EasyChair Preprint

№ 1734

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 Eylül University, Izmir, Turkey
latif.salum@deu.edu.tr & latif.salum@gmail.com

Abstract

This paper shows **P** = **NP** via exactly-1 3SAT (X3SAT). $C_k = (r_i \odot r_j \odot r_u)$ denotes a clause, an exactly-1 disjunction $\odot$ of literals, such that $\phi = \bigwedge C_k$, an X3SAT formula. $\phi(r_j) := r_j \wedge \phi$ denotes that the literal r_j is true, $r_j \in \{x_j, \overline{x}_j\}$. This truth assignment leads to reductions due to $\odot$ of any $C_k = (r_j \odot \overline{x}_i \odot x_u)$ into $c_k = r_j \wedge x_i \wedge \overline{x}_u$, and $C_k = (\overline{r}_j \odot r_u \odot r_v)$ into $C_{k'} = (r_u \odot r_v)$. As a result, $\phi(r_j) := r_j \wedge \phi$ transforms into $\phi(r_j) = \psi(r_j) \wedge \phi'(r_j)$, unless $\psi(r_j)$ involves $x_i \wedge \overline{x}_i$, that is, unless $\not\models \psi(r_j)$. Then, $\psi(r_j) = \bigwedge(c_k \wedge C_{k'})$ such that $C_{k'} = r_i$, and $\phi'(r_j) = \bigwedge(C_k \wedge C_{k'})$. Thus, $\psi(r_j)$ and $\phi'(r_j)$ are *disjoint*. It is *trivial* to check if $\not\models \psi(r_j)$, and *redundant* to check if $\not\models \phi'(r_j)$, in order to verify $\not\models \phi(r_j)$. Proof of this redundancy is sketched as follows. $\psi(r_i)$ is true, $\psi(r_i) \models \psi(r_i|r_j)$ holds, hence $\psi(r_i|r_j)$ is true for every r_i, because each r_j such that $\not\models \psi(r_j)$ is removed from ϕ. Then, any $\overline{r}_j$ consists in ψ so that ϕ transforms into $\psi \wedge \phi'$. If ψ involves $x_j \wedge \overline{x}_j$, then $\not\models \phi$. Otherwise, ϕ is satisfied, since any $\psi(.)$ is *disjoint* and *true*, and $\psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \dots, \psi(r_{i_n}|r_{i_m})$ compose ϕ. Thus, $\phi'(r_j)$ is *satisfied*, since $(r_j \wedge \phi) \equiv \big(\psi(r_j) \wedge \phi'(r_j)\big)$. 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 vs NP, NP-complete, 3SAT, one-in-three SAT, exactly-1 3SAT, X3SAT

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Acknowledgements

1 Introduction: Effectiveness of X3SAT in proving P = NP

P vs **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 $\odot$", the tool used to develop a polynomial time algorithm. It facilitates checking incompatibility of a literal r_j for satisfying some formula ϕ. When every r_j incompatible is removed, ϕ becomes un/satisfiable. Thus, each r_i becomes compatible to participate in some satisfiable assignment. Then, an assignment is constructed.

The truth assignment $r_j = \mathbf{T}$ (or r_j) is incompatible if $\phi(r_j)$ is unsatisfiable, denoted by $\not\models \phi(r_j)$, where $\phi(r_j) := r_j \wedge \phi$, and $r_j \in \{x_j, \overline{x}_j\}$. Then, the ϕ scan algorithm, introduced below, "scans" ϕ by checking incompatibility of every r_i, and removing each r_j incompatible.

Let $\phi = C_1 \wedge \cdots \wedge C_m$ be any X3SAT formula such that a clause $C_k = (r_i \odot r_j \odot r_u)$ is an exactly-1 disjunction $\odot$ of literals r_i, hence satisfied iff *exactly one* of $\{r_i, r_j, r_u\}$ is true. Note that a clause $(r_i \vee r_j \vee r_u)$ in a 3SAT formula is satisfied iff at least one of them is true.

© Latif Salum;
licensed under Creative Commons License CC-BY
42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Incompatibility of r_j is checked by a *deterministic* chain of *reductions* of any C_k in $\phi(r_j)$, which is constructed via $\odot$. This chain is initiated by $r_j = \mathbf{T}$, and followed by $\neg\overline{r}_j$, because $r_j \Rightarrow \neg\overline{r}_j$. That is, each $(r_j \odot \overline{x}_i \odot x_u)$ *collapses* to $(r_j \wedge x_i \wedge \overline{x}_u)$ due to $r_j \Rightarrow r_j \wedge \neg\overline{x}_i \wedge \neg x_u$, since there exists exactly one true literal in any clause C_k by the definition of X3SAT. Also, each $(\overline{r}_j \odot \overline{x}_u \odot x_v)$ *shrinks* to $(\overline{x}_u \odot x_v)$ due to $\neg\overline{r}_j$. Thus, r_j transforms $\phi(r_j) := r_j \wedge \phi$ into $\phi(r_j) = r_j \wedge x_i \wedge \overline{x}_u \wedge \phi^*$, and $x_i \wedge \overline{x}_u$ proceeds the reductions in ϕ^*, which involves $(\overline{x}_u \odot x_v)$.

The reductions over $\phi_s(r_j)$ terminate iff $r_j \wedge \phi_s$ transforms into $\psi_s(r_j) \wedge \phi'_s(r_j)$ such that $\psi_s(r_j)$ and $\phi'_s(r_j)$ are disjoint, where s denotes the current scan, and $\psi_s(r_j)$ is a conjunction of literals that are *true*. They are interrupted iff $\psi_s(r_j)$ involves $x_i \wedge \overline{x}_i$, thus $\not\models \phi_s(r_j)$, that is, r_j is incompatible. By *assumption*, $\not\models \phi_s(r_j)$ is verified *solely* via $\not\models \psi_s(r_j)$ (see Figure 1).

The reductions over ϕ terminate iff ϕ transforms into $\psi \wedge \phi'$ such that ψ and ϕ' are disjoint, where $\psi = \overline{x}_i \wedge x_u \wedge \cdots \wedge \overline{x}_v$ (see Figure 1). Then, ϕ is updated, that is, $\phi \leftarrow \phi'$. The ϕ_s scan is interrupted iff ψ_s involves $x_i \wedge \overline{x}_i$ for some s and i, thus $\not\models \phi$, that is, ϕ is unsatisfiable.

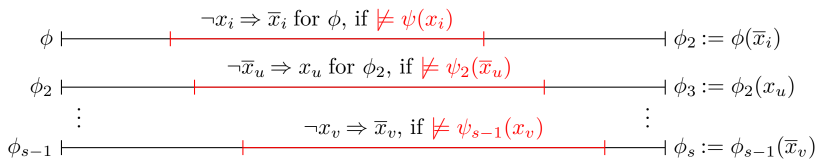

Figure 1 The ϕ_s scan: $\not\models \phi_s(r_j)$ is verified *solely* by $\not\models \psi_s(r_j)$, and whether $\not\models \phi'_s(r_j)$ is *ignored*

▷ Claim 1. It is *redundant* to check if $\not\models \phi'_s(r_j)$, thus $\not\models \phi(r_j)$ iff $\not\models \phi_s(r_j)$ iff $\not\models \psi_s(r_j)$ for some s. As a result, $\phi(r_i)$ reduces to $\psi(r_i)$ from $\phi(r_i) = \psi(r_i) \wedge \phi'(r_i)$, thus $\psi(r_i) \equiv \phi(r_i)$. Therefore, ϕ is satisfiable iff any truth assignment $\psi(r_i)$ *holds* (the scan *terminates*).

Sketch of proof. $\psi(r_i)/\psi(r_i|r_j)$ is constructed over $\phi/\phi'(r_j)$, thus $\psi(r_i)$ *covers* $\psi(r_i|r_j)$, hence $\psi(r_i) \models \psi(r_i|r_j)$ holds. Because $\psi(r_j)$ and $\phi'(r_j)$ are disjoint, $\psi(r_j)$ and $\psi(r_i|r_j)$ are disjoint (see Figure 2). Therefore, $\psi(r_{i_0})$, $\psi(r_{i_1}|r_{i_0})$, $\psi(r_{i_2}|r_{i_0}, r_{i_1})$, and $\psi(r_{i_3}|r_{i_0}, r_{i_1}, r_{i_2})$ form *disjoint* minterms $\psi(.) = \bigwedge r_i$ over ϕ such that $\psi(r_{i_0})$, $\psi(r_{i_1}|r_{i_0})$, $\psi(r_{i_2}|r_{i_0}, r_{i_1})$, and $\psi(r_{i_3}|r_{i_0}, r_{i_1}, r_{i_2})$ are true, because $\psi(r_i)$ is true for every r_i (the ϕ scan terminates), and $\psi(r_i) \models \psi(r_i|.)$ holds. Thus, ϕ is composed of $\psi(.)$ that are *disjoint* and *true* (see Figure 3), hence ϕ is satisfied. ◁

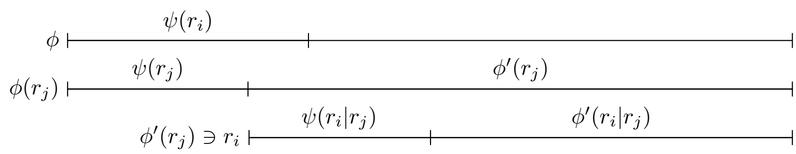

Figure 2 Since $\psi(r_i) = \bigwedge r_i$ is true and $\psi(r_i) \supseteq \psi(r_i|r_j)$, $\psi(r_i|r_j)$ is true, hence $\psi(r_i) \models \psi(r_i|r_j)$

A satisfiable assignment α is constructed by composing $\psi(.)$ that are *disjoint* and *true*. For example, $\alpha = \{\psi, \psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \psi(r_{i_2}|r_{i_0}, r_{i_1}), \psi(r_{i_3}|r_{i_0}, r_{i_1}, r_{i_2})\}$ (see Figure 3).

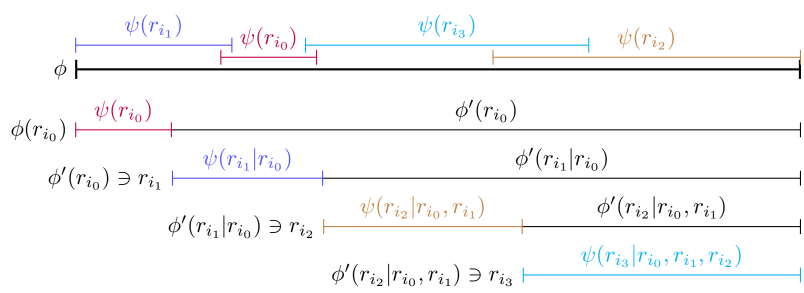

Figure 3 $\psi(r_{i_1}) \models \psi(r_{i_1}|r_{i_0})$, $\psi(r_{i_2}) \models \psi(r_{i_2}|r_{i_0}, r_{i_1})$, and $\psi(r_{i_3}) \models \psi(r_{i_3}|r_{i_0}, r_{i_1}, r_{i_2})$

2 Basic Definitions

A *literal* r_i is a variable x_i or its negation $\overline{x}_i$, i.e., $r_i \in \{x_i, \overline{x}_i\}$. A *clause* $C_k = (r_i \odot r_j \odot r_u)$, or $C_k = (r_{ik} \odot r_{jk} \odot r_{uk})$, is an exactly-1 disjunction $\odot$ 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). *$\varphi = \psi \wedge \phi$ such that $\psi = \bigwedge c_k$ and $\phi = \bigwedge C_k$.*

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

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

$\mathfrak{L} = \{1, 2, \ldots, n\}$ denotes the index set of the literals r_i in φ, and $\mathfrak{C} = \{1, 2, \ldots, m\}$ is an index set of the clauses C_k in ϕ, while $\mathfrak{C}^{r_i} = \{k \in \mathfrak{C} \,|\, r_i \in C_k\}$ denotes C_k that contain r_i.

▶ **Example 4.** Let $\hat{\varphi} = (x_{11} \odot \overline{x}_{31}) \wedge (x_{12} \odot \overline{x}_{22} \odot x_{32}) \wedge (x_{23} \odot \overline{x}_{33} \odot \overline{x}_{43}) \wedge \overline{x}_4$. Note that $C_3 = (x_2 \odot \overline{x}_3 \odot \overline{x}_4)$, and that $\overline{x}_4$ is a *conjunct*, thus $\overline{x}_4 = \mathbf{T}$ is *necessary* for satisfying $\hat{\varphi}$. Also, $\mathfrak{C} = \{1, 2, 3\}$, $\mathfrak{C}^{x_1} = \{1, 2\}$, and $\mathfrak{C}^{\overline{x}_4} = \{3\}$. Let $\varphi = (x_1 \odot \overline{x}_3) \wedge (x_1 \odot \overline{x}_4 \odot x_2) \wedge (x_2 \odot \overline{x}_3) \wedge x_4$. Then, $\mathfrak{C}^{x_4} = \emptyset$, and $C_1 = \{x_1, \overline{x}_3\}$, $C_2 = \{x_1, \overline{x}_4, x_2\}$ and $C_3 = \{x_2, \overline{x}_3\}$, while $\psi = \{x_4\}$ in φ.

▶ **Definition 5** (Collapse). *A clause $C_k = (r_i \odot x_j \odot \overline{x}_u)$ is said to* collapse *to the minterm $c_k = (r_i \wedge \overline{x}_j \wedge x_u)$, thus $r_i \notin C_k$, if r_i is* necessary*, denoted by $(r_i \odot x_j \odot \overline{x}_u) \searrow (r_i \wedge \overline{x}_j \wedge x_u)$.*

▶ **Definition 6** (Shrinkage). *A clause $C_k = (r_i \odot r_j \odot r_u)$ is said to shrink to another clause $C_{k'} = (r_j \odot r_u)$, if $\neg r_i$ (r_i* the incompatible is removed*), denoted by $(r_i \odot r_j \odot r_u) \rightarrowtail (r_j \odot r_u)$.*

▶ **Definition 7** (Truth assignment $r_i = \mathbf{T}$ over ϕ). *$\phi(r_i) = r_i \wedge \phi$ for any $r_i \in C_k$ and $C_k \in \phi$.*

▶ Note 8. r_i is *necessary* for $\phi(r_i)$, hence $\overline{r}_i$ is *removed*, i.e., $r_i \Rightarrow \neg\overline{r}_i$. Then, by the definition of X3SAT, $r_i \Rightarrow r_i \wedge \neg x_j \wedge \neg\overline{x}_u$ to satisfy a clause $(r_i \odot x_j \odot \overline{x}_u)$. As a result, $\neg x_j \Rightarrow \overline{x}_j$ and $\neg\overline{x}_u \Rightarrow x_u$, thus $\overline{x}_j$ and x_u become necessary. Therefore, the truth assignment $\phi(r_i)$ results in $(r_i \odot x_j \odot \overline{x}_u) \searrow (r_i \wedge \overline{x}_j \wedge x_u)$ and $(\overline{r}_i \odot r_v \odot r_y) \rightarrowtail (r_v \odot r_y)$ due to Definition 5 and 6.

▶ Remark (Reduction). The collapse or shrinkage of any clause C_k denotes its *reduction*, which in turn reduces φ_s, denoted by $\varphi_s \to \varphi_{s+1}$. Then, the number of $C_k \in \phi_{s+1}$ is less than the number of $C_k \in \phi_s$, or the number of literals in some $C_k \in \phi_{s+1}$ is less than that in some $C_k \in \phi_s$. Also, a collapse reduces nondeterminism to construct a satisfiable assignment.

▶ **Definition 9.** *ϕ denotes a general formula if $\{x_i, \overline{x}_i\} \nsubseteq C_k$ for any $i \in \mathfrak{L}$ and $k \in \mathfrak{C}$, hence $\mathfrak{C}^{x_i} \cap \mathfrak{C}^{\overline{x}_i} = \emptyset$. ϕ denotes a* special *formula if $\{x_i, \overline{x}_i\} \subseteq C_k$ for some k, hence $\mathfrak{C}^{x_i} \cap \mathfrak{C}^{\overline{x}_i} = \{k\}$.*

The φ scan algorithm accepts a general formula ϕ. Recall that $\varphi = \psi \wedge \phi$.

▶ **Lemma 10** (Conversion of a special formula). *Each* clause *$C_k = (r_j \odot x_i \odot \overline{x}_i)$ is replaced by the* conjunct *$\overline{r}_j$ so that $\mathfrak{C}^{x_i} \cap \mathfrak{C}^{\overline{x}_i} = \emptyset$ for any $i \in \mathfrak{L}$, if $\phi = \bigwedge C_k$ is a special formula.*

Proof. ϕ is unsatisfiable due to $r_j \Rightarrow \overline{x}_i \wedge x_i$. Then, $x_i \vee \overline{x}_i \Rightarrow \overline{r}_j$. That is, $\overline{r}_j$ is *necessary* for satisfying $C_k = (r_j \odot x_i \odot \overline{x}_i)$, which is sufficient also, thus $\overline{r}_j$ is equivalent to C_k. Therefore, each *clause* $C_k = (r_j \odot x_i \odot \overline{x}_i)$ is replaced by the *conjunct* $\overline{r}_j$ so that $\mathfrak{C}^{x_i} \cap \mathfrak{C}^{\overline{x}_i} = \emptyset$. ◀

▶ **Example 11.** $\phi = (x_1 \odot \overline{x}_2 \odot x_2) \wedge (x_1 \odot \overline{x}_3 \odot x_4) \wedge (x_2 \odot \overline{x}_1)$ denotes a special formula due to $C_1 = \{x_1, \overline{x}_2, x_2\}$. Note that $\mathfrak{C}^{\overline{x}_2} \cap \mathfrak{C}^{x_2} = \{1\}$. As a result, ϕ is converted by replacing the clause C_1 with the conjunct $\overline{x}_1$. Therefore, $\phi \leftarrow \overline{x}_1 \wedge (x_1 \odot \overline{x}_3 \odot x_4) \wedge (x_2 \odot \overline{x}_1)$. Likewise, if $\phi = (x_1 \odot \overline{x}_2 \odot x_2) \wedge (x_1 \odot \overline{x}_1 \odot \overline{x}_4) \wedge (x_2 \odot \overline{x}_1)$, then $\phi \leftarrow \overline{x}_1 \wedge x_4 \wedge (x_2 \odot \overline{x}_1)$. On the other hand, if ϕ involves $(x_u \odot \overline{x}_i \odot x_i) \wedge (\overline{x}_u \odot x_j \odot \overline{x}_j)$, then ϕ is unsatisfiable due to $\overline{x}_u \wedge x_u$.

3 The φ Scan

This section addresses the φ scan. Section 3.2 introduces the core algorithms. Section 3.3 tackles satisfiability of φ, and Section 3.4 tackles construction of a satisfiable assignment.

φ_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 $\varphi_s \equiv \varphi$. Then, $\phi_s^{r_i} = (r_{ik_1} \odot r_{u_1k_1} \odot r_{u_2k_1}) \wedge \cdots \wedge (r_{ik_r} \odot r_{v_1k_r} \odot r_{v_2k_r})$ denotes the formula over clauses $C_k \ni r_i$ in ϕ_s, where $r_i \in \{x_i, \overline{x}_i\}$. Hence, $\mathfrak{C}_s^{r_i} = \{k_1, \ldots, k_r\}$.

$\models_\alpha \varphi$ denotes that the assignment $\alpha = \{r_1, r_2, \ldots, r_n\}$ satisfies φ, and $\not\models \varphi$ denotes φ is unsatisfiable, while $\psi \models \psi'$ denotes ψ' is the logical consequence of ψ—as $\psi = \mathbf{T}$, $\psi' = \mathbf{T}$.

$\tilde{\psi}_s(r_i)$ is called the *local* effect of r_i and $\tilde{\phi}_s(\neg r_i)$ is the effect of $\neg r_i$. $\tilde{\varphi}_s(r_i)$ denotes its *overall* effect such that $\tilde{\varphi}_s(r_i) = \tilde{\psi}_s(r_i) \wedge \tilde{\phi}_s(\neg \overline{r}_i)$, specified below. Also, $\tilde{\psi}_s(r_i) = \bigwedge(c_k \wedge C_k)$ such that $|C_k| = 1$. Moreover, $\tilde{\phi}_s(\neg r_i) = \bigwedge C_k$ such that $|C_k| > 1$, or $\tilde{\phi}_s(\neg r_i)$ is empty.

3.1 Introduction: Incompatibility and Reductions

Example 12 and 13 introduces incompatibility and reductions, which drive the φ scan.

▶ **Example 12.** Consider $\phi(x_1)$ over $\varphi = \phi = (x_1 \odot \overline{x}_3) \wedge (x_1 \odot \overline{x}_2 \odot x_3) \wedge (x_2 \odot \overline{x}_3)$. Thus, x_1 is necessary (see Note 8), hence $x_1 \models \tilde{\psi}(x_1)$ such that $\tilde{\psi}(x_1) = (x_1 \wedge x_3) \wedge (x_1 \wedge x_2 \wedge \overline{x}_3)$. That is, $x_1 \Rightarrow \neg\overline{x}_3$ holds for $C_1 = (x_1 \odot \overline{x}_3)$, hence $\neg\overline{x}_3 \Rightarrow x_3$. Likewise, $x_1 \Rightarrow \neg\overline{x}_2 \wedge \neg x_3$ holds for $C_2 = (x_1 \odot \overline{x}_2 \odot x_3)$, hence $\neg\overline{x}_2 \Rightarrow x_2$ and $\neg x_3 \Rightarrow \overline{x}_3$. Thus, $\tilde{\varphi}(x_1) = \tilde{\psi}(x_1) \wedge \tilde{\phi}(\neg\overline{x}_1)$ becomes the overall effect, where $\tilde{\phi}(\neg\overline{x}_1)$ is empty. Then, the reductions initiated by x_1 are to proceed due to x_2. Nevertheless, they are interrupted by $x_3 \wedge \overline{x}_3$ due to $\tilde{\psi}(x_1)$, hence $\not\models \phi(x_1)$, where $\phi(x_1) = \tilde{\varphi}(x_1) \wedge (x_2 \odot \overline{x}_3)$. Therefore, x_1 is *incompatible* and *removed* from ϕ, thus $\neg x_1 \Rightarrow \overline{x}_1$.

▶ **Example 13.** $\overline{x}_1$ initiates *reductions* over φ (see Note 8). Then, $\tilde{\psi}(\overline{x}_1) = \overline{x}_1 \wedge \overline{x}_3$, $\tilde{\phi}(\neg x_1) = (\overline{x}_2 \odot x_3)$, and $\tilde{\varphi}(\overline{x}_1) = \tilde{\psi}(\overline{x}_1) \wedge \tilde{\phi}(\neg x_1)$ such that $\varphi_2 = \tilde{\varphi}(\overline{x}_1) \wedge (x_2 \odot \overline{x}_3)$. Note that $(x_2 \odot \overline{x}_3)$ is beyond $\tilde{\varphi}(\overline{x}_1)$ the overall effect. Note also that $\{\overline{x}_3\} \notin \tilde{\phi}(\neg x_1)$, while $\overline{x}_3 \in \tilde{\psi}(\overline{x}_1)$, because $C_1 \rightarrowtail c_1$, since $\tilde{\phi}(\neg x_1)$ contains no singleton. Then, φ_2 is the current formula due to the first reduction by $\overline{x}_1$ over φ. Thus, $\varphi \to \varphi_2$ due to $(x_1 \odot \overline{x}_3) \rightarrowtail (\overline{x}_3)$ and $(x_1 \odot \overline{x}_2 \odot x_3) \rightarrowtail (\overline{x}_2 \odot x_3)$. As a result, $\varphi_2 = \overline{x}_1 \wedge \overline{x}_3 \wedge (\overline{x}_2 \odot x_3) \wedge (x_2 \odot \overline{x}_3)$, in which $\psi_2 = \{\overline{x}_1, \overline{x}_3\}$ denotes the conjuncts, and $C_1 = \{\overline{x}_2, x_3\}$ and $C_2 = \{x_2, \overline{x}_3\}$ denote the clauses. Note that $\mathfrak{C}_2^{x_3} = \{1\}$ and $\mathfrak{C}_2^{\overline{x}_3} = \{2\}$. Then, $\overline{x}_3$ leads to the next reduction over φ_2: $\tilde{\psi}_2(\overline{x}_3) = (\overline{x}_2 \wedge \overline{x}_3)$, $\tilde{\phi}_2(\neg x_3)$ is empty, and $\tilde{\varphi}_2(\overline{x}_3) = \tilde{\psi}_2(\overline{x}_3) \wedge \tilde{\phi}_2(\neg x_3)$. Thus, $\varphi_2 \to \varphi_3$ due to $(x_2 \odot \overline{x}_3) \searrow (\overline{x}_2 \wedge \overline{x}_3)$ and $(\overline{x}_2 \odot x_3) \rightarrowtail (\overline{x}_2)$. Then, $\varphi_3 = \tilde{\varphi}(\overline{x}_1) \wedge \tilde{\varphi}_2(\overline{x}_3) = \overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3$, which denotes the cumulative effects of $\overline{x}_1$ and $\overline{x}_3$.

3.2 The Core Algorithms: `Scope` and `Scan`

This section specifies `Scope` and `Scan`, which incorporate the overall effect $\tilde{\varphi}_s(r_j)$, defined below. Recall that $\overline{r}_j$ is *removed*, if r_j is *necessary* for satisfying some formula, i.e., $r_j \Rightarrow \neg\overline{r}_j$. Note that $\phi_s^{r_j} = (r_{jk_1} \odot r_{i_1k_1} \odot r_{i_2k_1}) \wedge \cdots \wedge (r_{jk_r} \odot r_{u_1k_r} \odot r_{u_2k_r})$ for Lemma 14 and 15 below.

▶ **Lemma 14.** *$r_j \models \tilde{\psi}_s(r_j)$ such that $\tilde{\psi}_s(r_j) = r_j \wedge \overline{r}_{i_1} \wedge \overline{r}_{i_2} \wedge \cdots \wedge \overline{r}_{u_1} \wedge \overline{r}_{u_2}$, unless $\not\models \tilde{\psi}_s(r_j)$.*

Proof. Follows from Definition 5. That is, $r_j \Rightarrow (r_j \wedge \overline{r}_{i_1} \wedge \overline{r}_{i_2}) \wedge \cdots \wedge (r_j \wedge \overline{r}_{u_1} \wedge \overline{r}_{u_2})$. Hence, $r_j \Rightarrow r_j \wedge \overline{r}_{i_1} \wedge \overline{r}_{i_2} \wedge \cdots \wedge \overline{r}_{u_1} \wedge \overline{r}_{u_2}$. ◀

▶ **Lemma 15.** *If $\neg r_j$, then $\tilde{\phi}_s(\neg r_j)$ holds such that $\tilde{\phi}_s(\neg r_j) = (r_{i_1} \odot r_{i_2}) \wedge \cdots \wedge (r_{u_1} \odot r_{u_2})$.*

Proof. Follows from Definition 6. $\tilde{\phi}_s(\neg r_j) = \{\{\}\}$, or $|C_k| > 1$ for any C_k in $\tilde{\phi}_s(\neg r_j)$. ◀

▶ **Lemma 16** (Overall effect of r_j)**.** *$r_j \models \tilde{\varphi}_s(r_j)$ such that $\tilde{\varphi}_s(r_j) = \tilde{\psi}_s(r_j) \wedge \tilde{\phi}_s(\neg\overline{r}_j)$.*

Proof. Follows from $r_j \models r_j \wedge \neg\overline{r}_j$, as well as from Lemma 14, and Lemma 15 via $\phi_s^{\overline{r}_j}$. ◀

The algorithm `OvrlEft` (r_j, ϕ_*) below constructs the overall effect $\tilde{\varphi}_*(r_j)$ by means of the local effect $\tilde{\psi}_*(r_j)$ (see Lines 1-6, or L:1-6), as well as of the local effect $\tilde{\phi}_*(\neg\overline{r}_j)$ (L:7-10).

Algorithm 1 `OvrlEft` (r_j, ϕ_*) ▷ Construction of the overall effect $\tilde{\varphi}_*(r_j)$ due to r_j over ϕ_*

1: **for all** $k \in \mathfrak{C}_*^{r_j}$ over ϕ_* **do** ▷ Construction of the local effect $\tilde{\psi}_*(r_j)$ due to r_j (Lemma 14)
2: **for all** $r_i \in \big(C_k - \{r_j\}\big)$ **do** ▷ $\tilde{\psi}_*(r_j)$ gets r_j via r_e (see `Scope` L:4), or via $\overline{r}_j$ (`Remove` L:2)
3: $c_k \leftarrow c_k \cup \{\overline{r}_i\}$; ▷ $(r_{jk} \odot r_{i_1k} \odot r_{i_2k}) \searrow (\overline{r}_{i_1k} \wedge \overline{r}_{i_2k})$. That is, $C_k \searrow c_k$ (see Definition 2/5)
4: **end for**
5: $\tilde{\psi}_*(r_j) \leftarrow \tilde{\psi}_*(r_j) \cup c_k$; ▷ c_k consists in $\psi_s(r_j)$ (see `Scope` L:4), or in ψ_s (see `Remove` L:2)
6: **end for** ▷ L:1-6 are independent from L:7-10, since $\mathfrak{C}_*^{r_j} \cap \mathfrak{C}_*^{\overline{r}_j} = \emptyset$, i.e., $\mathfrak{C}_*^{x_j} \cap \mathfrak{C}_*^{\overline{x}_j} = \emptyset$ (Lemma 10)
7: **for all** $k \in \mathfrak{C}_*^{\overline{r}_j}$ over ϕ_* **do** ▷ Construction of the local effect $\tilde{\phi}_*(\neg\overline{r}_j)$ due to $\neg\overline{r}_j$ (Lemma 15)
8: $C_k \leftarrow C_k - \{\overline{r}_j\}$; ▷ $(\overline{r}_{jk} \odot r_{u_1k} \odot r_{u_2k}) \rightarrowtail (r_{u_1k} \odot r_{u_2k})$, or $(\overline{r}_{jk} \odot r_{uk}) \rightarrowtail (r_{uk})$ (Definition 6)
9: **if** $|C_k| = 1$ **then** $\tilde{\psi}_*(r_j) \leftarrow \tilde{\psi}_*(r_j) \cup C_k$; $C_k \leftarrow \emptyset$; ▷ $\tilde{\phi}_*(\neg\overline{r}_j)$ contains no singleton, $C_k \rightarrowtail c_k$
10: **end for** ▷ 3\2-literal C_k in $\phi_*^{\overline{r}_j}$ shrinks due to $\neg\overline{r}_j$ to 2-literal C_k in $\phi_*^{\overline{r}_j}$\to *conjunct* r_u in $\tilde{\psi}_*(r_j)$
11: **return** $\tilde{\psi}_*(r_j)$ & $\tilde{\phi}_*(\neg\overline{r}_j) \leftarrow \phi_*^{\overline{r}_j}$; ▷ $\tilde{\phi}_*(\neg\overline{r}_j) = \bigcup C_k$ such that $|C_k| > 1$, or $\tilde{\phi}_*(\neg\overline{r}_j) = \big\{\{\}\big\}$

▶ **Definition 17.** $\not\models \varphi_s(r_j)$ *iff* r_j *is incompatible, that is, the assumption* $r_j = \mathbf{T}$ *cannot hold.*

▶ Note. If $\not\models \varphi_s(r_j)$, r_j is incompatible, it is *removed* from ϕ_s, that is, $\neg r_j$ holds over ϕ_s.

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

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

▶ **Lemma 20** (Scope construction). $r_j \models \psi_s(r_j)$ *such that* $\psi_s(r_j) = \bigwedge c_k$*, unless* $\not\models \psi_s(r_j)$.

Proof. $\phi_s(r_j) = r_j \wedge \phi_s$ by Definition 7, as $r_j = \mathbf{T}$. Then, a *deterministic* chain of reductions is initiated (Note 8). That is, $r_j \Rightarrow r_j \wedge x_i \wedge \overline{x}_u$ due to any clause $(r_j \odot \overline{x}_i \odot x_u)$ containing r_j, as well as $\neg\overline{r}_j \Rightarrow (\overline{x}_u \odot x_v)$ due to any clause $(\overline{r}_j \odot \overline{x}_u \odot x_v)$ containing $\overline{r}_j$. 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 $\psi_s(r_j) = \bigwedge c_k$ and $\phi'_s(r_j) = \bigwedge C_k$. ◀

Algorithm 2 `Scope` (r_j, ϕ_s) ▷ Construction of $\psi_s(r_j)$ and $\phi'_s(r_j)$ due to r_j over ϕ_s; $\varphi_s = \psi_s \wedge \phi_s$

1: $\psi_s(r_j) \leftarrow \{r_j\}$; $\phi_* \leftarrow \phi_s$; ▷ $\phi_s(r_j) := r_j \wedge \phi_s$. ψ_s and ϕ_s are disjoint due to `Scan` L:1-3
2: **for all** $r_e \in \big(\psi_s(r_j) - R\big)$ **do** ▷ Reductions of C_k initiated by r_j over ϕ_s start off
3: `OvrlEft` (r_e, ϕ_*); ▷ It returns $\tilde{\psi}_*(r_e)$ for L:4 & $\tilde{\phi}_*(\neg\overline{r}_e)$ for L:6
4: $\psi_s(r_j) \leftarrow \psi_s(r_j) \cup \{r_e\} \cup \tilde{\psi}_*(r_e)$; ▷ $\tilde{\psi}_*(r_e)$ (see `OvrlEft` L:5,9) consists in the scope $\psi_s(r_j)$
5: **if** $\psi_s(r_j) \supseteq \{x_i, \overline{x}_i\}$ **then return** NULL; ▷ $r_j \Rightarrow x_i \wedge \overline{x}_i$, $i \in \mathfrak{L}^\phi$. $\not\models \psi_s(r_j)$, thus $\not\models \phi_s(r_j)$
6: $\tilde{\phi}_*(\neg r) \leftarrow \tilde{\phi}_*(\neg r) \cup \tilde{\phi}_*(\neg\overline{r}_e)$; ▷ $\tilde{\phi}_*(\neg r) = \big\{\{\}\big\}$ or $\tilde{\phi}_*(\neg r) = \bigcup C_k$, $|C_k| > 1$ (`OvrlEft` L:8-11)
7: $\phi_* \leftarrow \tilde{\phi}_*(\neg r) \wedge \phi'_*$; $R \leftarrow R \cup \{r_e\}$; ▷ $\tilde{\phi}_*(\neg r)$ and ϕ'_* consist in beyond the scope $\phi'_s(r_j)$
▷ $\phi'_* = \bigwedge C_k$ for $k \in \mathfrak{C}'_*$, where $\mathfrak{C}'_* = \mathfrak{C}_* - (\mathfrak{C}_*^{x_e} \cup \mathfrak{C}_*^{\overline{x}_e})$, and $\mathfrak{C}_*^{x_e} \cap \mathfrak{C}_*^{\overline{x}_e} = \emptyset$ due to Lemma 10
8: **end for** ▷ The reductions terminate if $\psi_s(r_j) = R$, which denotes conjuncts already reduced C_k
9: **return** $\psi_s(r_j)$ & $\phi'_s(r_j) \leftarrow \phi_*$; ▷ $\phi_s(r_j) = \psi_s(r_j) \wedge \phi'_s(r_j)$; $\psi_s(r_j) = \bigwedge c_k = \bigwedge r_j$, $\phi'_s(r_j) = \bigwedge C_k$

▶ Note 21. $\mathfrak{L}_s(r_j)$ being an index set of $\psi_s(r_j)$, $\mathfrak{L}_s(r_j) \cap \mathfrak{L}'_s(r_j) = \emptyset$ and $\mathfrak{L}_s(r_j) \cup \mathfrak{L}'_s(r_j) = \mathfrak{L}^\phi$, if `Scope` (r_j, ϕ_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` $(\varphi_{\hat{s}})$ terminates, then $\psi_{\hat{s}}$ and $\phi_{\hat{s}}$ are disjoint, and compose $\varphi_{\hat{s}}$ such that $\psi_{\hat{s}} = \bigwedge c_k$ (see Definition 2), and that $\phi_{\hat{s}} = \bigwedge C_k$, in which $|C_k| > 1$, because each $C_k = \{r_i\}$ in ϕ_s for any s transforms into r_i in $\psi_{\hat{s}}$. That is, $C_k = (r_i \odot r_j)$ or $C_k = (r_i \odot r_j \odot r_u)$ in $\phi_{\hat{s}}$.

▶ **Example 23.** Consider $\psi(x_1)$, `Scope` (x_1, ϕ), for $\phi = (x_1 \odot \overline{x}_3) \wedge (x_1 \odot \overline{x}_2 \odot x_3) \wedge (x_2 \odot \overline{x}_3)$. $\psi(x_1) \leftarrow \{x_1\}$ and $\phi_* \leftarrow \phi$ (L:1). Then, $\phi_*^{\overline{x}_1}$ is empty, and $\phi_*^{x_1} = (x_1 \odot \overline{x}_3) \wedge (x_1 \odot \overline{x}_2 \odot x_3)$ due to `OvrlEft` (x_1, ϕ_*). Also, $\mathfrak{C}_*^{x_1} = \{1, 2\}$, thus $c_1 \leftarrow \{x_3\}$ and $\tilde{\psi}_*(x_1) \leftarrow \tilde{\psi}_*(x_1) \cup c_1$, as well as $c_2 \leftarrow \{x_2, \overline{x}_3\}$ and $\tilde{\psi}_*(x_1) \leftarrow \tilde{\psi}_*(x_1) \cup c_2$ (see `OvrlEft` L:1-6). Then, $\tilde{\psi}_*(x_1) = \{x_3, x_2, \overline{x}_3\}$ & $\tilde{\phi}_*(\neg\overline{x}_1) \leftarrow \phi_*^{\overline{x}_1}$ (`OvrlEft` L:11). As a result, $\psi(x_1) \leftarrow \psi(x_1) \cup \{x_1\} \cup \tilde{\psi}_*(x_1)$ (`Scope` L:4), and $\psi(x_1) \supseteq \{x_3, \overline{x}_3\}$ (L:5), that is, $x_1 \Rightarrow x_3 \wedge \overline{x}_3$, hence x_1 is incompatible in the *first* scan.

▶ **Definition 24.** *$\mathfrak{L}^\psi = \{i \in \mathfrak{L} \,|\, r_i \in \psi_s\}$ and $\mathfrak{L}^\phi = \{i \in \mathfrak{L} \,|\, r_i \in C_k$ in $\phi_s\}$ due to $\varphi_s = \psi_s \wedge \phi_s$.*

Figure 4 illustrates `Scan` (φ_s). It decomposes $\phi_s = \bigwedge C_k$ into $\psi_s(x_1), \psi_s(\overline{x}_1), \ldots, \psi_s(x_n)$, $\psi_s(\overline{x}_n)$, thus checks if $\not\models \phi_s(x_i)$ and $\not\models \phi_s(\overline{x}_i)$, where $\psi_s(.) = \bigwedge c_k$ is true by Definition 2.

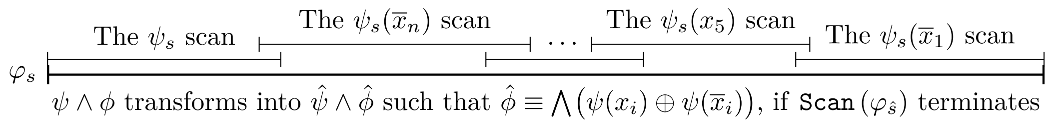

Figure 4 `Scan` decomposes ϕ_s into $\psi_s(x_1), \psi_s(\overline{x}_1), \ldots, \psi_s(\overline{x}_n)$, and transforms $\psi \wedge \phi$ into $\hat{\psi} \wedge \hat{\phi}$

`Scan` (φ_s) checks incompatibility of r_i for every $i \in \mathfrak{L}^\phi$. If $\overline{r}_i \in \psi_s$, then r_i is incompatible *trivially* (L:1-2). If $r_i \Rightarrow x_j \wedge \overline{x}_j$, then r_i is incompatible *nontrivially* (L:6). See also Note 18. For example, $\overline{x}_1$ is incompatible trivially due to $x_1 \wedge (x_1 \odot x_2 \odot \overline{x}_3)$, since $1 \in \mathfrak{L}^\phi$ and $x_1 \in \psi_s$. Note that $\overline{x}_1 \Rightarrow \overline{x}_1 \wedge x_1$. If `Scan` (φ_s) is interrupted (see `Remove` L:3), then φ is unsatisfiable. If the scan terminates (L:9), then a satisfiable assignment α is constructed (see Section 3.4).

Algorithm 3 `Scan` (φ_s) ▷ Checks if $\not\models \varphi_s(r_i)$ for all $i \in \mathfrak{L}^\phi$. See also Note 18. $\varphi_s = \psi_s \wedge \phi_s$

```
1: for all i ∈ 𝔏^φ and r̄_i ∈ ψ_s do ▷ φ_s(r_i) = ψ_s ∧ r_i ∧ φ_s, thus ⊭ (ψ_s ∧ r_i), that is, r_i ⇒ x_i ∧ x̄_i
2:     Remove (r_i, φ_s);  ▷ r̄_i is necessary, thus r_i is incompatible trivially, hence r̄_i ⇒ ¬r_i
3: end for ▷ If i ∈ 𝔏^ψ, r_i has been already removed, hence r̄_i ∈ ψ_s and r̄_i ∉ C_k ∀k ∈ ℭ_s, i.e., i ∉ 𝔏^φ
4: for all i ∈ 𝔏^φ do ▷ 𝔏^ψ ∩ 𝔏^φ = ∅ due to L:1-3. Hence, i ∈ 𝔏^ψ iff r_i = x_i is fixed or r_i = x̄_i is fixed
5:     for all r_i ∈ {x_i, x̄_i} do ▷ Each and every x_i and x̄_i assumed to be true is to be verified
6:         if Scope (r_i, φ_s) is NULL then Remove (r_i, φ_s); ▷ Incompatible nontrivially if ⊭ φ_s(r_i)
7:     end for ▷ If r_i ⇒ x_j ∧ x̄_j, hence ¬x_j ∨ ¬x̄_j ⇒ ¬r_i, then ¬r_i ⇒ r̄_i, where i ≠ j due to L:1-3
8: end for ▷ ¬r_i iff r̄_i, since ¬r_i ⇒ r̄_i due to nontrivial, and ¬r_i ⇐ r̄_i due to trivial incompatibility
9: return φ̂ = ψ̂ ∧ φ̂, and ψ(r_i) & φ′(r_i) for all i ∈ 𝔏^φ; ▷ ψ̂ ← ψ_ŝ and φ̂ ← φ_ŝ. See also Note 22
```

▶ Note 25. $\mathfrak{L}^\psi$ and $\mathfrak{L}^\phi$ form a partition of $\mathfrak{L}$ due to Definition 24 and `Scan` L:1-3.

`Remove` (r_j, ϕ_s) leads to reductions of any $C_k \ni \overline{r}_j$ due to $\overline{r}_j$, which consists in ψ_{s+1} (see L:1-2), as well as of any $C_k \ni r_j$ due to $\neg r_j$, which consists in ϕ_{s+1} (see L:1,5). Note that ψ_s denotes the current conjuncts (in φ_s), and that ψ denotes the initial conjuncts (in φ).

Algorithm 4 `Remove` (r_j, ϕ_s) ▷ r_j is incompatible/removed iff $\overline{r}_j$ is necessary, i.e., $\neg r_j$ iff $\overline{r}_j$

```
1: OvrlEft (r̄_j, φ_s); ▷ OvrlEft is defined over φ_s = ⋀C_k, |C_k| > 1, and returns ψ̃_s(r̄_j) & φ̃_s(¬r_j)
2: ψ_{s+1} ← ψ_s ∪ {r̄_j} ∪ ψ̃_s(r̄_j); ▷ ψ_{s+1} = ⋀c_k is true by Definition 2, unless ψ_{s+1} involves x_i ∧ x̄_i
3: if ψ_{s+1} ⊇ {x_i, x̄_i} for some i then return φ is unsatisfiable;   ▷ φ_s = ψ_s ∧ φ_s
4: 𝔏^φ ← 𝔏^φ − {j}; 𝔏^ψ ← 𝔏^ψ ∪ {j};
5: φ_{s+1} ← φ̃_s(¬r_j) ∧ φ′_s; Update {C_k} over φ_{s+1}; ▷ φ′_s denotes clauses beyond the entire ψ_s effect
       ▷ φ′_s = ⋀C_k for k ∈ ℭ′_s, where ℭ′_s = ℭ_s − (ℭ_s^{x̄_j} ∪ ℭ_s^{x_j}), and ℭ_s^{x̄_j} ∩ ℭ_s^{x_j} = ∅ due to Lemma 10
6: Scan (φ_{s+1}); ▷ r_i verified compatible for š ≤ s can be incompatible for s̃ > s due to ¬r_j in φ_s
```

3.3 Unsatisfiability of $\phi(r_j)$ vs Unsatisfiability of $\psi_s(r_j)$ for some s

This section tackles satisfiability of φ 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 $\not\models \phi'_s(r_j)$ for some s.*

Proof. Proof is obvious due to $\phi_s(r_j) = \psi_s(r_j) \wedge \phi'_s(r_j)$, transformed from $\phi_s(r_j) := r_j \wedge \phi_s$ through `Scope` (r_j, ϕ_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.

$\mathfrak{L}_s(r_i) = \mathfrak{L}(\psi_s(r_i))$ denotes the index set of the scope $\psi_s(r_i)$. Likewise, $\mathfrak{L}'_s(r_i) = \mathfrak{L}(\phi'_s(r_i))$. Also, we define the conditional scope $\psi_s(r_i|r_j)$ and beyond the scope $\phi'_s(r_i|r_j)$ over $\phi'_s(r_j)$ for any $j \neq i$, which are constructed by `Scope` $\big(r_i, \phi'_s(r_j)\big)$. Thus, $\mathfrak{L}_s(r_i|r_j) = \mathfrak{L}(\psi_s(r_i|r_j))$.

▶ **Lemma 27** (No conjunct exists in beyond the scope). *$\mathfrak{L}_s(r_j) \cap \mathfrak{L}'_s(r_j) = \emptyset$ for any $j \in \mathfrak{L}^\phi$.*

Proof. $\phi'_s(r_j) = \bigwedge C_k$ by `Scope` (r_j, ϕ_s). Let r_i the *conjunct* be in C_k, $i \in \big(\mathfrak{L}_s(r_j) \cap \mathfrak{L}'_s(r_j)\big)$. Then, for any $C_k \ni r_i$, $(r_i \odot x_j \odot \overline{x}_u) \searrow (r_i \wedge \overline{x}_j \wedge x_u)$, thus $r_i \notin C_k$. Moreover, for any $C_k \ni \overline{r}_i$, $(\overline{r}_i \odot r_v \odot r_y) \rightarrowtail (r_v \odot r_y)$, thus $\overline{r}_i \notin C_k$. See Definition 5/6. Hence, $i \notin \big(\mathfrak{L}_s(r_j) \cap \mathfrak{L}'_s(r_j)\big)$. ◀

▶ Note. No conjunct exists in any clause C_k due to Note 25, which states $\mathfrak{L}^\psi \cap \mathfrak{L}^\phi = \emptyset$.

▶ **Lemma 28.** *$\mathfrak{L}^\phi$ is partitioned into $\mathfrak{L}_s(r_j)$, $\mathfrak{L}_s(r_{j_1}|r_j), \ldots, \mathfrak{L}_s(r_{j_n}|r_{j_m})$ by means of* `Scope`.

▶ **Lemma 29.** *$\phi_s(r_j)$ is decomposed into* disjoint *$\psi_s(r_j)$, $\psi_s(r_{j_1}|r_j), \ldots, \psi_s(r_{j_n}|r_{j_m})$.*

Proof. `Scope` (r_j, ϕ_s) *partitions* $\mathfrak{L}^\phi$ into $\mathfrak{L}_s(r_j)$ and $\mathfrak{L}'_s(r_j)$ for any $j \in \mathfrak{L}^\phi$ (see Lemma 27). Thus, $\phi_s(r_j)$ is decomposed into *disjoint* $\psi_s(r_j)$ and $\phi'_s(r_j)$. `Scope` $\big(r_{j_1}, \phi'_s(r_j)\big)$ *partitions* $\mathfrak{L}'_s(r_j)$ into $\mathfrak{L}_s(r_{j_1}|r_j)$ and $\mathfrak{L}'_s(r_{j_1}|r_j)$ for any $j_1 \in \mathfrak{L}'_s(r_j)$. Thus, $\phi'_s(r_j)$ is decomposed into *disjoint* $\psi_s(r_{j_1}|r_j)$ and $\phi'_s(r_{j_1}|r_j)$. Finally, $\phi'_s(r_{j_m}|r_{j_l})$ is decomposed into *disjoint* $\psi_s(r_{j_n}|r_{j_m})$ and $\phi'_s(r_{j_n}|r_{j_m})$ for any $j_n \in \mathfrak{L}'_s(r_{j_m}|r_{j_l})$ such that $\mathfrak{L}'_s(r_{j_n}|r_{j_m}) = \emptyset$ (see also Note 21). ◀

Let the scan *terminate* (see `Scan` L:9), thus $\psi \wedge \phi$ transforms into $\hat{\psi} \wedge \hat{\phi}$. Let $\phi \leftarrow \hat{\phi}$, thus $\mathfrak{L} \leftarrow \mathfrak{L}^{\hat{\phi}}$. Also, $\psi(r_i) = \mathbf{T}$ for every $i \in \mathfrak{L}$ and $r_i \in \{x_i, \overline{x}_i\}$. 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 *$\psi(r_{j_1}|r_j)$, $\psi(r_{j_2}|r_{j_1}), \ldots, \psi(r_{j_n}|r_{j_m})$.*

Proof. Follows from Lemma 29, and from $\phi(r_j) = \psi(r_j) \wedge \phi'(r_j)$ due to `Scope` (r_j, ϕ). ◀

▶ **Lemma 31.** *$\phi \supseteq \phi'(r_j) \supseteq \phi'(r_{j_1}|r_j) \supseteq \phi'(r_{j_2}|r_{j_1}) \supseteq \cdots \supseteq \phi'(r_{j_n}|r_{j_m})$, since it* terminates.

Proof. Some C_k in ϕ collapse to some c_k in $\psi(r_j)$ due to `Scope` (r_j, ϕ) (see Lemma 20). As a result, the number of C_k in ϕ is greater than or equal to that of C_k in $\phi'(r_j)$, thus $|\mathfrak{C}| \geqslant |\mathfrak{C}'|$, where $\mathfrak{C}$ denotes an index set of C_k in ϕ. Also, some C_k in ϕ shrink to some $C_{k'}$ in $\phi'(r_j)$, thus $\forall k' \in \mathfrak{C}' \, \exists k \in \mathfrak{C} \, [C_k \supseteq C_{k'}]$. Hence, $\phi \supseteq \phi'(r_j)$. Likewise, $\phi'(r_j) \supseteq \phi'(r_{j_1}|r_j)$, since $\phi'(r_j)$ is decomposed into $\psi(r_{j_1}|r_j)$ and $\phi'(r_{j_1}|r_j)$ via `Scope` $\big(r_{j_1}, \phi'(r_j)\big)$. Therefore, $\phi \supseteq \phi'(r_j) \supseteq \phi'(r_{j_1}|r_j) \supseteq \phi'(r_{j_2}|r_{j_1}) \supseteq \cdots \supseteq \phi'(r_{j_n}|r_{j_m})$, where $\phi'(r_{j_n}|r_{j_m}) = \phi'(r_{j_n}|r_j, r_{j_1}, \ldots, r_{j_m})$. ◀

▶ **Lemma 32** (Any scope entails its conditional scope). *$\psi(r_i) \models \psi(r_i|r_j)$, since it* terminates.

Proof. $\phi \supseteq \phi'(r_j)$ due to Lemma 31. `Scope` (r_i, ϕ) constructs the scope $\psi(r_i)$ over ϕ, while `Scope` $\big(r_i, \phi'(r_j)\big)$ constructs the conditional scope $\psi(r_i|r_j)$ over $\phi'(r_j)$, thus $\psi(r_i) \supseteq \psi(r_i|r_j)$, where $\psi(r_i) = \bigwedge c_k$ by Definition 2 and Lemma 20. Since $\psi(r_i) \supseteq \psi(r_i|r_j)$ and $\psi(r_i)$ is true for all r_i in ϕ, $\psi(r_i|r_j)$ is true for all r_i in $\phi'(r_j)$. Hence, $\psi(r_i) \models \psi(r_i|r_j)$ (see Figure 2). ◀

▶ **Lemma 33.** *$\psi(r_i|r_j)$, $\psi(r_i|r_j, r_{j_1}), \ldots, \psi(r_i|r_j, r_{j_1}, \ldots, r_{j_m})$ is true for every $j \in \mathfrak{L}$, and for every $i \in \mathfrak{L}'(r_j)$, $i \in \mathfrak{L}'(r_{j_1}|r_j), \ldots, i \in \mathfrak{L}'(r_{j_m}|r_j, r_{j_1}, \ldots, r_{j_l})$, because the scan* terminates.

Proof. Recall that the scan *terminates*. Thus, $\hat{\varphi} = \hat{\psi} \wedge \hat{\phi}$, and $\phi := \hat{\phi}$ and $\mathfrak{L} := \mathfrak{L}^{\phi}$ (see also Note 22). Hence, a truth assignment $\psi(r_i)$ holds for every $i \in \mathfrak{L}$ and $r_i \in \{x_i, \overline{x}_i\}$. Moreover, $\phi \supseteq \phi'(r_j) \supseteq \phi'(r_{j_1}|r_j) \supseteq \phi'(r_{j_2}|r_{j_1}) \supseteq \cdots \supseteq \phi'(r_{j_n}|r_{j_m})$ due to Lemma 31 for any $j \in \mathfrak{L}$, and $j_1 \in \mathfrak{L}'(r_j), \ldots, j_n \in \mathfrak{L}'(r_{j_m}|r_{j_l})$. Then, $\psi(r_i) \supseteq \psi(r_i|r_j), \ldots, \psi(r_i) \supseteq \psi(r_i|r_j, r_{j_1}, \ldots, r_{j_m})$, in which $\psi(r_i) \supseteq \psi(r_i|r_j, r_{j_1})$ via `Scope` $\big(r_i, \phi'(r_{j_1}|r_j)\big)$, thus $\psi(r_i) \models \psi(r_i|r_j, r_{j_1})$. Therefore, any $\psi(r_i|r_j)$, $\psi(r_i|r_j, r_{j_1}), \ldots, \psi(r_i|r_j, r_{j_1}, \ldots, r_{j_m})$ is true, which generalizes Lemma 32. ◀

▶ **Theorem 34** (Unsatisfiability). *$\not\models \phi(r_j)$, r_j is incompatible, iff $\not\models \psi_s(r_j)$ for some s.*

▶ **Corollary 35** (Satisfiability). *$\models_\alpha \phi$ iff a truth assignment $\psi(r_i)$ holds $\forall i \in \mathfrak{L}$, $r_i \in \{x_i, \overline{x}_i\}$.*

Proof. $\psi(r_{j_1}|r_j)$, $\psi(r_{j_2}|r_{j_1}), \ldots, \psi(r_{j_n}|r_{j_m})$ form disjoint minterms over $\phi'(r_j)$ (Lemma 30) such that $\psi(r_{j_1}|r_j)$, $\psi(r_{j_2}|r_{j_1}), \ldots, \psi(r_{j_n}|r_{j_m})$ are true (Lemma 33) for any $j \in \mathfrak{L}$, $j_1 \in \mathfrak{L}'(r_j)$, $j_2 \in \mathfrak{L}'(r_{j_1}|r_j), \ldots, j_n \in \mathfrak{L}'(r_{j_m}|r_{j_l})$. Then, $\phi'(r_j)$ is composed of $\psi(.)$ the minterms true and disjoint, hence $\phi'(r_j)$ is *satisfied*, thus unsatisfiability of $\phi'_s(r_j)$ is *ignored* to verify $\not\models \phi_s(r_j)$. Therefore, Theorem 34 holds (cf. Proposition 26). Moreover, $\psi(r_i) \equiv \phi(r_i)$, since $\phi'(r_i)$ is satisfied, and $\phi(r_i) = \psi(r_i) \wedge \phi'(r_i)$. Therefore, Corollary 35 holds (see also Appendix A). ◀

▶ **Theorem 36** (Incompatibility is monotonic). *$\not\models \varphi_s(r_j)$ for all $s > \tilde{s}$ if $\not\models \varphi_{\tilde{s}}(r_j)$, even if $\neg r_i$.*

Proof. $\not\models \varphi_s(r_j)$, if $\not\models (\psi_s \wedge r_j)$ or $\not\models \phi_s(r_j)$ (`Scan` L:1,6). $\psi_s \supseteq \psi_{\tilde{s}}$ for all $s > \tilde{s}$ (`Remove` L:2), thus $\not\models (\psi_s \wedge r_j)$ for all $s > \tilde{s}$, if $\not\models (\psi_{\tilde{s}} \wedge r_j)$. Let $\not\models \phi_{\tilde{s}}(r_j)$ due to $x_i \wedge \overline{x}_i$, hence $\overline{x}_i \vee x_i \Rightarrow \overline{r}_j$, thus $\overline{r}_j \in \psi_s$, and $\not\models (\psi_s \wedge r_j)$ for all $s > \tilde{s}$. If $\not\models \varphi_{\check{s}}(r_i)$ for $\check{s} \leqslant \tilde{s}$, then $\neg r_i \Rightarrow \overline{r}_i$ and $\overline{r}_i \Rightarrow \overline{r}_j$, thus $\overline{r}_j \in \psi_s$ *still* holds, and $\not\models (\psi_s \wedge r_j)$ for all $s > \check{s}$, hence all $s > \tilde{s}$. If $\not\models \varphi_s(r_i)$ for $s > \tilde{s}$, then $\not\models (\psi_s \wedge r_j)$ *still* holds for all $s > \tilde{s}$, since $x_j \notin C_k$ and $\overline{x}_j \notin C_k$, while $r_i \in C_k$ in ϕ_s. ◀

▶ **Proposition 37.** *The time complexity of `Scan` is $O(mn^3)$.*

Proof. `OvrlEft`, and `Remove`, takes $4m$ steps by $\big(|\mathfrak{C}_*^{r_j}| \times |C_k|\big) + |\mathfrak{C}_*^{\overline{r}_j}| = 3m + m$. `Scope` takes $n4m$ steps by $|\psi_s(r_j)| \times 4m$. Then, `Scan` takes $n^2 4m$ steps due to L:1-3 by $|\mathfrak{L}^{\phi}| \times |\psi_s| \times 4m$, as well as $8n^2m + 8nm$ steps due to L:4-8 by $2|\mathfrak{L}^{\phi}| \times (4nm + 4m)$. Also, the number of the scans is $\hat{s} \leqslant |\mathfrak{L}^{\phi}|$ due to `Remove` L:6. Therefore, the time complexity of `Scan` is $O(n^3m)$. ◀

▶ **Example 38.** Let $\varphi = \big\{\{x_3, x_4, \overline{x}_5\}, \{x_3, x_6, \overline{x}_7\}, \{x_4, x_6, \overline{x}_7\}\big\}$. Let `Scope` (x_3, ϕ) execute *first* in the *first* scan, which leads to the reductions below over ϕ due to x_3. Note that $\psi = \emptyset$.

$$\begin{aligned} \phi(x_3) = {} & (x_3 \odot x_4 \odot \overline{x}_5) \wedge (x_3 \odot x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7) \wedge x_3 \\ x_3 \Rightarrow {} & (x_3 \wedge \overline{x}_4 \wedge x_5) \wedge (x_3 \wedge \overline{x}_6 \wedge x_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7) \wedge x_3 \\ \overline{x}_4 \Rightarrow {} & (x_3 \wedge \overline{x}_4 \wedge x_5) \wedge (x_3 \wedge \overline{x}_6 \wedge x_7) \wedge (\quad x_6 \odot \overline{x}_7) \wedge x_3 \\ \overline{x}_6 \Rightarrow {} & (x_3 \wedge \overline{x}_4 \wedge x_5) \wedge (x_3 \wedge \overline{x}_6 \wedge x_7) \wedge (\quad \overline{x}_7) \wedge x_3 \end{aligned}$$

Because $\not\models \big(\psi(x_3) = x_3 \wedge \overline{x}_4 \wedge x_5 \wedge \overline{x}_6 \wedge x_7 \wedge \overline{x}_7\big)$, x_3 is incompatible, hence $\overline{x}_3$ is necessary, i.e., $\neg x_3 \Rightarrow \overline{x}_3$. Thus, $\varphi \to \varphi_2$ by $(x_3 \odot x_4 \odot \overline{x}_5) \rightarrowtail (x_4 \odot \overline{x}_5)$ and $(x_3 \odot x_6 \odot \overline{x}_7) \rightarrowtail (x_6 \odot \overline{x}_7)$. As a result, $\varphi_2 = (x_4 \odot \overline{x}_5) \wedge (x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7) \wedge \overline{x}_3$. Let `Scope` (x_5, ϕ_2) execute next.

$$\begin{aligned} \phi_2(x_5) = {} & (\quad x_4 \odot \overline{x}_5) \wedge (\quad x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7) \wedge x_5 \\ x_5 \Rightarrow {} & (\quad x_4 \quad) \wedge (\quad x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7) \wedge x_5 \\ x_4 \Rightarrow {} & (\quad x_4 \quad) \wedge (\quad x_6 \odot \overline{x}_7) \wedge (x_4 \wedge \overline{x}_6 \wedge x_7) \wedge x_5 \\ \overline{x}_6 \Rightarrow {} & (\quad x_4 \quad) \wedge (\quad \overline{x}_7) \wedge (x_4 \wedge \overline{x}_6 \wedge x_7) \wedge x_5 \end{aligned}$$

Because $\not\models \big(\psi_2(x_5) = x_4 \wedge \overline{x}_7 \wedge \overline{x}_6 \wedge x_7 \wedge \overline{x}_3 \wedge x_5\big)$, x_5 is removed from ϕ_2, i.e., $\neg x_5 \Rightarrow \overline{x}_5$. Thus, $\varphi_2 \to \varphi_3$ by $(x_4 \odot \overline{x}_5) \searrow (\overline{x}_4 \wedge \overline{x}_5)$, where $\varphi_3 = (\overline{x}_4 \wedge \overline{x}_5) \wedge (x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7) \wedge \overline{x}_3$, and $\overline{x}_4$ leads to the next reduction by $(x_4 \odot x_6 \odot \overline{x}_7) \rightarrowtail (x_6 \odot \overline{x}_7)$. Then, `Scan` (φ_4) *terminates*, and $\varphi_4 = \overline{x}_3 \wedge \overline{x}_4 \wedge \overline{x}_5 \wedge (x_6 \odot \overline{x}_7)$, that is, $\hat{\varphi} = \hat{\psi} \wedge \hat{\phi}$, and $\hat{\psi} = \{\overline{x}_3, \overline{x}_4, \overline{x}_5\}$ and $\hat{\phi} = \big\{\{x_6, \overline{x}_7\}\big\}$.

In Example 38, if $\texttt{Scope}\,(x_5, \phi)$ executes *first*, then $\psi(x_5) = x_5$ becomes the scope, and $\phi'(x_5) = (x_3 \odot x_4) \wedge (x_3 \odot x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7)$ becomes beyond the scope of x_5 over ϕ. Then, x_5 is compatible (in ϕ) 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 $\psi_s(x_5)$ for some s. Then, x_5 becomes incompatible (in ϕ_2), because $\not\models \psi_2(x_5)$ holds, after $\varphi \to \varphi_2$ by removing x_3 from ϕ 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 ϕ, 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

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

Algorithm 5 ▷ Construction of a satisfiable assignment α over ϕ, $\mathfrak{L} := \mathfrak{L}^{\hat{\phi}}$ and $\phi := \hat{\phi}$

```
Pick j ∈ 𝔏;   ▷ The scope ψ(r_i) and beyond the scope φ'(r_i) for all i ∈ 𝔏 are available initially
α ← ψ(r_j); 𝔏 ← 𝔏 − 𝔏(r_j); φ ← φ'(r_j);
repeat
    Pick i ∈ 𝔏; Scope (r_i, φ);   ▷ It constructs ψ(r_i|r_j) and φ'(r_i|r_j) with respect to φ'(r_j)
    α ← α ∪ ψ(r_i);   ▷ ψ(r_i) := ψ(r_i|r_j), because ψ(r_i) is unconditional with respect to φ updated
    𝔏 ← 𝔏 − 𝔏(r_i);   ▷ 𝔏 ← 𝔏'(r_i|r_j) due to the partition {𝔏(r_j), 𝔏(r_i|r_j), 𝔏'(r_i|r_j)} over 𝔏
    φ ← φ'(r_i);   ▷ φ'(r_i) := φ'(r_i|r_j), because φ'(r_i) is unconditional with respect to φ updated
until 𝔏 = ∅
return α;   ▷ ψ(r_{i_n}|r_{i_m}) = ψ(r_{i_n}|r_j, r_{i_1}, ..., r_{i_m}) (see also Appendix A)
```

▶ **Definition 39.** *Let* $\big\langle \langle r_{\imath_1,1}, r_{\imath_2,1}, r_{\imath_3,1}\rangle, \langle r_{\jmath_1,2}, r_{\jmath_2,2}, r_{\jmath_3,2}\rangle, \ldots, \langle r_{u_1,m}, r_{u_2,m}, r_{u_3,m}\rangle \big\rangle$ *be in ascending order with respect to the index set* $\mathfrak{L}$. *If* $\imath_3 < \jmath_1$ *for any* $\langle r_{\imath_1,k}, r_{\imath_2,k}, r_{\imath_3,k}\rangle$ *and any* $\langle r_{\jmath_1,k+1}, r_{\jmath_2,k+1}, r_{\jmath_3,k+1}\rangle$*, then* ${}^{\imath}\phi \cup {}^{\jmath}\phi = \phi$ *and* ${}^{\imath}\phi \cap {}^{\jmath}\phi = \emptyset$ *such that* $C_k \in {}^{\imath}\phi$ *and* $C_{k+1} \in {}^{\jmath}\phi$.

▶ Note. ${}^{\imath}\phi$ and ${}^{\jmath}\phi$ form a *partition* of ϕ, hence their satisfiability check can be *independent*.

▶ **Example 40.** Let ${}^1\phi = (x_1 \odot \overline{x}_2 \odot x_6) \wedge (x_3 \odot x_4 \odot \overline{x}_5) \wedge (x_3 \odot x_6 \odot \overline{x}_7) \wedge (x_4 \odot x_6 \odot \overline{x}_7)$, ${}^2\phi = (x_8 \odot x_9 \odot \overline{x}_{10})$, and ${}^3\phi = (x_{11} \odot \overline{x}_{12} \odot x_{13})$ to form $\varphi = {}^1\phi \wedge {}^2\phi \wedge {}^3\phi$ (see Definition 39). Then, $\texttt{Scan}\,(\varphi_4)$ returns φ is satisfiable. Therefore, $\hat{\varphi} = \hat{\psi} \wedge \hat{\phi}$, where $\psi := \hat{\psi} = \overline{x}_3 \wedge \overline{x}_4 \wedge \overline{x}_5$ and $\phi := \hat{\phi} = (x_1 \odot \overline{x}_2 \odot x_6) \wedge (x_6 \odot \overline{x}_7) \wedge {}^2\phi \wedge {}^3\phi$ (see Example 38). Then, α is constructed by composing $\psi(.)$ based on $\phi'(.)$ below, where $\mathfrak{L}^{\psi} = \{3, 4, 5\}$ and $\mathfrak{L} := \mathfrak{L}^{\hat{\phi}} = \{1, 2, \ldots, 13\} - \mathfrak{L}^{\psi}$.

$$
\begin{aligned}
\psi(x_1) &= x_1 \wedge x_2 \wedge \overline{x}_6 \wedge \overline{x}_7 && \& & \phi'(x_1) &= {}^2\phi \wedge {}^3\phi \\
\psi(x_2) &= x_2 && \& & \phi'(x_2) &= (x_1 \odot x_6) \wedge (x_6 \odot \overline{x}_7) \wedge {}^2\phi \wedge {}^3\phi \\
\psi(\overline{x}_2) &= \overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_6 \wedge \overline{x}_7 && \& & \phi'(\overline{x}_2) &= {}^2\phi \wedge {}^3\phi \\
\psi(x_6) = \psi(x_7) &= \overline{x}_1 \wedge x_2 \wedge x_6 \wedge x_7 && \& & \phi'(x_6) = \phi'(x_7) &= {}^2\phi \wedge {}^3\phi \\
\psi(\overline{x}_6) = \psi(\overline{x}_7) &= \overline{x}_6 \wedge \overline{x}_7 && \& & \phi'(\overline{x}_6) = \phi'(\overline{x}_7) &= (x_1 \odot \overline{x}_2) \wedge {}^2\phi \wedge {}^3\phi \\
\psi(x_8) &= x_8 \wedge \overline{x}_9 \wedge x_{10} && \& & \phi'(x_8) &= (x_1 \odot \overline{x}_2 \odot x_6) \wedge (x_6 \odot \overline{x}_7) \wedge {}^3\phi \\
\psi(x_{11}) &= x_{11} \wedge x_{12} \wedge \overline{x}_{13} && \& & \phi'(x_{11}) &= (x_1 \odot \overline{x}_2 \odot x_6) \wedge (x_6 \odot \overline{x}_7) \wedge {}^2\phi
\end{aligned}
$$

▶ **Example 41.** A satisfiable assignment α is constructed by an order of indices over $\mathfrak{L}$, $\mathfrak{L} = \{1, \dots, 13\} - \mathfrak{L}^{\psi}$ (Example 40), such that $r_i := x_i$ for any $\psi(r_i)$ throughout the construction. First, pick $6 \in \mathfrak{L}$. As a result, $\alpha \leftarrow \psi(x_6)$ and $\mathfrak{L} \leftarrow \mathfrak{L} - \mathfrak{L}(x_6)$, where $\psi(x_6) = \{\overline{x}_1, x_2, x_6, x_7\}$, $\mathfrak{L}(x_6) = \{1, 2, 6, 7\}$, and $\mathfrak{L} \leftarrow \{8, 9, 10, 11, 12, 13\}$. Then, pick 8, hence $\alpha \leftarrow \alpha \cup \psi(x_8|x_6)$, where $\psi(x_8|x_6) = \{x_8, \overline{x}_9, x_{10}\}$. Also, $\mathfrak{L} \leftarrow \mathfrak{L} - \mathfrak{L}(x_8|x_6)$, where $\mathfrak{L}(x_8|x_6) = \{8, 9, 10\}$, hence $\mathfrak{L} \leftarrow \{11, 12, 13\}$. Finally, pick 11. Therefore, $\alpha \leftarrow \alpha \cup \psi(x_{11}|x_6, x_8)$ such that $\mathfrak{L} \leftarrow \emptyset$, which indicates its termination. Note that $\texttt{Scope}\left(x_{11}, \phi'(x_8|x_6)\right)$ constructs $\psi(x_{11}|x_6, x_8)$, in which $\phi'(x_8|x_6) = {}^3\phi$, and that $\phi'(x_{11}|x_6, x_8) = \emptyset$ iff $\mathfrak{L} \leftarrow \emptyset$. Note also that $\psi(x_8|x_6) = \psi(x_8)$ and $\psi(x_{11}|x_6, x_8) = \psi(x_{11})$, since ${}^1\phi$, ${}^2\phi$ and ${}^3\phi$ 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 φ is *decomposed* into ψ, $\psi(x_6)$, $\psi(x_8|x_6)$, and $\psi(x_{11}|x_6, x_8)$, which are *disjoint* (see also Note 22 and Lemma 29).

▶ **Example 42.** Let $(2, 1, 8, 11)$ be another order of indices in Example 40. This order leads to the assignment $\{\psi, \psi(x_2), \psi(x_1|x_2), \psi(x_8|x_2, x_1), \psi(x_{11}|x_2, x_1, x_8)\}$ for φ. This assignment corresponds to the partition $\left\{\mathfrak{L}^{\psi}, \{2\}, \{1, 6, 7\}, \{8, 9, 10\}, \{11, 12, 13\}\right\}$, where $\mathfrak{L}^{\psi} = \{3, 4, 5\}$ (see also Note 25 and Lemma 28). Note that the scope $\psi(x_1)$ is constructed over ϕ, and the conditional scope $\psi(x_1|x_2)$ is constructed over $\phi'(x_2)$, where $\phi \supseteq \phi'(x_2)$. Recall that $\phi := \hat{\phi}$. Hence, $\psi(x_1) \models \psi(x_1|x_2)$, in which $\psi(x_1) = x_1 \wedge x_2 \wedge \overline{x}_6 \wedge \overline{x}_7$, while $\psi(x_1|x_2) = x_1 \wedge \overline{x}_6 \wedge \overline{x}_7$. Moreover, $\psi(x_8) \models \psi(x_8|x_2, x_1)$ due to $\phi \supseteq \phi'(x_1|x_2)$, and $\psi(x_{11}) \models \psi(x_{11}|x_2, x_1, x_8)$ due to $\phi \supseteq \phi'(x_8|x_2, x_1)$, where $\phi'(x_1|x_2) = {}^2\phi \wedge {}^3\phi$ and $\phi'(x_8|x_2, x_1) = {}^3\phi$ (see Lemmas 31-33).

3.5 An Illustrative Example

This section illustrates $\texttt{Scan}\,(\varphi_s)$. Let $\varphi = \phi = (x_1 \odot \overline{x}_3) \wedge (x_1 \odot \overline{x}_2 \odot x_3) \wedge (x_2 \odot \overline{x}_3)$, which is adapted from Esparza [1], and denotes a general formula by Definition 9. Note that $C_1 = \{x_1, \overline{x}_3\}$, $C_2 = \{x_1, \overline{x}_2, x_3\}$, and $C_3 = \{x_2, \overline{x}_3\}$. Hence, $\mathfrak{C} = \{1, 2, 3\}$, and $\mathfrak{L} = \mathfrak{L}^{\phi} = \{1, 2, 3\}$.

$\texttt{Scan}\,(\varphi)$: There exists no conjunct in (the initial formula) φ. That is, ψ is empty (L:1). Recall that $\varphi := \varphi_1$, and that $r_i \in \{x_i, \overline{x}_i\}$. Recall also that *nontrivial* incompatibility of r_i is checked (L:4-8) via $\texttt{Scope}\,(r_i, \phi)$. Moreover, the order of incompatibility check is arbitrary (incompatibility is monotonic) by Theorem 36. Let $\texttt{Scope}\,(x_1, \phi)$ execute due to $\texttt{Scan}$ L:6.

$\texttt{Scope}\,(x_1, \phi)$: Since $\psi(x_1) \supseteq \{x_3, \overline{x}_3\}$, x_1 is incompatible *nontrivially* (see Example 23). Thus, $\overline{x}_1$ becomes necessary (a conjunct). Then, $\texttt{Remove}\,(x_1, \phi)$ executes due to $\texttt{Scan}$ L:6.

$\texttt{Remove}\,(x_1, \phi)$: $\mathfrak{C}^{\overline{x}_1} = \emptyset$ by $\texttt{OvrlEft}$ L:1. $\mathfrak{C}^{x_1} = \{1, 2\}$, thus $\phi^{x_1} = (x_1 \odot \overline{x}_3) \wedge (x_1 \odot \overline{x}_2 \odot x_3)$ by $\texttt{OvrlEft}$ L:7. As a result, $\tilde{\psi}(\overline{x}_1) = \{\overline{x}_3\}$ & $\tilde{\phi}(\neg x_1) = \{\{\}, \{\overline{x}_2, x_3\}\}$, the effects of $\overline{x}_1$ and $\neg x_1$. Note that $C_1 \leftarrow \emptyset$. Then, $\psi_2 \leftarrow \psi \cup \{\overline{x}_1\} \cup \tilde{\psi}(\overline{x}_1)$ ($\texttt{Remove}$ L:2), and $\mathfrak{L}^{\phi} \leftarrow \mathfrak{L}^{\phi} - \{1\}$ and $\mathfrak{L}^{\psi} \leftarrow \mathfrak{L}^{\psi} \cup \{1\}$ (L:4). Also, $\phi_2 \leftarrow \tilde{\phi}(\neg x_1) \wedge \phi'$, where $\tilde{\phi}(\neg x_1) = (\overline{x}_2 \odot x_3)$ and $\phi' = (x_2 \odot \overline{x}_3)$ (L:5). As a result, $\psi_2 = \overline{x}_1 \wedge \overline{x}_3$, and $\phi_2 = (\overline{x}_2 \odot x_3) \wedge (x_2 \odot \overline{x}_3)$. Note that $C_1 = \{\overline{x}_2, x_3\}$ and $C_2 = \{x_2, \overline{x}_3\}$. Consequently, $\varphi_2 = \psi_2 \wedge \phi_2$, and $\texttt{Scan}\,(\varphi_2)$ executes due to $\texttt{Remove}$ L:6.

$\texttt{Scan}\,(\varphi_2)$: $\mathfrak{C}_2 = \{1, 2\}$ and $\mathfrak{L}^{\phi} = \{2, 3\}$ hold in ϕ_2. Then, $\{x_2, \overline{x}_2\} \cap \psi_2 = \emptyset$ for $2 \in \mathfrak{L}^{\phi}$, while $\overline{x}_3 \in \psi_2$ for $3 \in \mathfrak{L}^{\phi}$ (L:1). As a result, $\overline{x}_3$ is *necessary* for satisfying φ_2, hence $\overline{x}_3 \Rightarrow \neg x_3$, that is, x_3 is incompatible *trivially*. Then, $\texttt{Remove}\,(x_3, \phi_2)$ executes due to $\texttt{Scan}$ L:2.

$\texttt{Remove}\,(x_3, \phi_2)$: $\mathfrak{C}_2^{\overline{x}_3} = \{2\}$, thus $\phi_2^{\overline{x}_3} = (x_2 \odot \overline{x}_3)$, and $\mathfrak{C}_2^{x_3} = \{1\}$, thus $\phi_2^{x_3} = (\overline{x}_2 \odot x_3)$. As a result, $\tilde{\psi}_2(\overline{x}_3) = \{\overline{x}_2\} \cup \{\overline{x}_2\}$ & $\tilde{\phi}_2(\neg x_3) = \{\{\}\}$, because $C_1 = \{\overline{x}_2\}$ consists in $\tilde{\psi}_2(\overline{x}_3)$, rather than in $\tilde{\phi}_2(\neg x_3)$ (see $\texttt{OvrlEft}$ L:9). Hence, $\psi_3 \leftarrow \psi_2 \cup \{\overline{x}_3\} \cup \tilde{\psi}_2(\overline{x}_3)$, $\mathfrak{L}^{\phi} \leftarrow \mathfrak{L}^{\phi} - \{3\}$, and $\mathfrak{L}^{\psi} \leftarrow \mathfrak{L}^{\psi} \cup \{3\}$, i.e., $\mathfrak{L}^{\phi} = \{2\}$. Therefore, $\phi_3 = \{\{\}\}$, thus $\mathfrak{C}_3 = \emptyset$, and $\psi_3 = \overline{x}_1 \wedge \overline{x}_3 \wedge \overline{x}_2$.

$\texttt{Scan}\,(\varphi_3)$: $\overline{x}_2 \in \psi_3$ for $2 \in \mathfrak{L}^{\phi}$ over ϕ_3. Then, $\texttt{Remove}\,(x_2, \phi_3)$ executes due to $\texttt{Scan}$ L:2.

$\texttt{Remove}\,(x_2, \phi_3)$: $\tilde{\psi}_3(\overline{x}_2) = \emptyset$ & $\tilde{\phi}_3(\neg x_2) = \{\{\}\}$ due to $\texttt{OvrlEft}\,(\overline{x}_2, \phi_3)$, because $\mathfrak{C}_3^{\overline{x}_2} = \emptyset$ and $\mathfrak{C}_3^{x_2} = \emptyset$, since $\mathfrak{C}_3 = \emptyset$. Hence, $\mathfrak{L}^{\phi} \leftarrow \{2\} - \{2\}$ and $\phi_4 \leftarrow \phi_3$. Then, $\texttt{Scan}\,(\varphi_4)$ executes.

$\texttt{Scan}\,(\varphi_4)$ *terminates*: $\hat{\varphi} = \hat{\psi} = \overline{x}_1 \wedge \overline{x}_3 \wedge \overline{x}_2$ (L:9), and φ collapses to a unique assignment.

Let `Scope` (x_3, ϕ) execute *before* `Scope` (x_1, ϕ) due to `Scan` L:6 (see Theorem 36).

`Scope` (x_3, ϕ): $\psi(x_3) \leftarrow \{x_3\}$ and $\phi_* \leftarrow \phi$ (L:1). Then, $\mathfrak{C}_*^{x_3} = \{2\}$ due to `OvrlEft` (x_3, ϕ_*) L:1, hence $\phi_*^{x_3} = (x_1 \odot \overline{x}_2 \odot x_3)$. As a result, $c_2 \leftarrow \{\overline{x}_1, x_2\}$ and $\tilde{\psi}_*(x_3) \leftarrow \tilde{\psi}_*(x_3) \cup c_2$ (L:3,5). Moreover, $\mathfrak{C}_*^{\overline{x}_3} = \{1, 3\}$ (L:7), hence $\phi_*^{\overline{x}_3} = (x_1 \odot \overline{x}_3) \wedge (x_2 \odot \overline{x}_3)$. Then, $C_1 \leftarrow \{x_1, \overline{x}_3\} - \{\overline{x}_3\}$, $\tilde{\psi}_*(x_3) \leftarrow \tilde{\psi}_*(x_3) \cup C_1$, and $C_1 \leftarrow \emptyset$. Likewise, $C_3 \leftarrow \{x_2, \overline{x}_3\} - \{\overline{x}_3\}$, $\tilde{\psi}_*(x_3) \leftarrow \tilde{\psi}_*(x_3) \cup C_3$, and $C_3 \leftarrow \emptyset$ (`OvrlEft` L:8-9). Consequently, $\tilde{\psi}_*(x_3) \leftarrow \{\overline{x}_1, x_2, x_1\}$ & $\tilde{\phi}_*(\neg x_3) \leftarrow \phi_*^{\overline{x}_3}$ (L:11). Note that $\phi_*^{\overline{x}_3} = \{\{\}, \{\}\}$, since $C_1 = C_3 = \emptyset$. Then, $\psi(x_3) \leftarrow \psi(x_3) \cup \{x_3\} \cup \tilde{\psi}_*(x_3)$ due to `Scope` L:4, hence $\psi(x_3) = \{x_3, \overline{x}_1, x_2, x_1\}$. Since $\psi(x_3) \supseteq \{\overline{x}_1, x_1\}$ (L:5), x_3 is incompatible *nontrivially*, i.e., $x_3 \Rightarrow \overline{x}_1 \wedge x_1$ and $\neg x_3 \Rightarrow \overline{x}_3$. Then, `Remove` (x_3, ϕ) executes due to `Scan` L:6.

`Remove` (x_3, ϕ): $\phi^{\overline{x}_3} = (x_1 \odot \overline{x}_3) \wedge (x_2 \odot \overline{x}_3)$ due to $\mathfrak{C}^{\overline{x}_3} = \{1, 3\}$, and $\phi^{x_3} = (x_1 \odot \overline{x}_2 \odot x_3)$ due to $\mathfrak{C}^{x_3} = \{2\}$. Then, `OvrlEft` $(\overline{x}_3, \phi)$ returns $\tilde{\psi}(\overline{x}_3) = \{\overline{x}_1, \overline{x}_2\}$ & $\tilde{\phi}(\neg x_3) = \{\{x_1, \overline{x}_2\}\}$ (`Remove` L:1), $\psi_2 \leftarrow \psi \cup \{\overline{x}_3\} \cup \tilde{\psi}(\overline{x}_3)$ (L:2), and $\mathfrak{L}^\phi \leftarrow \mathfrak{L}^\phi - \{3\}$ and $\mathfrak{L}^\psi \leftarrow \mathfrak{L}^\psi \cup \{3\}$ (L:4). As a result, $\psi_2 = \overline{x}_3 \wedge \overline{x}_1 \wedge \overline{x}_2$. Moreover, $\phi_2 \leftarrow \tilde{\phi}(\neg x_3) \wedge \phi'$ (L:5), in which $\tilde{\phi}(\neg x_3) = (x_1 \odot \overline{x}_2)$ and ϕ' is empty. Therefore, $\varphi_2 = \psi_2 \wedge \phi_2$. Note that $C_1 = \{x_1, \overline{x}_2\}$, hence $\mathfrak{C}_2 = \{1\}$. Recall that $\mathfrak{L}^\phi = \{1, 2\}$, and that $\mathfrak{L}^\psi = \{3\}$. Then, `Scan` (φ_2) executes due to `Remove` (x_3, ϕ) L:6.

`Scan` (φ_2): $\mathfrak{L}^\phi = \{1, 2\}$ such that $\overline{x}_2 \in \psi_2$ and $\overline{x}_1 \in \psi_2$. Thus, $\overline{x}_2$ and $\overline{x}_1$ are *necessary*, hence x_2 and x_1 are incompatible *trivially*. Then, `Remove` (x_1, ϕ_2) and `Remove` (x_2, ϕ_2) execute.

The fact that the order of incompatibility check is arbitrary (Theorem 36) is illustrated as follows. `Scope` (x_3, ϕ) returns x_3 is incompatible *nontrivially*, since $x_3 \Rightarrow \overline{x}_1 \wedge x_1$. Therefore, $\neg\overline{x}_1 \vee \neg x_1 \Rightarrow \neg x_3$, hence $x_1 \vee \overline{x}_1 \Rightarrow \overline{x}_3$. Then, $\overline{x}_3 \Rightarrow \overline{x}_1$ due to $C_1 = (x_1 \odot \overline{x}_3)$, and $\overline{x}_1 \Rightarrow \neg x_1$. Thus, x_1 is *still* incompatible, but trivially $\big($cf. `Scope` $(x_1, \phi)\big)$, even if $\neg x_3$ holds. That is, x_1 the *nontrivial* incompatible in ϕ due to $x_1 \Rightarrow \overline{x}_3 \wedge x_3$, i.e., $\neg\overline{x}_3 \vee \neg x_3 \Rightarrow \neg x_1$, is incompatible *trivially* in ψ_2 due to $\overline{x}_1 \Rightarrow \neg x_1$. See `Scan` (φ_2) above. Also, since $x_3 \notin C_k$ and $\overline{x}_3 \notin C_k$ in ϕ_s for any $s \geqslant 2$, $\not\models \varphi_s(x_3)$ for all $s \geqslant 2$, even if any r_i is removed from some C_k in ϕ_s, $s \geqslant 2$.

4 Conclusion

X3SAT has proved to be effective to show **P** = **NP**. A polynomial time algorithm checks unsatisfiability of a truth assignment $\phi(r_i)$ such that $\not\models \phi(r_i)$ iff $\psi_s(r_i)$ involves $x_j \wedge \overline{x}_j$ for some s. Thus, $\phi(r_i)$ reduces to $\psi(r_i)$. $\psi(r_i)$ denotes a conjunction of literals that are *true*, since each r_j such that $\not\models \psi_s(r_j)$ is removed from ϕ. Therefore, ϕ is satisfiable iff any truth assignment $\psi(r_i)$ holds, thus it is *easy* to verify satisfiability of ϕ through the *truth* of $\psi(r_i)$.

References

1. Javier Esparza. Decidability and complexity of Petri net problems – an introduction. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, volume 1491 of *LNCS*, pages 374–428. Springer Berlin Heidelberg, 1998.
2. Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM. URL: `http://doi.acm.org/10.1145/800133.804350`.

A Proof of Theorem 34/35

This section gives a rigorous proof of Theorem 34/35. Recall that the φ_s scan is *interrupted* iff ψ_s involves $x_i \wedge \overline{x}_i$ for some i and s, that is, φ is unsatisfiable, which is trivial to verify. Recall also that the $\varphi_{\hat{s}}$ scan *terminates* iff $\psi_{\hat{s}}(r_i) = \mathbf{T}$ for any $i \in \mathfrak{L}^\phi$, $r_i \in \{x_i, \overline{x}_i\}$. Moreover, $\hat{\varphi} = \hat{\psi} \wedge \hat{\phi}$ such that $\hat{\psi} = \mathbf{T}$ (see `Scan` L:9 and Note 22). Therefore, when the scan terminates, satisfiability of $\hat{\phi}$ is to be proved, which is addressed in this section. Let $\phi := \hat{\phi}$, i.e., $\mathfrak{L} := \mathfrak{L}^{\hat{\phi}}$.

▶ **Theorem 43** (cf. 34-35/Claim 1). *These statements are equivalent: a)* $\not\models \phi(r_j)$ *iff* $\not\models \psi_s(r_j)$ *for some s. b)* $\psi(r_i) = \mathbf{T}$ *for any* $i \in \mathfrak{L}$*. c)* $\models_\alpha \phi$ *by* $\alpha = \{\psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \ldots, \psi(r_{i_n}|r_{i_m})\}$.

Proof. We will show $a \Rightarrow b$, $b \Rightarrow c$, and $c \Rightarrow a$ (see Kenneth H. Rosen, Discrete Mathematics and its Applications, 7E, pg. 88). Firstly, $a \Rightarrow b$ holds, because a holds by assumption (see Note 19 and `Scope` L:5), and b holds by definition (`Scan` L:9). Also, $\psi(r_i|r_j)$ is true due to $\psi(r_i) \models \psi(r_i|r_j)$ (see Lemmas 32-33), where $\psi(.) = \bigwedge r_i$ by Lemma 20. Next, we will show $b \Rightarrow c$. We do this by showing that satisfiability of ϕ is *preserved* throughout the assignment $\alpha = \{\psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \ldots, \psi(r_{i_n}|r_{i_m})\}$ construction, because any *partial* truth assignment $\psi(r_i|r_j)$ is constructed *arbitrarily* through consecutive steps having the Markov property. 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 0: Pick any r_{i_0} in ϕ. The reductions due to r_{i_0} partition $\mathfrak{L}$ into $\mathfrak{L}(r_{i_0})$ and $\mathfrak{L}'(r_{i_0})$. Note that $i_0 \in \mathfrak{L}$, and that $i_0 \in \mathfrak{L}(r_{i_0})$. Hence, $i_0 \notin \mathfrak{L}'(r_{i_0})$ by Lemma 27. Thus, $r_{i_0} \Rightarrow \psi(r_{i_0})$ such that $\phi(r_{i_0}) = \psi(r_{i_0}) \wedge \phi'(r_{i_0})$ in Step 0. Then, pick an *arbitrary* r_{i_1} in $\phi'(r_{i_0})$ for Step 1.

Step 1: $\mathfrak{L}(r_{i_0}) \cap \mathfrak{L}'(r_{i_0}) = \emptyset$ in Step 0, and the reductions due to r_{i_1} over $\phi'(r_{i_0})$ partition $\mathfrak{L}'(r_{i_0})$ into $\mathfrak{L}(r_{i_1}|r_{i_0})$ and $\mathfrak{L}'(r_{i_1}|r_{i_0})$, thus $r_{i_1} \Rightarrow \psi(r_{i_1}|r_{i_0})$. See also Lemma 28. Therefore, $\mathfrak{L}(r_{i_0}) \cap \mathfrak{L}(r_{i_1}|r_{i_0}) = \emptyset$, because $\mathfrak{L}'(r_{i_0}) \supseteq \mathfrak{L}(r_{i_1}|r_{i_0})$. As a result, $\mathfrak{L}$ is partitioned into $\mathfrak{L}(r_{i_0})$, $\mathfrak{L}(r_{i_1}|r_{i_0})$, and $\mathfrak{L}'(r_{i_1}|r_{i_0})$ due to r_{i_0} and r_{i_1}. Thus, $\psi(r_{i_0})$ and $\psi(r_{i_1}|r_{i_0})$ are *disjoint*, as well as *true*. Hence, $\psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0}) = \mathbf{T}$, and $\phi(r_{i_0}, r_{i_1}) = \psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0}) \wedge \phi'(r_{i_1}|r_{i_0})$.

Step 2: The preceding steps have partitioned $\mathfrak{L}$ into $\mathfrak{L}(r_{i_0}) \cup \mathfrak{L}(r_{i_1}|r_{i_0})$ and $\mathfrak{L}'(r_{i_1}|r_{i_0})$, and r_{i_2} in $\phi'(r_{i_1}|r_{i_0})$ partitions $\mathfrak{L}'(r_{i_1}|r_{i_0})$ into $\mathfrak{L}(r_{i_2}|r_{i_1})$ and $\mathfrak{L}'(r_{i_2}|r_{i_1})$, i.e., $\mathfrak{L}'(r_{i_1}|r_{i_0}) \supseteq \mathfrak{L}(r_{i_2}|r_{i_1})$. Hence, $\big(\mathfrak{L}(r_{i_0}) \cup \mathfrak{L}(r_{i_1}|r_{i_0})\big) \cap \mathfrak{L}(r_{i_2}|r_{i_1}) = \emptyset$. Therefore, $\psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0})$ and $\psi(r_{i_2}|r_{i_1})$ are *disjoint*, as well as *true*. As a result, $\psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0}) \wedge \psi(r_{i_2}|r_{i_1}) = \mathbf{T}$, and $\phi(r_{i_0}, r_{i_1}, r_{i_2}) = \psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0}) \wedge \psi(r_{i_2}|r_{i_1}) \wedge \phi'(r_{i_2}|r_{i_1})$. Note that $\alpha \supseteq \{\psi(r_{i_0}), \psi(r_{i_1}|r_{i_0}), \psi(r_{i_2}|r_{i_1})\}$, and that $\mathfrak{L}$ is partitioned into $\mathfrak{L}(r_{i_0})$, $\mathfrak{L}(r_{i_1}|r_{i_0})$, $\mathfrak{L}(r_{i_2}|r_{i_1})$, and $\mathfrak{L}'(r_{i_2}|r_{i_1})$ such that $\mathfrak{L}'(r_{i_2}|r_{i_1}) \neq \emptyset$.

Step n: r_{i_n} partitions $\mathfrak{L}'(r_{i_m}|r_{i_l})$ into $\mathfrak{L}(r_{i_n}|r_{i_m})$ and $\mathfrak{L}'(r_{i_n}|r_{i_m})$ such that $\mathfrak{L}'(r_{i_n}|r_{i_m}) = \emptyset$. Then, $\mathfrak{L}(r_{i_0}) \cup \mathfrak{L}(r_{i_1}|r_{i_0}) \cup \cdots \cup \mathfrak{L}(r_{i_m}|r_{i_l})$ and $\mathfrak{L}'(r_{i_m}|r_{i_l})$, hence $\mathfrak{L}(r_{i_n}|r_{i_m})$, form a partition of $\mathfrak{L}$. Therefore, $\psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0}) \wedge \cdots \wedge \psi(r_{i_m}|r_{i_l})$ and $\psi(r_{i_n}|r_{i_m})$ are both *disjoint* and *true*, thus $\alpha = \phi(r_{i_0}, r_{i_1}, \ldots, r_{i_m}, r_{i_n}) = \psi(r_{i_0}) \wedge \psi(r_{i_1}|r_{i_0}) \wedge \cdots \wedge \psi(r_{i_m}|r_{i_l}) \wedge \psi(r_{i_n}|r_{i_m}) = \mathbf{T}$.

Consequently, ϕ is composed of $\psi(.)$ *disjoint* and *satisfied*, thus $\models_\alpha \phi$, hence $b \Rightarrow c$ holds. Finally, we show $c \Rightarrow a$. `Scope` (r_i, ϕ) transforms $r_i \wedge \phi$ into $\psi(r_i) \wedge \phi'(r_i)$, thus $(r_i \wedge \phi) \equiv \big(\psi(r_i) \wedge \phi'(r_i)\big)$. Since ϕ and $\psi(r_i)$ are satisfied, $\phi'(r_i)$ is satisfied. Therefore, unsatisfiability of $\psi_s(r_i)$ for some s is necessary and sufficient for the unsatisfiability of $\phi_s(r_i)$ for any s. ◀

▶ Note. $\psi(r_i) \equiv \phi(r_i)$ for all $i \in \mathfrak{L}$. Also, $\bigwedge C_k$ such that $C_k = (r_i \odot r_j \odot r_v)$ transforms into $\bigwedge \mathcal{C}_i$ such that $\mathcal{C}_i = \big(\psi(x_i) \oplus \psi(\overline{x}_i)\big)$, thus $\bigwedge C_k \equiv \bigwedge \mathcal{C}_i$. Recall that $\phi = \bigwedge C_k$, where $\phi := \hat{\phi}$.

▶ Note. The assignment α construction is driven by partitioning the set $\mathfrak{L}'(.)$ such that $\mathfrak{L} \leftarrow \mathfrak{L} - \mathfrak{L}(r_{i_0})$ in Step 1, and $\mathfrak{L} \leftarrow \mathfrak{L} - \mathfrak{L}(r_{i_{n-1}}|r_{i_{n-2}})$ for $i_n \in \mathfrak{L}'(r_{i_{n-1}}|r_{i_{n-2}})$ in Step $n \geqslant 2$.

▶ Note (Construction of α). In order to form a partition over the set ϕ, α is constructed such that $\psi(r_{i_1}|r_{i_0}) = \psi(r_{i_1}) - \psi(r_{i_0})$, and $\psi(r_{i_n}|r_{i_{n-1}}) = \psi(r_n) - \big(\psi(r_{i_0}) \cup \cdots \cup \psi(r_{i_{n-1}}|r_{i_{n-2}})\big)$ for $n \geqslant 2$. On the other hand, if the construction involves no set partition, then $\alpha = \bigcup \psi(r_i)$ for $i = (i_0, i_1, \ldots, i_n)$, where $i_0 \in \mathfrak{L}$, $i_1 \in \mathfrak{L}'(r_{i_0}), \ldots, i_n \in \mathfrak{L}'(r_{i_m}|r_{i_l})$, thus $r_{i_0} \prec r_{i_1} \prec \cdots \prec r_{i_n}$. Note that there is no need to construct $\phi'(r_i)$ in `Scan`/`Scope` L:9 (cf. Algorithm 5).

For instance, if Example 40 involves no set partition, then $\alpha = \{\psi(\overline{x}_7), \psi(x_2), \psi(x_1)\}$, in which $\psi(\overline{x}_7) = \{\overline{x}_7, \overline{x}_6\}$, $\psi(x_2) = \{x_2\}$, and $\psi(x_1) = \{x_1, x_2, \overline{x}_7, \overline{x}_6\}$. Also, $\overline{x}_7 \prec x_2 \prec x_1$ due to $x_2 \in \phi'(\overline{x}_7)$ and $x_1 \in \phi'(x_2|\overline{x}_7)$. Moreover, $\psi(\overline{x}_7)$, $\psi(x_2|\overline{x}_7)$, and $\psi(x_1|x_2)$ form a partition over the set ϕ, where $\psi(x_2|\overline{x}_7) = \psi(x_2) - \psi(\overline{x}_7)$ and $\psi(x_1|x_2) = \psi(x_1) - \big(\psi(x_2|\overline{x}_7) \cup \psi(\overline{x}_7)\big)$. As a result, $\alpha = \phi(\overline{x}_7, x_2, x_1) = \{\overline{x}_7, \overline{x}_6\} \cup \{x_2\} \cup \{x_1\}$ such that $\{\overline{x}_7, \overline{x}_6\} \cap \{x_2\} \cap \{x_1\} = \emptyset$.