Transforming Conditional Knowledge Bases into Renaming Normal Form

Christoph Beierle, Jonas Haldimann
FernUniversität in Hagen
58084 Hagen, Germany
{christoph.beierle, jonas.haldimann}@fernuni-hagen.de

Abstract

While for classical logics, the motto “Truth is invariant under the change of notation” has been studied extensively, less attention has been paid to this aspect in defeasible logics. In this paper, we address equivalences and transformations among conditional knowledge bases that take renamings into account. Extending previous proposals, we introduce the concepts of renaming normal form and renaming antecedent normal form for arbitrary knowledge bases and across different signatures. We present procedures to transform every knowledge base to corresponding, up to propositional normalization uniquely determined normal forms, and we study their properties. Using the obtained normal forms allows for systematically identifying equivalences among knowledge bases, for easier and more transparent comparisons, and for simplified descriptions of algorithms operating on knowledge bases by avoiding tedious, but uninteresting borderline cases.

1 Introduction

Electric cars do not need fossil fuel, although there might be exceptions, e.g., hybrid vehicles. Such a defeasible relationship is conveniently expressed by a conditional of the form If $A$ then usually $B$, formally denoted by $(B|A)$. A set of conditionals is a knowledge base that might express the explicit knowledge of an agent about a specific domain.

Example 1 ((Beierle, Eichhorn, and Kern-Isberner 2017)). $R_{e-car}$ is a knowledge base about e-cars containing:

- $(f|e)$ “Usually cars need fossil fuel.”
- $(\neg f|e)$ “Usually e-cars do not need fossil fuel.”
- $(e|e)$ “E-cars usually are cars.”
- $(e|\neg e)$ “E-cars not needing fossil fuel usually are e-cars.”
- $(\neg e|e)$ “E-cars usually are e-cars not needing fossil fuel.”
- $(\neg \top|\top)$ “Usually things are not e-cars.”
- $(e|\lor \neg f|e \lor e)$ “Things that are cars and e-cars or cars but not e-cars are cars that need fossil fuel or are not cars but need fossil fuel.”

Conditional knowledge bases and different semantic approaches have been defined for them (cf. (Adams 1975; Lewis 1973; Kraus, Lehmann, and Magidor 1990; Pearl 1990; Lehmann and Magidor 1992; Dubois and Prade 1994; Goldszmidt and Pearl 1996; Benferhat, Dubois, and Prade 1999; Kern-Isberner 2001)). In any of these approaches, there are typically many different possibilities to express the same semantic meaning. For easier comparisons, avoidance of cumbersome but uninteresting case distinctions, and for less complex descriptions of algorithms dealing with knowledge bases, normal forms are desirable. Here, we address normal forms for conditional knowledge bases that take renamings into account. Normal forms for conditional knowledge bases have been proposed in e.g. (Beierle, Eichhorn, and Kern-Isberner 2017; Beierle 2019) and in (Beierle and Kutsch 2019; Beierle and Haldimann 2020) with a focus on the systematic generation of knowledge bases in normal form; in particular, no general definition of renaming normal form and no algorithm transforming a knowledge base into renaming normal form is given there. The main contributions of this paper are:

- Extend the notion of renaming normal form ($\rho$NF) given in (Beierle and Haldimann 2020) only for normal form conditionals to arbitrary knowledge bases.
- Define a new unique renaming antecedent normal form ($\rho\text{ANF}$) ensuring model equivalence up to renamings.
- Define renamings, $\rho$NF, and $\rho\text{ANF}$ across different signatures.
- Propose Algorithms $\Theta^n$ and $\Theta^{an}$ transforming every $R$ into $\rho$NF and $\rho\text{ANF}$ and their properties. After recalling the required basics in Sec. 2, renamings and $\rho$NF are introduced in Section 3. Section 4 presents the algorithm transforming $R$ into its $\rho$NF, and Section 5 introduces the $\rho\text{ANF}$ and how it is obtained. In Sec. 6 we list conclusions and point out future work.

2 Background: Conditional Logic

Let $\mathcal{L}(\Sigma)$ be a propositional language over a finite signature $\Sigma$. Unless otherwise stated, $\Sigma$ consists of atoms $a, b, c, \ldots$. We call a signature $\Sigma$ with a linear ordering $\prec$ an ordered signature and denote it by $(\Sigma, \prec)$. The language may be denoted by $\mathcal{L}$ if the signature is clear from context. The formulas of $\mathcal{L}$ will be denoted by letters $A, B, C, \ldots$. We write
for $A \land B$ and $\overline{A}$ for $\neg A$. We identify the set of all complete conjunctions over $\Sigma$ with the set $\Omega$ of possible worlds over $\mathcal{L}$. For $\omega \in \Omega$, $\omega \models A$ means that $A \in \mathcal{L}$ holds in $\omega$, and the set of worlds satisfying $A$ is $\Omega_A = \{ \omega \mid \omega \models A \}$. Two formulas $A, B$ are equivalent, denoted as $A \equiv B$, if $\Omega_A = \Omega_B$. By introducing a new binary operator $\equiv$, we obtain the set $(\mathcal{L}_\equiv \mathcal{L}) = \{ (B|A) \mid A, B \in \mathcal{L}(\mathcal{L}) \}$ of conditionals over $\mathcal{L}(\mathcal{L})$. Again, $\equiv$ may be omitted. For a conditional $r = (B|A)$, $\operatorname{ant}(r) = A$ is the antecedent of $r$, and $\operatorname{con}(r) = B$ is its consequent. The counter conditional of $r = (B|A)$ is $\overline{r} = (\overline{B}|\overline{A})$. As semantics for conditionals, we use ordinal functional conditions (OCF) (Spohn 2012). An OCF is a function $\kappa : \Omega \rightarrow \mathbb{N}$ expressing degrees of plausibility of possible worlds where a lower degree denotes "less surprising". At least one world must be regarded as being normal; therefore, $\kappa(\emptyset) = 0$ for at least one $\omega \in \Omega$. Each $\kappa$ uniquely extends to a function mapping sentences to $\mathbb{N} \cup \{ \infty \}$ given by $\kappa(A) = \min \{ \kappa(\omega) \mid \omega \models A \}$ where $\min \emptyset = \infty$. An OCF $\kappa$ accepts a conditional $(B|A)$, written $\kappa \models (B|A)$, if the verification of the conditional is less surprising than its falsification, i.e., if $\kappa(AB) < \kappa(A\overline{B})$; equivalently, $\kappa \models (B|A)$ iff for every $\omega' \in \Omega_{\overline{1}1}$ there is $\omega \in \Omega_{1\overline{1}}$ with $\kappa(\omega) < \kappa(\omega')$. A conditional $(B|A)$ is trivial if it is self-fulfilling $(A \equiv B)$ or contradictory $(A \equiv \overline{B})$; a set of conditionals is self-fulfilling if every conditional in it is self-fulfilling. A finite set $\mathcal{R} \subseteq (\mathcal{L}_\equiv \mathcal{L})$ of conditionals is called a knowledge base. An OCF $\kappa$ accepts $\mathcal{R}$ if $\kappa$ accepts all conditionals in $\mathcal{R}$, and $\mathcal{R}$ is consistent if an OCF accepting $\mathcal{R}$ exists (Goldszmidt and Pearl 1996). We use $\equiv$ to denote an inconsistent knowledge base. $\operatorname{Mod}(\mathcal{R})$ denotes the set of all OFCs $\kappa$ accepting $\mathcal{R}$. Two knowledge bases $\mathcal{R}, \mathcal{R}'$ are model equivalent, denoted by $\mathcal{R} \equiv_{mod} \mathcal{R}'$, if $\operatorname{Mod}(\mathcal{R}) = \operatorname{Mod}(\mathcal{R}')$. We say $(B|A) \equiv (B'|A')$ if $A \equiv A'$ and $AB \equiv A'B'$ where $\equiv$ is propositional equivalence.

3 Renamings and Renaming Normal Form

There are knowledge bases that are identical except for the names of their variables. E.g., the knowledge bases $\mathcal{R}_1 = \{(a,b), (a,c)\}$ and $\mathcal{R}_2 = \{(c,b), (c,a)\}$ become equal if we swap the names for the variables $a$ and $c$ in one of them. When analysing the structure of a knowledge base, we are only interested in one of such knowledge bases that are identical except for a signature renaming.

Definition 2 (renaming, $\simeq$, $\equiv_{mod}$). Let $\Sigma, \Sigma'$ be signatures. We call a bijective function $\rho : \Sigma \rightarrow \Sigma'$ a (signature) renaming. A renaming is lifted canonically to formulas, worlds, conditionals, knowledge bases, and sets thereof as usual. Two worlds, formulas, conditionals, knowledge bases, or sets thereof are equivalent under signature renaming, denoted as $X \simeq X'$, if there exists a renaming $\rho$ such that $X' = \rho(X)$. Two knowledge bases $\mathcal{R}, \mathcal{R}'$ over $\Sigma, \Sigma'$ are model equivalent up to renamings, denoted $\mathcal{R} \equiv_{mod} \mathcal{R}'$, if there is a renaming $\rho : \Sigma' \rightarrow \Sigma$ such that $\mathcal{R} \equiv_{mod} \rho(\mathcal{R}')$.

Note that $\mathcal{R} \simeq \mathcal{R}'$ implies $\mathcal{R} \equiv_{mod} \mathcal{R}'$, but not vice versa; but each condition ensures inferential equivalence up to renamings: For formulas $A, B$ let $A \vdash_{\mathcal{R}} B$ denote that $\kappa \models (B|A)$ holds for all $\kappa \in \operatorname{Mod}(\mathcal{R})$; thus $\vdash_{\mathcal{R}}$ corresponds to system $P$ inference (Adams 1975; Lehmann and Magidor 1992) based on $\mathcal{R}$. Then we have:

**Proposition 3** ($\vdash_{\mathcal{R}}$ and renamings). If $\mathcal{R} \equiv_{mod} \rho(\mathcal{R}')$ then $A \vdash_{\mathcal{R}} B$ iff $\rho(A) \vdash_{\mathcal{R}} \rho(B)$.

In order to be able to deal with normal forms of formulas in $\mathcal{L}$ without having to select a specific representation, we assume a function $\nu$ mapping a propositional formula $A$ to a unique normal form $\nu(A)$ such that $A \equiv A'$ iff $\nu(A) = \nu(A')$.

**Definition 4** ($=_{\nu}$). Two propositional formulas $A, A'$ are equal under normalization, denoted as $A \equiv_{\nu} A'$, if $\nu(A) = \nu(A')$. This equivalence is lifted canonically to sets of formulas. Two conditionals $(B|A), (B'|A')$ are equal under normalization, if $A \equiv_{\nu} A'$ and $B \equiv_{\nu} B'$. This equivalence is lifted canonically to sets of conditionals.

Note that the definition of the normalization function implies that for formulas $A, B$ we have $A \equiv_{\nu} B$ iff $A \equiv B$.

**Definition 5** ($\simeq_{\nu}$). Two worlds or sets thereof are equivalent under signature renaming and normalization, denoted as $\omega \simeq_{\nu} \omega'$, if $\omega \equiv_{\nu} \omega'$. Two formulas $A, A'$ are equivalent under signature renaming and normalization, denoted as $A \simeq_{\nu} A'$, if there exists a renaming $\rho$ such that $\rho(A) = \nu(A')$. This equivalence is lifted canonically to sets of formulas. Two conditionals $(B|A), (B'|A')$ are equivalent under signature renaming and normalization, if there exists a renaming $\rho$ such that $\rho(A) = \nu(A')$ and $\rho(B) = \nu(B')$. This equivalence is lifted canonically to sets of conditionals.

In contrast to renaming, normalization might change the size of a knowledge base, as the following example shows.

**Example 6**. Let $\mathcal{R} = \{ (b|a), (b|a \lor a) \}$ and $\mathcal{R}' = \{ (b|a) \}$. As both conditionals in $\mathcal{R}$ are equivalent, they will be mapped to the same normal form by $\nu$. Therefore, $\mathcal{R} = \nu$, $\mathcal{R}'$ although $|\mathcal{R}| \neq |\mathcal{R}'|$.

For a set $M, m \in M$, and an equivalence relation $\equiv$ on $M$, the set of equivalence classes induced by $\equiv$ is denoted by $[M]_{\equiv}$, and the unique equivalence class containing $m$ is denoted by $[m]_{\equiv}$. It is easy to see that equivalence under signature renaming $\simeq$ and equivalence under signature renaming and normalization $\simeq_{\nu}$ are equivalence relations. Thus, for instance, the only non-identity renaming from $\Sigma_{ab} = \{ a, b \}$ to itself is the function $\rho_{ab}$ with $\rho_{ab}(a) = b$ and $\rho_{ab}(b) = a$, $[\Sigma_{ab}]_{\simeq} = \{ [ab], [\overline{ab}], [\overline{ab}] \}$ are the three equivalence classes of worlds over $\Sigma_{ab}$, and we have $[(ab|ab \lor \overline{b})]_{\simeq_{\nu}} = [(ab|\overline{a} \lor \overline{b})]_{\simeq_{\nu}}$. To define a normal form with respect to renaming, we need an ordering on $(\mathcal{L}_\equiv \mathcal{L})$. Specifically, $\prec$ on $(\mathcal{L}_\equiv \mathcal{L})$ allows to uniquely order names and allows to uniquely order formulas.

**Definition 7** (admissible $\prec$ on $(\mathcal{L}_\equiv \mathcal{L})$). We call a total order $\prec$ on $(\mathcal{L}_\equiv \mathcal{L})$ admissible if it fulfills the following conditions:

1. For any two conditionals $c_1, c_2$ with $c_1 \neq c_2$ we have either $c_1 \preceq c_2$ or $c_2 \preceq c_1$ but not both.
2. For any two conditionals $c_1, c_2$ with $c_1 = c_2$ we have both $c_1 \preceq c_2$ and $c_2 \preceq c_1$.
3. For any two equivalence classes $[c_1]_{\simeq_{\nu}}, [c_2]_{\simeq_{\nu}}$ with $[c_1]_{\simeq_{\nu}} \neq [c_2]_{\simeq_{\nu}}$ and $c_1 \preceq c_2$ it holds that for every $c' \in [c_1]_{\simeq_{\nu}}, c_2 \in [c_2]_{\simeq_{\nu}}$ we have $c'_1 \preceq c_2$. 

564
For developing such an order, we will represent each formula \( A \in \mathcal{L} \) uniquely by its set \( \Omega_A \) of satisfying worlds. Furthermore, we use the following notation. For an ordering relation \( \leq \) on a set \( M \), its lexicographic extension to strings over \( M \) is denoted by \( \leq_{lex} \). For ordered sets \( S, S' \subseteq M \) with \( S = \{e_1, \ldots, e_n\} \) and \( S' = \{e'_1, \ldots, e'_n'\} \) where \( e_i \leq e_{i+1} \) and \( e'_i \leq e'_{i+1} \) its extension \( \leq_{set} \) to sets is:

\[
S \leq_{set} S' \text{ iff } n < n',
\]
or \( n = n' \) and \( e_1 \ldots e_n \leq_{lex} e'_1 \ldots e'_{n'} \). (1)

For \( \Sigma \) with ordering \( \prec \), \([\omega]_\prec \) is the usual interpretation of a world \( \omega \) as a binary number; e.g., for \( \Sigma_{ab} \) with \( a \prec b \), \([ab]_\prec = 3 \), \([a\bar{b}]_\prec = 1 \), and \([\bar{a}\bar{b}]_\prec = 0 \).

**Definition 8** (induced ordering on formulas and conditionals \( \sqsubseteq \)). Let \( \Sigma \) be a signature with linear ordering \( \prec \). The orderings induced by \( \prec \) on worlds \( \omega \), \( \omega' \) and conditionals \( (B|A), (B'|A') \) over \( \Sigma \) are given by:

\[
\omega \sqsubseteq \omega' \iff [\omega]_\prec \geq [\omega']_\prec
\]

\[
(B|A) \sqsubseteq (B'|A') \text{ iff } \Omega_A \sqsubseteq_{set} \Omega_{A'},
\]
or \( \Omega_A = \Omega_{A'} \) and \( \Omega_B \sqsubseteq_{set} \Omega_{B'} \).

In order to ease our notation, we will omit the upper symbol in \( \sqsubseteq \) and \( \leq \), and write just \( \prec \) instead, and analogously \( \leq \) for the non-strict variants. For instance, for \( \Sigma_{ab} \) with \( a \prec b \), we have \( ab \prec ab \prec \bar{a} \prec \bar{a}b \prec \bar{b} \prec \bar{a}\bar{b} \prec \bar{a}b \bar{b} \) for worlds, and \( (ab)(ab) \prec (ab)(ab) \prec (ab)(ab) \prec (ab)(ab) \prec (ab)(ab) \).

**Definition 9** (canonical ordering \( \prec_n \)). Given a signature \( \Sigma \) with linear ordering \( \prec \), let \( [(\mathcal{L}|\mathcal{L})_\Sigma]_{\simeq} = \{[r_1]_{\simeq}, \ldots, [r_m]_{\simeq} \} \) be the equivalence classes of \( (\mathcal{L}|\mathcal{L})_\Sigma \) induced by renumberings and normalization such that for each \( i \in \{1, \ldots, m\} \), the conditional \( r_i \) is a minimal element in \( [r_i]_{\simeq} \) with respect to \( \prec \), and \( r_1 \ldots r_m \). The conditionals \( \{r_1, \ldots, r_m\} \) are the canonical conditionals over \( \Sigma \). With \( M_i = [r_i]_{\simeq} \), the canonical ordering on \( (\mathcal{L}|\mathcal{L})_\Sigma \) induced by \( \preceq \), denoted by \( \prec_n \), is given by the schema

\[
[r_1]_\simeq \prec_n [r_2]_\simeq \prec_n [r_2]_\simeq \prec_n \ldots \prec_n [r_m]_\simeq \prec_n M_m
\]

where \( r \prec_n r' \) for all \( r, r' \in M_i \) with \( i \in \{1, \ldots, m\} \).

**Proposition 10** (\( \prec_n \)). The canonical ordering \( \prec_n \) defined in Definition 9 is admissible (cf. Definition 7).

**Proof.** We show that \( \prec_n \) fulfills conditions 1 and 2 for admissibility first. Consider \( c_1 = (B_1|A_1), c_2 = (B_2|A_2) \). If \( c_1 \preceq c_2 \), then \( \Omega_{A_1} \preceq \Omega_{A_2} \), and \( \Omega_{B_1} = \Omega_{B_2} \). Hence, both \( c_1 \preceq c_2 \) and \( c_2 \preceq c_1 \) hold. If \( c_1 \npreceq c_2 \), then either \( \Omega_{A_1} \neq \Omega_{A_2} \) or \( \Omega_{B_1} \neq \Omega_{B_2} \). Hence either \( c_1 \npreceq c_2 \) or \( c_2 \npreceq c_1 \) holds. Because the ordering \( \prec_n \) compared to \( \prec \) only rearranges the ordering of the equivalence classes with respect to \( \prec_n \), the conditions 1 and 2 of admissibility can be transferred from \( \prec \) to \( \prec_n \). The schema of \( \prec_n \) given in Definition 9 shows that \( \prec_n \) fulfills condition 3 of admissibility.

In the following, we will abbreviate \( R \approx_{\text{set}} R' \) simply by \( R \approx R' \) for knowledge bases \( R, R' \), and analogously for the non-strict version \( \approx_{\text{set}} \). Using these notations, we can extend the notion of renaming normal form given in (Beierle and Haldimann 2020) for normal form conditionals only to arbitrary knowledge bases containing any conditionals over \( \Sigma \).

**Definition 11** (\( \rhoNF \)). A knowledge base \( R \) over an ordered signature \( \Sigma \) is in renaming normal form (\( \rhoNF \)) if for every knowledge base \( R' \) over \( \Sigma \) with \( R \approx R' \) we have \( R \not\approx R' \).

Note that while we use \( \approx \) in Definition 11, the next proposition shows that it does not make any difference if we use \( \approx_{\text{set}} \) instead.

**Proposition 12**. A knowledge base \( R \) over an ordered signature \( \Sigma \) is in \( \rhoNF \) iff for every knowledge base \( R' \) over \( \Sigma \) with \( R \approx_{\text{set}} R' \) we have \( R \not\approx_{\text{set}} R' \).

**Proof.** We prove this proposition by showing both implications of the “iff”.

\( \Rightarrow: \) Let \( R \) be a knowledge base in \( \rhoNF \) and \( R \approx_{\text{set}} R' \). Let \( \rho \) be the renaming such that \( \rho(R) = R' \). Because \( \rho \) is in \( \rhoNF \), we have \( R \not\approx \rho(R) \not\approx R' \) (cf. condition 2 of admissibility).

\( \Leftarrow: \) Let \( R \) be a knowledge base such that \( R \not\approx \rho(R) \) for all \( R' \) such that \( R \approx_{\text{set}} R' \). This implies that \( R \not\approx R' \) for all \( R' \) such that \( R \approx R' \) and therefore that \( R \) is in \( \rhoNF \).

For every knowledge base, a corresponding knowledge base in \( \rhoNF \) exists. More precisely, we have:

**Proposition 13** (\( \rhoNF \)). For every consistent conditional knowledge base \( R \) over an ordered signature \( \Sigma \) there is a knowledge base \( R' \) in \( \rhoNF \) over \( \Sigma \) such that \( R \approx R' \).

If two knowledge bases \( R', R'' \) over \( \Sigma \) are both in \( \rhoNF \) and \( R \approx R'' \), then \( R' \approx \rhoNF \).

**Proposition 14**. Let \( R, R' \) be two consistent conditional knowledge bases over the same ordered signature. Then \( R \approx R' \) implies \( \rhoNF(R) \approx \rhoNF(R') \).

**Proof.** Assume \( \rhoNF(R) \not\approx \rhoNF(R') \). W.l.o.g. we assume \( \rhoNF(R) \not\approx \rhoNF(R') \). Let \( \rho \) be the renaming that transforms \( R \) to \( \rhoNF(R) \). We have \( \rho(R') = \rho(R) \) and therefore \( \rho(R') \approx \rhoNF(R') \). This is a contradiction.
Two knowledge bases that are equivalent under renaming share the same \( \rho \text{NF} \).

**Proposition 15.** Let \( \mathcal{R}, \mathcal{R}' \) be two consistent conditional knowledge bases over the same ordered signature. Then \( \mathcal{R} \simeq \mathcal{R}' \) implies \( \rho \text{NF}(\mathcal{R}) \simeq \rho \text{NF}(\mathcal{R}') \). Furthermore: \( \mathcal{R} \simeq \mathcal{R}' \) iff \( \rho \text{NF}(\mathcal{R}) \simeq \rho \text{NF}(\mathcal{R}') \).

**Proof.** We will show both implications of the last “iff”.

\( \Leftarrow \): \( \mathcal{R} \simeq \mathcal{R}' \) implies that there is a renaming \( \rho \) such that \( \rho(\mathcal{R}) = \mathcal{R}' \). Let \( \rho' \) be the renaming that maps \( \mathcal{R}' \) to \( \rho \text{NF}(\mathcal{R}') \). Then \( \rho \circ \rho' \) maps \( \mathcal{R} \) to \( \rho \text{NF}(\mathcal{R}') \) which is in \( \rho \text{NF} \). With Proposition 13 it follows that \( \rho \text{NF}(\mathcal{R}) = \rho \circ \rho' \circ \rho(\mathcal{R}) = \rho \text{NF}(\mathcal{R}') \). \( \square \)

Finally, we consider renamings across different signatures.

**Proposition 16.** For every consistent conditional knowledge base \( \mathcal{R} \) over \( \langle \mathcal{L} | \mathcal{C} \rangle \Sigma \) and every ordered signature \( \langle \Sigma', \prec \rangle \) such that \( |\Sigma| = |\Sigma'| \) there is a knowledge base \( \mathcal{R}' \subseteq \langle \mathcal{L} | \mathcal{C} \rangle \Sigma \) in \( \rho \text{NF} \) such that \( \mathcal{R} \simeq \mathcal{R}' \).

If two knowledge bases \( \mathcal{R}', \mathcal{R}'' \subseteq \langle \mathcal{L} | \mathcal{C} \rangle \Sigma \) are both in \( \rho \text{NF} \) and \( \mathcal{R} \simeq \mathcal{R}' \) and \( \mathcal{R} \simeq \mathcal{R}'' \) then \( \mathcal{R}' \simeq \mathcal{R}'' \).

**Proof.** As \( |\Sigma| = |\Sigma'| \) there is a renaming \( \rho : \Sigma \rightarrow \Sigma' \) that maps \( \mathcal{R} \) to a knowledge base \( \mathcal{R}'' \) over \( \Sigma' \). With Proposition 13 it follows that there exists a knowledge base \( \mathcal{R}' \) in \( \rho \text{NF} \) such that \( \mathcal{R}' \simeq \mathcal{R}'' \). Therefore \( \mathcal{R} \simeq \mathcal{R}' \simeq \mathcal{R}'' \). From \( \mathcal{R}' \simeq \mathcal{R}' \) it follows with Proposition 15 that \( \mathcal{R}' \simeq \mathcal{R}'' \). \( \square \)

Again, in general, for every knowledge base \( \mathcal{R} \) there is a, possibly non-singleton, set \( S = \{ \mathcal{R}' \in \langle \mathcal{L} | \mathcal{C} \rangle \Sigma | \mathcal{R} \simeq \mathcal{R}' \text{ and } \mathcal{R}' \text{ is in } \rho \text{NF} \} \) of renaming equivalent knowledge bases in \( \rho \text{NF} \). Proposition 16 allows us to write \( \rho \text{NF} S(\mathcal{R}) \) to denote an arbitrary element from \( S \) if it does not make a difference which element is selected (cf. \( \rho \text{NF}(\mathcal{R}) \)).

Proposition 14 can be transferred to situations with multiple signatures.

**Proposition 17.** Let \( \mathcal{R}' \in \langle \mathcal{L} | \mathcal{C} \rangle \Sigma', \mathcal{R}'' \in \langle \mathcal{L} | \mathcal{C} \rangle \Sigma'' \) be two consistent knowledge bases and \( \langle \Sigma, \prec \rangle \) be an ordered signature. Then \( \mathcal{R}' \simeq \mathcal{R}'' \) iff \( \rho \text{NF} S(\mathcal{R}') \simeq \rho \text{NF} S(\mathcal{R}'') \).

Thus, for comparing knowledge bases over different signatures with respect to equivalence under renaming and normalization, we can simply compare their renaming normal forms with respect to some ordered signature \( \Sigma \).

### 4 Converting Knowledge Bases to \( \rho \text{NF} \)

In this section, we will introduce the algorithm \( \Theta^\rho \) to transform an arbitrary conditional knowledge base over a signature with an ordering into \( \rho \text{NF} \). Algorithm \( \Theta^\rho \) can be seen as an implementation of \( \rho \text{NF}(\mathcal{R}) \). Using \( \text{Perm}(\Sigma) \) to denote the set of all renamings from \( \Sigma \) to \( \Sigma \), \( \Theta^\rho \) is shown in Algorithm 1. The algorithm exploits two main observations:

**(O1)** For each equivalence class \( C_i \), every conditional \( c \in C_i \) is mapped to \( C_i \) by all renamings over \( \Sigma \).

**(O2)** For any two knowledge bases \( \mathcal{R}, \mathcal{R}' \) over \( \Sigma \) and \( 1 \leq i \leq n \) the following implication holds:

\[ \mathcal{R} \cap (C_1 \cup \cdots \cup C_i) \lessdot \mathcal{R}' \cap (C_1 \cup \cdots \cup C_i) \Rightarrow \mathcal{R} \lessdot \mathcal{R}' \]

The first observation follows from the definition of the equivalence classes \( C_1, \ldots, C_n \). The second observation follows from condition 3 of the admissibility of \( \lessdot \) (Definition 7). In combination these observations imply:

**(O3)** For any knowledge base \( \mathcal{R} \) over \( \Sigma \) and \( 1 \leq i \leq n \) the following implication holds:

If a renaming \( \rho \) maps \( \mathcal{R} \) to a knowledge base in \( \rho \text{NF} \), then it also maps \( \mathcal{R} \cap (C_1 \cup \cdots \cup C_i) \) to a knowledge base in \( \rho \text{NF} \).

The algorithm \( \Theta^\rho \) looks for a renaming that transforms a knowledge base \( \mathcal{R} \) to \( \rho \text{NF} \) in Lines 2 to 18. To do so it starts with the set of all renamings (cf. Line 2) and filters them in each iteration of the for loop starting in Line 3. Using observation (O3), in the \( i \)-th iteration we keep only the renamings that map \( \mathcal{R} \cap (C_1 \cup \cdots \cup C_i) \) to its \( \rho \text{NF} \). After the loop, only renamings that map \( \mathcal{R} \) to its \( \rho \text{NF} \) remain in \( P_n \). The algorithm chooses one of these renamings (cf. Line 18) and applies it to \( \mathcal{R} \) (cf. Line 19).

Inside the outer loop in Lines 3 to 17, we employ another optimization in Lines 5 to 14. We know, that at the beginning of the \( i \)-th iteration, all \( \rho \in P \) map \( \mathcal{R} \cap (C_1 \cup \cdots \cup C_{i-1}) \) to the same knowledge base (which is in \( \rho \text{NF} \)). Together with the second observation (O2) it follows that:

\[ \rho_1(\mathcal{R} \cap (C_1 \cup \cdots \cup C_i)) \lessdot \rho_2(\mathcal{R} \cap (C_1 \cup \cdots \cup C_i)) \leftarrow \rho_1(\mathcal{R} \cap C_i) \lessdot \rho_2(\mathcal{R} \cap C_i) \]

Therefore, it suffices to select those renamings, that minimize \( \mathcal{R} \cap C_i \). This is what happens in Lines 5 to 14. The

**Algorithm 1 \( \Theta^\rho \): Transform cond. knowledge base into \( \rho \text{NF} \)**

**Input:** conditional knowledge base \( \mathcal{R} \) over signature \( \Sigma \) with linear ordering \( \prec \)

**Output:** conditional knowledge base \( \mathcal{R}_{\rho \text{NF}} \) in \( \rho \text{NF} \) such that \( \mathcal{R} \simeq \mathcal{R}_{\rho \text{NF}} \)

1: \( \{C_1, \ldots, C_n\} \leftarrow \langle \mathcal{L} | \mathcal{C} \rangle \langle \Sigma, \prec \rangle \) with \( C_1 \prec \cdots \prec C_n \)
2: \( P_0 \leftarrow \text{Perm}(\Sigma) \)
3: for \( i = 1, \ldots, n \) do
4: \( R_i \leftarrow \mathcal{R} \cap C_i \)
5: \( P_{\text{tmp}} \leftarrow \emptyset \)
6: \( R_{\text{tmp}} \leftarrow \emptyset \)
7: for \( \rho \in P_{i-1} \) do
8: if \( P_{\text{tmp}} = \emptyset \) or \( \rho(R_i) \lessdot R_{\text{tmp}} \) then
9: \( P_{\text{tmp}} \leftarrow \{\rho\} \)
10: \( R_{\text{tmp}} \leftarrow \rho(R_i) \)
11: else
12: if \( \rho(R_i) \lessdot R_{\text{tmp}} \) then
13: \( P_{\text{tmp}} \leftarrow P_{\text{tmp}} \cup \{\rho\} \)
14: \( P_i \leftarrow P_{\text{tmp}} \)
15: if \( |P_i| = 1 \) then
16: \( P_n \leftarrow P_i \)
17: break
18: \( \rho \leftarrow \text{chooseOneOf}(P_n) \)
19: return \( \mathcal{R}_{\rho \text{NF}} \leftarrow \rho(\mathcal{R}) \)
check in Lines 15 to 17 avoids unnecessary iterations as there are no non-minimal elements in a singleton.

Formalizing these observations about $\Theta^\rho$ yields:

**Proposition 18 ($\Theta^\rho$).** Let $R$ be a knowledge base.

1. (termination) $\Theta^\rho$ terminates on input $R$.
2. ($\succeq$) If $\rho_1, \rho_2$ are different choices in Line 18 of executing $\Theta^\rho(R)$, then $\rho_1(R) = _\nu \rho_2(R)$.
3. ($\rho$NF) $\Theta^\rho(R)$ is in $\rho$NF.
4. ($\simeq$) $R \simeq \Theta^\rho(R)$.

In the remainder of this chapter, we will illustrate two approaches how the algorithm $\Theta^\rho$ can be used to check whether two knowledge bases are renaming equivalent. The first approach transforms two knowledge bases into $\rho$NF over their corresponding signatures. Afterwards the knowledge bases are compared. This approach makes use of the following proposition.

**Proposition 19.** Let $\Sigma_1 = \{a_1, \ldots, a_k\}$, $\Sigma_2 = \{b_1, \ldots, b_k\}$ be two ordered signatures with the same size with $a_1 \prec \cdots \prec a_k$ and $b_1 \prec \cdots \prec b_k$. Define $t : \Sigma_1 \rightarrow \Sigma_2, a_i \mapsto b_i$ for $i = 1, \ldots, k$. Let $R_1$ be a knowledge base over $\Sigma_1$ and $R_2$ be a knowledge base over $\Sigma_2$. Then $R_1 \simeq_{\nu} R_2$ iff $t(\Theta^\rho(R_1)) = _\nu \Theta^\rho(R_2)$.

**Proof.** We will prove both implications of the “iff”:

$\Rightarrow$: Because $t$ is a signature renaming we have $R_1 \simeq_{\nu} \Theta^\rho(R_1) \simeq_{\nu} \Theta^\rho(R_2)$.

$\Leftarrow$: Assume that $R_1 \simeq_{\nu} R_2$ but $t(\Theta^\rho(R_1)) \neq _\nu \Theta^\rho(R_2)$. Therefore $t(\Theta^\rho(R_1)) \prec \Theta^\rho(R_2)$ or $\Theta^\rho(R_2) \prec t(\Theta^\rho(R_1))$ (cf. admissibility of $\prec$, condition 3 in Definition 7.). The first case is impossible because $t(\Theta^\rho(R_2))$ is in $\rho$NF. The ordering of the elements in $\Sigma_1$ and $\Sigma_2$ together with the definition of $t$ implies that $R \prec R' \Leftrightarrow t(R) \prec t(R')$ for all knowledge bases $R, R'$. The second case therefore implies $t^{-1}(\Theta^\rho(R_2)) \prec \Theta^\rho(R_1)$. This is a contradiction, as $\Theta^\rho(R_1)$ is in $\rho$NF.

**Example 20.** We want to check whether the knowledge bases $R_1 = \{(d | f | e) \in R \mid e \in f\}$ over $\Sigma_1 = \{d, e, f\}$ and $R_2 = \{(b | a) \in R \mid a \in b\}$ over $\Sigma_2 = \{b, c\}$ are equivalent under signature renaming and normalization.

Defining $d < e < f$ on $\Sigma_1$ and $a < b < c$ on $\Sigma_2$ the results of applying $\Theta^\rho$ to both $R_1$ and $R_2$ yields $\Theta^\rho(R_1) = \{(e | d) \in R \mid (e \in d \cap (e | f \cap (f | T) \cap (T | T))\}$ and $\Theta^\rho(R_2) = \{(b | a) \in R \mid (a \in b \cap (a | c) \cap (a | T))\}$. Now we apply $t = \{d \mapsto a, e \mapsto b, f \mapsto c\}$ to $R_1$ and get $t(\Theta^\rho(R_1)) = \{(b | a) \in R \mid (a \in b \cap (a | c) \cap (a | T))\}$. This is equal under normalization to $\Theta^\rho(R_2)$. Therefore, $R_1$ and $R_2$ are equivalent under signature renaming and normalization.

The second approach renames both knowledge bases to the same signature and then transforms them into $\rho$NF.

**Proposition 21.** Let $\Sigma_1, \Sigma_2$ be two signatures and $(\Sigma, <)$ be an ordered signature all with the same size. Let $\rho_1 : \Sigma_1 \rightarrow \Sigma$ and $\rho_2 : \Sigma_2 \rightarrow \Sigma$ be arbitrary renamings. Let $R_1$ be a knowledge base over $\Sigma_1$ and $R_2$ be a knowledge base over $\Sigma_2$. Then $R_1 \simeq_{\nu} R_2$ iff $\Theta^\rho(\rho_1(R_1)) = _\nu \Theta^\rho(\rho_2(R_2))$.

**Proof.** This is a consequence of Propositions 17 and 18.

The following example illustrates the second approach.

**Example 22** (Example 20 continued). Based on Proposition 21, we want to check whether the knowledge bases $R_1, R_2$ given in Example 20 are equivalent under signature renaming and normalization. First, we transfer both knowledge bases to the ordered signature $(\Sigma, <)$ with $\Sigma = \{a_1, a_2, a_3\}$ and $a_1 \prec a_2 \prec a_3$:

- $R_1 \simeq R_1' = \{(a_1 a_3 | a_2), (a_2 a_3 | a_2), (a_1 a_2 | a_2), (a_1 a_3 | a_1), (a_2 a_3 | a_1), (a_2 | T)\}$

- $R_2 \simeq R_2' = \{(a_2 a_1 | a_1), (a_2 a_3 | a_1), (a_1 a_3 | a_1), (a_1 | T)\}$

Applying $\Theta^\rho$ yields the two knowledge bases

- $\Theta^\rho(R_1') = \{(a_2 a_3 | a_1), (a_1 a_3 | a_1), (a_2 a_1 | a_1), (a_1 a_3 | a_1), (a_2 a_3 | a_1), (a_1 | T)\}$

- $\Theta^\rho(R_2') = \{(a_2 a_3 | a_1), (a_2 a_3 | a_1), (a_1 a_3 | a_1), (a_1 | T)\}$

which are equivalent under normalization, showing that $R_1$ and $R_2$ are equivalent under signature renaming and normalization.

5 Renaming Antecedent Normal Form

In the previous sections, we deliberately developed and investigated the $\rho$NF independent of other normal forms that have been proposed for conditional knowledge bases. However, the benefits of $\rho$NF can be even better exploited in combination with other normal forms. For instance, let us call a conditional knowledge base $R$ to be in **propositional normal form (PNF)** if $\rho = \nu(R)$. If we restrict our attention to knowledge bases in PNF, then each consistent $R$ has a unique $\rho$NF because $\nu$ eliminates syntactic variants of propositional formulas; this also leads to slightly stricter versions of e.g. Propositions 13–17. The antecedent normal form (ANF) of $R$ goes considerably further than PNF.

**Definition 23** (ANF (Beierle and Kutsch 2019)). Let $R$ be a knowledge base.

- $\text{Ant}(R) = \{A \mid (B | A) \in R\}$ are the antecedents of $R$.
- For $A \in \text{Ant}(R)$, the set $R_A = \{(B' | A) \mid (B' | A') \in R \land A = A'\}$ is the set of $A$-conditionals in $R$.
- $R$ is in antecedent normal form (ANF) if either $R$ is inconsistent and $\rho = \infty$, or $R$ is consistent, does not contain any self-fulfilling conditional, contains only conditionals of the form $(AB | A)$, and $|\text{Ant}(R)| = 1$ for all $A \in \text{Ant}(R)$.

Using propositional normalization $\nu$, the transformation system $\Theta^\rho$ (Figure 1) maps every $R$ into a unique, model-equivalent $R'$ that is in ANF. (Beierle and Kutsch 2019, Prop. 2). Combining ANF and $\rho$NF, we get:

**Definition 24** ($\rho$ANF, $\Theta^{\rho\text{ANF}}$). A knowledge base $R$ is in renaming antecedent normal form ($\rho$ANF) if it is in ANF and in $\rho$NF. With $\Theta^{\rho\text{ANF}}(R)$, we denote $\Theta^\rho(\Theta^{\rho\text{ANF}}(R))$. $R_1, R_2$ may have different $\rho$NFs, but the same $\rho$ANF. The latter does not guarantee any more that $R_1$ and $R_2$ are renaming equivalent, but it still ensures that they are model equivalent up to renamings:

**Proposition 25** ($\rho$ANF, $\Theta^{\rho\text{ANF}}$). Let $R$ be a knowledge base.
Figure 1: Rules $\Theta^\rho$ (Beierle and Kutsch 2019) mapping every knowledge base to its ANF; $\Pi$ is a consistency test, e.g. based on the tolerance criterion (Goldszmidt and Pearl 1996), and $\circ$ represents an inconsistent knowledge base.

1. (completeness) There is $\mathcal{R}'$ in $\rho$ANF with $\mathcal{R} \simeq_{\text{mod}} \mathcal{R}'$.

2. (uniqueness) $\Theta^\rho(\mathcal{R})$ is uniquely determined.

3. (commutative) $\Theta^\rho(\mathcal{R} \cup \mathcal{R}) = \Theta^\rho(\Theta^\rho(\mathcal{R}))$.

4. ($\rho$ANF) $\Theta^\rho(\mathcal{R})$ is in $\rho$ANF.

5. ($\simeq_{\text{mod}}$) $\mathcal{R} \simeq_{\text{mod}} \Theta^\rho(\mathcal{R})$.

For model equivalence up to renaming of symbols we get:

**Proposition 26 ($\Theta^\rho$).** For $i = 1, 2$, let $\mathcal{R}_i$ be a knowledge base over $\Sigma_i$, and $\rho_i : \Sigma_i \mapsto \Sigma$ a signature. Then $\Theta^\rho(\rho_1(\mathcal{R}_1)) = \Theta^\rho(\rho_2(\mathcal{R}_2))$ implies $\mathcal{R}_1 \simeq_{\text{mod}} \mathcal{R}_2$.

Thus, two knowledge bases over different signatures are model equivalent up to renamings if they have the same $\rho$ANF for some (arbitrary) embeddings into some common signature. Note that Proposition 26 also covers the special case where $\Sigma_1 = \Sigma_2 = \Sigma$ and where $\rho_1$ is the identity.

**Example 27.** Let $\Sigma_{bcf} = \{b, c, f\}$ and $\mathcal{R}_{bcf} = \{(b \vee \neg b), (\neg c \vee c), (\neg f \vee f, (\neg f \vee \neg c \vee f)\}$. When comparing $\mathcal{R}_{bcf}$ to the knowledge base $\mathcal{R}_{car}$ from Example 1 we observe that their $\rho$ANFs are different and that they are not comparing renaming equivalent. For comparing their $\rho$ANFs, let us first apply $\Theta^\rho$ to both, yielding:

$\Theta^\rho(\mathcal{R}_{bcf}) = \{(\nu(c(e))\nu(f)), (\nu(c(e))\nu(f)), (\nu(\neg c)\nu(\neg f))\}$

Using an ordered standard signature like $\Sigma = \{a, b, c\}$ with $a \subset b \subset c$ and applying $\Theta^\rho$, we obtain $\rho$ANF$_\Sigma(\mathcal{R}_{bcf}) = \rho$ANF$_\Sigma(\mathcal{R}_{car}) = \{(\nu(\neg c)a)\nu(a)), (\nu(bc)\nu(b)), (\nu(\pi)\nu(\pi))\}$, implying $\mathcal{R}_{bcf} \simeq_{\text{mod}} \mathcal{R}_{car}$. Specifically, for $\rho(c) = b, \rho(e) = c, \rho(f) = c$, we have $\mathcal{R}_{car} \equiv_{\text{mod}} \rho(\mathcal{R}_{bcf})$. Due to Proposition 3, we thus have $A \vdash_{\mathcal{R}_{bcf}} B$ iff $\rho^{-1}(A) \sim_{\mathcal{R}_{bcf}} (B) \rho(C) \sim_{\mathcal{R}_{bcf}} (D) \equiv_{\text{mod}} \rho(\mathcal{R}_{bcf})$ for all formulas $A, B, C, D$ over the signatures of $\mathcal{R}_{bcf}$ and $\mathcal{R}_{car}$, respectively.

### 6 Conclusions and Further Work

We addressed the comparison of conditional knowledge bases when taking renamings of the underlying signature into account. We extended the notion of renaming normal form ($\rho$NF), capturing equivalences of knowledge bases under renamings, previously proposed only for normal form conditionals, to arbitrary knowledge bases. We introduced the new renaming antecedent normal form ($\rho$ANF), ensuring model equivalence up to renamings. Both normal forms are applicable also across different signatures. We presented procedures transforming any $\mathcal{R}$ into its unique $\rho$NF and $\rho$ANF, respectively, and studied their main properties. In future work, we will empirically evaluate the properties and benefits of knowledge bases in renaming normal form, and investigate the complexity of the normal form algorithms.

### References


