

On nesting of a nonmonotonic conditional

Citation for published version (APA):

Jurjus, H. (1995). *On nesting of a nonmonotonic conditional*. (Computing science reports; Vol. 9529). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/1995

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Eindhoven University of Technology
Department of Mathematics and Computing Science

On Nesting of a Nonmonotonic Conditional

by

Herman Jurjus

95/29

ISSN 0926-4515

All rights reserved

editors: prof.dr. J.C.M. Baeten
prof.dr. M. Rem

Computing Science Report 95/29
Eindhoven, October 1995

On Nesting of a Nonmonotonic Conditional

Herman Jurjus*

October 11, 1995

Abstract

The aim of this paper is to use (very) elementary topology to gain insights about nonmonotonic reasoning. In Section 0 it is proved that for finite spaces our approach is equivalent to the partial order based approach as found in [KLM 90]. Section 1 deals with nested conditional phrases with at least one interesting result, namely that it is possible to think of the monotonicity rule as valid in our (nonmonotonic) system, but valid only as a rule with possible exceptions. In Section 2, it is shown how the AGM postulates for revision are too strong, pointing in the direction of one postulate in particular.

0 Preliminaries

The reader of this paper is assumed to be familiar with (only) elementary notions from topology, such as "topology on a set", "closed set", "dense set", "induced topology on a subset", etc. (Say the first few pages of any elementary course on topology, for example [F&R 84]).

Note: In this paper variable names like O , O' etc. are used only for open sets. $P(X)$ denotes the power set of a set X .

0.1

Let (X, τ) be a topological space (τ is the collection of open sets).

Definition 0.1 A subset "a" of X is called full (in X) if every nonempty open subset of X contains a nonempty open subset of X which is also a subset of a.

[This is equivalent both to: "a^c is nowhere dense in X " and to: "the interior of a is dense in X " and is considered to be a possible interpretation of the phrase: "almost all elements of X are in a".]

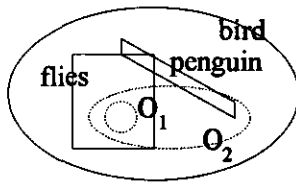
Example In the Euclidean plane, the complement of any line is a full subset; we intend to think about that as: given a line, almost all points in the plane are not on the line.

Definition 0.2 For $a, b \subseteq X$, we say "b is full in a" and write " $a \rightarrow b$ " if $a \cap b$ is a full subset of a with induced topology.

[This is equivalent to: "for all (open) O with $O \cap a \neq \emptyset$ there is an $O' \subseteq O$ with $O' \cap a \neq \emptyset$ and $O' \cap a \subseteq b$ ", and is considered to be a possible interpretation of the phrase "almost all elements of a are in b".]

* Department of Philosophy, Tilburg University, the Netherlands, email: H.H.Jurjus@kub.nl

Example Let X be some finite set, $\tau := \{\emptyset, O_1, O_2, X\}$, with $\emptyset \subset O_1 \subset O_2 \subset X$ (strict subsets). Instead of " X ", read "bird". Let "flies" and "penguin" be subsets of "bird" as in the picture below (sets that appear to be nonempty in the picture are supposed to be nonempty).



Then: $\text{bird} \rightarrow \text{flies}$
 $\text{bird} \cap \text{penguin} \not\rightarrow \text{flies}$
 $\text{bird} \cap \neg \text{flies} \not\rightarrow \emptyset$ (there are non-flying birds)
 $\text{penguin} \rightarrow \neg \text{flies}$
 $\text{bird} \rightarrow \neg \text{penguin}$
 $\text{penguin} \cap \text{flies} \not\rightarrow \emptyset$ (there are exceptional flying penguins)

(Where \neg denotes complement.)

Proposition 0.3 For all topological spaces (X, τ) and all $a, b, c \subseteq X$:

- $a \rightarrow a$
- if $a \rightarrow b$ and $b \subseteq c$ then $a \rightarrow c$
- if $a \rightarrow b$ and $a \cap b \rightarrow c$ then $a \rightarrow c$
- if $a \rightarrow b$ and $a \rightarrow c$ then $a \cap b \rightarrow c$
- if $a \rightarrow c$ and $b \rightarrow c$ then $a \cup b \rightarrow c$
- $a \rightarrow \emptyset$ iff $a = \emptyset$

And as a consequence:

- if $a \subseteq b$ then $a \rightarrow b$
- if $a \leftrightarrow b$ (that is: $a \rightarrow b, b \rightarrow a$) and $a \rightarrow c$, then $b \rightarrow c$
- if $a \rightarrow b, a \rightarrow c$ then $a \rightarrow b \cap c$
- if $a \rightarrow b^c \cup c$ and $a \rightarrow b$ then $a \rightarrow c$
- if $a \cap b \rightarrow c$ then $a \rightarrow b^c \cup c$
- $a \rightarrow b$ iff $a \rightarrow a \cap b$

(Proof: Elementary check.)

⊠

Proposition 0.4 NOT true (in general) are:

- if $a \rightarrow b$ then $a \subseteq b$
- if $a \rightarrow b$ then $a \cap c \rightarrow b$
- if $a \rightarrow b$ and $b \rightarrow c$ then $a \rightarrow c$
- if $a \rightarrow b$ then $b^c \rightarrow a^c$
- if $a \rightarrow b^c \cup c$ then $a \cap b \rightarrow c$

(Proof: Elementary check.)

⊠

Remark Readers familiar with the paper [KLM 90] will immediately recognize this set of properties as their "system P". They would write " \vdash " instead of " \rightarrow ", and call this relation a "preferential consequence relation".

Our topological presentation allows us to state some "additional" properties. As mentioned above, $a \rightarrow b$ does not imply $a \cap c \rightarrow b$. However:

Proposition 0.5 If $a \rightarrow b$ and O is open, then $a \cap O \rightarrow b$.

(Proof: Elementary check.)

Proposition 0.6 If $a \rightarrow b$, $d \subseteq X$ and d is dense in a , then $a \cap d \rightarrow b$.

(If $a, b \subseteq X$ then b is called *dense in a*, if $a \cap b$ is dense in a with induced topology. This is equivalent to: (for all open O) $O \cap a \neq \emptyset$ implies $O \cap a \cap b \neq \emptyset$.

For example, \mathbb{Q} is dense in \mathbb{R} , as is \mathbb{Q}^c . The open interval $(0,1)$ is not dense in \mathbb{R} , but it is dense in the closed interval $[0,1]$.)

Proof of Proposition 0.6: Suppose $a, b, d \subseteq X$, $a \rightarrow b$ and d is dense in a .

If $p \in a \cap d$ and $p \in O$ then there is an $O' \subseteq O$ with $O' \cap a \neq \emptyset$ and $O' \cap a \subseteq b$.

Since d is dense in a , $O' \cap a \cap d \neq \emptyset$ (and $O' \cap a \cap d \subseteq b$).

Hence $a \cap d \rightarrow b$.

⊠

Remark If (X, τ) is a topological space, then the following properties are equivalent:

i) for all $a, b, c \subseteq X$, $a \rightarrow b$ implies $a \cap c \rightarrow b$, ("(X, τ) is monotonic")

ii) for all $a, b \subseteq X$, $a \rightarrow b$ implies $a \subseteq b$.

(If ii) is not true, think about $a \cap b^c$.)

0.2 Finite spaces and partial order based semantics

In this section X is a finite set; every topology τ on X is said to *induce a conditional* on X (i.e. some binary relation on $P(X)$) according to the definitions in part 1, as follows: (for $a, b \subseteq X$:)

$a \rightarrow_\tau b$ iff $a \rightarrow b$ in (X, τ) .

Every partial ordering \leq on X induces a conditional on X as follows:

$a \rightarrow_\leq b$ iff all \leq -minimal elements of a are in b .

[$p \in X$ is called \leq -minimal in a if: i) $p \in a$, and ii) $q \in a$, $q \leq p$ implies $q = p$.]

Proposition 0.7 For every partial ordering on X there is a topology on X inducing the same conditional as that ordering.

Proof: Let (X, \leq) be a partial ordering.

Define $\tau := \{v \subseteq X \mid \text{if } p \in v \text{ and } q \leq p \text{ then } q \in v\}$ (which is a topology on X).

Then (for all $a, b \subseteq X$:) $a \rightarrow_\tau b$ iff $a \rightarrow_\leq b$:

[For every $p \in X$, let O_p denote $\{q \in X \mid q \leq p\}$.]

1) Suppose $a \rightarrow_\tau b$.

If p is \leq -minimal in a , then $O_p \cap a = \{p\}$.

Now there is an $O' \subseteq O_p$ with

$O' \cap a \neq \emptyset$, hence $O' \cap a = \{p\}$

and : $O' \cap a \subseteq b$, hence $p \in b$.

Hence $a \rightarrow_\leq b$.

2) Suppose $a \rightarrow_\leq b$.

If $p \in a$ and $p \in O$ then there is a q with $q \leq p$ and q \leq -minimal in a .

For such q : $O_q \subseteq O_p \subseteq O$ and $q \in b$ (since $a \rightarrow_\leq b$).

Hence: $O_q \subseteq O$, $O_q \cap a = \{q\} \neq \emptyset$, and $O_q \cap a \subseteq b$.

Hence $a \rightarrow_\tau b$.

⊠

Proposition 0.8 For every topology on X there is a partial ordering on X inducing the same conditional as that topology.

Proof: Let (X, τ) be a topological space.

For every $p \in X$, define O_p to be the smallest open set containing p . Now define the following strict partial ordering on X : (for $p, q \in X$):

$$p < q \text{ iff } O_p \subset O_q \quad (\text{strict subset})$$

and the associated ordinary partial ordering:

$$p \leq q \text{ iff } p < q \text{ or } p = q.$$

1) Suppose $a \rightarrow_\tau b$.

Let p be \leq -minimal in a .

There is an O with $O \subseteq O_p$, $O \cap a \neq \emptyset$, $O \cap a \subseteq b$.

Say $q \in O \cap a$, then $O_q \subseteq O_p$ and $O_q \cap a \subseteq b$.

Since p is \leq -minimal in a , $O_q = O_p$, hence $O_p \cap a \subseteq b$, hence $p \in b$.

Hence $a \rightarrow_\zeta b$.

2) Suppose $a \rightarrow_\zeta b$.

If $p \in a$, $p \in O$ then there is a q with $q \leq p$ and $q \leq$ -minimal in a .

For such q : if $r \in O_q \cap a$ then $O_r \subseteq O_q$, hence $O_r = O_q$ by the \leq -minimality (in a) of q .

Hence r is \leq -minimal in a , hence $r \in b$ (since $a \rightarrow_\zeta b$).

Hence $O_q \cap a \subseteq b$, and $O_q \cap a \neq \emptyset$ (since $q \in O_q \cap a$).

Hence $a \rightarrow_\tau b$.

⊠

Remarks 1) Proposition 0.8 obviously does not hold for infinite spaces in general. Think, for instance, about the Euclidean-plane example.

2) Given a conditional, there is at most one partial ordering inducing that conditional. On the other hand, a topology is in general not uniquely determined by the conditional that it induces. For example: if O_1 and O_2 are two disjoint, non-empty subsets of a set X , then the topology $\{\emptyset, O_1, O_2, O_1 \cup O_2, X\}$ induces the same conditional on X as the topology $\{\emptyset, O_1 \cup O_2, X\}$.

Conclusion of Section 0 For finite spaces, semantics using topology is the same thing as semantics using partial orderings, as found in [KLM 90]. For infinite spaces, our topological definitions are a natural generalization of the finite case.

1 Nesting

So far, we gave simple conditional phrases of the form " $a \rightarrow b$ " a non-standard interpretation. We will now investigate several possible ways to give nested conditional phrases an interpretation using the very same idea. Moreover, since rules of inference are implicational phrases by their very nature, it is only natural and consequent to use the new interpretation for inference rules as well.

Definition 1.1 An *implication* (\rightarrow -operator) on a topological space (X, τ) is a binary operation $I(_, _)$ on $P(X)$ satisfying

$$I(a, b) = X \text{ iff } a \rightarrow b \quad (\text{for all } a, b \subseteq X).$$

Let L be a propositional language closed under $\wedge, \vee, \rightarrow$ and \neg , having (finitely many) basic formulas as well as \top (for "true") and \perp (for "false").

Definition 1.2 A tuple (X, τ, φ) is called a *topological model based on I* if :

- (X, τ) is a topological space,
- $I(_, _)$ is an implication on X ,
- φ is a map $L \rightarrow P(X)$ such that (for all $a, b \in L$:)
- $\varphi(a \wedge b) = \varphi(a) \cap \varphi(b)$,
- $\varphi(a \vee b) = \varphi(a) \cup \varphi(b)$,
- $\varphi(\neg a) = \varphi(a)^c$,
- $\varphi(a \rightarrow b) = I(\varphi(a), \varphi(b))$,
- $\varphi(\top) = X$,
- $\varphi(\perp) = \emptyset$.

We will freely speak about "model (X, τ) ", "model X (based on I)" etc.

Definition 1.3 If $a \in L$ and $X = (X, \tau, \varphi)$ is a model, then

$$X \models a \text{ (" } a \text{ is true in } X \text{ ") iff } \varphi(a) = X$$

1.1 Nesting in finite spaces

Let (X, τ) be a finite topological space, and $a, b \subseteq X$.

Definition 1.4 $S(a, b)$ is the union of all $S \subseteq X$ with the property: $S \cap a \rightarrow b$.

Since $S_1 \cap a \rightarrow b$, $S_2 \cap a \rightarrow b$ implies $(S_1 \cup S_2) \cap a \rightarrow b$, and since X is finite, we have:

Proposition 1.5 (For all $a, b, c \subseteq X$:)

- $S(a, b) \cap a \rightarrow b$,
- $c \cap a \rightarrow b$ implies $c \rightarrow S(a, b)$.
- (it even implies $c \subseteq S(a, b)$)

corollary $S(a, b)$ is the largest $S \subseteq X$ with the property: $S \cap a \rightarrow b$.

corollary $S(a, b) = X$ iff $a \rightarrow b$ (i.e. $S(_, _)$ is an implication operator).

Furthermore it is easy (but important) to see:

$$a^c \cup b \subseteq S(a, b) \quad (\text{since } a^c \cap a \rightarrow b \text{ and } b \cap a \rightarrow b).$$

Let A_0, A_1, \dots, A_n be formulae from L .

Definition 1.6 We call

$$\frac{A_1, \dots, A_n}{A_0}$$

S-valid if $A_1 \wedge \dots \wedge A_n \rightarrow A_0$ is true in all finite models based on S .

It is called *strictly valid*, or *valid in the strict sense*, if for all such models :

$$X \models A_1 \wedge \dots \wedge A_n \text{ implies } X \models A_0$$

For example:

Proposition 1.7

Both $\frac{A \rightarrow B}{\neg A \vee B}$ and $\frac{\neg A \vee B}{A \rightarrow B}$ are *S-valid* rules.

Proof: 1) If X is a topological space, $a, b \subseteq X$ and $S = S(a, b)$ then $S \cap a \rightarrow b$, hence $S \rightarrow a^c \cup b$ (see Proposition 0.3). Hence $S(a, b) \rightarrow a^c \cup b$.

2) See the remark after the corollaries of Proposition 1.5.

⊠

Note that, since $a \rightarrow b$ does not imply $a \subseteq b$, $S(a,b) = X$ does not imply $a^c \cup b = X$, hence of the rules in the proposition, the latter is valid in the strict sense, the former is not.

Another example:

Theorem 1.8

$$\text{(mon)} \quad \frac{A \rightarrow B}{A \wedge C \rightarrow B}$$

is S-valid.

Proof: If $S = S(a,b)$ then $S \cap a \rightarrow b$, hence (for all $c \subseteq X$)

$$S \rightarrow a^c \cup b \subseteq (a \cap c)^c \cup b \subseteq S(a \cap c, b).$$

Hence $S(a,b) \rightarrow S(a \cap c, b)$.

⊠

Hence the rule of monotonicity is S-valid. But our system is non-monotonic, since $a \rightarrow b$ does *not* imply $a \cap c \rightarrow b$. What to think of such a result? Well, think of X as the set of possible worlds.

For $a, b \subseteq X$, read $a \rightarrow b$ as: "almost all worlds in a are in b", or: "a implies b (only) as a rule". The result now implies: if $a \rightarrow b$ is true and $c \subseteq X$, then there is a set $S \subseteq X$ (namely $S(a \cap c, b)$) such that

$$S \cap a \cap c \rightarrow b$$

$$\text{and } X \rightarrow S.$$

That is, under some extra condition "S", $a \cap c$ implies b (as a rule), and almost all worlds satisfy condition "S". Or: " $a \cap c \rightarrow b$ is true for almost all worlds". Hence in some way one could say that (mon) is a rule that might have exceptions, but that nevertheless is valid *as a rule*. ("The exception confirms the rule".)

Proposition 1.9 The following rules are S-valid:

$$\frac{A \rightarrow B}{\neg B \rightarrow \neg A}$$

$$\frac{A \wedge B \rightarrow C}{A \rightarrow (B \rightarrow C)}$$

$$\frac{A \rightarrow (B \rightarrow C)}{A \wedge B \rightarrow C}$$

Proof:

1) If $S = S(a,b)$ then $S \cap a \rightarrow b$, hence $S \rightarrow a^c \cup b = (b^c)^c \cup a^c \subseteq S(b^c, a^c)$.

Hence $S(a,b) \rightarrow S(b^c, a^c)$.

2) If $S = S(a \cap b, c)$ then $S \cap a \cap b \rightarrow c$, hence $S \cap a \rightarrow b^c \cup c$ (Proposition 0.5).

But $b^c \cup c \subseteq S(b,c)$, hence $S \cap a \rightarrow S(b,c)$. Hence $S \rightarrow S(a, S(b,c))$.

3) If $S = S(a, S(b,c))$ then $S \cap a \rightarrow S(b,c)$, hence $S \rightarrow a^c \cup S(b,c) \subseteq S$. So we have:

$$S \leftrightarrow a^c \cup S(b,c), \text{ and}$$

$$a^c \cup S(b,c) \rightarrow S(a \cap b, c) \text{ (because: } a^c \subseteq (a \cap b)^c \subseteq S(a \cap b, c) \text{ and}$$

$$S(b,c) \rightarrow S(a \cap b, c) \text{ by Theorem 1.8)}$$

Hence $S \rightarrow S(a \cap b, c)$ by Proposition 0.5.

⊠

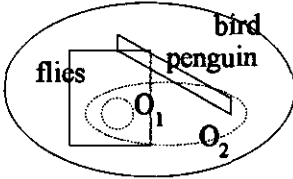
Of these rules, only the second is valid in the strict sense. There are also rules that are valid in the strict sense, but are not S-valid:

Proposition 1.10

$$\frac{A \rightarrow B, A \rightarrow C}{A \rightarrow B \wedge C}$$

is not S-valid.

Proof / Counterexample: Using an example from Section 0 :



Then:

$$S(\text{bird}, \text{flies}) = X$$

$$S(\text{bird}, \text{penguin}) = \text{penguin} \cup O_2^c$$

$$S(\text{bird}, \text{flies} \cap \text{penguin}) = \text{flies} \cap \text{penguin}$$

$$\text{but not } \text{penguin} \cup O_2^c \rightarrow \text{flies} \cap \text{penguin}.$$

Hence $S(\text{bird}, \text{flies}) \cap S(\text{bird}, \text{penguin}) \rightarrow S(\text{bird}, \text{flies} \cap \text{penguin})$ is not true.

⊠

Remarks

- 1) $\frac{A, A \rightarrow B}{B}$ is S-valid, $\frac{A, A \rightarrow B, \neg B}{B}$ and $\frac{A, A \rightarrow B, C}{B}$ are not.

(Stated without proof.)

2) $S(_, _)$ is, for finite spaces, the most natural candidate to use. However, it has some annoying properties as well (e.g. Proposition 1.10), and the idea is of little use for infinite spaces. Note that the definition of $S(_, _)$ doesn't really use our topological presentation, since for finite spaces our approach is equivalent to the approach using partial orderings. In the next section we will define, using the topology, a less natural operator, but one with better behaviour. The main result will be a generalization of Theorem 1.8 for all spaces (not only finite ones), in combination with the validity of most rules of ordinary proposition logic, including the one in Proposition 1.10.

1.2 Nesting in general

Let (X, τ) be a topological space and $a, b \subseteq X$.

Definition 1.11 $O(a,b)$ is the union of all (open) $O \subseteq X$ with the property: $O \cap a \rightarrow b$.

$O(a,b)$ is an open set for every a, b , and we have:

Proposition 1.12 $O(a,b) \cap a \rightarrow b$.

Proof: Suppose $O \subseteq X$ (open) and $O \cap (O(a,b) \cap a) \neq \emptyset$.

Say $p \in O \cap (O(a,b) \cap a)$, hence $p \in O(a,b)$. Say $p \in O_1$, and $O_1 \cap a \rightarrow b$.

Then $O \cap (O_1 \cap a) \neq \emptyset$. Hence there is an $O' \subseteq O$ with

$$O' \cap (O_1 \cap a) \neq \emptyset \text{ and } O' \cap (O_1 \cap a) \subseteq b.$$

Hence $O'' := O' \cap O_1$ satisfies $O'' \cap a \neq \emptyset$ and $O'' \cap a \subseteq b$.

Hence $O'' \cap a \rightarrow b$, hence $O'' \subseteq O(a,b)$.

Hence: $O'' \cap (O(a,b) \cap a) = O'' \cap a \subseteq b$ and $O'' \cap (O(a,b) \cap a) = (O' \cap O_1) \cap a \neq \emptyset$.

⊠

corollary $O(a,b)$ is the largest (open set) O satisfying $O \cap a \rightarrow b$.

corollary $a \rightarrow b$ iff $O(a,b) = X$ (i.e. $O(_, _)$ is an implication operator).

[(Stated without proof:) $O(_, _)$ does, in general, *not* satisfy $a^c \cup b \subseteq O(a,b)$, and $c \cap a \rightarrow b$ does *not* imply $c \rightarrow O(a,b)$]

The definitions of validity (Definition 1.6) are used mutatis mutandis:

Let A_0, A_1, \dots, A_n be formulae from L .

Definition 1.13

$$\frac{A_1, \dots, A_n}{A_0}$$

is called *O-valid* if $A_1 \wedge \dots \wedge A_n \rightarrow A_0$ is true in all models based on $O(_, _)$.

It is called *valid in the strict sense*, if for all such models :

$$X \models A_1 \wedge \dots \wedge A_n \text{ implies } X \models A_0$$

Proposition 1.14

$$\frac{A \rightarrow B}{A \wedge C \rightarrow B}$$

is *O-valid*.

Proof: Suppose $O = O(a,b)$ and $O \rightarrow O(a \cap c, b)$ is not true. Then $O(a \cap c, b)^c$ is somewhere dense in O , say $O' \subseteq O$, $O' \neq \emptyset$, and $O(a \cap c, b)^c$ is dense in O' . Then:

$$O' \subseteq O(a \cap c, b)^c \text{ (since the right hand side is closed and dense in } O')$$

i.e.: $O' \cap O(a \cap c, b) = \emptyset$.

This implies:

$a \cap c$ is dense in O' and

c is dense in $O' \cap a$.

By Proposition 0.5, $O' \cap a \rightarrow b$, hence by Proposition 0.6: $O' \cap a \cap c \rightarrow b$.

Hence $O' \subseteq O(a \cap c, b)$. Hence $O' = \emptyset$, contradiction.

⊠

As a corollary we have, for *all* topological spaces (X, τ) , $a, b \subseteq X$:

Theorem 1.15 If $a \rightarrow b$ is true, then there is a $v \subseteq X$ such that:

$$v \cap a \cap c \rightarrow b$$

v is a full subset

⊠

We could read that as: the monotonicity rule is valid *as a rule*, though it might have exceptions.

The same is true of the transitivity rule :

$$\frac{A \rightarrow C, C \rightarrow B}{A \rightarrow B}$$

(Sketch of the proof: $O(c,b) \rightarrow O(a \cap c, b)$ by Proposition 1.14, and $O(a,c) \cap O(a \cap c, b) \subseteq O(a,b)$.)

Theorem 1.16 The following rules are O-valid:

$$\begin{array}{c}
 \frac{C \rightarrow A, C \rightarrow B}{C \rightarrow A \wedge B} \\
 \\
 \frac{C \rightarrow A}{C \rightarrow A \vee B} \quad \frac{C \rightarrow B}{C \rightarrow A \vee B} \\
 \\
 \frac{C \wedge A \rightarrow B}{C \rightarrow \neg A \vee B} \\
 \\
 \frac{C \wedge A \rightarrow B, C \wedge A \rightarrow \neg B}{C \rightarrow \neg A} \\
 \\
 \frac{C \rightarrow A \wedge B}{C \rightarrow A} \quad \frac{C \rightarrow A \wedge B}{C \rightarrow B} \\
 \\
 \frac{C \rightarrow A \vee B, C \wedge A \rightarrow D, C \wedge B \rightarrow D}{C \rightarrow D} \\
 \\
 \frac{C \rightarrow A, C \rightarrow \neg A \vee B}{C \rightarrow B} \\
 \\
 \frac{C \rightarrow A, C \rightarrow \neg A}{C \rightarrow B}
 \end{array}$$

As well as:

$$\frac{C \rightarrow A}{C \wedge B \rightarrow A} \quad \frac{\top}{A \rightarrow A} \quad \frac{C \wedge A \wedge B \rightarrow D}{C \wedge B \wedge A \rightarrow D} \quad \frac{C \wedge A \wedge A \rightarrow B}{C \wedge A \rightarrow B} \quad \text{etc., etc.,}$$

Proof: (For X a topological space, $a, b, c, d \subseteq X$:)

It is easy to see that $O(c, a) \cap O(c, b) \subseteq O(c, a \cap b)$:

If $O_1 = O(c, a)$ and $O_2 = O(c, b)$ then $O_1 \cap O_2 \cap c \rightarrow a$ and $O_1 \cap O_2 \cap c \rightarrow b$ (Proposition 0.5).

Hence $O_1 \cap O_2 \cap c \rightarrow a \cap b$, hence $O_1 \cap O_2 \subseteq O(c, a \cap b)$.

Likewise:

$$O(c, a \cap b) \subseteq O(c, a), O(c, a \cap b) \subseteq O(c, b),$$

$$O(c, a) \subseteq O(c, a \cup b), O(c, b) \subseteq O(c, a \cup b),$$

$$O(c, a \cup b) \cap O(c \cap a, d) \cap O(c \cap b, d) \subseteq O(c, d),$$

$$[(O \cap c) \cap a \rightarrow d, (O \cap c) \cap b \rightarrow d \text{ implies } (O \cap c) \cap (a \cup b) \rightarrow d]$$

$$\text{and: } (O \cap c) \cap (a \cup b) \rightarrow d, (O \cap c) \rightarrow a \cup b \text{ implies } (O \cap c) \rightarrow d]$$

$$O(c \cap a, b) \subseteq O(c, a^c \cup b), \quad [v \cap a \rightarrow b \text{ implies } v \rightarrow a^c \cup b]$$

$$O(c, a^c \cup b) \cap O(c, a) \subseteq O(c, b), \quad [(a^c \cup b) \cap a \subseteq b]$$

$$O(c \cap a, b) \cap O(c \cap a, b^c) \subseteq O(c, a^c)$$

$$O(c, a) \cap O(c, a^c) \subseteq O(c, b).$$

The validity of the monotonicity rule is proved in Proposition 1.14, the rest is trivial.

⊠

Remarks 1) Note that under normal circumstances these are enough to determine ordinary classical proposition logic. Our system, however, is non-monotonic: $a \rightarrow b$ does *not* imply $a \cap c \rightarrow b$.

2) The rule of monotonicity is the only rule (in the theorem above) that is not valid in the strict sense.

3) The following variants of rules are O-valid as well:

$$\frac{C \rightarrow A, C \wedge A \rightarrow B}{C \rightarrow B} \quad \frac{C \wedge A \rightarrow D, C \wedge B \rightarrow D}{C \wedge (A \vee B) \rightarrow D}$$

(Proof is easy.)

Not O-valid are:

$$\text{i) } \frac{C \wedge A \rightarrow B}{C \rightarrow (A \rightarrow B)} \quad \text{ii) } \frac{A, A \rightarrow B, \neg B}{B} \quad \text{iii) } \frac{A \rightarrow B, A, C}{B}$$

Proof / Counterexamples: i) Let X be \mathbb{R} , $a = X$, $b = c = \mathbf{0}$. Then $O(c \cap a, b) = O(c, c) = X$, $O(a, b) = \emptyset$, $O(c, \emptyset) = \emptyset$. Hence $O(c \cap a, b) \not\rightarrow O(c, O(a, b))$.

ii) Choose X , a , b such that $a \rightarrow b$ is true in X and $a \cap b^c \neq \emptyset$, then

$O(a, b) = X$ and $O(a, b) \cap a \cap b^c \neq \emptyset$, but $O(a, b) \cap a \cap b^c \rightarrow b$ iff $O(a, b) \cap a \cap b^c \rightarrow \emptyset$.

iii) Choose X , a , b as in the preceding example, define c to be b^c , and proceed as before.

The non-validity of iii) could be thought of as: the information "bird", "green" and the rule that "bird" implies "flies" is not enough to conclude "flies". This is usually called the *irrelevancy problem*. It is a drawback common to most "conditional" approaches towards non-monotonic reasoning.

Conclusion of Section 1 The main result of this section is that, in some way, the monotonicity rule can be thought of as being valid in a nonmonotonic system like ours (be it valid only as a rule that might have exceptions).

Remark Using the topology on X , several other implication operators can easily be defined and investigated. In this paper however, we restrict our attention to $S(_, _)$ and $O(_, _)$.

2 Nesting and Theory Revision*

In this section (X, τ) is assumed to be a *finite, non-monotonic* space. (That is: there are $a, b, c \subseteq X$ with $a \rightarrow b$ but not $a \cap c \rightarrow b$. Or equivalently: there are $a, b \subseteq X$ with $a \rightarrow b$, but $a \not\subseteq b$.)

The word "implication" is used as before, i.e. an *implication* is a binary operation $I(_, _)$ satisfying

$$I(a, b) = X \text{ iff } a \rightarrow b \text{ (for all } a, b \subseteq X).$$

Now it is clear that $O(_, _)$ is an implication which does not satisfy the "deduction theorem"

$$(dt) \quad a \cap b \rightarrow c \text{ iff } a \rightarrow O(b, c).$$

(Proof: if $b \rightarrow c$ and *not* $a \cap b \rightarrow c$, then $O(b, c) = X$, hence $a \rightarrow O(b, c) \dots$)

This is a little bit disappointing, and nothing can be done about it, since the argument can be applied to any implication. Therefore, there is no alternative for $O(_, _)$ satisfying (dt).

However, there might be an alternative for the operator " \cap ".

Definition 2.1 A *combination* is a pair $(I, *)$ of binary operators on $P(X)$ with:

$$\begin{aligned} I(_, _) &\text{ is an implication on } X, \\ a * b \rightarrow c &\text{ iff } a \rightarrow I(b, c) \quad (\text{for all } a, b, c \subseteq X) \end{aligned}$$

One could think of $_ * _$ as a belief revision operator, cf. [Gär 92] or [F&L 94]. Here one should think of "a" as representing some theory, "b" as new information and "a*b" as the revised theory. In literature on revision, a central role is played by the *AGM postulates for revision*: a collection of proposed properties any revision operator ought to have. When translated into our terminology, these are:

* Sometimes also called 'belief revision'.

- K1 if a and b are subsets of X , $a*b$ is a subset of X ,
- K2 $a*b \subseteq b$,
- K3 $a \cap b \subseteq a*b$,
- K4 if $a \cap b \neq \emptyset$ then $a*b = a \cap b$,
- K5 if $a*b = \emptyset$ then $b = \emptyset$ or $a = \emptyset$,
- K6 if $b_1 = b_2$ then $a*b_1 = a*b_2$,
- K7 $(a*b_1) \cap b_2 \subseteq a*(b_1 \cap b_2)$,
- K8 if $(a*b_1) \cap b_2 \neq \emptyset$ then $a*(b_1 \cap b_2) = (a*b_1) \cap b_2$.

In our framework, K1 and K6 are trivially true, and our version of K1 is necessarily stronger than the original K1 (see, e.g., [F&L 94]). K5 has been modified as well, see remark 5) after Theorem 2.3.

Definition 2.2 A revision operator is said to satisfy the *Ramsey test* if it can be combined with some implication operator.

Let us now prove our version of the well known Gärdenfors impossibility theorem:

Theorem 2.3 There is no operator $*$ satisfying both the Ramsey test and K4.

Proof: Suppose $(I, *)$ is a combination, and $*$ satisfies : (for all $a, b, c \subseteq X$)

if $a \cap b \neq \emptyset$ then $a*b = a \cap b$.

Then (for all $p \in X$, $a, b \subseteq X$:)

$\{p\}*a \rightarrow b$ iff $p \in I(a, b)$

1) Now suppose $p \in I(a, b) \cap a$, then:

$\{p\} \cap a \neq \emptyset$, hence $\{p\}*a = \{p\} \cap a = \{p\}$.

But $\{p\}*a \rightarrow b$, hence $\{p\} \rightarrow b$, i.e. $p \in b$.

Hence $I(a, b) \cap a \subseteq b$, for all $a, b \subseteq X$.

2) Now $a \rightarrow b$ implies $I(a, b) = X$, hence $a \subseteq b$, contrary to our assumption at the beginning of this section.

⊠

Remarks 1) Theorem 2.3 sounds like an impossibility result, but it is in fact a triviality result:

if $(I, *)$ satisfies both the Ramsey test and K4 then the space (X, τ) is monotonic, hence not interesting.

2) Most researchers in the field of belief revision seem to accept without the slightest hesitation the principle that, as long as the new information is (classically) consistent with the knowledge so far, the revised theory should just be the classical expansion. This is the idea behind the AGM postulate K4.

3) There is no combination with $*$ satisfying:

if $a \not\rightarrow b^c$ then $a*b = a \cap b$

(proof is similar to the proof of Theorem 2.3, therefore skipped), which could be considered to be a better representation of the idea behind K4.

4) For stronger results, see, for example, [Gär 87], [Mak 90], [Mor 92].

[Gär 87] already blamed the combination Ramsey test + K4. (His version of the Ramsey test is slightly different from ours, however.) Some researchers blame the Ramsey test in particular, continuing research in the direction of the revision operator alone. Recent results [F&L 94] have shown that the AGM postulates are too strong for other reasons as well. [Mor 92] proved that the combination Ramsey test + K4 is possible in a different conceptual framework, hence blaming

technicalities. (For example, our version essentially needs $a*b$ to exist for $a = \{p\}$ for all p in X and all $b \subseteq X$. In view of [Mor 92] this is an unreasonably strong assumption.)

Nevertheless, we feel that K4 is to be blamed, or rather the general idea behind it, as sketched in remark 2) above. Our main argument will be Theorem 2.5 below.

5) There is no combination $(I, *)$ with $*$ satisfying $a*b \neq \emptyset$ whenever $b \neq \emptyset$, which is the direct translation of the original version of AGM postulate K5. (If $*$ is combinable, then $\emptyset*b = \emptyset$ for all $b \subseteq X$, since $\emptyset \rightarrow I(b, \emptyset)$ for all b .) This is why we have replaced the original postulate by our version.

Let us investigate the combinability of $S(_, _)$ and $O(_, _)$.

The implication operator $S(_, _)$ cannot be combined with any revision operator:

Theorem 2.4 There is no combination with I satisfying (for all $a, b, c \subseteq X$:)

$$\begin{aligned} I(a, b) \cap a &\rightarrow b \\ c \cap a &\rightarrow b \text{ implies } c \rightarrow I(a, b). \end{aligned}$$

Proof: Suppose $(I, *)$ is a combination and I satisfies those properties. Then:

1) $I(a, \emptyset) = a^c$ for all $a \subseteq X$:

$$\begin{aligned} \text{If } p \notin a, \text{ then } \{p\} \cap a &\rightarrow \emptyset, \text{ hence } \{p\} \rightarrow I(a, \emptyset), \text{ hence } p \in I(a, \emptyset) : a^c \subseteq I(a, \emptyset). \\ I(a, \emptyset) \cap a &\rightarrow \emptyset, \text{ hence } I(a, \emptyset) \subseteq a^c. \end{aligned}$$

2) $I(a, b) \cap a \subseteq b$ for all $a, b \subseteq X$:

$$\begin{aligned} \text{Suppose } p \in a \cap I(a, b). \text{ Then } \{p\} &\rightarrow I(a, b), \text{ hence } \{p\} * a \rightarrow b. \\ \text{Since } \{p\} \cap a &\rightarrow \{p\} \cap a : \{p\} \rightarrow I(a, \{p\} \cap a), \text{ hence } \{p\} * a \rightarrow \{p\} \cap a = \{p\}. \\ \text{Hence} \end{aligned}$$

$$\{p\} * a \cap \{p\} \rightarrow b.$$

$$I(a, \emptyset) = a^c, \text{ hence } p \notin I(a, \emptyset), \text{ hence } \textit{not} \{p\} * a \rightarrow \emptyset, \text{ hence } \{p\} * a \neq \emptyset.$$

$$\text{Now } \{p\} * a \rightarrow \{p\} \text{ implies}$$

$$p \in \{p\} * a.$$

$$\text{Hence } \{p\} * a \cap \{p\} = \{p\}, \text{ hence } \{p\} \rightarrow b, \text{ i.e. } p \in b.$$

3) The proof now proceeds like step 2) of Theorem 2.3.

⊠

Note that $S(_, _)$ does satisfy these conditions (Proposition 1.5).

But combinations do exist:

Theorem 2.5 If X is a *finite* space and I an implication operator on X satisfying (for all $a, b, c \subseteq X$:)

- (i) $I(a, b) \subseteq I(a, c)$ if $b \subseteq c$
- (ii) $I(a, b) \cap I(a, c) \subseteq I(a, b \cap c)$
- (iii) $I(a, b) \cap I(a \cap b, c) \subseteq I(a, c)$

then there is an operator $*$ such that $(I, *)$ is a combination.

Proof: Define, for $a, b \subseteq X$, $a*b$ to be the intersection of all $c \subseteq X$ with the property: $a \rightarrow I(b, c)$. Then, by i) and ii),

$$a*b \subseteq c \text{ iff } a \rightarrow I(b, c). \quad (\#)$$

We will prove: $a*b \rightarrow c$ iff $a*b \subseteq c$.

(We only prove the non-trivial part.) Suppose $a*b \rightarrow c$.

Then by (iii):

$I(b, a*b) \cap I(b \cap (a*b), c) \subseteq I(b, c)$.
 But $a*b \subseteq b$, (by (#))
 hence: $I(b \cap (a*b), c) = I(a*b, c) = X$, (since $a*b \rightarrow c$)
 hence: $I(b, a*b) \subseteq I(b, c)$. (by i))
 $a \rightarrow I(b, a*b)$, (by (#))
 hence: $a \rightarrow I(b, c)$,
 hence: $a*b \subseteq c$. (by (#))

⊠

Note that $O(_ , _)$ satisfies the three necessary conditions (see the proof of Theorem 1.16. It is easy to prove iii) using the same idea).

Remark There are revision operators satisfying K1, K2, K5 (that is, our version) K6 as well as the Ramsey test.

(Stated without proof : (for every X) there is an implication operator satisfying the three conditions of Theorem 2.5 and: $I(a, \emptyset) = \emptyset$ whenever $a \neq \emptyset$.)

Conclusions of Section 2

- 1) In this final section we investigated the implication operators $O(_ , _)$ and $S(_ , _)$, as defined in the previous section. The former is combinable with a revision operator, the latter is not.
- 2) The Ramsey test is a fair and plausible property for a revision operator to have, and should not be forgotten in the field of theory revision just because of the Gärdenfors-like Impossibility Theorems.
- 3) Several researchers agree about the AGM postulates being too strong, without agreement about a particular reason. Our set-up points in the direction of postulate K4 in particular.

Overall conclusions

- 1) It is possible to investigate non-monotonic reasoning using standard notions from elementary topology, without any reference to probability or preference of some possible worlds over others. For finite spaces our approach is equivalent to the partial order based approach as found in [KLM 90].
- 2) There are several possible ways to handle nested conditional phrases. We investigated a very natural one for finite spaces, and a slightly less natural one for the general case.
- 3) We used this "nesting" to interpret rules of inference, and it turned out that each of the standard rules of ordinary proposition logic, including the rule of monotonicity, is valid in this non-standard sense. In a strict sense, the system is non-monotonic however. This is a nice way to think of non-monotonic reasoning.

Directions of further research

- 1) Find a more convincing non-monotonic system that is based on the idea: "just accept (almost) each of the rules of ordinary logic, including the rule of monotonicity, but only as a rule that might have exceptions", hence creating a logic which is non-monotonic in the mathematical sense, but in which the rule of monotonicity is valid as a rule. This paper already presents such a system, but I'm looking for a more direct implementation of the idea.
- 2) Extend to predicate calculus.

3) Find a way to solve the irrelevancy problem (see the final remark after Theorem 1.16) using the general principle of 1) above.

References

- [Adams 74] E.W.Adams, *The Logic of "Almost all"*, Journal of Philosophical Logic 3 (1974), 3-17.
- [Adams 75] E.W.Adams, *The Logic of Conditionals*, Reidel, Netherlands, 1975.
- [F&L 94] M. Freund, D. Lehmann, *Belief revision and rational inference*, report of the Leibniz Center for Research in Computer Science, July 1994.
- [F&R 84] D.B. Fuks, V.A. Rokhlin, *Beginner's Course in Topology*, Springer 1984, page 5,6,7.
- [Gär 87] P. Gärdenfors, *Variations on the Ramsey test: more triviality results*, Studia Logica 46 (1987) 321-327.
- [Gär 92] P. Gärdenfors, *Belief Revision, An Introduction*, in: Belief Revision, ed. P. Gärdenfors, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1992, p 1-28.
- [KLM 90] S. Kraus, D. Lehmann, M. Magidor, *Nonmonotonic Reasoning, Preferential Models and Cumulative Logics*, Artificial Intelligence 44 (1990) 167-207.
- [L 92] D. Lehmann, *Plausibility Logic*; report of the Leibniz Center for Research in Computer Science, February 1992.
- [Mak 90] D. Makinson, *The Gärdenfors Impossibility Theorem in non-monotonic contexts*, Studia Logica 49 (1990) 1-6.
- [Mor 92] M. Morreau, *Epistemic Semantics for Counterfactuals*, Journal of Philosophical Logic 21 (1992) 33-62.

In this series appeared:

- 93/01 R. van Geldrop Deriving the Aho-Corasick algorithms: a case study into the synergy of programming methods, p. 36.
- 93/02 T. Verhoeff A continuous version of the Prisoner's Dilemma, p. 17
- 93/03 T. Verhoeff Quicksort for linked lists, p. 8.
- 93/04 E.H.L. Aarts
J.H.M. Korst
P.J. Zwietering Deterministic and randomized local search, p. 78.
- 93/05 J.C.M. Baeten
C. Verhoef A congruence theorem for structured operational semantics with predicates, p. 18.
- 93/06 J.P. Velkamp On the unavailability of metastable behaviour, p. 29
- 93/07 P.D. Moerland Exercises in Multiprogramming, p. 97
- 93/08 J. Verhoosel A Formal Deterministic Scheduling Model for Hard Real-Time Executions in DEDOS, p. 32.
- 93/09 K.M. van Hee Systems Engineering: a Formal Approach Part I: System Concepts, p. 72.
- 93/10 K.M. van Hee Systems Engineering: a Formal Approach Part II: Frameworks, p. 44.
- 93/11 K.M. van Hee Systems Engineering: a Formal Approach Part III: Modeling Methods, p. 101.
- 93/12 K.M. van Hee Systems Engineering: a Formal Approach Part IV: Analysis Methods, p. 63.
- 93/13 K.M. van Hee Systems Engineering: a Formal Approach Part V: Specification Language, p. 89.
- 93/14 J.C.M. Baeten
J.A. Bergstra On Sequential Composition, Action Prefixes and Process Prefix, p. 21.
- 93/15 J.C.M. Baeten
J.A. Bergstra
R.N. Bol A Real-Time Process Logic, p. 31.
- 93/16 H. Schepers
J. Hooman A Trace-Based Compositional Proof Theory for Fault Tolerant Distributed Systems, p. 27
- 93/17 D. Alstein
P. van der Stok Hard Real-Time Reliable Multicast in the DEDOS system, p. 19.
- 93/18 C. Verhoef A congruence theorem for structured operational semantics with predicates and negative premises, p. 22.
- 93/19 G-J. Houben The Design of an Online Help Facility for ExSpect, p.21.
- 93/20 F.S. de Boer A Process Algebra of Concurrent Constraint Programming, p. 15.
- 93/21 M. Codish
D. Dams
G. Filé
M. Bruynooghe Freeness Analysis for Logic Programs - And Correctness, p. 24
- 93/22 E. Poll A Typechecker for Bijective Pure Type Systems, p. 28.
- 93/23 E. de Kogel Relational Algebra and Equational Proofs, p. 23.
- 93/24 E. Poll and Paula Severi Pure Type Systems with Definitions, p. 38.
- 93/25 H. Schepers and R. Gerth A Compositional Proof Theory for Fault Tolerant Real-Time Distributed Systems, p. 31.
- 93/26 W.M.P. van der Aalst Multi-dimensional Petri nets, p. 25.
- 93/27 T. Kloks and D. Kratsch Finding all minimal separators of a graph, p. 11.
- 93/28 F. Kamareddine and R. Nederpelt A Semantics for a fine λ -calculus with de Bruijn indices, p. 49.
- 93/29 R. Post and P. De Bra GOLD, a Graph Oriented Language for Databases, p. 42.
- 93/30 J. Deogun
T. Kloks
D. Kratsch
H. Müller On Vertex Ranking for Permutation and Other Graphs, p. 11.

| | | |
|-------|--|---|
| 93/31 | W. Körver | Derivation of delay insensitive and speed independent CMOS circuits, using directed commands and production rule sets, p. 40. |
| 93/32 | H. ten Eikelder and H. van Geldrop | On the Correctness of some Algorithms to generate Finite Automata for Regular Expressions, p. 17. |
| 93/33 | L. Loyens and J. Moonen | ILLAS, a sequential language for parallel matrix computations, p. 20. |
| 93/34 | J.C.M. Baeten and J.A. Bergstra | Real Time Process Algebra with Infinitesimals, p.39. |
| 93/35 | W. Ferrer and P. Severi | Abstract Reduction and Topology, p. 28. |
| 93/36 | J.C.M. Baeten and J.A. Bergstra | Non Interleaving Process Algebra, p. 17. |
| 93/37 | J. Brunekreef J-P. Katoen R. Koymans S. Mauw | Design and Analysis of Dynamic Leader Election Protocols in Broadcast Networks, p. 73. |
| 93/38 | C. Verhoef | A general conservative extension theorem in process algebra, p. 17. |
| 93/39 | W.P.M. Nuijten E.H.L. Aarts D.A.A. van Erp Taalman Kip K.M. van Hee | Job Shop Scheduling by Constraint Satisfaction, p. 22. |
| 93/40 | P.D.V. van der Stok M.M.M.P.J. Claessen D. Alstein | A Hierarchical Membership Protocol for Synchronous Distributed Systems, p. 43. |
| 93/41 | A. Bijlsma | Temporal operators viewed as predicate transformers, p. 11. |
| 93/42 | P.M.P. Rambags | Automatic Verification of Regular Protocols in P/T Nets, p. 23. |
| 93/43 | B.W. Watson | A taxonomy of finite automata construction algorithms, p. 87. |
| 93/44 | B.W. Watson | A taxonomy of finite automata minimization algorithms, p. 23. |
| 93/45 | E.J. Luit J.M.M. Martin | A precise clock synchronization protocol.p. |
| 93/46 | T. Kloks D. Kratsch J. Spinrad | Treewidth and Patwidth of Cocomparability graphs of Bounded Dimension, p. 14. |
| 93/47 | W. v.d. Aalst P. De Bra G.J. Houben Y. Komatzky | Browsing Semantics in the "Tower" Model, p. 19. |
| 93/48 | R. Gerth | Verifying Sequentially Consistent Memory using Interface Refinement, p. 20. |
| 94/01 | P. America M. van der Kammen R.P. Nederpelt O.S. van Roosmalen H.C.M. de Swart | The object-oriented paradigm, p. 28. |
| 94/02 | F. Kamareddine R.P. Nederpelt | Canonical typing and Π -conversion, p. 51. |
| 94/03 | L.B. Hartman K.M. van Hee | Application of Markov Decision Processes to Search Problems, p. 21. |
| 94/04 | J.C.M. Baeten J.A. Bergstra | Graph Isomorphism Models for Non Interleaving Process Algebra, p. 18. |
| 94/05 | P. Zhou J. Hooman | Formal Specification and Compositional Verification of an Atomic Broadcast Protocol, p. 22. |
| 94/06 | T. Basten T. Kunz J. Black M. Coffin D. Taylor | Time and the Order of Abstract Events in Distributed Computations, p. 29. |
| 94/07 | K.R. Apt R. Bol | Logic Programming and Negation: A Survey, p. 62. |
| 94/08 | O.S. van Roosmalen | A Hierarchical Diagrammatic Representation of Class Structure, p. 22. |
| 94/09 | J.C.M. Baeten J.A. Bergstra | Process Algebra with Partial Choice, p. 16. |

| | | |
|-------|--|--|
| 94/10 | T. verhoeff | The testing Paradigm Applied to Network Structure. p. 31. |
| 94/11 | J. Peleska C. Huizing C. Petersohn | A Comparison of Ward & Mellor's Transformation Schema with State- & Activitycharts, p. 30. |
| 94/12 | T. Kloks D. Kratsch H. Müller | Dominoes, p. 14. |
| 94/13 | R. Seljée | A New Method for Integrity Constraint checking in Deductive Databases, p. 34. |
| 94/14 | W. Peremans | Ups and Downs of Type Theory, p. 9. |
| 94/15 | R.J.M. Vaessens E.H.L. Aarts J.K. Lenstra | Job Shop Scheduling by Local Search, p. 21. |
| 94/16 | R.C. Backhouse H. Doornbos | Mathematical Induction Made Computational, p. 36. |
| 94/17 | S. Mauw M.A. Reniers | An Algebraic Semantics of Basic Message Sequence Charts, p. 9. |
| 94/18 | F. Kamareddine R. Nederpelt | Refining Reduction in the Lambda Calculus, p. 15. |
| 94/19 | B.W. Watson | The performance of single-keyword and multiple-keyword pattern matching algorithms, p. 46. |
| 94/20 | R. Bloo F. Kamareddine R. Nederpelt | Beyond β -Reduction in Church's $\lambda \rightarrow$, p. 22. |
| 94/21 | B.W. Watson | An introduction to the Fire engine: A C++ toolkit for Finite automata and Regular Expressions. |
| 94/22 | B.W. Watson | The design and implementation of the FIRE engine: A C++ toolkit for Finite automata and regular Expressions. |
| 94/23 | S. Mauw and M.A. Reniers | An algebraic semantics of Message Sequence Charts, p. 43. |
| 94/24 | D. Dams O. Grumberg R. Gerth | Abstract Interpretation of Reactive Systems: Abstractions Preserving \forall CTL*, \exists CTL* and CTL*, p. 28. |
| 94/25 | T. Kloks | $K_{1,3}$ -free and W_4 -free graphs, p. 10. |
| 94/26 | R.R. Hoogerwoord | On the foundations of functional programming: a programmer's point of view, p. 54. |
| 94/27 | S. Mauw and H. Mulder | Regularity of BPA-Systems is Decidable, p. 14. |
| 94/28 | C.W.A.M. van Overveld M. Verhoeven | Stars or Stripes: a comparative study of finite and transfinite techniques for surface modelling, p. 20. |
| 94/29 | J. Hooman | Correctness of Real Time Systems by Construction, p. 22. |
| 94/30 | J.C.M. Baeten J.A. Bergstra Gh. Ştefănescu | Process Algebra with Feedback, p. 22. |
| 94/31 | B.W. Watson R.E. Watson | A Boyer-Moore type algorithm for regular expression pattern matching, p. 22. |
| 94/32 | J.J. Vereijken | Fischer's Protocol in Timed Process Algebra, p. 38. |
| 94/33 | T. Laan | A formalization of the Ramified Type Theory, p.40. |
| 94/34 | R. Bloo F. Kamareddine R. Nederpelt | The Barendregt Cube with Definitions and Generalised Reduction, p. 37. |
| 94/35 | J.C.M. Baeten S. Mauw | Delayed choice: an operator for joining Message Sequence Charts, p. 15. |
| 94/36 | F. Kamareddine R. Nederpelt | Canonical typing and Π -conversion in the Barendregt Cube, p. 19. |
| 94/37 | T. Basten R. Bol M. Voorhoeve | Simulating and Analyzing Railway Interlockings in ExSpect, p. 30. |
| 94/38 | A. Bijlsma C.S. Scholten | Point-free substitution, p. 10. |

| | | | |
|-------|--|--|--|
| 94/39 | A. Blokhuis T. Kloks | On the equivalence covering number of splitgraphs, p. 4. | |
| 94/40 | D. Alstein | Distributed Consensus and Hard Real-Time Systems, p. 34. | |
| 94/41 | T. Kloks D. Kratsch | Computing a perfect edge without vertex elimination ordering of a chordal bipartite graph, p. 6. | |
| 94/42 | J. Engelfriet J.J. Vereijken | Concatenation of Graphs, p. 7. | |
| 94/43 | R.C. Backhouse M. Bijsterveld | Category Theory as Coherently Constructive Lattice Theory: An Illustration, p. 35. | |
| 94/44 | E. Brinksma R. Gerth W. Janssen S. Katz M. Poel C. Rump | J. Davies S. Graf B. Jonsson G. Lowe A. Pruegli J. Zwiers | Verifying Sequentially Consistent Memory, p. 160 |
| 94/45 | G.J. Houben | Tutorial voor de ExSpect-bibliotheek voor "Administratieve Logistiek", p. 43. | |
| 94/46 | R. Bloo F. Kamareddine R. Nederpelt | The λ -cube with classes of terms modulo conversion, p. 16. | |
| 94/47 | R. Bloo F. Kamareddine R. Nederpelt | On Π -conversion in Type Theory, p. 12. | |
| 94/48 | Mathematics of Program Construction Group | Fixed-Point Calculus, p. 11. | |
| 94/49 | J.C.M. Baeten J.A. Bergstra | Process Algebra with Propositional Signals, p. 25. | |
| 94/50 | H. Geuvers | A short and flexible proof of Strong Normalization for the Calculus of Constructions, p. 27. | |
| 94/51 | T. Kloks D. Kratsch H. Müller | Listing simplicial vertices and recognizing diamond-free graphs, p. 4. | |
| 94/52 | W. Penczek R. Kuiper | Traces and Logic, p. 81 | |
| 94/53 | R. Gerth R. Kuiper D. Peled W. Penczek | A Partial Order Approach to Branching Time Logic Model Checking, p. 20. | |
| 95/01 | J.J. Lukkien | The Construction of a small CommunicationLibrary, p.16. | |
| 95/02 | M. Bezem R. Bol J.F. Groote | Formalizing Process Algebraic Verifications in the Calculus of Constructions, p.49. | |
| 95/03 | J.C.M. Baeten C. Verhoef | Concrete process algebra, p. 134. | |
| 95/04 | J. Hidders | An Isotopic Invariant for Planar Drawings of Connected Planar Graphs, p. 9. | |
| 95/05 | P. Severi | A Type Inference Algorithm for Pure Type Systems, p.20. | |
| 95/06 | T.W.M. Vossen M.G.A. Verhoeven H.M.M. ten Eikelder E.H.L. Aarts | A Quantitative Analysis of Iterated Local Search, p.23. | |
| 95/07 | G.A.M. de Bruyn O.S. van Roosmalen | Drawing Execution Graphs by Parsing, p. 10. | |
| 95/08 | R. Bloo | Preservation of Strong Normalisation for Explicit Substitution, p. 12. | |
| 95/09 | J.C.M. Baeten J.A. Bergstra | Discrete Time Process Algebra, p. 20 | |
| 95/10 | R.C. Backhouse R. Verhoeven O. Weber | Math \uparrow pad: A System for On-Line Preparation of Mathematical Documents, p. 15 | |

| | | |
|-------|---|---|
| 95/11 | R. Selj e | Deductive Database Systems and integrity constraint checking, p. 36. |
| 95/12 | S. Mauw and M. Reniers | Empty Interworkings and Refinement Semantics of Interworkings Revised, p. 19. |
| 95/13 | B.W. Watson and G. Zwaan | A taxonomy of sublinear multiple keyword pattern matching algorithms, p. 26. |
| 95/14 | A. Ponse, C. Verhoef, S.F.M. Vlijmen (eds.) | De proceedings: ACP95, p. |
| 95/15 | P. Niebert and W. Penczek | On the Connection of Partial Order Logics and Partial Order Reduction Methods, p. 12. |
| 95/16 | D. Dams, O. Grumberg, R. Gerth | Abstract Interpretation of Reactive Systems: Preservation of CTL*, p. 27. |
| 95/17 | S. Mauw and E.A. van der Meulen | Specification of tools for Message Sequence Charts, p. 36. |
| 95/18 | F. Kamareddine and T. Laan | A Reflection on Russell's Ramified Types and Kripke's Hierarchy of Truths, p. 14. |
| 95/19 | J.C.M. Baeten and J.A. Bergstra | Discrete Time Process Algebra with Abstraction, p. 15. |
| 95/20 | F. van Raamsdonk and P. Severi | On Normalisation, p. 33. |
| 95/21 | A. van Deursen | Axiomatizing Early and Late Input by Variable Elimination, p. 44. |
| 95/22 | B. Arnold, A. v. Deursen, M. Res | An Algebraic Specification of a Language for Describing Financial Products, p. 11. |
| 95/23 | W.M.P. van der Aalst | Petri net based scheduling, p. 20. |
| 95/24 | F.P.M. Dignum, W.P.M. Nuijten, L.M.A. Janssen | Solving a Time Tabling Problem by Constraint Satisfaction, p. 14. |
| 95/25 | L. Feijs | Synchronous Sequence Charts In Action, p. 36. |
| 95/26 | W.M.P. van der Aalst | A Class of Petri nets for modeling and analyzing business processes, p. 24. |
| 95/27 | P.D.V. van der Stok, J. van der Wal | Proceedings of the Real-Time Database Workshop, p. 106. |
| 95/28 | W. Fokkink, C. Verhoef | A Conservative Look at term Deduction Systems with Variable Binding, p. 29. |