Belief Revision Theory

Valentin Fouillard
Safouan TAHA
Frederic Boulanger
Nicolas Sabouret

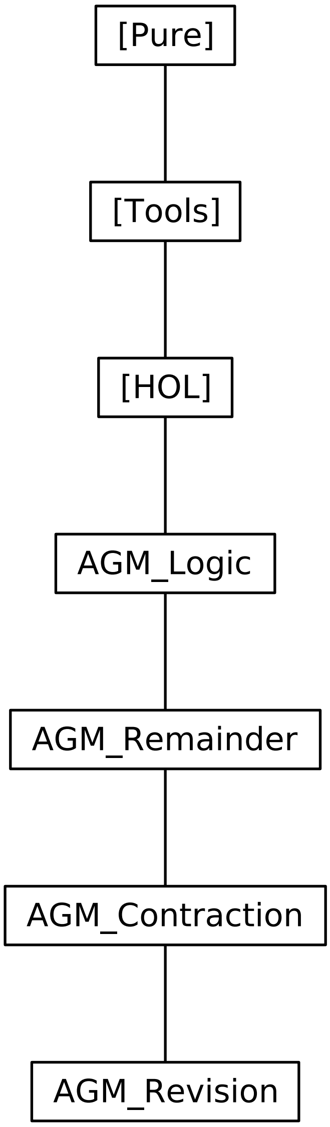
December 14, 2021

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


[Pure]
[Tools]
[HOL]
AGM_Logic
AGM_Remainder
AGM_Contraction
AGM_Revision

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 $Cn{::}\langle 'a\ set \Rightarrow 'a\ set\rangle$
assumes *monotonicity-L*: $\langle A \subseteq B \Longrightarrow Cn(A) \subseteq Cn(B)\rangle$
  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 $infer{::}\langle 'a\ set \Rightarrow 'a \Rightarrow bool\rangle$ (**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 $valid{::}\langle 'a \Rightarrow bool\rangle$ $(\langle\Vdash\rangle)$
defines $\langle \Vdash \varphi \equiv \{\} \vdash \varphi\rangle$

— $A \oplus \varphi$ is all that can be infered from A and φ.
fixes $expansion{::}\langle 'a\ set \Rightarrow 'a \Rightarrow 'a\ set\rangle$ (**infix** $\langle\oplus\rangle$ *57*)
defines $\langle A \oplus \varphi \equiv Cn(A \cup \{\varphi\})\rangle$

begin

lemma *idempotency-L*: $\langle Cn(Cn(A)) = Cn(A)\rangle$
  ⟨*proof*⟩

lemma *assumption-L*: $\langle \varphi \in A \Longrightarrow A \vdash \varphi\rangle$
  ⟨*proof*⟩

lemma *validD-L*: ‹$\Vdash \varphi \Longrightarrow \varphi \in Cn(A)$›
  ⟨*proof*⟩

lemma *valid-expansion*: ‹$K = Cn(A) \Longrightarrow \Vdash \varphi \Longrightarrow K \oplus \varphi = K$›
  ⟨*proof*⟩

lemma *transitivity2-L*:
  assumes ‹$\forall \varphi \in B.\ A \vdash \varphi$›
    and ‹$B \vdash \psi$›
  shows ‹$A \vdash \psi$›
⟨*proof*⟩

lemma *Cn-same*: ‹$(Cn(A) = Cn(B)) \longleftrightarrow (\forall C.\ A \subseteq Cn(C) \longleftrightarrow B \subseteq Cn(C))$›
⟨*proof*⟩
lemma *Cn-union*: ‹$Cn(Cn(A) \cup Cn(B)) = Cn(A \cup B)$›
⟨*proof*⟩
lemma *Cn-Union*: ‹$Cn(\bigcup\{Cn(B)|B.\ P\ B\}) = Cn(\bigcup\{B.\ P\ B\})$› (**is** ‹$?A = ?B$›)
⟨*proof*⟩
lemma *Cn-inter*: ‹$K = Cn(A) \cap Cn(B) \Longrightarrow K = Cn(K)$›
⟨*proof*⟩
lemma *Cn-Inter*: ‹$K = \bigcap\{Cn(B)|B.\ P\ B\} \Longrightarrow K = Cn(K)$›
⟨*proof*⟩

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. $.\wedge.$ 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*:: ‹$'a$› (‹$\bot$›)
  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*: ‹$\{\bot\} \vdash p$›

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*: ‹$A \vdash p \,.\!\longrightarrow\!.\; \bot \Longrightarrow A \vdash .\neg\; p$›
and *notE-PL*: ‹$A \vdash .\neg\; p \Longrightarrow A \vdash (p \,.\!\longrightarrow\!.\; \bot)$›

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 ‹$A \vdash q$›
⟨*proof*⟩
lemma *imp-PL*: ‹$A \vdash p \,.\!\longrightarrow\!.\; q \longleftrightarrow A \cup \{p\} \vdash q$›
⟨*proof*⟩

lemma *not-PL*: ‹$A \vdash .\neg\; p \longleftrightarrow A \cup \{p\} \vdash \bot$›
⟨*proof*⟩
lemma *notnot-PL*: ‹$A \vdash .\neg\; (.\neg\; p) \longleftrightarrow A \vdash p$›
⟨*proof*⟩

lemma *conj-PL*: ‹$A \vdash p \,.\!\wedge\!.\; q \longleftrightarrow (A \vdash p \wedge A \vdash q)$›
⟨*proof*⟩

lemma *disj-PL*: ‹$A \vdash p \,.\!\vee\!.\; q \longleftrightarrow A \cup \{.\neg\; p\} \vdash q$›
⟨*proof*⟩

lemma *equiv-PL*:‹$A \vdash p \mathrel{.\!\longleftrightarrow\!.} q \longleftrightarrow (A \cup \{p\} \vdash q \wedge A \cup \{q\} \vdash p)$›
⟨*proof*⟩

corollary *valid-imp-PL*: ‹$\Vdash (p \mathrel{.\!\longrightarrow\!.} q) = (\{p\} \vdash q)$›
and *valid-not-PL*: ‹$\Vdash (.\neg\, p) = (\{p\} \vdash \bot)$›
and *valid-conj-PL*: ‹$\Vdash (p \mathrel{.\!\wedge\!.} q) = (\Vdash p \wedge \Vdash q)$›
and *valid-disj-PL*: ‹$\Vdash (p \mathrel{.\!\vee\!.} q) = (\{.\neg\, p\} \vdash q)$›
and *valid-equiv-PL*:‹$\Vdash (p \mathrel{.\!\longleftrightarrow\!.} q) = (\{p\} \vdash q \wedge \{q\} \vdash p)$›
⟨*proof*⟩

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*: ‹$Cn(A) \neq \{\}$›
⟨*proof*⟩

lemma *Cn-true*: ‹$Cn(\{\top\}) = Cn(\{\})$›
⟨*proof*⟩

lemma *Cn-false*: ‹$Cn(\{\bot\}) = UNIV$›
⟨*proof*⟩

lemma *Cn-imp*: ‹$A \vdash (p \mathrel{.\!\longrightarrow\!.} q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(A \cup \{p\})$›
and *Cn-imp-bis*: ‹$A \vdash (p \mathrel{.\!\longrightarrow\!.} q) \longleftrightarrow Cn(A \cup \{q\}) \subseteq Cn(A \cup \{p\})$›
⟨*proof*⟩

lemma *Cn-not*: ‹$A \vdash .\neg\, p \longleftrightarrow Cn(A \cup \{p\}) = UNIV$›
⟨*proof*⟩

lemma *Cn-conj*: ‹$A \vdash (p \mathrel{.\!\wedge\!.} q) \longleftrightarrow Cn(\{p\}) \cup Cn(\{q\}) \subseteq Cn(A)$›
⟨*proof*⟩

lemma *Cn-conj-bis*: ‹$Cn(\{p \mathrel{.\!\wedge\!.} q\}) = Cn(\{p, q\})$›
⟨*proof*⟩

lemma *Cn-disj*: ‹$A \vdash (p \mathrel{.\!\vee\!.} q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(A \cup \{.\neg\, p\})$›
and *Cn-disj-bis*: ‹$A \vdash (p \mathrel{.\!\vee\!.} q) \longleftrightarrow Cn(A \cup \{q\}) \subseteq Cn(A \cup \{.\neg\, p\})$›
⟨*proof*⟩

lemma *Cn-equiv*: ‹$A \vdash (p \mathrel{.\!\longleftrightarrow\!.} q) \longleftrightarrow Cn(A \cup \{p\}) = Cn(A \cup \{q\})$›
⟨*proof*⟩

corollary *valid-nonemptyCn*: ‹$Cn(\{\}) \neq \{\}$›
and *valid-Cn-imp*: ‹$\Vdash (p \mathrel{.\!\longrightarrow\!.} q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(\{p\})$›
and *valid-Cn-not*: ‹$\Vdash (.\neg\, p) \longleftrightarrow Cn(\{p\}) = UNIV$›
and *valid-Cn-conj*: ‹$\Vdash (p \mathrel{.\!\wedge\!.} q) \longleftrightarrow Cn(\{p\}) \cup Cn(\{q\}) \subseteq Cn(\{\})$›

and *valid-Cn-disj*: ‹$\Vdash (p \mathbin{.\vee.} q) \longleftrightarrow Cn(\{q\}) \subseteq Cn(\{.\neg\ p\})$›
and *valid-Cn-equiv*: ‹$\Vdash (p \mathbin{.\longleftrightarrow.} q) \longleftrightarrow Cn(\{p\}) = Cn(\{q\})$›
⟨*proof*⟩

lemma *consistency*: ‹$Cn(\{p\}) \cap Cn(\{.\neg\ p\}) = Cn(\{\})$›
⟨*proof*⟩

lemma *Cn-notnot*: ‹$Cn(\{.\neg\ (.\neg\ \varphi)\}) = Cn(\{\varphi\})$›
⟨*proof*⟩

lemma *conj-com*: ‹$A \vdash p \mathbin{.\wedge.} q \longleftrightarrow A \vdash q \mathbin{.\wedge.} p$›
⟨*proof*⟩

lemma *conj-com-Cn*: ‹$Cn(\{p \mathbin{.\wedge.} q\}) = Cn(\{q \mathbin{.\wedge.} p\})$›
⟨*proof*⟩

lemma *disj-com*: ‹$A \vdash p \mathbin{.\vee.} q \longleftrightarrow A \vdash q \mathbin{.\vee.} p$›
⟨*proof*⟩

lemma *disj-com-Cn*: ‹$Cn(\{p \mathbin{.\vee.} q\}) = Cn(\{q \mathbin{.\vee.} p\})$›
⟨*proof*⟩

lemma *imp-contrapos*: ‹$A \vdash p \mathbin{.\longrightarrow.} q \longleftrightarrow A \vdash .\neg\ q \mathbin{.\longrightarrow.} .\neg\ p$›
⟨*proof*⟩

lemma *equiv-negation*: ‹$A \vdash p \mathbin{.\longleftrightarrow.} q \longleftrightarrow A \vdash .\neg\ p \mathbin{.\longleftrightarrow.} .\neg\ q$›
⟨*proof*⟩

lemma *imp-trans*: ‹$A \vdash p \mathbin{.\longrightarrow.} q \Longrightarrow A \vdash q \mathbin{.\longrightarrow.} r \Longrightarrow A \vdash p \mathbin{.\longrightarrow.} r$›
⟨*proof*⟩

lemma *imp-recovery0*: ‹$A \vdash p \mathbin{.\vee.} (p \mathbin{.\longrightarrow.} q)$›
⟨*proof*⟩

lemma *imp-recovery1*: ‹$A \cup \{p \mathbin{.\longrightarrow.} q\} \vdash p \Longrightarrow A \vdash p$›
⟨*proof*⟩

lemma *imp-recovery2*: ‹$A \vdash p \mathbin{.\longrightarrow.} q \Longrightarrow A \vdash (q \mathbin{.\longrightarrow.} p) \mathbin{.\longrightarrow.} p \Longrightarrow A \vdash q$›
⟨*proof*⟩

lemma *impI2*: ‹$A \vdash q \Longrightarrow A \vdash p \mathbin{.\longrightarrow.} q$›
⟨*proof*⟩

lemma *conj-equiv*: ‹$A \vdash p \Longrightarrow A \vdash ((p \mathbin{.\wedge.} q) \mathbin{.\longleftrightarrow.} q)$›
⟨*proof*⟩

lemma *conj-imp*: ‹$A \vdash (p \mathbin{.\wedge.} q) \mathbin{.\longrightarrow.} r \longleftrightarrow A \vdash p \mathbin{.\longrightarrow.} (q \mathbin{.\longrightarrow.} r)$›
⟨*proof*⟩

lemma *conj-not-impE-PL*: ‹$A \vdash (p \mathbin{.\wedge.} q) \mathbin{.\longrightarrow.} r \Longrightarrow A \vdash (p \mathbin{.\wedge.} .\neg\ q) \mathbin{.\longrightarrow.} r \Longrightarrow$

$A \vdash p\ .\!\longrightarrow\!.\ r$›
  ⟨*proof*⟩

lemma *disj-notE-PL*: ‹$A \vdash q \Longrightarrow A \vdash p\ .\!\vee\!.\ .\!\neg\ q \Longrightarrow A \vdash p$›
  ⟨*proof*⟩

lemma *disj-not-impE-PL*: ‹$A \vdash (p\ .\!\vee\!.\ q)\ .\!\longrightarrow\!.\ r \Longrightarrow A \vdash (p\ .\!\vee\!.\ .\!\neg\ q)\ .\!\longrightarrow\!.\ r \Longrightarrow A \vdash r$›
  ⟨*proof*⟩

lemma *imp-conj*: ‹$A \vdash p\ .\!\longrightarrow\!.\ q \Longrightarrow A \vdash r\ .\!\longrightarrow\!.\ s \Longrightarrow A \vdash (p\ .\!\wedge\!.\ r)\ .\!\longrightarrow\!.\ (q\ .\!\wedge\!.\ s)$›
  ⟨*proof*⟩

lemma *conj-overlap*: ‹$A \vdash (p\ .\!\wedge\!.\ q) \longleftrightarrow A \vdash (p\ .\!\wedge\!.\ ((.\!\neg\ p)\ .\!\vee\!.\ q))$›
  ⟨*proof*⟩

lemma *morgan*: ‹$A \vdash .\!\neg\ (p\ .\!\wedge\!.\ q) \longleftrightarrow A \vdash (.\!\neg\ p)\ .\!\vee\!.\ (.\!\neg\ q)$›
  ⟨*proof*⟩

lemma *conj-superexpansion1*: ‹$A \vdash .\!\neg\ (p\ .\!\wedge\!.\ q)\ .\!\wedge\!.\ .\!\neg\ p \longleftrightarrow A \vdash .\!\neg\ p$›
  ⟨*proof*⟩

lemma *conj-superexpansion2*: ‹$A \vdash (p\ .\!\vee\!.\ q)\ .\!\wedge\!.\ p \longleftrightarrow A \vdash p$›
  ⟨*proof*⟩

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*: ‹$A \vdash \varphi \Longrightarrow (\exists A'.\ A' \subseteq A \wedge \mathit{finite}\ A' \wedge A' \vdash \varphi)$›

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*: ‹$\neg \Vdash \varphi \Longrightarrow \mathit{subset.chain}\ \{B.\ \neg\ B \vdash \varphi\}\ C \Longrightarrow \neg \bigcup C \vdash \varphi$›
⟨*proof*⟩

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*::‹$'a\ set \Rightarrow 'a \Rightarrow 'a\ set\ set$› (**infix** ‹$.\bot.$› *55*)
  where *rem*: ‹$A\ .\bot.\ \varphi \equiv \{B.\ B \subseteq A \wedge \neg\ B \vdash \varphi \wedge (\forall B' \subseteq A.\ B \subset B' \longrightarrow B' \vdash \varphi)\}$›

lemma *rem-inclusion*: ‹$B \in A\ .\bot.\ \varphi \Longrightarrow B \subseteq A$›
  ⟨*proof*⟩

lemma *rem-closure*: $K = Cn(A) \Longrightarrow B \in K\ .\bot.\ \varphi \Longrightarrow B = Cn(B)$
  ⟨*proof*⟩

lemma *remainder-extensionality*: ‹$Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow A\ .\bot.\ \varphi = A\ .\bot.\ \psi$›
  ⟨*proof*⟩

lemma *nonconsequence-remainder*: ‹$A\ .\bot.\ \varphi = \{A\} \longleftrightarrow \neg\ A \vdash \varphi$›
  ⟨*proof*⟩
lemma *taut2emptyrem*: ‹$\Vdash \varphi \Longrightarrow A\ .\bot.\ \varphi = \{\}$›
  ⟨*proof*⟩

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*: ‹$K = Cn(A) \Longrightarrow K \vdash \psi \Longrightarrow B \in K\ .\bot.\ \varphi \Longrightarrow B \vdash \varphi\ .\longrightarrow.\ \psi$›
⟨*proof*⟩
lemma *remainder-recovery-bis*: ‹$K = Cn(A) \Longrightarrow K \vdash \psi \Longrightarrow \neg\ B \vdash \psi \Longrightarrow B \in K\ .\bot.\ \varphi \Longrightarrow B \in K\ .\bot.\ \psi$›

$\langle proof \rangle$

corollary *remainder-recovery-imp*: ‹$K = Cn(A) \Longrightarrow K \vdash \psi \Longrightarrow \Vdash (\psi \ .\!\longrightarrow\!. \ \varphi) \Longrightarrow B \in K \ .\bot. \ \varphi \Longrightarrow B \in K \ .\bot. \ \psi$›
$\langle proof \rangle$
lemma *remainder-expansion*: ‹$K = Cn(A) \Longrightarrow K \vdash \psi \Longrightarrow \neg\ B \vdash \psi \Longrightarrow B \in K \ .\bot. \ \varphi \Longrightarrow B \oplus \psi = K$›
$\langle proof \rangle$

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

lemma *remainder-conj*: ‹$K = Cn(A) \Longrightarrow K \vdash \varphi \ .\!\wedge\!. \ \psi \Longrightarrow K \ .\bot. \ (\varphi \ .\!\wedge\!. \ \psi) = (K \ .\bot. \ \varphi) \cup (K \ .\bot. \ \psi)$›
$\langle proof \rangle$

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*: ‹$B \subseteq A \Longrightarrow \neg\ B \vdash \varphi \Longrightarrow \exists B'. \ B \subseteq B' \wedge \ B' \in A \ .\bot. \ \varphi$›
$\langle proof \rangle$
corollary *emptyrem2taut*: ‹$A \ .\bot. \ \varphi = \{\} \Longrightarrow \Vdash \varphi$›
$\langle proof \rangle$

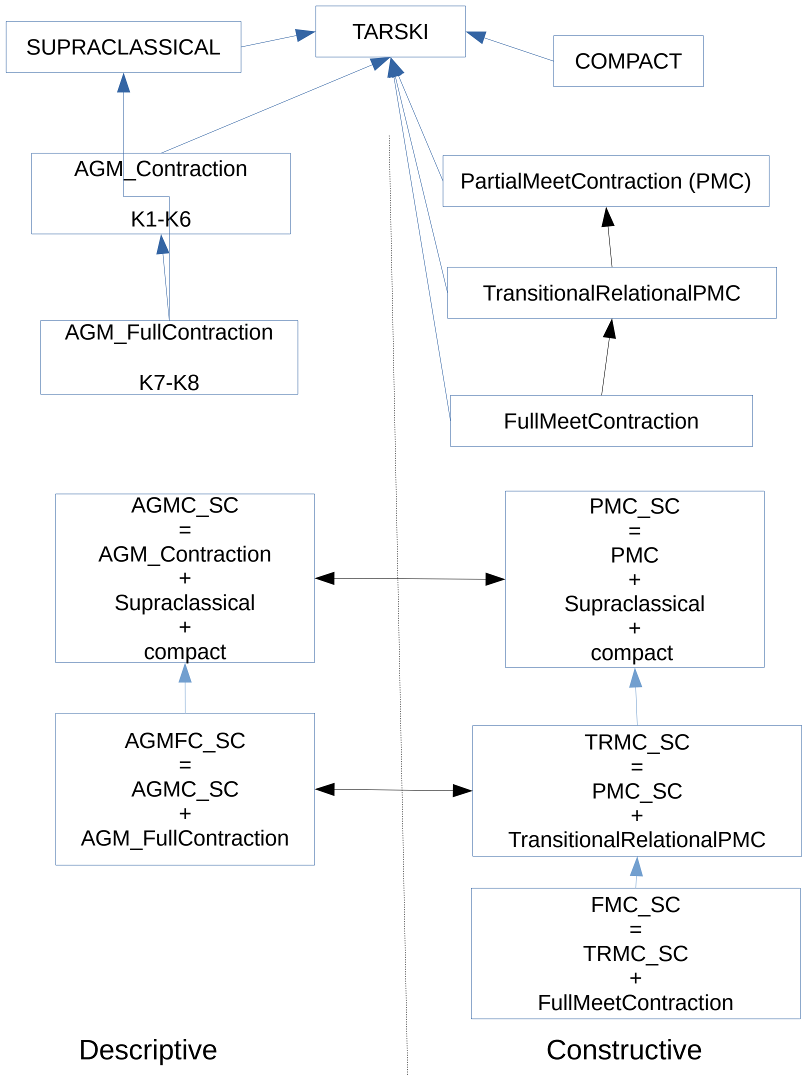
end

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 quivalent sentences to be freely substituted for each other

locale *AGM-Contraction* = *Tarskian-logic* +
fixes *contraction*::‹$'a$ *set* ⇒ $'a$ ⇒ $'a$ *set*› (**infix** ‹$\div$› *55*)
assumes *contract-closure*: ‹$K = Cn(A) \Longrightarrow K \div \varphi = Cn(K \div \varphi)$›
and *contract-inclusion*: ‹$K = Cn(A) \Longrightarrow K \div \varphi \subseteq K$›
and *contract-vacuity*: ‹$K = Cn(A) \Longrightarrow \varphi \notin K \Longrightarrow K \div \varphi = K$›
and *contract-success*: ‹$K = Cn(A) \Longrightarrow \varphi \notin Cn(\{\}) \Longrightarrow \varphi \notin K \div \varphi$›
and *contract-recovery*: ‹$K = Cn(A) \Longrightarrow K \subseteq ((K \div \varphi) \oplus \varphi)$›
and *contract-extensionality*: ‹$K = Cn(A) \Longrightarrow Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow K \div \varphi = K \div \psi$›

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 $K \div (\varphi \land \psi)$
- *contract-conj-inclusion* : If φ not in $K \div (\varphi \land \psi)$ 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*: ‹$K = Cn(A) \Longrightarrow (K \div \varphi) \cap (K \div \psi) \subseteq (K \div (\varphi .\land. \psi))$›
and *contract-conj-inclusion*: ‹$K = Cn(A) \Longrightarrow \varphi \notin (K \div (\varphi .\land. \psi)) \Longrightarrow ((K \div (\varphi .\land. \psi) \subseteq (K \div \varphi)))$›

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*: ‹$K = Cn(A) \Longrightarrow (K \div \varphi) \cap Cn(\{\varphi\}) \subseteq (K \div (\varphi .\wedge. \psi))$›
⟨*proof*⟩

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

corollary *contract-conj-inclusion-variant* : ‹$K = Cn(A) \Longrightarrow (K \div (\varphi .\wedge. \psi) \subseteq (K \div \varphi)) \vee (K \div (\varphi .\wedge. \psi) \subseteq (K \div \psi))$›
⟨*proof*⟩

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" ($K .\bot. \varphi$. 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*::‹$'a\ set \Rightarrow\ 'a\ \Rightarrow\ 'a\ set\ set$› (‹$\gamma$›)
assumes *is-selection*: ‹$K = Cn(A) \Longrightarrow (K .\bot. \varphi) \neq \{\} \Longrightarrow \gamma\ K\ \varphi \subseteq (K .\bot. \varphi)$›
assumes *tautology-selection*: ‹$K = Cn(A) \Longrightarrow (K .\bot. \varphi) = \{\} \Longrightarrow \gamma\ K\ \varphi = \{K\}$›
assumes *nonempty-selection*: ‹$K = Cn(A) \Longrightarrow \gamma\ K\ \varphi \neq \{\}$›
assumes *extensional-selection*: ‹$K = Cn(A) \Longrightarrow Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow \gamma\ K\ \varphi = \gamma\ K\ \psi$›

— 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 .⊥. φ 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*::‹$'a\ set \Rightarrow ('a\ set \Rightarrow 'a \Rightarrow 'a\ set\ set) \Rightarrow 'a \Rightarrow 'a\ set$› (‹- ÷- -› [*60,50,60*]*55*)
where *mc*: ‹$(A \div_\gamma \varphi) \equiv \bigcap(\gamma\ A\ \varphi)$›

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*: ‹$K = Cn(A) \Longrightarrow K \div_\gamma \varphi \subseteq K$›
⟨*proof*⟩

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*: ‹$K = Cn(A) \Longrightarrow \neg\ K \vdash \varphi \Longrightarrow K \div_\gamma \varphi = K$›
⟨*proof*⟩

pmc-closure : a partial meet contraction is logically closed

lemma *pmc-closure*: ‹$K = Cn(A) \Longrightarrow (K \div_\gamma \varphi) = Cn(K \div_\gamma \varphi)$›
⟨*proof*⟩

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*: ‹$K = Cn(A) \Longrightarrow Cn(\{\varphi\}) = Cn(\{\psi\}) \Longrightarrow K \div_\gamma \varphi = K \div_\gamma \psi$›
⟨*proof*⟩

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

lemma *pmc-tautology*: ‹$K = Cn(A) \Longrightarrow \Vdash \varphi \Longrightarrow K \div_\gamma \varphi = K$›
⟨*proof*⟩

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

definition (**in** *Tarskian-logic*) *completion*::‹('a set ⇒ 'a ⇒ 'a set set) ⇒ 'a set ⇒ 'a ⇒ 'a set set› (‹*›)
where ‹* γ A $\varphi \equiv$ *if* $(A .\bot. \varphi) = \{\}$ *then* $\{A\}$ *else* $\{B.\ B \in A .\bot. \varphi \wedge \bigcap (\gamma\ A\ \varphi) \subseteq B\}$›

lemma *selection-completion*: $K = Cn(A) \Longrightarrow \gamma\ K\ \varphi \subseteq * \gamma\ K\ \varphi$
⟨*proof*⟩

lemma (**in** *Tarskian-logic*) *completion-completion*: $K = Cn(A) \Longrightarrow * (* \gamma)\ K\ \varphi = * \gamma\ K\ \varphi$
⟨*proof*⟩

lemma *pmc-completion*: ‹$K = Cn(A) \Longrightarrow K \div_{*\gamma} \varphi = K \div_{\gamma} \varphi$›
⟨*proof*⟩

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 $K .\bot. \varphi$ 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*::‹'a set ⇒ 'a set ⇒ 'a set ⇒ bool› (‹- $\preceq_{-}$ -› [60,50,60]55)
assumes *trans-rel*: ‹$K = Cn(A) \Longrightarrow B \preceq_K C \Longrightarrow C \preceq_K D \Longrightarrow B \preceq_K D$›
assumes *nonempty-rel*: ‹$K = Cn(A) \Longrightarrow (K .\bot. \varphi) \neq \{\} \Longrightarrow \exists B \in (K .\bot. \varphi).\ (\forall C \in (K .\bot. \varphi).\ C \preceq_K B)$› — pas clair dans la litterature

fixes *rel-sel*::‹'a set ⇒ 'a ⇒ 'a set set› (‹γ_{TR}›)
defines *rel-sel*: ‹$\gamma_{TR}\ K\ \varphi \equiv$ *if* $(K .\bot. \varphi) = \{\}$ *then* $\{K\}$

else $\{B.\ B{\in}(K\ .\bot.\ \varphi) \wedge (\forall\ C{\in}(K\ .\bot.\ \varphi).\ C \preceq_K B)\}$›

begin

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

sublocale *PartialMeetContraction* **where** *selection* $= \gamma_{TR}$
⟨*proof*⟩

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*::‹'*a set* ⇒ '*a* ⇒ '*a set set*› (‹γ_{FC}›)
defines *full-sel*: ‹γ_{FC} $K\ \varphi \equiv$ *if* $K\ .\bot.\ \varphi = \{\}$ *then* $\{K\}$ *else* $K\ .\bot.\ \varphi$›

begin

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

sublocale *TransitivelyRelationalMeetContraction* **where** *relation* = ‹λ *K A B*. *True*› **and** *rel-sel*$=\gamma_{FC}$
⟨*proof*⟩

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*: ‹$K = Cn(A) \Longrightarrow K \subseteq ((K \div_\gamma \varphi) \oplus \varphi)$›
⟨*proof*⟩

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

lemma *pmc-success*: ‹$K = Cn(A) \Longrightarrow \varphi \notin Cn(\{\}) \Longrightarrow \varphi \notin K \div_\gamma \varphi$›
⟨*proof*⟩

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* = ‹$\lambda A\ \varphi.\ A \div_\gamma \varphi$›
⟨*proof*⟩

end

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

begin

obs 2.5 page 514

definition *AGM-selection*::‹'a set ⇒ 'a ⇒ 'a set set› (‹γ_{AGM}›)
where *AGM-sel*: ‹$\gamma_{AGM}\ A\ \varphi \equiv$ *if* A .⊥. $\varphi = \{\}$ *then* $\{A\}$ *else* $\{B.\ B \in A$.⊥. $\varphi \wedge A \div \varphi \subseteq B\}$›

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

sublocale *PartialMeetContraction* **where** *selection* = γ_{AGM}
⟨*proof*⟩

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

lemma *contraction-is-pmc*: ‹$K = Cn(A) \Longrightarrow K \div \varphi = K \div_{\gamma_{AGM}} \varphi$› — requires a supraclassical logic
⟨*proof*⟩

lemma *contraction-with-completion*: ‹$K = Cn(A) \Longrightarrow K \div \varphi = K \div_{*}\ \gamma_{AGM}\ \varphi$›
⟨*proof*⟩

end

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

begin

A transitively relational selection function respect conjuctive overlap.

lemma *rel-sel-conj-overlap*: ‹$K = Cn(A) \Longrightarrow \gamma_{TR}\ K\ (\varphi\ .\wedge.\ \psi) \subseteq \gamma_{TR}\ K\ \varphi \cup \gamma_{TR}\ K\ \psi$›
⟨*proof*⟩

A transitively relational meet contraction respect conjuctive overlap.

lemma *trmc-conj-overlap*: ‹$K = Cn(A) \Longrightarrow (K \div_{\gamma_{TR}} \varphi) \cap (K \div_{\gamma_{TR}} \psi) \subseteq (K \div_{\gamma_{TR}} (\varphi\ .\wedge.\ \psi))$›
⟨*proof*⟩

A transitively relational selection function respect conjuctive inclusion

lemma *rel-sel-conj-inclusion*: ‹$K = Cn(A) \Longrightarrow \gamma_{TR}\ K\ (\varphi\ .\wedge.\ \psi) \cap (K\ .\bot.\ \varphi) \neq \{\} \Longrightarrow \gamma_{TR}\ K\ \varphi \subseteq \gamma_{TR}\ K\ (\varphi\ .\wedge.\ \psi)$›
⟨*proof*⟩

A transitively relational meet contraction respect conjuctive inclusion

lemma *trmc-conj-inclusion*: ‹$K = Cn(A) \Longrightarrow \varphi \notin (K \div_{\gamma_{TR}} (\varphi\ .\wedge.\ \psi)) \Longrightarrow ((K \div_{\gamma_{TR}} (\varphi\ .\wedge.\ \psi) \subseteq (K \div_{\gamma_{TR}} \varphi)))$›
⟨*proof*⟩

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* = ‹$\lambda A\ \varphi.\ A\ \div_{\gamma_{TR}} \varphi$›
⟨*proof*⟩

end

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

begin

An AGM relation is defined as ?

definition *AGM-relation*::‹$'a\ set \Rightarrow 'a\ set \Rightarrow 'a\ set \Rightarrow bool$›
where *AGM-rel*: ‹$AGM\text{-}relation\ C\ K\ B \equiv (C = K \wedge B = K) \vee ((\exists\varphi.\ K \vdash \varphi \wedge C \in K\ .\bot.\ \varphi)$
$\wedge (\exists\varphi.\ K \vdash \varphi \wedge B \in K\ .\bot.\ \varphi \wedge K \div \varphi \subseteq B)$
$\wedge (\forall\varphi.\ (K \vdash \varphi \wedge C \in K\ .\bot.\ \varphi \wedge B \in K\ .\bot.\ \varphi \wedge K \div \varphi \subseteq C) \longrightarrow K \div \varphi \subseteq B))$›

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

definition *AGM-relational-selection*::‹$'a\ set \Rightarrow 'a \Rightarrow 'a\ set\ set$› (‹$\gamma_{AGMTR}$›)
where *AGM-rel-sel*: ‹$\gamma_{AGMTR}\ K\ \varphi \equiv$ *if* $(K\ .\bot.\ \varphi) = \{\}$
then $\{K\}$

else $\{B.\ B \in (K\ .\bot.\ \varphi) \wedge (\forall\ C \in (K\ .\bot.\ \varphi).\ \textit{AGM-relation}$ $C\ K\ B)\}$›

lemma *AGM-rel-sel-completion*: ‹$K = Cn(A) \Longrightarrow \gamma_{AGMTR}\ K\ \varphi = *\ \gamma_{AGM}\ K\ \varphi$›
⟨*proof*⟩

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

sublocale *TransitivelyRelationalMeetContraction* **where** *relation* = *AGM-relation* **and** *rel-sel* = ‹γ_{AGMTR}›
⟨*proof*⟩
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*: ‹$K = Cn(A) \Longrightarrow K \vdash \varphi \Longrightarrow (K \div_{\gamma_{FC}} \varphi) = K \cap Cn(\{.\neg\ \varphi\})$›
⟨*proof*⟩

lemma *full-meet-weak2*:‹$K = Cn(A) \Longrightarrow K \vdash \varphi \Longrightarrow Cn((K \div_{\gamma_{FC}} \varphi) \cup \{.\neg\ \varphi\}) = Cn(\{.\neg\ \varphi\})$›
⟨*proof*⟩

end

end

5 Revisions

The third operator of belief change introduce 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*:: ‹$'a$ *set* ⇒ $'a$ ⇒ $'a$ *set*› (**infix** ‹∗› *55*)

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

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*: ‹$K = Cn(A) \Longrightarrow K * (\varphi\ .\wedge.\ \psi) \subseteq (K * \varphi) \oplus \psi$›
  and *revis-subexpansion*: ‹$K = Cn(A) \Longrightarrow .\neg\ \psi \notin (K * \varphi) \Longrightarrow (K * \varphi) \oplus \psi \subseteq K * (\varphi\ .\wedge.\ \psi)$›

begin

— important lemmas/corollaries that can replace the previous assumptions
corollary *revis-superexpansion-ext* : ‹$K = Cn(A) \Longrightarrow (K * \varphi) \cap (K * \psi) \subseteq (K * (\varphi\ .\vee.\ \psi))$›
⟨*proof*⟩

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* $= \langle\lambda K\ \varphi.\ K \cap (K * .\neg\ \varphi)\rangle$
$\langle proof \rangle$

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* $= \langle\lambda K\ \varphi.\ (K \div .\neg\ \varphi) \oplus \varphi\rangle$
$\langle proof \rangle$

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

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

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

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

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.