

Belief Revision Theory

Valentin Fouillard Safouan TAHA Frederic Boulanger
Nicolas Sabouret

September 13, 2023

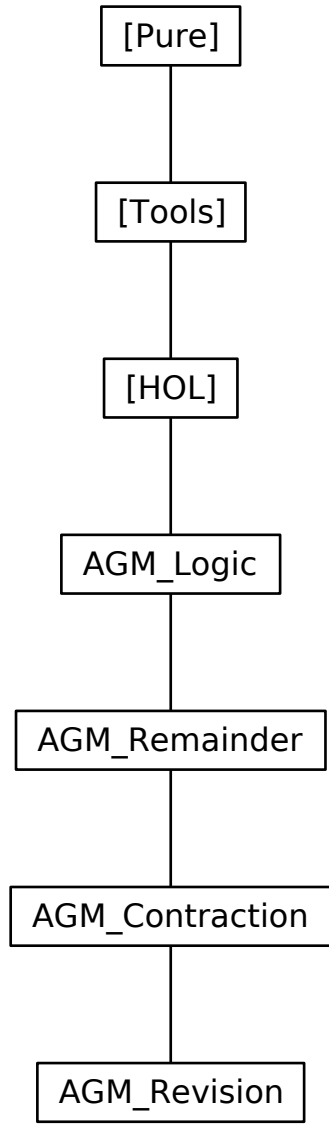
Abstract

The 1985 paper by Carlos Alchourrón, Peter Gärdenfors, and David Makinson (AGM), “On the Logic of Theory Change: Partial Meet Contraction and Revision Functions” launches a large and rapidly growing literature that employs formal models and logics to handle changing beliefs of a rational agent and to take into account new piece of information observed by this agent. In 2011, a review book titled “AGM 25 Years: Twenty-Five Years of Research in Belief Change” was edited to summarize the first twenty five years of works based on AGM.

This HOL-based AFP entry is a faithful formalization of the AGM operators (e.g. contraction, revision, remainder ...) axiomatized in the original paper. It also contains the proofs of all the theorems stated in the paper that show how these operators combine. Both proofs of Harper and Levi identities are established.

Contents

1	Introduction	5
2	Logics	6
2.1	Tarskian Logic	6
2.2	Supraclassical Logic	8
2.3	Compact Logic	14
3	Remainders	15
3.1	Remainders in a Tarskian logic	15
3.2	Remainders in a supraclassical logic	15
3.3	Remainders in a compact logic	17
4	Contractions	18
4.1	AGM contraction postulates	20
4.2	Partial meet contraction definition	22
4.3	Equivalence of partial meet contraction and AGM contraction	25
5	Revisions	35
5.1	AGM revision postulates	36
5.2	Relation of AGM revision and AGM contraction	37



1 Introduction

The 1985 paper by Carlos Alchourrón, Peter Gärdenfors, and David Makinson (AGM), “On the Logic of Theory Change: Partial Meet Contraction and Revision Functions” [1] launches a large and rapidly growing literature that employs formal models and logics to handle changing beliefs of a rational agent and to take into account new piece of information observed by this agent. In 2011, a review book titled ”AGM 25 Years: Twenty-Five Years of Research in Belief Change” was edited to summarize the first twenty five years of works based on AGM [2].

According to Google Scholar, the original AGM paper was cited 4000 times! This AFP entry is HOL-based and it is a faithful formalization of the logic operators (e.g. contraction, revision, remainder ...) axiomatized in the AGM paper. It also contains the proofs of all the theorems stated in the paper that show how these operators combine. Both proofs of Harper and Levi identities are established.

A belief state can be considered as a consistent set of beliefs (logical propositions) closed under logical reasoning. Belief changes represent the operations that apply on a belief state to remove some of it and/or to add new beliefs (propositions). In the latter case, it is possible that other beliefs are affected by these changes (to preserve consistency for example). AGM define several postulates to guarantee that such operations preserve consistency meaning that the agent keeps rational. Three kinds of operators are defined :

- The contraction \div : where a proposition is removed from a belief set
- The expansion \oplus : where a proposition is added to a belief set
- The revision $*$: where a proposition is added to a belief set such that the belief set remains consistent

In this AFP entry, there are three theory files:

1. The AGM Logic file contains a classification of logics used in the AGM framework.
2. The AGM Remainder defines a important operator used in the AGM framework.
3. The AGM Contraction file contains the postulates of the AGM contraction and its relation with the meet contraction.
4. The AGM Revision file contains the postulates of the AGM revision and its relation with the meet revision.

2 Logics

The AGM framework depends on the underlying logic used to express beliefs. AGM requires at least a Tarskian propositional calculus. If this logic is also supra-classical and/or compact, new properties are established and the main theorems of AGM are strengthened. To model AGM it is therefore important to start by formalizing this underlying logic and its various extensions. We opted for a deep embedding in HOL which required the redefinition of all the logical operators and an axiomatization of their rules. This is certainly not efficient in terms of proof but it gives a total control of our formalization and an assurance that the logic used has no hidden properties depending on the Isabelle/HOL implementation. We use the Isabelle *locales* feature and we take advantage of the inheritance/extension mechanisms between locales.

2.1 Tarskian Logic

The first locale formalizes a Tarskian logic based on the famous Tarski's consequence operator: $Cn A$ which gives the set of all propositions (**closure**) that can be inferred from the set of propositions A , Exactly as it is classically axiomatized in the literature, three assumptions of the locale define the consequence operator.

```

locale Tarskian-logic =
fixes Cn::'a set  $\Rightarrow$  'a set
assumes monotonicity-L:  $\langle A \subseteq B \implies Cn(A) \subseteq Cn(B) \rangle$ 
and inclusion-L:  $\langle A \subseteq Cn(A) \rangle$ 
and transitivity-L:  $\langle Cn(Cn(A)) \subseteq Cn(A) \rangle$ 

```

— Short notation for “ φ can be inferred from the propositions in A ”.

```

fixes infer::'a set  $\Rightarrow$  'a  $\Rightarrow$  bool (infix  $\langle \vdash \rangle$  50)
defines  $\langle A \vdash \varphi \equiv \varphi \in Cn(A) \rangle$ 

```

— φ is valid (a tautology) if it can be inferred from nothing.

```

fixes valid::'a  $\Rightarrow$  bool (infix  $\langle \Vdash \rangle$ )
defines  $\langle \Vdash \varphi \equiv \{\} \vdash \varphi \rangle$ 

```

— $A \oplus \varphi$ is all that can be inferred from A and φ .

```

fixes expansion::'a set  $\Rightarrow$  'a  $\Rightarrow$  'a set (infix  $\langle \oplus \rangle$  57)
defines  $\langle A \oplus \varphi \equiv Cn(A \cup \{\varphi\}) \rangle$ 

```

begin

```

lemma idempotency-L:  $\langle Cn(Cn(A)) = Cn(A) \rangle$ 
by (simp add: inclusion-L transitivity-L subset-antisym)

```

```

lemma assumption-L:  $\langle \varphi \in A \implies A \vdash \varphi \rangle$ 
using inclusion-L infer-def by blast

```

lemma *validD-L*: $\langle \Vdash \varphi \implies \varphi \in Cn(A) \rangle$
using *monotonicity-L valid-def infer-def* **by** *fastforce*

lemma *valid-expansion*: $\langle K = Cn(A) \implies \Vdash \varphi \implies K \oplus \varphi = K \rangle$
by (*simp add: idempotency-L insert-absorb validD-L valid-def expansion-def*)

lemma *transitivity2-L*:
assumes $\langle \forall \varphi \in B. A \vdash \varphi \rangle$
and $\langle B \vdash \psi \rangle$
shows $\langle A \vdash \psi \rangle$
proof –
from *assms(1)* **have** $\langle B \subseteq Cn(A) \rangle$ **by** (*simp add: infer-def subsetI*)
hence $\langle Cn(B) \subseteq Cn(A) \rangle$ **using** *idempotency-L monotonicity-L* **by** *blast*
moreover from *assms(2)* **have** $\langle \psi \in Cn(B) \rangle$ **by** (*simp add: infer-def*)
ultimately show *?thesis* **using** *infer-def* **by** *blast*
qed

lemma *Cn-same*: $\langle (Cn(A) = Cn(B)) \longleftrightarrow (\forall C. A \subseteq Cn(C) \longleftrightarrow B \subseteq Cn(C)) \rangle$
proof
{ **assume** $h: \langle Cn(A) = Cn(B) \rangle$
from h **have** $\langle \forall \varphi \in B. A \vdash \varphi \rangle$
by (*simp add: Tarskian-logic.assumption-L Tarskian-logic-axioms infer-def*)
moreover from h **have** $\langle \forall \varphi \in A. B \vdash \varphi \rangle$
by (*simp add: Tarskian-logic.assumption-L Tarskian-logic-axioms infer-def*)
ultimately have $\langle \forall C. A \subseteq Cn(C) \longleftrightarrow B \subseteq Cn(C) \rangle$
using h *idempotency-L inclusion-L monotonicity-L* **by** *blast*
} **thus** $\langle Cn(A) = Cn(B) \implies \forall C. (A \subseteq Cn(C)) = (B \subseteq Cn(C)) \rangle$.
next
{ **assume** $h: \langle \forall C. (A \subseteq Cn(C)) = (B \subseteq Cn(C)) \rangle$
from h **have** $\langle (A \subseteq Cn(A)) = (B \subseteq Cn(A)) \rangle$ **and** $\langle (A \subseteq Cn(B)) = (B \subseteq Cn(B)) \rangle$ **by** *simp+*
hence $\langle B \subseteq Cn(A) \rangle$ **and** $\langle A \subseteq Cn(B) \rangle$ **by** (*simp add: inclusion-L*)
hence $\langle Cn(A) = Cn(B) \rangle$
using *idempotency-L monotonicity-L* **by** *blast*
} **thus** $\langle (\forall C. (A \subseteq Cn(C)) = (B \subseteq Cn(C))) \implies Cn(A) = Cn(B) \rangle$.
qed

— The closure of the union of two consequence closures.
lemma *Cn-union*: $\langle Cn(Cn(A) \cup Cn(B)) = Cn(A \cup B) \rangle$
proof
have $\langle Cn(Cn(A) \cup Cn(B)) \subseteq Cn(Cn(A \cup B)) \rangle$ **by** (*simp add: monotonicity-L*)
thus $\langle Cn(Cn(A) \cup Cn(B)) \subseteq Cn(A \cup B) \rangle$ **by** (*simp add: idempotency-L*)
next
have $\langle (A \cup B) \subseteq (Cn(A) \cup Cn(B)) \rangle$ **using** *inclusion-L* **by** *blast*
thus $\langle Cn(A \cup B) \subseteq Cn(Cn(A) \cup Cn(B)) \rangle$ **by** (*simp add: monotonicity-L*)
qed

— The closure of an infinite union of consequence closures.

```

lemma Cn-Union:  $\langle Cn(\bigcup\{Cn(B)|B. P B\}) = Cn(\bigcup\{B. P B\}) \rangle$  (is  $\langle ?A = ?B \rangle$ )
proof
  have  $\langle ?A \subseteq Cn ?B \rangle$ 
    apply(rule monotonicity-L, rule Union-least, auto)
    by (metis Sup-upper in-mono mem-Collect-eq monotonicity-L)
  then show  $\langle ?A \subseteq ?B \rangle$ 
    by (simp add: idempotency-L)
next
  show  $\langle ?B \subseteq ?A \rangle$ 
    by (metis (mono-tags, lifting) Union-subsetI inclusion-L mem-Collect-eq mono-
tonicity-L)
qed

```

— The intersection of two closures is closed.

```

lemma Cn-inter:  $\langle K = Cn(A) \cap Cn(B) \implies K = Cn(K) \rangle$ 
proof –
  { fix K assume  $h:\langle K = Cn(A) \cap Cn(B) \rangle$ 
    from h have  $\langle K \subseteq Cn(A) \rangle$  and  $\langle K \subseteq Cn(B) \rangle$  by simp+
    hence  $\langle Cn(K) \subseteq Cn(A) \rangle$  and  $\langle Cn(K) \subseteq Cn(B) \rangle$  using idempotency-L mono-
tonicity-L by blast+
    hence  $\langle Cn(K) \subseteq Cn(A) \cap Cn(B) \rangle$  by simp
    with h have  $\langle K = Cn(K) \rangle$  by (simp add: inclusion-L subset-antisym)
  } thus  $\langle K = Cn(A) \cap Cn(B) \implies K = Cn(K) \rangle$  .
qed

```

— An infinite intersection of closures is closed.

```

lemma Cn-Inter:  $\langle K = \bigcap\{Cn(B)|B. P B\} \implies K = Cn(K) \rangle$ 
proof –
  { fix K assume  $h:\langle K = \bigcap\{Cn(B)|B. P B\} \rangle$ 
    from h have  $\langle \forall B. P B \implies K \subseteq Cn(B) \rangle$  by blast
    hence  $\langle \forall B. P B \implies Cn(K) \subseteq Cn(B) \rangle$  using idempotency-L monotonicity-L
by blast
    hence  $\langle Cn(K) \subseteq \bigcap\{Cn(B)|B. P B\} \rangle$  by blast
    with h have  $\langle K = Cn(K) \rangle$  by (simp add: inclusion-L subset-antisym)
  } thus  $\langle K = \bigcap\{Cn(B)|B. P B\} \implies K = Cn(K) \rangle$  .
qed

```

end

2.2 Supraclassical Logic

A Tarskian logic has only one abstract operator catching the notion of consequence. A basic case of such a logic is a **Supraclassical** logic that is a logic with all classical propositional operators (e.g. conjunction (\wedge), implication (\longrightarrow), negation (\neg) ...) together with their classical semantics.

We define a new locale. In order to distinguish the propositional operators of our supraclassical logic from those of Isabelle/HOL, we use dots (e.g. $\dot{\wedge}$ stands for conjunction). We axiomatize the introduction and elimination

rules of each operator as it is commonly established in the classical literature. As explained before, we give priority to a complete control of our logic instead of an efficient shallow embedding in Isabelle/HOL.

locale *Supraclassical-logic* = *Tarskian-logic* +

```

fixes true-PL:: ‹'a›           (‹ $\top$ ›)
and false-PL:: ‹'a›           (‹ $\perp$ ›)
and imp-PL:: ‹'a  $\Rightarrow$  'a  $\Rightarrow$  'a› (infix ‹ $\longrightarrow$ › 55)
and not-PL:: ‹'a  $\Rightarrow$  'a›       (‹ $\neg$ ›)
and conj-PL:: ‹'a  $\Rightarrow$  'a  $\Rightarrow$  'a› (infix ‹ $\wedge$ › 55)
and disj-PL:: ‹'a  $\Rightarrow$  'a  $\Rightarrow$  'a› (infix ‹ $\vee$ › 55)
and equiv-PL:: ‹'a  $\Rightarrow$  'a  $\Rightarrow$  'a› (infix ‹ $\longleftrightarrow$ › 55)

assumes true-PL: ‹ $A \vdash \top$ ›

and false-PL: ‹ $\{\perp\} \vdash p$ ›

and impI-PL: ‹ $A \cup \{p\} \vdash q \Longrightarrow A \vdash (p \longrightarrow q)$ ›
and mp-PL: ‹ $A \vdash p \longrightarrow q \Longrightarrow A \vdash p \Longrightarrow A \vdash q$ ›

and notI-PL: ‹ $A \vdash p \longrightarrow \perp \Longrightarrow A \vdash \neg p$ ›
and notE-PL: ‹ $A \vdash \neg p \Longrightarrow A \vdash (p \longrightarrow \perp)$ ›

and conjI-PL: ‹ $A \vdash p \Longrightarrow A \vdash q \Longrightarrow A \vdash (p \wedge q)$ ›
and conjE1-PL: ‹ $A \vdash p \wedge q \Longrightarrow A \vdash p$ ›
and conjE2-PL: ‹ $A \vdash p \wedge q \Longrightarrow A \vdash q$ ›

and disjI1-PL: ‹ $A \vdash p \Longrightarrow A \vdash (p \vee q)$ ›
and disjI2-PL: ‹ $A \vdash q \Longrightarrow A \vdash (p \vee q)$ ›
and disjE-PL: ‹ $A \vdash p \vee q \Longrightarrow A \vdash p \longrightarrow r \Longrightarrow A \vdash q \longrightarrow r \Longrightarrow A \vdash r$ ›

and equivI-PL: ‹ $A \vdash p \longrightarrow q \Longrightarrow A \vdash q \longrightarrow p \Longrightarrow A \vdash (p \longleftrightarrow q)$ ›
and equivE1-PL: ‹ $A \vdash p \longleftrightarrow q \Longrightarrow A \vdash p \longrightarrow q$ ›
and equivE2-PL: ‹ $A \vdash p \longleftrightarrow q \Longrightarrow A \vdash q \longrightarrow p$ ›

— non intuitionistic rules
and absurd-PL: ‹ $A \vdash \neg (\neg p) \Longrightarrow A \vdash p$ ›
and ex-mid-PL: ‹ $A \vdash p \vee (\neg p)$ ›

```

begin

In the following, we will first retrieve the classical logic operators semantics coming from previous introduction and elimination rules

lemma *non-consistency*:

```

assumes ‹ $A \vdash \neg p$ ›
and ‹ $A \vdash p$ ›
shows ‹ $A \vdash q$ ›
by (metis assms(1) assms(2) false-PL mp-PL notE-PL singleton-iff transitivity2-L)

```

— this direct result brings directly many remarkable properties of implication (i.e. transitivity)

lemma *imp-PL*: $\langle A \vdash p \longrightarrow. q \longleftrightarrow A \cup \{p\} \vdash q \rangle$
apply (*intro iffI impI-PL*)
apply(*rule mp-PL[where p=p], meson UnI1 assumption-L transitivity2-L*)
using *assumption-L by auto*

lemma *not-PL*: $\langle A \vdash \neg p \longleftrightarrow A \cup \{p\} \vdash \perp \rangle$
using *notE-PL notI-PL imp-PL by blast*

— Classical logic result

lemma *notnot-PL*: $\langle A \vdash \neg (\neg p) \longleftrightarrow A \vdash p \rangle$
apply(*rule iffI, simp add:absurd-PL*)
by (*meson mp-PL notE-PL Un-upper1 Un-upper2 assumption-L infer-def monotonicity-L not-PL singletonI subsetD*)

lemma *conj-PL*: $\langle A \vdash p \wedge. q \longleftrightarrow (A \vdash p \wedge A \vdash q) \rangle$
using *conjE1-PL conjE2-PL conjI-PL by blast*

lemma *disj-PL*: $\langle A \vdash p \vee. q \longleftrightarrow A \cup \{\neg p\} \vdash q \rangle$
proof

assume *a*: $\langle A \vdash p \vee. q \rangle$
have *b*: $\langle A \vdash p \longrightarrow. (\neg p \longrightarrow. q) \rangle$
by (*intro impI-PL*) (*meson Un-iff assumption-L insertI1 non-consistency*)
have *c*: $\langle A \vdash q \longrightarrow. (\neg p \longrightarrow. q) \rangle$
by (*simp add: assumption-L impI-PL*)
from *a b c* **have** $\langle A \vdash \neg p \longrightarrow. q \rangle$
by (*erule-tac disjE-PL*) *simp-all*
then show $\langle A \cup \{\neg p\} \vdash q \rangle$
using *imp-PL by blast*

next

assume *a*: $\langle A \cup \{\neg p\} \vdash q \rangle$
hence *b*: $\langle A \vdash \neg p \longrightarrow. q \rangle$ **by** (*simp add: impI-PL*)
then show $\langle A \vdash p \vee. q \rangle$
apply(*rule-tac disjE-PL[OF ex-mid-PL, of A p <p ∨. q>]*)
by (*auto simp add: assumption-L disjI2-PL disjI1-PL impI-PL imp-PL*)

qed

lemma *equiv-PL*: $\langle A \vdash p \longleftrightarrow. q \longleftrightarrow (A \cup \{p\} \vdash q \wedge A \cup \{q\} \vdash p) \rangle$
using *imp-PL equivE1-PL equivE2-PL equivI-PL by blast*

corollary *valid-imp-PL*: $\langle \vDash (p \longrightarrow. q) = (\{p\} \vdash q) \rangle$

and *valid-not-PL*: $\langle \vDash (\neg p) = (\{p\} \vdash \perp) \rangle$

and *valid-conj-PL*: $\langle \vDash (p \wedge. q) = (\vDash p \wedge \vDash q) \rangle$

and *valid-disj-PL*: $\langle \vDash (p \vee. q) = (\{\neg p\} \vdash q) \rangle$

and *valid-equiv-PL*: $\langle \vDash (p \longleftrightarrow. q) = (\{p\} \vdash q \wedge \{q\} \vdash p) \rangle$

using *imp-PL not-PL conj-PL disj-PL equiv-PL valid-def by auto*

Second, we will combine each logical operator with the consequence operator

Cn: it is a trick to profit from set theory to get many essential lemmas without complex inferences

declare *infer-def*[*simp*]

lemma *nonemptyCn*: $\langle Cn(A) \neq \{\} \rangle$
using *true-PL* **by** *auto*

lemma *Cn-true*: $\langle Cn(\{\top\}) = Cn(\{\}) \rangle$
using *Cn-same true-PL* **by** *auto*

lemma *Cn-false*: $\langle Cn(\{\perp\}) = UNIV \rangle$
using *false-PL* **by** *auto*

lemma *Cn-imp*: $\langle A \vdash (p \longrightarrow q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(A \cup \{p\}) \rangle$
and *Cn-imp-bis*: $\langle A \vdash (p \longrightarrow q) \longleftrightarrow Cn(A \cup \{q\}) \subseteq Cn(A \cup \{p\}) \rangle$
using *Cn-same imp-PL idempotency-L inclusion-L infer-def subset-insertI* **by** *force+*

lemma *Cn-not*: $\langle A \vdash \neg p \longleftrightarrow Cn(A \cup \{p\}) = UNIV \rangle$
using *Cn-false Cn-imp notE-PL not-PL* **by** *fastforce*

lemma *Cn-conj*: $\langle A \vdash (p \wedge q) \longleftrightarrow Cn(\{p\}) \cup Cn(\{q\}) \subseteq Cn(A) \rangle$
apply(*intro iffI conjI-PL, frule conjE1-PL, frule conjE2-PL*)
using *Cn-same Un-insert-right bot.extremum idempotency-L inclusion-L* **by** *auto*

lemma *Cn-conj-bis*: $\langle Cn(\{p \wedge q\}) = Cn(\{p, q\}) \rangle$
by (*unfold Cn-same*)
(*meson Supraclassical-logic.conj-PL Supraclassical-logic-axioms insert-subset*)

lemma *Cn-disj*: $\langle A \vdash (p \vee q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(A \cup \{\neg p\}) \rangle$
and *Cn-disj-bis*: $\langle A \vdash (p \vee q) \longleftrightarrow Cn(A \cup \{q\}) \subseteq Cn(A \cup \{\neg p\}) \rangle$
using *disj-PL Cn-same imp-PL idempotency-L inclusion-L infer-def subset-insertI*
by *force+*

lemma *Cn-equiv*: $\langle A \vdash (p \longleftrightarrow q) \longleftrightarrow Cn(A \cup \{p\}) = Cn(A \cup \{q\}) \rangle$
by (*metis Cn-imp-bis equivE1-PL equivE2-PL equivI-PL set-eq-subset*)

corollary *valid-nonemptyCn*: $\langle Cn(\{\}) \neq \{\} \rangle$
and *valid-Cn-imp*: $\langle \vDash (p \longrightarrow q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(\{p\}) \rangle$
and *valid-Cn-not*: $\langle \vDash (\neg p) \longleftrightarrow Cn(\{p\}) = UNIV \rangle$
and *valid-Cn-conj*: $\langle \vDash (p \wedge q) \longleftrightarrow Cn(\{p\}) \cup Cn(\{q\}) \subseteq Cn(\{\}) \rangle$
and *valid-Cn-disj*: $\langle \vDash (p \vee q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(\{\neg p\}) \rangle$
and *valid-Cn-equiv*: $\langle \vDash (p \longleftrightarrow q) \longleftrightarrow Cn(\{p\}) = Cn(\{q\}) \rangle$
using *nonemptyCn Cn-imp Cn-not Cn-conj Cn-disj Cn-equiv valid-def* **by** *auto*

— Finally, we group additional lemmas that were essential in further proofs

lemma *consistency*: $\langle Cn(\{p\}) \cap Cn(\{\neg p\}) = Cn(\{\}) \rangle$

proof

{ **fix** *q*

assume $\langle \{p\} \vdash q \rangle$ **and** $\langle \{\neg p\} \vdash q \rangle$
hence $\{\} \vdash p \longrightarrow q$ **and** $\{\} \vdash (\neg p) \longrightarrow q$
using *impI-PL* **by** *auto*
hence $\langle \{\} \vdash q \rangle$
using *ex-mid-PL* **by** (*rule-tac disjE-PL*[**where** $p=p$ **and** $q=\langle \neg p \rangle$]) *blast*
}
then show $\langle Cn(\{p\}) \cap Cn(\{\neg p\}) \subseteq Cn(\{\}) \rangle$ **by** (*simp add: subset-iff*)
next
show $\langle Cn(\{\}) \subseteq Cn(\{p\}) \cap Cn(\{\neg p\}) \rangle$ **by** (*simp add: monotonicity-L*)
qed

lemma *Cn-notnot*: $\langle Cn(\{\neg (\neg \varphi)\}) = Cn(\{\varphi\}) \rangle$
by (*metis (no-types, opaque-lifting) notnot-PL valid-Cn-equiv valid-equiv-PL*)

lemma *conj-com*: $\langle A \vdash p \wedge q \longleftrightarrow A \vdash q \wedge p \rangle$
using *conj-PL* **by** *auto*

lemma *conj-com-Cn*: $\langle Cn(\{p \wedge q\}) = Cn(\{q \wedge p\}) \rangle$
by (*simp add: Cn-conj-bis insert-commute*)

lemma *disj-com*: $\langle A \vdash p \vee q \longleftrightarrow A \vdash q \vee p \rangle$

proof –
{ **fix** $p q$
have $\langle A \vdash p \vee q \implies A \vdash q \vee p \rangle$
apply(*erule disjE-PL*)
using *assumption-L disjI2-PL disjI1-PL impI-PL* **by** *auto*
}
then show *?thesis* **by** *auto*
qed

lemma *disj-com-Cn*: $\langle Cn(\{p \vee q\}) = Cn(\{q \vee p\}) \rangle$
unfolding *Cn-same* **using** *disj-com* **by** *simp*

lemma *imp-contrapos*: $\langle A \vdash p \longrightarrow q \longleftrightarrow A \vdash \neg q \longrightarrow \neg p \rangle$
by (*metis Cn-not Un-insert-left Un-insert-right imp-PL notnot-PL*)

lemma *equiv-negation*: $\langle A \vdash p \longleftrightarrow q \longleftrightarrow A \vdash \neg p \longleftrightarrow \neg q \rangle$
using *equivE1-PL equivE2-PL equivI-PL imp-contrapos* **by** *blast*

lemma *imp-trans*: $\langle A \vdash p \longrightarrow q \implies A \vdash q \longrightarrow r \implies A \vdash p \longrightarrow r \rangle$
using *Cn-imp-bis* **by** *auto*

lemma *imp-recovery0*: $\langle A \vdash p \vee (p \longrightarrow q) \rangle$
apply(*subst disj-PL, subst imp-contrapos*)
using *assumption-L impI-PL* **by** *auto*

lemma *imp-recovery1*: $\langle A \cup \{p \longrightarrow q\} \vdash p \implies A \vdash p \rangle$
using *disjE-PL*[*OF imp-recovery0, of A p p q*] *assumption-L imp-PL* **by** *auto*

lemma *imp-recovery2*: $\langle A \vdash p \longrightarrow. q \implies A \vdash (q \longrightarrow. p) \longrightarrow. p \implies A \vdash q \rangle$
using *imp-PL imp-recovery1 imp-trans by blast*

lemma *impI2*: $\langle A \vdash q \implies A \vdash p \longrightarrow. q \rangle$
by (*meson assumption-L impI-PL in-mono sup-ge1 transitivity2-L*)

lemma *conj-equiv*: $\langle A \vdash p \implies A \vdash ((p \wedge. q) \longleftrightarrow. q) \rangle$
by (*metis Un-insert-right assumption-L conjE2-PL conjI-PL equiv-PL impI2 imp-PL insertI1 sup-bot.right-neutral*)

lemma *conj-imp*: $\langle A \vdash (p \wedge. q) \longrightarrow. r \longleftrightarrow A \vdash p \longrightarrow. (q \longrightarrow. r) \rangle$
proof

assume $A \vdash (p \wedge. q) \longrightarrow. r$
then have $Cn (A \cup \{r\}) \subseteq Cn (A \cup \{p, q\})$
by (*metis (no-types) Cn-conj-bis Cn-imp-bis Cn-union Un-insert-right sup-bot.right-neutral*)
then show $\langle A \vdash p \longrightarrow. (q \longrightarrow. r) \rangle$
by (*metis Un-insert-right impI-PL inclusion-L infer-def insert-commute insert-subset subset-eq sup-bot.right-neutral*)
next
assume $A \vdash p \longrightarrow. (q \longrightarrow. r)$
then have $A \cup \{p\} \cup \{q\} \vdash r$
using *imp-PL by auto*
then show $A \vdash (p \wedge. q) \longrightarrow. r$
by (*metis (full-types) Cn-conj-bis Cn-union impI-PL infer-def insert-is-Un sup-assoc*)
qed

lemma *conj-not-impE-PL*: $\langle A \vdash (p \wedge. q) \longrightarrow. r \implies A \vdash (p \wedge. \neg q) \longrightarrow. r \implies A \vdash p \longrightarrow. r \rangle$
by (*meson conj-imp disjE-PL ex-mid-PL imp-PL*)

lemma *disj-notE-PL*: $\langle A \vdash q \implies A \vdash p \vee. \neg q \implies A \vdash p \rangle$
using *Cn-imp Cn-imp-bis Cn-not disjE-PL notnot-PL by blast*

lemma *disj-not-impE-PL*: $\langle A \vdash (p \vee. q) \longrightarrow. r \implies A \vdash (p \vee. \neg q) \longrightarrow. r \implies A \vdash r \rangle$
by (*metis Un-insert-right disjE-PL disj-PL disj-com ex-mid-PL insert-commute sup-bot.right-neutral*)

lemma *imp-conj*: $\langle A \vdash p \longrightarrow. q \implies A \vdash r \longrightarrow. s \implies A \vdash (p \wedge. r) \longrightarrow. (q \wedge. s) \rangle$
apply(*intro impI-PL conjI-PL, unfold imp-PL[symmetric]*)
by (*meson assumption-L conjE1-PL conjE2-PL imp-trans infer-def insertI1 validD-L valid-imp-PL*) $+$

lemma *conj-overlap*: $\langle A \vdash (p \wedge. q) \longleftrightarrow A \vdash (p \wedge. ((\neg p) \vee. q)) \rangle$
by (*meson conj-PL disjI2-PL disj-com disj-notE-PL*)

lemma *morgan*: $\langle A \vdash \neg (p \wedge. q) \longleftrightarrow A \vdash (\neg p) \vee. (\neg q) \rangle$

by (meson conj-imp disj-PL disj-com imp-PL imp-contrapos notE-PL notI-PL)

lemma conj-superexpansion1: $\langle A \vdash \neg (p \wedge q) \wedge \neg p \longleftrightarrow A \vdash \neg p \rangle$
 using conj-PL disjI1-PL morgan by auto

lemma conj-superexpansion2: $\langle A \vdash (p \vee q) \wedge p \longleftrightarrow A \vdash p \rangle$
 using conj-PL disjI1-PL by auto

end

2.3 Compact Logic

If the logic is compact, which means that any result is based on a finite set of hypothesis

locale Compact-logic = Tarskian-logic +
 assumes compactness-L: $\langle A \vdash \varphi \implies (\exists A'. A' \subseteq A \wedge \text{finite } A' \wedge A' \vdash \varphi) \rangle$

begin

Very important lemma preparing the application of the Zorn's lemma. It states that in a compact logic, we can not deduce φ if we accumulate an infinity of hypothesis groups which locally do not deduce phi

lemma chain-closure: $\langle \neg \vdash \varphi \implies \text{subset.chain } \{B. \neg B \vdash \varphi\} C \implies \neg \bigcup C \vdash \varphi \rangle$
proof

assume a: $\langle \text{subset.chain } \{B. \neg B \vdash \varphi\} C \rangle$ and b: $\langle \neg \vdash \varphi \rangle$ and $\langle \bigcup C \vdash \varphi \rangle$
 then obtain A' where c: $\langle A' \subseteq \bigcup C \rangle$ and d: $\langle \text{finite } A' \rangle$ and e: $\langle A' \vdash \varphi \rangle$ using compactness-L by blast

define f where f: $\langle f \equiv \lambda a. \text{SOME } B. B \in C \wedge a \in B \rangle$

have g: $\langle \text{finite } (f \text{ ' } A') \rangle$ using f d by simp

have h: $\langle (f \text{ ' } A') \subseteq C \rangle$

unfolding f by auto (metis (mono-tags, lifting) Union-iff c someI-ex subset-eq)

have i: $\langle \text{subset.chain } \{B. \neg B \vdash \varphi\} (f \text{ ' } A') \rangle$ using a h

using a h by (meson subsetD subset-chain-def subset-trans)

have $\langle A' \neq \{\} \implies \bigcup (f \text{ ' } A') \in \{B. \neg B \vdash \varphi\} \rangle$ using g i

by (meson Union-in-chain image-is-empty subset-chain-def subset-eq)

hence j: $\langle A' \neq \{\} \implies \neg \bigcup (f \text{ ' } A') \vdash \varphi \rangle$ by simp

have $\langle A' \subseteq \bigcup (f \text{ ' } A') \rangle$

unfolding f by auto (metis (no-types, lifting) Union-iff c someI-ex subset-iff)

with j e b show False

by (metis infer-def monotonicity-L subsetD valid-def)

qed

end

end

3 Remainders

In AGM, one important feature is to eliminate some proposition from a set of propositions by ensuring that the set of retained clauses is maximal and that nothing among these clauses allows to retrieve the eliminated proposition

3.1 Remainders in a Tarskian logic

In a general context of a Tarskian logic, we consider a descriptive definition (by comprehension)

context *Tarskian-logic*

begin

definition *remainder*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle$ (**infix** $\langle .\perp. \rangle$ 55)

where *rem*: $\langle A .\perp. \varphi \equiv \{B. B \subseteq A \wedge \neg B \vdash \varphi \wedge (\forall B' \subseteq A. B \subset B' \longrightarrow B' \vdash \varphi)\} \rangle$

lemma *rem-inclusion*: $\langle B \in A .\perp. \varphi \Longrightarrow B \subseteq A \rangle$

by (*auto simp add:rem split:if-splits*)

lemma *rem-closure*: $\langle K = Cn(A) \Longrightarrow B \in K .\perp. \varphi \Longrightarrow B = Cn(B) \rangle$

apply (*cases* $\langle K .\perp. \varphi = \{\} \rangle$, *simp*)

by (*simp add:rem infer-def*) (*metis idempotency-L inclusion-L monotonicity-L psubsetI*)

lemma *remainder-extensionality*: $\langle Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow A .\perp. \varphi = A .\perp. \psi \rangle$

unfolding *rem infer-def* **apply** *safe*

by (*simp-all add: Cn-same*) *blast+*

lemma *nonconsequence-remainder*: $\langle A .\perp. \varphi = \{A\} \longleftrightarrow \neg A \vdash \varphi \rangle$

unfolding *rem* **by** *auto*

— As we will see further, the other direction requires compactness!

lemma *taut2emptyrem*: $\langle \vdash \varphi \Longrightarrow A .\perp. \varphi = \{\} \rangle$

unfolding *rem* **by** (*simp add: infer-def validD-L*)

end

3.2 Remainders in a supraclassical logic

In case of a supraclassical logic, remainders get impressive properties

context *Supraclassical-logic*

begin

— As an effect of being maximal, a remainder keeps the eliminated proposition in its propositions hypothesis

lemma *remainder-recovery*: $\langle K = Cn(A) \implies K \vdash \psi \implies B \in K \perp. \varphi \implies B \vdash \varphi \longrightarrow. \psi \rangle$
proof –
 { **fix** ψ **and** B
 assume $a:\langle K = Cn(A) \rangle$ **and** $c:\langle \psi \in K \rangle$ **and** $d:\langle B \in K \perp. \varphi \rangle$ **and** $e:\langle \varphi \longrightarrow. \psi \notin Cn(B) \rangle$
 with a **have** $f:\langle \varphi \longrightarrow. \psi \in K \rangle$ **using** *impI2 infer-def by blast*
 with d e **have** $\langle \varphi \in Cn(B \cup \{\varphi \longrightarrow. \psi\}) \rangle$
 apply (*simp add:rem, elim conjE*)
 by (*metis dual-order.order-iff-strict inclusion-L insert-subset*)
 with d **have** *False* **using** *rem imp-recovery1*
 by (*metis (no-types, lifting) CollectD infer-def*)
 }
thus $\langle K = Cn(A) \implies K \vdash \psi \implies B \in K \perp. \varphi \implies B \vdash \varphi \longrightarrow. \psi \rangle$
using *idempotency-L by auto*
qed

— When you remove some proposition φ several other propositions can be lost. An important lemma states that the resulting remainder is also a remainder of any lost proposition

lemma *remainder-recovery-bis*: $\langle K = Cn(A) \implies K \vdash \psi \implies \neg B \vdash \psi \implies B \in K \perp. \varphi \implies B \in K \perp. \psi \rangle$
proof–
assume $a:\langle K = Cn(A) \rangle$ **and** $b:\langle \neg B \vdash \psi \rangle$ **and** $c:\langle B \in K \perp. \varphi \rangle$ **and** $d:\langle K \vdash \psi \rangle$
hence $d:\langle B \vdash \varphi \longrightarrow. \psi \rangle$ **using** *remainder-recovery by simp*
with c **show** $\langle B \in K \perp. \psi \rangle$
by (*simp add:rem*) (*meson b dual-order.trans infer-def insert-subset monotonicity-L mp-PL order-refl psubset-imp-subset*)
qed

corollary *remainder-recovery-imp*: $\langle K = Cn(A) \implies K \vdash \psi \implies \Vdash (\psi \longrightarrow. \varphi) \implies B \in K \perp. \varphi \implies B \in K \perp. \psi \rangle$
apply(*rule remainder-recovery-bis, simp-all*)
by (*simp add:rem*) (*meson infer-def mp-PL validD-L*)

— If we integrate back the eliminated proposition into the remainder, we retrieve the original set!

lemma *remainder-expansion*: $\langle K = Cn(A) \implies K \vdash \psi \implies \neg B \vdash \psi \implies B \in K \perp. \varphi \implies B \oplus \psi = K \rangle$
proof
assume $a:\langle K = Cn(A) \rangle$ **and** $b:\langle K \vdash \psi \rangle$ **and** $c:\langle \neg B \vdash \psi \rangle$ **and** $d:\langle B \in K \perp. \varphi \rangle$
then show $\langle B \oplus \psi \subseteq K \rangle$
by (*metis Un-insert-right expansion-def idempotency-L infer-def insert-subset monotonicity-L rem-inclusion sup-bot.right-neutral*)

next

assume $a:\langle K = Cn(A) \rangle$ **and** $b:\langle K \vdash \psi \rangle$ **and** $c:\langle \neg B \vdash \psi \rangle$ **and** $d:\langle B \in K \perp. \varphi \rangle$
 { **fix** χ
 assume $\langle \chi \in K \rangle$
 hence $e:\langle B \vdash \varphi \longrightarrow. \chi \rangle$ **using** *remainder-recovery[OF a - d, of χ] assumption-L*


```

by blast
  have  $\langle \psi \in K \rangle$  using a b idempotency-L infer-def by blast
  hence  $f: \langle B \cup \{\psi\} \vdash \varphi \rangle$  using b c d apply(simp add:rem)
    by (meson inclusion-L insert-iff insert-subsetI less-le-not-le subset-iff)
  from e f have  $\langle B \cup \{\psi\} \vdash \chi \rangle$  using imp-PL imp-trans by blast
}
then show  $\langle K \subseteq B \oplus \psi \rangle$ 
  by (simp add: expansion-def subsetI)
qed

```

To eliminate a conjunction, we only need to remove one side

```

lemma remainder-conj:  $\langle K = Cn(A) \implies K \vdash \varphi \wedge \psi \implies K \perp. (\varphi \wedge \psi) = (K \perp. \varphi) \cup (K \perp. \psi) \rangle$ 
  apply(intro subset-antisym Un-least subsetI, simp add:rem)
  apply (meson conj-PL infer-def)
  using remainder-recovery-imp[of K A  $\langle \varphi \wedge \psi \rangle \varphi$ ]
  apply (meson assumption-L conjE1-PL singletonI subsetI valid-imp-PL)
  using remainder-recovery-imp[of K A  $\langle \varphi \wedge \psi \rangle \psi$ ]
  by (meson assumption-L conjE2-PL singletonI subsetI valid-imp-PL)

```

end

3.3 Remainders in a compact logic

In case of a supraclassical logic, remainders get impressive properties

```

context Compact-logic
begin

```

The following lemma is the Lindembaum's lemma requiring the Zorn's lemma (already available in standard Isabelle/HOL). For more details, please refer to the book "Theory of logical calculi" [5]. This very important lemma states that we can get a maximal set (remainder B') starting from any set B if this latter does not infer the proposition φ we want to eliminate

```

lemma upper-remainder:  $\langle B \subseteq A \implies \neg B \vdash \varphi \implies \exists B'. B \subseteq B' \wedge B' \in A \perp. \varphi \rangle$ 

```

proof –

```

  assume a:  $\langle B \subseteq A \rangle$  and b:  $\langle \neg B \vdash \varphi \rangle$ 
  have c:  $\langle \neg \Vdash \varphi \rangle$ 
    using b infer-def validD-L by blast
  define  $\mathcal{B}$  where  $\mathcal{B} \equiv \{B'. B \subseteq B' \wedge B' \subseteq A \wedge \neg B' \vdash \varphi\}$ 
  have d:  $\langle \text{subset.chain } \mathcal{B} C \implies \text{subset.chain } \{B. \neg B \vdash \varphi\} C \rangle$  for C
    unfolding  $\mathcal{B}$ -def
    by (simp add: le-fun-def less-eq-set-def subset-chain-def)
  have e:  $\langle C \neq \{\} \implies \text{subset.chain } \mathcal{B} C \implies B \subseteq \bigcup C \rangle$  for C
    by (metis (no-types, lifting)  $\mathcal{B}$ -def subset-chain-def less-eq-Sup mem-Collect-eq subset-iff)
  { fix C
    assume f:  $\langle C \neq \{\} \rangle$  and g:  $\langle \text{subset.chain } \mathcal{B} C \rangle$ 

```

```

have  $\langle \bigcup C \in \mathcal{B} \rangle$ 
  using  $\mathcal{B}$ -def e[OF f g] chain-closure[OF c d[OF g]]
    by simp (metis (no-types, lifting) CollectD Sup-least Sup-subset-mono g
subset.chain-def subset-trans)
  } note f=this
have  $\langle \text{subset.chain } \mathcal{B} \ C \implies \exists U \in \mathcal{B}. \forall X \in C. X \subseteq U \rangle$  for C
  apply (cases  $\langle C \neq \{\} \rangle$ )
  apply (meson Union-upper f)
  using  $\mathcal{B}$ -def a b by blast
with subset-Zorn[OF this, simplified] obtain B' where f:  $\langle B' \in \mathcal{B} \wedge (\forall X \in \mathcal{B}. B' \subseteq X \longrightarrow X = B') \rangle$  by auto
then show ?thesis
  by (simp add:rem  $\mathcal{B}$ -def, rule-tac x=B' in exI) (metis psubsetE subset-trans)
qed

```

— An immediate corollary ruling tautologies

corollary emptyrem2taut: $\langle A \perp. \varphi = \{\} \implies \Vdash \varphi \rangle$

by (metis bot.extremum empty-iff upper-remainder valid-def)

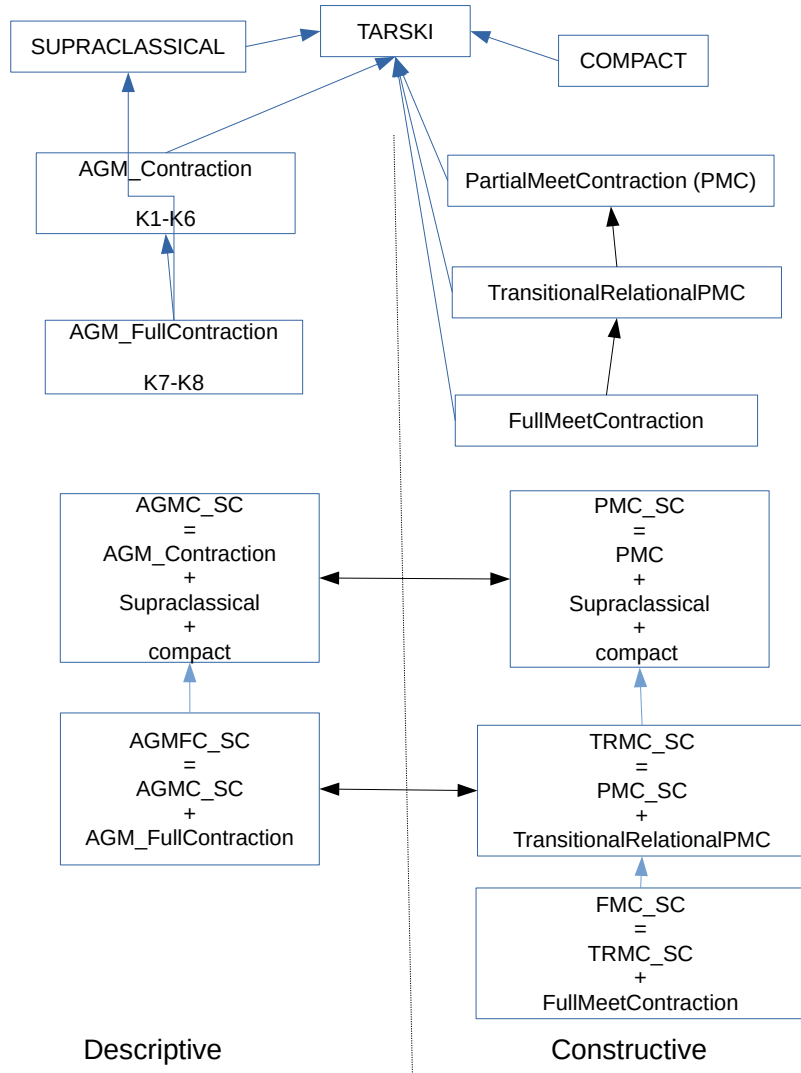
end

end

4 Contractions

The first operator of belief change of the AGM framework is contraction. This operator consist to remove a sentence φ from a belief set K in such a way that K no longer imply φ .

In the following we will first axiomatize such operators at different levels of logics (Tarskian, supraclassical and compact) and then we will give constructions satisfying these axioms. The following graph summarizes all equivalences we established:



We will use the extension feature of locales in Isabelle/HOL to incrementally define the contraction operator as shown by blue arrows in the previous figure. Then, using the interpretation feature of locales, we will prove the equivalence between descriptive and constructive approaches at each level depending on the adopted logics (black arrows).

4.1 AGM contraction postulates

The operator of contraction is denoted by the symbol \div and respects the six following conditions :

- *contract-closure* : a belief set K contracted by φ should be logically closed
- *contract-inclusion* : a contracted set K should be a subset of the original one
- *contract-vacuity* : if φ is not included in a set K then the contraction of K by φ involves no change at all
- *contract-success* : if a set K is contracted by φ then K does not imply φ
- *contract-recovery*: all propositions removed in a set K by contraction of φ will be recovered by expansion of φ
- *contract-extensionality* : Extensionality guarantees that the logic of contraction is extensional in the sense of allowing logically equivalent sentences to be freely substituted for each other

locale *AGM-Contraction* = *Tarskian-logic* +

fixes *contraction*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set} \rangle$ (**infix** $\langle \div \rangle$ 55)

assumes *contract-closure*: $\langle K = Cn(A) \Longrightarrow K \div \varphi = Cn(K \div \varphi) \rangle$
and *contract-inclusion*: $\langle K = Cn(A) \Longrightarrow K \div \varphi \subseteq K \rangle$
and *contract-vacuity*: $\langle K = Cn(A) \Longrightarrow \varphi \notin K \Longrightarrow K \div \varphi = K \rangle$
and *contract-success*: $\langle K = Cn(A) \Longrightarrow \varphi \notin Cn(\{\}) \Longrightarrow \varphi \notin K \div \varphi \rangle$
and *contract-recovery*: $\langle K = Cn(A) \Longrightarrow K \subseteq ((K \div \varphi) \oplus \varphi) \rangle$
and *contract-extensionality*: $\langle K = Cn(A) \Longrightarrow Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow K \div \varphi = K \div \psi \rangle$

A full contraction is defined by two more postulates to rule the conjunction. We base on a supraclassical logic.

- *contract-conj-overlap* : An element in both $K \div \varphi$ and $K \div \psi$ is also an element of $K \div (\varphi \wedge \psi)$
- *contract-conj-inclusion* : If φ not in $K \div (\varphi \wedge \psi)$ then all elements removed by this contraction are also removed from $K \div \varphi$

locale *AGM-FullContraction* = *AGM-Contraction* + *Supraclassical-logic* +

assumes *contract-conj-overlap*: $\langle K = Cn(A) \Longrightarrow (K \div \varphi) \cap (K \div \psi) \subseteq (K \div (\varphi \wedge \psi)) \rangle$

and *contract-conj-inclusion*: $\langle K = Cn(A) \Longrightarrow \varphi \notin (K \div (\varphi \wedge \psi)) \Longrightarrow ((K \div (\varphi \wedge \psi)) \subseteq (K \div \varphi)) \rangle$

begin

— two important lemmas/corollaries that can replace the two assumptions *contract-conj-overlap* and *contract-conj-inclusion*

contract-conj-overlap-variant does not need ψ to occur in the left side!

corollary *contract-conj-overlap-variant*: $\langle K = Cn(A) \implies (K \div \varphi) \cap Cn(\{\varphi\}) \subseteq (K \div (\varphi \wedge \psi)) \rangle$

proof –

assume $a: \langle K = Cn(A) \rangle$
{ assume $b: \langle K \vdash \varphi \rangle$ **and** $c: \langle K \vdash \psi \rangle$
hence $d: \langle K \div (\varphi \wedge \psi) = K \div (\varphi \wedge ((\neg \varphi) \vee \psi)) \rangle$
apply (*rule-tac contract-extensionality*[OF a])
using *conj-overlap*[of $\varphi \psi$] **by** (*simp add: Cn-same*)
have $e: \langle K \cap Cn \{\varphi\} \subseteq K \div (\neg \varphi \vee \psi) \rangle$
proof (*safe*)
fix χ
assume $f: \langle \chi \in K \rangle$ **and** $g: \langle \chi \in Cn \{\varphi\} \rangle$
have $\langle K \div (\neg \varphi \vee \psi) \vdash (\neg \varphi \vee \psi) \longrightarrow \chi \rangle$
by (*metis a contract-recovery expansion-def f impI-PL infer-def subset-eq*)
hence $\langle K \div (\neg \varphi \vee \psi) \vdash \neg \varphi \longrightarrow \chi \rangle$
by (*meson disjI1-PL imp-trans inclusion-L infer-def insert-subset validD-L valid-imp-PL*)
with g **show** $\langle \chi \in K \div (\neg \varphi \vee \psi) \rangle$
by (*metis a contract-closure disjE-PL ex-mid-PL infer-def validD-L valid-imp-PL*)
qed
have *?thesis*
unfolding d **using** e *contract-conj-overlap*[OF a, of $\varphi \langle (\neg \varphi \vee \psi) \rangle$] a
contract-inclusion **by force**
}
then show *?thesis*
apply (*cases* $\langle \neg K \vdash \varphi \vee \neg K \vdash \psi \rangle$)
by (*metis IntE a assumption-L conjE1-PL conjE2-PL contract-inclusion contract-vacuity subsetD subsetI*) *blast*
qed

contract-conj-inclusion-variant: Everything retained in $K \div (\varphi \wedge \psi)$ is retained in $K \div \psi$

corollary *contract-conj-inclusion-variant* : $\langle K = Cn(A) \implies (K \div (\varphi \wedge \psi) \subseteq (K \div \varphi)) \vee (K \div (\varphi \wedge \psi) \subseteq (K \div \psi)) \rangle$

proof –

assume $a: \langle K = Cn(A) \rangle$
{ assume $d: \langle \varphi \in (K \div (\varphi \wedge \psi)) \wedge \psi \in (K \div (\varphi \wedge \psi)) \rangle$
hence $\langle \varphi \wedge \psi \in (K \div (\varphi \wedge \psi)) \rangle$
using *Supraclassical-logic.conjI-PL Supraclassical-logic-axioms a contract-closure*
by *fastforce*
with d **have** *?thesis*
by (*metis (no-types, lifting) Supraclassical-logic.valid-conj-PL Supraclassical-logic-axioms*)

```

    Tarskian-logic.valid-expansion Tarskian-logic-axioms a contract-closure
contract-inclusion
    contract-recovery contract-success dual-order.trans expansion-def)
}
then show ?thesis
  by (metis a conj-com-Cn contract-conj-inclusion contract-extensionality)
qed

end

```

4.2 Partial meet contraction definition

A partial meet contraction of K by φ is the intersection of some sets that not imply φ . We define these sets as the "remainders" $(K \perp \varphi)$. The function of selection γ select the best set of the remainders that do not imply φ . This function respect these postulates :

- *is-selection* : if there exist some set that do not imply φ then the function selection γ is a subset of these sets
- *tautology-selection* : if there is no set that do not imply φ then the result of the selection function is K
- *nonempty-selection* : An empty selection function do not exist
- *extensional-selection* : Two proposition with the same closure have the same selection function

locale *PartialMeetContraction* = *Tarskian-logic* +

```

fixes selection::('a set  $\Rightarrow$  'a  $\Rightarrow$  'a set set) ( $\langle \gamma \rangle$ )
assumes is-selection:  $\langle K = Cn(A) \Rightarrow (K \perp \varphi) \neq \{\} \Rightarrow \gamma K \varphi \subseteq (K \perp \varphi) \rangle$ 
assumes tautology-selection:  $\langle K = Cn(A) \Rightarrow (K \perp \varphi) = \{\} \Rightarrow \gamma K \varphi = \{K\} \rangle$ 
assumes nonempty-selection:  $\langle K = Cn(A) \Rightarrow \gamma K \varphi \neq \{\} \rangle$ 
assumes extensional-selection:  $\langle K = Cn(A) \Rightarrow Cn(\{\varphi\}) = Cn(\{\psi\}) \Rightarrow \gamma K \varphi = \gamma K \psi \rangle$ 

```

— extensionality seems very hard to implement for a constructive approach, one basic implementation will be to ignore A and φ and only base on $A \perp \varphi$ that is already proved as extensional (lemma *remainder-extensionality*)

begin

A partial meet is the intersection of set of selected element.

definition (in *Tarskian-logic*) *meet-contraction*::('a set \Rightarrow ('a set \Rightarrow 'a \Rightarrow 'a set set) \Rightarrow 'a \Rightarrow 'a set) ($\langle - \div - \rightarrow [60,50,60]55 \rangle$)

where mc : $\langle (A \div_{\gamma} \varphi) \equiv \bigcap (\gamma A \varphi) \rangle$

Following this definition 4 postulates of AGM can be proved on a partial meet contraction:

- *contract-inclusion*
- *contract-vacuity*
- *contract-closure*
- *contract-extensionality*

pmc-inclusion : a partial meet contraction is a subset of the contracted set

lemma *pmc-inclusion*: $\langle K = Cn(A) \implies K \div_{\gamma} \varphi \subseteq K \rangle$

apply (*cases* $\langle (K \perp. \varphi) = \{\} \rangle$, *simp-all add: mc tautology-selection*)

by (*meson Inf-less-eq in-mono is-selection nonempty-selection rem-inclusion*)

pmc-vacuity : if φ is not included in a set K then the partial meet contraction of K by φ involves not change at all

lemma *pmc-vacuity*: $\langle K = Cn(A) \implies \neg K \vdash \varphi \implies K \div_{\gamma} \varphi = K \rangle$

unfolding *mc nonconsequence-remainder*

by (*metis Inf-superset-mono Un-absorb1 cInf-singleton insert-not-empty is-selection mc nonconsequence-remainder pmc-inclusion sup-commute*)

pmc-closure : a partial meet contraction is logically closed

lemma *pmc-closure*: $\langle K = Cn(A) \implies (K \div_{\gamma} \varphi) = Cn(K \div_{\gamma} \varphi) \rangle$

proof (*rule subset-antisym, simp-all add:inclusion-L mc transitivity-L, goal-cases*)

case 1

have $\langle \bigcap (\gamma (Cn A) \varphi) = \bigcap \{Cn(B) \mid B. B \in \gamma (Cn A) \varphi\} \rangle$

by *auto (metis idempotency-L insert-absorb insert-iff insert-subset is-selection rem-closure tautology-selection)+*

from *Cn-Inter[OF this]* **show** *?case by blast*

qed

pmc-extensionality : Extensionality guarantees that the logic of contraction is extensional in the sense of allowing logically equivalent sentences to be freely substituted for each other

lemma *pmc-extensionality*: $\langle K = Cn(A) \implies Cn(\{\varphi\}) = Cn(\{\psi\}) \implies K \div_{\gamma} \varphi = K \div_{\gamma} \psi \rangle$

by (*metis extensional-selection mc*)

pmc-tautology : if φ is a tautology then the partial meet contraction of K by φ is K

lemma *pmc-tautology*: $\langle K = Cn(A) \implies \Vdash \varphi \implies K \div_{\gamma} \varphi = K \rangle$

by (*simp add: mc taut2emptyrem tautology-selection*)

completion is a an operator that can build an equivalent selection from an existing one

definition (in *Tarskian-logic*) *completion*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle \Rightarrow 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle$ ($\langle * \rangle$)

where $\langle * \gamma A \varphi \equiv \text{if } (A \perp. \varphi) = \{\} \text{ then } \{A\} \text{ else } \{B. B \in A \perp. \varphi \wedge \bigcap (\gamma A \varphi) \subseteq B\} \rangle$

lemma *selection-completion*: $K = Cn(A) \Longrightarrow \gamma K \varphi \subseteq * \gamma K \varphi$
using *completion-def is-selection tautology-selection* **by** *fastforce*

lemma (in *Tarskian-logic*) *completion-completion*: $K = Cn(A) \Longrightarrow * (* \gamma) K \varphi = * \gamma K \varphi$
by (*auto simp add:completion-def*)

lemma *pmc-completion*: $\langle K = Cn(A) \Longrightarrow K \div * \gamma \varphi = K \div \gamma \varphi \rangle$
apply (*auto simp add: mc completion-def tautology-selection*)
by (*metis Inter-lower equals0D in-mono is-selection*)

end

A transitively relational meet contraction is a partial meet contraction using a binary relation between the elements of the selection function

A relation is :

- transitive (*trans-rel*)
- non empty (there is always an element preferred to the others (*nonempty-rel*))

A selection function γ_{TR} is transitively relational *rel-sel* with the following condition :

- If the the remainders $K \perp. \varphi$ is empty then the selection function return K
- Else the selection function return a non empty transitive relation on the remainders

locale *TransitivelyRelationalMeetContraction* = *Tarskian-logic* +

fixes *relation*:: $\langle 'a \text{ set} \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set} \Rightarrow \text{bool} \rangle$ ($\langle - \preceq - \rightarrow [60,50,60]55 \rangle$)

assumes *trans-rel*: $\langle K = Cn(A) \Longrightarrow B \preceq_K C \Longrightarrow C \preceq_K D \Longrightarrow B \preceq_K D \rangle$

assumes *nonempty-rel*: $\langle K = Cn(A) \Longrightarrow (K \perp. \varphi) \neq \{\} \Longrightarrow \exists B \in (K \perp. \varphi). (\forall C \in (K \perp. \varphi). C \preceq_K B) \rangle$ — pas clair dans la litterature

fixes *rel-sel*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle$ ($\langle \gamma_{TR} \rangle$)

defines *rel-sel*: $\langle \gamma_{TR} K \varphi \equiv \text{if } (K \perp. \varphi) = \{\} \text{ then } \{K\}$
 $\text{else } \{B. B \in (K \perp. \varphi) \wedge (\forall C \in (K \perp. \varphi). C$
 $\preceq_K B)\} \rangle$

begin

A transitively relational selection function respect the partial meet contraction postulates.

sublocale *PartialMeetContraction* **where** *selection* = γ_{TR}
apply(*unfold-locales*)
apply(*simp-all add: rel-sel*)
using *nonempty-rel* **apply** *blast*
using *remainder-extensionality* **by** *blast*

end

A full meet contraction is a limiting case of the partial meet contraction where if the remainders are not empty then the selection function return all the remainders (as defined by *full-sel*)

locale *FullMeetContraction* = *Tarskian-logic* +

fixes *full-sel*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle$ ($\langle \gamma_{FC} \rangle$)
defines *full-sel*: $\langle \gamma_{FC} K \varphi \equiv \text{if } K \perp. \varphi = \{\} \text{ then } \{K\} \text{ else } K \perp. \varphi$

begin

A full selection and a relation ? is a transitively relational meet contraction postulates.

sublocale *TransitivelyRelationalMeetContraction* **where** *relation* = $\langle \lambda K A B. \text{True} \rangle$ **and** *rel-sel*= γ_{FC}
by (*unfold-locales*, *auto simp add:full-sel*, *rule eq-reflection*, *simp*)

end

4.3 Equivalence of partial meet contraction and AGM contraction

locale *PMC-SC* = *PartialMeetContraction* + *Supraclassical-logic* + *Compact-logic*

begin

In a context of a supraclassical and a compact logic the two remaining postulates of AGM contraction :

- *contract-recovery*
- *contract-success* can be proved on a partial meet contraction.

pmc-recovery : all proposition removed by a partial meet contraction of φ will be recovered by the expansion of φ

lemma *pmc-recovery*: $\langle K = Cn(A) \implies K \subseteq ((K \div_{\gamma} \varphi) \oplus \varphi) \rangle$
apply(*cases* $\langle K \perp. \varphi = \{\} \rangle$, *simp-all* (*no-asm*) *add:mc expansion-def*)
using *inclusion-L tautology-selection* **apply** *fastforce*
proof –
assume $a:\langle K = Cn(A) \rangle$ **and** $b:\langle K \perp. \varphi \neq \{\} \rangle$
{ **fix** ψ
assume $d:\langle K \vdash \psi \rangle$
have $\langle \varphi \longrightarrow. \psi \in \bigcap (\gamma K \varphi) \rangle$
using *is-selection[OF a b]*
by *auto* (*metis a d infer-def rem-closure remainder-recovery subsetD*)
}
with a b **show** $\langle K \subseteq Cn(\text{insert } \varphi (\bigcap (\gamma K \varphi))) \rangle$
by (*metis* (*no-types, lifting*) *Un-commute assumption-L imp-PL infer-def insert-is-Un subsetI*)
qed

pmc-success : a partial meet contraction of K by φ not imply φ

lemma *pmc-success*: $\langle K = Cn(A) \implies \varphi \notin Cn(\{\}) \implies \varphi \notin K \div_{\gamma} \varphi \rangle$
proof
assume $a:\langle K = Cn(A) \rangle$ **and** $b:\langle \varphi \notin Cn(\{\}) \rangle$ **and** $c:\langle \varphi \in K \div_{\gamma} \varphi \rangle$
from c **show** *False* **unfolding** *mc*
proof(*cases* $\langle K \perp. \varphi = \{K\} \rangle$)
case *True*
then **show** *?thesis*
by (*meson assumption-L c nonconsequence-remainder pmc-inclusion[OF a] subsetD*)
next
case *False*
hence $\langle \forall B \in K \perp. \varphi. \varphi \notin B \rangle$ **using** *assumption-L rem* **by** *auto*
moreover **have** $\langle K \perp. \varphi \neq \{\} \rangle$ **using** b *emptyrem2taut validD-L* **by** *blast*
ultimately **show** *?thesis*
using b c *mc nonempty-selection[OF a] validD-L emptyrem2taut is-selection[OF a]*
by (*metis Inter-iff bot.extremum-uniqueI subset-iff*)
qed
qed

As a partial meet contraction has been proven to respect all postulates of AGM contraction the equivalence between the both are straightforward

sublocale *AGM-Contraction* **where** *contraction* = $\langle \lambda A \varphi. A \div_{\gamma} \varphi \rangle$
using *pmc-closure pmc-inclusion pmc-vacuity*
pmc-success pmc-recovery pmc-extensionality
expansion-def idempotency-L infer-def
by (*unfold-locales*) *metis+*

end

locale *AGMC-SC* = *AGM-Contraction* + *Supraclassical-logic* + *Compact-logic*

begin

obs 2.5 page 514

definition *AGM-selection*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle (\langle \gamma_{AGM} \rangle)$

where *AGM-sel*: $\langle \gamma_{AGM} A \varphi \equiv \text{if } A \perp. \varphi = \{\} \text{ then } \{A\} \text{ else } \{B. B \in A \perp. \varphi \wedge A \div \varphi \subseteq B\} \rangle$

The selection function γ_{AGM} respect the partial meet contraction postulates

sublocale *PartialMeetContraction* **where** *selection* = γ_{AGM}

proof(*unfold-locales, unfold AGM-sel, simp-all, goal-cases*)

case (1 *K A* φ) — *non-emptiness* of selection requires a compact logic

then show *?case* **using** *upper-remainder[of $\langle K \div \varphi \rangle K \varphi$] contract-success[OF 1(1)]*

by (*metis contract-closure contract-inclusion infer-def taut2emptyrem valid-def*)

next

case (2 *K A* $\varphi \psi$)

then show *?case*

by (*metis (mono-tags, lifting) contract-extensionality Collect-cong remainder-extensionality*)

qed

contraction-is-pmc : an AGM contraction is equivalent to a partial met contraction using the selection function γ_{AGM}

lemma *contraction-is-pmc*: $\langle K = Cn(A) \Longrightarrow K \div \varphi = K \div \gamma_{AGM} \varphi \rangle$ — requires a supraclassical logic

proof

assume *a*: $\langle K = Cn(A) \rangle$

show $\langle K \div \varphi \subseteq K \div \gamma_{AGM} \varphi \rangle$

using *contract-inclusion[OF a]* **by** (*auto simp add:mc AGM-sel*)

next

assume *a*: $\langle K = Cn(A) \rangle$

show $\langle K \div \gamma_{AGM} \varphi \subseteq K \div \varphi \rangle$

proof (*cases $\langle \vdash \varphi \rangle$*)

case *True*

hence $\langle K \perp. \varphi = \{\} \rangle$

using *nonconsequence-remainder taut2emptyrem* **by** *auto*

then show *?thesis*

apply(*simp-all add:mc AGM-sel*)

by (*metis a emptyrem2taut contract-closure contract-recovery valid-expansion*)

next

case *validFalse:False*

then show *?thesis*

proof (*cases $\langle K \vdash \varphi \rangle$*)

case *True*

hence *b*: $\langle K \perp. \varphi \neq \{\} \rangle$

```

    using emptyrem2taut validFalse by blast
    have d:⟨ψ ∈ K ⟹ φ ⟶. ψ ∈ K ÷ φ⟩ for ψ
    using Supraclassical-logic.impI-PL Supraclassical-logic-axioms a contract-closure
    contract-recovery expansion-def by fastforce
  { fix ψ
    assume e:⟨ψ ∈ K ⟩ and f:⟨ψ ∉ K ÷ φ⟩
    have ⟨(ψ ⟶. φ) ⟶. φ ∉ K ÷ φ⟩
      using imp-recovery2[of ⟨K ÷ φ⟩ φ ψ] a contract-closure d e f by auto
    hence g:⟨¬ (K ÷ φ) ∪ {ψ ⟶. φ} ⊢ φ⟩
      using a contract-closure impI-PL by fastforce
    then obtain B where h:⟨(K ÷ φ) ∪ {ψ ⟶. φ} ⊆ B⟩ and i:⟨B ∈ K .⊥.
    φ⟩
      using upper-remainder[of ⟨(K ÷ φ) ∪ {ψ ⟶. φ}⟩ K φ] a True
    contract-inclusion idempotency-L impI2 by auto
    hence j:⟨ψ ∉ Cn(B)⟩
      by (metis (no-types, lifting) CollectD mp-PL Un-insert-right a infer-def
    insert-subset rem rem-closure)
    have ⟨ψ ∉ K ÷ γAGM φ⟩
      apply(simp add:mc AGM-sel b, rule-tac x=B in exI)
    by (meson Tarskian-logic.assumption-L Tarskian-logic-axioms h i j le-sup-iff)
  }
  then show ?thesis
    using a pmc-inclusion by fastforce
next
case False
hence ⟨K .⊥. φ = {K}⟩
  using nonconsequence-remainder taut2emptyrem by auto
then show ?thesis
  using False a contract-vacuity idempotency-L pmc-vacuity by auto
qed
qed
qed

```

lemma *contraction-with-completion*: $\langle K = Cn(A) \implies K \div \varphi = K \div * \gamma_{AGM} \varphi \rangle$
 by (simp add: contraction-is-pmc pmc-completion)

end

locale *TRMC-SC* = *TransitivelyRelationalMeetContraction* + *PMC-SC* **where**
selection = γ_{TR}

begin

A transitively relational selection function respect conjunctive overlap.

lemma *rel-sel-conj-overlap*: $\langle K = Cn(A) \implies \gamma_{TR} K (\varphi \wedge \psi) \subseteq \gamma_{TR} K \varphi \cup \gamma_{TR} K \psi \rangle$

proof(*intro subsetI*)

```

fix B
assume a:⟨K = Cn(A)⟩ and b:⟨B ∈ γTR K (φ .∧. ψ)⟩
show ⟨B ∈ γTR K φ ∪ γTR K ψ⟩ (is ?A)
proof(cases ⟨⊢ φ ∨ ⊢ ψ ∨ ¬ K ⊢ φ ∨ ¬ K ⊢ ψ⟩, elim disjE)
  assume ⟨⊢ φ⟩
  hence c:⟨Cn({φ .∧. ψ}) = Cn({ψ})⟩
  using conj-equiv valid-Cn-equiv valid-def by blast
  from b show ?A
  by (metis Un-iff a c extensional-selection)
next
  assume ⟨⊢ ψ⟩
  hence c:⟨Cn({φ .∧. ψ}) = Cn({φ})⟩
  by (simp add: Cn-conj-bis Cn-same validD-L)
  from b show ?A
  by (metis Un-iff a c extensional-selection)
next
  assume ⟨¬ K ⊢ φ⟩
  then show ?A
  by (metis UnI1 a b conjE1-PL is-selection nonconsequence-remainder nonempty-selection
    tautology-selection subset-singletonD)
next
  assume ⟨¬ K ⊢ ψ⟩
  then show ?A
  by (metis UnI2 a b conjE2-PL is-selection nonconsequence-remainder nonempty-selection
    tautology-selection subset-singletonD)
next
  assume d:⟨¬ (⊢ φ ∨ ⊢ ψ ∨ ¬ K ⊢ φ ∨ ¬ K ⊢ ψ)⟩
  hence h:⟨K .⊥. φ ≠ {}⟩ and i:⟨K .⊥. ψ ≠ {}⟩ and j:⟨K .⊥. (φ .∧. ψ) ≠ {}⟩
and k:⟨K ⊢ φ .∧. ψ⟩
  using d emptyrem2taut valid-conj-PL apply auto
  by (meson Supraclassical-logic.conjI-PL Supraclassical-logic-axioms d)
  show ?A
  using remainder-conj[OF a k] b h i j rel-sel by auto
qed
qed

```

A transitively relational meet contraction respect conjunctive overlap.

lemma *trmc-conj-overlap*: ⟨K = Cn(A) ⟹ (K ÷_{γ_{TR}} φ) ∩ (K ÷_{γ_{TR}} ψ) ⊆ (K ÷_{γ_{TR}} (φ .∧. ψ))⟩
unfolding mc **using** rel-sel-conj-overlap **by** blast

A transitively relational selection function respect conjunctive inclusion

lemma *rel-sel-conj-inclusion*: ⟨K = Cn(A) ⟹ γ_{TR} K (φ .∧. ψ) ∩ (K .⊥. φ) ≠ {} ⟹ γ_{TR} K φ ⊆ γ_{TR} K (φ .∧. ψ)⟩
proof(intro subsetI)
fix B
assume a:⟨K = Cn(A)⟩ **and** b:⟨γ_{TR} K (φ .∧. ψ) ∩ (K .⊥. φ) ≠ {}⟩ **and** c:⟨B ∈ γ_{TR} K φ⟩
show ⟨B ∈ γ_{TR} K (φ .∧. ψ)⟩ (**is** ?A)

```

proof(cases ⟨ $\Vdash \varphi \vee \Vdash \psi \vee \neg K \vdash \varphi \vee \neg K \vdash \psi$ ⟩, auto)
  assume ⟨ $\Vdash \varphi$ ⟩
  then show ?A
    using b taut2emptyrem by auto
next
  assume ⟨ $\Vdash \psi$ ⟩
  hence ⟨ $Cn(\{\varphi \wedge \psi\}) = Cn(\{\varphi\})$ ⟩
    by (simp add: Cn-conj-bis Cn-same validD-L)
  then show ?A
    using a c extensional-selection by blast
next
  assume d:⟨ $\varphi \notin Cn K$ ⟩
  with d show ?A
    by (metis Int-emptyI Tarskian-logic.nonconsequence-remainder Tarskian-logic-axioms
a b c idempotency-L
      inf-bot-right is-selection nonempty-selection singletonD subset-singletonD)
next
  assume d:⟨ $\psi \notin Cn K$ ⟩
  hence e:⟨ $(\varphi \wedge \psi) \notin Cn K$ ⟩
    by (meson Supraclassical-logic.conjE2-PL Supraclassical-logic-axioms)
  hence f:⟨ $\gamma_{TR} K (\varphi \wedge \psi) = \{K\}$ ⟩
    by (metis Tarskian-logic.nonconsequence-remainder Tarskian-logic-axioms a
insert-not-empty is-selection
      nonempty-selection subset-singletonD)
  with b have g:⟨ $(K \perp \varphi) = \{K\}$ ⟩
    unfolding nonconsequence-remainder[symmetric] using rem by auto
  with d f show ?A
    using a c is-selection by fastforce
next
  assume d:⟨ $\neg \Vdash \varphi$ ⟩ and e:⟨ $\neg \Vdash \psi$ ⟩ and f:⟨ $\varphi \in Cn K$ ⟩ and g:⟨ $\psi \in Cn K$ ⟩
  hence h:⟨ $K \perp \varphi \neq \{\}$ ⟩ and i:⟨ $K \perp \psi \neq \{\}$ ⟩ and j:⟨ $K \perp (\varphi \wedge \psi) \neq \{\}$ ⟩
and k:⟨ $K \vdash \varphi \wedge \psi$ ⟩
    using e d emptyrem2taut valid-conj-PL apply auto
    by (meson Supraclassical-logic.conjI-PL Supraclassical-logic-axioms f g)
  have o:⟨ $B \in K \perp \varphi \implies B \in K \perp (\varphi \wedge \psi)$ ⟩ for B
    using a k remainder-conj by auto
  from b obtain B' where l:⟨ $B' \in K \perp (\varphi \wedge \psi)$ ⟩ and m:⟨ $\forall C \in K \perp (\varphi \wedge \psi). C \preceq_K B'$ ⟩
and n:⟨ $\varphi \notin B'$ ⟩
    apply (auto simp add:mc rel-sel j)
    using assumption-L rem by force
  have p:⟨ $B' \in K \perp \varphi$ ⟩
    apply (simp add: rem)
    by (metis (no-types, lifting) Supraclassical-logic.conjE1-PL Supraclassi-
cal-logic-axioms
      Tarskian-logic.rem Tarskian-logic-axioms a l mem-Collect-eq n
rem-closure)
  from c show ?A
    apply (simp add:rel-sel o j h)
    using m p trans-rel a by blast

```

qed
qed

A transitively relational meet contraction respect conjunctive inclusion

lemma *trmc-conj-inclusion*: $\langle K = Cn(A) \implies \varphi \notin (K \dot{\div} \gamma_{TR} (\varphi \wedge \psi)) \implies ((K \dot{\div} \gamma_{TR} (\varphi \wedge \psi) \subseteq (K \dot{\div} \gamma_{TR} \varphi)) \rangle$

proof –

assume $a: \langle K = Cn(A) \rangle$ **and** $b: \langle \varphi \notin (K \dot{\div} \gamma_{TR} (\varphi \wedge \psi)) \rangle$

then obtain B **where** $c: \langle B \in \gamma_{TR} K (\varphi \wedge \psi) \rangle$ **and** $d: \langle \neg B \vdash \varphi \rangle$ **apply** (*simp add:mc*)

by (*metis b emptyrem2taut is-selection pmc-tautology rem-closure subset-iff validD-L valid-conj-PL*)

hence $\langle B \in (K \perp \varphi) \rangle$

using *remainder-recovery-bis*[*OF a - d, of $\langle \varphi \wedge \psi \rangle$*]

by (*metis (no-types, opaque-lifting) a conj-PL emptyrem2taut insert-not-empty is-selection*

nonconsequence-remainder subsetD taut2emptyrem)

with c **have** $e: \langle \gamma_{TR} K (\varphi \wedge \psi) \cap (K \perp \varphi) \neq \{\} \rangle$ **by** *blast*

then show $\langle ((K \dot{\div} \gamma_{TR} (\varphi \wedge \psi) \subseteq (K \dot{\div} \gamma_{TR} \varphi)) \rangle$

unfolding *mc* **using** *rel-sel-conj-inclusion*[*OF a e*] **by** *blast*

qed

As a transitively relational meet contraction has been proven to respect all postulates of AGM full contraction the equivalence between the both are straightforward

sublocale *AGM-FullContraction* **where** *contraction* = $\langle \lambda A \varphi. A \dot{\div} \gamma_{TR} \varphi \rangle$

using *trmc-conj-inclusion trmc-conj-overlap*

by (*unfold-locales, simp-all*)

end

locale *AGMFC-SC* = *AGM-FullContraction* + *AGMC-SC*

begin

An AGM relation is defined as ?

definition *AGM-relation*:: $\langle 'a \text{ set} \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set} \Rightarrow \text{bool} \rangle$

where *AGM-rel*: $\langle \text{AGM-relation } C K B \equiv (C = K \wedge B = K) \vee ((\exists \varphi. K \vdash \varphi \wedge C \in K \perp \varphi) \wedge (\exists \varphi. K \vdash \varphi \wedge B \in K \perp \varphi \wedge K \dot{\div} \varphi \subseteq B) \wedge (\forall \varphi. (K \vdash \varphi \wedge C \in K \perp \varphi \wedge B \in K \perp \varphi \wedge K \dot{\div} \varphi \subseteq C) \longrightarrow K \dot{\div} \varphi \subseteq B) \rangle$

An AGM relational selection is defined as a function that return K if the remainders of $K \perp \varphi$ is empty and the best element of the remainders according to an AGM relation

definition *AGM-relational-selection*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set set} \rangle$ ($\langle \gamma_{AGMTR} \rangle$)
where *AGM-rel-sel*: $\langle \gamma_{AGMTR} K \varphi \equiv \text{if } (K \perp. \varphi) = \{\} \text{ then } \{K\} \text{ else } \{B. B \in (K \perp. \varphi) \wedge (\forall C \in (K \perp. \varphi). \text{AGM-relation } C K B)\} \rangle$

lemma *AGM-rel-sel-completion*: $\langle K = Cn(A) \Longrightarrow \gamma_{AGMTR} K \varphi = * \gamma_{AGM} K \varphi \rangle$
apply (*unfold AGM-rel-sel, simp add:completion-def split: if-splits*)
proof(*auto simp add:AGM-sel*)
fix $S B C$
assume $a: \langle S \in Cn(A) \perp. \varphi \rangle$ **and** $b: \langle B \in Cn(A) \perp. \varphi \rangle$ **and** $c: \langle \bigcap \{B \in Cn(A) \perp. \varphi. Cn(A) \div \varphi \subseteq B\} \subseteq B \rangle$
and $d: \langle C \in Cn(A) \perp. \varphi \rangle$
hence $e: \langle \varphi \notin Cn(A) \div \varphi \rangle$
using *Tarskian-logic.taut2emptyrem Tarskian-logic-axioms contract-success* **by** *fastforce*
show $\langle \text{AGM-relation } C (Cn(A)) B \rangle$
proof(*cases* $\langle \varphi \in Cn(A) \rangle$)
case *True*
{ fix ψ
assume $\langle Cn A \div \psi \subseteq C \rangle$
hence $\langle Cn A \div (\varphi \wedge. \psi) \subseteq Cn A \div \varphi \rangle$
using *contract-conj-inclusion-variant*[*of* $\langle Cn(A) \rangle A \varphi \psi$]
by (*metis (mono-tags, lifting) assumption-L contract-conj-inclusion d mem-Collect-eq rem subset-iff*)
} **note** $f = \text{this}$
{ fix $\psi \varphi'$
assume $g: \langle \psi \in Cn A \div \varphi' \rangle$ **and** $h: \langle B \in Cn A \perp. \varphi' \rangle$ **and** $j: \langle Cn A \div \varphi' \subseteq C \rangle$ **and** $i: \langle \psi \notin B \rangle$
hence $\langle \varphi' \vee. \psi \in Cn A \div \varphi' \rangle$
using *Supraclassical-logic.disjI2-PL Supraclassical-logic-axioms contract-closure*
by *fastforce*
hence $k: \langle \varphi' \vee. \psi \in Cn A \div \varphi \rangle$
using *contract-conj-overlap-variant*[*of* $\langle Cn(A) \rangle A \varphi' \varphi$] f [*OF* j]
by (*metis IntI Supraclassical-logic.disjI1-PL Supraclassical-logic-axioms conj-com-Cn contract-extensionality inclusion-L singletonI subsetD*)
hence $l: \langle Cn A \div \varphi \subseteq B \rangle$ **using** c **by** *auto*
from $k l$ **have** $m: \langle \varphi' \vee. \psi \in B \rangle$ **and** $n: \langle B = Cn(B) \rangle$
using b *rem-closure* **by** *blast+*
have $\langle B \cup \{\psi\} \vdash \varphi' \rangle$ **using** $g h i$
by (*simp add:rem*) (*metis contract-inclusion insertI1 insert-subsetI psubsetI subsetD subset-insertI*)
with $n m$ **have** $\langle B \vdash \varphi' \rangle$
by (*metis Cn-equiv assumption-L disjE-PL disj-com equiv-PL imp-PL*)
with h **have** *False*
using *assumption-L rem* **by** *auto*
} **note** $g = \text{this}$


```

with True show ?thesis
  apply(unfold AGM-rel, rule-tac disjI2)
  using d b c by (auto simp add:AGM-rel idempotency-L del:subsetI) blast+
next
  case False
  then show ?thesis
    by (metis AGM-rel b d idempotency-L infer-def nonconsequence-remainder
singletonD)
    qed
next
  fix S B ψ
  assume a:⟨S ∈ Cn(A) .⊥. φ⟩ and b:⟨B ∈ Cn(A) .⊥. φ⟩ and c:⟨∀ C ∈ Cn A .⊥.
φ. AGM-relation C (Cn A) B⟩
  and d:⟨∀ C'. C' ∈ Cn A .⊥. φ ∧ Cn A ÷ φ ⊆ C' ⟶ ψ ∈ C'⟩
  then show ⟨ψ ∈ B⟩
  unfolding AGM-rel
  by (metis (no-types, lifting) AGM-sel empty-Collect-eq insert-Diff insert-not-empty
nonconsequence-remainder nonempty-selection singletonD)
qed

```

A transitively relational selection and an AGM relation is a transitively relational meet contraction

```

sublocale TransitivelyRelationalMeetContraction where relation = AGM-relation
and rel-sel = ⟨γAGMTR⟩
proof(unfold-locales, simp-all (no-asm) only:atomize-eq, goal-cases)
  case a:(1 K A C B' B) — Very difficult proof requires litterature and high
automation of isabelle!
  from a(2,3) show ?case
    unfolding AGM-rel apply(elim disjE conjE, simp-all)
    proof(intro disjI2 allI impI, elim exE conjE, goal-cases)
      case (1 ψ - - φ)
      have b:⟨B ∈ K .⊥. (φ .∧. ψ)⟩ and c:⟨B' ∈ K .⊥. (φ .∧. ψ)⟩ and d:⟨C ∈ K
.⊥. (φ .∧. ψ)⟩
      using remainder-conj[OF a(1)] 1 conjI-PL by auto
      hence e:⟨K ÷ (φ .∧. ψ) ⊆ B⟩
      using contract-conj-inclusion-variant[OF a(1), of φ ψ]
      by (meson 1(1) 1(12) 1(16) 1(2) 1(3) 1(8) Supraclassical-logic.conj-PL
Supraclassical-logic-axioms dual-order.trans)
      { fix χ
        assume f:⟨χ ∈ K ÷ ψ⟩
        have ⟨ψ .∨. χ ∈ (K ÷ ψ) ∩ Cn {ψ}⟩
        by (metis Int-iff Supraclassical-logic.disjI1-PL Supraclassical-logic.disjI2-PL
Supraclassical-logic-axioms
          f a(1) contract-closure in-mono inclusion-L singletonI)
        hence g:⟨ψ .∨. χ ∈ B⟩
        using contract-conj-overlap-variant[OF a(1), of ψ]
        by (metis AGM-Contraction.contract-extensionality AGM-Contraction-axioms
a(1) conj-com-Cn e in-mono)
      }

```

```

    have ⟨ $\psi \longrightarrow \chi \in B$ ⟩
    by (metis a(1) 1(10) 1(15) 1(16) assumption-L f in-mono infer-def rem-closure
rem-inclusion remainder-recovery)
    with g have ⟨ $\chi \in B$ ⟩
    by (metis 1(15) a(1) disjE-PL infer-def order-refl rem-closure validD-L
valid-Cn-imp)
  }
  then show ?case by blast
qed
next
case (2 K A  $\varphi$ )
hence ⟨ $* \gamma_{AGM} K \varphi \neq \{\}$ ⟩
using nonempty-selection[OF 2(1), of  $\varphi$ ] selection-completion[OF 2(1), of  $\varphi$ ]
by blast
then show ?case
using AGM-rel-sel-completion[OF 2(1), of  $\varphi$ ] AGM-rel-sel 2(1,2) by force
next
case (3 K  $\varphi$ )
then show ?case using AGM-rel-sel-completion AGM-rel-sel by simp
qed

```

— ça marche tout seul! ==> Je ne vois pas où sont utilisés ces lemmas

lemmas *fullcontraction-is-pmc = contraction-is-pmc*

lemmas *fullcontraction-is-trmc = contraction-with-completion*

end

locale *FMC-SC = FullMeetContraction + TRMC-SC*

begin

lemma *full-meet-weak1*: ⟨ $K = Cn(A) \implies K \vdash \varphi \implies (K \div \gamma_{FC} \varphi) = K \cap Cn(\{\neg \varphi\})$ ⟩

proof(intro subset-antisym Int-greatest)

assume a:⟨ $K = Cn(A)$ ⟩ and b:⟨ $K \vdash \varphi$ ⟩

then show ⟨ $(K \div \gamma_{FC} \varphi) \subseteq K$ ⟩

by (simp add: Inf-less-eq full-sel mc rem-inclusion)

next

assume a:⟨ $K = Cn(A)$ ⟩ and b:⟨ $K \vdash \varphi$ ⟩

show ⟨ $(K \div \gamma_{FC} \varphi) \subseteq Cn(\{\neg \varphi\})$ ⟩

proof

fix ψ

assume c:⟨ $\psi \in (K \div \gamma_{FC} \varphi)$ ⟩

{ assume ⟨ $\neg \{\neg \varphi\} \vdash \psi$ ⟩

hence ⟨ $\neg \{\neg \psi\} \vdash \varphi$ ⟩

by (metis Un-insert-right insert-is-Un not-PL notnot-PL)

hence ⟨ $\neg \{\varphi \vee \neg \psi\} \vdash \varphi$ ⟩

by (metis assumption-L disjI2-PL singleton-iff transitivity2-L)

then obtain B **where** $d:\langle\{\varphi .\vee. \neg \psi\} \subseteq B\rangle$ **and** $e:\langle B \in K .\perp. \varphi\rangle$
by (*metis a b disjI1-PL empty-subsetI idempotency-L infer-def insert-subset upper-remainder*)
hence $f:\langle\neg \psi \in B\rangle$
by (*metis (no-types, lifting) CollectD assumption-L insert-subset disj-notE-PL rem*)
hence $\langle\neg \psi \in (K \dot{\div} \gamma_{FC} \varphi)\rangle$
using e *mc full-sel* **by** *auto*
}
then show $\langle\psi \in Cn(\{\neg \varphi\})\rangle$
using c *infer-def* **by** *blast*
qed
next
assume $a:\langle K = Cn(A)\rangle$ **and** $b:\langle K \vdash \varphi\rangle$
show $\langle K \cap Cn(\{\neg \varphi\}) \subseteq (K \dot{\div} \gamma_{FC} \varphi)\rangle$
proof(*safe*)
fix ψ
assume $c:\langle\psi \in K\rangle$ **and** $d:\langle\psi \in Cn \{\neg \varphi\}\rangle$
have $e:\langle B \vdash \neg \varphi \longrightarrow. \psi\rangle$ **for** B
by (*simp add: d validD-L valid-imp-PL*)
{ fix B
assume $f:\langle B \in K .\perp. \varphi\rangle$
hence $\langle B \vdash \varphi \longrightarrow. \psi\rangle$
using a *assumption-L c remainder-recovery* **by** *auto*
then have $f:\langle B \vdash \psi\rangle$ **using** $d e$
using *disjE-PL ex-mid-PL* **by** *blast*
}
then show $\langle\psi \in (K \dot{\div} \gamma_{FC} \varphi)\rangle$
apply(*simp-all add:mc c full-sel*)
using a *rem-closure* **by** *blast*
qed
qed
lemma *full-meet-weak2*: $\langle K = Cn(A) \implies K \vdash \varphi \implies Cn((K \dot{\div} \gamma_{FC} \varphi) \cup \{\neg \varphi\}) = Cn(\{\neg \varphi\})\rangle$
unfolding *full-meet-weak1*
by (*metis Cn-union idempotency-L inf.cobounded2 sup.absorb-iff2 sup-commute*)
end
end

5 Revisions

The third operator of belief change introduced by the AGM framework is the revision. In revision a sentence φ is added to the belief set K in such a way that other sentences of K are removed if needed so that K is consistent

5.1 AGM revision postulates

The revision operator is denoted by the symbol $*$ and respect the following conditions :

- *revis-closure* : a belief set K revised by φ should be logically closed
- *revis-inclusion* : a belief set K revised by φ should be a subset of K expanded by φ
- *revis-vacuity* : if $\neg\varphi$ is not in K then the revision of K by φ is equivalent of the expansion of K by φ
- *revis-success* : a belief set K revised by φ should contain φ
- *revis-extensionality* : Extensionality guarantees that the logic of contraction is extensional in the sense of allowing logically equivalent sentences to be freely substituted for each other
- *revis-consistency* : a belief set K revised by φ is consistent if φ is consistent

locale *AGM-Revision* = *Supraclassical-logic* +

fixes *revision*:: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set} \rangle$ (**infix** $\langle * \rangle$ 55)

assumes *revis-closure*: $\langle K = Cn(A) \Longrightarrow K * \varphi = Cn(K * \varphi) \rangle$
and *revis-inclusion*: $\langle K = Cn(A) \Longrightarrow K * \varphi \subseteq K \oplus \varphi \rangle$
and *revis-vacuity*: $\langle K = Cn(A) \Longrightarrow \neg\varphi \notin K \Longrightarrow K \oplus \varphi \subseteq K * \varphi \rangle$
and *revis-success*: $\langle K = Cn(A) \Longrightarrow \varphi \in K * \varphi \rangle$
and *revis-extensionality*: $\langle K = Cn(A) \Longrightarrow Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow K * \varphi = K * \psi \rangle$
and *revis-consistency*: $\langle K = Cn(A) \Longrightarrow \neg\varphi \notin Cn(\{\}) \Longrightarrow \perp \notin K * \varphi \rangle$

A full revision is defined by two more postulates :

- *revis-superexpansion* : An element of $K * (\varphi \wedge \psi)$ is also an element of K revised by φ and expanded by ψ
- *revis-subexpansion* : An element of $(K * \varphi) \oplus \psi$ is also an element of K revised by $\varphi \wedge \psi$ if $(K * \varphi)$ do not imply $\neg\psi$

locale *AGM-FullRevision* = *AGM-Revision* +

assumes *revis-superexpansion*: $\langle K = Cn(A) \Longrightarrow K * (\varphi \wedge \psi) \subseteq (K * \varphi) \oplus \psi \rangle$
and *revis-subexpansion*: $\langle K = Cn(A) \Longrightarrow \neg\psi \notin (K * \varphi) \Longrightarrow (K * \varphi) \oplus \psi \subseteq K * (\varphi \wedge \psi) \rangle$

begin

— important lemmas/corollaries that can replace the previous assumptions

corollary *revis-superexpansion-ext* : $\langle K = Cn(A) \implies (K * \varphi) \cap (K * \psi) \subseteq (K * (\varphi \vee \psi)) \rangle$

proof(*intro subsetI, elim IntE*)

fix χ

assume $a:\langle K = Cn(A) \rangle$ **and** $b:\langle \chi \in (K * \varphi) \rangle$ **and** $c:\langle \chi \in (K * \psi) \rangle$

have $\langle Cn(\{(\varphi' \vee \psi') \wedge \varphi'\}) = Cn(\{\varphi'\}) \rangle$ **for** $\varphi' \psi'$

using *conj-superexpansion2* **by** (*simp add: Cn-same*)

hence $d:\langle K * \varphi' \subseteq (K * (\varphi' \vee \psi')) \oplus \varphi' \rangle$ **for** $\varphi' \psi'$

using *revis-superexpansion*[*OF a, of $\langle \varphi' \vee \psi' \rangle \varphi'$*] *revis-extensionality a* **by** *metis*

hence $\langle \varphi \longrightarrow \chi \in (K * (\varphi \vee \psi)) \rangle$ **and** $\langle \psi \longrightarrow \chi \in (K * (\varphi \vee \psi)) \rangle$

using *d*[*of $\varphi \psi$*] *d*[*of $\psi \varphi$*] *revis-extensionality*[*OF a disj-com-Cn, of $\psi \varphi$*]

using *imp-PL a b c expansion-def revis-closure* **by** *fastforce+*

then show $c:\langle \chi \in (K * (\varphi \vee \psi)) \rangle$

using *disjE-PL a revis-closure revis-success* **by** *fastforce*

qed

end

5.2 Relation of AGM revision and AGM contraction

The AGM contraction of K by φ can be defined as the AGM revision of K by $\neg\varphi$ intersect with K (to remove $\neg\varphi$ from K revised). This definition is known as Harper identity [3]

sublocale *AGM-Revision* \subseteq *AGM-Contraction* **where** *contraction* = $\langle \lambda K \varphi. K \cap (K * \neg\varphi) \rangle$

proof(*unfold-locales, goal-cases*)

case *closure*:(1 $K A \varphi$)

then show *?case*

by (*metis Cn-inter revis-closure*)

next

case *inclusion*:(2 $K A \varphi$)

then show *?case* **by** *blast*

next

case *vacuity*:(3 $K A \varphi$)

hence $\langle \neg (\neg\varphi) \notin K \rangle$

using *absurd-PL infer-def* **by** *blast*

hence $\langle K \subseteq (K * \neg\varphi) \rangle$

using *revis-vacuity*[**where** $\varphi = \langle \neg\varphi \rangle$] *expansion-def inclusion-L vacuity(1)* **by** *fastforce*

then show *?case*

by *fast*

next

case *success*:(4 $K A \varphi$)

hence $\langle \neg (\neg\varphi) \notin Cn(\{\}) \rangle$

using *infer-def notnot-PL* **by** *blast*

hence $a:\langle \perp \notin K * (\neg\varphi) \rangle$

```

    by (simp add: revis-consistency success(1))
  have  $\langle \neg \varphi \in K * (\neg \varphi) \rangle$ 
    by (simp add: revis-success success(1))
  with a have  $\langle \varphi \notin K * (\neg \varphi) \rangle$ 
    using infer-def non-consistency revis-closure success(1) by blast
  then show ?case
    by simp
next
case recovery:(5 K A  $\varphi$ )
show ?case
proof
  fix  $\psi$ 
  assume a: $\langle \psi \in K \rangle$ 
  hence b: $\langle \varphi \longrightarrow \psi \in K \rangle$  using impI2 recovery by auto
  have  $\langle \neg \psi \longrightarrow \neg \varphi \in K * \neg \varphi \rangle$ 
    using impI2 recovery revis-closure revis-success by fastforce
  hence  $\langle \varphi \longrightarrow \psi \in K * \neg \varphi \rangle$ 
    using imp-contrapos recovery revis-closure by fastforce
  with b show  $\langle \psi \in Cn(K \cap (K * \neg \varphi) \cup \{\varphi\}) \rangle$ 
    by (meson Int-iff Supraclassical-logic.imp-PL Supraclassical-logic-axioms inclusion-L subsetD)
qed
next
case extensionality:(6 K A  $\varphi \psi$ )
hence  $\langle Cn(\{\neg \varphi\}) = Cn(\{\neg \psi\}) \rangle$ 
  using equiv-negation[of  $\langle \{\} \rangle \varphi \psi$ ] valid-Cn-equiv valid-def by auto
hence  $\langle (K * \neg \varphi) = (K * \neg \psi) \rangle$ 
  using extensionality(1) revis-extensionality by blast
then show ?case by simp
qed

```

locale *AGMC-S = AGM-Contraction + Supraclassical-logic*

The AGM revision of K by φ can be defined as the AGM contraction of K by $\neg\varphi$ followed by an expansion by φ . This definition is known as Levi identity [4].

sublocale *AGMC-S \subseteq AGM-Revision* **where** *revision = $\langle \lambda K \varphi. (K \div \neg \varphi) \oplus \varphi \rangle$*

```

proof(unfold-locales, goal-cases)
  case closure:(1 K A  $\varphi$ )
  then show ?case
    by (simp add: expansion-def idempotency-L)
next
case inclusion:(2 K A  $\varphi$ )
have  $K \div \neg \varphi \subseteq K \cup \{\varphi\}$ 
  using contract-inclusion inclusion by auto
then show ?case
  by (simp add: expansion-def monotonicity-L)

```

```

next
  case vacuity:(3 K A  $\varphi$ )
  then show ?case
    by (simp add: contract-vacuity expansion-def)
next
  case success:(4 K A  $\varphi$ )
  then show ?case
    using assumption-L expansion-def by auto
next
  case extensionality:(5 K A  $\varphi \psi$ )
  hence  $\langle Cn(\{\neg \varphi\}) = Cn(\{\neg \psi\}) \rangle$ 
    using equiv-negation[of  $\langle \{\} \rangle \varphi \psi$ ] valid-Cn-equiv valid-def by auto
  hence  $\langle (K \div \neg \varphi) = (K \div \neg \psi) \rangle$ 
    using contract-extensionality extensionality(1) by blast
  then show ?case
    by (metis Cn-union expansion-def extensionality(2))
next
  case consistency:(6 K A  $\varphi$ )
  then show ?case
    by (metis contract-closure contract-success expansion-def infer-def not-PL)
qed

```

The relationship between AGM full revision and AGM full contraction is the same as the relation between AGM revision and AGM contraction

sublocale *AGM-FullRevision* \subseteq *AGM-FullContraction* **where** *contraction* = $\langle \lambda K \varphi. K \cap (K * \neg \varphi) \rangle$

```

proof(unfold-locales, goal-cases)
  case conj-overlap:(1 K A  $\varphi \psi$ )
  have  $a:\langle Cn(\{\neg(\varphi \wedge \psi)\}) = Cn(\{(\neg \varphi) \vee (\neg \psi)\}) \rangle$ 
    using Cn-same morgan by simp
  show ?case (is ?A)
    using revis-supereexpansion-ext[OF conj-overlap(1), of  $\langle \neg \varphi \rangle \langle \neg \psi \rangle$ ]
    revis-extensionality[OF conj-overlap(1) a] by auto
next
  case conj-inclusion:(2 K A  $\varphi \psi$ )
  have  $a:\langle Cn(\{\neg(\varphi \wedge \psi) \wedge \neg \varphi\}) = Cn(\{\neg \varphi\}) \rangle$ 
    using conj-supereexpansion1 by (simp add: Cn-same)
  from conj-inclusion show ?case
  proof(cases  $\langle \varphi \in K \rangle$ )
    case True
    hence  $b:\langle \neg(\neg \varphi) \notin K * \neg(\varphi \wedge \psi) \rangle$ 
      using absurd-PL conj-inclusion revis-closure by fastforce
    show ?thesis
      using revis-subexpansion[OF conj-inclusion(1) b] revis-extensionality[OF
    conj-inclusion(1) a]
    expansion-def inclusion-L by fastforce
  next
  case False
  then show ?thesis

```

```

    by (simp add: conj-inclusion(1) contract-vacuity)
  qed
qed

locale AGMFC-S = AGM-FullContraction + AGMC-S

sublocale AGMFC-S  $\subseteq$  AGM-FullRevision where revision =  $\langle \lambda K \varphi. (K \div \neg \varphi) \oplus \varphi \rangle$ 
proof (unfold-locales, safe, goal-cases)
  case super: (1 K A  $\varphi \psi \chi$ )
  hence a:  $\langle \varphi \wedge \psi \rangle \longrightarrow \chi \in Cn(Cn(A) \div \neg (\varphi \wedge \psi))$ 
  using Supraclassical-logic.imp-PL Supraclassical-logic-axioms expansion-def by
  fastforce
  have b:  $\langle \varphi \wedge \psi \rangle \longrightarrow \chi \in Cn(\{\neg (\varphi \wedge \psi)\})$ 
  by (meson Supraclassical-logic.imp-recovery0 Supraclassical-logic.valid-disj-PL
  Supraclassical-logic-axioms)
  have c:  $\langle \varphi \wedge \psi \rangle \longrightarrow \chi \in Cn(A) \div (\neg (\varphi \wedge \psi) \wedge \neg \varphi)$ 
  using contract-conj-overlap-variant[of  $\langle Cn(A) \rangle A \langle \neg (\varphi \wedge \psi) \rangle \langle \neg \varphi \rangle$ ] a b
  using AGM-Contraction.contract-closure AGM-FullContraction-axioms AGM-FullContraction-def
  by fastforce
  have d:  $\langle Cn(\{\neg (\varphi \wedge \psi) \wedge \neg \varphi\}) = Cn(\{\neg \varphi\}) \rangle$ 
  using conj-superexpansion1 by (simp add: Cn-same)
  hence e:  $\langle \varphi \wedge \psi \rangle \longrightarrow \chi \in Cn(A) \div \neg \varphi$ 
  using AGM-Contraction.contract-extensionality[OF - - d] c
  AGM-FullContraction-axioms AGM-FullContraction-def by fastforce
  hence f:  $\langle \varphi \longrightarrow (\psi \longrightarrow \chi) \rangle \in Cn(A) \div \neg \varphi$ 
  using conj-imp AGM-Contraction.contract-closure AGM-FullContraction-axioms
  AGM-FullContraction-def conj-imp by fastforce
  then show ?case
  by (metis assumption-L expansion-def imp-PL infer-def)
next
  case sub: (2 K A  $\psi \varphi \chi$ )
  hence a:  $\langle \varphi \longrightarrow (\psi \longrightarrow \chi) \rangle \in Cn(A) \div \neg \varphi$ 
  by (metis AGMC-S.axioms(1) AGMC-S-axioms AGM-Contraction.contract-closure
  expansion-def impI-PL infer-def revis-closure)
  have b:  $\langle Cn(\{\neg (\varphi \wedge \psi) \wedge \neg \varphi\}) = Cn(\{\neg \varphi\}) \rangle$ 
  using conj-superexpansion1 by (simp add: Cn-same)
  have c:  $\langle \neg (\varphi \wedge \psi) \notin Cn A \div (\neg \varphi) \rangle$ 
  using sub(1) by (metis assumption-L conj-imp expansion-def imp-PL infer-def
  not-PL)
  have c':  $\langle Cn(A) \div \neg \varphi \subseteq Cn(A) \div (\neg (\varphi \wedge \psi)) \rangle$ 
  using contract-conj-inclusion[of  $\langle Cn(A) \rangle A \langle \neg (\varphi \wedge \psi) \rangle \langle \neg \varphi \rangle$ ]
  by (metis AGM-Contraction.contract-extensionality AGM-FullContraction.axioms(1)
  AGM-FullContraction-axioms b c)
  then show ?case
  by (metis a assumption-L conj-imp expansion-def imp-PL in-mono infer-def)
qed

```


end

References

- [1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The journal of symbolic logic*, 50(2):510–530, 1985.
- [2] E. Fermé and S. O. Hansson. Agm 25 years: Twenty-five years of research in belief change. *Journal of Philosophical Logic*, 40:295–331, 04 2011.
- [3] W. L. Harper. Rational conceptual change. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 1976:462–494, 1976.
- [4] I. Levi. Subjunctives, dispositions and chances. *Synthese*, 34:303–335, 1977.
- [5] R. Wójcicki. *Theory of logical calculi: basic theory of consequence operations*, volume 199. Springer Science & Business Media, 2013.