CAISL: Simplification Logic for Conditional Attribute Implications

Estrella Rodríguez-Lorenzo[1], Pablo Cordero[1], Manuel Enciso[1], Rokia Missaoui[2], Ángel Mora[1]

[1]Universidad de Málaga, Andalucía Tech, Spain,
e-mail: {estrellarodlor,amora}@ctima.uma.es, {pcordero,enciso}@uma.es
[2] Université du Québec en Outaouais, Canada,
e-mail: {rokia.missaoui}@uqo.ca

Abstract. In this work, we present a sound and complete axiomatic system for conditional attribute implications (*CAI*s) in Triadic Concept Analysis (TCA). Our approach is strongly based on the Simplification paradigm, allowing a more suitable approach to automated reasoning than those based on Armstrong's Axioms. We also present an automated method to prove the derivability of a *CAI* from a set of *CAI*s.

1 Introduction

Implications in FCA represent associations between two attribute sets, denoted by $X \to Y$, and capture an important knowledge hidden in the input data. They also allow an alternate representation of the concept lattice and open the door to their automated management through logic. Such a management is used, for instance, to characterize representations of the whole knowledge by means of the notion of implicational systems. There exist different axiomatic systems in FCA, the first one is called *Armstrong's Axioms* [1], but later, other equivalent logics emerged [4, 5, 9].

The first study on triadic implications has been investigated by Biedermann [2] and then an extended work has been poposed by Ganter and Obiedkov [6]. In addition to a formal definition of implications and their language, we believe that the introduction of a sound and complete inference system is needed to reason about such implications and determine whether a given implication can be derived from an implication basis. Soundness ensures that implications derived by using the axiomatic system are valid in the formal context and completeness guarantees that all valid implications can be derived from the implicational system.

As far as we know, there does not exist an axiomatic system in Triadic Concept Analysis. The main goal of this paper is then to define a new axiomatic system based on Simplification Logic [4] as an alternate view of the inference system recently developed by the authors [12]. This new way also allows an efficient automated reasoning, commonly called the implication problem, to determine if a conditional attribute implication (*CAI*) can be derived from a set of *CAI*s.

Given a set of dependencies Σ and a further dependency σ, the implication problem means that one would like to check whether σ holds in all datasets

satisfying Σ. This problem occurs in research areas such as database theory and knowledge reasoning, and its solution allows the search for associations in an interactive and exploratory way rather than an exhaustive manner. Using Armstrong's axioms, many polynomial time algorithms for implication problem decision have been defined and the closure of an attribute set has been exploited to solve it.

The remainder of this paper is organized as follows. In Section 2 we provide a background on TCA. Section 3 briefly presents a logic for conditional attribute implications called CAIL [12] while Section 4 describes a new axiomatic system called CAISL that is more suitable for solving the implication problem in the triadic framework. In Section 5 we establish equivalences derived from CAISL between sets of *CAI*sand show how we can syntactically transform and simplify a set of *CAI*s while preserving their semantics in the CAISL context. To check whether a *CAI* holds for a given set of *CAI*s, we propose and illustrate a new procedure in Section 6. Finally, Section 7 summarizes our contribution and presents further work.

2 Triadic concept analysis

As a natural extension to Formal Concept Analysis (FCA), theoretical foundations of Triadic Concept Analysis have been investigated by Lehmann and Wille [8] who were inspired by the philosophical framework of Charles S. Peirce [11] of three universal categories. The input is a formal triadic context describing objects in terms of attributes that hold under given conditions and the output is a concept trilattice that allows the generation of triadic association rules, including implications [2, 6, 7, 10].

Definition 1. *A triadic context* $\mathbb{K} = \langle G, M, B, I\rangle$ *consists of three sets: a set of objects (G), a set of attributes (M) and a set of conditions (B) together with a ternary relation* $I \subseteq G \times M \times B$*. A triple* (g, m, b) *in* I *means that object* g *possesses attribute* m *under condition* b*.*

Figure 1 shows a triadic context $\mathbb{K} := \langle G, M, B, I\rangle$, where $G = \{1, 2, 3, 4, 5\}$ is a set of customers, $M = \{\mathrm{P}, \mathrm{N}, \mathrm{R}, \mathrm{K}, \mathrm{S}\}$ a set of suppliers and $B = \{\mathrm{a}, \mathrm{b}, \mathrm{d}, \mathrm{e}\}$ represents a set of products. The ternary relation gives information about the customers and the suppliers from whom they buy products. For instance, Customer 1 buys from Supplier P products a, b and e.

$\mathbb{K}$	P	N	R	K	S
1	abe	abe	ad	ab	a
2	ae	bde	abe	ae	e
3	abe	e	ab	ab	a
4	abe	be	ab	ab	e
5	ae	ae	abe	abd	a

Fig. 1. A triadic context

The derivation operators in triadic concept analysis were introduced in [8]. If X_1, X_2 and X_3 are subsets of G, M and B respectively, then one can get:

$$X_1' = \{(a_j, a_k) \in M \times B \mid (a_i, a_j, a_k) \in I \text{ for all } a_i \in X_1\}.$$

$$(X_2, X_3)' = \{a_i \in G \mid (a_i, a_j, a_k) \in I \text{ for all } (a_j, a_k) \in X_2 \times X_3\}.$$

In a similar way, X_2', $(X_1, X_3)'$, X_3' and $(X_1, X_2)'$ can be defined. As shown in [13], the above family of operators, by setting a subset of objects, attributes or conditions (respectively) yields Galois connections. In this paper, we use the family of Galois connections associated with condition subsets. That is, given $\mathcal{C} \subseteq B$ we consider the Galois connection between the lattices $(2^M, \subseteq)$ and $(2^G, \subseteq)$ as the pair of mappings:

$$\begin{array}{rcl} (-, \mathcal{C})' : 2^G & \longrightarrow & 2^M \\ X_1 & \longmapsto & (X_1, \mathcal{C})' \end{array} \qquad \begin{array}{rcl} (-, \mathcal{C})' : 2^M & \longrightarrow & 2^G \\ X_2 & \longmapsto & (X_2, \mathcal{C})' \end{array}$$

Thus, for each $X_1 \subseteq G$ and $X_2 \subseteq M$, one has $X_2 \subseteq (X_1, \mathcal{C})'$ if and only if $X_1 \subseteq (X_2, \mathcal{C})'$.

In a similar way as in dyadic FCA, the composition of both derivation operators leads to the notion of triadic concept.

Definition 2. *A triadic concept of a triadic context is a triple (A_1, A_2, A_3) with $A_1 \subseteq G$, $A_2 \subseteq M$, $A_3 \subseteq B$ and $A_1 \times A_2 \times A_3 \subseteq I$ such that for $X_1 \subseteq G, X_2 \subseteq M$, and $X_3 \subseteq B$ with $X_1 \times X_2 \times X_3 \subseteq I$, the containments $A_1 \subseteq X_1, A_2 \subseteq X_2$, and $A_3 \subseteq X_3$ always lead to $(A_1, A_2, A_3) = (X_1, X_2, X_3)$. The subsets A_1, A_2 and A_3 are called the* extent*, the* intent *and the* modus *of the triadic concept (A_1, A_2, A_3) respectively.*

There are a few kinds of triadic implications with different semantics. Biedermann [3] defines a triadic implication to be an expression of the form: $(A \to B)_{\mathcal{C}}$ where A and B are attribute sets and $\mathcal{C}$ is a set of conditions. This implication is interpreted as: *If an object has all attributes from A under all conditions from $\mathcal{C}$, then it also has all attributes from B under all conditions from $\mathcal{C}$.* Its formal definition is the following:

Definition 3. *Let $\mathbb{K} = \langle G, M, B, I \rangle$ be a triadic context, $A, B \subseteq M$ and $\mathcal{C} \subseteq B$. The implication $(X \to Y)_{\mathcal{C}}$ holds in the context $\mathbb{K}$ iff $(X, \mathcal{C})' \subseteq (Y, \mathcal{C})'$.*

Ganter and Obiedkov [6] consider three kinds of triadic implications. We will describe and make use of the following one which is stronger than Biedermann's expression and has another notation: $X \xrightarrow{\mathcal{C}} Y$, where $X, Y \subseteq M$ and $\mathcal{C} \subseteq B$. Such implication is called *conditional attribute implication* (CAI) and is read as "X implies Y under all conditions in $\mathcal{C}$ or any subset of it".

Definition 4 (Conditional attribute implication). *Let $\mathbb{K} = \langle G, M, B, I \rangle$ be a triadic context, $X, Y \subseteq M$ and $\mathcal{C} \subseteq B$. The implication $X \xrightarrow{\mathcal{C}} Y$ holds in the context $\mathbb{K}$ when $(X, \{c\})' \subseteq (Y, \{c\})'$ for all $c \in \mathcal{C}$.*

Notice that *CAI*s preserve the dyadic implications that hold for each elementary condition in $\mathcal{C}$. The following proposition relates both notions of implications and also shows that Biedermann's definition is weaker than the *CAI* definition.

Proposition 1 ([6]). *Let $\mathbb{K} = \langle G, M, B, I\rangle$ be a triadic context, $X, Y \subseteq M$ and $\mathcal{C} \subseteq B$. Then $X \xrightarrow{\mathcal{C}} Y$ holds in $\mathbb{K}$ iff $(X \to Y)_{\mathcal{N}}$ also holds in $\mathbb{K}$ for all $\mathcal{N} \subseteq \mathcal{C}$.*

The following example illustrates the above proposition.

Example 1. Let $\mathbb{K}$ be the triadic formal context given in Figure 1.

i) The *CAI* $N \xrightarrow{ae} P$ holds in $\mathbb{K}$ since the following implications are satisfied: $(N \to P)_a, (N \to P)_e, (N \to P)_{ae}$.
ii) The Biedermann's implication $(N \to P)_{abe}$ is satisfied but the *CAI* $N \xrightarrow{abe} P$ does not hold because, for instance, $(N \to P)_b$ is not satisfied.

Our objective in this paper is to provide inference mechanisms for a set of *CAI*s.

To that end, a sound and complete axiomatic system is needed. As mentioned earlier, we have introduced in [12] a novel logic for computing *CAI*s and reasoning about them. This logic is briefly presented in the following section.

3 CAIL: Conditional Attribute Implication Logic

In this section, we describe CAIL, a logic for reasoning about *CAI*s in the framework of TCA [12]. This logic is presented in a classical style by considering three pillars: the language, the semantics and the inference system.

Language: As it has been outlined, we use the following language: given an attribute set Ω and a set of conditions Γ, the set of well-formed formulas (hereinafter, formulas or implications) is $\mathcal{L}_{\Omega,\Gamma} = \{A \xrightarrow{\mathcal{C}} B \mid A, B \subseteq \Omega, \mathcal{C} \subseteq \Gamma\}$.

In the sequel we use X, Y, Z, W to mean subsets of attributes ($X, Y, Z, W \subseteq \Omega$) and $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2$ for subsets of conditions ($\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2 \subseteq \Gamma$). For the sake of readability of formulas, we omit the brackets and commas (e.g. *abc* denotes the set $\{a, b, c\}$) and, as usual, the union is denoted by set juxtaposition (e.g. XY denotes $X \cup Y$).

Semantics: Based on Definition 4, the semantics is introduced by means of the notions of interpretation and model. From a language $\mathcal{L}_{\Omega,\Gamma}$, an interpretation is a triadic context $\mathbb{K} = \langle G, M, B, I\rangle$ such that $M = \Omega$ and $B = \Gamma$. A model for a formula $X \xrightarrow{\mathcal{C}} Y \in \mathcal{L}_{\Omega,\Gamma}$ is an interpretation that satisfies $X \xrightarrow{\mathcal{C}} Y$ in $\mathbb{K}$. In this case, we write $\mathbb{K} \models X \xrightarrow{\mathcal{C}} Y$.

As usual, for $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$, an interpretation $\mathbb{K}$ is a model for Σ (briefly, $\mathbb{K} \models \Sigma$) if $\mathbb{K} \models X \xrightarrow{\mathcal{C}} Y$ for each $X \xrightarrow{\mathcal{C}} Y \in \Sigma$. Similarly, $\Sigma \models X \xrightarrow{\mathcal{C}} Y$ states that $X \xrightarrow{\mathcal{C}} Y$ is a semantic consequence of Σ, i.e. every model for Σ is also a model for $X \xrightarrow{\mathcal{C}} Y$.

Syntactic inference: The syntactic derivation in CAIL is denoted by the symbol $\vdash_{\mathscr{C}}$ and covers two axiom schemes and four inference rules.

Definition 5. *The CAIL axiomatic system consists of the following rules:*

`[Non-constraint]` $\vdash_{\mathscr{C}} \emptyset \xrightarrow{\emptyset} \Omega$.
`[Inclusion]` $\vdash_{\mathscr{C}} XY \xrightarrow{\Gamma} X$.
`[Augmentation]` $X \xrightarrow{\mathcal{C}} Y \vdash_{\mathscr{C}} XZ \xrightarrow{\mathcal{C}} YZ$.
`[Transitivity]` $\{X \xrightarrow{\mathcal{C}_1} Y, Y \xrightarrow{\mathcal{C}_2} Z\} \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} Z$.
`[Conditional Decomposition]` $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}_1} Y$.
`[Conditional Composition]` $\{X \xrightarrow{\mathcal{C}_1} Y, Z \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{C}} XZ \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y \cap W$.

The *derivation* notion is introduced as usual: For a given set $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$ and $\varphi \in \mathcal{L}_{\Omega,\Gamma}$, we state that φ is derived (or inferred) from Σ by using the CAIL axiomatic system, denoted by $\Sigma \vdash_{\mathscr{C}} \varphi$, if there exists a chain of formulas $\varphi_1, \ldots, \varphi_n \in \mathcal{L}_{\Omega,\Gamma}$ such that $\varphi_n = \varphi$ and, for all $1 \leq i \leq n$, φ_i is either an axiom, an implication in Σ or is obtained by applying the CAIL inference rules to formulas in $\{\varphi_j \mid 1 \leq j < i\}$.

Soundness and completeness: In [12], we prove that every model of Σ is a model of $X \xrightarrow{\mathcal{C}} Y$ iff such implication can be derived syntactically from Σ using the CAIL axiomatic system, i.e.

$$\Sigma \models X \xrightarrow{\mathcal{C}} Y \quad \text{if and only if} \quad \Sigma \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}} Y$$

4 CAISL: Simplification Logic for *CAI*s

Once the preliminary results have been introduced, we now present a new axiomatic system which is more suitable for automated reasoning. We will use the same language and semantics provided in the previous section but give a novel equivalent axiomatic system based on simplification paradigm [4]. For this axiomatic system, the symbol $\vdash_{\mathscr{S}}$ denotes the syntactic derivation.

Definition 6. *The* CAISL *axiomatic system has two axiom schemes:*

`[Non-constraint]` $\vdash_{\mathscr{S}} \emptyset \xrightarrow{\emptyset} \Omega$.
`[Reflexivity]` $\vdash_{\mathscr{S}} X \xrightarrow{\Gamma} X$.

and four inference rules:

`[Decomposition]` $\{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} YZ\} \vdash_{\mathscr{S}} X \xrightarrow{\mathcal{C}_1} Y$.
`[Composition]` $\{X \xrightarrow{\mathcal{C}_1} Y, Z \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{S}} XZ \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW$.
`[Conditional Composition]` $\{X \xrightarrow{\mathcal{C}_1} Y, Z \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{S}} XZ \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y \cap W$.
`[Simplification]` *If* $X \cap Y = \emptyset$,

$$\{X \xrightarrow{\mathcal{C}_1} Y, XZ \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{S}} XZ \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y.$$

The two axiom schemes in CAISL have the following interpretations respectively: (1) all attributes hold for all objects under a void condition, and (2) X always implies itself under all conditions.

The key statement is that both axiomatic systems are equivalent as the following theorem proves. However, as we will show below, CAISL is more appropriate for developing automated methods to reason about implications.

Theorem 1 (Equivalence between CAIL and CAISL). *For any* $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$ *and* $X \xrightarrow{\mathcal{C}} Y \in \mathcal{L}_{\Omega,\Gamma}$*, one has*

$$\Sigma \vdash_{\mathscr{S}} X \xrightarrow{\mathcal{C}} Y \quad \textit{if and only if} \quad \Sigma \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}} Y$$

Proof. To prove the equivalence between both logics, we will show that the inference rules of CAISL can be derived from those in CAIL and vice versa.

i) Inference rules derived from CAIL

`[Reflexivity]:`

1. $X \xrightarrow{\Gamma} X$ ..... Inclusion.

`[Decomposition]:`

1. $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} YZ$ .....Hypothesis.
2. $YZ \xrightarrow{\Gamma} Y$ ..... Inclusion.
3. $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y$ .....1, 2 Trans.
4. $X \xrightarrow{\mathcal{C}_1} Y$ .....3 Cond. Decomp.

`[Composition]:`

1. $X \xrightarrow{\mathcal{C}_1} Y$ ..... Hypothesis.
2. $Z \xrightarrow{\mathcal{C}_2} W$ ..... Hypothesis.
3. $XZ \xrightarrow{\mathcal{C}_1} YZ$.....1 Augm.
4. $YZ \xrightarrow{\mathcal{C}_2} YW$ ..... 2 Augm.
5. $XZ \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} YW$ .....3, 4 Trans.

`[Simplification]:`

1. $X \xrightarrow{\mathcal{C}_1} Y$ ..... Hypothesis.
2. $XZ \xrightarrow{\mathcal{C}_2} W$ ..... Hypothesis.
3. $XZ \setminus Y \xrightarrow{\Gamma} X$ ..... Inclusion.
4. $W \xrightarrow{\Gamma} W \setminus Y$ ..... Inclusion.
5. $XZ \xrightarrow{\mathcal{C}_2} W \setminus Y$ ..... 2, 4 Trans.
6. $XZ \setminus Y \xrightarrow{\mathcal{C}_1} Y$ .....3, 1 Trans.
7. $XZ \setminus Y \xrightarrow{\mathcal{C}_1} XYZ$ .... 6 Augm.
8. $XYZ \xrightarrow{\mathcal{C}_2} WY$ ..... 5 Augm.
9. $XZ \setminus Y \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} WY$ 7, 8 Trans.
10. $XZ \setminus Y \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} W \setminus Y$. 9 Decomp.

ii) Inference rules derived from CAISL

`[Inclusion]:`

1. $XY \xrightarrow{\Gamma} XY$ ..... Reflexivity.
2. $XY \xrightarrow{\Gamma} Y$ ..... 1 Decomp.

`[Augmentation]:`

1. $X \xrightarrow{\mathcal{C}} Y$ ..... Hypothesis.
2. $Z \xrightarrow{\Gamma} Z$.....Reflexivity.
3. $XZ \xrightarrow{\mathcal{C}} YZ$ .....1, 2 Comp.

`[Transitivity]:`

1. $X \xrightarrow{\mathcal{C}_1} Y$ ..... Hypothesis.
2. $Y \xrightarrow{\mathcal{C}_2} Z$ ..... Hypothesis.
3. $X \xrightarrow{\mathcal{C}_1} Y \setminus X$ ..... 1 Decomp.
4. $Y \xrightarrow{\mathcal{C}_2} Z \setminus Y$.....2 Decomp.
5. $X \xrightarrow{\Gamma} X$ ..... Reflexivity.
6. $X \xrightarrow{\Gamma} \emptyset$ ..... 5 Decomp.
7. $XY \xrightarrow{\mathcal{C}_2} Z \setminus Y$ .....4, 6 Comp.
8. $Y \setminus X \xrightarrow{\Gamma} Y \setminus X$ ....Reflexivity.
9. $X \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} Z \setminus Y$ ..... 3, 7 Simp.
10. $X \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} YZ$ ..... 1, 9 Comp.
11. $X \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} Z$ ..... 10 Decomp.

□

Since the two axiomatic systems are equivalent, in the sequel we will omit the subscript in the syntactic derivation symbol using simply $\vdash$.

5 CAISL Equivalences

In this section, we introduce several results which constitute the basis of the automated reasoning method that will be introduced in the next section. These results illustrate how we can use CAISL as a framework to syntactically transform and simplify a set of *CAI*s while entirely preserving their semantics. This is the common feature of the family of Simplification Logics.

The notion of equivalence is introduced as usual: two sets of *CAI*s, Σ_1 and Σ_2, are equivalent, denoted by $\Sigma_1 \equiv \Sigma_2$, when their models are the same. Equivalently, $\Sigma_1 \equiv \Sigma_2$ iff $\Sigma_1 \vdash \varphi$ for all $\varphi \in \Sigma_2$, and $\Sigma_2 \vdash \varphi$ for all $\varphi \in \Sigma_1$.

Lemma 1. *The following equivalences hold:*

$$\{X \xrightarrow{\mathcal{C}_1} Y, X \xrightarrow{\mathcal{C}_2} W\} \equiv \{X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW, X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y, X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W\} \quad (1)$$

$$\{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2} W\} \equiv \{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W, X(V \smallsetminus Y) \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y\} \quad (2)$$

Proof. For Equivalence (1), first, we prove that $X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW$, $X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y$, and $X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$ can be inferred from $\{X \xrightarrow{\mathcal{C}_1} Y, X \xrightarrow{\mathcal{C}_2} W\}$:

- By applying Composition to $X \xrightarrow{\mathcal{C}_1} Y$ and $X \xrightarrow{\mathcal{C}_2} W$, we get $X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW$.
- $X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y$ and $X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$ are obtained by Decomposition.

On the other hand, we prove that $X \xrightarrow{\mathcal{C}_1} Y$ and $X \xrightarrow{\mathcal{C}_2} W$ can be inferred from $\{X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW, X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y, X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W\}$ by applying Conditional Composition.

For Equivalence (2), from $\{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2} W\}$, we infer $XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$ by applying Decomposition to $V \xrightarrow{\mathcal{C}_2} W$. In addition, we infer $XV \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$ by applying Simplification to $X \xrightarrow{\mathcal{C}_1} Y$ and $XV \xrightarrow{\mathcal{C}_2} W$.

Finally, $\{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W, XV \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y\} \vdash XV \xrightarrow{\mathcal{C}_2} W$ is proved. By applying Reflexivity and Decomposition, we get $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} XV \smallsetminus Y$ and, by Transitivity with $XV \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$, one has $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$. Now, by applying Composition to $X \xrightarrow{\mathcal{C}_1} Y$ and $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$, we infer $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} WY$ and, by Decomposition, $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W$. At last, by applying Conditional Composition to $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W$ and $XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$, we obtain $XV \xrightarrow{\mathcal{C}_2} W$. □

The following theorem highlights a common characteristic of Simplification Logics, which shows that inference rules can be read as equivalences that allow redundancy removal.

Theorem 2. *The following equivalences hold:*

`Axiom Eq.:` $\{X \xrightarrow{\emptyset} Y\} \equiv \{X \xrightarrow{\mathcal{C}} \emptyset\} \equiv \emptyset$

`Decomposition Eq.:` $\{X \xrightarrow{\mathcal{C}} Y\} \equiv \{X \xrightarrow{\mathcal{C}} Y \smallsetminus X\}$

`Composition Eq.:` $\{X \xrightarrow{\mathcal{C}} Y, X \xrightarrow{\mathcal{C}} W\} \equiv \{X \xrightarrow{\mathcal{C}} YW\}$

`Conditional Composition Eq.:` $\{X \xrightarrow{\mathcal{C}_1} Y, X \xrightarrow{\mathcal{C}_2} Y\} \equiv \{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y\}$

`Simplification Eq.:` *If* $X \cap Y = \emptyset$, *then*

$$\{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y, XV \xrightarrow{\mathcal{C}_2} W\} \equiv \{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y, XV \smallsetminus Y \xrightarrow{\mathcal{C}_2} W \smallsetminus Y\}$$

Proof. The first equivalence is straightforward because both implications are axioms. For the rest of equivalences, the left to right inference is directly obtained by applying the homonymous inference rule. Thus, we prove the right to left inference:

i) $X \xrightarrow{\mathcal{C}} XY$ is inferred by Composition of $X \xrightarrow{\mathcal{C}} Y \smallsetminus X$ and $X \xrightarrow{\mathcal{C}} X$ obtained by reflexivity. Then, by applying Decomposition, one has $X \xrightarrow{\mathcal{C}} Y$.
ii) $X \xrightarrow{\mathcal{C}} Y$ and $X \xrightarrow{\mathcal{C}} W$ are inferred by applying Decomposition to $X \xrightarrow{\mathcal{C}} YW$.
iii) $X \xrightarrow{\mathcal{C}_1} Y$ and $X \xrightarrow{\mathcal{C}_2} Y$ are inferred from $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y$ by applying Decomposition.
iv) It is a consequence of Axiom Equivalence and Equivalence (2) in Lemma 1.

□

This section has been devoted to equivalences in CAISL of a *CAI*s set in order to remove redundancy or, dually, to extend the set. The effect depends on the direction we apply the equivalence. In next section, we are going to use other equivalences where the empty set plays a main role. The Deduction Theorem presented below gives to the empty set such a role. This theorem establishes the necessary and sufficient condition to ensure the derivability of a *CAI* from a set of *CAI*s.

6 Automated reasoning

This section shows the merits of CAISL for the development of automated methods. Specifically, we present a method that checks whether a *CAI* is derived from a set of *CAI*s. The next theorem is the core of our approach in the design of the automated prover.

Theorem 3 (Deduction). *For any* $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$ *and* $X \xrightarrow{\mathcal{C}} Y \in \mathcal{L}_{\Omega,\Gamma}$, *one has*

$$\Sigma \vdash X \xrightarrow{\mathcal{C}} Y \text{ if and only if } \Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\} \vdash \emptyset \xrightarrow{\mathcal{C}} Y$$

Proof. Straightforwardly, we have $\Sigma \vdash X \xrightarrow{\mathcal{C}} Y$ implies $\Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\} \vdash \emptyset \xrightarrow{\mathcal{C}} Y$. Conversely, assuming $\Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\} \vdash \emptyset \xrightarrow{\mathcal{C}} Y$, we have to prove that $\mathbb{K} \models \Sigma$ implies $\mathbb{K} \models X \xrightarrow{\mathcal{C}} Y$ for each model $\mathbb{K}$.

Consider $\mathbb{K} = \langle G, M, B, I\rangle$ as a model of Σ. In order to prove $(X, \{c\})' \subseteq (Y, \{c\})'$ for all $c \in C$ in $\mathbb{K}$, we build the context $\mathbb{K}_1 = \langle G_1, M, B, I_1\rangle$ where $G_1 = (X, \{c\})'$ and $I_1 = I \cap (G_1 \times M \times B)$.

Since $\mathbb{K} \models \Sigma$, we have $\mathbb{K}_1 \models \Sigma \cup \{\emptyset \xrightarrow{\{c\}} X\}$ and therefore, by hypothesis, $\mathbb{K}_1 \models \{\emptyset \xrightarrow{\{c\}} Y\}$. That is, $(Y, \{c\})' \supseteq (\emptyset, \{c\})' = G_1 = (X, \{c\})'$.

If we go back to the original triadic context $\mathbb{K}$, $(X, \{c\})'$ remains unchanged whereas $(Y, \{c\})'$ could grow up. Therefore, in $\mathbb{K}$, one has $(X, \{c\})' \subseteq (Y, \{c\})'$ for all $c \in C$. □

Function CAISL-Prover(Σ,$X \xrightarrow{\mathcal{C}} Y$)

input : A set of implications Σ, and a *CAI* $X \xrightarrow{\mathcal{C}} Y$

output: A boolean answer

begin

$\Delta_X := X \times \mathcal{C}$

$\Delta_Y := (Y \times \mathcal{C}) \setminus (X \times \mathcal{C})$

repeat

flag:=false

foreach $U \xrightarrow{\mathcal{C}_1} V \in \Sigma$ *with* $\mathcal{C}_1 \cap \mathcal{C} \neq \emptyset$ **do**

$\Delta_C := \{c \in \mathcal{C}_1 \cap \mathcal{C} \mid U \times \{c\} \subseteq \Delta_X\}$

if $\Delta_C \neq \emptyset$ **then** ..... Equivalence (4)

$\Delta_X := \Delta_X \cup (V \times \Delta_C)$

$\Delta_Y := \Delta_Y \setminus (V \times \Delta_C)$

$\Sigma := \Sigma \setminus \{U \xrightarrow{\mathcal{C}_1} V\}$

$\mathcal{C}_1 := \mathcal{C}_1 \setminus \Delta_C$

if $\mathcal{C}_1 \neq \emptyset$ **then** $\Sigma := \Sigma \cup \{U \xrightarrow{\mathcal{C}_1} V\}$

flag:=true

$\Delta_C := \{c \in \mathcal{C}_1 \cap \mathcal{C} \mid V \times \{c\} \subseteq \Delta_X\}$

if $\Delta_C \neq \emptyset$ **then** ..... Equivalence (5)

if $\Delta_C = \mathcal{C}_1$ **then** $\Sigma := \Sigma \setminus \{U \xrightarrow{\mathcal{C}_1} V\}$

else $\Sigma := (\Sigma \setminus \{U \xrightarrow{\mathcal{C}_1} V\}) \cup \{U \xrightarrow{\mathcal{C}_1 \setminus \Delta_C} V\}$

until ($\Delta_Y = \emptyset$) *or* (flag=false)

return the boolean value ($\Delta_Y = \emptyset$)

Theorem 3 guides the design of the automated prover. To check that the formula $X \xrightarrow{\mathcal{C}} Y$ is inferred from the set Σ we apply the family of simplification equivalences iteratively - while it is possible - to the set $\Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\}$ looking for $\emptyset \xrightarrow{\mathcal{C}} Y$.

The following proposition revisits Theorem 2 by instantiating the particular case of having the empty premise.

Proposition 2. *The following equivalences hold:*

$$\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} \equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \setminus X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} V \setminus X, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\} \quad (3)$$

$$\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} \equiv \{\emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} XV, \emptyset \xrightarrow{\mathcal{C}_1 \setminus \mathcal{C}_2} X, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\},\ \textit{when } U \subseteq X \quad (4)$$

$$\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} \equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\},\ \textit{when } V \subseteq X \quad (5)$$

Proof. Equivalence (3) is a particular case of Equivalence (2). In particular, when $U \subseteq X$, Equivalence (4) is obtained from (3) by applying Conditional Composition

and Composition equivalences:

$$\begin{aligned}\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} V \setminus X, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\} \\ &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1 \setminus \mathcal{C}_2} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} V \setminus X, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\} \\ &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1 \setminus \mathcal{C}_2} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} XV, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\}\end{aligned}$$

Analogously, when $V \subseteq X$, by applying Equivalence (3) and Axiom equivalence, one has Equivalence (5):

$$\begin{aligned}\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \setminus X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} \emptyset, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\} \\ &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2 \setminus \mathcal{C}_1} V\}\end{aligned}$$

□

The equivalences introduced in Proposition 2 constitute the core of the function called `CAISL-Prover`, which acts as an automated prover for CAISL. The prover works by splitting the original formula into its left and right hand sides (see Theorem 2) and, by applying the equivalences, we check whether its right side can be reduced to the empty set. The derivability is proved if and only if such reduction is fulfilled.

Finally, we conclude this section with an illustrative example.

Steep	State
0	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
1	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathbf{Q} \xrightarrow{\mathbf{de}} \mathbf{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
2	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \cancel{\mathbf{M} \xrightarrow{\mathbf{a}} \mathbf{T}}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
3	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \cancel{\mathbf{Q} \xrightarrow{\mathbf{bd}} \mathbf{L}}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
4	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a),(Q,b),(Q,d)\}$ $\Delta_Y = \{(Q,a)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \mathbf{ML} \xrightarrow{\cancel{\mathbf{bde}}} \mathbf{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
5	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a),(Q,b),(Q,d),(R,a)\}$ $\Delta_Y = \{(Q,a)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \mathrm{ML} \xrightarrow{\mathrm{e}} \mathrm{Q}, \mathbf{T} \xrightarrow{\cancel{\mathbf{ab}}} \mathbf{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
6	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a),(Q,b),(Q,d),(R,a),(Q,a)\}$ $\Delta_Y = \emptyset$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \mathrm{ML} \xrightarrow{\mathrm{e}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{b}} \mathrm{R}, \mathbf{R} \xrightarrow{\cancel{\mathbf{ae}}} \mathbf{Q}\}$
Output	Return `TRUE`

Table 1. Illustration of the derivability of $\mathrm{ML} \xrightarrow{\mathrm{abd}} \mathrm{Q}$

Example 2. Let $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$ be a set of *CAI*s. The `CAISL-Prover` function proves that $\mathrm{ML} \xrightarrow{\mathrm{abd}} \mathrm{Q}$ is inferred from Σ (see Table 1) and $\mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{Q}$ is not inferred from Σ (see Table 2).

Step	State
0	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M},\ \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
1	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathbf{Q} \xrightarrow{\mathbf{de}} \mathbf{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
2	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \cancel{\mathbf{M} \xrightarrow{\mathbf{a}} \mathbf{T}}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
3	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathbf{Q} \xrightarrow{\mathbf{bd}} \mathbf{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
4	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathbf{ML} \xrightarrow{\mathbf{bde}} \mathbf{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
5	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \cancel{\mathbf{T} \xrightarrow{\mathbf{ab}} \mathbf{RL}}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
6	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathbf{R} \xrightarrow{\cancel{\mathbf{ae}}} \mathbf{Q}\}$
Loop 2	
7	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathbf{Q} \xrightarrow{\mathbf{de}} \mathbf{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{R} \xrightarrow{\mathrm{e}} \mathrm{Q}\}$
8	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathbf{Q} \xrightarrow{\cancel{\mathbf{bd}}} \mathbf{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{R} \xrightarrow{\mathrm{e}} \mathrm{Q}\}$
9	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{d}} \mathrm{L}, \mathbf{ML} \xrightarrow{\mathbf{bde}} \mathbf{Q}, \mathrm{R} \xrightarrow{\mathrm{e}} \mathrm{Q}\}$
10	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{d}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathbf{R} \xrightarrow{\mathbf{e}} \mathbf{Q}\}$
Output	Return `FALSE`

Table 2. Illustration of the non derivability of $\mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{Q}$

7 Conclusion and Future Work

We have proposed a logic named CAISL to deal with conditional attribute implications in Triadic Concept Analysis. Its soundness and completeness have been

proved by establishing the equivalence between the axioms and rules of CAISL and those of CAIL - a recently developed solution [12]. This novel approach is strongly based on the Simplification paradigm which is an useful formalism to develop automated methods. In this direction, CAISL has been used to build an automated prover to check the derivability of a *CAI* from a set of *CAI*s, which is an important issue in data and knowledge management.

The use of a logic-based approach is a challenging but very interesting issue that has not been explored in the TCA framework yet. Our short-term research activity is to adapt our proposal to other kinds of triadic implications and analyze the interplay between them.

Acknowledgment

This work is supported by regional project TIN2014-59471-P of the Science and Innovation Ministry of Spain, co-funded by the European Regional Development Fund (ERDF) as well as by a discovery grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Armstrong, W.: Dependency structures of data base relationships. Proc. IFIP Congress. North Holland, Amsterdam pp. 580–583 (1974)
2. Biedermann, K.: How triadic diagrams represent conceptual structures. In: ICCS. pp. 304–317 (1997)
3. Biedermann, K.: Powerset trilattices. In: ICCS. pp. 209–224 (1998)
4. Cordero, P., Enciso, M., Mora, A., P. de Guzmn, I.: Slfd logic: Elimination of data redundancy in knowledge representation. In: IBERAMIA. Lecture Notes in Computer Science, vol. 2527, pp. 141–150. Springer Berlin Heidelberg (2002)
5. Fagin, R.: Functional dependencies in a relational database and propositional logic. IBM. Journal of research and development 21 (6), 534–544 (1977)
6. Ganter, B., Obiedkov, S.A.: Implications in triadic formal contexts. In: ICCS. pp. 186–195 (2004)
7. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Trias - an algorithm for mining iceberg tri-lattices. In: ICDM. pp. 907–911 (2006)
8. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: ICCS. pp. 32–43 (1995)
9. Maier, D.: The theory of Relational Databases. Computer Science Press (1983)
10. Missaoui, R., Kwuida, L.: Mining triadic association rules from ternary relations. In: Formal Concept Analysis - 9th International Conference, ICFCA 2011, Nicosia, Cyprus, May 2-6, 2011. Proceedings. pp. 204–218 (2011)
11. Peirce, C.S.: Collected Papers. Harvard University Press, Cambridge (1931-1935)
12. Rodríguez-Lorenzo, E., Cordero, P., Enciso, M., Missaoui, R., Mora, A.: An axiomatic system for conditional attribute implications in triadic concept analysis. Submitted to International Journal of Intelligent Systems (2016)
13. Wille, R.: The basic theorem of triadic concept analysis. Order 12(2), 149–158 (1995)