FernUniversität in Hagen
Faculty of Mathematics and Computer Science

Artificial Intelligence Group

An Investigation of a Notion of „Variable Forgetting“ that Minimizes Truth Values

Bachelor's Thesis

in partial fulfillment of the requirements for
the degree of Bachelor of Science (B.Sc.)
in Computer Science

submitted by
Christoph Kaplan

First examiner: Dr. Kai Sauerwald
Artificial Intelligence Group

Advisor: Dr. Kai Sauerwald
Artificial Intelligence Group

Statement

Ich erkläre, dass ich die Bachelorarbeit selbstständig und ohne unzulässige Inanspruchnahme Dritter verfasst habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht. Die Versicherung selbstständiger Arbeit gilt auch für enthaltene Zeichnungen, Skizzen oder graphische Darstellungen. Die Bachelorarbeit wurde bisher in gleicher oder ähnlicher Form weder derselben noch einer anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Mit der Abgabe der elektronischen Fassung der endgültigen Version der Bachelorarbeit nehme ich zur Kenntnis, dass diese mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate geprüft werden kann und ausschließlich für Prüfungszwecke gespeichert wird.

	Yes	No
I agree to have this thesis published in the library.	☑	☐
I agree to have this thesis published on the webpage of the artificial intelligence group.	☑	☐
The thesis text is available under a Creative Commons License (CC BY-SA 4.0).	☑	☐
The source code is available under a GNU General Public License (GPLv3).	☑	☐
The collected data is available under a Creative Commons License (CC BY-SA 4.0).	☑	☐

Berlin, 03.07.2024 C. Kaplan

. . . .

(Place, Date) (Signature)

Zusammenfassung

Diese Bachelorarbeit widmet sich dem Thema "Variable Forgetting" – einer Technik der Aussagenlogik und der Künstlichen Intelligenz. Aussagenlogik ist eine der grundlegendsten Logiken, die in der Informatik sowie darüber hinaus eine zentrale Rolle spielt. In der KI findet sie Anwendung in den Bereichen der Wissensrepräsentation, des Belief Revision sowie in Multiagentensystemen. Da Wissen einem ständigen Wandel unterliegt, ist eine kontinuierliche Revision und Verfeinerung der formalen Repräsentation von Wissen notwendig. Des Weiteren stellt die Reduktion wachsender Komplexität von logischen Sätzen oder Wissensbasen ein weiteres Problem dar. Variable Forgetting bietet einen Ansatz zur Reduktion dieser Komplexität durch Fokussierung oder Aktualisierung der Wissensdarstellung.

Das klassische Variable Forgetting in der Logik geht auf die Arbeit von George Boole aus dem Jahr 1854 zurück, mit bedeutenden Beiträgen von Lin und Reiter in ihrem wegweisenden Paper "Forget It!" von 1994. Diese Bachelorarbeit stellt einen "zurückhaltenden" Ansatz zum Variable Forgetting vor, genannt "Skeptical Variable Forgetting". Eine Besonderheit des Variable Forgetting im Allgemeinen ist, dass es in bestimmten Szenarien zu drastischen Ergebnissen wie Tautologien oder Kontradiktionen führen kann, was ein zentrales epistemisches Interesse dieser Arbeit darstellt. Wir untersuchen Skeptical Variable Forgetting im Vergleich zum klassischen Variable Forgetting und heben dabei Eigenschaften und Unterschiede hervor. Die Studie beginnt mit einer syntaktischen Analyse und geht dann zu einer semantischen Untersuchung und Charakterisierung des Skeptical Variable Forgetting über. Darüber hinaus werden verwandte Themen kurz angeschnitten, um ein holistischeres Bild zu vermitteln.

Abstract

This bachelor thesis explores "Variable Forgetting" – a technique in propositional logic and artificial intelligence. Propositional logic is one of the fundamental logics in computer science and beyond. In AI, it finds applications in Knowledge Representation and Reasoning, Belief Revision and Multi-Agent Systems. Since knowledge is constantly changing, continuous revision and refinement of its formal representation are necessary. Additionally, reducing the growing complexity of logical sentences or knowledge bases presents another challenge. Variable Forgetting offers an approach to reduce this complexity by focusing or revising the representation of knowledge.

Classical Variable Forgetting in logic traces back to George Boole's work in 1854, with significant contributions from Lin and Reiter in their seminal paper "Forget It!" in 1994. This thesis introduces a cautious approach to "Variable Forgetting", termed "Skeptical Variable Forgetting". An intriguing aspect of Variable Forgetting is its potential to lead to drastic outcomes such as tautologies or contradictions in specific scenarios, which is a central epistemic interest of this work. We examine Skeptical Variable Forgetting compared to classical Variable Forgetting, highlighting their properties and distinctions. The study begins with a syntactic analysis and proceeds to a semantic examination and characterization of Skeptical Variable Forgetting. Additionally, related topics are briefly addressed to provide a more holistic understanding.

Contents

1 Introduction

Whether considering Knowledge Representation and Reasoning (KRR), Belief Revision or Multi-Agent Systems (MAS), many fields in artificial intelligence depend critically on modeling and representation of knowledge. Hence, it is nearby that an artificial agent should be capable not only of acquiring knowledge but also of eliminating or forgetting knowledge. An important aspect is recognizing that knowledge is subject to change. Consider a simple example: an agent models the world through a knowledge base. If the material world "out there" changes—say a statue is removed from the town square—the knowledge about this statue's presence should update correspondingly. This requires the agent to revise it's knowledge base, potentially eliminating outdated statements.

Not only does the external world undergo change; the very notions we use to apprehend and articulate also evolve. Whether this distinction can be made or not, pure descriptions or facts are crucial, but normative views and values also play a role in modeling thought and are subject to change as well. We suppose that knowledge undergoes transformation and is shaped not only by the object but also by the subject, by the observer. What is considered valid knowledge evolves with time, history, and societal context. Facts are compelling but not "dogmatic" and can be refuted or altered by scientific progress. Reflection on knowledge also leads us to the notion that it revolves around the determination of statements as "true" or "false". Yet, a frequently neglected dimension is that of scope or focus—what is included, why it is included, and what is excluded. This reveals that knowledge is not solely concerned with factual accuracy but also with the reasons for relevance, attention, and exclusion—what matters and what does not. Overlooking certain connections can alter the whole picture. Those familiar with "whodunit" movies understand how adding new information creates twists, and the same applies to removing information. A similar aspect is that knowledge might be irrelevant in certain contexts and therefore unnecessary. Attribution of relevance shapes knowledge, and what is deemed irrelevant is or can be consequently forgotten. There are many different approaches to understanding the notion of forgetting, some of which are discussed in [VDHLM09]. We will approach forgetting from the perspective of relevance.

In the technical landscapes of AI, we confront material challenges—complex computations for querying knowledge bases and deducing conclusions demand substantial computational power. Filtering out irrelevant knowledge also simplifies matters, offering potential technical benefits across diverse AI domains.

In Knowledge Representation and Reasoning (KRR), propositional logic serves as the formalism for modeling knowledge. To this end there are a few approaches to forgetting in logic yet it is not a well researched topic. The concept of forgetting itself can be interpreted in various ways, with differing opinions on its definition and operational mechanics. This paper focuses specifically on variable forgetting within propositional logic. George Boole, in his work "The Laws of Thought" [Boo21], introduced the idea of "elimination of the middle terms", wich can be seen as an early version of variable forget in the field. The $Forget$ operation, which we aim to build upon, was introduced in Lin and Reiter's 1994 seminal paper titled "Forget It!" [LR94]. Their approach involves the susbtitution of a variable by logical constants. This notion is semantically defined by establishing two interpretations that agree on all aspects except possibly the truth value of the variable being forgotten, thereby deeming it irrelevant.

In this thesis, we introduce a conjunction-based variation of the forgetting operation, termed $SkepForget$, which we intend to explore in depth. We will begin by investigating the syntactic aspects, comparatively analyzing the behavior of $SkepForget$. Next, we will examine the semantic patterns and implications to describe the general properties of this operation. Finally, we will discuss related topics to highlight the connecting research in the field.

1.1 Preliminaries

Before delving into the topic of forgetting, let's establish some fundamental concepts in propositional logic.

We denote a signature Σ as a set of propositional atomic variables. Let $\mathcal{L}_\Sigma$ denote a propositional language over Σ. Atomic variables are signified by small Latin letters; for example, $\Sigma = \{a, b, c\}$. Formulae (or sentences) in $\mathcal{L}_\Sigma$ are denoted by capital Latin letters (e.g., F, G, H) and are inductively defined using common logical connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) as well as logical constants $\top$ for true and $\bot$ for false. Every propositional variable $a \in \Sigma$ is a formula F. If F is a formula, then so is $\neg F$. If F and G are formulae, then so are $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, and $(F \leftrightarrow G)$. For instance, $F = (\neg(a \wedge b) \vee (c \rightarrow d))$ is a formula. We refer to a literal as an atomic variable or its negation for instance a and $\neg a$ are both a-literals.

Definition 1 (Truth Values). *The set of truth values, denoted as $BOOL$, is defined as $\{0, 1\}$, where 0 represents false and 1 represents true.*

Definition 2 (Σ-Interpretation). *Let Σ be a signature, F be a formula over the language $\mathcal{L}_\Sigma$. Let $\mathcal{A}_\Sigma \subseteq \mathcal{L}_\Sigma$ denote all the atomic formulae in $\mathcal{L}_\Sigma$. We define the function $\omega_a : \mathcal{A}_\Sigma \rightarrow BOOL$ as Σ-Interpretation. We extend ω_a to $\omega : \mathcal{L}_\Sigma \rightarrow BOOL$.*

- *For every atomic formula $A \in \mathcal{A}_\Sigma$, $\omega(A) = \omega_a(A)$.*
- $\omega(\neg F) = \begin{cases} 1, & \text{if } \omega(F) = 0 \\ 0, & \text{otherwise} \end{cases}$
- $\omega(F = (G \wedge H)) = \begin{cases} 1, & \text{if } \omega(G) = 1 \text{ and } \omega(H) = 1 \\ 0, & \text{otherwise} \end{cases}$
- $\omega(F = (G \vee H)) = \begin{cases} 1, & \text{if } \omega(G) = 1 \text{ or } \omega(H) = 1 \\ 0, & \text{otherwise} \end{cases}$

We will refer to Σ-Interpretation just by saying "interpretation" or "possible world". As for semantic notions we denote Ω_Σ as the set of all interpretations under Σ.

Definition 3 (Semantic Consequence). *Let Σ be a signature, F be a formula and $\omega \in \Omega_\Sigma$. If $\omega(F) = 1$ then the interpretation ω is a model of F, denoted by $\omega \models_\Sigma F$. We say ω satisfies or models F.*

We denote $Mod_\Sigma(F) = \{\omega \mid \omega \models_\Sigma F\}$ as the subset of Ω_Σ in which F holds true such that $\omega(F) = 1$ where $\omega \in \Omega_\Sigma$. Semantic equivalence under Σ is denoted as $\equiv_\Sigma$. For instance, let $\Sigma = \{a, b\}$, $A \in \mathcal{L}_\Sigma$ and $A = a \wedge b$ then $A \equiv_\Sigma \neg\neg A$ or $A \vee \neg A \equiv_\Sigma \top$ holds true.

Let $\omega \in \Omega_\Sigma$, a mapping $\omega[a \mapsto 1]$ denotes the interpretation $\omega' \in \Omega_\Sigma$ such that $\omega'(b) = \omega(b)$ for all atoms b that are not equal to a, and $\omega'(a) = 1$. Similarly, $\omega[a \mapsto 0]$ denotes an interpretation ω' with $\omega'(a) = 0$. To express assignments of an interpretation ω, we will use a notation such that $\bar{a}bc$ means $\omega(a) = 0$, $\omega(b) = 1$, and $\omega(c) = 1$, and $a\bar{b}\bar{c}$ means $\omega(a) = 1$, $\omega(b) = 0$, and $\omega(c) = 0$.

For $\omega \in \Omega_\Sigma$ and $a \in \Sigma$ we denote $Switch(\omega, a)$ as the interpretation in Ω_Σ that maintains the same truth values of ω for all variables except a, but assigns the opposite truth value to a compared to ω. For a subsignature $\Gamma \subseteq \Sigma$ and an interpretation $\omega \in \Omega_\Sigma$, we denote the Γ-part of ω as $\omega^\Gamma \in \Omega_\Gamma$, mentioning exactly the atoms from Γ. That is, $\omega^\Gamma : \Gamma \to BOOL$ with $\omega^\Gamma(b) = \omega(b)$ for all $b \in \Gamma$.

For each formula F, there exists an equivalent formula in conjunctive normal form, denoted as $F_{\text{cnf}} \in \mathcal{L}_\Sigma$. Let $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m_i\}$. A formula in CNF can be represented as:

$$F_{\text{cnf}} = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \mathrm{l}_{i,j} \right),$$

where each $C_i = \bigvee_{j=1}^{m_i} \mathrm{l}_{i,j}$ denotes the i-th disjunction of literals, referred to as a "clause".

Here, $\mathrm{l}_{i,j}$ represents the j-th literal in the i-th clause, and m_i is the number of literals in the clause C_i.

We represent F_{cnf} as a set of clauses, known as a "clause-set". Let $\{C_1, C_2, \ldots, C_n\}$ denote the clause-set of F_{cnf}. Thus, F_{cnf} can be notated using clause sets as:

$$F_{\text{cnf}} = \{\{l_{1,1}, l_{1,2}, \ldots, l_{1,m_1}\}, \{l_{2,1}, l_{2,2}, \ldots, l_{2,m_2}\}, \ldots, \{l_{n,1}, l_{n,2}, \ldots, l_{n,m_n}\}\}.$$

A multiset is an extension of a set, denoted as $\{\{a, a, b, c\}\}$. While a set can contain only one occurrence of any given element, a multiset may contain multiple occurrences of the same element.

We now provide some definitions related to sets, specifying the appearing variables and indicating whether an atom appears positively or negatively (as a negated atom) in a formula.

Definition 4 (Sig). *For a formula $F \in \mathcal{L}_\Sigma$ and a variable $a \in \Sigma$, we denote $Sig(F)$ as the set of atomic variables that appear in F given inductively by:*

- $Sig(a) = \{a\}$
- $Sig(\neg F) = Sig(F)$
- $Sig(F = G \vee H) = Sig(G) \cup Sig(H)$
- $Sig(F = G \wedge H) = Sig(G) \cap Sig(H)$

Example 5. *Let $F = \neg((a \wedge b) \vee \neg c)$ in $\mathcal{L}_\Sigma$.*

$$\begin{aligned} Sig(F) &= Sig((a \wedge b) \vee \neg c) \\ &= Sig(a \wedge b) \cup Sig(\neg c) \\ &= Sig(a) \cup Sig(b) \cup Sig(c) \\ &= \{a\} \cup \{b\} \cup \{c\} \\ &= \{a, b, c\} \end{aligned}$$

Definition 6 (PosAtom and NegAtom). *Let $F \in \mathcal{L}_\Sigma$ be a formula, and let $a \in \Sigma$ be an atomic variable. Then $PosAtom(F, a)$ denotes the set of positive atomic variables of the sentence F, given inductively by:*

- $PosAtom(a) = \{a\}$
- $PosAtom(\neg F) = NegAtom(F)$
- $PosAtom(F = G \vee H) = PosAtom(G) \cup PosAtom(H)$
- $PosAtom(F = G \wedge H) = PosAtom(G) \cap PosAtom(H)$

And $NegAtom(F, a)$ denotes the set of negative atomic variables of the sentence F, given inductively by:

- $NegAtom(a) = \emptyset$
- $NegAtom(\neg F) = PosAtom(F)$
- $NegAtom(F = G \vee H) = NegAtom(G) \cup NegAtom(H)$
- $NegAtom(F = G \wedge H) = NegAtom(G) \cap NegAtom(H)$

Example 7. *Let $F = \neg((a \wedge \neg b) \vee \neg c)$ in $\mathcal{L}_\Sigma$.*

$$\begin{aligned} PosAtom(F) &= NegAtom((a \wedge \neg b) \vee \neg c) \\ &= NegAtom(a \wedge \neg b) \cup NegAtom(\neg c) \\ &= NegAtom(a) \cup NegAtom(\neg b) \cup NegAtom(\neg c) \\ &= \emptyset \cup PosAtom(b) \cup PosAtom(c) \\ &= \{b, c\} \end{aligned}$$

$$\begin{aligned} NegAtom(F) &= PosAtom((a \land \neg b) \lor \neg c) \\ &= PosAtom(a \land \neg b) \cup PosAtom(\neg c) \\ &= PosAtom(a) \cup PosAtom(\neg b) \cup PosAtom(\neg c) \\ &= \{a\} \cup \emptyset \cup \emptyset \\ &= \{a\} \end{aligned}$$

Finally, we provide a definition for propositional variable substitution as follows:

Definition 8 (Variable Substitution). *Let F and B be formulae over $\mathcal{L}_\Sigma$ where, and let $a \in \Sigma$ be an atomic variable. Then $F[a/B]$ denotes the substitution of every a in F by B, given inductively by:*

- *If $F = a$ then $F[a/B] = B$*
- *If $F = b$ and $a \neq b$ then $F[a/B] = F$*
- *If $F = (\neg F)$ then $F[a/B] = \neg(F[a/B])$*
- *If $F = (G \land H)$ then $F[a/B] = G[a/B] \land H[a/B]$*
- *If $F = (G \lor H)$ then $F[a/B] = G[a/B] \lor H[a/B]$*

Example 9. *Let $F = a \lor b$ a formula over $\mathcal{L}_\Sigma$ and $a, b \in \Sigma$ then:*

$$\begin{aligned} F[a/\top] &= a[a/\top] \lor b[a/\top] \\ &= \top \lor b \end{aligned}$$

$$\begin{aligned} F[a/\bot] &= a[a/\bot] \lor b[a/\bot] \\ &= \bot \lor b \end{aligned}$$

1.2 Classical vs Skeptical Forget

For the notion of the classical operation, we refer to variable forgetting as described in [LR94] and [LLM03]. We will now present both definitions for classical and skeptical forgetting. Through examples, we aim to provide an initial impression of the behavior and challenges associated with both operations. We define (classical) variable forgetting inductively:

Definition 10 (Variable Forgetting). *Let $F \in \mathcal{L}_\Sigma$ be a formula, let $a \in \Sigma$ an atomic variable and $A \subseteq \Sigma$ a set of variables. The* forgetting of a in F *$Forget(F, a)$, respectively, the* forgetting of A in F *$Forget(F, A)$, is given inductively by:*

- $Forget(F, \emptyset) = F$
- $Forget(F, \{a\}) = F[a/\top] \lor F[a/\bot]$

- $\mathit{Forget}(F, a) = \mathit{Forget}(F, \{a\})$
- $\mathit{Forget}(F, A \cup \{a\}) = \mathit{Forget}(\mathit{Forget}(F, \{a\}), A)$

For some formulae F, the process of variable forgetting can yield significant reductions or even drastic outcomes. To illustrate, let's begin by examining an instance where the operation $Forget$ behaves quite appropriate. Let $F_1 = a \wedge b$ and we choose to forget about b:

$$\begin{aligned} Forget(F_1, b) &= F_1[b/\top] \vee F_1[b/\bot] \\ &\equiv_\Sigma ((a \wedge \top) \vee (a \wedge \bot)) \\ &\equiv_\Sigma (a \vee \bot) \\ &\equiv_\Sigma a \end{aligned}$$

This demonstrates that $\mathit{Forget}(F_1, b)$ simplifies to a, which intuitively aligns with the expected outcome when forgetting about b in F_1. Now, let's explore the forgetting of b in the formula $F_2 = a \vee b$.

$$\begin{aligned} Forget(F_2, b) &= F_2[b/\top] \vee F_2[b/\bot] \\ &\equiv_\Sigma ((a \vee \top) \vee (a \vee \bot)) \\ &\equiv_\Sigma (\top \vee a) \\ &\equiv_\Sigma \top \end{aligned}$$

We observe that $\mathit{Forget}(F_2, b)$ results in a *tautology*, which is a drastic consequence when singularly forgetting b. A variant of the forget operation can be defined as follows:

Definition 11 (Skeptical Variable Forgetting). *Let $F \in \mathcal{L}_\Sigma$ be a formula , $a \in \Sigma$ an atomic variable and $A \subseteq \Sigma$ a set of variables. The* skeptical forgetting of a in F *$\mathit{SkepForget}(F, a)$, respectively, the skeptical forgetting of A in F $\mathit{SkepForget}(F, A)$, is given inductively by:*

- $\mathit{SkepForget}(F, \emptyset) = F$
- $\mathit{SkepForget}(F, \{a\}) = F[a/\top] \wedge F[a/\bot]$
- $\mathit{SkepForget}(F, a) = \mathit{SkepForget}(F, \{a\})$
- $\mathit{SkepForget}(F, A \cup \{a\}) = \mathit{SkepForget}(\mathit{SkepForget}(F, \{a\}), A)$

The proposal is to utilize conjunction instead of disjunction. This adjustment leads to the following outcomes for our earlier examples:

$$\begin{aligned} SkepForget(F_1, b) &= F_1[b/\top] \vee F_1[b/\bot] \\ &\equiv_\Sigma ((a \wedge \top) \wedge (a \wedge \bot)) \\ &\equiv_\Sigma (a \wedge \bot) \\ &\equiv_\Sigma \bot \end{aligned}$$

$$
\begin{aligned}
SkepForget(F_2, b) &= F_2[b/\top] \vee F_2[b/\bot] \\
&\equiv_\Sigma ((a \vee \top) \wedge (a \vee \bot)) \\
&\equiv_\Sigma (\top \wedge a) \\
&\equiv_\Sigma a
\end{aligned}
$$

Upon observation, our alternative operation does not necessarily yield less drastic results, at least in these examples. In the first case, we encounter a contradiction, while in the second case, we arrive at an equivalent to a. This suggests a duality between the two operations.

2 Syntactic Investigation and Characterization

The introductory examples in the previous section have illuminated the fundamental problem, in which variable forgetting yields drastic results such as $\top$ or $\bot$ in some cases. When simulating the "phenomena of forgetting", one might naively anticipate the removal of a part of the statement while preserving the remainder. Yet, as illustrated by the preceding examples, success is not always guaranteed. This observation directs our epistemic interest in the first place. Why do these operations yield in tautological or contradictory outcomes? How do the top-layer connectives in the syntax tree influence the results? What impact do deeply nested expressions exert on the final outcome? Eventually, we will explore the semantic aspects to understand their properties. For now, however, we will begin our investigation with a syntactical approach, systematically examining the outcomes from various sentences. To derive meaningful conclusions, it is essential to select examples that facilitate generalization while maintaining specificity to preserve key properties and characteristics. How can we identify such examples? What criteria are essential for their selection? To understand an object and identify its properties, it can be helpful to compare it with other entities, as this can highlight differences. For example, comparing $SkepForget$ with a square root operation might not make sense, since a common denominator would be to far, to abstract. Fortunately, we have a very similar operation already at hand that is suitable for comparison. In our case, we want to compare our two forgetting operations to see if they behave for example dual to each other. To explore the properties of our $SkepForget$ operation, it is crucial to understand the conditions under which particular results arise. Given that propositional formulae can become very complex, and we are initially proceeding empirically, we want to select examples that are reasonably representative. We have identified several dimensions to consider. First, we need to test our operations with various unary and binary connectives. Additionally, it is important to consider which variable is being forgotten. For instance, with a sentence like $F = (a \vee b \vee c)$, there are three options for forgetting variables. We should also consider the position of the variable in the syntax tree, as this may play a significant role. Let's begin our investigation with the following sentences:

$$\begin{array}{llllll} F_1 & = a & Forget(F_1, a) & \equiv_\Sigma \top & SkepForget(F_1, a) & \equiv_\Sigma \bot \\ F_2 & = \neg a & Forget(F_2, a) & \equiv_\Sigma \top & SkepForget(F_2, a) & \equiv_\Sigma \bot \end{array}$$

In F_1 and its negation F_2, we loose or forget the entire sentence itself. The result is $\top$ for $Forget$ and $\bot$ for $SkepForget$, suggesting a fundamental characteristic of each operation.

We want to briefly emphasize an interesting distinction between negation and irrelevance (or affirmation and relevance). In our context, forgetting a variable doesn't merely erase its truth values; it erases the ability to even refer to the variable itself.

Without a notion or signifier for it, we cannot assert its falsity. For example we can not ask the question wether a is true or is false within a possible world. In the case of $SkepForget(F_1, a)$, the variable a effectively ceases to exist; our statement $SkepForget(F_1, a)$ no longer addresses a at all. We will later see that this may not hold true depending on the signature under which we observe the formula, but we find it a noteworthy point to consider.

Before we delve into more complex sentences, let's examine a special edge case:

$$\begin{aligned} F_1 &= (a \wedge \neg a) \equiv_\Sigma \bot & Forget(F_1, a) &\equiv_\Sigma \bot & SkepForget(F_1, a) &\equiv_\Sigma \bot \\ F_2 &= (a \vee \neg a) \equiv_\Sigma \top & Forget(F_2, a) &\equiv_\Sigma \top & SkepForget(F_2, a) &\equiv_\Sigma \top \end{aligned}$$

In this case, we observe a drastic result, as discussed, primarily because the original formula is already a tautology or a contradiction. We also note that F_1, $Forget(F_1, a)$ and $SkepForget(F_1, a)$ are equivalent. The same holds for F_2, $Forget(F_2, a)$ and $SkepForget(F_2, a)$.

Now, let's investigate the behavior with more complex sentences that include other variables besides a.

$$\begin{aligned} F_1 &= (a \wedge b) & Forget(F_1, a) &\equiv_\Sigma b & SkepForget(F_1, a) &\equiv_\Sigma \bot \\ F_2 &= \neg(a \wedge b) & Forget(F_2, a) &\equiv_\Sigma \top & SkepForget(F_2, a) &\equiv_\Sigma \neg b \\ F_3 &= (a \vee b) & Forget(F_3, a) &\equiv_\Sigma \top & SkepForget(F_3, a) &\equiv_\Sigma b \\ F_4 &= \neg(a \vee b) & Forget(F_4, a) &\equiv_\Sigma \neg b & SkepForget(F_4, a) &\equiv_\Sigma \bot \end{aligned}$$

We observe that $Forget(F_1, a) \equiv_\Sigma \neg(SkepForget(F_2, a))$. Given $F_1 \equiv_\Sigma \neg F_2$, we derive $Forget(F_1, a) \equiv \neg SkepForget(\neg F_1, a)$. Similarly, $\neg F_1 \equiv_\Sigma F_2$ leads to $Forget(\neg F_2, a) \equiv_\Sigma \neg SkepForget(F_2, a)$. More general, we can establish the equivalence:

Proposition 12 (De Morgan Relation). *For any formula $F \in \mathcal{L}_\Sigma$ and variable $a \in \Sigma$, we have:*

$$\neg Forget(F, a) \equiv_\Sigma SkepForget(\neg F, a)$$
$$Forget(\neg F, a) \equiv_\Sigma \neg SkepForget(F, a)$$

Proof.

$$\begin{aligned} \neg Forget(F, a) &= \neg(F[a/\top] \vee F[a/\bot]) \\ &\equiv_\Sigma \neg F[a/\top] \wedge \neg F[a/\bot] \\ &\equiv_\Sigma SkepForget(\neg F, a) \end{aligned}$$

$$\begin{aligned} Forget(\neg F, a) &= \neg F[a/\top] \vee \neg F[a/\bot] \\ &\equiv_\Sigma \neg(F[a/\top] \wedge F[a/\bot]) \\ &\equiv_\Sigma \neg SkepForget(F, a) \end{aligned}$$

□

We have now established our first property. However, when it comes to regular negation, $SkepForget$ (as well as $Forget$) does not hold true as we can demonstrate:

Proposition 13 (Negation)**.** *There is a formula $F = \neg A$ over $\mathcal{L}_\Sigma$ and variable $a \in \Sigma$, the following holds:*

$$SkepForget(F, a) \not\equiv_\Sigma \neg SkepForget(A, a).$$

Proof. Let $F \in \mathcal{L}_\Sigma$ and $a \in \Sigma$. There is a formula $F = \neg A$ with $A = a$ such that

$$SkepForget(\neg A, a) \equiv_\Sigma \bot \text{ and } \neg SkepForget(A, a) \equiv_\Sigma \neg\bot.$$

Thus,

$$SkepForget(\neg A, a) \not\equiv_\Sigma SkepForget(A, a). \quad \square$$

Note that consequently the following **does not** hold for all formulae:

$$SkepForget(F, a) \equiv_\Sigma \neg SkepForget(A, a).$$

Let's continue the investigation with a focus on binary connectives:

$$\begin{array}{llllll} F_1 & = (a \wedge b) & Forget(F_1, a) & \equiv_\Sigma b & SkepForget(F_1, a) & \equiv_\Sigma \bot \\ F_2 & = (a \vee b) & Forget(F_2, a) & \equiv_\Sigma \top & SkepForget(F_2, a) & \equiv_\Sigma b \end{array}$$

In F_1 and F_2, we selectively forget one variable of a binary connective. Given the commutativity of logical connectives, forgetting the other variable does not introduce new information. Consequently, both forget operations result in either $\top$ or $\bot$, or the retention of the variable b. This comparison reveals a dualistic behavior between the two operations. A necessary condition appears to be the connective itself. We notice that $Forget$ with $\wedge$ yields $\bot$, whereas in combination with $\vee$, we obtain b. For $SkepForget$, it's the opposite, including the constant being $\top$, which is the inverse. In summary, in two cases, we lose exactly the variable we expect to lose. In the other two cases, we lose more information than expected.

If we compare the outcomes with the original sentences, we observe that from F_1, we can derive b, whereas this is not necessarily true for F_2. Conversely, from b, we can derive F_2. Furthermore, it's noteworthy that $\top$ can always be derived from any sentence because we can disjunct its negation, and from $\bot$, any sentence can be derived. Once again, we encounter a dualistic behavior, this time concerning the direction of entailment—both from the original sentence to the forgotten sentence and vice versa. We will also discuss this later in section 4.4.

Let's explore whether this pattern holds true with more complex or longer sentences:

$$\begin{array}{llllll} F_1 & = (a \wedge (c \wedge d)) & Forget(F_1, a) & \equiv_\Sigma (c \wedge d) & SkepForget(F_1, a) & \equiv_\Sigma \bot \\ F_2 & = (a \wedge (c \vee d)) & Forget(F_2, a) & \equiv_\Sigma (c \vee d) & SkepForget(F_2, a) & \equiv_\Sigma \bot \\ F_3 & = (a \vee (c \wedge d)) & Forget(F_3, a) & \equiv_\Sigma \top & SkepForget(F_3, a) & \equiv_\Sigma (c \wedge d) \\ F_4 & = (a \vee (c \vee d)) & Forget(F_4, a) & \equiv_\Sigma \top & SkepForget(F_4, a) & \equiv_\Sigma (c \vee d) \end{array}$$

When we introduce additional complexity by adding one more connective, it appears that due to the inductive or recursive structure of logic formulae, we still observe the same dualistic result as in the previous example. In half of the cases, we lose exactly what we expect, while in the other half, we lose more information.

Let's examine if this pattern persists when we introduce one more atomic variable.

$$\begin{array}{llllll}
F_1 & = ((a \wedge b) \wedge (c \wedge d)) & Forget(F_1, a) & \equiv_\Sigma (b \wedge (c \wedge d)) & SkepForget(F_1, a) & \equiv_\Sigma \bot \\
F_2 & = ((a \wedge b) \wedge (c \vee d)) & Forget(F_2, a) & \equiv_\Sigma (b \wedge (c \vee d)) & SkepForget(F_2, a) & \equiv_\Sigma \bot \\
F_3 & = ((a \wedge b) \vee (c \wedge d)) & Forget(F_3, a) & \equiv_\Sigma (b \vee (c \wedge d)) & SkepForget(F_3, a) & \equiv_\Sigma (c \wedge d) \\
F_4 & = ((a \wedge b) \vee (c \vee d)) & Forget(F_4, a) & \equiv_\Sigma (b \vee (c \vee d)) & SkepForget(F_4, a) & \equiv_\Sigma (c \vee d) \\
F_5 & = ((a \vee b) \wedge (c \wedge d)) & Forget(F_5, a) & \equiv_\Sigma (c \wedge d) & SkepForget(F_5, a) & \equiv_\Sigma (b \wedge (c \wedge d)) \\
F_6 & = ((a \vee b) \wedge (c \vee d)) & Forget(F_6, a) & \equiv_\Sigma (c \vee d) & SkepForget(F_6, a) & \equiv_\Sigma (b \wedge (c \vee d)) \\
F_7 & = ((a \vee b) \vee (c \wedge d)) & Forget(F_7, a) & \equiv_\Sigma \top & SkepForget(F_7, a) & \equiv_\Sigma (b \vee (c \wedge d)) \\
F_8 & = ((a \vee b) \vee (c \vee d)) & Forget(F_8, a) & \equiv_\Sigma \top & SkepForget(F_8, a) & \equiv_\Sigma (b \vee (c \vee d))
\end{array}$$

Once again, we can observe a split, this time between F_4 and F_5. However, on each side of the group of drastic results, we notice that there are now partially less drastic outcomes. These can again be categorized into halves. It appears that the recurring pattern is related to the recursive structure of the syntax tree. Additionally, it seems that the connective that connects the forgotten variable, as well as the subsequent connectives, have a significant impact on the outcome.

Now, let's examine forgetting different variables within the same sentences. For a sentence $a \wedge (b \vee c)$, we have three options, but only two layers in depth. Considering the commutativity of logical connectives, we can reduce this to two options.

$$\begin{array}{llllll}
F_1 & = (a \wedge (b \wedge c)) & Forget(F_1, a) & \equiv_\Sigma (b \wedge c) & SkepForget(F_1, a) & \equiv_\Sigma \bot \\
F_2 & = (a \wedge (b \vee c)) & Forget(F_2, a) & \equiv_\Sigma (b \vee c) & SkepForget(F_2, a) & \equiv_\Sigma \bot \\
F_3 & = (a \vee (b \wedge c)) & Forget(F_3, a) & \equiv_\Sigma \top & SkepForget(F_3, a) & \equiv_\Sigma (b \wedge c) \\
F_4 & = (a \vee (b \vee c)) & Forget(F_4, a) & \equiv_\Sigma \top & SkepForget(F_4, a) & \equiv_\Sigma (b \vee c)
\end{array}$$

$$\begin{array}{llllll}
F_1 & = (a \wedge (b \wedge c)) & Forget(F_1, c) & \equiv_\Sigma (a \wedge b) & SkepForget(F_1, c) & \equiv_\Sigma \bot \\
F_2 & = (a \wedge (b \vee c)) & Forget(F_2, c) & \equiv_\Sigma a & SkepForget(F_2, c) & \equiv_\Sigma (a \wedge b) \\
F_3 & = (a \vee (b \wedge c)) & Forget(F_3, c) & \equiv_\Sigma (a \vee b) & SkepForget(F_3, c) & \equiv_\Sigma a \\
F_4 & = (a \vee (b \vee c)) & Forget(F_4, c) & \equiv_\Sigma \top & SkepForget(F_4, c) & \equiv_\Sigma (a \vee b)
\end{array}$$

Once again, in half of the cases, we lose exactly what we expect to lose. Moreover, the outcomes exhibit further distinctions: the variable a, positioned in the top layer, yields the most significant consequences, whereas the deeper nested variable c results in less drastic information loss.

Thus far, our examination has been limited to formulae composed solely of atoms. Now, we aim to broaden our scope to include more general sentences. Specifically, we want to determine if $SkepForget$ is compatible under conjunction and disjunction, meaning we need to investigate whether we can "pull" out logical connectives from our $SkepForget$ operator. As for negation, we have already demonstrated its incompatibility in Proposition 13.

Proposition 14 (Conjunction)**.** *Let $F = A \wedge B$ a formula over $\mathcal{L}_\Sigma$ and variable $a \in \Sigma$, it holds that:*

$$SkepForget(F, a) \equiv_\Sigma SkepForget(A, a) \wedge SkepForget(B, a).$$

Proof. Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$.

$$\begin{aligned} SkepForget(F, a) &= F[a/\top] \wedge F[a/\bot] \\ &\equiv_\Sigma (A[a/\top] \wedge B[a/\top]) \wedge (A[a/\bot] \wedge B[a/\bot]) \\ &\equiv_\Sigma A[a/\top] \wedge A[a/\bot] \wedge B[a/\top] \wedge B[a/\bot] \\ &\equiv_\Sigma SkepForget(A, a) \wedge SkepForget(B, a) \end{aligned}$$ □

Proposition 15 (Disjunction)**.** *There is a formula $F = A \vee B$ over $\mathcal{L}_\Sigma$ and a variable $a \in \Sigma$, for which the following holds:*

$$SkepForget(F, a) \not\equiv_\Sigma SkepForget(A, a) \vee SkepForget(B, a)$$

Note that the following **does not** hold for all formulae:

$$SkepForget(F, a) \equiv_\Sigma SkepForget(A, a) \vee SkepForget(B, a).$$

Proof. Let $F = (A \vee B) \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$. If $A = \neg a$ and $B = (a \vee b)$, then $SkepForget(F, a) \equiv_\Sigma \top$, $SkepForget(A, a) \equiv_\Sigma \bot$, and $SkepForget(B, a) \equiv_\Sigma b$. Consequently,

$$SkepForget(A, a) \vee SkepForget(B, a) \not\equiv_\Sigma SkepForget(F, a).$$ □

As Proposition 15 shows, conversely to $Forget$, $SkepForget$ is compatible under conjunction but incompatible under disjunction. By imposing certain constraints, we can formally state that:

Proposition 16 (Disjunction Constrained)**.** *Let $F \in \mathcal{L}_\Sigma$, a variable $a \in \Sigma$ where $F = A \vee B$ and where $a \notin Sig(B)$, the following holds:*

$$SkepForget(F, a) \equiv_\Sigma SkepForget(A, a) \vee B.$$

Proof. Let $F \in \mathcal{L}_\Sigma$, a variable $a \in \Sigma$ where $F = A \vee B$ and where $a \notin Sig(B)$. First observe that if $a \notin Sig(B)$ we have $B[a/\top] \equiv_\Sigma B[a/\bot] \equiv_\Sigma B$.

$$\begin{aligned} SkepForget(F, a) &= F[a/\top] \wedge F[a/\bot] \\ &\equiv_\Sigma (A[a/\bot] \vee B[a/\bot]) \wedge (A[a/\top] \vee B[a/\top]) \\ &\equiv_\Sigma (A[a/\bot] \vee B) \wedge (A[a/\top] \vee B) \\ &\equiv_\Sigma (A[a/\bot] \wedge A[a/\top]) \vee B \\ &\equiv_\Sigma SkepForget(A, a) \vee B \end{aligned}$$ □

2.1 Behavior of Truth Values in Syntax Trees

As hinted in the previous section, we suspect a correlation between the syntactic structure, the connectives, and the outcomes of the operations. Therefore, we aim to examine the syntax trees more closely to potentially gain further insights. Let us first investigate the relationship between syntax trees and the constants $\top$ and $\bot$ in general.

We know that $\top \wedge \bot \equiv \bot$, and $\top \vee \bot \equiv \top$. Syntax trees provide an excellent way to visualize sentences. To gain a comprehensive overview, we will explore all permutations of connectives over a given variable count n, which results in $2^{(n-1)}$ possibilities.

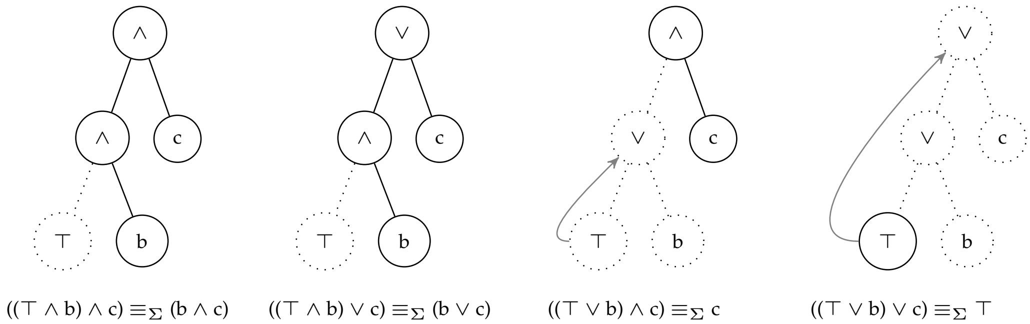


Figure 1: $\top$ Propagation

In figure 1 we can observe the bottomup propagation of $\top$. Starting from the leafes $\top$ effectively "absorbs" all siblings in conjunction with $\vee$ and stops propagating at $\wedge$.

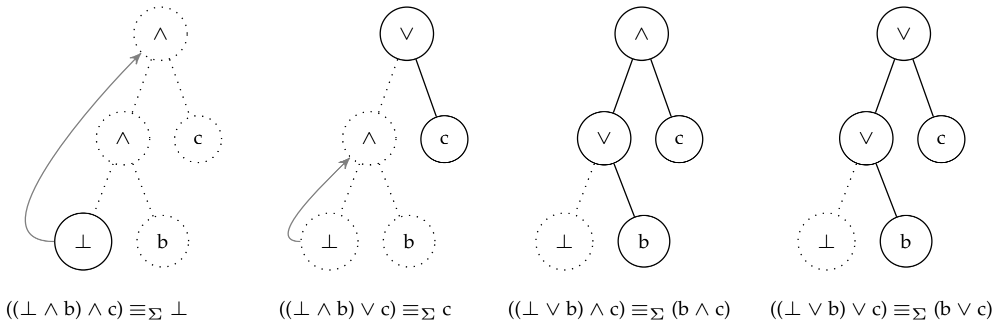
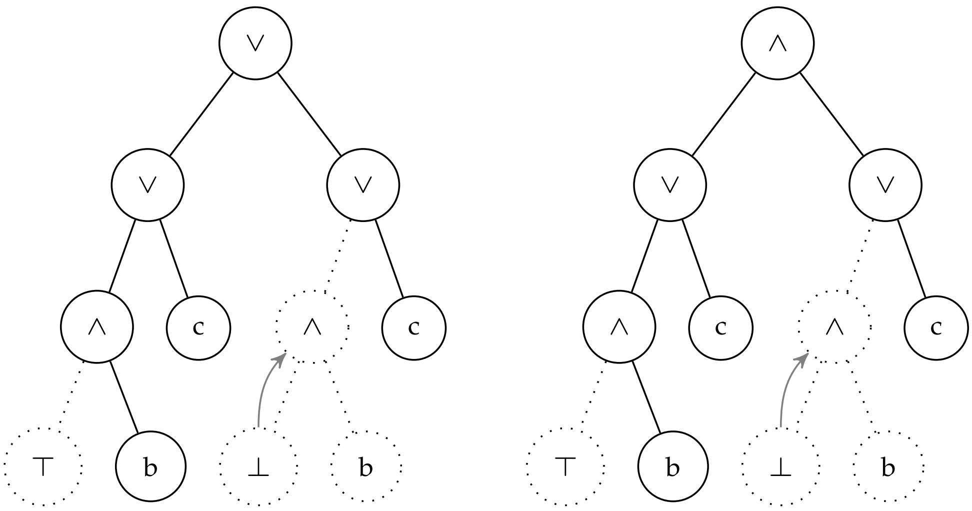


Figure 2: $\bot$ Propagation

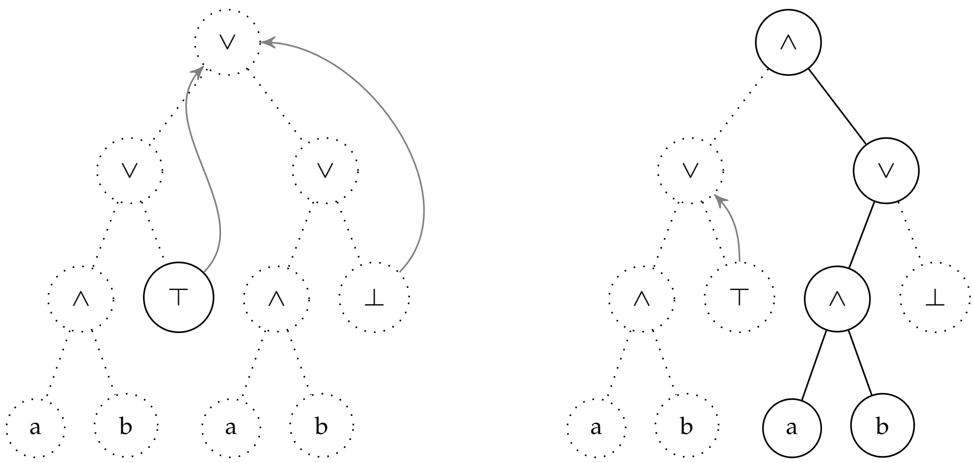
Figure 2 demonstrates, in a manner dual to Figure 1, how $\bot$ "annihilates" all siblings when paired with a logical $\wedge$ and halts at $\vee$. This dynamic interaction suggests that logical constants have the capability to selectively "eliminate" variables and entire propositions.

Let us now examine syntax trees for our forget operations over specific formulae. On the left, we display $Forget$, and on the right, $SkepForget$.



$Forget((a \wedge b) \vee c, a) \equiv_{\Sigma} (b \vee c)$ $SkepForget((a \wedge b) \vee c, a) \equiv_{\Sigma} c$

Figure 3: Forgetting a in $(a \wedge b) \vee c$



$Forget((a \wedge b) \vee c, c) \equiv_{\Sigma} \top$ $SkepForget((a \wedge b) \vee c, c) \equiv_{\Sigma} (a \wedge b)$

Figure 4: Forgetting c in $(a \wedge b) \vee c$

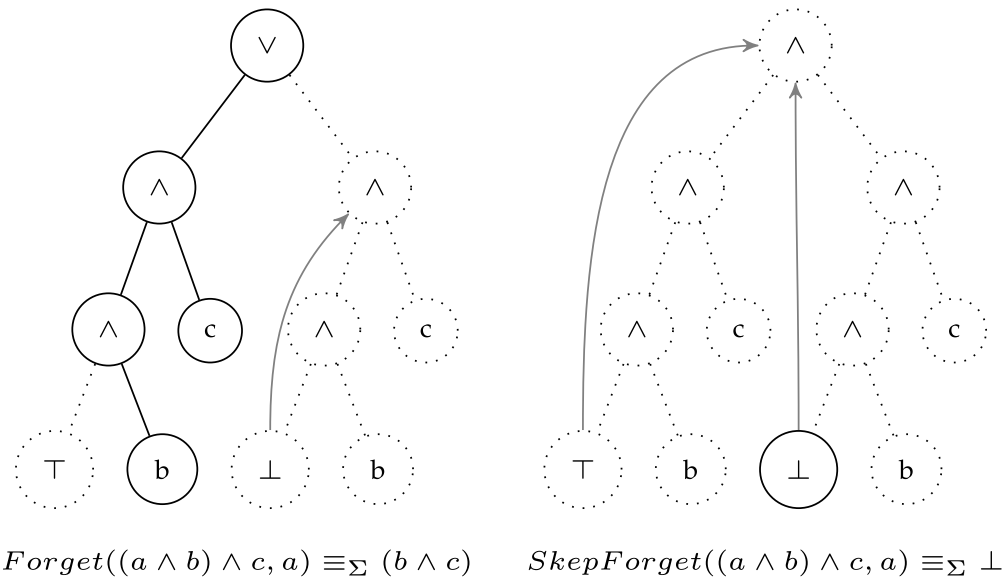


Figure 5: Forgetting a in $(a \wedge b) \wedge c$

We can observe that, due to the definition of the forget operations, we essentially have two branches. Ultimately, the truth value is determined at the topmost connective. $Forget$ allows a model if only one of the two branches is true, whereas $SkepForget$ requires both branches to be true. As we have already noted, $\top$ or $\bot$ are propagated upwards; the more complex or lengthy the formula, the more possibilities there are to "hold back" logical constants and retain information.

At this point we do not see any other way to predict whether and how much information is lost, except by examining the specific logical connectives. Extreme cases arise with $SkepForget$ when we only have conjunctions, while with $Forget$, it's disjunctions. Adding negation further complicates the matter. Let us expand our syntactic investigation and explore skeptical forgetting in the context of conjunctive normal form (CNF) in the following subsection.

2.2 Skeptical Forgetting and CNF

Let $F \in \mathcal{L}_\Sigma$. For all formulae F, there exists an equivalent formula in conjunctive normal form, denoted as $F_{\text{cnf}} \in \mathcal{L}_\Sigma$. For each clause C_i in F_{cnf}, where $i \in \{1, 2, \ldots, n\}$ (with n being the total number of clauses), and each literal $\text{lit}_{i,j}$ within C_i, where $j \in \{1, 2, \ldots, m_i\}$ (with m_i being the number of literals in the i-th clause), we denote $\text{lit}_{i,j}$ as a literal that does **not** contain the propositional variable a.

For instance:

$$F_{\text{cnf}} = a \wedge \neg a \wedge (\neg a \vee \text{lit}_{1,1} \vee \text{lit}_{1,2}) \wedge (a \vee \text{lit}_{i,j} \vee \text{lit}_{i,j+1}) \wedge \ldots \wedge (a \vee \neg a \vee \text{lit}_{n,m_n}).$$

For every clause in F_{cnf}, we can due to commutativity, group similar clauses. For clauses containing disjunctions of a, $\neg a$, or both, we can isolate ("pull out") a, $\neg a$, or $(a \vee \neg a)$ using the distributive law. Assuming we have all possible connections of a-literals with $\text{lit}_{i,j}$, we can categorize the clauses as follows:

- "a-Single" for a single a, respectively "¬a-Single" for a single $\neg a$.
- "a-Group", "¬a-Group", and "Both" for parts where $\text{lit}_{i,j}$ disjunctions appear with a or $\neg a$ or $(a \vee \neg a)$.
- "None" for clauses without any a-literals, only $\text{lit}_{i,j}$ disjunctions.

We split n into n_1, n_2, n_3, and n_4, representing the number of clauses in each category. Expressed as a CNF formula, we obtain:

$$\begin{aligned}F_{\text{cnf}} = \underbrace{a}_{\text{a-Single}} \wedge \underbrace{\neg a}_{\neg\text{a-Single}} \wedge \underbrace{\left(a \vee \left(\bigwedge_{i=1}^{n_1}\left(\bigvee_{j=1}^{m_i} \text{lit}_{i,j}\right)\right)\right)}_{\text{a-Group}} \\ \wedge \underbrace{\left(\neg a \vee \left(\bigwedge_{i=1}^{n_2}\left(\bigvee_{j=1}^{m_i} \text{lit}_{i,j}\right)\right)\right)}_{\neg\text{a-Group}} \\ \wedge \underbrace{\left((a \vee \neg a) \vee \left(\bigwedge_{i=1}^{n_3}\left(\bigvee_{j=1}^{m_i} \text{lit}_{i,j}\right)\right)\right)}_{\text{Both}} \\ \wedge \underbrace{\left(\bigwedge_{i=1}^{n_4}\left(\bigvee_{j=1}^{m_i} \text{lit}_{i,j}\right)\right)}_{\text{None}}.\end{aligned}$$

Let's represent these grouped clauses in a simpler form as follows:

$$F_{\text{cnf}} = S_a \wedge S_{\neg a} \wedge G_a \wedge G_{\neg a} \wedge G_{\text{both}} \wedge G_{\text{none}}.$$

Now that we have separated a in all possible forms, we can examine it under the skeptical forgetting operation:

$$\text{SkepForget}(F_{\text{cnf}}, a) = \text{SkepForget}(S_a \wedge S_{\neg a} \wedge G_a \wedge G_{\neg a} \wedge G_{\text{both}} \wedge G_{\text{none}}, a).$$

In Proposition 14 we have shown that SkepForget is compatible under conjunction, hence we can transform such that:

$$\begin{aligned}SkepForget(F_{cnf}, a) =& SkepForget(S_a, a) \wedge SkepForget(S_{\neg a}, a) \wedge \\ & SkepForget(G_a, a) \wedge SkepForget(G_{\neg a}, a) \wedge \\ & SkeptForget(G_{both}, a) \wedge SkepForget(G_{none}, a).\end{aligned} \quad (*)$$

First, let's simplify each conjunctive forget formula within the previous sentence under (*):

$$\begin{aligned} SkepForget(S_a, a) &= (S_a[a/\bot] \wedge S_a[a/\top]) \\ &\equiv_\Sigma \bot \wedge \top \\ &\equiv_\Sigma \bot \end{aligned} \tag{R1}$$

$$\begin{aligned} SkepForget(S_{\neg a}, a) &= (S_{\neg a}[a/\bot] \wedge S_{\neg a}[a/\top]) \\ &\equiv_\Sigma \neg\bot \wedge \neg\top \\ &\equiv_\Sigma \top \wedge \bot \\ &\equiv_\Sigma \bot \end{aligned} \tag{R2}$$

$$\begin{aligned} SkepForget(G_a, a) &= (G_a[a/\bot] \wedge G_a[a/\top]) \\ &\equiv_\Sigma (\bot \vee \bigwedge_{i=1}^{n_1}(\bigvee_{j=1}^{m_i} lit_{i,j})) \wedge (\top \vee \bigwedge_{i=1}^{n_1}(\bigvee_{j=1}^{m_i} lit_{i,j})) \\ &\equiv_\Sigma (\bigwedge_{i=1}^{n_1}(\bigvee_{j=1}^{m_i} lit_{i,j})) \wedge \top \\ &\equiv_\Sigma \bigwedge_{i=1}^{n_1}(\bigvee_{j=1}^{m_i} lit_{i,j}) \end{aligned} \tag{R3}$$

$$\begin{aligned} SkepForget(G_{\neg a}, a) &= (G_{\neg a}[a/\bot] \wedge G_{\neg a}[a/\top]) \\ &\equiv_\Sigma (\neg\bot \vee \bigwedge_{i=1}^{n_2}(\bigvee_{j=1}^{m_i} lit_{i,j})) \wedge (\neg\top \vee \bigwedge_{i=1}^{n_2}(\bigvee_{j=1}^{m_i} lit_{i,j})) \\ &\equiv_\Sigma (\top \vee \bigwedge_{i=1}^{n_2}(\bigvee_{j=1}^{m_i} lit_{i,j})) \wedge (\bot \vee \bigwedge_{i=1}^{n_2}(\bigvee_{j=1}^{m_i} lit_{i,j})) \\ &\equiv_\Sigma \top \wedge (\bot \vee \bigwedge_{i=1}^{n_2}(\bigvee_{j=1}^{m_i} lit_{i,j})) \\ &\equiv_\Sigma \bigwedge_{i=1}^{n_2}(\bigvee_{j=1}^{m_i} lit_{i,j}) \end{aligned} \tag{R4}$$

$$SkepForget(G_{both}, a) = \top \tag{R5}$$

$$SkepForget(G_{none}, a) = G_{none} \tag{R6}$$

$$\begin{aligned} SkepForget(F_{cnf}, a) =& SkepForget(S_a, a) \wedge SkepForget(S_{\neg a}, a) \wedge \\ & SkepForget(G_a, a) \wedge SkepForget(G_{\neg a}, a) \wedge \\ & SkeptForget(G_{both}, a) \wedge SkepForget(G_{none}, a). \end{aligned} \tag{*}$$

We substitute the equation under (*) with results (R1-R6) and obtain:

$$SkepForget(F_{cnf}, a) \equiv_\Sigma \underbrace{\bot}_{\text{Case A}} \wedge \underbrace{\bot}_{\text{Case B}} \wedge \underbrace{\bigwedge_{i=1}^{n_1} \left(\bigvee_{j=1}^{m_i} lit_{i,j} \right)}_{\text{Case C}} \wedge \underbrace{\bigwedge_{i=1}^{n_2} \left(\bigvee_{j=1}^{m_i} lit_{i,j} \right)}_{\text{Case D}} \wedge \underbrace{\top}_{\text{Case F}} \wedge \underbrace{G_{none}}_{\text{Case G}}$$

Note that these groups of clauses may or may not appear in a specific sentence; therefore, we treat them as distinct cases, but they are **not** mutually exclusive. To simplify our analysis, equivalent cases of clause-groups are combined: Case A and Case B merge into Case 1, Case C and Case D combine into Case 2, Case F becomes Case 3, and Case G is designated as Case 4. Thus, we simplify to:

$$SkepForget(F_{cnf}, a) \equiv_\Sigma \underbrace{\bot}_{\text{Case 1}} \wedge \underbrace{\bigwedge_{i=1}^{n_1+n_2} \left(\bigvee_{j=1}^{m_i} lit_{i,j} \right)}_{\text{Case 2}} \wedge \underbrace{\top}_{\text{Case 3}} \wedge \underbrace{G_{none}}_{\text{Case 4}}.$$

We can specify the conditions under which these cases occur. If F_{cnf} includes a clause where:

- a is the sole literal (a or $\neg a$), then Case 1 occurs.
- a appears as a literal in conjunction with other literals $lit_{i,j}$, but not with its complement, then Case 2 occurs.
- $a \vee \neg a$ appears, possibly alongside other literals $lit_{i,j}$, then Case 3 occurs.
- a does not appear at all in any clause, then Case 4 occurs.

Now, we gain insight into the scenarios where we obtain drastic results ($\bot$ or $\top$) from $SkepForget$. Let F_{cnf} denote the CNF of F, and let $CS_{F_{cnf}}$ represent its clause set:

- If $CS_{F_{cnf}} \cap \{a, \neg a\} = \emptyset$ then $SkepForget(F, a) \equiv_\Sigma \top$.
- If $\{a\} \in CS_{F_{cnf}}$ or $\{\neg a\} \in CS_{F_{cnf}}$ then $SkepForget(F, a) \equiv_\Sigma \bot$.

In situations where only Case 2 and/or Case 4 appear, no drastic results are yielded. As observed, we found a way to determine, via the clause set $CS_{F_{cnf}}$, whether a drastic result ($\bot$ or $\top$) will be entailed from $SkepForget(F, a)$. Now, let's move away from the CNF topic and provide one final syntactic characterization or simplification for this part of the thesis.

For sentences subsentences that are in the form $F = a \circ G$ and where $a \notin Sig(G)$. We can give following characterization:

Proposition 17. *Let $a \in \Sigma$. Let $F \in \mathcal{L}_\Sigma$ be in such a form $F = a \circ G$ (or $\neg a \circ G$) where $\circ$ is either $\wedge$ or $\vee$ and where $a \notin Sig(G)$ then the following holds:*

- $Forget(F, a) = \top \circ G$
- $SkepForget(F, a) = \bot \circ G$

Proof. Let $F \in \mathcal{L}_\Sigma$, $a \in \Sigma$ and let $\circ$ be $\wedge$ or $\vee$.

$$\begin{aligned} Forget(F, a) &= F[a/\top] \vee F[a/\bot] \\ &\equiv_\Sigma (\bot \circ G) \vee (\top \circ G) \\ &\equiv_\Sigma (\bot \vee \top) \circ G \\ &\equiv_\Sigma \top \circ G \end{aligned}$$

$$\begin{aligned} SkepForget(F, a) &= F[a/\top] \wedge F[a/\bot] \\ &\equiv_\Sigma (\bot \circ G) \wedge (\top \circ G) \\ &\equiv_\Sigma (\bot \wedge \top) \circ G \\ &\equiv_\Sigma \bot \circ G \end{aligned}$$

□

We can extend Proposition 17 to the following:

Proposition 18. *Let $a \in \Sigma$. If we have a sentence $F \in \mathcal{L}_\Sigma$ in such a form $F = (a \circ_1 G) \circ_2 H$ (or $\neg a \circ_1 G \circ_2 H$) where $\circ_1$ and $\circ_2$ is either $\wedge$ or $\vee$ and where $a \notin Sig(G)$ and $\notin Sig(H)$ then:*

- $Forget(F, a) = Forget(a \circ_1 G, a) \circ_2 H = (\top \circ_1 G) \circ_2 H$
- $SkepForget(F, a) = SkepForget(a \circ_1 G, a) \circ_2 H = (\bot \circ_1 G) \circ_2 H$

Proof. Let $F \in \mathcal{L}_\Sigma$, $a \in \Sigma$ and let $\circ_1$, $\circ_2$ be $\wedge$ or $\vee$.

$$\begin{aligned}
Forget(F, a) &= F[a/\top] \vee F[a/\bot] \\
&\equiv_\Sigma ((\bot \circ_1 G) \circ_2 H) \vee ((\top \circ_1 G) \circ_2 H) \\
&\equiv_\Sigma ((\bot \circ_1 G) \vee (\top \circ_1 G)) \circ_2 H \\
&\equiv_\Sigma Forget((a \circ_1 G), a) \circ_2 H \\
&\equiv_\Sigma (\top \circ_1 G) \circ_2 H
\end{aligned}$$

$$\begin{aligned}
SkepForget(F, a) &= F[a/\top] \wedge F[a/\bot] \\
&\equiv_\Sigma ((\bot \circ_1 G) \circ_2 H) \wedge ((\top \circ_1 G) \circ_2 H) \\
&\equiv_\Sigma ((\bot \circ_1 G) \wedge (\top \circ_1 G)) \circ_2 H \\
&\equiv_\Sigma SkepForget((a \circ_1 G), a) \circ_2 H \\
&\equiv_\Sigma (\bot \circ_1 G) \circ_2 H
\end{aligned}$$

□

We have observed how forget operations behave at the syntactic level. To eliminate a variable, it is replaced by a constant. This process "flattens out" the distinction that a variable holds regarding its truth values. In one scenario, it universally holds true, becoming a tautology; in another, it universally holds false, becoming a contradiction. Recognizing the implications for truth values, we now intend to expand our investigation to the semantic level in the subsequent sections.

3 Semantic Investigation

In this section, our aim is to develop a straightforward understanding or intuition regarding how our variable forgetting operates at a semantic level. Fortunately, our exploration is confined to propositional logic, where truth tables offer a clear depiction of truth value assignments across all possible interpretations or worlds.

3.1 Reduction of Signature

We observe that through the process of forgetting, at least one variable is substituted by constants such as $\top$ and $\bot$. This raises the pertinent question: What happens to the signature over which our formulae are defined? If the variable is no longer necessary, does it become redundant? Is there any further significance to this redundancy? To explore this question, let us consider a sentence $F = (a \wedge b)$ over a language $\mathcal{L}_\Sigma$.

a	b	F	$Forget(F, a) \equiv_\Sigma b$	$SkepForget(F, a) \equiv_\Sigma \bot$
0	0	0	0	0
0	1	0	1	0
1	0	0	0	0
1	1	1	1	0

Figure 6: Crossed Out Variable a

Figure 6 shows the respective equivalences of the two forget operations, namely b and $\bot$, indicating that the variable a no longer "plays a role". To illustrate this, we have crossed out a in the table with a red line. Our signature $\Sigma = \{a, b\}$ has now been reduced to a subset of Σ. We denote this subset as a subsignature $\Gamma \subseteq \Sigma$. Since a variable has been crossed out, can we omit it entirely? Let's examine the truth tables of our reduced signatures.

b	$Forget(F, a) \equiv_{\Gamma_1} b$	$SkepForget(F, a) \equiv_{\Gamma_2} \bot$
0	0	0
1	1	0

Figure 7: Collapsed Truth Tables

In Figure 7, we see that the resulting signatures are $\Gamma_1 = \{b\}$ for $Forget(F, a)$ and $\Gamma_2 = \emptyset$ for $SkepForget(F, a)$ with respect to F.

We observe that we can consider formulae under an extended (super) signature. For instance, we can consider $Forget(F, a)$ under Γ, but as shown in Figure 6, we can also consider it under the extended signature Σ. For example we can give the models of $Forget(F, a)$ in two regards: $Mod_\Sigma(Forget(F, a)) = \{\bar{a}b, ab\}$ and $Mod_\Gamma(Forget(F, a)) = \{b\}$. It should be noted that we cannot consider F under the smaller signature Γ, as F mentions a variable that is not included in Γ.

To formally describe signatures, we denoted $Sig(F)$ as the set of atomic variables that appear in F. Now, consider a formula $G = \bot \wedge b$, which is equivalent to $SkepForget(F, a)$. Then we have $Sig(F) = \{b\}$, even though we know that $G \equiv_\Sigma \bot$ and hence the signature should be empty? What we need here is a notion of the **minimal** signature of a formula.

In [SBKI24], the authors provide the possibility through the following proposition:

Proposition 19 (Minimal Signature [SBKI24]). *Let $F \in \mathcal{L}_\Sigma$ then $Sig_{min}(F)$ is the set of those atoms that distinguish models of F from non-modelsof F by exactly one signature element.*

$$Sig_{min}(F) = \{a \in \Sigma \mid \exists \omega_1, \omega_2 \in \Omega.\ \omega_1 \models_\Sigma F \; and \; \omega_2 \not\models_\Sigma F \; and \; \omega_1^{\Sigma \setminus \{a\}} = \omega_2^{\Sigma \setminus \{a\}}\}.$$

Example 20. *Let $G = Forget(F, a)$ over $\mathcal{L}_\Sigma$. The pair of interpretations that are the same except for one variable and the following holds: $\omega_1 \models_\Sigma G$ and $\omega_2 \not\models_\Sigma G$.*

- $\omega_1(a) = 1,\ \omega_1(b) = 1$ and $\omega_2(a) = 1,\ \omega_2(b) = 0$
- $\omega_1(a) = 0,\ \omega_1(b) = 1$ and $\omega_2(a) = 0,\ \omega_2(b) = 0$

Intuitively, one can say: b is necessary or relevant to be a model of G. Therefore, the result is $Sig_{min}(G) = \{b\}$, which is also equal to Γ_1 in Figure 7. Essentially, we exclude all irrelevant variables from the signature and retain the relevant ones. Now, let's assume $G = SkepForget(F, a)$. We cannot find any $\omega \in Mod_\Sigma(G)$, hence $Sig_{min}(G) = \emptyset$, which is equal to Γ_2 in Figure 7.

Indeed, we now observe that through the forgetting of variables, the signature gradually shrinks as more variables are forgotten, until we reach the extreme case of $\Gamma = \emptyset$. This behavior is expected and desired from our perspective.

Another intriguing side point arises when we examine formulae such as $Forget(F, a)$ or more precisely through a substitution like $F[a/\bot]$. From a purely "symbolic" perspective, a appears in the string "$F[a/\top]$". However, the equivalence of the formula $F[a/\top]$ disregards a, as we see $F[a/\bot] \equiv_\Sigma \bot \wedge b$.

3.2 Ambiguities

In general, when examining interpretations in a truth table, each assignment or truth value of a variable serves as a crucial criterion for distinguishing one interpretation from another. However, as we saw in the previous section, it must be noted that not every variable is necessarily required to satisfy a particular formula.

Forgetting implies that after the "act of forgetting", the information is no longer available to us. Applied to our context, we can observe that this principle is reflected in the behavior of our two forgetting approaches. When we forget a variable $a \in \Sigma$ in a formula $F \in \mathcal{L}_\Sigma$, we observe that after applying the operation, the forget formula

$SkepForget(F, a)$ (or $Forget(F, a)$) no longer "speaks" about a. As discussed in the previous section, their interpretations can be represented in or mapped to Ω_Γ, where $\Gamma \subseteq \Sigma$. Since our variable forgetting suppresses the variable a, rendering its truth values as irrelevant, ambiguities arise. We can no longer determine how or which interpretations from $\omega \in \Omega_\Sigma$ should be mapped to Ω_Γ. Because, originally, these interpretations depended on the truth value $\{1, 0\}$ of a, we have two reduction or mapping possibilities: one where a was 1, and one where a was 0.

Option 1:

~~a~~	b	F_1
~~0~~	0	0
~~0~~	1	0
~~1~~	~~0~~	~~0~~
~~1~~	~~1~~	~~1~~

Option 2:

~~a~~	b	F_1
~~0~~	~~0~~	~~0~~
~~0~~	~~1~~	~~0~~
~~1~~	0	0
~~1~~	1	1

Figure 8: Collapse Options

In Figure 8 we observe that there are two interpretations that yield the same truth value for a and we cannot ascertain which pair should be retained in order to make a "indifferent". The question arises: Is there a "correct" reduction here, or is it a matter of contingent definition? We anticipate that our two forgetting operations resolve this ambiguity in different ways, which we will demonstrate in the next section.

3.3 Two ways to resolve the Ambiguity

Through a simple example, we have already gained significant insights into the behavior of both operations: the reduction of the signature and the emergence of ambiguity. We now wish to examine how our two operations represent forgetting differently, with a greater focus on the actual truth values rather than the signature. Additionally, we intend to consider more complex examples and observe their behavior. Consider the truth values in a specific example for a formula $F = (a \wedge b)$ over $\mathcal{L}_\Sigma$.

a	b	F	$Forget(F, a) \equiv_\Sigma b$	$SkepForget(F, a) \equiv_\Sigma \bot$
0	0	0	0	0
0	1	0	1	0
1	0	0	0	0
1	1	1	1	0

Figure 9: Two mapping Patterns

As illustrated in Figure 9, we observe two mappings. Firstly, we see that the truth values of F for all $\omega \in Mod(a)$ are essentially mapped to all interpretations of $Forget(F, a)$. Similarly, we observe the same for $\omega \notin Mod(a)$ in relation to F and $SkepForget(F, a)$.

a	b	c	d	F	$Forget(F, a) \equiv_\Sigma (b \vee (c \wedge d))$	$SkepForget(F, a) \equiv_\Sigma (c \wedge d)$
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	1	1
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	1

Figure 10: More complex Example

In Figure 10 that the same pattern holds for more complex sentences $F = (a \wedge b) \vee (c \wedge d)$ as well, but i does not hold for negation, lets say $F = \neg(a \wedge b)$ as we can see in the following figure.

a	b	$\neg F$	$Forget(\neg F, a) \equiv_\Sigma \top$	$SkepForget(\neg F, a) \equiv_\Sigma \neg b$
0	0	1	1	1
0	1	1	1	0
1	0	1	1	1
1	1	0	1	0

Figure 11: Negation Example

As illustrated in Figure 11, under negation, there is a notable "swap" pattern between the operations compared to the previous patterns.

We have observed that both operations exhibit a distinct mapping pattern. The mapping is not precisely identical but suggests a dual relationship. We also observe a specific duplication effect. In the next section, we will delve deeper into this phenomenon.

4 Semantic Characterization

Up to this point, we observed and analyzed the behavior of our object of interest. Building upon these intuitions, our next objective is to develop comprehensive semantic characterizations.

4.1 $\top$ and $\bot$ Substitutions

In both classical and skeptical variable forgetting, we utilize, by definition, the substitution of a variable by logical constants, such as $F[a/\top]$ and $F[a/\bot]$. To address the semantic properties of $SkepForget$ (and $Forget$), let's first specifically examine the semantic properties of these substitutions to gain foundational knowledge to build upon.

Consider three formulae $F_1 = \neg a$, $F_2 = a \wedge b$, and $F_3 = a \vee b$ over the language $\mathcal{L}_\Sigma$. We will compare the truth values of the a-substitutions of each formula with $\top$ and $\bot$ using a truth table.

a	b	F_1	F_2	F_3	$F_1[a/\top]$	$F_2[a/\top]$	$F_3[a/\top]$	$F_1[a/\bot]$	$F_2[a/\bot]$	$F_3[a/\bot]$
0	0	1	0	0	0	0	1	1	0	0
0	1	1	0	1	0	1	1	1	0	1
1	0	0	0	1	0	0	1	1	0	0
1	1	0	1	1	0	1	1	1	0	1

Figure 12: Behaviour of Substitutions

In Figure 12, we have arranged the columns based on the truth values of the "variable to be forgotten" to visually highlight specific patterns. The red axis separates the truth values of a. By examining these patterns, we can observe how the substitutions relate to their original forms (F_1-F_3). Let's focus on $F_1[a/\top]$-$F_3[a/\top]$, indicated in blue. When we observe interpretations where $\omega(a) = 1$, we notice that the same truth values are "duplicated" across interpretions where $\omega(a) = 0$. The same group of truth values correspond to their respective sentences of origin ($F_1 - F_3$). Conversely, for $F_1[a/\bot]$-$F_3[a/\bot]$, we observe a similar but reversed pattern. This duplication echoes the pattern discussed in the previous section. To gain deeper insights into why certain interpretations are duplicated, let us explore the semantic sets for substitutions, namely $\{\omega \mid \omega[a \mapsto 1]\}$ for $F[a/\top]$ and $\{\omega \mid \omega[a \mapsto 0]\}$ for $F[a/\bot]$. Given that our truth values are bivalent ({0,1}), for each variable assignment we have two possibilities or interpretations. Now, if we map a single truth value to each and therefore "override" the two complementary assigments, duplication "naturally" arises. From this standpoint, the mapping results in duplication, while from another viewpoint, it represents the reduction we discussed earlier. Essentially, we consolidate two interpretations into one.

We observe that we still retain all the necessary information to reconstruct our sentences from the substitutions. Trivially, the set of all interpretations results as

$\Omega = \{\omega \mid \omega \models_\Sigma a \cup \omega \not\models_\Sigma a\}$. We can formally reconstruct and "reassemble" $Mod_\Sigma(F)$ from the substitutions as follows:

- $Mod_\Sigma(F) = \{\omega \mid \omega \models_\Sigma F[a/\top], \omega \models a\} \cup \{\omega \mid \omega \models_\Sigma F[a/\bot], \omega \not\models_\Sigma a\}$
- $Mod_\Sigma(F) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F$ or $\omega[a \mapsto 0] \models_\Sigma F\}$

To sum it up:

- $Mod_\Sigma(F[a/\top])$ contains all models ω where either $\omega(a) = 1$ or where a is irrelvant for $\omega \models_\Sigma F$.
- $Mod_\Sigma(F[a/\bot])$ contains all models ω where either $\omega(a) = 0$ or where a is irrelvant for $\omega \models_\Sigma F$.

Now, before we proceed to the next section, let us provide a semantic characterization of the syntactic definition of substitutions.

Proposition 21 (Semantic characterization of $\top$ Substituion). *Let $\omega \in \Omega_\Sigma$, $F \in \mathcal{L}_\Sigma$ be a formula and a a variable over Σ.*

$$Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}.$$

Proof. By structural induction of $F \in \mathcal{L}_\Sigma$, we can show $Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}$ holds.

Base case: $F = a$**.** Then, we have $F[a/\top] \equiv_\Sigma \top$. Consequently we optain:

$$Mod_\Sigma(F[a/\top]) = Mod_\Sigma(\top) = \Omega$$

We show that $Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}$ holds:

$\subseteq$ Direction: Let $\omega \in Mod_\Sigma(F[a/\top])$ and let $\omega' = \omega[a \mapsto 1]$. Then, we have $\omega' \models_\Sigma F$, as $\omega'(a) = 1$. Consequently, we have:

$$\omega \in \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}$$

$\supseteq$ Direction:

$$\{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\} \subseteq \Omega$$

holds trivially.

Base case: $F = b$, for $b \in \Sigma$ and $b \neq a$. Then, we have $F[a/\top] = b$, and consequently, $F[a/\top] = F$. We show that: $Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}$ holds. Observe that

$$Mod_\Sigma(F) = \{\omega \mid \omega(b) = 1\}$$

holds. Consequently, we have that,

$$Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega(b) = 1\}.$$

We have the following chain of equivalences:

$$\begin{aligned} Mod_\Sigma(F) &= Mod_\Sigma(F[a/\top]) \\ &= \{\omega \mid \omega(b) = 1\} \\ &= \{\omega \mid \omega(b) = 1, \omega(a) = 1\} \cup \{\omega | \omega(b) = 1, \omega(a) = 0\} \\ &= \underbrace{\{\omega \mid \omega \models_\Sigma F, \omega(a) = 1\}}_{X_1} \cup \underbrace{\{\omega \mid \omega \models_\Sigma F, \omega(a) = 0\}}_{X_2} \end{aligned}$$

Now observe that for each $\omega' \in X_2$ exists $\omega \in X_1$ such that $\omega = \omega'[a \mapsto 1]$. Consequently we have:

$$\begin{aligned} X_1 \cup X_2 &= \{\omega \mid (\omega \models_\Sigma F \, and \, \omega(a) = 1) \, or \, \omega[a \mapsto 1] \models_\Sigma F\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}. \end{aligned}$$

Induction case: $F = \neg G$**.**
From the structural induction we obtain,

$$Mod_\Sigma(G[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\}.$$

Now, observe that:

$$\begin{aligned} Mod_\Sigma(F[a/\top]) &= Mod(\neg G[a/\top]) \\ &= \Omega \setminus Mod_\Sigma(G[a/\top]) \end{aligned}$$

Hence, we have:

$$\begin{aligned} Mod_\Sigma(F[a/\top]) &= \Omega \setminus Mod_\Sigma(G[a/\top]) \\ &= \Omega \setminus \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\} \\ &= \{\omega \mid \omega[a \mapsto 1] \not\models_\Sigma G\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma \neg G\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\} \end{aligned}$$

Induction case: $F = G \wedge H$**.** The following chain of quivalences holds:

$$\begin{aligned} Mod_\Sigma(F[a/\top]) &= Mod_\Sigma((G \wedge H)[a/\top]) \\ &= Mod_\Sigma(G[a/\top] \wedge H[a/\top]) \\ &= Mod_\Sigma(G[a/\top]) \cap Mod_\Sigma(H[a/\top]) \end{aligned}$$

From the induction, we obtain:

$$Mod_\Sigma(G[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\}$$
$$Mod_\Sigma(H[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma H\}$$

So, we obtain the following equivalence:

$$Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\} \cap \{\omega \mid \omega[a \mapsto 1] \models_\Sigma H\}. \qquad (*)$$

Using set theory, we obtain from (*):

$$\begin{aligned} Mod_\Sigma(F[a/\top]) &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\} \cap \{\omega \mid \omega[a \mapsto 1] \models_\Sigma H\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G, \omega[a \mapsto 1] \models_\Sigma H\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma (G \wedge H)\} \end{aligned}$$

Induction case: $F = G \vee H$**.** The following chain of quivalences holds:

$$\begin{aligned} Mod_\Sigma(F[a/\top]) &= Mod_\Sigma((G \vee H)[a/\top]) \\ &= Mod_\Sigma(G[a/\top] \vee H[a/\top]) \\ &= Mod_\Sigma(G[a/\top]) \cup Mod_\Sigma(H[a/\top]) \end{aligned}$$

From the induction, we obtain:

$$Mod_\Sigma(G[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\}$$
$$Mod_\Sigma(H[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma H\}$$

So, we obtain the following equivalence:

$$Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G\} \cup \{\omega \mid \omega[a \mapsto 1] \models_\Sigma H\}. \qquad (**)$$

Using set theory, we obtain from (**):

$$\begin{aligned} Mod_\Sigma(F[a/\top]) &= \{\omega|\omega[a \mapsto 1] \models_\Sigma G\} \cup \{\omega \mid \omega[a \mapsto 1] \models_\Sigma H\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma G \text{ or } \omega[a \mapsto 1] \models_\Sigma H\} \\ &= \{\omega \mid \omega[a \mapsto 1] \models_\Sigma (G \vee H)\} \end{aligned}$$ □

Proposition 22 (Semantic characterization of $\bot$ Substituion)**.** *Let $\omega \in \Omega_\Sigma$, $F \in \mathcal{L}_\Sigma$ be a formula and a a variable over Σ.*

$$Mod_\Sigma(F[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}.$$

Proof. By structural induction of $F \in \mathcal{L}_\Sigma$, we can show

$$Mod_\Sigma(F[a/\bot]) = \{\omega|\omega[a \mapsto 0] \models_\Sigma F\}$$

holds.

Base case: $F = a$**.** Then, we have $F[a/\bot] \equiv_\Sigma \bot$. Consequently we optain:

$$Mod_\Sigma(F[a/\bot]) = Mod_\Sigma(\bot) = \emptyset.$$

We show that $Mod_\Sigma(F[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}$ holds:

$\subseteq$ Direction:

$$\emptyset \subseteq \{\omega | \omega[a \mapsto 0] \models_\Sigma F\}$$

holds trivially.

$\supseteq$ Direction: Let $\omega' = \omega[a \mapsto 0]$. Then, we have $\omega' \not\models_\Sigma F$, as $\omega'(a) = 0$. Consequently, we have:

$$\{\omega | \omega[a \mapsto 0] \models_\Sigma F\} \subseteq \emptyset.$$

Base case: $F = b$, for $b \in \Sigma$ and $b \neq a$. Then, we have $F[a/\bot] = b$, and consequently, $F[a/\bot] = F$. We show that

$$Mod_\Sigma(F[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}$$

holds. Observe that $Mod_\Sigma(F) = \{\omega | \omega(b) = 1\}$ holds. Consequently, we have that,

$$Mod_\Sigma(F[a/\bot]) = \{\omega | \omega(b) = 0\}.$$

We have the following chain of equivalences:

$$\begin{aligned}
Mod_\Sigma(F) &= Mod_\Sigma(F[a/\bot]) \\
&= \{\omega \mid \omega(b) = 1\} \\
&= \{\omega \mid \omega(b) = 1, \omega(a) = 1\} \cup \{\omega | \omega(b) = 1, \omega(a) = 0\} \\
&= \underbrace{\{\omega \mid \omega \models_\Sigma F, \omega(a) = 1\}}_{X_1} \cup \underbrace{\{\omega \mid \omega \models_\Sigma F, \omega(a) = 0\}}_{X_2}
\end{aligned}$$

Now observe that for each $\omega' \in X_1$ exists $\omega \in X_2$ such that $\omega = \omega'[a \mapsto 0]$. Consequently we have:

$$\begin{aligned}
X_1 \cup X_2 &= \{\omega | (w \models_\Sigma F \, and \, \omega(a) = 0) \, or \, \omega[a \mapsto 0] \models_\Sigma F\} \\
&= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}
\end{aligned}$$

Induction case: $F = \neg G$**.** From the structural induction we obtain,

$$Mod(G[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma G\}.$$

Now, observe that:

$$
\begin{aligned}
Mod_\Sigma(F[a/\bot]) &= Mod_\Sigma(\neg G[a/\bot]) \\
&= \Omega \setminus Mod_\Sigma(G[a/\bot])
\end{aligned}
$$

Hence, we have:

$$
\begin{aligned}
Mod_\Sigma(F[a/\bot]) &= \Omega \setminus Mod_\Sigma(G[a/\bot]) \\
&= \Omega \setminus \{\omega|\omega[a \mapsto 0] \models_\Sigma G\} \\
&= \{\omega \mid \omega[a \mapsto 0] \not\models_\Sigma G\} \\
&= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma \neg G\} \\
&= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}
\end{aligned}
$$

Induction case: $F = G \wedge H$**.** The following chain of quivalences holds:

$$
\begin{aligned}
Mod_\Sigma(F[a/\bot]) &= Mod_\Sigma((G \wedge H)[a/\bot]) \\
&= Mod_\Sigma(G[a/\bot] \wedge H[a/\bot]) \\
&= Mod_\Sigma(G[a/\bot]) \cap Mod_\Sigma(H[a/\bot])
\end{aligned}
$$

From the induction, we obtain:

$$
\begin{aligned}
Mod_\Sigma(G[a/\bot]) &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma G\} \\
Mod_\Sigma(H[a/\bot]) &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma H\}
\end{aligned}
$$

So, we obtain the following equivalence:

$$
Mod_\Sigma(F[a/\bot]) = \{\omega|\omega[a \mapsto 0] \models_\Sigma G\} \cap \{\omega \mid \omega[a \mapsto 0] \models_\Sigma H\}. \qquad (*)
$$

Using set theory, we obtain from (*):

$$
\begin{aligned}
Mod_\Sigma(F[a/\bot]) &= \{\omega|\omega[a \mapsto 0] \models_\Sigma G\} \cap \{\omega|\omega[a \mapsto 0] \models_\Sigma H\} \\
&= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma G, \omega[a \mapsto 0] \models_\Sigma H\} \\
&= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma (G \wedge H)\}
\end{aligned}
$$

Induction case: $F = G \vee H$**.** The following chain of quivalences holds:

$$
\begin{aligned}
Mod_\Sigma(F[a/\bot]) &= Mod_\Sigma((G \vee H)[a/\bot]) \\
&= Mod_\Sigma(G[a/\bot] \vee H[a/\bot]) \\
&= Mod_\Sigma(G[a/\bot]) \cup Mod_\Sigma(H[a/\bot])
\end{aligned}
$$

From the induction, we obtain:

$$
\begin{aligned}
Mod_\Sigma(G[a/\bot]) &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma G\} \\
Mod_\Sigma(H[a/\bot]) &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma H\}
\end{aligned}
$$

So, we obtain the following equivalence:

$$Mod_\Sigma(F[a/\bot]) = \{\omega | \omega[a \mapsto 0] \models_\Sigma G\} \cup \{\omega \mid \omega[a \mapsto 0] \models_\Sigma H\}. \quad (**)$$

Using set theory, we obtain from (**):

$$\begin{aligned} Mod_\Sigma(F[a/\bot]) &= \{\omega | \omega[a \mapsto 0] \models_\Sigma G\} \cup \{\omega \mid \omega[a \mapsto 0] \models_\Sigma H\} \\ &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma G \text{ or } \omega[a \mapsto 0] \models H\} \\ &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma (G \vee H)\} \end{aligned}$$

□

4.2 Forget Relation and Relevance

Lin and Reiter describe this forget relation in their seminal paper [LR94] for first-order logic. However, this concept also holds relevance in propositional logic, as discussed further in related research [EKI19]. Formally, we can define this forget relation as follows:

Definition 23 (Forget Relation)**.** *Given two interpretations $\omega_1, \omega_2 \subseteq \Omega_\Sigma$, we say $\omega_1 \sim_a \omega_2$ if ω_1 and ω_2 agree on all variables except possibly on the truth value of a.*

Let $x_1, x_2, y_1, y_2 \in \Omega_\Sigma$ be interpretations and let $F \in \mathcal{L}_\Sigma$ and $a \in \Sigma$.

$\omega_1 \sim_a \omega_2$	a	b	F	$Forget(F, a) \equiv_\Sigma \top$	$SkepForget(F, a) \equiv_\Sigma b$
x_2	0	0	0	1	0
x_1	1	0	1	1	0
y_2	0	1	1	1	1
y_1	1	1	1	1	1

Figure 13: Forget Relation Table

We have organized the truth table as depicted in Figure 13 based on pairs of a forget relation $\omega_1 \sim_a \omega_2$ over a. For each interpretation ω_1, there exists another interpretation ω_2 such that ω_2 agrees with ω_1 on all variables except for a. For example, consider the interpretations x_1 and x_2. This illustrates the duplication effect we discussed earlier induced by the substitutions.

There exists a relationship between the model set of F and the model set of $Forget(F, a)$ or $SkepForget(F, a)$, which is determined by pairs of interpretations in the forget relation. Let's explore this further: For each interpretation $\omega_1 \in \Omega_\Sigma$, there exists an interpretation ω_2 such that $\omega_1 \sim_a \omega_2$. We make four case distinctions:

(1) $\omega_1 \models_\Sigma F$ and $\omega_2 \models_\Sigma F$

(2) $\omega_1 \not\models_\Sigma F$ and $\omega_2 \models_\Sigma F$

(3) $\omega_1 \models_\Sigma F$ and $\omega_2 \not\models_\Sigma F$

(4) $\omega_1 \not\models_\Sigma F$ and $\omega_2 \not\models_\Sigma F$

Figure 14: Cases of Forget Relation and Models of F

Let's discuss the cases for $\omega_1 \sim_a \omega_2$ as stated in Figure 14:

- In Case (1), both ω_1 and ω_2 satisfy F, the assignment of a does not affect the satisfaction of F because both interpretations necessarily yield opposite assignments for a. Thus, we can infer that a is irrelevant for satisfying F.
- In Cases (2) and (3), where either ω_1 or ω_2 satisfies F while the other does not. The assignment of a is a necessary condition for satisfaction, both interpretations agree on all assignments except for a. Here, we can conclude that a is relevant for satisfying F.
- In Case (4), where neither ω_1 nor ω_2 satisfies F, it is trivially observed that a is irrelevant for satisfying F.

Now, let us examine the outcome of forgetting, focussing on the Γ-parts of ω_1 and ω_2, denoted as ω_1^Γ and ω_2^Γ. We obtain the same (collapsed) interpretation $\omega_f \in \Gamma$ where $\omega_f = \omega_1^\Gamma = \omega_2^\Gamma$. This implies that, for a forget formula (e.g., $Forget(F, a)$), there must exist a truth value that is identical for both ω_1 and ω_2, even if the truth values for F possibly differ between them. In retrospect, we can state that the "essence" of forgetting lies in the requirement that pairs within the forgetting relation yield the equal truth value for the respective forget formulae.

We have now established the interrelation between the forget relation $\omega_1 \sim_a \omega_2$, the variable a, and the models of F. We observed that forgetting necessitates assigning equal truth values to the forget formulae under ω_1 and ω_2. This raises the question of how the substitutions $F[a/\bot]$ and $F[a/\top]$ relate to these insights, given their crucial role in determining the outcome. A key inquiry here is under what conditions both substitutions are true (or false), considering that they "insert" opposite values for a. We will delve into these questions in the next subsection.

4.3 Inclusion or Exclusion

This subsection aims to show the step-by-step transition from the truth values of F to those of the substitutions $F[a/\bot]$ and $F[a/\top]$, and then, in the final step, to the values of the forget formulae. Afterward, we can characterize the key distinctions between $Forget$ and $SkepForget$. Let's revisit the three examples: $F_1 = \neg a$, $F_2 = a \wedge b$, and $F_3 = a \vee b$, and analyze their respective truth tables.

a	b	F_1	$F_1[a/\top]$	$F_1[a/\bot]$	$Forget(F_1, a)$	$SkepForget(F_1, a)$
0	0	1	0	1	1	0
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	1	0	0	1	1	0

a	b	F_2	$F_2[a/\top]$	$F_2[a/\bot]$	$Forget(F_2, a)$	$SkepForget(F_2, a)$
0	0	0	0	0	0	0
0	1	0	1	0	1	0
1	0	0	0	0	0	0
1	1	1	1	0	1	0

a	b	F_3	$F_3[a/\top]$	$F_3[a/\bot]$	$Forget(F_3, a)$	$SkepForget(F_3, a)$
0	0	0	1	0	1	0
0	1	1	1	1	1	1
1	0	1	1	0	1	0
1	1	1	1	1	1	1

Figure 15: Compare Substitutions with Forget Operations

In Figure 15, we observe a pattern, highlighted in blue and red, illustrating the relationship between F and the substitutions under the forget relation as follows: Let F be a formula, $a \in \Sigma$ and $\omega_1, \omega_2 \in \Omega_\Sigma$ such that $\omega_1 \sim_a \omega_2$. Then:

$$\omega_1(F) = \omega_2(F) \text{ if and only if } \omega_1(F[a/\top]) = \omega_1(F[a/\bot]) \text{ and } \omega_2(F[a/\top]) = \omega_2(F[a/\bot]) \tag{A}$$

$$\omega_1(F) \neq \omega_2(F) \text{ if and only if } \omega_1(F[a/\top]) \neq \omega_1(F[a/\bot]) \text{ and } \omega_2(F[a/\top]) \neq \omega_2(F[a/\bot]) \tag{B}$$

Given a forget relation $\omega_1 \sim_a \omega_2$, the statement referenced under (A) describes that the substitutions $F[a/\bot]$ and $F[a/\top]$ yield equal truth values under the same interpretation (respectively ω_1 and ω_2), if and only if the truth values of F under ω_1 and ω_2 are also equal. This scenario is depicted in blue in Figure 15. Conversely, the statement referenced under (B), highlighted in red, describes that the substitutions yield opposing truth values under the same respective interpretations, if and only if the truth values of F under ω_1 and ω_2 are also opposing.

Note that the left-hand side of the statements referenced under (A) and (B) pertains to the equality of truth values of the **same** formula under **different** interpretations ω_1 and ω_2, while on the right-hand side, we want to highlighting the relation of the equality of truth values of **different** formulae under the **same** interpretation.

Additionally, we observe that,

$$\text{If } \omega_1 \sim_a \omega_2, \text{ then } \omega_1(F[a/\top]) = \omega_2(F[a/\top]) \text{ and} \\ \omega_1(F[a/\bot]) = \omega_2(F[a/\bot])$$

wich describes the duplication effect we discussed in Subsection 3.3. To draw a connection to the case distinctions we discussed in Figure 14, consider the following. For each $\omega \in \Omega_\Sigma$, $a \in \Sigma$ and a formula F over $\mathcal{L}_\Sigma$:

$$\text{If } \omega(F[a/\bot]) = \omega(F[a/\top]) \text{ then } a \text{ is irrelevant to satisfy } F.$$
$$\text{If } \omega(F[a/\bot]) \neq \omega(F[a/\top]) \text{ then } a \text{ is relevant to satisfy } F.$$

Recall, in previous section we learned that forgetting involves assigning the same truth value to the forget formulae under ω_1 and ω_2 where $\omega_1 \sim_a \omega_2$, such that they collapse into $\omega_f \in \Omega_\Gamma$ where $\omega_f = \omega_1^\Gamma = \omega_2^\Gamma$. How to determine this value? If $\omega_1(F) = \omega_2(F)$ then then a is irrelevant to satisfy F and the decision is straightforward since both interpretations already assign the same truth value to F; we can simply adopt the same value.

The dilemma arises when one interpretation is a model of F while the other is not. This is the situation $\omega_1(F) \neq \omega_2(F)$ and wich we observed previously in Figure 14 as Cases 2 and 3. This is also the situation of ambiguity we discussed earlier in Subsection 3.2. Here, we face the necessity of deciding a unified truth value for the forget formula under both interpretations ω_1 and ω_2 while the sentence of origin F yields opposing values. This is essentially the point of "forgetting", the point where we lose information. One could argue that it necessitates the introduction of "false" information. However, this decision results in either a surplus or a deficit of models in the outcome, thereby explaining the occurrence of tautologies or contradictions, respectively.

Similar to the equations referenced under (A) and (B) we can relate truth values of F to the respective forget formulae as well. Again, let $F \in \mathcal{L}_\Sigma$, $a \in \Sigma$ and $\omega_1, \omega_2 \in \Omega_\Sigma$ such that $\omega_1 \sim_a \omega_2$. Then:

$$\omega_1(F) = \omega_2(F) \text{ if and only if } \omega_1(Forget(F,a)) = \omega_1(SkepForget(F,a)) \text{ and} \\ \omega_2(Forget(F,a)) = \omega_2(SkepForget(F,a)). \tag{C}$$

$$\omega_1(F) \neq \omega_2(F) \text{ if and only if } \omega_1(Forget(F,a)) \neq \omega_1(SkepForget(F,a)) \text{ and} \\ \omega_2(Forget(F,a)) \neq \omega_2(SkepForget(F,a)). \tag{D}$$

The equations given under (C) and (D) demonstrate what we just described. Under (C) we have the straighforward assigment and under (D) we have the situation of ambiguity that each operation resolves differently.

Now, lets us observe how each forget operation resolves the aforementioned ambiguity differently. The elimination of a through substitutions has possibly already introduced the drastic results. At this point, the decision only differs in how these results are further resolved. In Figure 15, we see that for $Forget$ formulae, the decision is determined by the maximum value, $Max(\{\omega(F[a/\bot]), \omega(F[a/\top])\})$. In contrast, for $SkepForget$ formulae, we obtain $Min(\{\omega(F[a/\bot]), \omega(F[a/\top])\})$, hence the thesis title. This aspect is not surprising given their definitions.

In other words, if ω_1 and ω_2 are models of F then both are included in the set $Mod_\Sigma(SkepForget(F, a))$. Also then both are models for $F[a/\bot]$ and $F[a/\top]$. However, if only one of ω_1 or ω_2 is a model, neither belongs to $Mod_\Sigma(SkepForget(F, a))$; instead, both interpretations are elements of $Mod_\Sigma(Forget(F, a))$. The key distinction is that $Forget$ includes interpretations of the forget relation where a is relevant to satisfy F, while $SkepForget$ excludes them. In summary, we can formally capture both behaviors as follows:

For each $\omega_1 \in Mod_\Sigma(F)$ there is an ω_2 with $\omega_1 \sim_a \omega_2$ such that:

- $\omega_1 \in Mod_\Sigma(F) \lor \omega_2 \notin Mod_\Sigma(F)$ implies $\omega_1, \omega_2 \in Mod_\Sigma(Forget(F, a))$
- $\omega_1, w_2 \in Mod_\Sigma(F)$ implies $\omega_1, \omega_2 \in Mod_\Sigma(SkepForget(F, a))$.

Additionally, we can obtain a set-theoretic characterization for the models of both operations, which we will use in the following sections as well:

Proposition 24. *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$.*

$$\begin{aligned} Mod_\Sigma(Forget(F, a)) &= Mod_\Sigma(F[a/\top]) \cup Mod_\Sigma(F[a/\bot]) \\ Mod_\Sigma(SkepForget(F, a)) &= Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot]) \end{aligned}$$

Proof. By the definition of Mod we obtain:

$$\begin{aligned} Mod_\Sigma(Forget(F, a)) &= Mod_\Sigma(F[a/\top] \lor F[a/\bot]) \\ &= Mod_\Sigma(F[a/\top]) \cup Mod_\Sigma(F[a/\bot]) \end{aligned}$$

$$\begin{aligned} Mod_\Sigma(SkepForget(F, a)) &= Mod_\Sigma(F[a/\top] \land F[a/\bot]) \\ &= Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot]) \end{aligned}$$ □

4.4 Minimizer or Maximizer

As noted earlier in Subsection 3.1, after applying both forget operations, the signature Σ is reduced to a subsignature Γ, shrinking to the extreme case of $\emptyset$. However, we can observe different behaviors for the model sets of F.

- $Mod_\Sigma(F) \subseteq Mod_\Sigma(Forget(F, a))$
- $Mod_\Sigma(F) \supseteq Mod_\Sigma(SkepForget(F, a))$

We observe contrasting effects between $Forget$ and $SkepForget$: $Forget$ tends to expand the set of models, while $SkepForget$ tends to contract it. As noted in [LR94], $F \models_\Sigma Forget(F, a)$. Conversely, $SkepForget(F, a) \models_\Sigma F$. This preference for disjunctive variable forgetting likely stems from its ability to extract "weaker" knowledge from a presumed "richer" base. Additionally, in extreme cases, $Forget$ can lead to a tautology, while $SkepForget$ may result in a contradiction. The implications of the "principle of explosion", where any arbitrary formula can be derived from a contradiction, are relevant considerations in the application of this technique.

Proposition 25. *For a formula $F \in \mathcal{L}_\Sigma$ and a variable $a \in \Sigma$. The following holds:*

$$SkepForget(F, a) \models_\Sigma F.$$

Proof. Let $\omega \in \Omega_\Sigma$, $F \in \mathcal{L}_\Sigma$ be a formula and let $a \in \Sigma$ be a variable. We want to proof that the following holds:

$$SkepForget(F, a) \models_\Sigma F.$$

By definition of $SkepForget$ we have:

$$F[a/\top] \wedge F[a/\bot] \models_\Sigma F.$$

Now we can obtain the model set relations such that:

$$Mod_\Sigma(F[a/\top] \wedge F[a/\bot]) \subseteq Mod_\Sigma(F).$$

By definition of Mod we get:

$$Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot]) \subseteq Mod_\Sigma(F). \qquad (*)$$

Consider Proposition 21 and 22:

$Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\}$ and $Mod_\Sigma(F[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}$.

We can now substitute the respective model sets in (*) with Proposition 21 and 22 to obtain:

$$\{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\} \cap \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\} \subseteq \{\omega \mid \omega \models_\Sigma F\}$$

Using set theory we can express our equation like so:

$$\underbrace{\{\omega \mid \omega[a \mapsto 1] \models_\Sigma F, \omega[a \mapsto 0] \models_\Sigma F\}}_{Set_1} \subseteq \underbrace{\{\omega \mid \omega \models_\Sigma F\}}_{Set_2}$$

Now, we have to show that $\omega \in Set_1$ implies $\omega \in Set_2$ or, equivalently, the contrapositive

$$\omega \notin Set_2 \text{ implies } \omega \notin Set_1.$$

For each $\omega \notin Set_2$, we have $\omega \not\models_\Sigma F$. Now considering Set_1, if ω does not satisfy F, then $\omega[a \mapsto 1]$ or $\omega[a \mapsto 0]$ might still satisfy F in the following two cases:

(1) $\omega \not\models_\Sigma F$ and $\omega_1 = \omega[a \mapsto 1]$, $\omega_1 \models_\Sigma F$ where $\omega_1(a) = 1$

(2) $\omega \not\models_\Sigma F$ and $\omega_0 = \omega[a \mapsto 0]$, $\omega_0 \models_\Sigma F$ where $\omega_0(a) = 0$

Observe that while ω does not satisfy F, the mappings ω_0 and ω_1 do. The only alteration is the a-assignment. Thus, we conclude that the criterion for satisfying F is determined solely by the assignment of a. For an element ω to be in Set_1, it must be that both $\omega[a \mapsto 0]$ and $\omega[a \mapsto 1]$ satisfy F simultaneously. However, given the opposing assignments of a, it is impossible for both $\omega[a \mapsto 0] \models_\Sigma F$ and $\omega[a \mapsto 1] \models_\Sigma F$ to be true concurrently. Consequently, ω cannot belong to Set_1, thus the implication holds. □

Proposition 26. *For a formula $F \in \mathcal{L}_\Sigma$ and a variable $a \in \Sigma$. The following holds:*

$$F \models_\Sigma Forget(F, a).$$

Proof. Let $\omega \in \Omega_\Sigma$, $F \in \mathcal{L}_\Sigma$ be a formula and let $a \in \Sigma$ be a variable. We want to proof that the following holds:

$$F \models_\Sigma Forget(F, a).$$

By definition of $Forget$ we have:

$$F \models_\Sigma F[a/\top] \vee F[a/\bot].$$

Now we can obtain the model set relations such that:

$$Mod_\Sigma(F) \subseteq Mod_\Sigma(F[a/\top] \vee F[a/\bot]).$$

By definition of Mod we get:

$$Mod_\Sigma(F) \subseteq Mod_\Sigma(F[a/\top]) \cup Mod_\Sigma(F[a/\bot]). \tag{*}$$

Consider Proposition 21 and 22:

$$Mod_\Sigma(F[a/\top]) = \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\} \quad \text{and} \quad Mod_\Sigma(F[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}.$$

We can now substitute the respective model sets in (*) with Proposition 21 and 22 to obtain:

$$\{\omega \mid \omega \models_\Sigma F\} \subseteq \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\} \cap \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\}$$

Using set theory we can express our equation like so:

$$\underbrace{\{\omega \mid \omega \models_\Sigma F\}}_{Set_1} \subseteq \underbrace{\{\omega \mid \omega[a \mapsto 1] \models_\Sigma F, \omega[a \mapsto 0] \models_\Sigma F\}}_{Set_2}$$

Now, we have to show that $\omega \in Set_1$ implies $\omega \in Set_2$.

For each $\omega \in Set_1$, we have $\omega \models_\Sigma F$. Now considering Set_2, if ω does satisfy F, then $\omega[a \mapsto 1]$ or $\omega[a \mapsto 0]$ might still not satisfy F in the following two cases:

(1) $\omega \models_\Sigma F$ and $\omega_1 = \omega[a \mapsto 1]$, $\omega_1 \not\models_\Sigma F$ where $\omega_1(a) = 1$

(2) $\omega \models_\Sigma F$ and $\omega_0 = \omega[a \mapsto 0]$, $\omega_0 \not\models_\Sigma F$ where $\omega_0(a) = 0$

Observe that ω would satisfy F, but the mappings ω_0 and ω_1 do not. The only change is the a-assignment. Therefore, we conclude that the criterion for **not** satisfying F is determined solely by the assignment of a. For an element ω to be in Set_2, it must hold that either $\omega[a \mapsto 0]$ or $\omega[a \mapsto 1]$ satisfy F. Due to the opposing assignments of a, there is always either $\omega[a \mapsto 0] \models_\Sigma F$ or $\omega[a \mapsto 1] \models_\Sigma F$ true. Consequently, ω belongs to Set_2, and thereby the implication holds. □

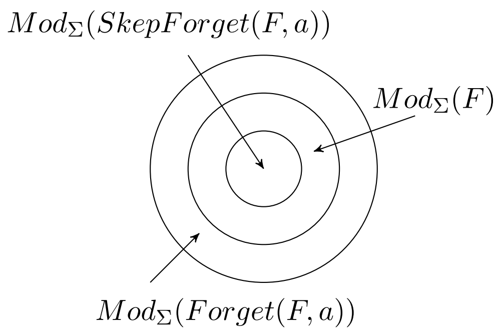


Figure 16

Figure 16 illustrates following Proposition 27, showcasing the relationship between models in the context discussed.

Proposition 27. *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$. The following relations hold true.*

$$Mod_\Sigma(SkepForget(F, a)) \subseteq Mod_\Sigma(F) \subseteq Mod_\Sigma(Forget(F, a))$$

and thus

$$SkepForget(F, a) \models_\Sigma F \models_\Sigma Forget(F, a).$$

Proof. From proposition 26 we have $F \models_\Sigma Forget(F, a)$ and obtain:

$$Mod(F) \subseteq Mod(Forget(F, a))$$

From proposition 25 we have $SkepForget(F, a) \models_\Sigma F$ and obtain:

$$Mod(SkepForget(F, a)) \subseteq Mod(F)$$

Hence we have this relations:

$$Mod_\Sigma(SkepForget(F, a)) \subseteq Mod_\Sigma(F) \subseteq Mod_\Sigma(Forget(F, a))$$

We can conclude:

$$SkepForget(F, a) \models_\Sigma F \models_\Sigma Forget(F, a).$$ □

4.5 Switch Intuition

We will briefly examine the $Switch$ operation, as defined in preliminaries, to develop a deeper intuition and understanding of its connection to the forget relation. Let us now consider this operation as if it were a mapping from all interpretations $\omega \in \Omega_\Sigma$ to their respective results. Let's consider three formulae $F_1 = (\neg a)$, $F_2 = (a \wedge b)$ and $F_3 = (a \vee b)$ in $\mathcal{L}_\Sigma$.

a	b	F_1	F_2	F_3
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

$Switch(\omega, a)$ and $\omega \in \Omega_\Sigma$ yields:

a	b	F_1	F_2	F_3
1	0	0	0	1
1	1	0	1	1
0	0	1	0	0
0	1	1	0	1

Figure 17: switch mapping

Figure 17 illustrates mirroring-like behaviour across the "truth value" axis of the variable a. Intuitivly, the term "Switch" suggests an operation where we simply inverse the truth value of a. Revisiting the forget relation discussed earlier, it involves a single variable that differs between two interpretations while all other variables remain consistent. This aligns precisely with what the Switch operation accomplishes, where we can describe the forget relation as

$$\omega_1 \sim_a \omega_2 = \omega_1 \sim_a Switch(\omega_1, a).$$

We denote the set of interpretations satisfying F under $Switch(F, a)$ as

$$SwMod(F, a) = \{\omega \mid Switch(\omega, a) \models F\}$$

and which we will refer to as "switched models".

4.6 Model Set Dynamics

We aim to explore the interconnections among the model-sets of substitutions, formulae, and switched models. As previously established, $Mod_\Sigma(SkepForget(F, a))$ can be described as $\{\omega \mid \omega[a \mapsto 1] \models_\Sigma F, \omega[a \mapsto 0] \models_\Sigma F\}$. These mappings notably lead to contradictory assignments for a. Moreover, the $Switch$ operation similarly results in contradictory a-assignments, highlighting a correlation between these sets. By drawing this connection, we can characterize $SkepForget$ as follows:

$$Mod_\Sigma(SkepForget(F, a)) = \{\omega \mid \omega \models_\Sigma F \text{ and } Switch(\omega, a) \models_\Sigma F\}.$$

Let's begin by visually examining the relationships among these sets using set diagrams. We will consider three formulae again: a, $(a \wedge b)$, and $(a \vee b)$, along with their negations over a language $\mathcal{L}_\Sigma$. Each diagram will indicate the corresponding formula (and forget-formula) below in the left corner.

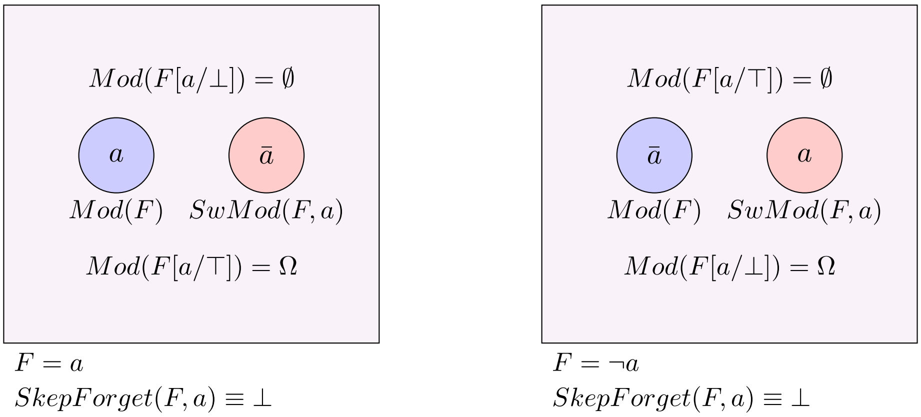


Figure 18: Diagrams of $F = a$ and $F = \neg a$

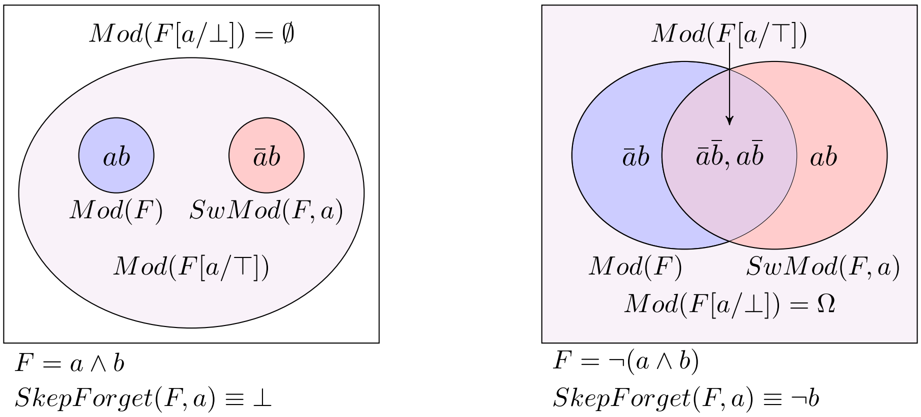
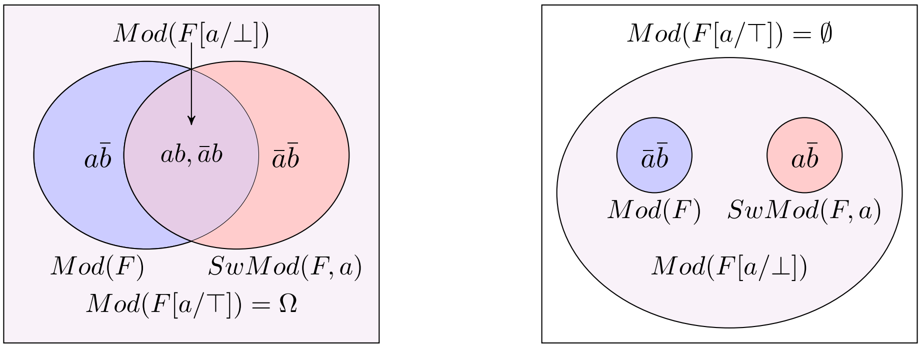


Figure 19: Diagrams of $F = a \wedge b$ and $F = \neg(a \wedge b)$



$F = a \vee b$
$SkepForget(F, a) \equiv b$

$F = \neg(a \vee b)$
$SkepForget(F, a) \equiv \bot$

Figure 20: Diagrams of $F = a \vee b$ and $F = \neg(a \vee b)$

In extreme cases, consider for instance $F = (\neg a \vee (a \vee b))$, where all mentioned sets become Ω. Conversely, for $F = (\neg a \wedge (a \wedge b))$, we have $Mod_\Sigma(F[a/\bot]) = Mod_\Sigma(F[a/\top]) = Mod_\Sigma(F) = SwMod_\Sigma(F, a) = \emptyset$.

In the Diagrams 18, 19, and 20, we visually observe a behaviour that we anticipated earlier regarding the exclusion of models where a is relevant to satisfy F. For an intersection set S_{mod} between $Mod_\Sigma(F)$ and $SwMod_\Sigma(F, a)$, precisely those models are included from which F can be derived, where the assignment of a is irrelevant to satisfy F. The set S_{mod} is either the empty set, or we find pairs of interpretations that satisfy the forget relation $\omega_1 \sim_a \omega_2$. The same holds for an intersection set $S_{sub} = Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot])$. S_{sub} is the empty set unless there are interpretations $\omega \in S_{sub}$ that derive F regardless of the assignment of a. The examples of the diagrams are divided such that on the left side, a is only positively included in the formulae, while on the right side, a is only negatively (i.e., $\neg a$) included in F. The following four cases, illustrate the diagrams formally:

(1) For $a \in PosAtom(F) \wedge a \notin NegAtom(F)$ we have:

a) $Mod_\Sigma(F[a/\bot]) = Mod_\Sigma(F) \cap SwMod_\Sigma(F, a)$

b) $Mod_\Sigma(F[a/\top]) = Mod_\Sigma(F) \cup SwMod_\Sigma(F, a)$

(2) For $a \notin PosAtom(F) \wedge a \in NegAtom(F)$ we have:

a) $Mod_\Sigma(F[a/\bot]) = Mod_\Sigma(F) \cup SwMod_\Sigma(F, a)$

b) $Mod_\Sigma(F[a/\top]) = Mod_\Sigma(F) \cap SwMod_\Sigma(F, a)$

(3) For $a \in PosAtom(F) \wedge a \in NegAtom(F)$ and

(4) For $a \notin PosAtom(F) \wedge a \notin NegAtom(F)$ we have:

a) $Mod_\Sigma(F[a/\top]) = Mod_\Sigma(F[a/\bot]) = Mod_\Sigma(F) = SwMod_\Sigma(F, a)$

For cases (3) and (4), all sets are identical, consistent with our observations in the extreme cases. In cases (1) and (2), we observe their duality in terms of set operations. With this improved understanding of these relationships, we can now characterize our skeptical notion of variable forgetting as follows:

Proposition 28. *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$, we have:*

$$\begin{aligned} Mod_\Sigma(SkepForget(F,a)) &= \{\omega \mid \omega \models_\Sigma F \, and \, Switch(\omega,a) \models_\Sigma F\} \\ &= Mod_\Sigma(F) \cap \{\omega \mid \ Switch(\omega,a) \models_\Sigma F\} \\ &= Mod_\Sigma(F) \cap SwMod_\Sigma(F,a) \end{aligned}$$

Proof. We need to show that

$$Mod_\Sigma(SkepForget(F,a)) = Mod_\Sigma(F) \cap SwMod_\Sigma(F,a)$$

holds.

The following equivalence is due to Proposition 21 and 22:

$$Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot]) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\} \cap \{\omega \mid \omega[a \mapsto 1] \models_\Sigma F\} \quad \text{(*)}$$

By Proposition 24, we have the following set equation :

$$\begin{aligned} Mod_\Sigma(SkepForget(F,a)) &= Mod_\Sigma(F[a/\top] \wedge F[a/\bot]) \\ &= Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot]) \end{aligned}$$

With (*) we can substitue our equation such that:

$$\begin{aligned} Mod_\Sigma(SkepForget(F,a)) &= Mod_\Sigma(F[a/\top]) \cap Mod_\Sigma(F[a/\bot]) \\ &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F\} \cap \{\omega|\omega[a \mapsto 1] \models_\Sigma F\} \end{aligned}$$

Applying set-theoretic transformations, we can express this as:

$$Mod_\Sigma(SkepForget(F,a)) = \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F, \omega[a \mapsto 1] \models_\Sigma F\}$$

Observe that for each interpretation ω, the operation $Switch(\omega,a)$ results in an interpretation with opposite assignments for a:

$$(1) \;\; \omega = \omega[a \mapsto 1], \text{ then } Switch(\omega,a) = \omega[a \mapsto 0]$$

$$(2) \;\; \omega = \omega[a \mapsto 0], \text{ then } Switch(\omega,a) = \omega[a \mapsto 1].$$

Hence, we have the equivalence:

$$\{\omega \mid \omega[a \mapsto 0] \models_\Sigma F, \omega[a \mapsto 1] \models_\Sigma F\} = \{\omega \mid \omega \models_\Sigma F, Switch(\omega, a) \models_\Sigma F\} \qquad (**)$$

Using (**) with the previous equation, we obtain the following substitution:

$$\begin{aligned} Mod_\Sigma(SkepForget(F, a)) &= \{\omega \mid \omega[a \mapsto 0] \models_\Sigma F, \omega[a \mapsto 1] \models_\Sigma F\} \\ &= \{\omega \mid \omega \models_\Sigma F, Switch(\omega, a) \models_\Sigma F\} \end{aligned}$$

From here we can easily show that the proposition holds true by the following transformation:

$$\begin{aligned} Mod_\Sigma(SkepForget(F, a)) &= \{\omega \mid \omega \models_\Sigma F, Switch(\omega, a) \models_\Sigma F\} \\ &= \{\omega \mid \omega \models_\Sigma F\} \cap \{Switch(\omega, a) \models_\Sigma F\} \\ &= Mod_\Sigma(F) \cap SwMod_\Sigma(F, a) \end{aligned}$$

□

In [LLM03] the authors state the following characterization of $Forget$.

Proposition 29 ([LLM03]). *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$, we have:*

$$\begin{aligned} Mod_\Sigma(Forget(F, a)) &= \{\omega \mid \omega \models_\Sigma F \text{ or } Switch(\omega, a) \models_\Sigma F\} \\ &= Mod_\Sigma(F) \cup \{\omega \mid \; Switch(\omega, a) \models_\Sigma F\} \\ &= Mod_\Sigma(F) \cup SwMod_\Sigma(F, a) \end{aligned}$$

5 Related Topics

In the context of variable forgetting, there are several related topics that have been the subject of research. We will briefly introduce two of these topics. First, there is the concept of marginalization, which emphasizes semantic aspects, signature reduction and builds upon variable forgetting. Second, we have the concept of independence, which examines the dependency of a propositional variable on a given formula.

5.1 Marginalisation

In [SBKI24], the authors have demonstrated that variable forgetting can be understood as "marginalization", from a semantic perspective. This aligns with the reduction discussed earlier, where a signature is simplified to a subset of its own, thereby excluding certain variables from consideration.

Definition 30 (Model Marginalization, $ModMg_\Sigma(\omega, \Gamma)$, $ModMg_\Sigma(M, \Gamma)$)**.** *Let* $\omega \in \Omega_\Sigma$*,* $M \subseteq \Omega_\Sigma$*, and let* $\Gamma \subseteq \Sigma$*. We define* $ModMg_\Sigma(\omega, \Gamma) = \omega^\Gamma$ *as the (model) marginalization of* ω *from* Σ *to* Γ*. The element-wise marginalization of all* $\omega \in M$ *from* Σ *to* Γ *is called the (model) marginalization of* M *from* Σ *to* Γ*, denoted by* $ModMg_\Sigma(M, \Gamma) = \{ModMg_\Sigma(\omega, \Gamma) \mid \omega \in M\}$*.*

Definition 31 (Syntactic Marginalization)**.** *Let* $F \in \mathcal{L}_\Sigma$ *and let* $\Gamma \subseteq \Sigma$*. The syntactic marginalization of* F *(from* Σ*) to* Γ*, written* $SynMg_\Sigma(F, \Gamma)$*, is* $Forget(F, \Sigma \setminus \Gamma)$

Theorem 32 ([SBKI24])**.** *For all* $F \in \mathcal{L}_\Sigma$ *and* $\Gamma \subseteq \Sigma$*, the following holds:*

$$\begin{aligned} \mathit{ModMg}_\Sigma(\mathit{Mod}_\Sigma(F), \Gamma) &= \mathit{Mod}_\Gamma(\mathit{SynMg}_\Sigma(F, \Gamma)) \\ &= \mathit{Mod}_\Gamma(\mathit{Forget}(F, \Sigma \setminus \Gamma)) \end{aligned}$$

To explore our topic of skeptical forgetting in the context of marginalization, we introduce the following definition:

Definition 33 (Skeptical Syntactic Marginalization)**.** *Let* $F \in \mathcal{L}_\Sigma$ *and let* $\Gamma \subseteq \Sigma$*. The skeptical syntactic marginalization of* F *(from* Σ*) to* Γ*, written* $SkepSynMg_\Sigma(F, \Gamma)$*, is* $SkepForget(F, \Sigma \setminus \Gamma)$*.*

The authors in [SBKI24] demonstrate that the models of classical variable forgetting for a formula $F \in \mathcal{L}_\Sigma$ and a variable $a \in \Sigma$ are equivalent to the set of interpretations obtained through model marginalization from Σ to Γ where $\Gamma = \Sigma \setminus \{a\}$, formally expressed as:

$$ModMg_\Sigma(Mod_\Sigma(F), \Sigma \setminus \{a\}) = Mod_\Gamma(Forget(F, a)).$$

Example 34. *Consider the formula* $F = a \vee (b \wedge c)$ *over the signature* $\Sigma = \{a, b, c\}$*. Let* $\Gamma = \Sigma \setminus \{a\} = \{b, c\}$ *be a subsignature of* Σ*. First, we examine the models of* F*:*

$$\mathit{Mod}_\Sigma(F) = \{\{\bar{a}bc\}, \{a\bar{b}\bar{c}\}, \{a\bar{b}c\}, \{ab\bar{c}\}, \{abc\}\}.$$

Now, consider the marginalization of the models of F:

$$ModMg_{\Sigma}(Mod_{\Sigma}(F), \Gamma) = \{\{bc\}, \{\bar{b}\bar{c}\}, \{\bar{b}c\}, \{b\bar{c}\}\} = \Omega_{\Gamma}.$$

For $SynMg_{\Sigma}(F, \Gamma)$ = Forget(F, a), we obtain the tautology. Consequently, the models are Mod$_{\Sigma}$(Forget$(F, a)) =$ Mod$_{\Sigma}(\top) = \Omega_{\Sigma}$ and Mod$_{\Gamma}$(Forget$(F, a)) =$ Mod$_{\Gamma}(\top) = \Omega_{\Gamma}$.

Example 35. *Next, we consider Skeptical Syntactic Marginalization. Again, let $F = a \vee (b \wedge c)$ over the signature $\Sigma = \{a, b, c\}$. For $SkepSynMg_{\Sigma}(F, \Gamma)$ = SkepForget(F, a), we derive $b \wedge c$. Revisiting the models of F from the previous example:*

$$Mod_{\Sigma}(F) = \{\{\bar{a}bc\}, \{a\bar{b}\bar{c}\}, \{a\bar{b}c\}, \{ab\bar{c}\}, \{abc\}\}.$$

We convert the set of $Mod_{\Sigma}(F)$ into a multiset of sets such that:

$$\{\{\{\bar{a}bc\}, \{a\bar{b}\bar{c}\}, \{a\bar{b}c\}, \{ab\bar{c}\}, \{abc\}\}\}$$

Notice that removing the variable a results in duplicate remainders such as $\{bc\}$:

$$\{\{\{bc\}, \{\bar{b}\bar{c}\}, \{\bar{b}c\}, \{b\bar{c}\}, \{bc\}\}\}.$$

The model set $Mod_{\Gamma}(SkepForget(F, a)) = Mod_{\Gamma}(b \wedge c) = \{bc\}$ corresponds precisely to the set of duplicate remainders.

Note that while models of F are elements of Ω_{Σ}, the models of $SkepForget(F, a)$ are elements of Ω_{Γ}. In that regard, we can consider them as marginalized. Semantically, Skeptical Syntactic Marginalization results in a distinct set of models compared to classical Syntactic Marginalization and can be viewed as an alternative syntactic realization of model marginalization.

Proposition 36. *Let $F \in \mathcal{L}_{\Sigma}$, $a \in \Sigma$ and $\Gamma \subseteq \Sigma$ with $\Gamma = \Sigma \setminus \{a\}$, then*

$$Mod_{\Gamma}(SkepForget(F, a)) \subseteq ModMg_{\Sigma}(Mod_{\Sigma}(F), \Gamma)$$

Proof. From Propositon 28 and 29 we have

$$Mod_{\Sigma}(SkepForget(F, a)) = \{\omega_1 \mid \omega_1 \models_{\Sigma} F\} \cap \{\omega_1 \mid Switch(\omega_1, a) \models_{\Sigma} F\}, \quad (1)$$

$$Mod_{\Sigma}(Forget(F, a)) = \{\omega_1 \mid \omega_1 \models_{\Sigma} F\} \cup \{\omega_1 \mid Switch(\omega_1, a) \models_{\Sigma} F\} \quad (2)$$

Proposition 27 states the following holds:

$$Mod_{\Sigma}(SkepForget(F, a)) \subseteq Mod_{\Sigma}(Forget(F, a)). \quad (*)$$

We want to show that:

$$Mod_{\Gamma}(SkepForget(F, a)) \subseteq Mod_{\Gamma}(Forget(F, a)).$$

We subsitute (*) with (1) and (2) and obtain:

$$\{\omega_1 \mid \omega_1 \models_{\Sigma} F, \omega_2 = Switch(\omega_1, a) \models_{\Sigma} F\} \subseteq \{\omega_1 \mid \omega_1 \models_{\Sigma} F \text{ or } \omega_2 = Switch(\omega_1, a) \models_{\Sigma} F\}$$

Now, we transition from Σ to Γ. Observe that on both sides of the equation, the forget relation $\omega_1 \sim_a \omega_2$, as defined in Definition 23, holds. Therefore, we have $\omega_1^\Gamma = \omega_2^\Gamma = \omega_f$, where $\omega_f \in \Omega_\Gamma$.

To prove that for all ω_f in $Mod_\Gamma(SkepForget(F, a))$ implies ω_f is element of $Mod_\Gamma(Forget(F, a))$ holds, observe that:

- If $\omega_1 \models_\Sigma F$ and $\omega_2 \models_\Sigma F$ then $\omega_f \in Mod_\Gamma(SkepForget(F, a))$
- If $\omega_1 \models_\Sigma F$ or $\omega_2 \models_\Sigma F$ then $\omega_f \in Mod_\Gamma(Forget(F, a))$

Hence we conclude that every $\omega_f \in Mod_\Gamma(SkepForget(F, a))$ must also be an element of $Mod_\Gamma(Forget(F, a))$. Therefore:

$$\begin{aligned} Mod_\Gamma(SkepForget(F, a)) &\subseteq Mod_\Gamma(Forget(F, a)) \\ &\subseteq Mod_\Gamma(Forget(F, \Sigma \setminus \Gamma)) \end{aligned} \tag{**}$$

We subsitute (**) with Theorem 32:

$$\text{ModMg}_\Sigma(\text{Mod}_\Sigma(F), \Gamma) = \text{Mod}_\Gamma(\text{Forget}(F, \Sigma \setminus \Gamma)),$$

and obtain:

$$Mod_\Gamma(SkepForget(F, a)) \subseteq ModMg_\Sigma(Mod_\Sigma(F), \Gamma). \quad \square$$

5.2 Variable Inpependence

We have seen that pairs of interpretations satisfying the forget relation regarding a variable a and our formula F are always models of $SkepForget(F, a)$. For these pairs, the variable a is essentially irrelevant or, one could say, independent. There is relevant research on the topic of variable independence closely linked to the concept of forgetting. Therefore, we would like to delve into this subject briefly.

In [LLM03], the authors define propositional variable independence both syntactically and semantically as follows:

Definition 37 (Syntactical Inpependence)**.** *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$.*

- *F is variable dependent on a if $a \in Sig(F)$ holds.*
- *F is variable dependent on $V \subseteq \Sigma$, if there is a variable $a \in V$ such that F is variable dependent on a.*

Definition 38 (Semantical Independence)**.** *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$.*

- *F is variable independent from a denoted $a \not\mapsto^+_- F$ if there exists a formula ϕ such that $\phi \equiv F$ and ϕ is syntactically var independent from a. Otherwise its dependent $a \mapsto^+_- F$. We denote $DepVar(F)$ the set of all variables a such that $a \mapsto^+_- F$.*

- *F is variable independent from $V \subseteq \Sigma$, denoted $V \not\mapsto^{+}_{-} F$, if $V \cap DepVar(F) = \emptyset$ holds. Otherwise F is variable dependent on V, denoted $V \mapsto^{+}_{-} F$.*

Example 39. *Consider a sentence $F = (a \wedge b) \wedge (a \vee c)$. Then we have a sentence $\phi = (a \wedge b)$ that is equal to F. Therefore $DepVar(F) = \{a, b\}$ and $c \not\mapsto^{+}_{-} F$, means F is var-independent from c.*

The authors of [LLM03] also highlight that:

$$a \not\mapsto^{+}_{-} F \text{ if } (\forall \omega \in \Omega,\ \omega \models_{\Sigma} F \iff Switch(\omega, a) \models_{\Sigma} F) \text{ holds .}$$

This reminds us of our semantic characterization of

$$Mod_{\Sigma}(SkepForget(F, a)) = \{\omega \mid \omega \models_{\Sigma} F\, and\, Switch(\omega, a) \models_{\Sigma} F\}.$$

We also know that for

$$Mod_{\Sigma}(Forget(F, a)) = \{\omega \mid \omega \models_{\Sigma} F\, or\, Switch(\omega, a) \models_{\Sigma} F\}.$$

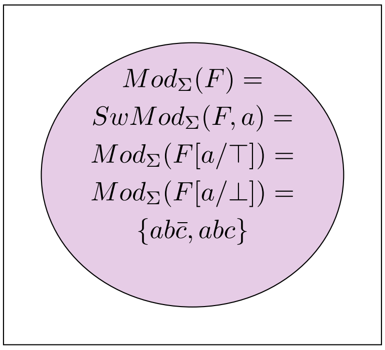
However, it's essential to distinguish between a biconditional and a conjunction (or disjunction). It's important to note that forgetting is an operation that constructs sets, whereas variable independence is a property that can be either true or false for a given set. Variable independence is a result of variable forgetting, as indicated the statements

$$a \not\mapsto^{+}_{-} Forget(F, a) \text{ and } a \not\mapsto^{+}_{-} SkepForget(F, a).$$

Lets try to forget an independent variable:

$$\begin{aligned} SkepForget(F, c) &= F[c/\top] \wedge F[c/\bot] \\ &\equiv_{\Sigma} (((a \wedge b) \wedge (a \vee \top)) \wedge ((a \wedge b) \wedge (a \vee \bot))) \\ &\equiv_{\Sigma} (((a \wedge b) \wedge \top) \wedge ((a \wedge b) \wedge a)) \\ &\equiv_{\Sigma} ((a \wedge b) \wedge ((a \wedge b) \wedge a)) \\ &\equiv_{\Sigma} ((a \wedge b) \wedge ((a \wedge b))) \\ &\equiv_{\Sigma} (a \wedge b) \end{aligned}$$

For $Forget(F, c) = (a \wedge b)$, the result is equal as well as it is equal to ϕ. This observation suggests that if $a \notin DepVar(F)$ then $Forget(F, a) \equiv_{\Sigma} SkepForget(F, a)$? Lets consider another example with a disjunction, $G = (a \vee b) \wedge (a \vee b \vee c)$ is equal to $\gamma = (a \vee b)$. We have same $DepVar(G) = \{a, b\}$. Then we get $Forget(G, c) \equiv_{\Sigma} SkepForget(G, c) \equiv_{\Sigma} (a \vee b) \equiv_{\Sigma} \gamma$. This strengthens our suspicion.



$F = (a \wedge b) \wedge (a \vee c))$
$SkepForget(F, c) \equiv_\Sigma a \wedge b$

Figure 21: Forgetting an independent Variable c.

Proposition 40. *Let $F \in \mathcal{L}_\Sigma$ and let $a \in \Sigma$, then*

$$a \notin DepVar(F) \text{ if and only if } Forget(F, a) \equiv_\Sigma SkepForget(F, a)$$

or

$$a \not\mapsto^{+}_{-} F \text{ if and only if } Forget(F, a) \equiv_\Sigma SkepForget(F, a).$$

Proof. If $a \notin DepVar(F)$ or $a \not\mapsto^{+}_{-} F$ if and only if there exists a formula ϕ such that $\phi \equiv_\Sigma F$ and ϕ is syntactically var independent from a and therefore $a \notin Sig(\phi)$. Then

$$Forget(\phi, a) \equiv_\Sigma \phi \text{ and } SkepForget(\phi, a) \equiv_\Sigma \phi,$$

hence

$$Forget(\phi, a) \equiv_\Sigma SkepForget(\phi, a). \quad \square$$

Only when we forget a dependent variable we possibly yield a drastic results as demonstrated in the beginning.

6 Conclusion

We have shown that while $SkepForget$ is compatible under conjunction, it is not compatible under disjunction or negation. Furthermore, $SkepForget$ is related to $Forget$ in a de Morgan relationship. On a syntactic level, we have identified equivalences to simplify forget sentences under certain sentence forms, making it easier to recognize whether and which drastic results ($\top$, $\bot$) are to be expected. Furthermore, we have formalized a method using clause sets to ascertain the conditions under which SkepForget yields drastic outcomes.

On a semantic level, we have analyzed step by step the different ways both operations resolve the discussed ambiguity and dissected why tautologies and contradictions arise when forgetting is applied. When the decision of whether an interpretation ω is a model of F depends on the "variable to be forgotten", $SkepForget$ always excludes both interpretations of a forget relation, whereas classical forgetting would include both. In other words, in a forget relation, both interpretations of the pair must satisfy F for $SkepForget$ to be satisfied. Consequently, we obtain fewer models for $SkepForget(F, a)$. Thus, it can indeed be said that skeptical forgetting minimizes truth values. We have seen that $SkepForget$ can be used as an alternative syntactic realization of model marginalization. When the variable to be forgotten is independent, $SkepForget$ and $Forget$ are equivalent. Under this condition, we also do not obtain drastic results. In summary, it could be argued that $Forget$ reacts more affirmatively when there is uncertainty. Conversely, $SkepForget$ tends to make fewer claims and takes a more cautious stance.

The meaningfulness of this technique is certainly debatable. Let's focus on what doesn't seem particularly sensible, namely the drastic results. If we imagine forgetting as something known losing its contour, merging into "everything else", or into "the void", then equating the variable with a constant, namely a tautology or contradiction, makes some sense. However, if we consider the result of forgetting as the complete disappearance of a mental entity, simply replacing a symbol might not suffice. In our case, there still remains a reference to something—the logical constant, an entity being "spoken about". We do forget, but we shift the reference and point to something else, thereby introducing something new. This referenced entity, in the form of a tautology or contradiction, introduces misinformation or continues to influence our forget statement; one could say it distorts the statement in an unexpected way.

However, the concept of relevance is highly intriguing. Framing forgetting as a form of relevance-loss leads us to consider its inverse: how relevance can be attributed? This suggests a reverse function. One might envision this process expanding a signature by introducing new variables from "outside" or differentiating existing atoms, literaly splitting atoms. Such perspectives offer exciting avenues for future research, in our view.

References

[Boo21] George Boole. An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities (1854). 2021.

[EKI19] Thomas Eiter and Gabriele Kern-Isberner. A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI-Künstliche Intelligenz*, 33:9–33, 2019.

[LLM03] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence-formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.

[LR94] Fangzhen Lin and Ray Reiter. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, pages 154–159, 1994.

[SBKI24] Kai Sauerwald, Christoph Beierle, and Gabriele Kern-Isberner. *Propositional Variable Forgetting and Marginalization: Semantically, Two Sides of the Same Coin*, pages 144–162. 03 2024.

[VDHLM09] Hans Van Ditmarsch, Andreas Herzig, Jérôme Lang, and Pierre Marquis. Introspective forgetting. *Synthese*, 169:405–423, 2009.