j) \supseteq \psi(r_j, r_j, r_j) \supseteq \psi(r_j, r_j, \ldots, r_j) \), in which \( \psi(r_j) \supseteq \psi(r_j | r_j) \) via Scope \( (r_j, \phi(r_j, r_j)) \), thus \( \psi(r_j) \models \psi(r_j, r_j) \). Therefore, any \( \psi(r_j, r_j, \ldots, r_j) \) is true, which generalizes Lemma 32.

Theorem 34 (Unsatisfiability). \( \not\models \phi(r_j) \), \( r_j \) is incompatible, \( \not\models \psi_x(r_j) \) for some \( s \).

Corollary 35 (Satisfiability). \( \models \phi(r_j) \) if \( \not\models \phi_x(r_j) \) (Satisfaction). \( \not\models \phi_x(r_j) \) is true for all \( r_j \) by \( \not\models \phi_x(r_j) \), even if \( \neg r_j \).

Proof. \( \not\models \phi_x(r_j) \), if \( \not\models \psi_x(r_j) \) or \( \not\models \phi_x(r_j) \) (Scan L.16). \( \psi_x \supseteq \psi_x \) for all \( s \) by L.2, thus \( \not\models \psi_x(r_j) \) for all \( s \) by L.2. \( \not\models \psi_x(r_j) \) due to \( x_1 \land x_2 \not\models \phi_x \), hence \( x_1 \lor x_2 \not\models \phi_x \).

Proposition 37. The time complexity of Scan is \( O(mn^3) \).

Proof. 0vRLef, and Remove, takes \( 4m \) steps by \( \langle \mathcal{L} \times |C_k| \rangle + |C_k| = 3m + m \). \( \mathcal{L} \) takes \( 4m \) steps by \( \psi(r_j) \times 4m \). Then, \( \mathcal{L} \) takes \( n^24m \) steps due to L.1-3 by \( \mathcal{L} \times |\psi| \times 4m \), as well as \( 8n^2m + 8nm \) steps due to L.4-8 by \( 2|\mathcal{L}| \times (4m + 4m) \). Also, the number of the scan is \( s \leq |\mathcal{L}| \) due to Remove L.6. Therefore, the time complexity of Scan is \( O(nm^3) \).

Example 38. Let \( \varphi = \{ \{ x_1, x_4, \mathcal{T}_j \}, \{ x_3, x_6, \mathcal{T}_j \}, \{ x_4, x_6, \mathcal{T}_j \} \} \). Let \( \mathcal{S} = \mathcal{T}_{3, \phi} \) execute first in the first scan, which leads to the reductions below over \( \varphi \) due to \( x_3 \).

\[ \begin{align*}
\phi(x_1) &= (x_3 \land x_4 \land \mathcal{T}_4) \land (x_3 \land x_4 \land \mathcal{T}_7) \land x_3 \\
n(x_3) &= (x_3 \land \mathcal{T}_4 \land x_3) \land (x_3 \land \mathcal{T}_6 \land x_7) \land (x_4 \land x_6 \land x_7) \land x_3 \\
\mathcal{T}_4 &= (x_3 \land \mathcal{T}_4 \land x_3) \land (x_3 \land \mathcal{T}_6 \land x_7) \land (x_4 \land x_6 \land x_7) \land x_3 \\
\mathcal{T}_6 &= (x_3 \land \mathcal{T}_4 \land x_3) \land (x_3 \land \mathcal{T}_6 \land x_7) \land (x_4 \land x_6 \land x_7) \land x_3
\end{align*} \]

Because \( \not\models (\varphi(x_1) = x_3 \land x_4 \land x_5 \land x_6 \land x_7 \land \mathcal{T}_7) \), \( x_3 \) is incompatible, hence \( x_3 \) is necessary, i.e., \( \neg x_3 \models \mathcal{T}_3 \). Thus, \( \varphi \models \varphi_2 \) by \( (x_3 \land x_4 \land x_5) \models (x_4 \land x_6 \land x_7) \) and \( (x_3 \land x_4 \land x_7) \models (x_4 \land x_6 \land x_7) \).

As a result, \( \varphi_2 = (x_3 \land \mathcal{T}_4) \land (x_4 \land \mathcal{T}_7) \land (x_3 \land x_6 \land \mathcal{T}_7) \land x_3 \). Let \( \mathcal{S} = \mathcal{T}_{3, \phi} \) execute next. \( \phi_2(x_2) \) is given by

\[ \begin{align*}
\phi_2(x_2) &= (x_4 \land \mathcal{T}_4 \land x_5) \land (x_4 \land \mathcal{T}_7 \land x_5) \\
x_5 &= \langle x_4 \rangle \land (x_6 \land \mathcal{T}_7 \land (x_4 \land x_6 \land \mathcal{T}_7) \land x_5) \\
x_4 &= \langle x_4 \rangle \land (x_6 \land \mathcal{T}_7 \land (x_4 \land \mathcal{T}_6 \land x_7) \land x_5) \\
\mathcal{T}_6 &= \langle x_4 \rangle \land (x_6 \land \mathcal{T}_7 \land (x_4 \land \mathcal{T}_6 \land x_7) \land x_5)
\end{align*} \]

Because \( \not\models (\psi(x_5) = x_4 \land \mathcal{T}_7 \land \mathcal{T}_6 \land x_7 \land \mathcal{F}_3 \land x_3) \), \( x_5 \) is removed from \( \phi_2 \), i.e., \( \neg x_5 \models \mathcal{T}_3 \).

Thus, \( \varphi_2 \models \varphi_3 \) by \( (x_3 \land \mathcal{T}_4) \land (x_4 \land \mathcal{T}_7) \), where \( \varphi_3 = (x_4 \land \mathcal{T}_4) \land (x_4 \land x_6 \land \mathcal{T}_7) \land \mathcal{T}_3 \) and \( \mathcal{T}_3 \) leads to the next reduction by \( (x_4 \land x_6 \land \mathcal{T}_7) \models (x_4 \land \mathcal{T}_7) \). Then, \( \mathcal{S}(x_4) \) terminates, and \( \varphi_4 = \mathcal{T}_4 \land \mathcal{T}_6 \land \mathcal{T}_7 \land \mathcal{F}_3 \), that is, \( \phi = \hat{\phi} \land \phi_\phi \), and \( \hat{\phi} = \{ x_3, x_4, x_5 \} \) and \( \hat{\phi} = \{ x_6, \mathcal{T}_7 \} \).
In Example 38, if Scope \( x_5, \phi \) executes first, then \( \psi(x_5) = x_5 \) becomes the scope, and
\[
\phi'(x_5) = (x_3 \circ x_4) \land (x_1 \circ x_6 \circ \tau_7) \land (x_4 \circ x_6 \circ \tau_7)
\]
becomes beyond the scope of \( x_5 \) over \( \phi \).
Then, \( x_5 \) is compatible (in \( \phi \)) due to Theorem 34, since \( \psi(x_5) \) is true, while it is incompatible
due to Proposition 26, since \( \not\models \phi'(x_5) \) holds. On the other hand, the fact that \( \not\models \phi'(x_5) \) holds
is verified indirectly. That is, incompatibility of \( x_5 \) is checked by means of \( \psi_i(x_5) \) for some \( i \).
Then, \( x_5 \) becomes incompatible (in \( \phi_2 \)), because \( \not\models \psi_2(x_5) \) holds, after \( \varphi \rightarrow \varphi_2 \) by removing
\( x_3 \) from \( \phi \) due to \( \not\models \psi(x_3) \). As a result, \( \not\models \phi'(x_5) \) holds due to \( \neg x_3 \). Thus, there exists no
\( r_j \) such that \( \not\models \phi'(r_j) \), when the scan terminates, because \( \psi(r_i) \) is true for all \( r_i \) in \( \phi \), hence
\( \psi(r_i|r_j) \) is true for all \( r_i \) in \( \phi'(r_j) \), after each \( r_j \) is removed if \( \not\models \psi_s(r_j) \) (see also Figures 1-4).

### 3.4 Construction of a satisfiable assignment by composing minterms

\[ \hat{\psi} = \hat{\psi} \land \hat{\phi} \], when Scan \( \varphi_2 \) terminates. Let \( \psi := \hat{\psi} \land \hat{\phi} \), i.e., \( \mathcal{L} := \mathcal{L}^\psi \). Then, \( \models_a \phi \) holds
by Corollary 35, where \( \alpha \) is a satisfiable assignment, and constructed by Algorithm 5 through
any \( \{i_0, i_1, i_2, \ldots, i_m, i_n\} \) over \( \mathcal{L} \) such that \( \alpha = \{\psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \psi(r_{i_2}|r_{i_1}), \ldots, \psi(r_{i_m}|r_{i_{m-1}})\} \).
Thus, \( \varphi \) is decomposed into disjoint minterms \( \psi, \psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \psi(r_{i_2}|r_{i_1}), \ldots, \psi(r_{i_m}|r_{i_{m-1}}) \)
(see Note 25, and Lemmas 28-29). Note that \( \psi \) is fixed in each satisfiable assignment for \( \varphi \).
Recall that Scope \( (r_i, \phi) \) constructs the scope \( \psi(r_i) \) and beyond the scope \( \phi'(r_i) \) to determine
any assignment \( \alpha \), unless \( \varphi \) itself collapses to a unique assignment, i.e., unless \( \alpha = \hat{\psi} \). See
also Appendix A to determine \( \alpha \) without constructing \( \psi(r_i|\ldots) \) and \( \phi'(r_i|\ldots) \) by Scope \( (r_i, \phi'(\ldots)) \).

---

**Algorithm 5**  
Construction of a satisfiable assignment \( \alpha \) over \( \mathcal{L} := \mathcal{L}^\psi \) and \( \phi := \hat{\phi} \)

**Pick** \( j \in \mathcal{L} \);  
\( \Rightarrow \) The scope \( \psi(r_i) \) and beyond the scope \( \phi'(r_i) \) for all \( i \in \mathcal{L} \) are available initially  
\( \alpha := \psi(r_j); \mathcal{L} := \mathcal{L} - \{r_j\}; \phi := \phi'(r_j); \)

**repeat**  
\( i \in \mathcal{L}; \text{ Scope}(r_i, \phi); \)  
\( \Rightarrow \) It constructs \( \psi(r_i|r_j) \) and \( \phi'(r_i|r_j) \) with respect to \( \phi'(r_j) \)  
\( \alpha := \alpha \cup \psi(r_i); \Rightarrow \psi(r_i) := \psi(r_i|r_j), \text{ because } \psi(r_i) \text{ is unconditional with respect to } \phi \text{ updated} \)  
\( \mathcal{L} := \mathcal{L} - \{r_i\}; \Rightarrow \mathcal{L} := \mathcal{L} - \{r_i\} \) due to the partition \( \{L, L', L'' \} \) over \( \mathcal{L} \)  
\( \phi := \phi'(r_i); \Rightarrow \phi'(r_i) := \phi'(r_i|r_j), \text{ because } \phi'(r_i) \text{ is unconditional with respect to } \phi \text{ updated} \)

until \( \mathcal{L} = \emptyset \)

**return** \( \alpha; \)

---

**Definition 39.** Let \( \langle r_{i_1,1}, r_{i_2,1}, r_{i_3,1} \rangle, \langle r_{i_1,2}, r_{i_2,2}, r_{i_3,2} \rangle, \ldots, \langle r_{i_m,1}, r_{i_m,2}, r_{i_m,3} \rangle \) be in ascending order with respect to the index set \( \mathcal{L} \). If \( i_3 < i_1 \) for any \( \langle r_{i_1,k}, r_{i_2,k}, r_{i_3,k} \rangle \) and any
\( \langle r_{i_3,k+1}, r_{i_2,k+1}, r_{i_1,k+1} \rangle \), then \( \phi^1 \cup \phi^2 = \phi \) and \( \phi^1 \cap \phi^2 = \emptyset \) such that \( C_k \subseteq \phi^1 \) and \( C_{k+1} \subseteq \phi^2 \).

**Note.** \( \phi \) and \( \phi \) form a partition of \( \phi \), hence their satisfiability check can be independent.

**Example 40.** Let \( \phi = (x_1 \circ \tau_2 \circ x_6) \land (x_3 \circ x_4 \circ \tau_5) \land (x_4 \circ x_6 \circ \tau_7), \)
\( \phi = (x_1 \circ x_5 \circ \tau_6) \), and \( \phi = (x_{11} \circ \tau_{12} \circ x_{13}) \) to form \( \varphi = \phi \cup \phi \cup \phi \) (see Definition 39).
Then, Scan \( \varphi_2 \) returns \( \varphi \) is satisfiable. Therefore, \( \hat{\phi} = \hat{\psi} \land \hat{\phi} \), where \( \psi := \hat{\psi} = \tau_3 \land \tau_4 \land \tau_5 \)
and \( \phi := \hat{\phi} = (x_1 \circ \tau_2 \circ x_6) \land (x_5 \circ \tau_7) \land \phi \) (see Example 38). Then, \( \alpha \) is constructed by
compressing \( \psi(\ldots) \) based on \( \phi'(\ldots) \), below, where \( \mathcal{L} = \{3, 4, 5\} \) and \( \mathcal{L} := \mathcal{L}^\psi = \{1, 2, \ldots, 13\} - \mathcal{L}^\psi. \)
As a result, while $\psi$ becomes necessary (a conjunct). Then, $\alpha \leftarrow \psi(x_6)$ and $\psi(x_{11}|x_6, x_8) = \psi(x_{11})$, since $\phi \wedge \phi_1 \wedge \phi_2$ are disjoint (see Definition 39). Consequently, Algorithm 5 constructs $\alpha = \{\psi(x_6), \psi(x_8|x_6), \psi(x_{11}|x_6, x_8)\}$. Note that $\psi$ is decomposed into $\psi(x_6), \psi(x_8|x_6)$, and $\psi(x_{11}|x_6, x_8)$, which are disjoint (see also Note 22 and Lemma 29).

This section illustrates Scan ($\varphi$). Let $\varphi = \phi = (x_1 \circ T_3) \wedge (x_1 \circ T_2 \circ x_3) \wedge (x_2 \circ T_3)$, which is adapted from Esparza [1], and denotes a general formula by Definition 9. Note that $C_1 = \{x_1, \bar{T}_3\}, C_2 = \{x_1, \bar{T}_2, x_3\}$, and $C_3 = \{x_2, \bar{T}_3\}$. Hence, $\varphi = \{1, 2, 3\}$, and $\varphi = \varphi^\circ = \{1, 2, 3\}$.

Scan ($\varphi$): There exists no conjunct in the (initial formula) $\varphi$. That is, $\psi$ is empty (L.1).

Recall that $\varphi := \varphi_1$, and that $r_i \in \{x_1, \bar{T}_1\}$. Recall also that nontrivial incompatibility of $r_i$ is checked (L.4-8) via $\text{Scope}(\alpha_2, \phi)$. Moreover, the order of incompatibility check is arbitrary (incompatibility is monotonic) by Theorem 36. Let $\text{Scope}(x_1, \phi)$ execute due to Scan L.6.

$\text{Scope}(x_1, \phi)$: Since $\psi(x_1) \supseteq \{x_2, \bar{T}_3\}$, $x_1$ is incompatible nontrivially (see Example 23).

Thus, $\bar{T}_1$ becomes necessary (a conjunct). Then, Remove ($x_1, \phi$) executes due to Scan L.6.

Remove ($x_1, \phi$): $C^T_1 = \emptyset$ by $\text{OvrLefT}$ L.1. $C^T_1 = \{1, 2\}$, thus $\phi^T_1 = (x_1 \circ T_3) \wedge (x_1 \circ T_2 \circ x_3)$ by $\text{OvrLefT}$ L.7. As a result, $\psi(\bar{T}_1) = \{\bar{T}_1\}$ and $\phi(\bar{x}_1) = \emptyset$, the effects of $\varphi_1$, and $\bar{T}_1 \neq 0$. Then, $\psi(\bar{T}_1) \cup \psi(\bar{T}_1)$ (Remove L.2), and $\varphi^\circ \leftarrow \varphi^\circ - \{1\}$ and $\varphi^\circ \leftarrow \varphi^\circ - \{1\}$ (L.4). Also, $\phi_2 \leftarrow \phi_2(-x_1) \wedge \phi_2$, where $\phi_2(-x_1) = (T_2 \circ x_3)$ and $\phi_2 = (x_1 \circ T_2 \circ x_3)$. (L.5). As a result, $\psi(\bar{T}_2) \wedge \bar{T}_3$, and $\phi_2 = (T_2 \circ x_3) \wedge (x_2 \circ \bar{T}_3)$. Note that $C_1 = \{T_2, x_3\}$ and $C_2 = \{T_2, x_3\}$. Consequently, $\varphi_2 = \varphi_2 \wedge \varphi_2$, and Scan ($\varphi_2$) executes due to Remove L.6.

Scan ($\varphi_2$): $C_2 = \{1, 2\}$ and $\varphi^\circ = \{2, 3\}$ hold in $\varphi_2$. Then, $\{x_2, \bar{T}_2\} \cap \psi_2 = \emptyset$ for $2 \in \varphi^\circ$, while $\varphi_3 \subseteq \psi_2$ for $3 \in \varphi^\circ$ (L.1). As a result, $\varphi_3$ is necessary for satisfying $\varphi_2$, hence $\varphi_2 \leftarrow \neg x_3$, that is, $\bar{T}_3$ becomes incompatible trivially. Then, Remove ($x_3, \varphi_2$) executes due to Scan L.2.

Remove ($x_3, \varphi_2$): $C^T_3 = \{2\}$, thus $\phi^T_3 = (x_2 \circ T_3)$, and $C^T_3 = \{1\}$, thus $\phi^T_3 = (T_2 \circ x_3)$.

As a result, $\psi(\bar{T}_3) = \{T_3\} \cup \{\bar{T}_3\}$ and $\phi_2(-x_3) = \emptyset$, because $C_1 = \{\bar{T}_3\}$ consists in $\phi_2(\bar{\varphi}_2)$, rather than in $\phi_2(-x_3)$ (see $\text{OvrLefT}$ L.9). Hence, $\psi_3 \leftarrow \psi(\bar{T}_3) \cup \psi_3(\bar{T}_3)$, $\psi^\circ \leftarrow \psi^\circ - \{3\}$, and $\varphi^\circ \leftarrow \varphi^\circ - \{3\}$, i.e., $\varphi^\circ = \emptyset$. Therefore, $\varphi_3 = \emptyset$, and $\varphi_3 = \{T_3 \wedge \bar{T}_3 \wedge \bar{T}_2\}$.

Scan ($\varphi_3$): $\varphi_3$ for $2 \in \varphi^\circ$ over $\varphi_3$. Then, Remove ($x_2, \varphi_3$) executes due to Scan L.2.

Remove ($x_2, \varphi_3$): $\varphi_3(\bar{T}_2) = \emptyset$ and $\phi_2(-x_2) = \emptyset$ due to $\text{OvrLefT}$ (T_2, \phi_2)$, because $C^T_2 = \emptyset$ and $C^T_2 = \emptyset$, where $C_3 = \emptyset$. Hence, $\varphi^\circ \leftarrow \varphi^\circ - \{2\}$ and $\varphi_3 \leftarrow \alpha$. Then, Scan ($\varphi_3$) executes.

Scan ($\varphi_4$) terminates: $\psi = \psi_3 \wedge \bar{T}_3 \wedge \bar{T}_2$ (L.9), and $\varphi$ collapses to a unique assignment.
Let \( \text{Scope} (x_3, \phi) \) execute before \( \text{Scope} (x_1, \phi) \) due to \( \text{Scan} L:6 \) (see Theorem 36).

\( \text{Scope} (x_3, \phi) : \phi (x_3) \leftarrow \{ x_3 \} \) and \( \phi \leftarrow \phi (L:1) \). Then, \( \mathcal{C}^3 = \{2\} \) due to 0vlEft \((x_1, \phi_s)\).

L:1, hence \( \phi^3 = (x_1 \lor x_3 \land x_1) \). As a result, \( c_2 \leftarrow \{ t_1, x_2 \} \) and \( \psi (x_3) \leftarrow \psi (x_3) \cup c_2 \) (L:3.5). Moreover, \( \mathcal{C}^3 = \{1, 3\} \) (L:7), hence \( \phi^3 = (x_1 \lor x_3) \land (x_2 \lor x_3) \). Then, \( C_1 = \{ x_1, x_3 \} \) (L:9).

\( \psi (x_3) = \psi (x_3) \cup C_1, \) and \( C_1 \leftarrow \emptyset \). Likewise, \( C_3 = \{ x_2, x_3 \} \leftarrow \{ x_3 \}, \) \( \psi (x_3) = \psi (x_3) \cup C_3, \) and \( C_3 \leftarrow \emptyset \) (0vlEft L:8.9).

Consequently, \( \psi (x_3) \leftarrow \{ t_1, x_2, x_1 \} \) \& \( \phi (\neg x_3) \leftarrow \phi^3 (L:11) \).

Note that \( \phi^3 = \{\} \) since \( C_3 = C_3 = \emptyset \). Then, \( \psi (x_3) \leftarrow \psi (x_3) \cup \{ x_1 \} \) due to \( \text{Scope} L:4 \), hence \( \psi (x_3) = \{ x_3, x_1, x_2, x_1 \} \). Since \( \psi (x_3) \subseteq \{ t_1, x_1 \} \) (L:5), \( x_3 \) is incompatible nontrivially, i.e., \( x_3 \Rightarrow t_1 \land x_1 \) and \( \neg x_3 \Rightarrow t_3 \). Then, \( \text{Remove} (x_3, \phi) \) executes due to \( \text{Scan} L:6 \).

\( \text{Remove} (x_3, \phi) : \phi^3 = (x_1 \lor x_3) \land (x_2 \lor x_3) \) due to \( \mathcal{C}^3 = \{1, 3\} \), and \( \phi^3 = (x_1 \lor x_3) \land x_3 \) due to \( \mathcal{C}^3 = \{2\} \). Then, \( 0vlEft (t_3, \phi) \) returns \( \psi (t_3) = \{ t_1, t_2 \} \) \& \( \neg \psi (t_3) = \{ x_1, t_2 \} \) (Remove L:1).

\( \psi_2 \leftarrow \psi \cup \{ t_3 \} \cup \neg \psi (t_3) \) (L:2), and \( \mathcal{L}^e \leftarrow \mathcal{L}^e - \{3\} \) and \( \mathcal{L}^e \leftarrow \mathcal{L}^e \cup \{3\} \) (L:4). As a result, \( \psi_2 = \mathcal{F}_2 \land \mathcal{F}_2 \land t_2 \). Moreover, \( \phi_2 \leftarrow \phi (\neg x_3) \land \phi' (L:5) \), in which \( \phi (\neg x_3) = (x_1 \lor \mathcal{F}_2) \) and \( \phi' \) is empty. Therefore, \( \phi_2 = \psi_2 \land \phi_2. \) Note that \( C_1 = \{ x_1, t_2 \}, \) hence \( \mathcal{C}_2 = \{1\} \). Recall that \( \mathcal{L}^e = \{1, 2\} \), and that \( \mathcal{L}^e = \{3\} \). Then, \( \text{Scan} (\phi_2) \) executes due to \( \text{Remove} (x_3, \phi) \) L:6.

\( \text{Scan} (\phi_2) : \mathcal{L}^e = \{1, 2\} \) such that \( t_2 \in \psi_2 \) and \( t_3 \in \psi_2 \). Thus, \( t_2 \) and \( t_3 \) are necessary, hence \( x_2 \) and \( x_1 \) are incompatible trivially. Then, \( \text{Remove} (x_1, \phi_2) \) and \( \text{Remove} (x_2, \phi_2) \) execute.

The fact that the order of incomparability check is arbitrary (Theorem 36) is illustrated as follows. \( \text{Scope} (x_3, \phi) \) returns \( x_3 \) is incompatible nontrivially, since \( x_3 \Rightarrow t_1 \land x_1 \). Therefore, \( \neg \mathcal{F}_1 \lor \neg x_1 \Rightarrow \neg x_3, \) hence \( x_1 \lor \mathcal{F}_1 \Rightarrow \mathcal{F}_3 \). Then, \( \mathcal{F}_1 \Rightarrow \mathcal{F}_1 \) due to \( C_1 = \{ x_1, x_3 \}, \) and \( \mathcal{F}_1 \Rightarrow \neg x_1 \).

Thus, \( x_1 \) is still incompatible, but trivially (cf. \( \text{Scope} (x_1, \phi) \)), even if \( \neg x_3 \) holds. That is, \( x_1, \) the nontrivially incompatible in \( \phi \) due to \( x_1 \Rightarrow t_3 \land x_3, i.e., \neg \mathcal{F}_3 \lor \neg x_3 \Rightarrow \neg x_1, \) is incompatible trivially in \( \psi \) due to \( \mathcal{F}_1 \Rightarrow \neg x_1 \). See \( \text{Scan} (\phi_2) \) above. Also, since \( x_3 \notin \mathcal{F}_k \) and \( \mathcal{F}_3 \notin \mathcal{F}_k \) in \( \phi, \) for any \( s \geq 2, \) \( \neg \varphi (x_3) \) for all \( s \geq 2, \) even if any \( r_1 \) is removed from some \( C_k \) in \( \phi, s \geq 2 \).

4 Conclusion

X3SAT has proved to be effective to show \( P = NP. \) A polynomial time algorithm checks unsatisfiability of a truth assignment \( \phi (r_1) \) such that \( \neg \phi (r_1) \) iff \( \psi (r_1) \) involves \( x_1 \land t_1 \) for some \( s. \) Thus, \( \phi (r_1) \) reduces to \( \psi (r_1) \) \( \psi (r_1) \) denotes a conjunction of literals that are true, since each \( r_1 \) such that \( \neg \psi (r_1) \) is removed from \( \phi. \) Therefore, \( \phi \) is satisfiable iff any truth assignment \( \psi (r_1) \) holds, thus it is easy to verify satisfiability of \( \phi \) through the truth of \( \psi (r_1). \)

References


A Proof of Theorem 34/35

This section gives a rigorous proof of Theorem 34/35. Recall that the \( \varphi \) scan is interrupted iff \( \psi \) involves \( x_i \lor t_i \) for some \( i = s, \) that is, \( \psi \) is unsatisfiable, which is trivial to verify.

Recall also that the \( \varphi \) scan terminates iff \( \psi (r_i) = T \) for any \( r_i \in \mathcal{L}^e, r_i \in \{ x_1, t_1 \}. \) Moreover, \( \psi = \psi \lor \phi \) such that \( \psi = T \) (see Scan L:9 and Note 22). Therefore, when the scan terminates, satisfiability of \( \phi \) is to be proved, which is addressed in this section. Let \( \phi := \phi, \) i.e., \( \mathcal{L} := \mathcal{L}^e. \)
Then, construction of $\psi(r_i)$ over the set $L$ for some $i \in V$. Firstly, $\psi(r_i)$ holds because $a$ holds by assumption (see Note 19 and Scope L:5), and $b$ holds by definition (Scan L:9). Also, $\psi(r_i)$ is true due to $\psi(r_i) = \psi(r_i | r_j)$ (see Lemmas 32–33), where $\psi(\cdot) = \bigwedge r_i$ by Lemma 20. Next, we will show $b \Rightarrow a$ by showing that satisfiability of $\phi$ is preserved throughout the assignment.

Thus, construction of $\psi(r_i | r_j)$ in the next step is independent from the preceding steps, and depends only upon $\psi(r_j | r_k)$ in the present step. The construction process is specified below.

Step 1: Pick any $r_{i_0}$ in $\phi$. The reductions due to $r_{i_0}$ partition $L$ into $\Sigma(r_{i_0})$ and $\Sigma'(r_{i_0})$. Hence, $i_0 \notin \Sigma(r_{i_0})$. Thus, $r_{i_0} \Rightarrow \psi(r_{i_0})$ such that $\phi(r_{i_0}) = \psi(r_{i_0}) \land \phi'(r_{i_0})$ in Step 0. Then, pick an arbitrary $r_{i_1}$ in $\phi'(r_{i_0})$ for Step 1.

Step 2: The preceding steps have partitioned $L$ into $\Sigma(r_{i_0}) \cup \Sigma(r_{i_1}) \cup \Sigma'(r_{i_0})$ and $\Sigma'(r_{i_1})$, and $r_{i_1}$ into $\phi'(r_{i_1})$ partitions $\Sigma'(r_{i_1})$ into $\Sigma(r_{i_1})$ and $\Sigma'(r_{i_1})$, i.e., $\Sigma'(r_{i_1}) = \bigcup \Sigma'(r_{i_1}) \neq \emptyset$. Therefore, $\psi(r_{i_1}) \land \psi(r_{i_1})$ and $\psi(r_{i_1})$ are disjoint, as well as true. As a result, $\psi(r_{i_1}) \land \psi(r_{i_1}) \land \psi(r_{i_1}) \neq \psi(r_{i_1})$. Thus, $r_{i_1}$ is constructed such that $\phi(r_{i_1}) = \psi(r_{i_1}) \land \phi'(r_{i_1})$. Note that $\Sigma'(r_{i_1}) = \bigcup \Sigma'(r_{i_1}) \neq \emptyset$. Therefore, $\psi(r_{i_1}) \land \psi(r_{i_1}) \land \psi(r_{i_1}) \land \phi(r_{i_1})$. Hence, $\Sigma'(r_{i_1}) = \bigcup \Sigma'(r_{i_1}) \neq \emptyset$. Therefore, $\psi(r_{i_1}) \land \psi(r_{i_1})$ and $\psi(r_{i_1})$ are disjoint, as well as true. As a result, $\psi(r_{i_1}) \land \psi(r_{i_1}) \land \psi(r_{i_1}) \neq \psi(r_{i_1})$. Thus, $r_{i_1}$ is constructed such that $\phi(r_{i_1}) = \psi(r_{i_1}) \land \phi'(r_{i_1})$. Note that $\Sigma'(r_{i_1}) = \bigcup \Sigma'(r_{i_1}) \neq \emptyset$. Consequently, $\phi$ is composed of $\psi(\cdot)$ disjoints and satisfied, thus $\exists i_0 \epsilon. b \Rightarrow c$ holds. Finally, we show $c \Rightarrow a$. Scope $(r_i, \phi)$ transforms $r_i \land \phi$ into $\psi(r_i) \land \phi'(r_i)$, thus $(r_i \land \phi) \equiv (\psi(r_i) \land \phi'(r_i))$. Since $\phi$ and $\psi$ are satisfied, $\phi'(r_i)$ is satisfied. Therefore, unsatisfiability of $\psi(r_i)$ for some $s$ is necessary and sufficient for the unsatisfiability of $\phi(r_i)$ for any $s$.