



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-00406635v8

HAL Id: hal-00406635

<https://hal.science/hal-00406635v8>

Preprint submitted on 13 Jun 2011 (v8), 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 ||

June 13, 2011

Abstract

We investigate several technical and conceptual questions.

Contents

1	Introduction	2
2	Countably many disjoint sets	2
3	Independence as ternary relation	3
3.1	Situation and definitions	3
3.2	New rules	5
3.3	There is no finite characterization	8
3.3.1	Discussion	8
3.3.2	Composition of layers	8
3.3.3	Systematic construction	9
3.3.4	The cases to consider	10
3.3.5	Solution of the cases	11
	References	11

*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>

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

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 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).

We show here that, independent of the cardinality of the language, one can define only countably many inconsistent formulas.

The problem is due to D. Makinson (personal communication).

Example 2.1

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 2.1

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: Rewrite 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 $\phi \in X_i$. Without loss of generality, let $\phi = p_1 \wedge \dots \wedge p_i$. 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, $card(X_i)$ 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 two truth values. Obviously, we can easily generalize to finitely many truth values.

3 Independence as ternary relation

3.1 Situation and definitions

Independence as an abstract ternary relation for probability and other situations has been examined by Pearl et al., see, e.g., [Pea88]. We interpret independence here differently, but in a related way.

Definition 3.1

We consider function sets Σ etc. over a fixed, arbitrary domain I , into some fixed codomain K .

For pairwise disjoint subsets X, Y, Z of I , we define

$\langle X | Y | Z \rangle$ iff for all $f, g \in \Sigma$ such that $f \upharpoonright Y = g \upharpoonright Y$ ($f \upharpoonright Y$ the restriction of f to elements of Y , etc.), there is $h \in \Sigma$ such that $h \upharpoonright X = f \upharpoonright X, h \upharpoonright Y = f \upharpoonright Y = g \upharpoonright Y, h \upharpoonright Z = g \upharpoonright Z$. Y may be empty, then the condition $f \upharpoonright Y = g \upharpoonright Y$ is void.

Thus, the definition is relative to Σ .

$\langle X | Y | Z \rangle$ means thus, that we can piece functions together, or that we have a sort of decomposition of Σ into a product. This is an independence property, we can put parts together independently.

Notation 3.1

In more complicated cases, we will often write ABC for $\langle A | B | C \rangle$, and $\neg ABC$ or $-ABC$ if $\langle A | B | C \rangle$ does not hold. Moreover, we will often just write $f(A)$ for $f \upharpoonright A$, etc.

For $\langle A \cup A' | B | C \rangle$, we will then write $(AA')BC$, etc.

If only singletons are involved, we will sometimes write abc instead of ABC , etc.

When we speak about fragments of functions, we will often write just $A : \sigma$ for $\sigma \upharpoonright A, B : \sigma = \tau$ for $\sigma \upharpoonright B = \tau \upharpoonright B$, etc.

We use the following notations for functions:

Definition 3.2

The constant functions 0_c and 1_c :

$$0_c(i) = 0 \text{ for all } i \in I$$

$$1_c(i) = 1 \text{ for all } i \in I$$

Moreover, when we define a function $\sigma : I \rightarrow \{0, 1\}$ argument by argument, we abbreviate $\sigma(a) = 0$ by $a = 0$, etc.

Sometimes, we also give (a fragment of) a function just by the sequence of the values, so instead of writing $a = 0, b = 1, c = 1$, we just write 011 - context will disambiguate.

E.g., Pearl discusses the following rules for the ternary relation:

Definition 3.3

- (a) Symmetry: $\langle X | Y | Z \rangle \leftrightarrow \langle Z | Y | X \rangle$
- (b) Decomposition: $\langle X | Y | Z \cup W \rangle \rightarrow \langle X | Y | Z \rangle$
- (c) Weak Union: $\langle X | Y | Z \cup W \rangle \rightarrow \langle X | Y \cup W | Z \rangle$
- (d) Contraction: $\langle X | Y | Z \rangle$ and $\langle X | Y \cup Z | W \rangle \rightarrow \langle X | Y | Z \cup W \rangle$
- (e) Intersection: $\langle X | Y \cup W | Z \rangle$ and $\langle X | Y \cup Z | W \rangle \rightarrow \langle X | Y | Z \cup W \rangle$
- (\emptyset) Empty outside: $\langle X | Y | Z \rangle$ if $X = \emptyset$ or $Z = \emptyset$.

We show now that above Rules (a) – (d) hold in our context, but (e) does not hold.

Fact 3.1

In our interpretation,

- (1) rule (e) does not hold,
- (2) all $\langle X | Y | \emptyset \rangle$ (and thus also all $\langle \emptyset | Y | Z \rangle$) hold.
- (3) rules (a) – (d) hold, even when one or both of the outside elements of the triples is the empty set.

Proof

- (1) (e) does not hold:

Consider $I := \{x, y, z, w\}$ and $U := \{1111, 0100\}$. Then $x(yw)z$ and $x(yz)w$, as for all $\sigma \upharpoonright yw$ there is just one τ this σ can be. The same holds for $x(yz)w$. But for $y = 1$, there are two different paths through $y = 1$, which cannot be combined.

- (2) This is a trivial consequence of the fact that $\{f : f : \emptyset \rightarrow U\} = \{\emptyset\}$.

- (3) Rules (a), (b), (c) are trivial, by definition, also for $X, Z = \emptyset$. In (c), if $W = \emptyset$, there is nothing to show.

Rule (d): The cases for $X, W, Z = \emptyset$ are trivial. Assume σ, τ such that $\sigma \upharpoonright Y = \tau \upharpoonright Y$, we want to combine $\sigma \upharpoonright X$ with $\tau \upharpoonright Z \cup W$. By $\langle X | Y | Z \rangle$, there is ρ such that $\rho \upharpoonright X = \sigma \upharpoonright X$, $\rho \upharpoonright Y = \sigma \upharpoonright Y = \tau \upharpoonright Y$, $\alpha \upharpoonright X = \rho \upharpoonright Z = \tau \upharpoonright Z$. Thus ρ and τ satisfy the prerequisite of $\langle X | Y \cup Z | W \rangle$, and there is α such that $\alpha \upharpoonright X = \rho \upharpoonright X = \sigma \upharpoonright X$, $\alpha \upharpoonright X = \rho \upharpoonright Y = \sigma \upharpoonright Y = \tau \upharpoonright Y$, $\alpha \upharpoonright W = \tau \upharpoonright W$.

□

Next, we give an example which shows that increasing the center set can change validity of the triple in any way.

Example 3.1

Consider $I := \{x, a, b, c, d, z\}$.

Let $\Sigma := \{111111, 011110, 011101, 111100, 110111, 010000\}$.

Then $\neg x(abcd)z, x(abc)z, \neg x(ab)z$.

For $\neg x(abcd)z$, fix $abcd = 1111$, then $111111, 011110 \in \Sigma$, but, e.g., $011111 \notin \Sigma$.

For $x(abc)z$, the following combinations of abc exist: $111, 101, 100$. The result is trivial for 101 and 100 . For 111 , all combinations for x and z with 0 and 1 exist.

For $\neg x(ab)z$, fix $ab = 10$, then $110111, 010000 \in \Sigma$, but there is, e.g., no $110xy0 \notin \Sigma$.

□

Validity of $ABC, ACD, ADE, AEB \Rightarrow ABE$						
	A	B	C	D	E	$ABE?$
	σ	$\sigma = \tau$			τ	
(1) ρ_1	σ	$\sigma = \tau$	τ			ABC
(2) ρ_2	σ		τ	τ		ACD
(3) ρ_3	σ			τ	τ	ADE
(4) ρ_4	σ	$\sigma = \tau$			τ	AEB

3.2 New rules

Above rules (a) – (d) are not the only ones to hold, and we introduce now more complicated ones, and show that they hold in our situation. Of the possibly infinitary rules, only (Loop1) is given in full generality, (Loop2) and (Struc) are only given to illustrate that even the infinitary rule (Loop1) is not all there is.

For warming up, we consider the following short version of (Loop1):

Example 3.2

$ABC, ACD, ADE, AEB \Rightarrow ABE$.

We show that this rule holds in all Σ .

Suppose $A : \sigma, B : \sigma = \tau, C : \tau$, so by ABC , there is ρ_1 such that

$A : \rho_1 = \sigma, B : \rho_1 = \sigma = \tau, C : \rho_1 = \tau$. So by ACD , there is ρ_2 such that

$A : \rho_2 = \sigma, C : \rho_2 = \rho_1 = \tau, D : \rho_2 = \tau$. So by ADE , there is ρ_3 such that

$A : \rho_3 = \sigma, D : \rho_3 = \rho_2 = \tau, E : \rho_3 = \tau$. So by AEB , there is ρ_4 such that

$A : \rho_4 = \sigma, E : \rho_4 = \rho_3 = \tau, B : \rho_4 = \tau = \sigma$.

So ABE .

We abbreviate this reasoning by:

(1) $ABC : A : \sigma, B : \sigma = \tau, C : \tau$

(2) $ACD : (1) + \tau$

(3) $ADE : (2) + \tau$

(4) $AEB : (3) + \tau$

So ABE .

It is helpful to draw a little diagram as in the following Table 3.1 (page 5).

.

We introduce now some new rules.

Definition 3.4

- (Bin1)
 $XYZ, XY'Z, Y(XZ)Y' \Rightarrow X(YY')Z$
- (Bin2)
 $XYZ, XZY', Y(XZ)Y' \Rightarrow X(YY')Z$
- (Loop1)
 $ABC, ACD_1, AD_1D_2, \dots, AD_{i-1}D_i, AD_iD_{i+1}, AD_{i+1}D_{i+2}, \dots, AD_{n-1}D_n, AD_nG, AGB \Rightarrow ABG$
so we turn AGB around to ABG .
When we have to be more precise, we will denote this condition ($Loop1_n$) to fix the length.

- (Loop2)
 $ABC, ACD, DAE, DEF, FDG, FGH, HFB \Rightarrow HBF :$
- (Struc)
 $ABC, ACD, AEF, AFG, DAG, AHI, AIJ, GAJ, BAJ \Rightarrow ABJ$

The complicated structure of these rules suggests already that the ternary relations are not the right level of abstraction to speak about construction of functions from fragments. This is made formal by our main result below, which shows that there is no finite characterization by such relations. In other words, the main things happen behind the screen.

Fact 3.2

The new rules are valid in our situation.

Proof

- (Bin1)
 - (1) $XYZ : X : \sigma, Y : \sigma = \tau, Z : \tau$
 - (2) $XY'Z : X : \sigma, Y' : \sigma = \tau, Z : \tau$
 - (3) $Y(XZ)Y' : (1) + (2)$
 So $X(YY')Z$.
- (Bin2)

Let $X : \sigma, Y : \sigma = \tau, Y' : \sigma = \tau, Z : \tau$

 - (1) $XYZ : X : \sigma, Y : \sigma = \tau, Z : \tau$
 - (2) $XZY' : (1) + \tau$
 - (3) $Y(XZ)Y' : (1) + (2)$
 So $X(YY')Z$.
- (Loop1)
 - (1) $ABC : A : \sigma, B : \sigma = \tau, C : \tau$
 - (2) $ACD_1 : (1) + \tau$
 - (3) $AD_1D_2 : (2) + \tau$
 -
 - ($i + 1$) $AD_{i-1}D_i : (i) + \tau$
 - ($i + 2$) $AD_iD_{i+1} : (i + 1) + \tau$
 - ($i + 3$) $AD_{i+1}D_{i+2} : (i + 2) + \tau$
 -
 - ($n + 1$) $AD_{n-1}D_n : (n) + \tau$
 - ($n + 2$) $AD_nG : (n + 1) + \tau$
 - ($n + 3$) $AGB : (n + 2) + \tau = \sigma$
 So ABG .
- (Loop2)

Let

 - (1) $ABC : A : \sigma, B : \sigma = \tau, C : \tau$
 - (2) $ACD : 1 + \tau$

(3) $DAE : 2 + \sigma$

(4) $DEF : 3 + \sigma$

(5) $FDG : 4 + \tau$

(6) $FGH : 5 + \tau$

(7) $HFB : 6 + \sigma$

So HBF by $B : \sigma = \tau$.

• (Struc)

(1) $ABC : A : \sigma, B : \sigma = \tau, C : \tau$

(2) $ACD : (1) + \tau$

(3) $AEF : A : \sigma, E : \sigma = \tau, F : \tau$

(4) $AFG : (3) + \tau$

(5) $DAG : (2) + (4)$

(6) $AHI : A : \sigma, H : \sigma = \tau, I : \tau$

(7) $AIJ : (6) + \tau$

(8) $GAJ : (5) + (7)$

(9) $BAJ : (1) + (8)$

So ABJ .

Note that we use here $B : \sigma = \tau, E : \sigma = \tau, H : \sigma = \tau$, whereas the other tripels are used for other functions.

□

Next we show that the full (Loop1) cannot be derived from the basic rules (a) – (d) and (Bin1), and shorter versions of (Loop1). (This is also a consequence of the sequel, but we want to point it out right away.)

Fact 3.3

Let $n \geq 1$, then $(Loop1_n)$ does not follow from the rules (a) – (d), (\emptyset), (Bin1), and the shorter versions of (Loop1)

Proof

Consider the following set of tripels $L \cup L'$ over $I := \{a, b, c, g, d_1, \dots, d_n\}$:

$L := \{abc, acd_1, ad_1d_2, \dots, ad_id_{i+1}, \dots, ad_n g, agb\}$,

$L' := \{\emptyset AB : A \cap B = \emptyset, A \cup B \subseteq I\}$,

and close this set under symmetry (rule (a)). Call the resulting set \mathcal{A} .

Note that, on the outside, we have \emptyset or singletons, inside singletons or \emptyset . If the inside is \emptyset , one of the outside sets must also be \emptyset .

When we look at L , and define a relation $<$ by $x < y$ iff $axy \in L$, we see that the only $<$ -loop is $b < c < d_1 < \dots < d_n < g < b$.

We show first that \mathcal{A} is closed under rules (a) – (d) (see Definition 3.3 (page 4)).

(a) is trivial.

(b) If $W = \emptyset$ or $Z = \emptyset$, this is trivial, if $W = Z$, this is trivial, too.

(c) If $Z \cup W = \emptyset$, this is trivial, if $Z \cup W$ is a singleton, so $Z = \emptyset$ or $W = \emptyset$ or $Z = W$. $Z = \emptyset$ or $W = \emptyset$ are trivial, otherwise $Z = W$ contradicts disjointness.

(d) $Z = \emptyset$ is trivial, so is $W = \emptyset$, otherwise $Z = W$ contradicts disjointness.

(Bin1) $X = \emptyset$ or $Z = \emptyset$ are trivial, otherwise $X = Z$ is excluded by disjointness. So we are in L' for $Y(XZ)Y'$. So $Y = \emptyset$ or $Y' = \emptyset$ and it is trivial.

Obviously, $(Loop1_n)$ does not hold.

We show now that all $(Loop1_k)$, $0 \leq k < n$ hold.

We do the case $k = 0$ first.

Consider $ABC, ACG, AGB \Rightarrow ABG$.

If $A = \emptyset$ or $G = \emptyset$, the condition holds.

So assume $A, G \neq \emptyset$. Thus, by above remark, $C \neq \emptyset$, and then $B \neq \emptyset$. Thus, the prerequisites have to be in L . Moreover, A has to be a , which is the only element occurring repeatedly on the outside. Consider now a relation $<'$ defined by $U <' V$ iff AUV is among the prerequisites. We then have $B <' C <' G <' B$, but this contradicts the fact that the only existing loop goes through all elements. So the prerequisites cannot be all in L , a contradiction.

Consider the case $0 < k < n$.

This has the form $ABC, ACD_1, AD_1D_2, \dots, AD_kG, AGB \Rightarrow ABG$.

Again, the cases $A = \emptyset$ or $G = \emptyset$ are obvious, assume $A, G \neq \emptyset$. Then $D_k \neq \emptyset$, so descending all $D_i \neq \emptyset$, and $C \neq \emptyset$ and $B \neq \emptyset$. Thus, all prerequisites are in L . Defining again a relation $<'$ as above, we see again that the resulting $<'$ -loop is too short, and we have again a contradiction.

□

3.3 There is no finite characterization

We turn to our main result.

3.3.1 Discussion

Consider the following simple, short, loop for illustration:

$ABC, ACD, ADE, AEF, AFG, AGB \Rightarrow ABG$ - so we can turn AGB around to ABG .

Of course, this construction may be arbitrarily long.

The idea is now to make ABG false, and, to make it coherent, to make one of the interior conditions false, too, say ADE . We describe this situation fully, i.e. enumerate all conditions which hold in such a situation. If we make now ADE true again, we know this is not valid, so any (finite) characterization must say "NO" to this. But as it is finite, it cannot describe all the interior tripels of the type ADE in a sufficiently long loop, so we just change one of them which it does not "see" to FALSE, and it must give the same answer NO, so this fails.

Basically, we cannot describe parts of the loop, as the $\langle || \rangle$ -language is not rich enough to express it, we see only the final outcome.

The problem is to fully describe the situation.

3.3.2 Composition of layers

A very helpful fact is the following:

Definition 3.5

Let Σ_j be function sets over I into some set K , $j \in J$.

Let $\Sigma := \{ f : I \rightarrow K^J : f(i) = \{ \langle f_j(i), j \rangle : j \in J, f_j \in \Sigma_j \} \}$.

So any $f \in \Sigma$ has the form $f(i) = \langle f_1(i), f_2(i), \dots, f_n(i) \rangle$, $f_m \in \Sigma_m$ (we may assume J to be finite).

Thus, given $f \in \Sigma$, $f_m \in \Sigma_m$ is defined.

Fact 3.4

For the above $\Sigma \langle A | B | C \rangle$ holds iff it holds for all Σ_j .

Thus, we can destroy the $\langle A | B | C \rangle$ independently, and collect the results.

Proof

The proof is trivial, and a direct consequence of the fact that $f = f'$ iff for all components $f_j = f'_j$.

Suppose for some Σ_k , $k \in J$, $\neg \langle A | B | C \rangle$.

So for this k there are $f_k, f'_k \in \Sigma_k$ such that $f_k(B) = f'_k(B)$, but there is no $f''_k \in \Sigma_k$ such that $f''_k(A) = f_k(A)$, $f''_k(B) = f_k(B) = f'_k(B)$, $f''_k(C) = f'_k(C)$ (or conversely). Consider now some $h \in \Sigma$ such that $h_k = f_k$, and h' is like h , but $h'_k = f'_k$, so also $h' \in \Sigma$. Then $h(B) = h'(B)$, but there is no $h'' \in \Sigma$ such that $h''(A) = h(A)$, $h''(B) = h(B) = h'(B)$, $h''(C) = h'(C)$.

Conversely, suppose $\langle A | B | C \rangle$ for all Σ_j . Let $h, h' \in \Sigma$ such that $h(B) = h'(B)$, so for all $j \in J$ $h_j(B) = h'_j(B)$, where $h_j \in \Sigma_j$, $h'_j \in \Sigma_j$, so there are $h''_j \in \Sigma_j$ with $h''_j(A) = h_j(A)$, $h''_j(B) = h_j(B) = h'_j(B)$, $h''_j(C) = h'_j(C)$ for all $j \in J$. Thus, h'' composed of the h''_j is in Σ , and $h''(A) = h(A)$, $h''(B) = h(B) = h'(B)$, $h''(C) = h'(C)$.

□

3.3.3 Systematic construction

Recall the general form of (Loop1) for singletons:

$$abc, acd_1, ad_1d_2, \dots, ad_{i-1}d_i, ad_id_{i+1}, ad_{i+1}d_{i+2}, \dots, ad_{n-1}d_n, ad_n g, agb \Rightarrow abg$$

so we turn agb around to abg .

We will fully describe a model of above tripels, with the exception of abg and ad_id_{i+1} which will be made to fail, and all other $\langle X | Y | Z \rangle$ which are not in above list of tripels to preserve, will fail, too (except for $X = \emptyset$ or $Z = \emptyset$).

We use the following fact:

Fact 3.5

Let $X \subseteq I$, $card(X) > 1$, $\Sigma_X := \{ \sigma : I \rightarrow \{0, 1\} : card\{x \in X : \sigma(x) = 0\} \text{ is even} \}$

Then $\neg ABC$ iff $A \cap X \neq \emptyset$, $C \cap X \neq \emptyset$, $X \subseteq A \cup B \cup C$.

Proof

“ \Leftarrow ”:

Suppose $A \cap X \neq \emptyset$, $C \cap X \neq \emptyset$, $X \subseteq A \cup B \cup C$.

Take σ such that $card\{x \in X : \sigma(x) = 0\}$ is odd, then $\sigma \notin \Sigma_X$. As $X \not\subseteq A \cup B$, there is $\tau \in \Sigma_X$ such that $\sigma \upharpoonright A \cup B = \tau \upharpoonright A \cup B$. As $X \not\subseteq B \cup C$, there is $\rho \in \Sigma_X$ such that $\rho \upharpoonright B \cup C = \sigma \upharpoonright B \cup C$. Thus, $\tau \upharpoonright B = \rho \upharpoonright B$. If there were $\alpha \in \Sigma_X$ such that $\alpha \upharpoonright A \cup B = \tau \upharpoonright A \cup B$ and $\alpha \upharpoonright B \cup C = \rho \upharpoonright B \cup C$, then $\alpha \upharpoonright A \cup B \cup C = \sigma \upharpoonright A \cup B \cup C$, contradiction

“ \Rightarrow ”:

Suppose $A \cap X = \emptyset$ or $C \cap X = \emptyset$, or $X \not\subseteq A \cup B \cup C$. We show ABC .

Case 1: $C \cap X = \emptyset$. Let $\sigma, \tau \in \Sigma_X$ such that $\sigma \upharpoonright B = \tau \upharpoonright B$. As $C \cap X = \emptyset$, we can continue $\sigma \upharpoonright A \cup B$ as we like.

Case 2, $A \cap X = \emptyset$, analogous.

Case 3: $X \not\subseteq A \cup B \cup C$. But then there is no restriction in $A \cup B \cup C$.

□

We will have to make abg false, but agb true. On the other hand, we will make abd_1 false, but ad_1b need not be preserved.

This leads to the following definition, which helps to put order into the cases.

Definition 3.6

Suppose we have to destroy axy . Then

$dmin(axy) := \min\{d(\{a, x, y\}, \{a, u, v\}) : auv \text{ has to be preserved}\}$ - d the counting Hamming distance.

Thus, $dmin(abg) = 0$ (as agb has to be preserved), $dmin(abd_1) = 1$ (because abc has to be preserved, but not ad_1b).

We introduce the following order defined from the loop prerequisites to be preserved.

Definition 3.7

Order the elements by following the string of sequences to be preserved as follows:

$d_{i+1} \prec d_{i+2} \prec \dots \prec d_{n-1} \prec d_n \prec g \prec b \prec c \prec d_1 \prec d_2 \prec \dots \prec d_{i-1} \prec d_i$

Note that the interruption at $ad_i d_{i+1}$ is crucial here - otherwise, there would be a cycle.

As usual, \preceq will stand for \prec or $=$.

3.3.4 The cases to consider

The elements to consider are: $a, b, c, g, d_1, \dots, d_n$.

The tripels to preserve are:

$P := \{abc, acd_1, ad_1d_2, \dots, ad_{i-1}d_i, \text{ (BUT NOT } ad_i d_{i+1}), ad_{i+1}d_{i+2}, \dots, ad_{n-1}d_n, ad_n g, agb\}$

The $\langle X | Y | Z \rangle$ to destroy are (except when $X = \emptyset$ or $Z = \emptyset$):

- (1) all $\langle X || Z \rangle$
- (2) all $\langle X | Y | Z \rangle$ such that $X \cup Y \cup Z$ has > 3 elements
- (3) all tripels which do not have a on the outside, e.g. bgc
- (4) and the following tripels:
 (the (0) will be explained below - for the moment, just ignore it)
 abg (0), abd_1, \dots, abd_n
 acb (0), acg, acd_2, \dots, acd_n
 agc, agd_1, \dots, agd_n (0)
 ad_1b, ad_1c (0), $ad_1g, ad_1d_3, \dots, ad_1d_n$
 $ad_2b, ad_2c, ad_2g, ad_2d_1$ (0), ad_2d_4, \dots, ad_2d_n

 $ad_i b, ad_i c, ad_i g, ad_i d_1, ad_i d_2, \dots, \text{ ALSO } ad_i d_{i+1}, \dots, ad_i d_n$

 $ad_{n-1}b, ad_{n-1}c, ad_{n-1}g, ad_{n-1}d_1, \dots, ad_{n-1}d_{n-2}$ (0)
 $ad_n b, ad_n c, ad_n d_1, \dots, ad_n d_{n-1}$ (0)

3.3.5 Solution of the cases

We show how to destroy all tripels mentioned above, while preserving all tripels in P .

- (1) all $\langle X | Y | Z \rangle$ where $X \cup Y \cup Z$ has > 3 elements:

See Fact 3.5 (page 9) with the X there with 4 elements, for all such X, Y, Z separately, so all tripels in P are preserved.

- (2) all $\langle X | Y | Z \rangle$ with 1 element: -

- (3) all $\langle X || Z \rangle$:

This can be done by considering $\Sigma_j := \{0_c, 1_c\}$. Then, say for a, c , we have to examine the fragments 00 and 11, but there is no 10 or 01. For $\langle a | b | c \rangle$ this is no problem, as we have only the two 000, 111, which do not agree on b .

- (4) all $\langle X | Y | Z \rangle$ with 2 elements: eliminated by $\langle X || Z \rangle$

- (5) all $\langle X | Y | Z \rangle$ with 3 elements:

- (5.1) a is not on the outside

(5.1.1) a is in the middle, we need $\neg xay$: Consider Σ with 2 functions, 0_c , and the second defined by $a = 0$, and all $u = 1$ for $u \neq a$. Obviously, $\neg xay$. Recall that all tripels to be preserved have a on the outside, and some other element x in the middle. Then the two functions are different on x .

(5.1.2) a is not in xyz , we need $\neg xyz$: Consider Σ with 2 functions, 0_c , and the second defined by $a = y = 0$, all $u = 1$ for $u \neq a, u \neq y$. As a is neither x nor z , $\neg xyz$. If some uvw has a on the outside, say $u = a$, then both functions are 000 or 0vw on this tripel, so uvw holds.

- (5.2) a is on the outside, we destroy ayz :

- (5.2.1) Case $dmin(ayz) > 0$:

Take as Σ the set of all functions with values in $\{0, 1\}$, but eliminate those with $a = y = z = 0$. Then $\neg ayz$ (we have 100, 001, 101, but not 000), but for all auv with $d(\{a, y, z\}, \{a, u, v\}) > 0$ auv has all possible combinations, as all combinations for ay and az exist.

- (5.2.2) Case $dmin(ayz) = 0$.

The elements with $dmin = 0$ are:

$abg, acb, agd_n, ad_1c, ad_2d_1, \dots, ad_id_{i-1}$, NOT $ad_{i+1}d_i, ad_{i+2}d_{i+1}, \dots, ad_nd_{n-1}$ they were marked with (0) above.

Σ will again have 2 functions, the first is always 0_c .

The second function: Always set $a = 1$.

We see that the tripels with $dmin = 0$ to be destroyed have the form ayz , where z is the immediate \prec -predecessor of y in above order - see Definition 3.7 (page 10). Conversely, those to be preserved (in P) have the form azy , where again z is the immediate \prec -predecessor of y .

We set $z' = 1$ for all $z' \preceq z$, and $y' = 0$ for all $y' \succeq y$. Recall that $z \prec y$, so we have the picture $d_{i+1} = 1, \dots, z = 1, y = 0, \dots, d_i = 0$.

Then $\neg ayz$, as we have the fragments 000, 101. But azy , as we have the fragments 000, 110. Moreover, considering the successors of the sequence, we give the values 11, or 10, or 00. This results in the function fragments for auv as 111, or 110, or 100. But the resulting fragment sets (together with 0_c) are then: $\{000, 111\}, \{000, 110\}, \{000, 100\}$. They all make auv true. Thus, all tripels in P are preserved.

References

[Pea88] J. Pearl, "Probabilistic Reasoning in Intelligent Systems", Morgan Kaufmann, San Mateo, Cal., 1988