



HAL
open science

Equilibria und weiteres Heiteres

Dov Gabbay, Karl Schlechta

► **To cite this version:**

| Dov Gabbay, Karl Schlechta. Equilibria und weiteres Heiteres. 2009. hal-00406635v4

HAL Id: hal-00406635

<https://hal.science/hal-00406635v4>

Preprint submitted on 7 Dec 2009 (v4), last revised 15 Aug 2011 (v10)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Equilibria und weiteres Heiteres

Dov M Gabbay *
King's College, London †
and
Bar-Ilan University, Israel ‡
and
University of Luxembourg §

Karl Schlechta ¶
Laboratoire d'Informatique Fondamentale de Marseille ||

December 7, 2009

Abstract

We investigate several technical questions.

Contents

1	Introduction	2
2	Equilibrium logic	2
2.1	Semantics and interpolation of many valued logics	2
2.1.1	Generalization of model sets and (in)essential variables	2
2.1.2	Implication	2
2.1.3	Liberation of a variable (generalization of the Cartesian product)	3
2.2	Some logical results for the base logic Here/There HT	3
2.2.1	Basic definitions and results	3
2.2.2	Interpolation	5
2.3	The equilibrium logic EQ	8
2.3.1	EQ has no interpolation of the form $\phi \vdash \alpha \sim \psi$	8
2.3.2	EQ has no interpolation of the form $\phi \sim \alpha \vdash \psi$	8
2.3.3	EQ has interpolation of the form $\phi \sim \alpha \sim \psi$	8
2.3.4	Definability of minimal models	9
3	Countably many disjoint sets	9
4	Operations on logical structures	10
4.1	Introduction	10
4.1.1	Conditionals	10
4.2	Theory revision	11
4.2.1	“Meta-revision”	11
4.3	Preferential systems	11
4.3.1	Why does \sim not modify itself?	11
4.3.2	(Meta) Operations on logics	12
4.3.3	Implementation	12
4.3.4	Operations on linear and ranked structures	12
4.3.5	Intuitionistic preferential logic	12

*Dov.Gabbay@kcl.ac.uk, www.dcs.kcl.ac.uk/staff/dg

†Department of Computer Science, King's College London, Strand, London WC2R 2LS, UK

‡Department of Computer Science, Bar-Ilan University, 52900 Ramat-Gan, Israel

§Computer Science and Communications, Faculty of Sciences, 6, rue Coudenhove-Kalergi, L-1359 Luxembourg

¶ks@cmi.univ-mrs.fr, karl.schlechta@web.de, http://www.cmi.univ-mrs.fr/~ks

||UMR 6166, CNRS and Université de Provence, Address: CMI, 39, rue Joliot-Curie, F-13453 Marseille Cedex 13, France

4.4	Non-monotonic interpolation	13
5	Independence and multiplication of abstract size	13
5.1	Definition of independence	13
5.1.1	Discussion	14
5.1.2	Independence and multiplication of abstract size	14
5.2	Multiplication of size for subsets	15
5.2.1	Properties	15
5.2.2	Corresponding preferential relations	16
5.3	Multiplication of size for subspaces	16
5.3.1	Properties	16
5.3.2	Corresponding preferential relations	17
5.3.3	Language change	19
5.3.4	Interpolation	19
5.3.5	A relevance problem	20
5.3.6	Small subspaces	21
5.3.7	Distance relations	21
5.3.8	Reactivity and language change	21
5.4	Summary of properties	21
6	General semantic interpolation	23
6.1	Introduction	23
6.1.1	Generalization of model sets and (in)essential variables	23
6.2	Many-valued propositional interpolation	23
6.3	Many-valued propositional modal interpolation	25
6.4	Many-valued first order interpolation	25
	References	25

1 Introduction

We present here various small results, which may one day be published in a bigger paper, and which we wish to make already available to the community.

2 Equilibrium logic

2.1 Semantics and interpolation of many valued logics

We generalize the concepts and ideas of [GS09c].

2.1.1 Generalization of model sets and (in)essential variables

See GIN.

2.1.2 Implication

We now discuss implication. $\phi \rightarrow \psi$ will, just as any formula, have a certain truth value in a model m . But we can also look only at the maximal truth value (if there is an order on those values), i.e., we are only interested in those models where $\phi \rightarrow \psi$ has maximal value. In this case, the interpolation problem is the following:

Suppose $\phi \rightarrow \psi$ has everywhere maximal truth value v , i.e., $f_{\phi \rightarrow \psi}$ is the constant function v . Find a “simple” formula α , such that $\phi \rightarrow \alpha$ and $\alpha \rightarrow \psi$ both have everywhere maximal truth value. “Simple” means here: formulated in the common language of ϕ and ψ . In the present and similar cases, there is an order on the set of truth values, and implication respects that order, in the following sense:

$\phi \rightarrow \psi$ has maximal truth value in m iff the truth value of ϕ in m is smaller or equal to the truth value of ψ in m , i.e., $f_\phi(m) \leq f_\psi(m)$.

The semantic version of interpolation is now (the generalization to arbitrary $f : M \rightarrow V$ is straightforward):

Definition 2.1

(1) Given f_ϕ and f_ψ such that $\forall m \in M (f_\phi(m) \leq f_\psi(m))$, in short $f_\phi \leq f_\psi$, find α such that $\forall m \in M (f_\phi(m) \leq f_\alpha(m) \leq f_\psi(m))$, in short $f_\phi \leq f_\alpha \leq f_\psi$, where α is “simple” relative to ϕ and ψ . I.e., α contains only variables common to ϕ and ψ , or, more robust: $R(\alpha) \subseteq R(\phi) \cap R(\psi)$.

(2) Given arbitrary $f, g : M \rightarrow V$ such that $f \leq g$, find $h : M \rightarrow V$ such that $f \leq h \leq g$, where h is simple relative to f and g , i.e., $R(h) \subseteq R(f) \cap R(g)$.

2.1.3 Liberation of a variable (generalization of the Cartesian product)

In classical logic, we started with a model set Σ (written as a set of sequences), and “liberated” the variables we were not allowed to use, admitting now any value for the liberated variables, i.e., taking the Cartesian product for those variables. E.g, if our model set contained just the model $\langle p, q \rangle$, and we liberated q , the model set was now $\{\langle p, q \rangle, \langle p, \neg q \rangle\}$. When we write the first model set as a function f , the second as a function g , we have:

$$f(\langle p, q \rangle) = 1, f(\langle p, \neg q \rangle) = f(\langle \neg p, q \rangle) = f(\langle \neg p, \neg q \rangle) = 0$$

$$g(\langle p, q \rangle) = g(\langle p, \neg q \rangle) = 1, g(\langle \neg p, q \rangle) = g(\langle \neg p, \neg q \rangle) = 0$$

Written otherwise, and in a way we can generalize:

$$g(m) := \sup\{f(m') : m \text{ and } m' \text{ differ at most on } q\}$$

Thus, we define for multi-valued logics:

Definition 2.2

Let $f : M \rightarrow V$ be any function, $x \in V$. We define

$$f_{(x)} : M_{\mathcal{L}} \rightarrow V \text{ by } f_{(x)}(m) := \sup\{f(m') : m' =_{(x)} m\}$$

where $=_{(x)}$ is equality with the possible exception of x .

If $f = f_\phi$ for some ϕ , and there is some (usually not unique) ϕ' such that $f_{\phi'} = f_{\phi, (x)}$, we also write by abuse of language $\phi_{(x)}$ for any such ϕ' .

THIS DEFINITION DOES NOT WORK

In the classical case, one had to be a model, i.e. max value was taken, as otherwise the product would have been empty. E.g., take $\sigma \models \phi$. Then all values for all variables x were $max = 1$. Here this is not the case, i.e. they can all have value 1 on x , the variable to be neglected. But we want 2 as neglected value now. What we can do: we take the set X , take the projection on $V - \{x\}$, $Pro_x(X)$ and consider the value of $\sigma \upharpoonright (V - \{x\})$ in $Pro_x(X)$. As the whole thing is multiplicative, it should work.

Fact 2.1

If ϕ does not contain the variable x , and validity is defined as usual by induction on the components, then $\phi_{(x)} = \phi$.

Proof

Validity of ϕ in a model does not depend on the value of x in that model. \square

Remark 2.2

This definition is well-behaved:

(1) x is inessential for $f_{(x)}$, as it should be. But we also had to make sure that we took the right one of such functions, not defined through the inf, or so.

(2) This works also for things like $a \wedge \neg a \wedge b$ - gives the empty set, or its analogue, i.e. the function which is 0 everywhere.

(3) Consider now classical \vee , for a disjunction of conjunctions. We have $f_{\phi \vee \psi}(m) = \sup\{f_\phi(m), f_\psi(m)\}$, and thus $f_{\phi \vee \psi, (x)} = \sup\{f_{\phi, (x)}, f_{\psi, (x)}\}$, so \vee commutes with liberation.

(4) If, in an arbitrary logic with an order on the truth values, \vee is semantically again the sup, we can do the same as for classical logic, and work with the disjuncts separately, and put them together at the end.

\square

2.2 Some logical results for the base logic Here/There HT

2.2.1 Basic definitions and results

There are 3 truth values, 0, 1, 2, so any model is a function from the set of propositional variables into $\{0, 1, 2\}$.

For a model σ , and a propositional variable x , set

$$\sigma(x) = 0 \text{ iff } x \text{ holds neither here nor there}$$

$$\sigma(x) = 1 \text{ iff } x \text{ holds only there}$$

$$\sigma(x) = 2 \text{ iff } x \text{ holds here and there}$$

We introduce the operators $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ by their semantic matrices.

Negation:

a	$\neg a$
0	2
1	0
2	0

Implication $a \rightarrow b$:

	b	0	1	2
a	$a \rightarrow b$			
0		2	2	2
1		0	2	2
2		0	1	2

Conjunction $a \wedge b$:

	b	0	1	2
a	$a \wedge b$			
0		0	0	0
1		0	1	1
2		0	1	2

Disjunction $a \vee b$:

	b	0	1	2
a	$a \vee b$			
0		0	1	2
1		1	1	2
2		2	2	2

We can add as derived operator

Equivalence $a \leftrightarrow b$:

	b	0	1	2
a	$a \leftrightarrow b$			
0		2	0	0
1		0	2	1
2		0	1	2

The following results were checked with a small computer program:

Fact 2.3

(1)

The following formulas are definable with 1 variable a (values given for a). This means, that all formulas ϕ generated with a single variable a have a semantic function f_ϕ which is exactly the same as the semantic function of one of the 6 formulas below. Thus, this set is semantically closed. (And, of course, the 6 formulas below all have different semantic functions.)

a	0	1	2
a	0	1	2
$\neg a$	2	0	0
$\neg\neg a$	0	2	2
$a \rightarrow a$	2	2	2
$\neg(a \rightarrow a)$	0	0	0
$\neg\neg a \rightarrow a$	2	1	2

(2) With 2 variables a, b are definable, using the operators $\neg, \rightarrow, \wedge, \vee$, 174 semantically different formulas. \vee is not needed, i.e. with or without \vee we have the same set of definable formulas. We have, e.g., $a \vee b \leftrightarrow \left((b \rightarrow (\neg\neg a \rightarrow a)) \rightarrow ((\neg a \rightarrow b) \wedge (\neg\neg a \rightarrow a)) \right)$.

(3) With the operators \neg, \wedge, \vee only are 120 semantically different formulas definable. Thus, \rightarrow cannot be expressed by the other operators.

Fact 2.4

The following semantic equivalences hold:

(Note: all except (14) hold also for 4 and 6 truth values, so probably for arbitrarily many truth values, but this is not checked so far.)

Triple negation can be simplified:

(1) $\neg\neg\neg a \leftrightarrow \neg a$

Disjunction and conjunction combine classically:

(2) $\neg(a \vee b) \leftrightarrow \neg a \wedge \neg b$

(3) $\neg(a \wedge b) \leftrightarrow \neg a \vee \neg b$

(4) $a \wedge (b \vee c) \leftrightarrow (a \wedge b) \vee (a \wedge c)$

(5) $a \vee (b \wedge c) \leftrightarrow (a \vee b) \wedge (a \vee c)$

Implication can be eliminated from combined negation and implication:

$$(6) \neg(a \rightarrow b) \leftrightarrow \neg\neg a \wedge \neg b$$

$$(7) (a \rightarrow \neg b) \leftrightarrow (\neg a \vee \neg b)$$

$$(8) (\neg a \rightarrow b) \leftrightarrow (\neg\neg a \vee b)$$

Implication can be put inside when combined with \wedge and \vee :

$$(9) (a \vee b \rightarrow c) \leftrightarrow ((a \rightarrow c) \wedge (b \rightarrow c))$$

$$(10) (a \wedge b \rightarrow c) \leftrightarrow ((a \rightarrow c) \vee (b \rightarrow c))$$

$$(11) (a \rightarrow b \wedge c) \leftrightarrow ((a \rightarrow b) \wedge (a \rightarrow c))$$

$$(12) (a \rightarrow b \vee c) \leftrightarrow ((a \rightarrow b) \vee (a \rightarrow c))$$

Nested implication can be flattened:

$$(13) (a \rightarrow (b \rightarrow c)) \leftrightarrow ((a \wedge b \rightarrow c) \wedge (a \wedge \neg c \rightarrow \neg b))$$

$$(14) ((a \rightarrow b) \rightarrow c) \leftrightarrow ((\neg a \rightarrow c) \wedge (b \rightarrow c) \wedge (a \vee \neg b \vee c))$$

(Thanks to D.Pearce for (14) and other help.)

2.2.2 Interpolation

Normal form

Fact 2.5

Every formula ϕ can be transformed into a semantically equivalent formula ψ of the following form:

$$(1) \psi \text{ has the form } \phi_1 \vee \dots \vee \phi_n$$

$$(2) \text{ every } \phi_i \text{ has the form } \phi_{i,1} \wedge \dots \wedge \phi_{i,m}$$

$$(3) \text{ every } \phi_{i,m} \text{ has one of the following forms:}$$

p , or $\neg p$, or $\neg\neg p$, or $p \rightarrow q$ - where p and q are propositional variables.

Proof

The numbers refer to Fact 2.4 (page 4).

We first push \neg downward, towards the interior:

- $\neg(\phi \wedge \psi)$ is transformed to $\neg\phi \vee \neg\psi$ by (3).
- $\neg(\phi \vee \psi)$ is transformed to $\neg\phi \wedge \neg\psi$ by (2).
- $\neg(\phi \rightarrow \psi)$ is transformed to $\neg\neg\phi \wedge \neg\psi$ by (6).

We next eliminate any $\phi \rightarrow \psi$ where ϕ and ψ are not propositional variables:

- $\neg\phi \rightarrow \psi$ is transformed to $\neg\neg\phi \vee \psi$ by (8).
- $\phi \wedge \phi' \rightarrow \psi$ is transformed to $(\phi \rightarrow \psi) \vee (\phi' \rightarrow \psi)$ by (10).
- $\phi \vee \phi' \rightarrow \psi$ is transformed to $(\phi \rightarrow \psi) \wedge (\phi' \rightarrow \psi)$ by (9).
- $(\phi \rightarrow \phi') \rightarrow \psi$ is transformed to $(\neg\phi \rightarrow \psi) \wedge (\phi' \rightarrow \psi) \wedge (\phi \vee \neg\phi' \vee \psi)$ by (14).
- $\phi \rightarrow \neg\psi$ is transformed to $\neg\phi \vee \neg\psi$ by (7).
- $\phi \rightarrow \psi \wedge \psi'$ is transformed to $(\phi \rightarrow \psi) \wedge (\phi \rightarrow \psi')$ by (11).
- $\phi \rightarrow \psi \vee \psi'$ is transformed to $(\phi \rightarrow \psi) \vee (\phi \rightarrow \psi')$ by (12).
- $\phi \rightarrow (\psi \rightarrow \psi')$ is transformed to $(\phi \wedge \psi \rightarrow \psi') \wedge (\phi \wedge \neg\psi' \rightarrow \neg\psi)$ by (13).

Finally, we push \wedge inside:

$$\phi \wedge (\psi \vee \psi') \text{ is transformed to } (\phi \wedge \psi) \vee (\phi \wedge \psi') \text{ by (4).}$$

□

Liberating a variable

We turn to liberating a propositional variable x .

Given ϕ , what is $\phi_{(x)}$, i.e., do we find ϕ' such that $f_{\phi_{(x)}} := f_{(\phi_{(x)})} = f_{\phi'}$?

Fact 2.6

$$(1) \text{ If } \phi = \phi' \vee \phi'', \text{ and } \phi'_{(x)}, \phi''_{(x)} \text{ both exist, then so does } \phi_{(x)}, \text{ and } \phi_{(x)} = \phi'_{(x)} \vee \phi''_{(x)}.$$

$$(2) \text{ If } \phi = \phi' \wedge \phi'', \phi'_{(x)} \text{ exists, and } \phi'' \text{ does not contain } x, \text{ then } \phi_{(x)} \text{ exists, and } \phi_{(x)} = \phi'_{(x)} \wedge \phi''.$$

Table 1: Table Neglecting a variable

Neglecting a variable		
ϕ	neglected variable(s)	$\phi_{(x)}$
$a \rightarrow b$	a	TRUE
$a \rightarrow b$	b	TRUE
$a \wedge (a \rightarrow b)$	a	b
$\neg a \wedge (a \rightarrow b)$	a	TRUE
$\neg\neg a \wedge (a \rightarrow b)$	a	$\neg\neg b$
$(a \rightarrow b) \wedge b$	b	TRUE
$(a \rightarrow b) \wedge \neg b$	b	$\neg a$
$(a \rightarrow b) \wedge \neg\neg b$	b	TRUE
$(a \rightarrow b) \wedge (b \rightarrow c)$	a	$b \rightarrow c$
$(a \rightarrow b) \wedge (b \rightarrow c)$	b	$a \rightarrow c$
$(a \rightarrow b) \wedge (b \rightarrow c)$	c	$a \rightarrow b$
$(a \rightarrow b) \wedge (b \rightarrow c) \wedge (b \rightarrow d)$	b	$(a \rightarrow c) \wedge (a \rightarrow d)$
$(a \rightarrow b) \wedge (b \rightarrow c) \wedge b$	b	$(a \rightarrow c) \wedge c$
$(a \rightarrow b) \wedge (b \rightarrow c) \wedge (c \rightarrow d)$	a, b	$a \rightarrow d$
$(c \rightarrow a) \wedge (a \rightarrow b) \wedge (b \rightarrow d) \wedge b \wedge \neg a$	a, b	$(c \rightarrow d) \wedge d \wedge \neg c$
$(b \rightarrow a) \wedge (a \rightarrow c) \wedge (d \rightarrow a) \wedge (a \rightarrow e)$	a	$b \vee d \rightarrow c \wedge e$
$(b \vee d \rightarrow a) \wedge (a \rightarrow c \wedge e)$	a	$b \vee d \rightarrow c \wedge e$

Proof

(1) We have to show $f_{\phi_{(x)}} = f_{(\phi'_{(x)} \vee \phi''_{(x)})}$.

By definition of validity of \vee , we have $f_{(\phi'_{(x)} \vee \phi''_{(x)})}(m) = \sup\{f_{\phi'_{(x)}}(m), f_{\phi''_{(x)}}(m)\}$. $f_{\phi_{(x)}}(m) := \sup\{f_{\phi}(m') : m' =_{(x)} m\}$, so $f_{(\phi'_{(x)} \vee \phi''_{(x)})}(m) = \sup\{\sup\{f_{\phi'}(m') : m' =_{(x)} m\}, \sup\{f_{\phi''}(m') : m' =_{(x)} m\}\} = \sup\{\sup\{f_{\phi'}(m'), f_{\phi''}(m')\} : m' =_{(x)} m\} =$ (again by definition of validity of \vee) $\sup\{f_{\phi' \vee \phi''}(m') : m' =_{(x)} m\} = \sup\{f_{\phi}(m') : m' =_{(x)} m\} = f_{(\phi_{(x)})}(m)$.

(2) We have to show $f_{\phi_{(x)}} = f_{(\phi'_{(x)} \wedge \phi''_{(x)})}$. By definition of validity of \wedge , we have $f_{(\phi'_{(x)} \wedge \phi''_{(x)})}(m) = \inf\{f_{\phi'_{(x)}}(m), f_{\phi''_{(x)}}(m)\}$. So $f_{(\phi'_{(x)} \wedge \phi''_{(x)})}(m) = \inf\{\sup\{f_{\phi'}(m') : m' =_{(x)} m\}, \sup\{f_{\phi''}(m') : m' =_{(x)} m\}\} =$ (as ϕ'' does not contain x) $\inf\{\sup\{f_{\phi'}(m') : m' =_{(x)} m\}, f_{\phi''}(m)\} = \sup\{\inf\{f_{\phi'}(m'), f_{\phi''}(m)\} : m' =_{(x)} m\} =$ (again by definition of validity of \wedge , and by the fact that ϕ'' does not contain x) $\sup\{f_{\phi' \wedge \phi''}(m') : m' =_{(x)} m\} = \sup\{f_{\phi}(m') : m' =_{(x)} m\} = f_{(\phi_{(x)})}(m)$.

□

Thus, we can calculate disjunctions separately, and also conjunctions, as long as they have no variables in common. In classical logic, we are finished, as we can break down conjunctions into parts which have no variables in common. The problem here are formulas of the type $a \rightarrow b$, as they may have variables in common with other conjuncts, and, as we saw in Fact 2.3 (page 4) (2) and (3), they cannot be eliminated.

We have to consider situations like $(a \rightarrow b) \wedge (b \rightarrow c)$, $a \wedge (a \rightarrow b)$, etc.

Table 2.2.2 (page 6) presents some results, calculated by computer. The left hand column gives the formula ϕ , the center column the neglected variable x , and the right hand column $\phi_{(x)}$, or $\phi_{(x,y)}$, when we neglected two variables.

We tested that these projections hold for 3, 4, 6 truth values. Probably they will hold for arbitrarily many truth values. (TRUE is the maximal truth value.)

We turn to the general situation.

A variable a may occur in the following way:

- (1) a
- (2) $\neg a$
- (3) $\neg\neg a$
- (4) $a \rightarrow b$
- (5) $b \rightarrow a$
- (6) and any finite combinations of the above

To do as many cases together as possible, it is useful to use Fact 2.4 (page 4) (9) and (11) backwards, to obtain general formulas. We then see that the cases to examine are of the form:

$$\phi = ((b_1 \vee \dots \vee b_n) \rightarrow a) \wedge (a \rightarrow (c_1 \wedge \dots \wedge c_m)) \wedge \sigma a \wedge \tau a,$$

where n, m may be 0, and σ, τ are absence (\emptyset , no a), $1a = a$, $\neg a$, or $\neg\neg a$.

First, we may simplify, as $\sigma a \wedge \tau a$ is equivalent to some $\sigma' a$, where $\sigma' a$ may also be FALSE. Moreover, and $a \rightarrow a$ may be replaced by TRUE, i.e. omitted.

Conjecture

(1) If $\sigma'a = FALSE$, then $\phi_{(a)} = FALSE$.

Assume now that $\sigma'a$ is consistent.

(2) If $n = m = 0$, then $\phi_{(a)} = TRUE$.

(3) If $n = 0$, and $\sigma'a = a$, then $\phi_{(a)} = c_1 \wedge \dots \wedge c_m$.

(4) If $n = 0$, and $\sigma'a = \neg a$, then $\phi_{(a)} = \neg c_1 \wedge \dots \wedge \neg c_m$.

(5) If $n = 0$, and $\sigma'a = \neg a$ or $\sigma'a = TRUE$, then $\phi_{(a)} = TRUE$.

(7) If $m = 0$, and $\sigma'a = a$ or $\sigma'a = TRUE$, then $\phi_{(a)} = TRUE$.

(8) If $m = 0$, and $\sigma'a = \neg a$, then $\phi_{(a)} = TRUE$.

(9) If $m = 0$, and $\sigma'a = \neg a$, then $\phi_{(a)} = \neg b_1 \wedge \dots \wedge \neg b_n$.

Suppose now in addition that $n, m \neq 0$.

(10) If $\sigma'a = TRUE$, then $\phi_{(a)} = (b_1 \vee \dots \vee b_n) \rightarrow (c_1 \wedge \dots \wedge c_m)$.

(11) If $\sigma'a = a$, then $\phi_{(a)} = ((b_1 \vee \dots \vee b_n) \rightarrow (c_1 \wedge \dots \wedge c_m)) \wedge (c_1 \wedge \dots \wedge c_m)$.

(12) If $\sigma'a = \neg a$, then $\phi_{(a)} = ((b_1 \vee \dots \vee b_n) \rightarrow (c_1 \wedge \dots \wedge c_m)) \wedge (\neg c_1 \wedge \dots \wedge \neg c_m)$.

(13) If $\sigma'a = \neg a$, then $\phi_{(a)} = ((b_1 \vee \dots \vee b_n) \rightarrow (c_1 \wedge \dots \wedge c_m)) \wedge (\neg c_1 \vee \dots \vee \neg c_m)$.

Remark 2.7

We cannot improve the value of $\phi \rightarrow \psi$ by taking a detour $\phi \rightarrow \alpha_1 \rightarrow \dots \rightarrow \alpha_n \rightarrow \psi$ because the destination determines the value: in any column of \rightarrow , there is only max and a constant value. And if we go further down than needed, we get only worse, going from right to left deteriorates the values in the lines. \square

We can achieve this also as follows:

We close under the following rules:

(1) \rightarrow under transitivity, i.e.

$$((b_1 \vee \dots \vee b_n) \rightarrow a) \wedge (a \rightarrow (c_1 \wedge \dots \wedge c_m)) \Rightarrow ((b_1 \vee \dots \vee b_n) \rightarrow (a \rightarrow (c_1 \wedge \dots \wedge c_m)))$$

(2) $\sigma'a$ and \rightarrow as follows:

$$((b_1 \vee \dots \vee b_n) \rightarrow a) \wedge (a \rightarrow (c_1 \wedge \dots \wedge c_m)), a \Rightarrow c_1 \wedge \dots \wedge c_m$$

$$((b_1 \vee \dots \vee b_n) \rightarrow a) \wedge (a \rightarrow (c_1 \wedge \dots \wedge c_m)), \neg a \Rightarrow \neg c_1 \wedge \dots \wedge \neg c_m$$

$$((b_1 \vee \dots \vee b_n) \rightarrow a) \wedge (a \rightarrow (c_1 \wedge \dots \wedge c_m)), \neg a \Rightarrow \neg b_1 \wedge \dots \wedge \neg b_n$$

and finally erase all formulas containing a .

For transitivity of the implication: look where the sup exists. if $value(b) \leq value(c)$, then the max is clear. Otherwise, look at the cases.

The other cases must be similar.

It seems that we can close first under the following operations, and then just forget the formulas, where the neglected variables are:

$$a \wedge (a \rightarrow b) \Rightarrow b$$

$$\neg a \wedge (a \rightarrow b) \Rightarrow \neg b$$

$$(a \rightarrow b) \wedge \neg b \Rightarrow \neg a$$

$$(a \rightarrow b) \wedge (b \rightarrow c) \Rightarrow a \rightarrow c$$

Once these formulas are added, just omit the formulas with the variable in question inside. This is the description of the generalized product.

We do this for all variables first, so even if we treat a , and then b , and have erased $a \rightarrow b$ without trace for a , the perhaps needed trace for b is still there.

This allows us probably to use Fact 3.1 (semantic interpolation) of [GS09c], and transform it into syntactic interpolation. (The result is known, we just give our proof.)

Let ψ be the new conjunction, resulting from the old formula ϕ .

We have to prove that above construction satisfies the sup-condition (*SUP*), i.e. that $f_\psi = f_{\phi,(x)} = \sup\{f_\phi(m') : m' =_{(x)} m\}$.

We also have to show that we can do this simultaneously for several variables, perhaps it suffices to show that the order in which we do it does not matter.

2.3 The equilibrium logic EQ

See [PV09] for the background.

Definition 2.3

- (1) A model m is total iff no value of any variable in m is 1.
- (2) $m \prec m'$ iff there is a variable a such that $m'(a) = 2$, $m(a) = 1$.
- ($\sigma \prec \tau$ iff T is preserved, and H goes down. Thus, only changes from 2 to 1 are possible when $\sigma \prec \tau$.)
- (3) $m \in X$ is an equilibrium model of X iff m is total, and there is no $m' \prec m$, $m' \in X$.
- (4) $\eta(X)$ will be the set of equilibrium models of X .

Remark 2.8

Consequently, we have a sort of “anti-smoothnes”: if a model is not minimal, then any model below it is NOT chosen.

Fact 2.9

It is not necessary to make it uniform, we can also just distinguish cases.

Definability of EQ models for a formula ϕ :

- (1) ϕ has $\neg a$ models, then $\phi \wedge \neg a$ defines this part.
- (2) ϕ has $\neg\neg a$ models, but there is a model of value 1 for a , then $\phi \wedge \neg a$ defines this part.
- (3) ϕ has no model with value 1 for a : then ϕ itself defines the EQ models of ϕ .

2.3.1 EQ has no interpolation of the form $\phi \vdash \alpha \mid \sim \psi$

Example 2.1

Work with 3 variables, a, b, c .

Consider $\Sigma := \{\langle 0, 2, 2 \rangle, \langle 2, 1, 0 \rangle, \langle 2, 2, 0 \rangle\}$.

By the above, and classical behaviour of “or” and “and”, Σ is definable by $(\neg a \wedge b \wedge c) \vee (a \wedge \neg\neg b \wedge \neg c)$.

Note that $\langle 2, 2, 0 \rangle$ is total, but $\langle 2, 1, 0 \rangle \prec \langle 2, 2, 0 \rangle$, thus $\mu(\Sigma) = \{\langle 0, 2, 2 \rangle\}$.

So $\Sigma \vdash c = 2$ (or $\Sigma \vdash \Box c$). Let $X' := \{a, b\}$, $X'' := \{c\}$.

All possible interpolants Γ must not contain a or b as essential variables, and they must contain Σ . The smallest candidate Γ is $\Pi X' \times \{0, 2\}$. But $\sigma := \langle 0, 0, 0 \rangle \in \Gamma$, σ is total, and there cannot be any $\tau \prec \sigma$, so $\sigma \in \mu(\Gamma)$, so $\Gamma \not\vdash c = 2$.

For completeness' sake, we write all elements of Γ :

- $\langle 0, 0, 0 \rangle$ $\langle 0, 0, 2 \rangle$
- $\langle 0, 1, 0 \rangle$ $\langle 0, 1, 2 \rangle$
- $\langle 0, 2, 0 \rangle$ $\langle 0, 2, 2 \rangle$
- $\langle 1, 0, 0 \rangle$ $\langle 1, 0, 2 \rangle$
- $\langle 1, 1, 0 \rangle$ $\langle 1, 1, 2 \rangle$
- $\langle 1, 2, 0 \rangle$ $\langle 1, 2, 2 \rangle$
- $\langle 2, 0, 0 \rangle$ $\langle 2, 0, 2 \rangle$
- $\langle 2, 1, 0 \rangle$ $\langle 2, 1, 2 \rangle$
- $\langle 2, 2, 0 \rangle$ $\langle 2, 2, 2 \rangle$

Recall that no sequence containing 1 is total, and when we go from 2 to 1, we have a smaller model. Thus, $\mu(\Gamma) = \{\langle 0, 0, 0 \rangle, \langle 0, 0, 2 \rangle\}$.

2.3.2 EQ has no interpolation of the form $\phi \mid \sim \alpha \vdash \psi$

Example 2.2

Consider 2 variables, a, b , and $\Sigma := \{0, 2\} \times \{0, 1, 2\}$

No σ containing 1 can be in $\mu(\Sigma)$, as a matter of fact, $\mu(\Sigma) = \{\langle 0, 0 \rangle, \langle 2, 0 \rangle\}$. Σ is defined by $a \vee \neg a$, $\mu(\Sigma)$ is defined by $(a \vee \neg a) \wedge \neg b$.

So we have $a \vee \neg a \vdash b \vee \neg b$, even $a \vee \neg a \vdash \neg b$.

The only possible interpolants are TRUE or FALSE. $a \vee \neg a \not\vdash FALSE$, and $TRUE \not\vdash \neg b$.

2.3.3 EQ has interpolation of the form $\phi \mid \sim \alpha \mid \sim \psi$

We give a rough argument for the semantic version, but we do not know whether the set in question is definable.

We consider $\phi \mid \sim \psi$.

Let X' be the variables not in ψ , X'' the others.

Fact 2.10

Let $\mu(\Sigma) \neq \emptyset$. Let $\Pi' := \Pi X'$.

$$\mu(\Pi' \times \mu(\Sigma) \upharpoonright X'') \upharpoonright X'' = \mu(\Sigma) \upharpoonright X''$$

Proof

Let $\sigma \in \mu(\Sigma)$. Consider τ s.t. $\sigma \upharpoonright X'' = \tau \upharpoonright X''$, $\tau(x) = 0$ elsewhere. As $\sigma \in \mu(\Sigma)$, $\tau \in \mu(\Pi' \times \mu(\Sigma) \upharpoonright X'')$.

Conversely, let $\sigma \in \mu(\Pi' \times \mu(\Sigma) \upharpoonright X'')$, then $\sigma \upharpoonright X'' \in \mu(\Sigma) \upharpoonright X''$

The reason for both is that we look at the coordinates independently.

□

Corollary 2.11

$\Pi' \times \mu(\Sigma) \upharpoonright X''$ is a semantical interpolant for $\phi \sim \psi$.

Is it definable? See below.

Proof

Suppose it is definable by α .

$\mu(\Sigma) \subseteq \Pi' \times \mu(\Sigma) \upharpoonright X''$, so $\phi \sim \alpha$.

By prerequisite, $\mu(\Sigma) \subseteq M(\psi)$, and ψ contains no $x' \in X'$ as relevant variable, so $M(\psi)$ can be written as $\Pi' \times \Gamma$ for some $\Gamma \subseteq \Pi X''$.
 $\mu(\Pi' \times \mu(\Sigma) \upharpoonright X'') \upharpoonright X'' = \mu(\Sigma) \upharpoonright X''$, so $\mu(\Pi' \times \mu(\Sigma) \upharpoonright X'') \subseteq \Pi X' \times (\mu(\Pi' \times \mu(\Sigma) \upharpoonright X'') \upharpoonright X'') = \Pi' \times (\mu(\Sigma) \upharpoonright X'') \subseteq \Pi' \times \Gamma = M(\psi)$, as $\mu(\Sigma) \subseteq \Pi' \times \Gamma$. Thus $\alpha \sim \psi$.

2.3.4 Definability of minimal models

Perhaps we can do like for base logic.

Consider $\vee \wedge$ again, and a conjunction.

If a is only in $a \rightarrow b$ or $b \rightarrow a$: make a, b wrong ($a = b = 0$).

If a is alone: $a = 2$ (prerequisite, look at models where $\phi = 2$)

If $\neg a$ is alone: $a = 0$

If $\neg\neg a$ is alone: $FALSE = a \wedge \neg a$

If $(a \rightarrow b) \wedge a$: $a = b = 2$

If $(a \rightarrow b) \wedge \neg a$: $a = b = 0$

If $(a \rightarrow b) \wedge a \neg\neg$: $FALSE$

If $(a \rightarrow b) \wedge b$: $a = 0, b = 2$

If $(a \rightarrow b) \wedge \neg b$: $a = b = 0$

If $(a \rightarrow b) \wedge \neg\neg b$: $FALSE$

Perhaps this solves the thing.

Remark 2.12

By Fact 2.9 (page 8), EQ-models of ϕ are definable with a formula (in the finite case). The technique to go from EQ models to the interpolant is the same as for the HT logic. So the interpolant is definable, too. □

3 Countably many disjoint sets

We show here that - independent of the cardinality of the language - one can define only countably many inconsistent formulas. The question is due to D.Makinson (personal communication).

Example 3.1

There is a countably infinite set of formulas s.t. the defined model sets are pairwise disjoint.

Let $p_i : i \in \omega$ be propositional variables.

Consider $\phi_i := \bigwedge \{\neg p_j : j < i\} \wedge p_i$ for $i \in \omega$.

Obviously, $M(\phi_i) \neq \emptyset$ for all i .

Let $i < i'$, we show $M(\phi_i) \cap M(\phi_{i'}) = \emptyset$. $M(\phi_{i'}) \models \neg p_i$, $M(\phi_i) \models p_i$.

□

Fact 3.1

Any set X of consistent formulas with pairwise disjoint model sets is at most countable.

Proof

Let such X be given.

(1) We may assume that X consists of conjunctions of propositional variables or their negations.

Proof: Re-write all $\phi \in X$ as disjunctions of conjunctions ϕ_j . At least one of the conjunctions ϕ_j is consistent. Replace ϕ by one such ϕ_j . Consistency is preserved, as is pairwise disjointness.

(2) Let X be such a set of formulas. Let $X_i \subseteq X$ be the set of formulas in X with length i , i.e. a consistent conjunction of i many propositional variables or their negations, $i > 0$.

As the model sets for X are pairwise disjoint, the model sets for all $\phi \in X_i$ have to be disjoint.

(3) It suffices now to show that each X_i is at most countable, we even show that each X_i is finite.

Proof by induction:

Consider $i = 1$. Let $\phi, \phi' \in X_1$. Let ϕ be p or $\neg p$. If ϕ' is not $\neg\phi$, then ϕ and ϕ' have a common model. So one must be p , the other $\neg p$. But these are all possibilities, so $\text{card}(X_1)$ is finite.

Let the result be shown for $k < i$.

Consider now X_i . Take arbitrary $\phi \in X_i$. Wlog, $\phi = p_1 \wedge \dots \wedge p_i$. Take arbitrary $\phi' \neq \phi$. As $M(\phi) \cap M(\phi') = \emptyset$, ϕ' must be a conjunction containing one of $\neg p_k$, $1 \leq k \leq i$. Consider now $X_{i,k} := \{\phi' \in X_i : \phi' \text{ contains } \neg p_k\}$. Thus $X_i = \{\phi\} \cup \bigcup \{X_{i,k} : 1 \leq k \leq i\}$. Note that all $\psi, \psi' \in X_{i,k}$ agree on $\neg p_k$, so the situation in $X_{i,k}$ is isomorphic to X_{i-1} . So, by induction hypothesis, $\text{card}(X_{i,k})$ is finite, as all $\phi' \in X_{i,k}$ have to be mutually inconsistent. Thus, $\text{card}(X_i)$ is finite. (Note that we did not use the fact that elements from different $X_{i,k}, X_{i,k'}$ also have to be mutually inconsistent, our rough proof suffices.)

□

Note that the proof depends very little on logic. We needed normal forms, and used 2 truth values. Obviously, we can easily generalize to finitely many truth values.

4 Operations on logical structures

4.1 Introduction

In many cases, one wants more than a static structure:

- (1) dynamic theory revision a la Pearl etc.
- (2) revising a preferential logic
- (3) changing the language in interpolation
- (4) intuitionistic preferential logic: arrows are added

etc.

One can ask about such “meta-operations” for instance:

- (1) are properties preserved, e.g. is the result of working on a ranked structure again a ranked structure?
- (2) do we lose properties?
- (3) do we win new properties?
- (4) is it reasonable to require higher operators to follow the same laws as the basic operators, e.g. minimal change, and if so, e.g. minimal change of what?
- (5) what can be a structural semantics for such higher operators?

4.1.1 Conditionals

$A > B$ may mean: A becomes true (in the world), or the agent learns/believes A then: B becomes true, or the agent believes B , or the agent does B , or the agent brings B about, or so. Similarly, ternary conditionals $(A, B) > C$ can have very different meanings, and their formal properties may reflect this.

4.2 Theory revision

4.2.1 “Meta-revision”

AGM left K (A below) constant, and this may have contributed to subsequent confusion.

One sees sometimes a conditional $B > C$ expressing that after revising with B , C will hold.

But this hides the fact that it is in reality a 3-place conditional:

$(A, B) > C$: after revising A with B , C will hold, $A * B \models C$

A, B , etc. are formulas, i.e. partial information.

$(A, B) > C$ is partial information about the revision strategy, it describes just a bit of the whole picture. In the LMS tradition, see [LMS01], a revision strategy is just a distance between models. So $(A, B) > C$ describes one part of the distance.

Thus, $(A, B) > C$ is a partial revision strategy, or a set of distances which are compatible with $(A, B) > C$, just as a formula is a set of models.

But now, we have a perfect analogy:

We had $A * B \models C$ for formulas A, B, C , and now we can revise partial strategies:

$((A, B) > C, (A', B') > C') >> (A'', B'') > C''$, i.e.

if we revise the partial strategy $(A, B) > C$ with the partial strategy $(A', B') > C'$, then the new strategy gives $(A'', B'') > C''$. $>>$ is the “meta-conditional”.

It all becomes transparent, and we can iterate the whole thing as often as we want.

In the distance language, we have a set of distances on models which all satisfy $(A, B) > C$, i.e. the B -models closest to the A -models all satisfy C , and another set of distances which all satisfy $(A', B') > C'$, we revise the first with the second using a “meta-distance” (a distance on the set of all distances between models of the base language), and get a new set of distances which all satisfy $(A'', B'') > C''$. If the first two sets are consistent, i.e. there is a distance which satisfies $(A, B) > C$ and $(A', B') > C'$, then the result is the intersection of the two distance sets. This corresponds, as usual, to the respect of 0 by a distance: $d(x, y) = 0$ iff $x = y$. Of course, we can consider here special distances like variants of the Hamming distance, working on a suitable set.

Of course, just as a formula may correspond to exactly 1 model, i.e. a complete consistent theory, we may also work with $*$ (i.e. the full revision strategy) instead of with $(A, B) > C$. So we may have $(*, *) >> *$. Still, as shown in [LMS01], the distance will usually not be fully determined, so we still work with sets of distance.

Note that we can also construct mixed systems, which allow to evaluate expressions like $(A, (B, C) > D) > E$, where factual information/models are mixed with conditional structures - this might be needed e.g. for natural language. We can go as high as we want, or even go down, evaluating “on the fly”.

One problem is TR (global distances). The CFC approach will not work, as we cannot consider the left hand side individually. We have to express it via quantifiers (modal operators). We have to say $x \in X$ is such that there is $y \in Y$ s.t. for no $x' \in X$, $y' \in Y$ $d(x', y') < d(x, y)$. We use global modality. Let X be defined by ϕ , Y by ψ , then $\diamond\psi \neg(\diamond\phi \diamond\psi \langle x', y', x, y \rangle)$ where $\langle x', y', x, y \rangle$ expresses that $d(x', y') < d(x, y)$ (axiomatize suitably so it is a distance). Etc, should work?

The distance need not be defined everywhere, so it may return “unknown”.

Pearl et al. For Pearl et al. (see [DP94]), Boutilier (see [Bou94]), and Kern-Isberner (see [Ker99]), an epistemic state \mathcal{E} is a pair $(\mathcal{B}, \mathcal{C})$, where \mathcal{B} is a set of beliefs (classical formulas), and \mathcal{C} a (perhaps partial) revision strategy coded by a set of conditionals, whose elements are classical formulas. In our terminology such an conditional will have the form $(\bigwedge \mathcal{B}, A) > C$, where $\bigwedge \mathcal{B}$ is the conjunction of factual beliefs, A, C are classical formulas, expressing: On the basis of \mathcal{B} , if I were to learn A , then I would believe C .

Pearl’s criticism of the AGM approach was that revising an epistemic state by some factual information A should not only modify factual beliefs, i.e. \mathcal{B} , but also the revision strategy \mathcal{C} . Pearl et al. gave some conditions this modification of \mathcal{C} should satisfy.

In our above notation, we then have $\mathcal{E} * A = (\mathcal{B}, \mathcal{C}) * A = (\mathcal{B}', \mathcal{C}') = \mathcal{E}'$, where \mathcal{C} determines the modification of \mathcal{B} to \mathcal{C} (by the conditionals $A > X$), but *not* the modification of \mathcal{C} to \mathcal{C}' .

Boutilier, see [Bou94], and Kern-Isberner, see [Ker99], extended this idea to revising epistemic states not only by factual information, but also by conditional information. (Kern-Isberner codes factual information X by the conditional $TRUE > X$, and thus avoids a distinction between the two.)

So we have $\mathcal{E} * C = (\mathcal{B}, \mathcal{C}) * (X > Y) = (\mathcal{B}', \mathcal{C}') = \mathcal{E}'$.

Revising by the factual conditional $X > Y$ imposes restrictions also on the transformation of \mathcal{C} to \mathcal{C}' .

4.3 Preferential systems

4.3.1 Why does \sim not modify itself?

We saw in Section 4.2 (page 11) how Pearl et al. introduced a revision operator $*$ whose application changes (the conditional part of) $*$ itself.

The question is obvious: Is there a logical formalism \sim which, applied to some formula ϕ , will not only produce a consequence ψ , but also a new logic \sim' ?

To the authors’ knowledge, this does not exist.

We may use Gabbay’s idea of reactivity to build such a logic: applying the logic changes it - this would give a formal motivation to the enterprise, from the other side, so to say.

In more detail: simple arrows obey the fundamental law of preferential structures. Adding higher arrows allows us to restrict from above, and thus describe any set, in a static way, see [GS08b]. So we can describe $M(\phi)$. When we “activate” now $M(\phi)$, we use higher arrows to modify the basic preferential relation.

4.3.2 (Meta) Operations on logics

It is natural to consider the operations of deduction and revision on logical systems.

Given some logical system \sim , we might deduce a new logic \sim' from it and some formula ϕ , i.e. $\phi \sim (\psi, \sim')$. E.g., \sim' might be weaker than \sim (this corresponds to classical logic, which is weakening), or we might deduce a new, bolder logic, corresponding to a more daring reasoning (this corresponds to non-monotonic logic, where we go beyond classical logic, win more conclusions, at the price of less certainty).

Perhaps even more useful, we may see that our logic does not give the desired conclusions, and may want to revise it, by some minimal change which obtains the desired result.

As in the case of theory revision, we can take as arguments the whole logic \sim , or just one or some pairs (ϕ, ψ) with $\phi \sim \psi$. For instance, we might want to revise \sim with some new pair (ϕ, ψ) , and see whether $\phi' \sim' \psi'$ holds in the new logic \sim' .

4.3.3 Implementation

Usually, working on the semantic side is easier. There are different ways to do it.

- (1) We can work with canonical structures (if they exist) - this may generate different results when we consider different structures as canonical. (This was a problem with [ALS98-1].)
- (2) We can work with the set of all structures corresponding to the logic, e.g. all preferential structures generating the logic
- (3) We can work with the algebraic semantics, i.e. usually with the smallest set of the filter, corresponding to $\mu(\phi)$, the set of minimal models of ϕ .
- (4) The reactive idea was carried out in [GS08b], where we modified preferential structures by adding higher order arrows. There, the view was static, but we can turn it dynamic to achieve revision and “meta-logic”.

Algebraic semantics As the algebraic semantics usually is the most robust notion, this is perhaps the easiest to work with. For each ϕ , $\mu(\phi)$ is defined. So a natural distance between \sim and \sim' is the set of ϕ where $\mu(\phi) \neq \mu'(\phi)$, and for each such ϕ the symmetrical set distance between $\mu(\phi)$ and $\mu'(\phi)$. This gives a distance based revision of \sim to \sim' .

For a “meta-logic”, we can as usual consider a preference relation between logics (which are now simple objects, just as classical models, given by their $\mu(\phi)$ for all ϕ), and work with the algebraic representation results of the second author, see e.g. [Sch04].

Structural semantics It is natural to define a distance between two preferential structures by looking at the arrow sets, or sets of pairs (m, m') , such that $m < m'$. Again, some Hamming distance would be a first answer.

We treated one technique of modifying general (and smooth) preferential structures in [GS08b].

4.3.4 Operations on linear and ranked structures

Making a linear or ranked structure simply reactive will usually result in a mess, where the central properties of such structures are destroyed.

It seems more reasonable to investigate operations which leave the structure more intact, and postpone questions about their realization.

We may consider here operations which

- (1) cut the linear or ranked structure in two parts, such that within in each part the structure stays as it was, and the two parts are incomparable (they look a bit like a tree trunk, which was cut with a saw in 2 parts)
- (2) do elementary exchange operations (permutations) in the case of linear orders.
- (3) for ranked structures, we may have an operation $\alpha(x, y)$, which puts x on y 's level, $\beta(x, y)$ which changes the levels of x and y , etc.

4.3.5 Intuitionistic preferential logic

We are not sure about all arrows. Some arrows are definitely there, others definitely out, some come and go.

We have successively better information about arrows, and thus about size. Int. rules about size.

The following seems new: We do not only have $\Box(\phi \sim \psi)$, but also $\Box\neg(\phi \sim \psi)$. This has to be treated, especially for ranked structures.

4.4 Non-monotonic interpolation

We investigated non-monotonic interpolation in [GS09c].

The main property needed can be summarized as follows:

Let X be a set, and $X' \cup X'' = X$ be a disjoint cover of X . Consider $\Sigma \subseteq \Pi X$. Suppose $\mu(\Sigma) \subseteq \Sigma'$, where the variables defining Σ' are all in X'' . We now have to consider $\mu(\Pi X' \times \Sigma'')$, where $\Sigma'' = \Sigma \upharpoonright X''$, the restriction of Σ to X'' . We want $\mu(\Pi X' \times \Sigma'') \subseteq \Sigma'$, this gives the desired interpolation.

Formally:

$$(\mu * 3) \mu(\Pi' \times \Sigma'') \upharpoonright X'' \subseteq \mu(\Sigma) \upharpoonright X''.$$

The point here is that, logically, only the X'' -part matters, as Σ' is the full product on X' : $\Sigma' = \Pi X' \times (\Sigma' \upharpoonright X'')$. When we go from $\mu(\Sigma)$ to $\mu(\Delta)$, where $\Delta := \Pi X' \times \Sigma''$, Δ is bigger than Σ in the X' -part, and identical in the X'' -part. So increasing Σ outside X'' does not increase $\mu(\Sigma)$ inside X'' .

The (set variant) of the Hamming order satisfies this property: If $\sigma = \sigma' \circ \sigma''$, $\tau = \tau' \circ \tau''$ (\circ is concatenation), then $\sigma \prec \tau$ iff $\sigma' \prec \tau'$ and $\sigma'' \prec \tau''$. Thus, if e.g. $\sigma, \tau \in \Sigma$, $\sigma'' \prec \tau''$, but $\tau' \prec \sigma'$, then τ may be minimal, but adding $\rho' \circ \sigma''$ with $\rho' \prec \tau'$ will eliminate τ .

This behaviour motivates the following reflections:

- (1) This condition $(\mu * 3)$ points to a weakening of the Hamming condition:

Adding new “branches” in X' will not give new minimal elements in X'' , but may destroy other minimal elements in X'' . This can be achieved by a sort of semi-rankedness: If ρ and σ are different only in the X' -part, then $\tau \prec \rho$ iff $\tau \prec \sigma$, but not necessarily $\rho \prec \tau$ iff $\sigma \prec \tau$.

- (2) In more abstract terms:

When we separate support from attack (support: a branch σ' in X' supports a continuation σ'' in X'' iff $\sigma \circ \sigma''$ is minimal, i.e. not attacked, attack: a branch τ in X' attacks a continuation σ'' in X'' iff it prevents all $\sigma \circ \sigma''$ to be minimal), we see that new branches will not support any new continuations, but may well attack continuations.

More radically, we can consider paths σ'' as positive information, σ' as potentially negative information. Thus, Π' gives maximal negative information, and thus smallest set of accepted models.

The concept of size looks only at the result of support and attack, so it is necessarily somewhat coarse. Future research should also investigate both concepts separately.

- (3) We can interpret this as follows:

- (1) X'' determines the base set.

- (2) X' is the context. The context determines the choice (i.e. a subset of the base set).

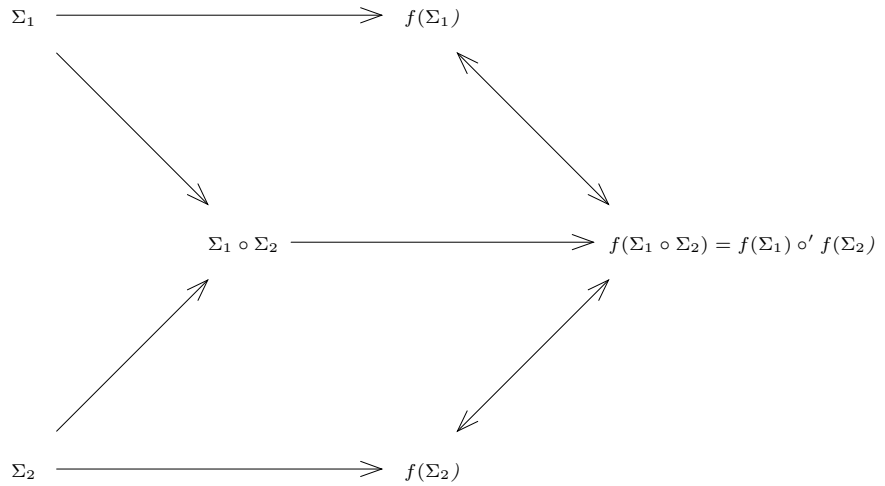
- (3) When we compare this to preferential structures, we see that also in preferential structures the bigger the set, the more attacks are possible.

We broaden these considerations:

- (1) Following a tradition begun by Kripke, one has added structure to the set of classical models, reachability, preference, etc. Perhaps one should emphasize a more abstract approach, in the line of [Sch92], and elaborated in [Sch04], see in particular the distinction between structural and algebraic semantics in the latter. So we should separate structure from logic in the semantics, and treat what we called context above by a separate “machinery”. Thus, given a set X of models, we have some abstract function f , which chooses the models where the consequences hold, $f(X)$.
- (2) Now, we can put into this “machinery” whatever we want, e.g. the abstract choice function of preferential structures.
- (3) But we can also investigate non-static f , where f changes in function of what we already did - “reacting” to the past.
- (4) We can also look at properties of f , like complexity, generation by some simple structure like a simple automaton, etc.
- (5) So we advocate the separation of usual, classical semantics, from the additional properties, which are treated “outside”.

5 Independence and multiplication of abstract size

5.1 Definition of independence



Note that \circ and \circ' might be different

Diagram 5.1

Independence

The right notion of independence seems to be:

We have composition \circ , and operation f . We can calculate $f(\Sigma_1 \circ \Sigma_2)$ from $f(\Sigma_1)$ and $f(\Sigma_2)$, but also conversely, given $f(\Sigma_1 \circ \Sigma_2)$ we can calculate $f(\Sigma_1)$ and $f(\Sigma_2)$.

Definition 5.1

Let $f : \mathcal{D} \rightarrow \mathcal{C}$ be any function from domain \mathcal{D} to co-domain \mathcal{C} . Let \circ be a “composition function” $\circ : \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$, likewise for $\circ' : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$.

We say that $\langle f, \circ, \circ' \rangle$ are independent iff for any $\Sigma_i \in \mathcal{D}$

- (1) $f(\Sigma_1 \circ \Sigma_2) = f(\Sigma_1) \circ' f(\Sigma_2)$,
- (2) we can recover $f(\Sigma_i)$ from $f(\Sigma_1 \circ \Sigma_2)$, provided we know how $\Sigma_1 \circ \Sigma_2$ splits into Σ_i .

5.1.1 Discussion

- (1) Ranked structures satisfy it:

Let $\circ = \circ' = \cup$. Let f be the minimal model operator μ of preferential logic. Let $X, Y \subseteq X \cup Y$ have (at least) medium size. Then $\mu(X \cup Y) = \mu(X) \cup \mu(Y)$, and $\mu(X) = \mu(X \cup Y) \cap X$, $\mu(Y) = \mu(X \cup Y) \cap Y$.

- (2) Consistent classical formulas and their interpretation satisfy it:

Let \circ be conjunction in the composed language, \circ' be model set intersection, $f(\phi) = M(\phi)$. Let ϕ, ψ be classical formulas, defined on disjoint language fragments $\mathcal{L}, \mathcal{L}'$ of some language \mathcal{L}'' . Then $f(\phi \wedge \psi) = M(\phi) \cap M(\psi)$, and $M(\phi)$ is the projection of $M(\phi) \cap M(\psi)$ onto the (models of) language \mathcal{L} , likewise for $M(\psi)$. This is due to the way validity is defined, using only variables which occur in the formula.

As a consequence, monotonic logic has interpolation - see [GS09c].

- (3) It does not hold for inconsistent classical formulas: We cannot recover $M(a \wedge \neg a)$ and $M(b)$ from $M(a \wedge \neg a \wedge b)$, as we do not know where the inconsistency came from. The basic reason is trivial: One empty factor suffices to make the whole product empty, and we do not know which factor was the culprit. See Section 5.3.5 (page 20) for the discussion of a remedy.
- (4) Preferential logic satisfies it under certain conditions:
If $\mu(X \times Y) = \mu(X) \times \mu(Y)$ holds for model products and \vdash , then it holds by definition. An important consequence is that such a logic has interpolation of the form $\vdash \circ \vdash$, see Section 5.3.4 (page 19).
- (5) Modular revision a la Parikh is based on a similar idea.

5.1.2 Independence and multiplication of abstract size

We are mainly interested in nonmonotonic logic. In this domain, independence seems to be strongly connected to multiplication of abstract size, and much of the present paper treats this connection and its repercussions.

We have at least two scenarios for multiplication, one is decribed in Diagram 5.2 (page 15), the second in Diagram 5.3 (page 16). In the first scenario, we have nested sets, in the second, we have set products. In the first scenario, we consider subsets which behave as the big set does, in the second scenario we consider subspaces, and decompose the behaviour of the big space into behaviour of the subspaces. In both cases, this results naturally in multiplication of abstract sizes. When we look at the corresponding relation properties, they are quite different (rankedness vs. some kind of modularity). But this is perhaps to be expected, as the two scenarios are quite different.

We do not know whether there are still other, interesting scenarios to consider in our framework.

5.2 Multiplication of size for subsets

Here we have nested sets, $X \subseteq Y \subseteq Z$, X is a certain proportion of Y , and Y of Z , resulting in a multiplication of relative size or proportions. This is a classical subject of nonmonotonic logic, see e.g. [GS09a], it is partly repeated here only to stress the common points with the other scenario.

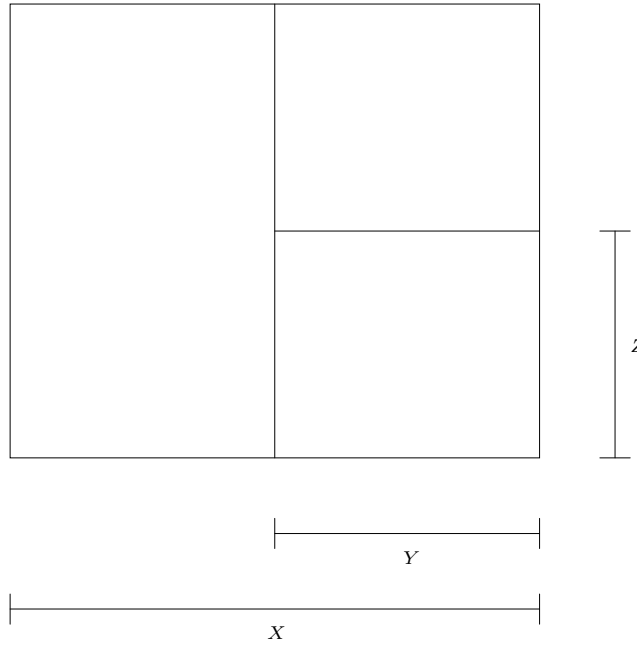


Diagram 5.2

Scenario 1

5.2.1 Properties

Diagram 5.2 (page 15) is to be read as follows: The whole set X is split in Y and $X-Y$, Y is split in Z and $Y-Z$. Y is a small/medium/big part of X , Z is a small/medium/big part of Y . The question is: is Z a small/medium/big part of X ?

Note that the relation of Z to Y is conceptually different from that of Y to Z , as we change the base set by going from Y to Z , but not when going from Z to Y . Thus, in particular, when we read the diagram as expressing multiplication, commutativity is not necessarily true.

We looked at this scenario already in [GS09a], but there from an additive point of view, using various basic properties like (iM), (eMT), (eMF). Here, we use just multiplication - except sometimes for motivation.

We examine different rules:

If $X = Y$ or $Y = Z$, there is nothing to show, so 1 is the neutral element of multiplication.

If $Y \in \mathcal{I}(X)$ or $Z \in \mathcal{I}(Y)$, then we should have $Z \in \mathcal{I}(X)$. (Use for motivation (iM) or (eMT) respectively.)

So it remains to look at the following cases, with the “natural” answers given already:

- (1) $Y \in \mathcal{F}(X)$, $Z \in \mathcal{F}(Y) \Rightarrow Z \in \mathcal{F}(X)$,
- (2) $Y \in \mathcal{M}^+(X)$, $Z \in \mathcal{F}(Y) \Rightarrow Z \in \mathcal{M}^+(X)$,
- (3) $Y \in \mathcal{F}(X)$, $Z \in \mathcal{M}^+(Y) \Rightarrow Z \in \mathcal{M}^+(X)$,
- (4) $Y \in \mathcal{M}^+(X)$, $Z \in \mathcal{M}^+(Y) \Rightarrow Z \in \mathcal{M}^+(X)$.

But (1) is case (3) of (\mathcal{M}_ω^+) in [GS09a],

- (2) is case (2) of (\mathcal{M}_ω^+) there,
- (3) is case (1) of (\mathcal{M}_ω^+) there, finally,
- (4) is (\mathcal{M}^{++}) there.

So the first three correspond to various expressions of AND/OR/CM, and the last one to RatM.

But we can read them also the other way round, e.g.:

- (1) corresponds to: $\alpha \sim \beta, \alpha \wedge \beta \vdash \gamma \Rightarrow \alpha \vdash \gamma,$
- (2) corresponds to: $\alpha \not\sim \neg\beta, \alpha \wedge \beta \vdash \gamma \Rightarrow \alpha \not\sim \neg\beta \vee \neg\gamma,$
- (3) corresponds to: $\alpha \sim \beta, \alpha \wedge \beta \not\sim \neg\gamma \Rightarrow \alpha \not\sim \neg\beta \vee \neg\gamma.$

All these rules might be seen as too idealistic, so just as we did in [GS09a], we can consider milder versions: We might for instance consider a rule which says that $big * \dots * big$, n times is not small, big^n is not small. Consider for instance the case $n = 2$. So we would conclude that Z is not small in X . In terms of logic, we then have: $\alpha \sim \beta, \alpha \wedge \beta \vdash \gamma \Rightarrow \alpha \not\sim (\neg\beta \vee \neg\gamma)$. We can obtain the same logic property from $3 * small \neq all$.

Fact 5.1

$Small + small = small$ plus $1 * big = big$ entails $big * big = big$

Proof

$$1 * big \cap big * 1 = big * big \quad \square$$

5.2.2 Corresponding preferential relations

It is well-known that rankedness corresponds to the rule (\mathcal{M}^{++}) , see, e.g., [GS09a], and we will not discuss this here any further.

5.3 Multiplication of size for subspaces

5.3.1 Properties

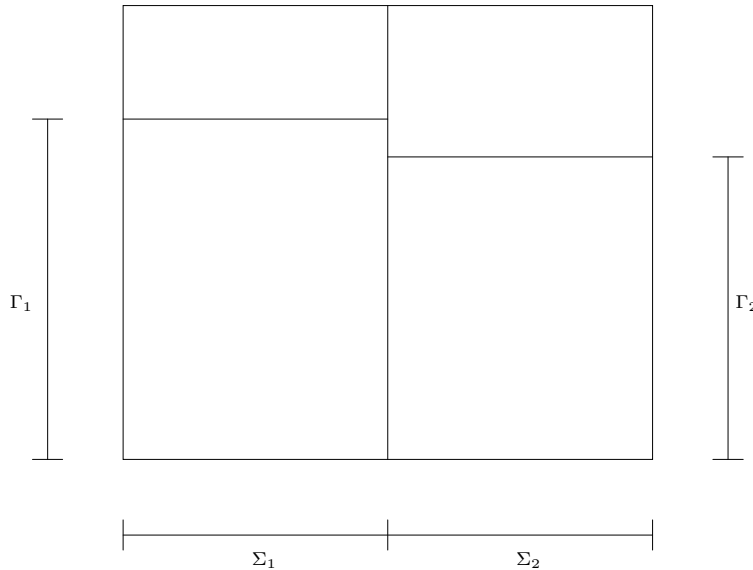


Diagram 5.3

Scenario 2

In this scenario, Σ_i are sets of sequences. (Corresponding, intuitively, to a set of models in language \mathcal{L}_i , Σ_i will be the set of α_i -models, and the subsets Γ_i are to be seen as the “best” models, where β_i will hold. The languages are supposed to be disjoint sublanguages of a common language \mathcal{L} .)

In this scenario, the Σ_i have symmetrical roles, so there is no intuitive reason for multiplication not to be commutative.

We can interpret the situation twofold:

First, we work separately in sublanguage \mathcal{L}_1 and \mathcal{L}_2 , and, say, α_1 and β_1 are both defined in \mathcal{L}_1 , and we look at $\alpha_1 \sim \beta_1$ in the sublanguage \mathcal{L}_1 , or, we consider both α_1 and β_1 in the common language \mathcal{L} , and look at $\alpha_1 \sim \beta_1$ in \mathcal{L} . These two ways are a priori completely different. Speaking in preferential terms, it is not at all clear why the orderings on the submodels should have anything to do with the orderings on the whole models. It seems a very desirable property, but we have to postulate it, which we do now:

(*big * 1 = big*) Let $\Gamma_1 \subseteq \Sigma_1$, then $\Gamma_1 \times \Sigma_2 \in \mathcal{F}(\Sigma_1 \times \Sigma_2)$ iff $\Gamma_1 \in \mathcal{F}(\Sigma_1)$, (and the dual rule for Σ_2 and Γ_2).

This property preserves proportions, so it seems intuitively quite uncontested, whenever we admit coherence over products. (Recall that there was nothing to show in the first scenario.)

When we re-consider above case: suppose $\alpha \sim \beta$ in the sublanguage, so $M(\beta) \in \mathcal{F}(M(\alpha))$, so by (*big * 1 = big*) read from right to left, $M(\beta) \in \mathcal{F}(M(\alpha))$ in the big language \mathcal{L} .

We obtain the dual rule for small (and likewise, medium size) sets:

(*small * 1 = small*) Let $\Gamma_1 \subseteq \Sigma_1$, then $\Gamma_1 \times \Sigma_2 \in \mathcal{I}(\Sigma_1 \times \Sigma_2)$ iff $\Gamma_1 \in \mathcal{I}(\Sigma_1)$, (and the dual rule for Σ_2 and Γ_2), establishing *All* as the neutral element for multiplication.

We look now at other, plausible rules:

(*small * x = small*) $\Gamma_1 \in \mathcal{I}(\Sigma_1), \Gamma_2 \subseteq \Sigma_2 \Rightarrow \Gamma_1 \times \Gamma_2 \in \mathcal{I}(\Sigma_1 \times \Sigma_2)$

(*big * big = big*) $\Gamma_1 \in \mathcal{F}(\Sigma_1), \Gamma_2 \in \mathcal{F}(\Sigma_2) \Rightarrow \Gamma_1 \times \Gamma_2 \in \mathcal{F}(\Sigma_1 \times \Sigma_2)$

(*big * medium = medium*) $\Gamma_1 \in \mathcal{F}(\Sigma_1), \Gamma_2 \in \mathcal{M}(\Sigma_2) \Rightarrow \Gamma_1 \times \Gamma_2 \in \mathcal{M}(\Sigma_1 \times \Sigma_2)$

(*medium * medium = medium*) $\Gamma_1 \in \mathcal{M}^+(\Sigma_1), \Gamma_2 \in \mathcal{M}^+(\Sigma_2) \Rightarrow \Gamma_1 \times \Gamma_2 \in \mathcal{M}^+(\Sigma_1 \times \Sigma_2)$

When we accept all, we can invert (*big * big*), as a big product must be composed of big components. Likewise, at least one component of a small product has to be small.

We see that these properties give a lot of modularity. We can calculate the consequences of α and α' separately - provided α, α' use disjoint alphabets - and put the results together afterwards. Such properties are particularly interesting for classification purposes, where subclasses are defined with disjoint alphabets.

We quote from [GS09c] the following pairwise equivalent conditions for a principal filter generated by μ :

Definition 5.2

(*S * 1*) $\Delta \subseteq \Sigma' \times \Sigma''$ is big iff there is $\Gamma = \Gamma' \times \Gamma'' \subseteq \Delta$ s.t. $\Gamma' \subseteq \Sigma'$ and $\Gamma'' \subseteq \Sigma''$ are big

(*$\mu * 1$*) $\mu(\Sigma' \times \Sigma'') = \mu(\Sigma') \times \mu(\Sigma'')$

(*S * 2*) $\Gamma \subseteq \Sigma$ is big $\Rightarrow \Gamma \upharpoonright X' \subseteq \Sigma \upharpoonright X'$ is big - where Σ is not necessarily a product.

(*$\mu * 2$*) $\mu(\Sigma) \subseteq \Gamma \Rightarrow \mu(\Sigma \upharpoonright X') \subseteq \Gamma \upharpoonright X'$

5.3.2 Corresponding preferential relations

Definition 5.3

Call a relation \prec a GH (= general Hamming) relation iff the following two conditions hold:

(GH1) $\sigma \preceq \tau \wedge \sigma' \preceq \tau' \wedge (\sigma \prec \tau \vee \sigma' \prec \tau') \Rightarrow \sigma\sigma' \prec \tau\tau'$

(where $\sigma \preceq \tau$ iff $\sigma \prec \tau$ or $\sigma = \tau$)

(GH2) $\sigma\sigma' \prec \tau\tau' \Rightarrow \sigma \prec \tau \vee \sigma' \prec \tau'$

(GH2) means that some compensation is possible.

We use (GH) for (*GH1*) + (*GH2*).

Example 5.1

The following are example of GH relations:

Define on all components X_i a relation \prec_i .

(1) The set variant Hamming relation:

Then the relation \prec defined on $\Pi\{X_i : i \in I\}$ by $\sigma \prec \tau$ iff there is at least one i s.t. $\sigma_i \prec_i \tau_i$ and for all other $j \neq i$ $\sigma_j \preceq_j \tau_j$.

(2) The counting variant Hamming relation:

Then the relation \prec defined on $\Pi\{X_i : i \in I\}$ by $\sigma \prec \tau$ iff the number of i such that $\sigma_i \prec_i \tau_i$ is bigger than the number of i such that $\tau_i \prec_i \sigma_i$.

(3) The weighed counting Hamming relation:

Like the counting relation, but we give different (numerical) importance to different i . E.g., $\sigma_1 \prec \tau_1$ may count 1, $\sigma_2 \prec \tau_2$ may count 2, etc.

□

Proposition 5.2

Let $\sigma \prec \tau \Leftrightarrow \tau \notin \mu(\{\sigma, \tau\})$ and \prec be smooth. Then μ satisfies (*$\mu * 1$*) iff \prec is a GH relation.

Proof

(1) $(\mu * 1)$ entails the GH relation conditions

(GH1): Suppose $\sigma \prec \tau$ and $\sigma' \preceq \tau'$. Then $\mu(\{\sigma, \tau\}) = \{\sigma\}$, and $\mu(\{\sigma', \tau'\}) = \{\sigma'\}$ (either $\sigma' \prec \tau'$ or $\sigma' = \tau'$, so in both cases $\mu(\{\sigma', \tau'\}) = \{\sigma'\}$). As $\tau \notin \mu(\{\sigma, \tau\})$, $\tau\tau' \notin \mu(\{\sigma, \tau\} \times \{\sigma', \tau'\}) =_{(\mu * 1)} \mu(\{\sigma, \tau\}) \times \mu(\{\sigma', \tau'\}) = \{\sigma\} \times \{\sigma'\} = \{\sigma\sigma'\}$, so by smoothness $\sigma\sigma' \prec \tau\tau'$.

(GH2): Let $X := \{\sigma, \tau\}$, $Y := \{\sigma', \tau'\}$, so $X \times Y = \{\sigma\sigma', \sigma\tau', \tau\sigma', \tau\tau'\}$. Suppose $\sigma\sigma' \prec \tau\tau'$, so $\tau\tau' \notin \mu(X \times Y) =_{(\mu * 1)} \mu(X) \times \mu(Y)$. If $\sigma \not\prec \tau$, then $\tau \in \mu(X)$, likewise if $\sigma' \not\prec \tau'$, then $\tau' \in \mu(Y)$, so $\tau\tau' \in \mu(X \times Y)$, contradiction.

(2) The GH relation conditions generate $(\mu * 1)$.

$\mu(X \times Y) \subseteq \mu(X) \times \mu(Y)$: Let $\tau \in X$, $\tau' \in Y$, $\tau\tau' \notin \mu(X) \times \mu(Y)$, then $\tau \notin \mu(X)$ or $\tau' \notin \mu(Y)$. Suppose $\tau \notin \mu(X)$, let $\sigma \in X$, $\sigma \prec \tau$, so by condition (GH1) $\sigma\tau' \prec \tau\tau'$, so $\tau\tau' \notin \mu(X \times Y)$.

$\mu(X) \times \mu(Y) \subseteq \mu(X \times Y)$: Let $\tau \in X$, $\tau' \in Y$, $\tau\tau' \notin \mu(X \times Y)$, so there is $\sigma\sigma' \prec \tau\tau'$, $\sigma \in X$, $\sigma' \in Y$, so by (GH2) either $\sigma \prec \tau$ or $\sigma' \prec \tau'$, so $\tau \notin \mu(X)$ or $\tau' \notin \mu(Y)$, so $\tau\tau' \notin \mu(X) \times \mu(Y)$.

□

Fact 5.3

Let $\Gamma \subseteq \Sigma$, $\Gamma' \subseteq \Sigma'$, $\Gamma \times \Gamma' \subseteq \Sigma \times \Sigma'$ be small, let (GH2) hold, then $\Gamma \subseteq \Sigma$ is small or $\Gamma' \subseteq \Sigma'$ is small.

Proof

Suppose $\Gamma \subseteq \Sigma$ is not small, so there is $\gamma \in \Gamma$ and no $\sigma \in \Sigma$ with $\sigma \prec \gamma$. Fix this γ . Consider $\{\gamma\} \times \Gamma'$. As $\Gamma \times \Gamma' \subseteq \Sigma \times \Sigma'$ is small, there is for each $\gamma\gamma'$, $\gamma' \in \Gamma'$ some $\sigma\sigma' \in \Sigma \times \Sigma'$, $\sigma\sigma' \prec \gamma\gamma'$. By (GH2) $\sigma \prec \gamma$ or $\sigma' \prec \gamma'$, but $\sigma \prec \gamma$ was excluded, so for all $\gamma' \in \Gamma'$ there is $\sigma' \in \Sigma'$ with $\sigma' \prec \gamma'$, so $\Gamma' \subseteq \Sigma'$ is small. □

Fact 5.4

Let $\Gamma \subseteq \Sigma$ be small, $\Gamma' \subseteq \Sigma'$, let (GH1) hold, then $\Gamma \times \Gamma' \subseteq \Sigma \times \Sigma'$ is small.

Proof

Let $\gamma \in \Gamma$, so there is $\sigma \in \Sigma$ and $\sigma \prec \gamma$. By (GH1), for any $\gamma' \in \Gamma'$ $\sigma\gamma' \prec \gamma\gamma'$, so no $\gamma\gamma' \in \Gamma \times \Gamma'$ is minimal. □

Proposition 5.5

$\Gamma \times \Gamma' \subseteq \Sigma \times \Sigma'$ is small iff at least one of $\Gamma \subseteq \Sigma$, $\Gamma' \subseteq \Sigma'$ is small entails (GH1) and (GH2).

Proof

(GH1): Let $\sigma\tau, \sigma' \preceq \tau'$, we have to show $\sigma\sigma' \prec \tau\tau'$. $\sigma \prec \tau \Rightarrow \{\tau\} \subseteq \{\sigma, \tau\}$ is small $\Rightarrow \{\tau\} \times \{\sigma', \tau'\} \subseteq \{\sigma, \tau\} \times \{\sigma', \tau'\}$ is small, so some element has to be smaller than $\tau\tau'$, by smoothness, there has to be a minimal element smaller than $\tau\tau'$, so $\sigma\sigma' \prec \tau\tau'$ or $\sigma\tau' \prec \tau\tau'$. Case 1: $\sigma' = \tau'$. Then $\sigma\sigma' \prec \tau\tau'$. Case 2: $\sigma' \prec \tau'$. Then $\{\sigma, \tau\} \times \{\tau'\}$ is small, so $\sigma\tau'$ is not minimal. Thus, again by smoothness, $\sigma\sigma' \prec \tau\tau'$.

(GH2): Let $\sigma\sigma' \prec \tau\tau'$, we have to show $\sigma \prec \tau$ or $\sigma' \prec \tau'$. Suppose $\sigma \not\prec \tau$, $\sigma' \not\prec \tau'$, then $\{\tau\} \subseteq \{\sigma, \tau\}$ is not small, $\{\tau'\} \subseteq \{\sigma', \tau'\}$ is not small, so $\{\tau\tau'\} \subseteq \{\sigma\tau\} \times \{\sigma'\tau'\}$ is not small, so $\sigma\sigma' \not\prec \tau\tau'$, contradiction.

□

Definition 5.4

For a relation \prec on $\Sigma \times \Sigma$ we define:

$(pr(\prec)) \sigma\sigma' \prec \tau\tau' \Rightarrow \sigma \prec \tau$ and $\sigma' \prec \tau'$

Fact 5.6

Connect $(pr(\prec))$ to $(\mu * 1)$ and $(\mu * 2)$, perhaps just quote from SIN?

Proof**Note**

Note that already $(\mu * 1)$ results in a strong independence result in the second scenario: Let $\sigma\rho' \prec \tau\rho'$, then $\sigma\rho'' \prec \tau\rho''$ for all ρ'' . Thus, whether $\{\rho''\}$ is small, or medium size (i.e. $\rho'' \in \mu(\Sigma'')$), the behaviour of $\Sigma \times \{\rho''\}$ is the same. This we do not have in the first scenario, as small sets may behave very differently from medium size sets. (But, still, their internal structure is the same, only the minimal elements change.) When $(\mu * 2)$ holds, then if $\sigma\sigma' \prec \tau\tau'$ and $\sigma \neq \tau$, then $\sigma \prec \tau$, i.e. we need not have $\sigma' = \tau'$.

5.3.3 Language change

Independence of language fragments gives us the following perspectives:

- (1) it makes independent and parallel treatment of fragments possible, and offers thus efficient treatment in applications (descriptive logics etc.).
- (2) it results in new rules similar to the classical ones like AND, OR, Cumulativity, etc. We can thus obtain postulates about reasonable behaviour, but also classification by those rules.
- (3) it sheds light on notions like “ceteris paribus”, which we saw in the context of obligations, see [GS08g].
- (4) it clarifies notions like “normal with respect to ϕ , but not ψ ”
- (5) it helps to understand e.g. inheritance diagrams where arrows make other information accessible, and we need an underlying mechanism to combine bits of information.

5.3.4 Interpolation

Proposition 5.7

$(\mu * 1)$ entails interpolation of the form $\phi \sim \alpha \sim \psi$.

Proof

Let the product be defined on $J \cup J' \cup J''$ (i.e., $J \cup J' \cup J''$ is the set of propositional variables in the intended application). See Diagram 5.4 (page 19).

We abuse notation and write $\phi \sim \Sigma$ if $\mu(\phi) \subseteq \Sigma$. As usual, $\mu(\phi)$ abbreviates $\mu(M(\phi))$.

For clarity, even if it clutters up notation, we will be precise about where μ is formed. Thus, we write $\mu_{J \cup J' \cup J''}(X)$ when we take the minimal elements in the full product, $\mu_J(X)$ when we consider only the product on J , etc.

Let $\phi \sim \psi$, i.e., $\mu_{J \cup J' \cup J''}(\phi) \subseteq M(\psi)$, ϕ be defined on $J' \cup J''$, ψ on $J \cup J'$.

As $\mu_{J \cup J' \cup J''}(\phi) \subseteq X_J \times (\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J') \times X_{J''}$, $\phi \sim X_J \times (\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J') \times X_{J''}$.

We show $\mu_{J \cup J' \cup J''}(X_J \times (\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J') \times X_{J''}) \subseteq M(\psi)$.

(1) As $M(\phi) = X_J \times M(\phi) \upharpoonright (J' \cup J'')$, $\mu_{J \cup J' \cup J''}(\phi) = \mu_J(X_J) \times \mu_{J' \cup J''}(M(\phi) \upharpoonright (J' \cup J''))$ by $(\mu * 1)$.

(2) By $(\mu * 1)$ again, $\mu_{J \cup J' \cup J''}(X_J \times \mu_{J \cup J' \cup J''}(\phi) \upharpoonright J' \times X_{J''}) \subseteq \mu_J(X_J) \times \mu_{J'}((\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J')) \times \mu_{J''}(X_{J''})$.

So it suffices to show $\mu_J(X_J) \times \mu_{J'}(\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J') \times \mu_{J''}(X_{J''}) \models \psi$.

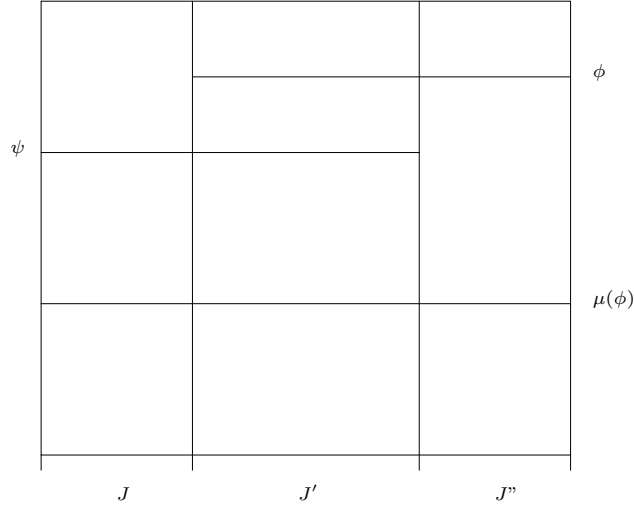
Proof: Let $\sigma = \sigma_J \sigma_{J'} \sigma_{J''} \in \mu_J(X_J) \times \mu_{J'}(\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J') \times \mu_{J''}(X_{J''})$, so $\sigma_J \in \mu_J(X_J)$.

By definition and $\mu_{J'}(\mu_{J \cup J' \cup J''}(\phi) \upharpoonright J') \subseteq \mu_{J \cup J' \cup J''}(\phi) \upharpoonright J'$, there is $\sigma' = \sigma'_J \sigma'_{J'} \sigma'_{J''} \in \mu_{J \cup J' \cup J''}(\phi)$ s.t. $\sigma'_{J'} = \sigma_{J'}$, i.e. $\sigma' = \sigma'_J \sigma_{J'} \sigma'_{J''}$. As $\sigma' \in \mu_{J \cup J' \cup J''}(\phi)$, $\sigma' \models \psi$.

By (1) and $\sigma_J \in \mu_J(X_J)$ also $\sigma_J \sigma_{J'} \sigma'_{J''} \in \mu_{J \cup J' \cup J''}(\phi)$, so also $\sigma_J \sigma_{J'} \sigma'_{J''} \models \psi$.

But ψ does not depend on J'' , so also $\sigma = \sigma_J \sigma_{J'} \sigma_{J''} \models \psi$.

□



Non-monotonic interpolation

Diagram 5.4

Remarks for the converse: from interpolation to $(\mu * 1)$

Example 5.2

We show here in (1) and (2) that half of the condition $(\mu * 1)$ is not sufficient for interpolation, and in (3) that interpolation may hold, even if $(\mu * 1)$ fails. When looking closer, the latter is not surprising: μ of sub-products may be defined in a funny way, which has nothing to do with the way μ on the big product is defined.

Consider the language based on p, q, r .

For (1) and (2) define the order $<$ on sequences of length 3 by $\neg p\neg q\neg r < p\neg q\neg r$, leave all other 3-sequences incomparable.

Let $\phi = \neg q \wedge \neg r$, $\psi = \neg p \wedge \neg q$, so $\mu(\phi) = \neg p \wedge \neg q \wedge \neg r$, and $\phi \vdash \psi$. Suppose there is α , $\phi \vdash \alpha \vdash \psi$, α written with q only, so α is equivalent to FALSE, TRUE, q , or $\neg q$. $\phi \not\vdash \text{FALSE}$, $\phi \not\vdash q$. $\text{TRUE} \not\vdash \psi$, $\neg q \not\vdash \psi$. Thus, there is no such α , and \vdash has no interpolation. We show in (1) and (2) that we can make both directions of $(\mu * 1)$ true separately, so they do not suffice to obtain interpolation.

(1) We make $\mu(X \times Y) \subseteq \mu(X) \times \mu(Y)$ true, but not the converse.

Do not order any sequences of length 2 or 1, i.e. μ is there always identity. Thus, $\mu(X \times Y) \subseteq X \times Y = \mu(X) \times \mu(Y)$ holds trivially.

For (2) and (3), consider the following ordering $<$ between sequences: $\sigma < \tau$ iff there is $\neg x$ is σ , x in τ , but for no y y in σ , $\neg y$ in τ . E.g., $\neg p < p$, $\neg pq < pq$, $\neg p\neg q < pq$, but $\neg pq \not< p\neg q$.

(2) We make $\mu(X \times Y) \supseteq \mu(X) \times \mu(Y)$ true, but not the converse.

We order all sequences of length 1 or 2 by $<$.

Suppose $\sigma \in X \times Y - \mu(X \times Y)$. Case 1: $X \times Y$ consists of sequences of length 2. Then, by definition, $\sigma \notin \mu(X) \times \mu(Y)$. Case 2: $X \times Y$ consists of sequences of length 3. Then $\sigma = p\neg q\neg r$, and there is $\tau = \neg p\neg q\neg r \in X \times Y$. So $\{p, \neg p\} \subseteq X$ or $\{p\neg q, \neg p\neg q\} \subseteq X$, but in both cases $\sigma \upharpoonright X \notin \mu(X)$.

Finally, note that $\mu(\text{TRUE}) \not\subseteq \{\neg p\neg q\neg r\}$, so full $(\mu * 1)$ does not hold.

(3) We make interpolation hold, but $\mu(X) \times \mu(Y) \not\subseteq \mu(X \times Y)$:

We order all sequences of length 3 by $<$. Shorter sequences are made incomparable, so for shorter sequences $\mu(X) = X$.

Obviously, in general $\mu(X) \times \mu(Y) \not\subseteq \mu(X \times Y)$.

But the proof of Proposition 5.7 (page 19) goes through as above, only directly, without the use of factorizing and taking μ of the factors.

□

5.3.5 A relevance problem

Consider the formula $\phi := a \wedge \neg a \wedge b$. Then $M(\phi) = \emptyset$. But we cannot recover where the problem came from, and this results in the EFQ rule. We now discuss one, purely algebraic, approach to remedy.

Consider 3 valued models, with a new value b for both, in addition to t and f . Above formula would then have the model $m(a) = b$, $m(b) = t$. So there is a model, EFQ fails, and we can recover the culprit.

To have the usual behaviour of \wedge as intersection, it might be good to change the definition so that $m(x) = b$ is always a model. Then $M(b) = \{m(b) = t, m'(b) = b\}$, $M(\neg b) = \{m(b) = f, m'(b) = b\}$, and $M(b \wedge \neg b) = \{m'(b) = b\}$.

It is not yet clear which version to choose, and we have no syntactic characterization.

5.3.6 Small subspaces

When considering small subsets in nonmonotonic logic, we neglect small subsets of models. What is the analogue when considering small subspaces, i.e. when $J = J' \cup J''$, with J'' small in J in nonmonotonic logic?

It is perhaps easiest to consider the relation based approach first. So we have an order on $\Pi J'$ and one on $\Pi J''$, J'' is small, and we want to know how to construct a corresponding order on ΠJ . Two solutions come to mind:

- a less radical one: we make a lexicographic ordering, where the one on $\Pi J'$ has precedence over the one on $\Pi J''$,
- a more radical one: we totally forget about the ordering of $\Pi J''$, i.e. we do as if the ordering on $\Pi J''$ were the empty set, i.e. $\sigma' \sigma'' \prec \tau' \tau''$ iff $\sigma' \prec \tau'$ and $\sigma'' = \tau''$.

We call this condition *forget*(J'').

The more radical one is probably more interesting. Suppose ϕ' is written in language J' , ϕ'' in language J'' , we then have $\phi' \wedge \phi'' \vdash \psi' \wedge \psi''$ iff $\phi' \vdash \psi'$ and $\phi'' \vdash \psi''$.

5.3.7 Distance relations

The corresponding properties will be:

For GH distances:

- (1) $d(\sigma, \tau) \leq d(\alpha, \beta)$ and $d(\sigma', \tau') \leq d(\alpha', \beta')$ and $d(\sigma, \tau) < d(\alpha, \beta)$ or $d(\sigma', \tau') < d(\alpha', \beta')$ $\Rightarrow d(\sigma\sigma', \tau\tau') < d(\alpha\alpha', \beta\beta')$
- (2) $d(\sigma\sigma', \tau\tau') < d(\alpha\alpha', \beta\beta')$ $\Rightarrow d(\sigma, \tau) < d(\alpha, \beta)$ or $d(\sigma', \tau') < d(\alpha', \beta')$

and for small sets J' :

$$d(\sigma\sigma', \tau\tau') < d(\alpha\alpha', \beta\beta') \Leftrightarrow d(\sigma, \tau) < d(\alpha, \beta).$$

5.3.8 Reactivity and language change

E.g., for small J' : big J erases the J' -relation.

5.4 Summary of properties

Note that $A \times B \subseteq X \times Y$ big $\Rightarrow A \subseteq X$ big etc. is intuitively better justified than the other direction, as the proportion might increase in the latter, decrease in the former. Cf. [GS09a], the Size Table, increasing proportions.

Multiplication law	Scenario 1, $Y * X$ in Diagram 5.2 (page 15)			Scenario 2		
	Addition law	Logical property	Relation property	Logical property α, β in \mathcal{L}_1 , α', β' in \mathcal{L}_2 $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ (disjoint)	Relation property	Interpolation
$x * 1 = x$		trivial		$\alpha \sim_{\mathcal{L}_1} \beta \Leftrightarrow \alpha \sim_{\mathcal{L}} \beta$		
$1 * x = x$		trivial				
$x * s = s$	(iM)	$\alpha \sim \neg\beta \Rightarrow \alpha \sim \neg\beta \vee \gamma$	-	$\alpha \sim_{\mathcal{L}_1} \beta \Leftrightarrow \alpha \sim_{\mathcal{L}} \beta$		
$s * x = s$	(eML)	$\alpha \wedge \beta \sim \neg\gamma \Rightarrow \alpha \neg\beta \vee \neg\gamma$	-			
$b * b = b$	$(\langle \omega * s \rangle, (\mathcal{M}_\omega^+))$ (3)	$\alpha \sim \beta, \alpha \wedge \beta \sim \gamma \Rightarrow \alpha \sim \gamma$	- (Filter)	$\alpha \sim_{\mathcal{L}_1} \beta, \alpha' \sim_{\mathcal{L}_2} \beta' \Leftrightarrow \alpha \wedge \alpha' \sim_{\mathcal{L}} \beta \wedge \beta'$	(GH)	$\sim \circ \sim$
$b * m = m$	$(\langle \omega * s \rangle, (\mathcal{M}_\omega^+))$ (2)	$\alpha \not\sim \neg\beta, \alpha \wedge \beta \sim \gamma \Rightarrow \alpha \not\sim \neg\beta \vee \neg\gamma$	- (Filter)	$\alpha \not\sim_{\mathcal{L}_1} \neg\beta, \alpha' \sim_{\mathcal{L}_2} \beta' \Rightarrow \alpha \wedge \alpha' \not\sim_{\mathcal{L}} \neg\beta \vee \beta'$		
$m * b = m$	$(\langle \omega * s \rangle, (\mathcal{M}_\omega^+))$ (1)	$\alpha \sim \beta, \alpha \wedge \beta \not\sim \neg\gamma \Rightarrow \alpha \not\sim \neg\beta \vee \neg\gamma$	- (Filter)			
$m * m = m$	(\mathcal{M}^{++})	Rational Monotony	ranked	$\alpha \not\sim_{\mathcal{L}_1} \neg\beta, \alpha' \sim_{\mathcal{L}_2} \neg\beta' \Rightarrow \alpha \wedge \alpha' \not\sim_{\mathcal{L}} \neg\beta \vee \neg\beta'$		
$pr(b) = b$				$\alpha \sim \beta \Rightarrow \alpha \uparrow \mathcal{L}_1 \sim \beta \uparrow \mathcal{L}_1$	$pr(\prec)$	
$pr(b) = b + m * m = m$						$\vdash \circ \sim$
J' small		analogue: System P		$\alpha \wedge \alpha' \sim \beta \wedge \beta' \Leftrightarrow \alpha \wedge \beta, \alpha' \vdash \beta'$	$forget(J')$	

Table 2: Multiplication laws

6 General semantic interpolation

6.1 Introduction

We assume that \rightarrow respects the order \leq on truth values.

6.1.1 Generalization of model sets and (in)essential variables

See Table 3 (page 24) for notation and definitions.

Example 6.1

This example shows that 2 different formulas ϕ and ϕ' may define the same $f_\phi = f_{\phi'}$, but neglecting a certain variable should give different results.

Define two new unary operators $K(x) := 1$ (constant), $M(x) := \min\{1, x\}$. Consider 3 truth values.

a	b	$\phi = K(a) \wedge b$	$\phi' = K(a) \wedge M(b)$
0	0	0	0
0	1	1	1
0	2	1	1
1	0	0	0
1	1	1	1
1	2	1	1
2	0	0	0
2	1	1	1
2	2	1	1

So they define the same model function $f : \mathcal{M}_{\mathcal{L}} \rightarrow Val$. But when we forget about a , the first should just be b , but the second should be $M(b)$.

□

6.2 Many-valued propositional interpolation

The situation:

We prefer to think of models as sequences, whose elements are truth-values so we change notation slightly. Let L be the set of propositional variables of a given language \mathcal{L} . Let (V, \leq) be a finite, totally ordered set (of values). Let $\Gamma \subseteq \mathcal{M}_{\mathcal{L}}$.

Let $L = J \cup J' \cup J''$ be a disjoint union. If $f : \mathcal{M}_L \rightarrow V$ is insensitive to $J \cup J''$, we can define for $\sigma_{J'} : J' \rightarrow V$ $f(\sigma_{J'})$ as any $f(\sigma')$ such that $\sigma' \upharpoonright J' = \sigma_{J'}$.

Definition 6.1

Call Γ rich iff for all $\sigma, \sigma' \in \Gamma$, $J \subseteq L$ $(\sigma \upharpoonright J) \cup (\sigma' \upharpoonright (L - J)) \in \Gamma$.

(I.e., we may cut and paste models.)

Fact 6.1

Let Γ be rich, $f, g : \Gamma \rightarrow V$, $f(\sigma) \leq g(\sigma)$ for all $\sigma \in \Gamma$.

Then $f^+(\sigma_{J'}) \leq g^-(\sigma_{J'})$ for all $\sigma_{J'} \in \Gamma \upharpoonright J'$, and any $h : \Gamma \upharpoonright J' \rightarrow V$ which is insensitive to $J \cup J''$ is an interpolant iff $f^+(\sigma_{J'}) \leq h(\sigma_{J'}) \leq g^-(\sigma_{J'})$ for all $\sigma_{J'} \in \Gamma \upharpoonright J'$.

Proof

Let $L = J \cup J' \cup J''$ be a pairwise disjoint union. Let f be insensitive to J , g be insensitive to J'' .

$h : \Gamma \rightarrow V$ will have to be insensitive to $J \cup J''$, so we will have to define h on $\Gamma \upharpoonright J'$.

Let $\sigma_{J'} : \Gamma \upharpoonright J' \rightarrow V$. Choose $\sigma_{J''}$ such that $f^+(\sigma_{J'}) = f(\sigma_J \sigma_{J'} \sigma_{J''})$ for any σ_J . Recall that f is insensitive to J . Likewise, choose σ_J such that $g^-(\sigma_{J'}) = g(\sigma_J \sigma_{J'} \sigma_{J''})$ for any $\sigma_{J''}$.

We have $f^+(\sigma_{J'}) \leq g^-(\sigma_{J'})$.

Notation and definitions			
		2-valued $\{0, 1\}$	many-valued (V, \leq)
propositional $L' \subseteq L$	language L	propositional variables s, \dots	
	model S	$S : L \rightarrow \{0, 1\}$	$S : L \rightarrow V$
	\mathcal{M}_L set of all L -models		
	$S \upharpoonright L'$	like S , but restricted to L'	
	$S \sim_{L'} S'$	$S \sim_{L'} S'$ iff $\forall s \in L'. S(s) = S'(s)$	
	model set of formula ϕ	$M(\phi) \subseteq \mathcal{M}_L, f_\phi : \mathcal{M}_L \rightarrow \{0, 1\}$	$f_\phi : \mathcal{M}_L \rightarrow V$
	general model set	$M \subseteq \mathcal{M}_L, f : \mathcal{M}_L \rightarrow \{0, 1\}$	$f : \mathcal{M}_L \rightarrow V$
	f insensitive to L'	$\forall S, S' \in \mathcal{M}_L. (S \sim_{L-L'} S' \Rightarrow f(S) = f(S'))$	
	$f^+(S \upharpoonright L'), f^-(S \upharpoonright L')$	$f^+(S \upharpoonright L') = \max\{f(S') : S' \in \mathcal{M}_L, S \sim_{L'} S'\}$ $f^-(S \upharpoonright L') = \min\{f(S') : S' \in \mathcal{M}_L, S \sim_{L'} S'\}$	
$f \leq g$	$\forall S \in \mathcal{M}_L. f(S) \leq g(S)$		
propositional modal $L' \subseteq L$	language L	propositional variables s, \dots	
	(Kripke) structure $S = \langle U_S, R_S, I_S \rangle$	$I_S : U_S \times L \rightarrow \{0, 1\}$	$I_S : U_S \times L \rightarrow V$
	U_S : points, R_S rel., I_S interpret. \mathcal{M}_L set of all L -structures with desired R_S -properties		
	$S \upharpoonright L'$	like S , but I_S restricted to L'	
	$S \sim_{L'} S'$	$S \sim_{L'} S'$ iff $U_S = U_{S'}$ and $R_S = R_{S'}$ and $\forall s \in L' \forall u \in U_S. I_S(u, s) = I_{S'}(u, s)$ (iff $S \upharpoonright L' = S' \upharpoonright L'$)	
	for all S : model set of formula ϕ	$M_S(\phi) \subseteq U_S, f_{S,\phi} : U_S \rightarrow \{0, 1\}$	$f_{S,\phi} : U_S \rightarrow V$
	for all S : general model set	$M_S \subseteq U_S, f_S : U_S \rightarrow \{0, 1\}$	$f_S : U_S \rightarrow V$
	f_S insensitive to L'	$\forall S, S' \in \mathcal{M}_L (S \sim_{L-L'} S' \Rightarrow \forall u \in U_S = U_{S'}. f_S(u) = f_{S'}(u))$	
	$f_S^+(S \upharpoonright L'), f_S^-(S \upharpoonright L')$	$f_S^+(S \upharpoonright L') = \max\{f_{S'}(u) : S' \in \mathcal{M}_L, S \sim_{L'} S'\}$ $f_S^-(S \upharpoonright L') = \min\{f_{S'}(u) : S' \in \mathcal{M}_L, S \sim_{L'} S'\}$	
$f_S \leq g_S$	$\forall u \in U_S. f_S(u) \leq g_S(u)$		
first order L_1 unary pred., L_2 binary, ... $L' \subseteq L$	language $L = L_1 \cup L_2, \dots$	predicates $p(\cdot), q(\cdot, \cdot)$	
	structure $S = \langle U_S, I_S \rangle$	$I_S(p(\cdot)) \subseteq U_S,$	$I_S : U_S \times L_1 \rightarrow V,$
	\mathcal{M}_L set of all L -structures	$I_S(q(\cdot, \cdot)) \subseteq U_S \times U_S$	$I_S : (U_S \times U_S) \times L_2 \rightarrow V$
	$S \upharpoonright L'$	like S , but I_S restricted to L'	
	$S \sim_{L'} S'$	$S \sim_{L'} S'$ iff $U_S = U_{S'}$ and $\forall s \in L' \forall \langle u \rangle \in U_S^n. I_S(\langle u \rangle, s) = I_{S'}(\langle u \rangle, s)$ (iff $S \upharpoonright L' = S' \upharpoonright L'$)	
	for all S : model set of formula ϕ with n free variables	$M_S(\phi(x_1, \dots, x_n)) \subseteq U_S^n$ $f_{S,\phi} : U_S^n \rightarrow \{0, 1\}$	$f_{S,\phi} : U_S^n \rightarrow V$
	for all S : gen. model set, n var.	$f_S : U_S^n \rightarrow \{0, 1\}$	$f_S : U_S^n \rightarrow V$
	f insensitive to L'	$\forall S, S' \in \mathcal{M}_L (S \sim_{L-L'} S' \Rightarrow \forall u_i \in U_S = U_{S'}. f_S(u_1, \dots, u_n) = f_{S'}(u_1, \dots, u_n))$	
	$f_S^+(S \upharpoonright L'), f_S^-(S \upharpoonright L')$	$f_S^+(S \upharpoonright L') = \max\{f_{S'}(\langle u \rangle) : S' \in \mathcal{M}_L, S \sim_{L'} S'\}$ $f_S^-(S \upharpoonright L') = \min\{f_{S'}(\langle u \rangle) : S' \in \mathcal{M}_L, S \sim_{L'} S'\}$	
$f_S \leq g_S$	$\forall \langle u_1, \dots, u_n \rangle \in U_S^n. f_S(u_1, \dots, u_n) \leq g_S(u_1, \dots, u_n)$		
for all cases	semantic equivalence of ϕ, ψ	$f_\phi = f_\psi$ (or for all $S f_{S,\phi} = f_{S,\psi}$)	
	definability of f	$\exists \phi : f_\phi = f$ (or for all $S f_{S,\phi} = f_S$)	
	$\Gamma \upharpoonright L'$	(for $\Gamma \subseteq \mathcal{M}_L$) $\Gamma \upharpoonright L' := \{S \upharpoonright L' : S \in \Gamma\}$	
	(ir)relevant	$s \in L$ is irrelevant for f iff f is insensitive to s , $I(f) := \{s \in L : s \text{ is irrelevant for } f\}, R(f) := L - I(f)$ $I(\phi) := I(f_\phi), R(\phi) := R(f_\phi)$	

Table 3: Notation and Definitions

Proof: Fix $\sigma_{J'}$. Choose $\sigma_{J''}$ such that $f(\sigma_J\sigma_{J'}\sigma_{J''})$ is maximal for $\sigma_{J'}$ and any σ_J . Let $\tau_{J''}$ be one such $\sigma_{J''}$. Choose σ_J such that $g(\sigma_J\sigma_{J'}\sigma_{J''})$ is minimal for $\sigma_{J'}$ and any $\sigma_{J''}$. Let τ_J be one such σ_J . Consider $\tau_J\sigma_{J'}\tau_{J''} \in \Gamma$ (recall that Γ is rich). By definition, $f^+(\sigma_{J'}) := f(\tau_J\sigma_{J'}\tau_{J''})$ and $g^-(\sigma_{J'}) := g(\tau_J\sigma_{J'}\tau_{J''})$, but by prerequisite $f(\tau_J\sigma_{J'}\tau_{J''}) \leq g(\tau_J\sigma_{J'}\tau_{J''})$, so $f^+(\sigma_{J'}) \leq g^-(\sigma_{J'})$.

Thus, any h such that h is insensitive to $J \cup J''$ and

$$(Int) \quad f^+(\sigma_{J'}) \leq h(\sigma_{J'}) \leq g^-(\sigma_{J'})$$

is an interpolant for f and g .

But (Int) is also a necessary condition.

Proof:

Suppose h is an interpolant and $h(\sigma_{J'}) < f^+(\sigma_{J'})$. Let $\tau_{J''}$ be as above, i.e., $f(\sigma_J\sigma_{J'}\tau_{J''}) = f^+(\sigma_{J'})$ for any σ_J . Then $h(\sigma_J\sigma_{J'}\tau_{J''}) = h(\sigma_{J'}) < f^+(\sigma_{J'}) = f(\sigma_J\sigma_{J'}\tau_{J''})$, so h is not an interpolant.

The proof that $h(\sigma_{J'})$ has to be $\leq g^-(\sigma_{J'})$ is analogous.

We summarize:

f and g have an interpolant h , and h is an interpolant for f and g iff h is insensitive to $J \cup J''$ and $f^+(\sigma_{J'}) \leq h(\sigma_{J'}) \leq g^-(\sigma_{J'})$.

□

6.3 Many-valued propositional modal interpolation

Fix $S \in \mathcal{M}_L$. By prerequisite, $f_S \leq g_S$. Set $f := f_S, g := g_S$. Find an interpolant $h, f \leq h \leq g$ as above in the propositional case, set $h_S := h$. Then $\{h_S : S \in \mathcal{M}_L\}$ is an interpolant for all f_S, g_S .

6.4 Many-valued first order interpolation

Analogous to the modal logic case:

Fix S . Let $f_S \leq g_S$ be defined for $\langle u_1, \dots, u_n \rangle \in U_S^n$. Set $f := f_S, g := g_S$. Find an interpolant $h, f \leq h \leq g$ as above in the propositional case, set $h_S := h$. Then $\{h_S : S \in \mathcal{M}_L\}$ is an interpolant for all f_S, g_S .

References

- [ALS98-1] L.Audibert, C.Lhoussaine, K.Schlechta: “Distance based revision of preferential logics”, Belief Revision Workshop of KR98, Trento, Italy, 1998 (electronic proceedings), and Logic Journal of the Interest Group in Pure and Applied Logics, Vol. 7, No. 4, pp. 429-446, 1999
- [Bou94] C.Boutilier, “Unifying default reasoning and belief revision in a modal framework”, Artificial Intelligence, 68 : 33–85, 1994
- [DP94] A.Darwiche, J.Pearl, “On the Logic of Iterated Belief Revision”, in: “Proceedings of the fifth Conference on Theoretical Aspects of Reasoning about Knowledge”, R.Fagin ed., pp. 5-23, Morgan Kaufman, Pacific Grove, CA, 1994
- [GS08b] D.Gabbay, K.Schlechta, “Reactive preferential structures and nonmonotonic consequence”, hal-00311940, arXiv 0808.3075, Review of Symbolic Logic, Vol. 2, No. 2, pp. 414-450,
- [GS08g] D.Gabbay, K.Schlechta, “A semantics for obligations”, submitted, hal-00339393, arXiv 0811.2754
- [GS09a] D.Gabbay, K.Schlechta, “Size and logic”, arXiv 0903.1367, Review of Symbolic Logic, Vol. 2, No. 2, pp. 396-413
- [GS09c] D.Gabbay, K.Schlechta, “Semantic interpolation”, submitted, preliminary version: arXiv.org 0906.4082
- [Ker99] G.Kern-Isberner, “Postulates for conditional belief revision”, Proceedings IJCAI 99, T.Dean ed., Morgan Kaufmann, pp.186-191, 1999
- [LMS01] D.Lehmann, M.Magidor, K.Schlechta: “Distance Semantics for Belief Revision”, Journal of Symbolic Logic, Vol.66, No. 1, March 2001, p. 295-317
- [PV09] D.Pearce, A.Valverde, “Interpolation in equilibrium logic and answer set programming”, submitted
- [Sch04] K.Schlechta: “Coherent Systems”, Elsevier, Amsterdam, 2004
- [Sch92] K.Schlechta: “Some results on classical preferential models”, Journal of Logic and Computation, Oxford, Vol.2, No.6 (1992), p. 675-686