Belief Base Contraction by Cutting Connections

Matthew James Lynn , **James P. Delgrande**
Simon Fraser University, Canada
mlynn@cs.sfu.ca, jim@cs.sfu.ca

Abstract

This paper presents a methodology for constructing belief base contraction operators which preserve the syntactic structure of the initial belief base. We believe that preserving the structure of the initial belief base is an important requirement for practical belief base contraction. In our approach, individual occurrences of propositional variables are differentiated by the introduction of tags. Using the characterisation of unsatisfiability provided by the connection method, we may identify which specific variable occurrences result in a belief being entailed, and thus apply selective substitutions for these occurrences in order to block that entailment by effectively *cutting connections*. The resulting belief base has a structure almost identical to that of the initial belief base. We demonstrate that these contraction operators satisfy a number of desirable properties. Next, we present an algorithm for path-contraction and use it to show the corresponding decision problem is in **NP**. Finally, we introduce the notion of path-entailment to capture very precisely what is preserved after a contraction, and show that the class of *regular path-contraction operators* satisfy an analogue of Parikh's postulate.

1 Introduction

Belief contraction is a form of belief change which occurs whenever an agent realises that it holds a belief which is no longer justified, and subsequently must modify its existing beliefs to ensure that this specified belief is no longer entailed by those beliefs it decides to retain. The challenge is to preserve as many of its existing beliefs as possible. This process is formalised as a *belief contraction operator* $-$ which takes a belief state κ alongside an existing belief ϕ and produces a contracted knowledge base $\kappa - \phi$.

Our contention is that belief contraction operators, and belief change functions more generally, should satisfy a *principle of structural preservation* analogous to the principle of categorical matching, which requires that the structure of the contracted beliefs should resemble the structure of the initial beliefs to the greatest extent possible. This is in conflict with purely semantic approaches to belief change, such as the Katsuno–Mendelzon approach, which require syntax-independence. This is also in conflict with approaches such as prime implicate based belief revision (Bienvenu, Herzig, and Qi 2008) which always convert the knowledge base into disjunctive normal form, which would generally involve an exponential cost as knowledge bases generally have the form of a large conjunction of small beliefs. We believe that pursuing the principle of structural preservation will help close the gap between practical belief representation and the representations convenient for naive implementations of belief change functions.

In this paper, we will introduce the class of *path-contraction operators* which satisfy the principle of structural preservation. These will work by tagging every occurrence of a propositional variable within the existing belief base with a unique tag, and then applying the connection method (Bibel 1981) to determine which particular occurrences contribute to the unwanted belief being entailed. Using this information, a process of selective substitution of $\top$ or $\bot$ for these particular occurrences, which we call *attenuation*, is employed to produce the resulting contracted belief base. The nature of this construction means that these path-contraction operators preserve the initial structure to a great extent.

This can be understood from a tableaux perspective: in order to compute $\kappa - \phi$ we proceed as follows. Assuming $\kappa \vdash \phi$ it follows that $\kappa \wedge \neg\phi$ is unsatisfiable, and therefore we may construct a closed fully expanded tableaux for $\kappa \wedge \neg\phi$. At this point, we select one or more branches, and selectively remove literals appearing along these branches which originate in κ until at least one branch is open. This results in a formula κ' obtained from κ by our process of *attenuation*, with the property that $\kappa \vdash \kappa'$ and $\kappa' \nvdash \phi$. Then, define $\kappa - \phi$ as κ'.

In Section 2 we present background material on propositional matrices, the connection method, existing approaches to belief base contraction, and on the distinction between explicit an implicit beliefs. Section 3 introduces the technique of attenuation, the notion of a cutting, and uses these to define the class of path-contraction operators which are shown to satisfy a handful of desirable properties. Section 4 presents a concrete algorithm for performing path-contraction alongside a complexity analysis. In Section 5 we introduce the notion of path-entailment and path-independence, which allow for characterising the preservation properties of path-contraction operators, and show an analogue of Parikh's Postulate to be satisfied by all *regular* path-contraction operators. We next compare this to existing literature in Section 6, and finally offer a summary of our contributions in Section 7.

2 Background Material

2.1 Propositional Logic

Let $V = \{p, q, r, \dots\}$ be a finite set of propositional variables. The corresponding propositional language L is constructed from V by applying the propositional connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$. We will use $\phi, \psi, \kappa, \dots$ to range over propositional formulae in L. Propositional formulae of the form $\neg p$ or p are called *literals*. When every negation occurring in ϕ is the negation of a variable we say that ϕ is in *negation normal form*. When ϕ is a disjunction of conjunctions of literals we say it is in *disjunctive normal form*, and when ϕ is a conjunction of disjunctions of literals we say it is in *conjunctive normal form*.

Functions $\nu, \mu : V \rightarrow \{T, F\}$ are referred to as *truth-value assignments* or just as *assignments*. Given a propositional formula $\mathcal{A}$ we will write $[\phi]$ for the set of assignments satisfying ϕ, with $\phi \vdash \psi$ indicating that $[\phi] \subseteq [\psi]$, and $\phi \equiv \psi$ indicating that $[\phi] = [\psi]$. In the case $[\phi] \neq \varnothing$ we say that ϕ is **satisfiable**.

2.2 Belief Base Contraction Operators

Belief contraction operators were formalised by Alchourron, Gärdenfors, and Makinson (1985) as binary functions $-$ which map a belief state κ alongside a belief to contract ϕ into a new contracted belief state $\kappa - \phi$ such that $\kappa - \phi \nvdash \phi$. Working with a finite vocabulary, both the belief state κ and the belief ϕ may be represented as propositional formulae alongside the lines of (Katsuno and Mendelzon 1991). Among the many postulates discussed in the aforementioned, there is an assumption that whenever $\kappa_1 \equiv \kappa_2$ then $\kappa_1 - \phi \equiv \kappa_2 - \phi$ meaning that belief contraction is meant to be syntax-independent. Rejecting this assumption leads to the subject of *belief base contraction*.

In (Hansson 2012) a number of different properties are proposed that a belief base contraction operator may be required to satisfy. For our purposes, we will work with a subset of those postulates discussed in (Caridroit, Konieczny, and Marquis 2017).

Definition 2.1. *A binary function $-$ from $L \times L \rightarrow L$ is a* ***belief base contraction operator*** *iff it satisfies the following postulates:*

C1. *If $\nvdash \phi$ then $\kappa - \phi \nvdash \phi$.*

C2. *If $\vdash \phi$ then $\kappa - \phi = \kappa$.*

C3. *$\kappa \vdash \kappa - \phi$.*

C4. *If $\kappa \nvdash \phi$ then $\kappa - \phi = \kappa$.*

Postulate (C1) states that whenever ϕ is not a tautology, then $\kappa - \phi$ must not entail ϕ. Postulate (C2) states that whenever ϕ is a tautology, then $\kappa - \phi$ should not change anything as there is nothing which can be done to stop the entailment of ϕ anyways. Postulate (C3) states that $\kappa - \phi$ must be a consequence of κ, so that the process of contraction cannot result in new beliefs being adopted. Finally, postulate (C4) states that whenever ϕ is not a consequence of κ then contracting κ by ϕ should result in nothing being changed. We regard these postulates as serving to demarcate the broadest class of functions worth considering as belief base contraction operators, as the postulates capture very little of the requirement of minimal change.

We will consider one further postulate later on. In order to capture the requirement of minimal change, (Parikh 1999) proposed to require that when some beliefs κ are being revised by a new belief ϕ then those beliefs in κ irrelevant to ϕ should remain unchanged. It is natural enough to rephrase this for contraction as follows:

Definition 2.2. *A belief base contraction operator $-$ satisfies* ***Parikh's postulate*** *if and only if for any formulae κ_1, κ_2, and ϕ such that $V(\kappa_1) \cap (V(\kappa_2) \cup V(\phi)) = \varnothing$ then it follows that*

P. $(\kappa_1 \wedge \kappa_2) - \phi \equiv \kappa_1 \wedge (\kappa_2 - \phi)$.

2.3 Propositional Matrices

Our approach to belief contraction will involve the selective substitution of $\top$ or $\bot$ for propositional variables appearing within the belief base κ. In order to facilitate this, we will attach distinct *tags* to each separate occurrence of a propositional variable in κ. Our examples will use positive integers for tags, but the choice is arbitrary. We will refer to propositional formulae in negation normal form which have been annotated with tags as *propositional matrices*, and use the variables $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{K}, \dots$ to range over them.

Definition 2.3. *A* ***matrix*** *is an expression constructed via the following rules:*

1. *The symbol $\top$ is a matrix.*
2. *If p is a variable and i is an tag then p^i and $\neg p^i$ are matrices.*
3. *If $\mathcal{A}$ and $\mathcal{B}$ are matrices with no tags in common, then $(\mathcal{A} \wedge \mathcal{B})$ is a matrix.*
4. *If $\mathcal{A}$ and $\mathcal{B}$ are matrices with no tags in common, then $(\mathcal{A} \vee \mathcal{B})$ is a matrix.*
5. *Nothing else is a matrix.*

It is worth noting that the use of the term "matrix" for a formula in negation normal form within the connection method literature is motivated by a graphical notation where-in disjunctions are represented by vertical juxtaposition, and conjunctions are represented by horizontal juxtaposition, or vice versa depending on the author. To illustrate, one might write

$$p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4 = \begin{bmatrix} p^1 & \begin{bmatrix} \neg p^2 \\ q^3 \end{bmatrix} & \neg q^4 \end{bmatrix}.$$

Our introduction of tags into the definition of a matrix amounts to a slight simplification of the approach in (Kreitz and Otten 1999; Otten 2011) which instead associates every subformula with a position label of its own.

Although matrices are required to be in negation normal form, we will sometimes write $\neg\mathcal{A}$ to refer to the matrix obtained by temporarily treating $\mathcal{A}$ as a formula, and computing the negation normal form of $\neg\mathcal{A}$ by pushing negations down while retaining the tags. For example, $\neg(p^1 \vee q^2)$ refers to the matrix $\neg p^1 \wedge \neg q^2$

Definition 2.4. *We will write $T(\mathcal{A})$ for the set of tags occurring in $\mathcal{A}$, and say that matrices $\mathcal{A}$ and $\mathcal{B}$ are **tag-disjoint** when $T(\mathcal{A}) \cap T(\mathcal{B}) = \varnothing$.*

Definition 2.5. *If p is a propositional variable then a matrix of the form p^i or $\neg p^i$ is called a **literal**. In the case the literal p^i or $\neg p^i$ appears in a matrix $\mathcal{A}$ we will say that i **tags** the variable p.*

Definition 2.6. *The **erasure** of a matrix $\mathcal{A}$ is the propositional formula ϕ obtained by deleting the tags from $\mathcal{A}$, which we denote by $\epsilon(\mathcal{A})$.*

Example 2.1. *The matrix $p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4$ has erasure $p \wedge (\neg p \vee q) \wedge \neg q$.*

When working with matrices we will say a truth-value assignment ν satisfies $\mathcal{A}$ when ν satisfies $\epsilon(\mathcal{A})$. We will also say that $\mathcal{A}$ entails $\mathcal{B}$ and write $\mathcal{A} \vdash \mathcal{B}$ when $\epsilon(\mathcal{A})$ entails $\epsilon(\mathcal{B})$.

2.4 Connections in Propositional Matrices

Unsatisfiability of propositional matrices may be characterised in terms of *paths* and *connections*, where paths correspond roughly to the disjuncts of a disjunctive normal form of a formula, and connections correspond to pairs of complementary literals in those disjuncts. In the context of automated reasoning, this has become known as the *connection method* which originates with (Bibel 1981) and (Andrews 1976). Our presentation below is a variation on that of (Wallen 1987) and (Otten 2011).

Definition 2.7. *A **path** is a set $\mathfrak{p}$ of literal matrices such that each tag occurring in $\mathfrak{p}$ occurs exactly once. If there exists a variable p and tags i and j such that $\mathfrak{p}$ contains p^i and $\neg p^j$ then $\{p^i, \neg p^j\}$ is called a **connection** in $\mathfrak{p}$, and $\mathfrak{p}$ is said to be **connected**. If $\mathfrak{p}$ contains no connection, then $\mathfrak{p}$ is **unconnected**.*

In order to construct the set of paths through a particular matrix, we will employ the following two functions defined on sets of paths:

Definition 2.8. *If X and Y are sets of paths, then $X \oplus Y$ and $X \otimes Y$ are defined as follows:*

$$\begin{aligned} X \oplus Y &:= X \cup Y, \\ X \otimes Y &:= \{\mathfrak{p} \cup \mathfrak{q} \mid \mathfrak{p} \in X \text{ and } \mathfrak{q} \in Y\}\,. \end{aligned}$$

Definition 2.9. *If $\mathcal{A}$ is a matrix then the set of **paths through** $\mathcal{A}$, denoted by $[\![\mathcal{A}]\!]$, is defined by the following rules:*

1. *If $\mathcal{A}$ is $\top$ then $[\![\mathcal{A}]\!] = \{\varnothing\}$.*
2. *If $\mathcal{A}$ is p^i or $\neg p^i$ then $[\![\mathcal{A}]\!] = \{\{\mathcal{A}\}\}$.*
3. *If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ then $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \oplus [\![\mathcal{C}]\!]$.*
4. *If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]$.*

Computing the paths through a matrix effectively amounts to converting the matrix into a disjunctive normal form by relying solely on the associative, commutative, distributive, and De Morgan laws.

Example 2.2. *Consider the matrix $p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4$, which is unsatisfiable and has the following paths:*

$$\begin{aligned} &[\![p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4]\!] \\ &= [\![p^1]\!] \otimes ([\![\neg p^2]\!] \oplus [\![q^3]\!]) \otimes [\![\neg q^4]\!] \\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}\} \oplus \{\{q^3\}\}) \otimes \{\{\neg q^4\}\} \\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}, \{q^3\}\}) \otimes \{\{\neg q^4\}\} \\ &= \{\{p^1, \neg p^2, \neg q^4\}, \{p^1, q^3, \neg q^4\}\}. \end{aligned}$$

Observe that the first path contains the connection $\{p^1, \neg p^2\}$ whereas the second path contains the connection $\{q^3, \neg q^4\}$, so that every path is connected. Recalling the graphical notation

$$p^1 \wedge (\neg p^2 \vee q^3) \wedge \neg q^4 = \begin{bmatrix} p^1 & \begin{bmatrix} \neg p^2 \\ q^3 \end{bmatrix} & \neg q^4 \end{bmatrix},$$

we see that the paths through a matrix correspond to horizontal lines drawn across the matrix which intersect one literal from every column.

That every path through our example matrix is connected, and the matrix itself is unsatisfiable, is not a coincidence. At the heart of the connection method in (Bibel 1981; Andrews 1976) is a theorem stating that a matrix is unsatisfiable if and only if every path through the matrix is connected. Although this characterisation is well known, given our modified definitions for matrices and paths, we will take a moment to prove this result for the convenience of the reader. We start with the following lemma:

Lemma 2.1. *An interpretation ν satisfies a matrix $\mathcal{A}$ if and only if for some path $\mathfrak{p}$ through $\mathcal{A}$ it follows that ν satisfies every element of $\mathfrak{p}$.*

Proof. Suppose $\mathcal{A}$ is a matrix and ν is an interpretation, and proceed by induction on the complexity of $\mathcal{A}$ followed by case analysis on the primary connective of the matrix underlying $\mathcal{A}$.

1. In the case $\mathcal{A}$ is $\top$ then it follows that ν satisfies $\mathcal{A}$, and ν satisfies every element of the single path $\varnothing \in [\![\mathcal{A}]\!]$.
2. In the case $\mathcal{A}$ is a literal p^i or $\neg p^i$, then ν satisfies $\mathcal{A}$ if and only if it satisfies every element of the path $\{\mathcal{A}\}$. As $[\![\mathcal{A}]\!] = \{\{\mathcal{A}\}\}$ the conclusion follows.
3. In the case $\mathcal{A}$ is a disjunction $(\mathcal{B} \vee \mathcal{C})$ suppose ν is a valuation satisfying $\mathcal{A}$. It follows that ν satisfies either $\mathcal{B}$ or $\mathcal{C}$, and therefore by the induction hypothesis there either exists a path $\mathfrak{p} \in [\![\mathcal{B}]\!]$ such that ν satisfies every element of $\mathfrak{p}$, or there exists a path $\mathfrak{p} \in [\![\mathcal{C}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \oplus [\![\mathcal{C}]\!]$, it follows in either case that there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$ as required. Conversely, suppose that ν is a valuation such that there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \oplus [\![\mathcal{C}]\!]$ it follows that either $\mathfrak{p} \in [\![\mathcal{B}]\!]$ in which case the induction hypothesis shows that ν satisfies $\mathcal{B}$, or $\mathfrak{p} \in [\![\mathcal{C}]\!]$ in which case the induction hypothesis shows that ν satisfies $\mathcal{C}$. In either case, it follows that ν satisfies $\mathcal{A} = (\mathcal{B} \vee \mathcal{C})$ so the conclusion follows.

4. In the case $\mathcal{A}$ is a conjunction $(\mathcal{B} \wedge \mathcal{C})$ suppose ν is a valuation satisfying $\mathcal{A}$. It follows that ν satisfies both $\mathcal{B}$ and $\mathcal{C}$, and therefore by the induction hypothesis there exists a path $\mathfrak{q} \in [\![\mathcal{B}]\!]$ such that ν satisfies every element of $\mathfrak{q}$, and there also exists a path $\mathfrak{r} \in [\![\mathcal{B}]\!]$ such that ν satisfies every element of $\mathfrak{r}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]$ it follows that $\mathfrak{p} = \mathfrak{q} \cup \mathfrak{r}$ is a path through $\mathfrak{p}$ such that ν satisfies every element of $\mathfrak{p}$, as required. Conversely, suppose that ν is a valuation for which there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Observing that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]$ it follows that there exist paths $\mathfrak{q} \in [\![\mathcal{B}]\!]$ and $\mathfrak{r} \in [\![\mathcal{C}]\!]$ such that $\mathfrak{p} = \mathfrak{q} \cup \mathfrak{r}$. Therefore, as ν satisfies every element of $\mathfrak{q}$ and every element of $\mathfrak{r}$, by applying the induction hypothesis it follows that ν satisfies $\mathcal{B}$ and $\mathcal{C}$, which is to say ν satisfies $\mathcal{A} = (\mathcal{B} \wedge \mathcal{C})$ so the conclusion follows.

□

Theorem 2.1. *A matrix $\mathcal{A}$ is unsatisfiable if and only if every path through $\mathcal{A}$ is connected.*

Proof. Suppose that $\mathcal{A}$ is unsatisfiable. Assume for the sake of contradiction that there exists a path $\mathfrak{p}$ through $\mathcal{A}$ which contains no connection. Then consider an interpretation ν such that $\nu(p) = \top$ if $p^i \in \mathfrak{p}$, $\nu(p) = \bot$ if $\neg p^i \in \mathfrak{p}$, with $\nu(p)$ chosen arbitrarily otherwise. It follows that ν satisfies every element of the path $\mathfrak{p}$, and hence by the prior Lemma 2.1 ν satisfies $\mathcal{A}$. This is a contradiction, so it must be that every path through $\mathcal{A}$ was connected.

Conversely, suppose every path through $\mathcal{A}$ is connected and assume for the sake of contradiction that ν is an interpretation satisfying $\mathcal{A}$. It follows that there exists a path $\mathfrak{p}$ such that ν satisfies every element of $\mathfrak{p}$. However, because every path is connected, $\mathfrak{p}$ is connected and therefore there exists a variable p alongside tags i and j such that $\mathfrak{p}$ contains both p^i and $\neg p^j$. However, this means that $\nu(p) = \top$ and $\nu(p) = \bot$ which is a contradiction. Therefore, $\mathcal{A}$ must be unsatisfiable. □

2.5 Explicit and Implicit Beliefs

Requiring belief contraction operators to be invariant under logical equivalence is unreasonable when studying resource-limited agents. It becomes impossible to differentiate between the beliefs the agent holds, and the logical consequences of those beliefs. Effectively, every consequence of its beliefs must be treated as if it is instantaneously known, and any contradictory beliefs results in every sentence being believed.

These sorts of concerns motivated (Levesque 1984) to differentiate between the *explicit beliefs* which an agent possesses, and those *implicit beliefs* which it would be able to conclude based off of inferences from its explicit beliefs given adequate time.

One concern is that explicitly believing $\mathcal{A} \wedge \mathcal{B}$ seems to imply one should explicitly believe $\mathcal{A}$ as well. Hence, there is a need for an intermediate approach, wherein certain immediate consequences of explicit beliefs are regarded as among the explicit beliefs, while consequences involving more elaborate inferences are relegated to the category of implicit belief.

Definition 2.10. *Matrices $\mathcal{A}$ and $\mathcal{B}$ are **path-equivalent** iff $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$.*

Example 2.3. *The matrices $p^1 \vee (q^2 \vee r^3)$ and $r^3 \vee (q^2 \vee p^1)$ are path-equivalent, whereas the matrices $p^1 \vee \neg p^2$ and $\top$ are not path-equivalent.*

In our approach, almost everything is invariant under path-equivalence, or can be chosen to be so. It follows from Lemma 2.1 that path-equivalence implies logical equivalence, however path-equivalence is far more restrictive. We consider path-equivalence to offer an intermediary between completely syntax-insensitive approaches which fail to differentiate implicit and explicit beliefs, and completely syntax-sensitive approaches which risk becoming ad-hoc.

3 Path-Contraction via Matrix Attenuation

In this section we introduce the class of *path-contraction operators* which operate by applying selective substitutions of $\top$ or $\bot$ for particular occurrences of variables within a matrix $\mathcal{K}$ in order to construct a matrix $\mathcal{K}'$ which does not entail another matrix $\mathcal{A}$. We refer to this process of selective substitution as *matrix attenuation*. An advantage of matrix attenuation is that it amounts to a straightforward edit to the original knowledge base, without any requirement of a costly conversion to a conjunctive or disjunctive normal form. Hence, the path-contraction operators we obtain will leave the structure of the knowledge bases being contracted relatively unchanged.

Definition 3.1. *The **attenuation of** a matrix $\mathcal{A}$ **at** a tag i is the matrix $\mathcal{A}_i$ defined by the following rules:*

1. *If $\mathcal{A}$ is $\top$ then $\mathcal{A}_i = \top$.*
2. *If $\mathcal{A}$ is p^j or $\neg p^j$ and $i \neq j$ then $\mathcal{A}_i = \mathcal{A}$.*
3. *If $\mathcal{A}$ is p^j or $\neg p^j$ and $i = j$ then $\mathcal{A}_i = \top$.*
4. *If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ then $\mathcal{A}_i = (\mathcal{B}_i \vee \mathcal{C}_i)$.*
5. *If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then $\mathcal{A}_i = (\mathcal{B}_i \wedge \mathcal{C}_i)$.*

It is possible to characterise the paths through an attenuation of a matrix as being attenuations of the paths through the matrix itself, where attenuations of paths are defined as follows:

Definition 3.2. *The **attenuation of** a path $\mathfrak{p}$ **at** a tag i is the path $\mathfrak{p}_i$ consisting of those literals in $\mathfrak{p}$ not containing the tag i.*

Theorem 3.1. *If $\mathcal{A}$ is a matrix and i is a tag then $[\![\mathcal{A}_i]\!] = \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}$.*

Proof. Proceed by induction on the complexity of $\mathcal{A}$, and within the induction by case analysis.

1. If $\mathcal{A}$ is $\top$ then $[\![\mathcal{A}]\!] = \{\varnothing\}$ and $\mathcal{A}_i = \mathcal{A}$ so it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\top]\!] \\ &= \{\varnothing\} \\ &= \{\varnothing_i\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in \{\varnothing\}\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

2. If $\mathcal{A}$ is p^j or $\neg p^j$ then there are two cases. In the case $i = j$ then $\mathcal{A}_i = \top$ it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\top]\!] \\ &= \{\varnothing\} \\ &= \{\{\mathcal{A}\}_i\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in \{\{\mathcal{A}\}\}\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

In the case $i \neq j$ then $\mathcal{A}_i = \mathcal{A}$ and it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= \{\{\mathcal{A}\}\} \\ &= \{\{\mathcal{A}\}_i\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in \{\{\mathcal{A}\}\}\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

In either case, our choice satisfies the requirement.

3. If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ then by the induction hypothesis $[\![\mathcal{B}_i]\!] = \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\}$ and $[\![\mathcal{C}_i]\!] = \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\}$. Observing $\mathcal{A}_i = (\mathcal{B}_i \vee \mathcal{C}_i)$ it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\mathcal{B}_i \vee \mathcal{C}_i]\!] \\ &= [\![\mathcal{B}_i]\!] \cup [\![\mathcal{C}_i]\!] \\ &= \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\} \cup \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{B}]\!] \cup [\![\mathcal{C}]\!]\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

Thus, the requirement is satisfied.

4. If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then by the induction hypothesis $[\![\mathcal{B}_i]\!] = \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\}$ and $[\![\mathcal{C}_i]\!] = \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\}$. Observing $\mathcal{A}_i = (\mathcal{B}_i \wedge \mathcal{C}_i)$ it follows that

$$\begin{aligned}[\![\mathcal{A}_i]\!] &= [\![\mathcal{B}_i \wedge \mathcal{C}_i]\!] \\ &= [\![\mathcal{B}_i]\!] \otimes [\![\mathcal{C}_i]\!] \\ &= \{\mathfrak{q}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!]\} \otimes \{\mathfrak{r}_i \mid \mathfrak{r} \in [\![\mathcal{C}]\!]\} \\ &= \{\mathfrak{q}_i \cup \mathfrak{r}_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!], \mathfrak{r} \in [\![\mathcal{C}]\!]\} \\ &= \{(\mathfrak{q} \cup \mathfrak{r})_i \mid \mathfrak{q} \in [\![\mathcal{B}]\!], \mathfrak{r} \in [\![\mathcal{C}]\!]\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{B}]\!] \otimes [\![\mathcal{C}]\!]\} \\ &= \{\mathfrak{p}_i \mid \mathfrak{p} \in [\![\mathcal{A}]\!]\}.\end{aligned}$$

Thus, the requirement is satisfied.

□

Example 3.1. *Consider the matrix $p^1 \wedge (\neg p^2 \vee q^3) \wedge (\neg q^4 \vee r^5)$ which logically entails r, and has the following paths:*

$$\begin{aligned}&[\![p^1 \wedge (\neg p^2 \vee q^3) \wedge (\neg q^4 \vee r^5)]\!] \\ &= [\![p^1]\!] \otimes ([\![\neg p^2]\!] \oplus [\![q^3]\!]) \otimes ([\![\neg q^4]\!] \oplus [\![r^5]\!]) \\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}\} \oplus \{\{q^3\}\}) \otimes (\{\{\neg q^4\}\} \oplus \{\{r^5\}\}) \\ &= \{\{p^1\}\} \otimes \{\{\neg p^2\}, \{q^3\}\} \otimes \{\{\neg q^4\}, \{r^5\}\}) \\ &= \{\{p^1, \neg p^2\}, \{p^1, q^3\}\} \otimes \{\{\neg q^4\}, \{r^5\}\}) \\ &= \{\{p^1, \neg p^2, \neg q^4\}, \{p^1, q^3, \neg q^4\}, \\ &\qquad \{p^1, \neg p^2, r^5\}, \{p^1, q^3, r^5\}\}.\end{aligned}$$

Attenuating this matrix at the tag 3, we obtain the following paths:

$$\begin{aligned}&[\![p^1 \wedge (\neg p^2 \vee \top) \wedge (\neg q^4 \vee r^5)]\!] \\ &= [\![p^1]\!] \otimes ([\![\neg p^2]\!] \oplus [\![\top]\!]) \otimes ([\![\neg q^4]\!] \oplus [\![r^5]\!]) \\ &= \{\{p^1\}\} \otimes (\{\{\neg p^2\}\} \oplus \{\varnothing\}) \otimes (\{\{\neg q^4\}\} \oplus \{\{r^5\}\}) \\ &= \{\{p^1\}\} \otimes \{\{\neg p^2\}, \varnothing\} \otimes \{\{\neg q^4\}, \{r^5\}\}) \\ &= \{\{p^1, \neg p^2\}, \{p^1\}\} \otimes \{\{\neg q^4\}, \{r^5\}\}) \\ &= \{\{p^1, \neg p^2, \neg q^4\}, \{p^1, \neg q^4\}, \{p^1, \neg p^2, r^5\}, \{p^1, r^5\}\})\end{aligned}$$

Notice that it is possible to build a valuation satisfying $\{p^1, \neg q^4\}$ but not r. Therefore, it follows that via attenuation we have prevented the logical entailment of r.

In the subsequent development we will make use of iterated attenuations:

Definition 3.3. *If $I = \{i_1, i_2, \ldots, i_k\}$ is a finite set of tags and $\mathcal{A}$ is a matrix then the **attenuation of $\mathcal{A}$ by I** is defined as the iterated attenuation $(\ldots((\mathcal{A}_{i_1})_{i_2})\ldots)_{i_k}$.*

That this definition is well-defined follows from the following observation:

Observation 3.1. *If i and j are tags and $\mathcal{A}$ is a matrix then $(\mathcal{A}_i)_j = (\mathcal{A}_j)_i$ and $(\mathcal{A}_i)_i = \mathcal{A}_i$.*

Path-contraction operators will compute a contraction of $\mathcal{K}$ by $\mathcal{A}$ via attenuating a number of tags within a matrix $\mathcal{K}$ to obtain a matrix $\mathcal{K}_I$ which does not entail a matrix $\mathcal{A}$. We refer to these sets of tags I as *cuttings*.

Definition 3.4. *If $\mathcal{K}$ and $\mathcal{A}$ are tag-disjoint matrices then a **cutting** of $\mathcal{K}$ by $\mathcal{A}$ is either $\varnothing$ in the case $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$, or a subset I of $T(\mathcal{K})$ such that $\mathcal{K}_I \nvdash \mathcal{A}$ otherwise. It is a **regular cutting** iff every tag in I tags a variable in $\mathcal{K}$ which also appears in $\mathcal{A}$.*

Example 3.2. *In our previous example of $\mathcal{K} = p^1 \wedge (\neg p^2 \vee q^3) \wedge (\neg q^4 \vee r^5)$ it follows that $\{3\}$ is a non-regular cutting of $\mathcal{K}$ by r whereas $\{5\}$ is a regular cutting of $\mathcal{K}$ by r.*

We will see in Section 5 that working with regular cuttings results in an analogue of Parikh's Postulate being satisfied. It is conceivable that additional restrictions on cuttings may prove desirable. For instance, whenever I and J are sets of tags with $I \subseteq J$ then $\mathcal{K}_I \vdash \mathcal{K}_J$ and hence in the case $\mathcal{K}_I \nvdash \mathcal{A}$ it follows that $\mathcal{K}_J \nvdash \mathcal{A}$ as well. Seeking belief base contraction operators which result in minimal change, i.e. which preserve as many of the existing beliefs as possible, suggests that we should always prefer $\mathcal{K}_I$ to $\mathcal{K}_J$, in effect imposing a requirement that a cutting must be minimal with respect to set inclusion. However, this will increase the complexity of computing a cutting, and thus we do not take this to be a defining feature of our approach. We leave the question of additional restrictions on cuttings to the designers of concrete path-contraction operators.

Definition 3.5. *A binary function $- : L \times L \to L$ is a **path-contraction operator** iff for all satisfiable $\mathcal{K}$ and $\mathcal{A}$ it follows that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some cutting I for $\mathcal{A}$ in $\mathcal{K}$. It is **regular** in the case $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some regular cutting I.*

Note our restriction to satisfiable formulae in the prior definition. This is a matter of convenience. When $\mathcal{A}$ is unsatisfiable then $\mathcal{K} \vdash \mathcal{A}$ only holds when $\mathcal{K}$ is unsatisfiable as well, making the situation rather uninteresting. Further, when $\mathcal{K}$ is unsatisfiable, then it is implausible as a belief state for an agent anyways.

There are a number of desirable properties satisfied by path-contraction operators:

Theorem 3.2. *Suppose that $-$ is a path-contraction operator, then the following properties are satisfied:*

1. $\mathcal{K} \vdash \mathcal{K} - \mathcal{A}$.
2. *If* $\vdash \mathcal{A}$ *then* $\mathcal{K} - \mathcal{A} = \mathcal{K}$.
3. *If* $\mathcal{K} \nvdash \mathcal{A}$ *then* $\mathcal{K} - \mathcal{A} = \mathcal{K}$.
4. *If* $\nvdash \mathcal{A}$ *then* $\mathcal{K} - \mathcal{A} \nvdash \mathcal{A}$.

Proof.

1. It follows that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ and therefore $\mathcal{K} \Vdash \mathcal{K}_I = \mathcal{K} - \mathcal{A}$.
2. In this case $\vdash \mathcal{A}$ it follows by definition that $\varnothing$ is the only cutting of $\mathcal{K}$ by $\mathcal{A}$, and hence $\mathcal{K} - \mathcal{A} = \mathcal{K}_\varnothing = \mathcal{K}$.
3. In the case $\mathcal{K} \nvdash \mathcal{A}$ then $\varnothing$ is a cutting of $\mathcal{K}$ by $\mathcal{A}$. As $\varnothing \subseteq I$ for every tag set I in $\mathcal{K}$, it follows that $\varnothing$ is the only cutting of $\mathcal{K}$ by $\mathcal{A}$ and thus $\mathcal{K} - \mathcal{A} = \mathcal{K}_\varnothing = \mathcal{K}$ showing that $\mathcal{K} - \mathcal{A} \vdash \mathcal{K}$.
4. In the case $\nvdash \mathcal{A}$ let I be a cutting of $\mathcal{K}$ by $\mathcal{A}$ such that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$. Being that $\nvdash \mathcal{A}$ it follows that I is a minimal set of tags such that $\mathcal{K}_I \nvdash \mathcal{A}$, which is to say $\mathcal{K} - \mathcal{A} \nvdash \mathcal{A}$.

□

It follows that every path-contraction operator produces a belief base contraction operator in the following manner. Given κ and ϕ choose tag-disjoint matrices $\mathcal{K}$ and $\mathcal{A}$ such that κ is the erasure of $\mathcal{K}$ and ϕ is the erasure of $\mathcal{A}$. Apply the path-contraction operator to compute $\mathcal{K} - \mathcal{A}$, then define $\kappa - \phi$ as the erasure of $\mathcal{K} - \mathcal{A}$. It follows by Theorem 3.2 that the binary function $-$ on propositional formulae defined above satisfies the requirements to be a belief base contraction operator. Example 4.1 in the next section shows this in action.

4 An Algorithm for Regular Path-Contraction

In this section we will present an algorithm for implementing a regular path-contraction operator, and show that this algorithm results in a decision procedure for testing whether $\mathcal{K} - \mathcal{A}$ entails $\mathcal{B}$ with complexity **NP**. Our algorithm will make use of the notion of an *extracted path*, and the notion of a *cross-cut*, which we now present.

Definition 4.1. *If $\mathcal{A}$ is a satisfiable matrix with satisfying assignment ν, then the **extracted path** $\mathrm{ext}(\mathcal{A}, \nu)$ is defined by the following rules:*

1. *If $\mathcal{A}$ is $\top$ then* $\mathrm{ext}(\mathcal{A}, \nu) = \varnothing$.
2. *If $\mathcal{A}$ is p^i or $\neg p^i$ then* $\mathrm{ext}(\mathcal{A}, \nu) = \{\mathcal{A}\}$.
3. *If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ then* $\mathrm{ext}(\mathcal{A}, \nu) = \mathrm{ext}(\mathcal{B}, \nu) \cup \mathrm{ext}(\mathcal{C}, \nu)$.
4. *If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ and ν satisfies $\mathcal{B}$ then* $\mathrm{ext}(\mathcal{A}, \nu) = \mathrm{ext}(\mathcal{B}, \nu)$, *otherwise* $\mathrm{ext}(\mathcal{A}, \nu) = \mathrm{ext}(\mathcal{C}, \nu)$.

Lemma 4.1. *If ν is an assignment satisfying $\mathcal{A}$ then* $\mathrm{ext}(\mathcal{A}, \nu)$ *is an unconnected path through $\mathcal{A}$.*

Definition 4.2. *If $\mathfrak{p}$ and $\mathfrak{q}$ are unconnected paths then the **cross-cut**, denoted by $\mathrm{cr}(\mathfrak{p}, \mathfrak{q})$, is the set of tags i in $\mathfrak{p}$ for which there exists a tag j in $\mathfrak{q}$ alongside a propositional variable p such that either $\{p^i, \neg p^j\}$ or $\{\neg p^i, p^j\}$ is a connection in $\mathfrak{p} \cup \mathfrak{q}$.*

Lemma 4.2. *If $\mathcal{K} \vdash \mathcal{A}$, $\mathfrak{p}$ is an unconnected path through $\mathcal{K}$, and $\mathfrak{q}$ is an unconnected path through $\neg\mathcal{A}$ then $\mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$.*

Proof. Suppose $\mathfrak{p}$ is an unconnected path through $\mathcal{K}$ and $\mathfrak{q}$ is an unconnected path through $\neg\mathcal{A}$. It follows that $\mathfrak{r} = \mathfrak{p} \cup \mathfrak{q}$ is a path through $\mathcal{K} \wedge \neg\mathcal{A}$. By the assumption that $\mathfrak{p}$ and $\mathfrak{q}$ are unconnected, it follows that every connection in $\mathfrak{r}$ may be written as $\{\ell^i, \ell^j\}$ where ℓ^i is in $\mathcal{K}$ and ℓ^j is in $\neg\mathcal{A}$. Enumerating these connections as $\{\ell^{i_1}, \ell^{j_1}\}, \{\ell^{i_2}, \ell^{j_2}\}, \ldots, \{\ell^{i_n}, \ell^{j_n}\}$ it follows that $\mathrm{cr}(\mathfrak{p}, \mathfrak{q}) = \{i_1, i_2, \ldots, i_n\}$. Letting $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ it follows that $\mathfrak{r}_I$ is a unconnected path through $(\mathcal{K} \wedge \neg\mathcal{A})_I$. Being that I contains only tags appearing in $\mathcal{K}$ it follows that $(\mathcal{K} \wedge \neg\mathcal{A})_I = \mathcal{K}_I \wedge \neg\mathcal{A}$. Therefore $\mathfrak{r}_I$ is an unconnected path through $\mathcal{K}_I \wedge \neg\mathcal{A}$. By Theorem 2.1 it follows that $\mathcal{K}_I \wedge \neg\mathcal{A}$ is satisfiable, showing that $\mathcal{K}_I \nvdash \mathcal{A}$. If $\vdash \mathcal{A}$ then there would exist no unconnected path through $\neg\mathcal{A}$, contradicting our hypotheses. If $\mathcal{K} \nvdash \mathcal{A}$ this would also contradict our hypothesis. Hence, $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ is a cutting of $\mathcal{K}$ by $\mathcal{A}$. Furthermore, as every tag in I appears in $\mathcal{K}$, it follows that I is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$. □

Without further ado, we can present our algorithm as follows:

Algorithm 1 Path-Contraction Algorithm

```
Input: Initial beliefs $\mathcal{K}$
Input: Contractum $\mathcal{A}$
Output: The contracted beliefs $\mathcal{K} - \mathcal{A}$

 1: if $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$ then
 2:     return $\mathcal{K}$
 3: else
 4:     $\nu :=$ a satisfying assignment for $\mathcal{K}$
 5:     $\mu :=$ a satisfying assignment for $\neg\mathcal{A}$
 6:     $\mathfrak{p} := \mathrm{ext}(\mathcal{K}, \nu)$
 7:     $\mathfrak{q} := \mathrm{ext}(\neg\mathcal{A}, \mu)$
 8:     $I := \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$
 9:     return $\mathcal{K}_I$
10: end if
```

Depending on the satisfiability solver used, this algorithm will produce different path-contraction operators. Further, if the satisfiability solver is non-deterministic, as many systems with randomised restarts are, then the path-contraction operator produced will be non-deterministic. However, fixing a deterministic satisfiability solver, we obtain a deterministic algorithm. Before turning to the correctness and complexity of this algorithm, we present the following example:

Example 4.1. *Consider a propositional vocabulary where p symbolises that Tweety is a penguin, b symbolises that Tweety is a bird, and f symbolises that Tweety flies. Consider the naive knowledge base κ defined as $(p \to b) \wedge (b \to f)$ which has the undesirable consequence $\phi := (p \to f)$ suggesting that were Tweety a penguin then Tweety could fly. We will use our path-contraction algorithm to compute $\kappa - \phi$. First we tag everything to obtain the matrices $\mathcal{K} := (\neg p^1 \vee b^2) \wedge (\neg b^3 \vee f^4)$ and $\mathcal{A} := (\neg p^5 \vee f^6)$.*

As $\mathcal{K} \vdash \mathcal{A}$ and $\nvdash \mathcal{A}$ our algorithm must do some work. We start by choosing a truth-value assignment ν satisfying $\mathcal{K}$ such as the one satisfying $p \wedge b \wedge f$, alongside a truth-value assignment μ satisfying $\neg\mathcal{A} = p^5 \wedge \neg f^6$ such as the one satisfying $p \wedge b \wedge \neg f$. Using these assignments, we extract the path $\mathfrak{p} = \{b^2, f^4\}$ through $\mathcal{K}$ alongside the path $\mathfrak{q} = \{p^5, \neg f^6\}$ through $\neg\mathcal{A}$.

Computing the cross-cut, we find $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q}) = \{4\}$. Our algorithm now returns $\mathcal{K}_I = (\neg p^1 \vee b^2) \wedge (\neg b^3 \vee \top) = (\neg p^1 \vee b^2)$. Erasing the tags gives the new knowledge base $\neg p \vee b$.

Theorem 4.1. *Algorithm 1 defines a regular path-contraction operator.*

Proof. For any satisfiable matrices $\mathcal{K}$ and $\mathcal{A}$ let $\mathcal{K} - \mathcal{A}$ be defined as the result returned by Algorithm 1. This is well-defined as Algorithm 1 is deterministic once we fix deterministic satisfiability solvers, and always terminates regardless. We must show that $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ for some regular cutting of $\mathcal{K}$ by $\mathcal{A}$. In the case $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$ then, by definition, $I = \varnothing$ is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$, and furthermore the algorithm returns $\mathcal{K} = \mathcal{K}_I$. Otherwise, we proceed under the assumption $\mathcal{K} \vdash \mathcal{A}$ and $\nvdash \mathcal{A}$. It follows that there exists a truth value assignment ν satisfying $\mathcal{K}$ as well as a truth value assignment μ satisfying $\neg\mathcal{A}$. By Lemma 4.1 it follows that $\mathfrak{p} = \mathrm{ext}(\nu, \mathcal{K})$ is an unconnected path through $\mathcal{K}$, and $\mathfrak{q} = \mathrm{ext}(\mu, \neg\mathcal{K})$ is an unconnected path through $\mathcal{A}$. Further, recalling our assumption that $\mathcal{K} \vdash \mathcal{A}$ it follows by Lemma 4.2 that $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ is a regular cutting of $\mathcal{K}$ by $\mathcal{A}$. As the algorithm returns with $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ this is also fine. Hence, it follows that $-$ is a regular path-contraction operator. □

Theorem 4.2. *If $-$ is the regular path-contraction operator defined by Algorithm 1 then the decision problem of determining whether or not $\mathcal{K} - \mathcal{A} \vdash \mathcal{B}$ has worst-case time complexity **NP**.*

Proof. Let n be the sum of the size of $\mathcal{K}$, the size of $\mathcal{A}$, and the number of variables in the language. We start by computing $\mathcal{K} - \mathcal{A}$. Initially the algorithm calls into a satisfiability solver twice on line (1) to determine whether $\mathcal{K} \nvdash \mathcal{A}$ or $\vdash \mathcal{A}$ hold. Assuming this is not the case, it then calls into a satisfiability solver twice to produce the satisfying assignments ν and μ on lines (4) and (5). Recursing over the matrices $\mathcal{K}$ and $\mathcal{A}$ and labelling every subformula with the value it is assigned by ν and μ respectively requires only $O(n \log n)$ time, and after we can then compute $\mathfrak{p} = \mathrm{ext}(\nu, \kappa)$ and $\mathfrak{q} = \mathrm{ext}(\mu, \neg\phi)$ on lines (6) and (7) using only $O(n)$ time in the size of the formula. Computing $I = \mathrm{cr}(\mathfrak{p}, \mathfrak{q})$ on line (8) can be done in time $O(n^2)$, and computing $\mathcal{K}_I$ on line (9) can be done in time $O(n)$. With $\mathcal{K} - \mathcal{A} = \mathcal{K}_I$ computed, we now call into a satisfiability solver one last time to determine whether $\mathcal{K} - \mathcal{A} \vdash \mathcal{B}$. In total, we have done a polynomial-time amount of work in addition to solving 5 instances of the satisfiability problem. As satisfiability has worst-case time complexity **NP**, it follows that our decision problem has worst-case time complexity **NP** as well. □

The existence of a regular path-contraction operator with a worst-case time complexity of **NP** is in sticking contrast to other concrete belief change functions discussed in (Eiter and Gottlob 1992) whose complexity is often Π^2_p-complete. Though, it must be pointed out that path-contraction operators comprise a class of operators of which Algorithm 1 contributes only a small subset, and it may ultimately prove desirable to sacrifice some additional performance in order to ensure stronger guarantees over the properties of the overall path-contraction operator.

5 Path-Entailment and Path-Independence

In this section we introduce the notion of *path-entailment* which strengthens logical entailment to a structural property of matrices. We will see that every matrix path-entails its attenuations, and further that attenuation preserves the path-entailment of matrices not containing the attenuated tag. Using path-entailment, we will be able to characterise further the preservation properties of regular path-contraction operators.

Definition 5.1. *If $\mathcal{A}$ and $\mathcal{B}$ are matrices such that for every path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ there exists a path $\mathfrak{q} \in [\![\mathcal{B}]\!]$ with $\mathfrak{p} \supseteq \mathfrak{q}$ then $\mathcal{A}$ **path-entails** $\mathcal{B}$, which we indicate by writing $\mathcal{A} \Vdash \mathcal{B}$. We also say that $\mathcal{B}$ is a **path-consequence** of $\mathcal{A}$.*

Example 5.1. *Consider the matrix $\mathcal{K}$ defined as $(\neg p^1 \vee q^2) \wedge (\neg q^3 \vee r^4)$. The paths through $\mathcal{K}$ are as follows: $\{\neg p^1, \neg q^3\}$, $\{\neg p^1, r^4\}$, $\{q^2, \neg q^3\}$, and finally $\{q^2, r^4\}$. As the paths through $\neg p^1 \vee q^2$ are $\{\neg p^1\}$ and $\{q^2\}$ it follows that $\mathcal{K}$ path-entails $\neg p^1 \vee q^2$, as each path through $\mathcal{K}$ contains either $\neg p^1$ or q^2. However, despite $\neg p^1 \vee r^4$ being logically entailed by $\mathcal{K}$, this is not a path-consequence of $\mathcal{K}$ for the reason that the path $\{q^2, \neg q^3\}$ contains no path through $\neg p^1 \vee r^4$.*

Theorem 5.1. *If $\mathcal{A} \Vdash \mathcal{B}$ then $\mathcal{A} \vdash \mathcal{B}$.*

Proof. Suppose that $\mathcal{A} \Vdash \mathcal{B}$, and consider an interpretation ν which satisfies $\mathcal{A}$. By Lemma 2.1 this means that there exists a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$ such that ν satisfies every element of $\mathfrak{p}$. Under our assumption that $\mathcal{A} \Vdash \mathcal{B}$ it follows that there exists some path $\mathfrak{q} \in [\![\mathcal{B}]\!]$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. However, this means that ν satisfies every element of $\mathfrak{q}$, which by Lemma 2.1 implies ν satisfies $\mathcal{B}$. With ν being arbitrary, it follows that $\mathcal{A} \vdash \mathcal{B}$. □

Theorem 5.2.

1. *$\mathcal{A} \Vdash \mathcal{A}$.*
2. *If $\mathcal{A} \Vdash \mathcal{B}$ and $\mathcal{B} \Vdash \mathcal{C}$ then $\mathcal{A} \Vdash \mathcal{C}$.*
3. *If $\mathcal{A} \Vdash \mathcal{C}$ and $\mathcal{B}$ is tag-disjoint with $\mathcal{A}$ then $\mathcal{A} \wedge \mathcal{B} \Vdash \mathcal{C}$.*

4. *If $\mathcal{A}$ is a matrix and i is a tag then $\mathcal{A} \Vdash \mathcal{A}_i$.*
5. *If $\mathcal{A} \Vdash \mathcal{B} \wedge \mathcal{C}$ then $\mathcal{A} \Vdash \mathcal{B}$.*
6. *If $\mathcal{A} \Vdash \mathcal{B}$ then $\mathcal{A} \Vdash \mathcal{B} \vee \mathcal{C}$.*

Proof.

1. Immediate.
2. Suppose $\mathcal{A} \Vdash \mathcal{A}$ and $\mathcal{A} \Vdash \mathcal{A}$. Suppose that $\mathfrak{p} \in [\![\mathcal{A}]\!]$ and observe that $\mathcal{A} \Vdash \mathcal{B}$ implies there exists some $\mathfrak{q} \in [\![\mathcal{B}]\!]$ such that $\mathfrak{p} \supseteq \mathfrak{q}$, and furthermore observe that $\mathcal{B} \Vdash \mathcal{C}$ implies there exists some $\mathfrak{r} \in [\![\mathcal{C}]\!]$ with $\mathfrak{q} \supseteq \mathfrak{r}$. By transitivity it follows that $\mathfrak{p} \supseteq \mathfrak{r}$. With $\mathfrak{p}$ being arbitrary, it follows that $\mathcal{A} \Vdash \mathcal{C}$.
3. Suppose $\mathfrak{p}$ is a path through $\mathcal{A} \wedge \mathcal{B}$ and observe there exist paths $\mathfrak{q}_1 \in [\![\mathcal{A}]\!]$ and $\mathfrak{q}_2 \in [\![\mathfrak{q}]\!]$ with $\mathfrak{p} = \mathfrak{q}_1 \cup \mathfrak{q}_2$. As $\mathcal{A} \Vdash \mathcal{C}$ there exists a path $\mathfrak{r} \in [\![\mathcal{C}]\!]$ such that $\mathfrak{q}_1 \supseteq \mathfrak{r}$. As $\mathfrak{p} = \mathfrak{q}_1 \cup \mathfrak{q}_2$ it follows that $\mathfrak{p} \supseteq \mathfrak{r}$. With $\mathfrak{p}$ being arbitrary, it then follows that $\mathcal{A} \wedge \mathcal{B} \Vdash \mathcal{C}$.
4. Suppose $\mathfrak{p}$ is a path through $\mathcal{A}$, then it follows that $\mathfrak{p} \supseteq \mathfrak{p}_i$ and $\mathfrak{p}_i \in [\![\mathcal{A}_i]\!]$. Hence, $\mathcal{A} \Vdash \mathcal{A}_i$.
5. Suppose $\mathcal{A} \Vdash \mathcal{B} \wedge \mathcal{C}$ and consider a path $\mathfrak{p} \in [\![\mathcal{A}]\!]$. It follows that there exists a path $\mathfrak{q} \in [\![\mathcal{B} \wedge \mathcal{C}]\!]$ such that $\mathfrak{p} \supseteq \mathfrak{q}$, however as $[\![\mathcal{B} \wedge \mathcal{C}]\!]$ it follows that $\mathfrak{q} = \mathfrak{q}_1 \cup \mathfrak{q}_2$ for some $\mathfrak{q}_1 \in [\![\mathcal{B}]\!]$ and $\mathfrak{q}_2 \in [\![\mathcal{B}_2]\!]$ showing that $\mathfrak{p} \supseteq \mathfrak{q}_1$ where $\mathfrak{q}_1 \in [\![\mathfrak{q}_1]\!]$. With $\mathfrak{p}$ being arbitrary, it then follows that $\mathcal{A} \Vdash \mathcal{B}$.
6. Suppose $\mathfrak{p}$ is a path through $\mathcal{A}$. It follows from the assumption that $\mathcal{A} \Vdash \mathcal{B}$ then there exists a path $\mathfrak{q} \in [\![\mathcal{B}]\!]$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. As $[\![\mathcal{B} \vee \mathcal{C}]\!] = [\![\mathcal{B}]\!] \cup [\![\mathcal{C}]\!]$, this implies that there exists a path $\mathfrak{q} \in [\![\mathcal{B} \vee \mathcal{C}]\!]$ with $\mathfrak{p} \supseteq \mathfrak{q}$. Hence, $\mathcal{A} \Vdash \mathcal{B} \vee \mathcal{C}$.

□

Note that properties (1), (2), and (3) of Theorem 5.2 correspond to the requirements for path-entailment to be a Tarskian consequence relation, modulo the proviso of the tag-disjointness for (3). Regardless, the motivation for studying path-entailment is that we can easily formulate a criterion for attenuation to preserve an individual path-entailment, whereas in the case of logical entailment the situation is not straightforward.

Theorem 5.3. *If $\mathcal{A} \Vdash \mathcal{B}$ and i is an tag not occurring in $\mathcal{B}$ then $\mathcal{A}_i \Vdash \mathcal{B}$.*

Proof. Suppose that $\mathfrak{p} \in [\![\mathcal{A}_i]\!]$. Then there exists a path $\mathfrak{p}' \in [\![\mathcal{A}]\!]$ such that $\mathfrak{p} = \mathfrak{p}'_i$, and as $\mathcal{A} \Vdash \mathcal{B}$ there exists a path $\mathfrak{q} \in [\![\mathcal{B}]\!]$ such that $\mathfrak{p} \supseteq \mathfrak{q}$. However, as i does not occur in $\mathcal{B}$ it follows that $\mathfrak{p} = \mathfrak{p}'_i \supseteq \mathfrak{q}$. With $\mathfrak{p}$ being arbitrary, it follows that $\mathcal{A}_i \Vdash \mathcal{B}$. □

As a corollary of Theorem 5.3, we will see that regular path-contraction operators satisfy a structural analogue of Parikh's postulate. Rather than consider formulae κ logically equivalent to some conjunction $\kappa_1 \wedge \kappa_2$ with $V(\kappa_1) \cap V(\kappa_2) = \varnothing$, we will consider matrices $\mathcal{K}$ which are path-equivalent to a conjunction $\mathcal{K}_1 \wedge \mathcal{K}_2$ with $V(\mathcal{K}_1) \cap V(\mathcal{K}_2) = \varnothing$. This is a stronger requirement, and as such this version of Parikh's Postulate will apply less frequently. Though, this is not unreasonable given our position on the relevance of syntax.

Definition 5.2. *If X and Y are disjoint subsets of V with $V = X \cup Y$ then X and Y are **path-independent modulo** $\mathcal{K}$ iff there exist tag-disjoint matrices $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $V(\mathcal{K}_1) \subseteq X$, $V(\mathcal{K}_2) \subseteq Y$, and $[\![\mathcal{K}]\!] = [\![\mathcal{K}_1 \wedge \mathcal{K}_2]\!]$. In this case we say that $(\mathcal{K}_1, \mathcal{K}_2)$ is a (X, Y)-**splitting** of $\mathcal{K}$.*

Example 5.2. *Although the matrices $p^1 \wedge (\neg p^2 \vee q^3)$ and $p^1 \wedge q^3$ are logically equivalent, and it follows that $\{p\}$ and $\{q\}$) are path-independent modulo $p^1 \wedge q^3$, it follows that $[\![p^1 \wedge (\neg p^2 \vee q^3)]\!] = \{\{p^1, \neg p^2\}, \{p^1, q^3\}\}$ cannot be expressed as $[\![\mathcal{K}_1 \wedge \mathcal{K}_2]\!] = [\![\mathcal{K}_1]\!] \otimes [\![\mathcal{K}_2]\!]$ for any $\mathcal{K}_1$ and $\mathcal{K}_2$ with $V(\mathcal{K}_1) = \{p\}$ and $V(\mathcal{K}_2) = \{q\}$ showing that $\{p\}$ and $\{q\}$ are not path-independent modulo $p^1 \wedge (\neg p^2 \vee q^3)$.*

Theorem 5.4. *Suppose that $-$ is a regular path-contraction operator, $(\mathcal{K}_1, \mathcal{K}_2)$ is an (X, Y)-splitting of $\mathcal{K}$, and that $\mathcal{A}$ is a formula with $V(\mathcal{A}) \subseteq Y$, then it follows that*

$$(\mathcal{K}_1 \wedge \mathcal{K}_2) - \mathcal{A} \Vdash \mathcal{K}_2.$$

Proof. As $-$ is a regular path-contraction operator it follows that there exists a regular cutting I of $(\mathcal{K}_1 \wedge \mathcal{K}_2)$ by $\mathcal{A}$ such that $(\mathcal{K}_1 \wedge \mathcal{K}_2) - \mathcal{A} = (\mathcal{K}_1 \wedge \mathcal{K}_2)_I$. As $V(\mathcal{A}) \cap V(\mathcal{K}_2) = \varnothing$ it follows that every tag i in I tags a variable in $\mathcal{A}$ which does not appear in $\mathcal{K}_2$. Hence, as $\mathcal{K}_1 \wedge \mathcal{K}_2 \Vdash \mathcal{K}_2$ it follows from Theorem 5.2 that $(\mathcal{K}_1 \wedge \mathcal{K}_2) - \mathcal{A} = (\mathcal{K}_1 \wedge \mathcal{K}_2)_I \Vdash \mathcal{K}_2$. □

6 Discussion

6.1 Related Work

There are attempts to leverage the connection method and similar techniques for belief contraction already in literature, however we believe our work to be unique in utilising matrix attenuation to preserve the structure of the original formula.

In (Bienvenu, Herzig, and Qi 2008), knowledge bases are converted into prime implicate normal form, resulting in a syntax-independent but nevertheless syntactic belief revision function.

In (Schwind 2010) and (Schwind 2012) belief revision functions are introduced which operate on implicants, which are taken there to be roughly paths through matrices with tags erased. These functions are required to satisfy the AGM postulates, and thus correspond to belief revision rather than belief base revision.

In (Gabbay, Rodrigues, and Russo 2010) a formulation of the connection method for knowledge bases in disjunctive normal form is utilised to repair inconsistent knowledge bases. This is accomplished by replacing the knowledge base with the disjunction formed by the conjunctions associated with each maximally consistent subset of a path. This can be further adapted to the process of belief revision by assigning priorities to the different tags, and selecting maximal consistent subpaths which retain the highest priority tags if at all possible. Our approach differs in that we do not require conversion to disjunctive normal form, we produce a modification to the original formula, our approach performs belief contraction, and our approach results in a class of path-contraction operators.

6.2 Future Work

Clarifying the connection between path-contraction operators and other approaches to belief base contraction via hitting sets and incision functions, as well as attempting to obtain versions of properties such as core-retainment suitable for path-contraction remains an open problem. We believe that investigating "path-remainders" of $\mathcal{K}$ modulo $\mathcal{A}$, defined as those logically strongest $\mathcal{K}'$ such that $\mathcal{K} \Vdash \mathcal{K}'$ yet $\mathcal{K}' \nvdash \mathcal{A}$ will prove illuminating.

Variants of the connection method have been developed for intuitionistic and modal logics (Wallen 1987), as well as for the description logic $\mathcal{ALC}$ (Freitas and Otten 2016). We believe that the theory of path-contraction operators introduced here will generalise well to these formalisms. It would also be interesting to investigate whether this approach also extends to tableaux methods for non-monotonic logics (Olivetti 1999) such as sceptical default reasoning (Bonatti and Olivetti 1997b) or circumscription (Bonatti and Olivetti 1997a).

We are also interested in conducting an empirical study of the performance of path-contraction operators. In order to carry out path-contraction, an algorithm would essentially have to locate an open branch in a tableau for $\mathcal{K}$, an open branch in a tableau for $\mathcal{A}$, select a minimal cutting based off these branches, and then perform a quadratic amount of work to attenuate by that cutting in order to produce $\mathcal{K} - \mathcal{A}$. Therefore, the majority of the complexity arises in locating those open branches, and ensuring that resulting cutting is minimal. Hence, we suspect that the performance of path-contraction should be comparable to tableaux-based satisfiability checking. It would also be interesting to obtain some probabilistic estimates on the likelihood of obtaining a cutting via randomly sampling paths, as a random path should be able to be sampled from $\mathcal{K} \wedge \neg\mathcal{A}$ in linear time. Finally, we conjecture that path-entailment can be decided in polynomial-time.

7 Conclusion

In this paper we introduced the class of path-contraction operators, which utilise the process of matrix attenuation to carry out a form of belief base contraction in a manner which leaves the syntactic structure of the original belief base minimally changed. We have presented an algorithm for implementing a path-contraction operator and shown it to have complexity **NP**. We have further introduced the notion of path-entailment, and shown that regular path-contraction operators have desirable preservation properties which substantiate the claim that path-contraction operators are carrying out only minimal changes to the original formula. Finally, we discussed where our approach fits in with other related approaches to belief change.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 510–530.

Andrews, P. B. 1976. Refutations by matings. *IEEE Computer Architecture Letters* 25(08):801–807.

Bibel, W. 1981. On matrices with connections. *Journal of the ACM (JACM)* 28(4):633–645.

Bienvenu, M.; Herzig, A.; and Qi, G. 2008. Prime implicate-based belief revision operators. In *ECAI*, volume 178, 741–742. Citeseer.

Bonatti, P. A., and Olivetti, N. 1997a. A sequent calculus for circumscription. In *International Workshop on Computer Science Logic*, 98–114. Springer.

Bonatti, P. A., and Olivetti, N. 1997b. A sequent calculus for skeptical default logic. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 107–121. Springer.

Caridroit, T.; Konieczny, S.; and Marquis, P. 2017. Contraction in propositional logic. *International Journal of Approximate Reasoning* 80:428–442.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial intelligence* 57(2-3):227–270.

Freitas, F., and Otten, J. 2016. A connection calculus for the description logic $\mathcal{ALC}$. In *Canadian Conference on Artificial Intelligence*, 243–256. Springer.

Gabbay, D. M.; Rodrigues, O. T.; and Russo, A. 2010. *Revision, Acceptability and Context: Theoretical and Algorithmic Aspects*. Springer Science & Business Media.

Hansson, S. O. 2012. *A textbook of belief dynamics: Theory change and database updating*. Springer.

Katsuno, H., and Mendelzon, A. O. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(3):263–294.

Kreitz, C., and Otten, J. 1999. Connection-based theorem proving in classical and non-classical logics. *Journal of Universal Computer Science* 5(3):88–112.

Levesque, H. J. 1984. A logic of implicit and explicit belief. In *AAAI*, 198–202.

Olivetti, N. 1999. Tableaux for nonmonotonic logics. In *Handbook of Tableau Methods*. Springer. 469–528.

Otten, J. 2011. A non-clausal connection calculus. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, 226–241. Springer.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. *Logic, Language and Computation* 2(96):266–268.

Schwind, C. 2010. From inconsistency to consistency: Knowledge base revision by tableaux opening. In *Ibero-American Conference on Artificial Intelligence*, 120–132. Springer.

Schwind, C. 2012. Belief base revision as a binary operation on implicant sets: a finitary approach.

Wallen, L. A. 1987. Automated proof search in non-classical logics: efficient matrix proof methods for modal and intuitionistic logics.