Disjunctions and Specificity in Suppositional Defeasible Argumentation.

Article *in* Logic Journal of IGPL · January 2002

DOI: 10.1093/jigpal/10.1.23 · Source: DBLP

Gustavo Adrian Bodanza
Universidad Nacional del Sur

LOGIC JOURNAL of the IGPL

Volume 10 Number 1 January 2002

Interest Group in Pure and Applied Logics

OXFORD UNIVERSITY PRESS

ISSN 1367-0751

Subscription Information

Volume 10, 2002 (bimonthly) Full: Europe pounds sterling 275; Rest of World US$ 450. Personal: pounds sterling 138 (US$ 225). Please note that personal rates apply only when copies are sent to a private address and payment is made by a personal cheque or credit card.

Order Information

Subscriptions can be accepted for complete volumes only. Prices include air-speeded delivery to Australia, Canada, India, Japan, New Zealand, and the USA. Delivery elsewhere is by surface post. Payment is required with all orders and may be made in the following ways:

Cheque (made payable to Oxford University Press)
National Girobank (account 500 1056)
Credit card (Access, Visa, American Express)
UNESCO Coupons
Bankers: Barclays Bank plc, PO Box 333, Oxford, UK. Code 20-65-18, Account 00715654.

Requests for sample copies, subscription enquiries, orders and changes of address should be sent to the Journals Subscriptions Department, Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, UK. Tel: 01865 267907. Fax: 01865 267485.

Advertisements

Copyright

Logic Journal of the IGPL (ISSN 1367-0751) is published bimonthly in January, March, May, July, September and November by Oxford University Press, Oxford, UK. Annual subscription price is US$ 450.00. *Logic Journal of the IGPL* is distributed by M.A.I.L. America, 2323 Randolph Avenue, Avenel, NJ 07001. Periodical postage paid at Rahway, New Jersey, USA and at additional entry points.
US Postmasters: Send address changes to *Logic Journal of the IGPL*, c/o Mercury International, 365 Blair Road, Avenel, NJ 07001, USA.

Back Issues

The current plus two back volumes are available from Oxford University Press. Previous volumes can be obtained from the Periodicals Service Company, 11 Main Street, Germantown, NY 12526 USA. Tel: +1 (518) 537 4700, Fax: +1 (518) 537 5899.

Logic Journal of the IGPL

Volume 10, Number 1, January 2002

Contents

Please visit the journal's World Wide Web site at
http://www.jigpal.oupjournals.org

Logic Journal of the Interest Group in Pure and Applied Logics

Scope of the Journal

The *Journal* is the official publication of the International Interest Group in Pure and Applied Logics (IGPL), which is sponsored by The European Foundation for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, computational, mathematical, philosophical, etc.) from all over the world.

The *Journal* is published in hardcopy and in electronic form six times per year. Publication is fully electronic: submission, refereeing, revising, typesetting, checking proofs, and publishing, all is done via electronic mailing and electronic publishing.

Papers are invited in all areas of pure and applied logic, including: pure logical systems, proof theory, model theory, recursion theory, type theory, nonclassical logics, nonmonotonic logic, numerical and uncertainty reasoning, logic and AI, foundations of logic programming, logic and computation, logic and language, and logic engineering.

The *Journal* is an attempt to solve a problem in the logic (in particular, IGPL) community:

- Long delays and large backlogs in publication of papers in current journals.
- Very tight time and page number limits on submission.

Papers in the final form should be in LaTeX. The review process is quick, and is made mainly by other IGPL members.

Submissions
Submissions are made by sending a submission letter to the e-mail address: jigpl@dcs.kcl.ac.uk, giving the title and the abstract of the paper, and informing: of how to obtain the file electronically or, by sending 5 (five) hardcopies of the paper to the Editor-in-Chief.

URL: http://www.jigpal.oupjournals.org

A Spatial Similarity Measure based on Games: Theory and Practice

MARCO AIELLO, *ILLC and ISIS, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam. The Netherlands.*
Email: aiellom@ieee.org

Abstract

Model comparison games can be used not only to decide whether two specific models are equivalent or not, but also to establish a measurement of difference among a whole class of models. We show how this is possible in the case of the spatial modal logic $S4_u$ of Bennett. The approach results in a spatial similarity measure based on topological model comparison games. After establishing the theoretical framework, we move towards practice by giving an algorithm to effectively compute the similarity measure for a class of topological models widely used in computer science applications: polygons of the real plane. In the appendix, we briefly overview an implemented system based on the theoretical framework.

Keywords: model comparison games, similarity measures, modal logics of space, image retrieval based on spatial relationships

1 Introduction

There are various ways to take space qualitatively. Topology, orientation or distance have been investigated in a non-quantitative manner. The literature especially is abundant in mereotopological theories, i.e., theories of parthood P and connection C. Even though the two primitives can be axiomatized independently, the definition of part in terms of connection suffices for AI applications. Usually, some fragment of topology is axiomatized and set inclusion is used to interpret parthood, [13].

Most of the efforts in mereotopology have gone into the axiomatization of specific theories, disregarding important model theoretic questions. Issues such as model equivalence are seldom (if ever) addressed. Seeing an old friend from high-school yields an immediate comparison with the image one had from the school days. Most often, one immediately notices how many aesthetic features have changed. Recognizing a place as one already visited involves comparing the present sensory input against memories of the past sensory inputs. “Are these trees the same as I saw six hours ago, or are they arranged differently?” An image retrieval system seldom yields an exact match, more often it yields a series of ‘close’ matches. In computer vision, object occlusion cannot be disregarded. One ‘sees’ a number of features of an object and compares them with other sets of features to perform object recognition. Vision is not a matter of precise matching, it is more closely related to similarity. The core of the problem lies in the precise definition of ‘close’ match, thus the question shall be: *How similar are two spatial patterns?*

The fundamental issues in order to answer this question involve finding an agreement on spatial representation and finding an agreement on a language to describe spatial patterns. Our choice here falls on modal logics topologically interpreted. The language, called $S4_u$, is a multi-modal S4*S5 logic interpreted on topological spaces equipped with valuation functions. $S4_u$ is an extension of the simple modal logic S4 with universal and existential modal operators. Thank to the extension one can get rid of S4's intrinsic 'locality', a known technique used in modal logic, [21]. Bennett introduced $S4_u$ in the spatial setting [10] to encode decidable fragments of the region connection calculus RCC (a fundamental qualitative spatial reasoning calculus in AI [25] extending Allen's ideas [5] from temporal reasoning to spatial reasoning). The encoding also proved essential to identify maximal tractable fragments of RCC [26]. $S4_u$ has recently been used in [28] as a logic complete with respect to connected topological spaces. Finally, in the recent and important trend of combining spatial and temporal formalism, $S4_u$ plays an important role [35]. Even though the logical technique we deploy is similar to that of [10, 26], we would like to remark a shift in perspective. First, we consider $S4_u$ not as a decidable access to RCC but as a general language of (mereo-)topology: a way to 'logically' access topological and mereological notions. Second, we stress issues of model equivalence and model comparison, not only spatial representation.

Spatial representation is not only interesting in itself, but also when considering its applications. It is essential in vision, in spatial reasoning for robotics, in geographical information systems, just to name a few. Of paramount importance in applications is the comparison of spatial patterns which must be represented in the same way, in short, similarity measures are of great importance. We consider similarity measures and look at their application to image retrieval. Image retrieval is concerned with the indexing and retrieval of images from a database, according to some desired set of image features. These features can be as diverse as textual annotations, color, texture, object shape, and spatial relationships among objects. The way the features from different images are compared, in order to have a measure of similarity among images, is what really distinguishes an image retrieval architecture from another one. We refer to [17] for an overview of image retrieval and more specifically to [27] for image similarity measures. Here we concentrate on image retrieval based on spatial relationships at the qualitative level of mereotopology, that is, part-whole relations, topological relations and topological properties of individual regions (see for instance [6]). Other image retrieval systems are based on spatial relationships as the main retrieval feature. The work in [29] is founded on transformation of Voronoi diagrams and that in [23] on graph matching. An older and known approach to image retrieval by spatial relationship is in [14]. This seminal work considers the projections of regions onto two axes imposed on the picture and simple interval relations over the projections. This approach suffers from not being orientation invariant and from the inability to deal with overlapping objects. On the positive side is the compactness of the topological representation of spatial relationships (called 2D strings).

The organization of the paper reflects the transition from theory to practice we are interested in. We begin by giving the formal (theoretical) details of the modal logic $S4_u$ in Section 2. To get a feeling for its expressive power, we place it in the taxonomy of mereotopological theories of Cohn-Varzi. In Section 3, we introduce the notion of topological bisimulation for $S4_u$ and show its adequacy. The main theoretical result of the paper is provided in Section 4 where Ehrenfeucht-Fraïssé style[1] model comparison

[1] For an introduction to Ehrenfeucht-Fraïssé games see, for instance, [18].

games adequate for $S4_u$ are presented and it is shown how the games can be turned into a distance measure on the space of all topological models of $S4_u$. In Section 5, we make an ontological commitment and show the distance measure to be decidable by providing an algorithm to compute it. The proofs of the theorems throughout the paper are collected at the end in Appendix A. The techniques described in the paper have been used to implement an image retrieval prototype named `IRIS` (`I`mage `R`et`rI`eval based on `S`patial relationships) which is rapidly overviewed in Appendix B. The paper is based and extends [1, 2].

2 A general framework for Mereotopology

The proposed framework takes the beaten road of mereotopology by extending topology with a mereological theory based on the interpretation of set inclusion as parthood. Hence, a brief recall of the basic topological definitions is in order.

A *topological space* is a couple $\langle X, O\rangle$, where X is a set and $O \subseteq \mathcal{P}(X)$ such that: $\emptyset \in O$, $X \in O$, O is closed under arbitrary union, O is closed under finite intersection. An element of O is called an *open*. A subset A of X is called *closed* if $X - A$ is open. The *interior* of a set $A \subseteq X$ is the union of all open sets contained in A. The *closure* of a set $A \subseteq X$ is the intersection of all closed sets containing A.

To capture a considerable fragment of topological notions a multi-modal language $S4_u$ interpreted on topological spaces (à la Tarski [30]) is used. A *topological model* $M = \langle X, O, \nu\rangle$ is a topological space $\langle X, O\rangle$ equipped with a valuation function $\nu : P \to \mathcal{P}(X)$, where P is the set of proposition letters of the language.

The definition and interpretation of $S4_u$ follows that given in [3], which in turn is a rewriting of the one in [10]. In [3] though, emphasis is given to the topological expressivity of the language rather than the mereotopological implications. Every formula of $S4_u$ represents a region. Two modalities are available. $\Box\varphi$ to be interpreted as "interior of the region φ", and $U\varphi$ to be interpreted as "it is the case everywhere that φ." The truth definition can now be given. Consider a topological model $M = \langle X, O, \nu\rangle$ and a point $x \in X$:

$$
\begin{array}{lll}
M, x \models p & \text{iff} & x \in \nu(p) (\text{with } p \in P) \\
M, x \models \neg\varphi & \text{iff} & \text{not } M, x \models \varphi \\
M, x \models \varphi \to \psi & \text{iff} & \text{not } M, x \models \varphi \text{ or } M, x \models \psi \\
M, x \models \Box\varphi & \text{iff} & \exists o \in O :\ x \in o \wedge \\
 & & \forall y \in o :\ M, y \models \varphi \\
M, x \models \Diamond\varphi & \text{iff} & \forall o \in O :\ x \notin o \vee \\
 & & \exists y \in o :\ M, y \models \varphi \\
M, x \models U\varphi & \text{iff} & \forall y \in X :\ M, y \models \varphi \\
M, x \models E\varphi & \text{iff} & \exists y \in X,\ M, y \models \varphi
\end{array}
$$

Since $\Box$ is interpreted as interior and $\Diamond$ (defined dually as $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$, for all φ) as closure, it is not a surprise that these modalities obey the following axioms:

$$\Box A \to A \tag{T}$$
$$\Box A \to \Box\Box A \tag{4}$$
$$\Box\top \tag{N}$$
$$\Box A \wedge \Box B \leftrightarrow \Box(A \wedge B) \tag{R}$$

(4) is idempotence, while (N) and (R) are immediately identifiable in the definition of topological space. For the universal—existential modalities U and E (defined dually: $E\varphi \leftrightarrow \neg U\neg\varphi$) the axioms are those of S5:

$$U(\varphi \to \psi) \to (U\varphi \to U\psi) \tag{K}$$
$$U\varphi \to \varphi \tag{T}$$
$$U\varphi \to UU\varphi \tag{4}$$
$$\varphi \to UE\varphi \tag{B}$$

In addition, the following 'connecting' principle is part of the axioms:

$$\Diamond\varphi \to E\varphi \tag{Con}$$

The axiomatization of $\Box$ as interior is due to [30] and is generally known as S4 in modal logics. Though, in the context of Kripke semantics one gives an equivalent set of axioms to the one here provided. The axiomatization of the full $\mathrm{S4}_u$ was first introduced in [21], then by Bennett [10] with the topological interpretation for spatial reasoning.

Before defining the similarity measure based on model comparison games for $\mathrm{S4}_u$, we take a look at the mereotopological expressive power of the language. This to get acquainted with the language and to get an intuition for what $\mathrm{S4}_u$ can and what it cannot express.

2.1 Expressivity

The language $\mathrm{S4}_u$ is perfectly suited to express mereotopological concepts. The relation of parthood $\mathtt{P(A,B)}$ of a region A being inside the region B holds whenever it is the case everywhere that A implies B:

$$\mathtt{P(A,B)} := U(A \to B)$$

This captures exactly the set-inclusion relation of the models. As for connection $\mathtt{C}$, two regions A and B are connected if there exists a point where both A and B are true:

$$\mathtt{C(A,B)} := E(A \wedge B)$$

From here it is immediate to define all the usual mereotopological predicates such as proper part, tangential part, overlap, external connection, and so on. Notice that the choice made in defining $\mathtt{P}$ and $\mathtt{C}$ is arbitrary. So, why not take a more restrictive definition of parthood? Say, A is part of B whenever the closure of A is contained in the interior of B?

$$\mathtt{P(A,B)} := U(\Diamond A \to \Box B)$$

As this formula shows, $S4_u$ is expressive enough to capture also this definition of parthood. In [15], the logical space of mereotopological theories is systematized. Based on the intended interpretation of the connection predicate $\mathtt{C}$, and the consequent interpretation of $\mathtt{P}$ (and fusion operation), a type is assigned to mereotopological theories. More precisely, a *type* is a triple $\tau = \langle i, j, k \rangle$, where the first i refers to the adopted definition of $\mathtt{C_i}$, j to that of $\mathtt{P_j}$ and k to the sort of fusion. The index i, referring to the connection predicate $\mathtt{C}$, accounts for the different definition of connection at the topological level. Using $S4_u$ one can repeat here the three types of connection:

$$\mathtt{C_1(A,B)} := E(A \wedge B)$$
$$\mathtt{C_2(A,B)} := E(A \wedge \Diamond B) \vee E(\Diamond A \wedge B)$$
$$\mathtt{C_3(A,B)} := E(\Diamond A \wedge \Diamond B)$$

Looking at previous mereotopological literature, one remarks that RCC uses a $\mathtt{C_3}$ definition, while the system proposed in [6] uses a $\mathtt{C_1}$. Similarly to connectedness, one can distinguish the various types of parthood, again in terms of $S4_u$:

$$\mathtt{P_1(A,B)} := U(A \rightarrow B)$$
$$\mathtt{P_2(A,B)} := U(A \rightarrow \Diamond B)$$
$$\mathtt{P_3(A,B)} := U(\Diamond A \rightarrow \Diamond B)$$

In [15], the definitions of the $\mathtt{C_i}$ are given directly in terms of topology, and the definitions of $\mathtt{P_j}$ in terms of a first order language with the addition of a predicate $\mathtt{C_i}$. Finally, a general fusion $\phi_{\mathtt{k}}$ is defined in terms of a first order language with a $\mathtt{C_i}$ predicate. Fusion operations are like algebraic operations on regions, such as adding two regions (product), or subtracting two regions. One cannot repeat the general definition given in [15] at the $S4_u$ level. Though, one can show that various instances of fusion operations are expressible in $S4_u$. For example, the product $A \times_k B$:

$$\mathtt{A} \times_1 \mathtt{B} := A \wedge B$$
$$\mathtt{A} \times_2 \mathtt{B} := (\Diamond A \wedge B) \vee (A \wedge \Diamond B)$$
$$\mathtt{A} \times_3 \mathtt{B} := (\Diamond A \wedge \Diamond B)$$

The above discussion has shown that $S4_u$ is a general language for mereotopology. All the different types $\tau = \langle i, j, k \rangle$ of mereotopological theories are expressible within $S4_u$.

3 When are two spatial patterns the same?

One is now ready to address questions such as: *When are two spatial patterns the same?* or *When is a pattern a sub-pattern of another one?* More formally, one wants to define a notion of equivalence adequate for $S4_u$ and the topological models. In first-order logic the notion of 'partial isomorphism' is the building block of model equivalence. Since $S4_u$ is multi-modal language, one resorts to bisimulation, which is the modal analogue of partial isomorphism. Bisimulations compare models in a structured sense, 'just enough' to ensure the truth of the same modal formulas [32, 22].

Definition 3.1 (Topological bisimulation) Given two topological models $\langle X, O, \nu\rangle$, $\langle X', O', \nu'\rangle$, a *total topological bisimulation* is a non-empty relation $\leftrightarrows \subseteq X \times X'$ defined for all $x \in X$ and for all $x' \in X'$ such that if $x \leftrightarrows x'$:

(base): $x \in \nu(p)$ iff $x' \in \nu'(p)$ (for any proposition p)

(forth condition): if $x \in o \in O$ then
$\exists o' \in O' : x' \in o'$ and $\forall y' \in o' : \exists y \in o : y \leftrightarrows y'$

(back condition): if $x' \in o' \in O'$ then
$\exists o \in O : x \in o$ and $\forall y \in o : \exists y' \in o' : y \leftrightarrows y'$

If only conditions (i) and (ii) hold, the second model *simulates* the first one.

The notion of bisimulation is used to answer questions of 'sameness' of models, while simulation will serve the purpose of identifying sub-patterns. Though, one must show that the above definition is adequate with respect to the mereotopological framework provided in this paper.

Theorem 3.2 Let $M = \langle X, O, \nu\rangle$, $M' = \langle X', O', \nu'\rangle$ be two models, $x \in X$, and $x' \in X'$ bisimilar points. Then, for any modal formula φ in $\mathrm{S4}_u$, $M, x \models \varphi$ iff $M', x' \models \varphi$.

Theorem 3.3 Let $M = \langle X, O, \nu\rangle$, $M' = \langle X', O', \nu'\rangle$ be two models with finite O, O', $x \in X$, and $x' \in X'$ such that for every φ in $\mathrm{S4}_u$, $M, x \models \varphi$ iff $M', x' \models \varphi$. Then there exists a total bisimulation between M and M' connecting x and x'.

In words, extended modal formulas are invariant under total bisimulations, while finite modally equivalent models are totally bisimilar. One may notice, that in Theorem 3.3 a finiteness restriction is posed on the open sets. This will not surprise the modal logician, since the same kind of restriction holds for Kripke semantics and does not affect the proposed use for bisimulations in the mereotopological framework.

4 How different are two spatial patterns?

If topological bisimulation is satisfactory from the formal point of view, one needs more to address qualitative spatial reasoning problems and computer vision issues. If two models are not bisimilar, or one does not simulate the other, one must be able to quantify the difference between the two models. Furthermore, this difference should behave in a coherent manner across the class of all models. Informally, one needs to answer questions like: *How different are two spatial patterns?*

To this end, we recall the game theoretic definition of topo-games [3], and then prove the main theoretical result of this paper, namely the fact that topo-games induce a distance on the space of all topological models for $\mathrm{S4}_u$. First, we give the definition and the theorem that ties together the topo-games, $\mathrm{S4}_u$ and topological models.

Definition 4.1 (topo-game) Consider two topological models $\langle X, O, \nu\rangle$, $\langle X', O', \nu'\rangle$ and a natural number n. A *topo-game* of length n, notation $TG(X, X', n)$, consists of n rounds between two players, Spoiler and Duplicator, who move alternatively.

Spoiler is granted the first move and always chooses which type of round to engage. The two sorts of rounds are as follows:

$$\textbf{global}\begin{cases} (i) & \text{Spoiler chooses a model } X_s \text{ and picks a point} \\ & \bar{x}_s \text{ anywhere in } X_s \\ (ii) & \text{Duplicator chooses a point } \bar{x}_d \text{ anywhere in t he} \\ & \text{other model } X_d \end{cases}$$

$$\textbf{local}\begin{cases} (i) & \text{Spoiler chooses a model } X_s \text{ and an open } o_s \\ & \text{containing the current point } x_s \text{ of that model} \\ (ii) & \text{Duplicator chooses an open } o_d \text{ in the other} \\ & \text{model } X_d \text{ containing its current point } x_d \\ (iii) & \text{Spoiler picks a point } \bar{x}_d \text{ in Duplicator's open} \\ & o_d \text{ in the } X_d \text{ model} \\ (iv) & \text{Duplicator replies by picking a point } \bar{x}_s \text{ in} \\ & \text{Spoiler's open } o_s \text{ in } X_s \end{cases}$$

The points $\bar{x}_s$ and $\bar{x}_d$ become the new current points. A game always starts by a global round. By this succession of actions, two sequences are built: $\{x_1, x_2, \ldots x_n\}$ and $\{x'_1, x'_2, \ldots x'_n\}$. After n rounds, if x_i and x'_i (with $i \in [1, n]$) satisfy the same propositional atoms, Duplicator *wins*, otherwise, Spoiler wins. A *winning strategy (w.s.)* for Duplicator is a function from any sequence of moves by Spoiler to appropriate responses which always ends in a win for him. Spoiler's winning strategy is defined dually.

The *multi-modal rank* of a $\mathrm{S4}_u$ formula is the maximum number of nested modal operators appearing in it (i.e. $\Box$, $\Diamond$, U and E modalities). The following adequacy of the games with respect to the mereotopological language holds.

Theorem 4.2 (Adequacy) Duplicator has a winning strategy for n rounds in $TG(X, X', n)$ iff X and X' satisfy the same formulas of multi-modal rank at most n.

Various examples of plays and a discussion of winning strategies can be found in [3].[2]

The interesting result is that of having a game theoretic tool to compare topological models. Given any two models, they can be played upon. If Spoiler has a winning strategy in a certain number of rounds, then the two models are different up to a certain degree. The degree is exactly the minimal number of rounds needed by Spoiler to win. On the other hand, one knows (see [3]) that if Spoiler has no w.s. in any number of rounds, and therefore Duplicator has in all games, including the infinite round game, then the two models are bisimilar.

A way of comparing any two given models is not of great use by itself. It is essential instead to have some kind of measure. It turns out that topo-games can be used to define a distance measure.

---

[2] For example, one may find interesting that a normal form is available for the language (one for which every formula has one universal modal operator ranging over boolean combinations of local modal operators). The normal form is tied to the winning strategies of either player.

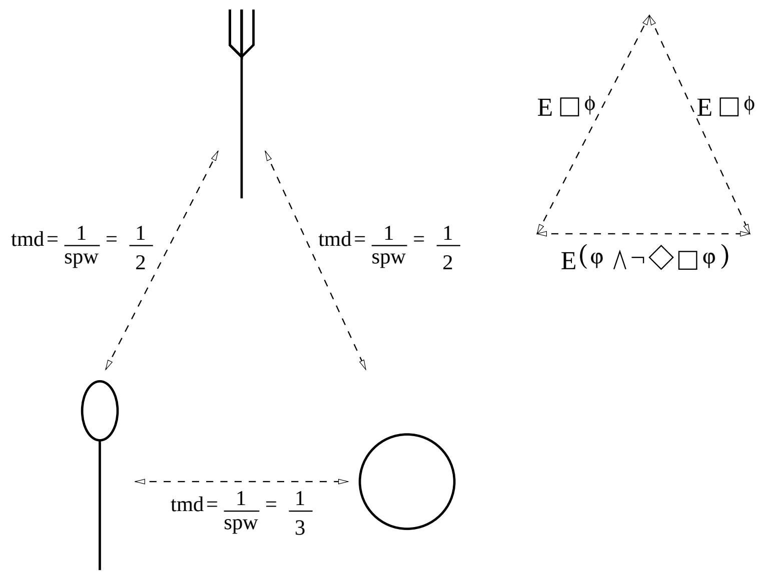


FIG. 1: On the left, three models and their relative distance. On the right, the distinguishing formulas.

Definition 4.3 (isosceles topo-distance) Consider the space of all topological models T. *Spoiler's shortest possible win* is the function $spw : T \times T \to I\!N \cup \{\infty\}$, defined as:

$$spw(X_1, X_2) = \begin{cases} n & \text{if Spoiler has a winning strategy in } TG(X_1, X_2, n), \\ & \text{but not in } TG(X_1, X_2, n-1) \\ & \\ \infty & \text{if Spoiler does not have a winning strategy in} \\ & TG(X_1, X_2, \infty) \end{cases}$$

The *isosceles topo-model distance (topo-distance,* for short*)* between X_1 and X_2 is the function $tmd : T \times T \to [0, 1]$ defined as:

$$tmd(X_1, X_2) = \frac{1}{spw(X_1, X_2)}$$

The distance was named 'isosceles' since it satisfies the triangular property in a peculiar manner. Given three models, two of the distances among them (two sides of the triangle) are always the same and the remaining distance (the other side of the triangle) is smaller or equal. On the left of Figure 1, three models are displayed: a spoon, a fork and a plate. Think these cutlery objects as subsets of a dense space, such as the real plane, which evaluate to ϕ, while the background of the items evaluates to $\neg\phi$. The isosceles topo-distance is displayed on the left next to the arrow connecting two models. For instance, the distance between the fork and the spoon is $\frac{1}{2}$ since the minimum number of rounds that Spoiler needs to win the game is 2.

To see this, consider the formula $E\Box\phi$, which is true on the spoon (there exists an interior point of the region ϕ associated with the spoon) but not on the fork (which has no interior points). On the right of the figure, the formulas used by spoiler to win the three games between the fork, the spoon and the plate are shown. Next the proof that *tmd* is really a distance, in particular the triangular property, exemplified in Figure 1, is always satisfied by any three topological models.

Theorem 4.4 (isosceles topo-model distance) *tmd* is a distance measure on the space of all topological models.

The nature of the isosceles topo-distance triggers a question. Why, given three spatial models, the distance between two couples of them is always the same?

First an example, consider a spoon, a chop-stick and a sculpture by Henry Moore. It is immediate to distinguish the Moore's sculpture from the spoon and from the chop-stick. The distance between them is high and the same. On the other hand, the spoon and the chop-stick look much more similar, thus, their distance is much smaller. Mereotopologically, it may even be impossible to distinguish them, i.e., the distance may be null.

In fact one is dealing with models of a qualitative spatial reasoning language of mereotopology. Given three models, via the isosceles topo-distance, one can easily distinguish the very different patterns. In some sense they are far apart as if they were belonging to different equivalence classes. Then, to distinguish the remaining two can only be harder, or equivalently, the distance can only be smaller.

5 Computing similarities

The fundamental step to move from theory to practice has been taken when shifting from model comparison games to a distance. To complete the journey towards practice one needs to identify ways of effectively compute the distance in cases actually occurring in real life domains. We do not have an answer to the general question of whether the topo-distance is computable for any two topological models or not. Though, by restricting to a specific class of topological models widely used in real life applications, we can show the topo-distance to be computable when one makes an ontological commitment. The commitment consists of considering topological spaces made of polygons. This is common practice in various application domains such as geographical information systems (GIS), in many branches of image retrieval and of computer vision, in robot planning, just to mention the most common.

Consider the real plane $\mathbb{R}^2$, any line in $\mathbb{R}^2$ cuts it into two half-planes. We call a half-plane *closed* if it includes the cutting line, *open* otherwise.

Definition 5.1 (region) A *polygon* is the intersection of finitely many open or closed half-planes. An *atomic region* of $\mathbb{R}^2$ is the union of finitely many polygons.

An atomic region is denoted by one propositional letter. More in general, any set of atomic regions, simply called *region*, is denoted by a $S4_u$ formula. The polygons of the plane equipped with a valuation function, denoted by $M_{\mathbb{R}^2}$, are in full rights a topological model as defined in Section 2, a basic topological fact. A similar definition of region can be found in [24]. In that article Pratt and Lemon also provide a collection of fundamental results regarding the plane, polygonal ontology just defined (actually one in which the regions are open regular).

From the model theoretic point of view, the advantage of working with $M_{\mathbb{R}^2}$ is that we can prove a logical finiteness result and thus give a terminating algorithm to compute the topo-distance. The preliminary step is thus that of proving a finiteness lemma for $S4_u$ over $M_{\mathbb{R}^2}$ models.[3]

Lemma 5.2 (finiteness) There are only finitely many modally definable subsets of a finite set of regions $\{r_i | r_i \text{ is an atomic region}\}$.

Here is a proof sketch.[4] We work by enumerating cases, i.e., considering boolean

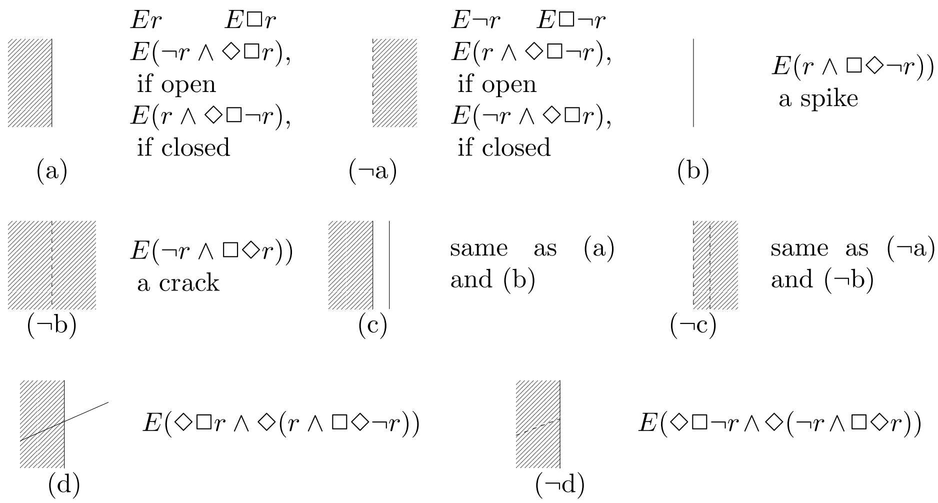


Fig. 2. Basic formulas defined by one region.

combinations of planes, adding to an 'empty' space one half-plane at the time, first to build one region r, and then to build a finite set of regions. The goal is to show that only finitely many possibilities exist. We begin by placing a half plane denoted by r on an empty bidimensional space, Figure 2.a. Let us follow what happens to points in the space from left to right. On the left, points satisfy the formula $E(r \wedge \Box r)$ and its subformulas Er and $E\Box r$. This is true until we reach the frontier point of the half-plane. Either $E(\neg r \wedge \Diamond\Box r)$ or $E(r \wedge \Diamond\Box\neg r)$ are true depending on whether the half-plane is open or closed, respectively. Once the frontier has been passed to the right, the points satisfy $E(\neg r \wedge \Box\neg r)$ and its subformulas $E\neg r$ and $E\Box\neg r$, better seen in Figure 2.¬a. In fact, if we consider negation in the formulas the role of r and $\neg r$ switch. Consider now a second plane in the picture:

- **Intersection:** the intersection may be empty (no new formula), may be a polygon with two sides and vertices (no new formula, the same situation as with one polygon), or it may be a line, the case of two closed polygons that share the side (in this last case depicted in Figure 2.b—*spike*—we have a new formula, namely, $E(r \wedge \Box\Diamond\neg r))$.

[3] Of course, in general this is not true. There are infinitely many non equivalent $S4_u$ formulas and one can identify appropriate Kripke models to show this. See, e.g., [11].

[4] The finiteness lemma is the extension to two dimensions of the theorem for serial sets of [4]. In two-dimensions one has 8 non-equivalent formulas rather than 6, as in the one dimensional case proved in [4].

- **Union:** the union may be a polygon with either one or two sides (no new formula), two separated polygons (no new formula), or two open polygons sharing the open side (this last case depicted in Figure 2.¬b—*crack*—is like the spike, one inverts the roles r and $\neg r$ in the formula: $E(\neg r \wedge \Box\Diamond r)$).

Finally, consider combining cases (a) and (b). By union, we get Figure 2.a, 2.c, 2.d. The only situation bringing new formulas is the latter. In particular, the point where the line intersects the plane satisfies the formula: $E(\Diamond\Box r \wedge \Diamond(r \wedge \Box\Diamond\neg r))$. By intersection, we get a segment, or the empty space, thus, no new formula.

The four basic configurations just identified yield no new configuration from the S4$_u$ point of view. To see this, consider the boolean combinations of the above configurations. We begin by negation (complement):

¬	a	b	c	d
	¬a	¬b	¬a, ¬b	¬d

Union straightforwardly follows (where a stands for both a and ¬a, as both configurations always appear together):

⋃	a	b	c	d
a	a, ¬b, ¬d	a, c, d	a, ¬b, c, d, ¬d	a, ¬b, d, ¬d
b	a, c, d	b	c, d	d
c	a, ¬b, c, d, ¬d	c, d	a, ¬b, c, d, ¬d	a, ¬b, c, d, ¬d
d	a, ¬b, d, ¬d	d	a, ¬b, c, d, ¬d	a, ¬b, d, ¬d

The table for intersection follows, with the proviso that the combination of the two regions can always be empty (not reported in the table) and again a and ¬a are represented simply by a:

⋂	a	b	c	d
a	a, b, c, d	b	a, b, c	a, b, d
b	b	b	b	b
c	a, b, c	b	a, b, c, d	a, b, c, d
d	a, b, d	b	a, b, c, d	a, b, c, d

We call *topo-vector* associated with the region r, notation $\vec{r}$, an ordered sequence of ten boolean values. The values represent whether the region r satisfies or not the ten formulas

$$\{Er, E\neg r, E\Box r, E\Box\neg r, E(\neg r \wedge \Diamond\Box r), E(r \wedge \Diamond\Box\neg r), E(r \wedge \Box\Diamond\neg r)), E(\neg r \wedge \Box\Diamond r)), E(\Diamond\Box r \wedge \Diamond(r \wedge \Box\Diamond\neg r)), E(\Diamond\Box\neg r \wedge \Diamond(\neg r \wedge \Box\Diamond r))\}.$$

The ten formulas are those identified in Figure 2 which we have shown to be the only one definable by boolean combinations of planes denoting the same one region r. For example, the topo-vector associated with a plate—a closed square r in the plane—is {true, true, true, true, false, true, false, false, false, false}.

Adding half-planes with different denotations $r_2, r_3, \ldots$ increases the number of defined formulas. The definition of topo-vector is extended to an entire $M_{\mathbb{R}^2}$ model:

$$\{E\bigwedge_i[\neg]r_i,\ E\bigwedge_i\Box[\neg]r_i,\ E(\bigwedge_i[\neg]^+r_i\wedge\bigwedge_i\Diamond\Box[\neg]^*r_i),\ E(\bigwedge_i[\neg]^+r_i\wedge\bigwedge_i\Box\Diamond[\neg]^*r_i),$$
$$E(\bigwedge_i\Diamond\Box[\neg]^+r_i\wedge\Diamond(\bigwedge_i[\neg]^+r_i\wedge\bigwedge_i\Box\Diamond[\neg]^*r_i))\},$$

where $[\,]$ denotes an option and if the option $[\,]^+$ is used then the option $[\,]^*$ is not and vice-versa. The topo-vector is built such that the modal rank of the formulas is not decreasing going from the positions with lower index to those with higher. The size of such a vector is $5\cdot 2^i$ where i is the number of denoted regions of the model. The fact that the size of the topo-vector grows exponentially with the number of regions might seem a serious drawback. Though, as we shall show in a moment, the topo-vector stores all the information relevant for $S4_u$ about the model. Furthermore, the size of a topo-vector is most often considerably smaller than that of a topological model. In fact, a topo-vector is of exponential size in the number of regions, while a topological model is of exponential size in the number of points of the space because of the set of opens. As a final argument, one should add that in practical situations the number of regions is always much smaller than the number of points of the space.

We are now in a position to devise an algorithm to compute the topo-distance between two topological models. The algorithm works by first computing the associated topo-vectors and then comparing them. By the comparison it is possible to establish which formulas differentiate the two models and therefore the distance between the two models. Here is the general algorithm (in pseudo-code) to compute the topo-distance between two topological models M_1 and M_2:

```
topo-distance(M1, M2)
  v1 = topo-vector (M1)
  v2 = topo-vector (M2)
  align v1 and v2
  loop on v1 v2 with index i
    if v1(i) ≠ v2(i)
      return 1 / modal rank(v1(i))
  return 0
```

The idea is of retrieving the topo-vectors associated with the two input models and then looping over their elements. The inequality check can also be thought of as a **xor**, since the elements of the array are booleans. If the condition is never satisfied, the two topo-vectors are identical, the two-models are topo-bisimilar and thus the topo-distance is null. The `align` command makes the topo-vectors of the same length and aligns the formulas of the two, i.e., such that to the same index in the vector

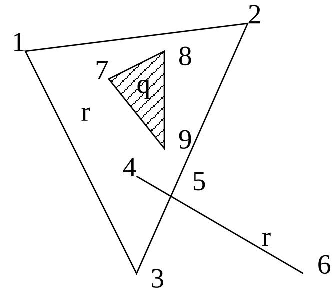


FIG. 3. Computing the topo-vector on a simple model.

corresponds the same formula. If a topo-vector contains a formula that the other one does not, the entry is added to the vector missing it with a false value. To complete the description of the algorithm, we provide the function to compute the topo-vector associated with an $M_{\mathbb{R}^2}$ model:

`topo-vector(`M`)`
    $\vec{v}$ `= initialized to all false values`
    loop `on regions` r `of` M `with index` i
        loop `on atomic regions` a `of` $r(i)$ `with index` j
            loop `on vertices` v `of` $a(j)$ `with index` k
                `update` $\vec{v}$ `with the point` $v(k)$
                if $v(k)$ `is not free`
                    loop `on intersections` x `of` $a(j)$ `with all regions of` M `with index` l
                        `update` $\vec{v}$ `with the point` $x(l)$
    return $\vec{v}$

If a point $v(k)$ of an atomic region $a(j)$ is contained in any polygon different from $a(j)$ and it is not contained in any other region, then the condition $v(k)$ `is not free` is satisfied. Standard computational geometry algorithms exist for this task, [16]. When the "`update` $\vec{v}$ `with the point` p" function is called, one checks in which case p is (as shown after Lemma 5.2), then one considers the position of the corresponding topo-vector and puts in a true value. An obvious optimization to the algorithm is to avoid checking points for which all the associated formulas are already true. Consider the simple model of Figure 3 composed of two closed regions r and q. Since there are two regions, the topo-vector will be of size $5 \cdot 2^2 = 20$ elements: $\{E(r \wedge q), E(r \wedge \neg q), \ldots E(\Diamond\Box\neg r \wedge \Diamond\Box\neg p \wedge \Diamond(\neg r \wedge \neg q \wedge \Box\Diamond r \wedge \Box\Diamond q))\}$. After initialization, the region r is considered and one starts looping on the vertices of its polygons, first the point 1. The point is `free`, it is the vertex of a full polygon (not a segment) and therefore the topo-vector is updated directly in the positions corresponing to $Er \wedge \neg q$, $E\Box r \wedge \Box\neg q$, $Er \wedge \neg q \wedge \Box r \wedge \Box\neg q$, $Er \wedge \neg q \wedge \Diamond\Box r \wedge \Diamond\Box q$. The points 2 and 3 would update the values for the same formula and are not considered. The point 4 falls inside the first polygon of r, the topo-vector does not need update. Intersections are then computed and the point 5 is found. The point needs to update the vector for the formula $E\Diamond\Box r \wedge \Diamond\Box\neg q \wedge \Diamond(r \wedge \neg q \wedge \Box\Diamond\neg r \wedge \Box\Diamond\neg q)$. Finally, the point 6 is considered and the point needs to update the formula $E(r \wedge \neg q \wedge \Diamond\Box\neg r \wedge \Diamond\Box\neg q)$. The algorithm proceeds by considering the second region, q and its vertices 7, 8, and 9. The three vertices all fall inside the region r and provide for the satisfaction of the formulas $Er \wedge q$, $E\Box r \wedge \Box q$, ...

Lemma 5.3 (termination) The topo-distance algorithm terminates.

The property is easily shown by noticing that a segment (a side of a polygon) can have at most one intersection with any other segment, that the number of polygons forming a region of $M_{\mathbb{R}^2}$ is finite, and that the number of regions of $M_{\mathbb{R}^2}$ is finite. Putting this result together with Lemma 5.2 one gets the hoped decidability result for polygonal topological models.

Theorem 5.4 (decidability of the topo-distance) In the case of polygonal topological models $M_{\mathbb{R}^2}$ over the real plane, the problem of computing the topo-distance among any two models is decidable.

Given the definition of topo-distance, the fact that two models have a null topo-distance implies that in the topo-game Duplicator has a winning strategy in the infinite round game. In the case of $M_{\mathbb{R}^2}$, Theorem 5.4 implies that the two models are topo-bisimilar. Note that, in general, this is not the case: Duplicator may have a winning strategy in an infinite model comparison game adequate for some modal language and the models need not be bisimilar [9].

Corollary 5.5 (decidability of topo-bisimulations) In the case of polygonal topological models over the real plane, the problem of identifying whether two models are topo-bisimilar or not is decidable.

6 Conclusions

We have followed the line from theory to practice in a context of spatial reasoning. First, we have considered a general mereotoplogical framework, placing the language $S4_u$ where it belongs: $S4_u$ is a general mereotopological language not committed to any specific definition of connection, but rather with high topological discriminating power. We addressed issues of model equivalence and especially of model comparison, thus, looking at mereotopology from a new angle. Defining a distance that encodes the mereotopological difference between spatial models has important theoretical and application implications, as we have shown. Our journey has ended by illustrating the actual decidability of the devised similarity measure for a practically interesting class of models.

The theoretical framework proposed is much more general than what we have shown here. We were interested in a mereotopological framework and have therefore used the language $S4_u$ interpreted on topological models, but an isosceles distance can be used for any modal language equipped with negation and for which one has adequate notions of model comparison games and bisimulation. Even the restriction to modal logic is not necessary, one can think of first-order logic, of the usual Ehrenfeucht-Fraïssé games, of elementary equivalence in place of bisimulation, and an isosceles distance is then definable. The decidabilty result for the distance is the only thing that does not necessarily extend, rather one has to consider the class of models and the logic case by case. Of particular importance is then how the adequate topological games are defined.

We would like to stress the fact that the use of model comparison games presented in this paper is novel. Model comparison games have been used only to compare two given models, but the issue of setting a distance among a whole class of models has not been previously addressed. The technique employed in Theorem 4.4 for the language $S4_u$ is, as we have just mentioned, much more general. A question interesting *per se*, but out of the scope of the present paper, is: which is the class of games (over which languages) for which a notion of isosceles distance holds? We belive the class of such languages and model comparison games to be quite vast.

Having implemented a system based on the above framewok is also an important step in the presented research. Experimentation is essential to asses applicability, but some preliminar considerations are possible. We have noticed that the prototype is very sensible to the labeling of segmented areas of images, i.e., to the assignment of propositional letters to regions. We have also noticed that the mereotopological expressive power appears to enhance the quality of retrieval and indexing over pure textual searches, but the expressive power of $S4_u$ is still too limited. Notions of qualitative orientation, shape or geometry appear to be important, especially when the user expresses his desires in the form of an image query or of a sketch.

The generality of the framework described in the paper allows for optimism about future developments. Once one has identified an appropriate language of, say, qualitative shape with adequate model comparison games, a newer version of `IRIS` can be built. We will be researching in two directions. On the one hand, qualitative notions of shape are expressible via mathematical morpholoy, which in turn is closely related to modal logics [12]. On the other hand, axiomatizations of the notions of betweenness (also originated by Tarski in [31]) may provide for qualitative notions of geometry. Again, one can stay on the ground of modal logics and one can look at languages for incidence geometries. In the approach, one distinguishes the sorts of elements that populate space and considers the incidence relation between elements of the different sorts (see [8, 7, 33]).

All in all, there is much more to model-comparison games than simply laying down two peculiar models and start playing on them. We have looked at spatial reasoning and at image similarity, but many more roads are viable.

Aknowledgements

The author is thankful to Johan van Benthem for fruitful discussions on the topic, to Kees Doets for feedback on a previous version of the proof of Theorem 4.4, and to Michael Zakharyaschev for helpful discussion. This work was supported in part by CNR grants 203.15.10 and 203.7.27.

References

[1] M. Aiello. Topo-distance: Measuring the Difference between Spatial Patterns. In M. Ojeda-Aciego, I. P. de Guzmán, G. Brewka, and L. Moniz Pereira, editors, *Logics in Artificial Intelli-*

gence (JELIA 2000), LNAI, 1919, pages 73–86. Springer, 2000.

[2] M. Aiello. Computing spatial similarity by games. In F. Esposito, editor, *AI*IA 2001: Advances in Artificial Intelligence*, LNAI, 2175, pages 99–110. Springer, 2001.

[3] M. Aiello and J. van Benthem. Logical Patterns in Space. In D. Barker-Plummer, D. Beaver, J. van Benthem, and P. Scotto di Luzio, editors, *Logic Unleashed: Language, Diagrams and Computation*, Stanford, 2002. CSLI. To appear.

[4] M. Aiello, J. van Benthem, and G. Bezhanishvili. Reasoning about Space: the Modal Way. Manuscirpt, 2001.

[5] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.

[6] N. Asher and L. Vieu. Toward a Geometry of Common Sense: a semantics and a complete axiomatization of mereotopology. In *IJCAI95*, pages 846–852. International Joint Conference on Artificial Itelligence, 1995.

[7] Ph. Balbiani. The modal multilogic of geometry. *Journal of Applied Non-Classical Logics*, 8:259–281, 1998.

[8] Ph. Balbiani, L. Fariñas del Cerro, T. Tinchev, and D. Vakarelov. Modal logics for incidence geometries. *Journal of Logic and Computation*, 7:59–78, 1997.

[9] J. Barwise and L. Moss. *Vicious Circles*. CSLI, 1996.

[10] B. Bennett. Modal Logics for Qualitative Spatial Reasoning. *Bulletin of the IGPL*, 3:1 – 22, 1995.

[11] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[12] I. Bloch. Using Mathematical Morphology Operators as Modal Operators for Spatial Reasoning. In *ECAI 2000, Workshop on Spatio-Temporal Reasoning*, pages 73–79, 2000.

[13] R. Casati and A. Varzi. *Parts and Places*. MIT Press, 1999.

[14] S.K. Chang and S.H. Liu. Picture indexing and abstraxtion techniques for pictorial databases. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(7):475–484, 1984.

[15] A. G. Cohn and A. Varzi. Connection Relations in Mereotopology. In H. Prade, editor, *Proc. 13th European Conf. on AI (ECAI98)*, pages 150–154. John Wiley, 1998.

[16] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2000. Second edition.

[17] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, 1999.

[18] K. Doets. *Basic Model Theory*. CSLI Publications, Stanford, 1996.

[19] M. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, 8(4):403–424, 1997.

[20] W. Frakes and R. Baeza-Yates, editors. *Information Retrieval, Data Structures and Algorithms*. Prentice Hall, 1992.

[21] V. Goranko and S. Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.

[22] D. Park. Concurrency and Automata on Infinite Sequences. In *Proceedings of the 5th GI Conference*, pages 167–183, Berlin, 1981. Springer Verlag.

[23] E. Petrakis, C. Faloutsos, and L. King-Ip. ImageMap: An Image Indexing Method Based on Spatial Similarity. *IEEE Transactions on Knowledge and Data Engineering*, 2001. To appear.

[24] I. Pratt and O. Lemon. Ontologies for Plane, Polygonal Mereotopology. *Notre Dame Journal of Formal Logic*, 38(2):225–245, 1997.

[25] D. Randell, Z. Cui, and A. G. Cohn. A Spatial Logic Based on Regions and Connection. In *Proc. of Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 165–176. San Mateo, 1992.

[26] J. Renz and B. Nebel. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *Artificial Intelligence*, 108(1-2):69–123, 1999.

[27] S. Santini and J. Ramesh. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[28] V. Shehtman. “Everywhere” and “Here”. *Journal of Applied Non-Classical Logics*, 9(2-3):369–379, 1999.

[29] H. Tagare, F. Vos, C. Jaffe, and J. Duncan. Arrangement: A spatial relation between part for evaluating similarity of tomographic section. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):880–893, 1995.

[30] A. Tarski. Der Aussagenkalkül und die Topologie. *Fund. Math.*, 31:103–134, 1938.

[31] A. Tarski. What is Elementary Geometry? In L. Henkin and P. Suppes and A. Tarski, editor, *The Axiomatic Method, with Special Reference to Geometry ad Physics*, pages 16–29. North-Holland, 1959.

[32] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, University of Amsterdam, 1976.

[33] Y. Venema. Points, Lines and Diamonds: a Two-Sorted Modal Logic for Projective Planes. *Journal of Logic and Computation*, 9(5):601–621, 1999.

[34] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, 2nd edition, 1999.

[35] F. Wolter and M. Zakharayaschev. Spatio-temporal representation and reasoning based on RCC-8. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, pages 3–14, 2000.

A Proofs of Theorems

Proof of Theorem 3.2 on page 6

Induction on φ. The case of a proposition letter p is the first condition on $\leftrightharpoons$. As for conjunction, $M, x \models \varphi \wedge \psi$ is equivalent by the truth definition to $M, x \models \varphi$ and $M, x \models \psi$, which by the induction hypothesis is equivalent to $M', x' \models \varphi$ and $M', x' \models \psi$, which by the truth definition amounts to $M', x' \models \varphi \wedge \psi$. The other boolean cases are similar. For the modal case, we do one direction. First the 'local' modal operators $\Box$ and $\Diamond$: If $M, x \models \Box\varphi$, then by the truth definition we have that $\exists o \in O : x \in o \wedge \forall y \in o : M, y \models \varphi$. By the forth condition, corresponding to o, there must exist an $o' \in O'$ such that $\forall y' \in o'\ \exists y \in o\ y \leftrightharpoons y'$. By the induction hypothesis applied to y and y' with respect to φ, then $\forall y' \in o' : M', y' \models \varphi$. By the truth definition of the modal operator we have $M', x' \models \Box\varphi$. Using the back condition one proves the other direction likewise. Now the 'global' modal operators U and E: If $M, x \models E\varphi$, then by the truth definition we have that $\exists y\ M, y \models \varphi$. The point x must be in the open set X (the whole space). By the forth condition, we know that there must be a corresponding open set o' in X'. Though, we need a little bit more, therefore we show next that X' itself is one of such o'.

Fact A.1 If the topological models $\langle X, O, \nu\rangle$ and $\langle X', O', \nu'\rangle$ are totally topologically bisimilar, a special instance of the forth condition holds (similarily for the back condition):

$$\text{if } x \in X \in O \text{ then } \exists X' \in O' : x' \in X' \text{ and } \forall y' \in X' : \exists y \in X : y \leftrightharpoons y'$$

In general, if we instantiate the forth condition on the whole space X, we do not know which open set o' will correspond on $\langle X', O', \nu'\rangle$:

$$\text{if } x \in X \in O \text{ then } \exists o' \in O' : x' \in o' \text{ and } \forall y' \in o' : \exists y \in X : y \leftrightharpoons y'$$

Obviously, $o' \subseteq X'$. Ab absurdum, $o' \neq X'$ for all o' as just defined by the forth condition. Thus, there must exist an element p' of X' that belongs to none of the sets o'. Two cases are now possible. Either there exists a $p \in X$ such that $p \leftrightharpoons p'$ or it does not exist. In both cases one reaches a contradiction. If it exists, then X' is an o' open set. If it does not, then the two models are not totally topologically bisimilar.

We can now proceed in the original proof. By the forth condition and the above fact, corresponding to X there is the open whole space X' such that $\forall z \in X'\ \exists z \in X\ z \leftrightharpoons z'$. By the induction hypothesis applied to z and z' with respect to φ, $\forall z' \in X' : M', z' \models \varphi$. One of such z is the y of the truth definition for $E\varphi$, therefore $M', y' \models \varphi$. By the truth definition, we have $M', x' \models E\varphi$. Using the back condition one proves the other direction likewise.

Proof of Theorem 3.3 on page 6

To get a bisimulation between the two finite models, we stipulate that $u \leftrightharpoons u'$ if and only if u and u' satisfy the same modal formulas. The atomic preservation condition for a bisimulation holds since the modal φ include all proposition letters. We now prove the forth condition. Suppose that $u \leftrightharpoons u'$ where $u \in o$. We must find an open o' such that $u' \in o'$ and $\forall y' \in o' \exists y \in o : y \leftrightharpoons y'$. Now, suppose there is no such o'. Then for every o' containing x' $\exists y' \in o' : \forall y \in o : \exists \varphi_y : y \not\models \varphi_y$ and $y' \models \varphi_y$. In words, every open o' contains a point y' with no modally equivalent point in o. Taking the finite conjunction of all formulas φ_y, we get a formula $\Phi_{o'}$ such that $y' \models \Phi_{o'}$ and $\neg\Phi_{o'}$ is true everywhere in o. Slightly abusing notation, we write $o \models \neg\Phi_{o'}$. This line of reasoning holds for any open o' containing x' as chosen. Therefore, there exists a collection of formulas $\neg\Phi_{o'}$ for which $o \models \bigwedge_{o'} \neg\Phi_{o'}$. Since $x \in o$, by the truth definition we have $x \models \Box \bigwedge_{o'} \neg\Phi_{o'}$. By the fact that x and x' satisfy the same modal formulas, it follows that $x' \models \Box \bigwedge_{o'} \neg\Phi_{o'}$. But then, there exists an open o^* (with $x' \in o^*$) such that $o^* \models \bigwedge_{o'} \neg\Phi_{o'}$. Since o^* is an open containing x', is one of the o', i.e. $o^* \models \neg\Phi_{o^*}$. But we had supposed that for all opens o' there was a point $y' \models \Phi_{o'}$, so in particular the y' of o^* satisfies Φ_{o^*}. We have thus reached a contradiction: which shows that some appropriate open o' must exist. The back clause is proved analogously.

Proof of Theorem 4.2 on page 7

The left to right direction is proven by induction on the length n of the game $TG(X, X', n)$. If $n = 0$ and Duplicator has a winning strategy, this means that X and X' satisfy the same propositional letters, hence the same boolean combinations of propositional letters, i.e., the same modal formulas of modal rank 0. Now for the inductive step. Suppose that Duplicator has a winning strategy σ in $TG(X, X', n)$. We want to show that $X, x \models \varphi$ iff $X', x' \models \varphi$ when the modal rank of φ is n. By simple syntactic inspection, φ must be a boolean combination of formulas of the form $\Box\psi$ or $U\psi$ where ψ has modal rank less or equal to $n - 1$. Thus, it suffices to prove that $X \models \Box\psi$ iff $X' \models \Box\psi$ and that $X \models U\psi$ iff $X' \models U\psi$. Without loss of generality, let us consider the first model. Suppose that $X \models \Box\psi$. By the truth definition there exists an open o (with $x \in o$) such that $\forall u \in o : X, z \models \psi$. Now, assume that the n-round game starts with Spoiler choosing o in X. Using the strategy σ, Duplicator can pick an open o' such that $x' \in o'$ and $\forall u' \in o' : X, u' \models \psi$. Now Spoiler can pick any point u' in o'. Duplicator can use the information in σ to respond with a point $u \in oBox$, concluding the first round, so that the remaining strategy σ' is still winning for Duplicator in $TG(X, X', n-1)$. By the inductive hypothesis, the fact that $X, u \models \psi$ (where ψ has modal rank $n - 1$) implies that $X', x' \models \psi$. Thus we have shown that all $u' \in o'$ satisfy ψ, and hence $X', x' \models \Box\psi$. The other direction is analogous. Suppose now that $X \models U\psi$. By the truth definition for all $x \in X$ such that $X, x \models \psi$. Ab absurdum, $X\neg \models U\psi$, hence $X \models E\neg\psi$. By the truth definition, $\exists x' \in X'$ such that $X', x' \models \neg\psi$. Spoiler can choose the x' point as his first move. Duplicator's choice on X is necessarily a point x such that $X, x \models \psi$, hence Duplicator cannot win the game $TG(X, X', n-1)$, contradicting the induction hypothesis. The other direction is analogous.

The right to left direction is again proven by induction on n. If $n = 0$, then X and X' satisfy the same non-modal formulas. In particular, they satisfy the same atoms, which is winning for Duplicator, by the definition of topological game. For the inductive step, without loss of generality, let us assume that in the first (global) round of $TG(X, X', n)$ Spoiler chooses the point x. Consider a generic open o containing x. Now, take the set $\{\mathrm{DES}_{n-1}(z) : z \in o\}$, where $\mathrm{DES}_{n-1}(z)$ denotes all the formulas up to modal rank $n-1$ satisfied at z. This set is not finite per se, but we can simply prove the following.

Fact A.2 (Logical Finiteness) There are only finitely many formulas of modal depth k up to logical equivalence.

Therefore, we can write one boolean formula to describe this open set o, namely $\bigvee \bigwedge \mathrm{DES}_{n-1}(z)$. Since this is true for all $z \in o$, by the truth definition we have that $X, x \models \Box \bigvee \bigwedge \mathrm{DES}_{n-1}(z)$ (a formula of modal rank n). By hypothesis, x and x' satisfy the same modal formulas of modal rank n, so $X', x' \models \Box \bigvee \bigwedge \mathrm{DES}_{n-1}(z)$. This last fact, together with the truth definition implies that there exists an open o' such that $\forall z' \in o' : X', z' \models \bigvee \bigwedge \mathrm{DES}_{n-1}(z)$. This is the open that Duplicator must choose to reply to Spoiler's move. Now Spoiler can pick any point u' in o'. Such a point satisfies at least one disjunct $\bigwedge \mathrm{DES}_{n-1}(z)$, and we let Duplicator respond with $z \in o$. As a result of this first round, z, u' satisfy the same modal formulas up to modal depth $n-1$. Hence by the inductive hypothesis, Duplicator has a winning strategy for $TG(X, X', n-1)$. Putting this together with our first instruction, we have a winning strategy for Duplicator in the n-round game.

Proof of Theorem 4.4 on page 9

tmd satisfies the three properties of distances; i.e., for all X_1, $X_2 \in T$:

(i) $tmd(X_1, X_2) \geq 0$ and $tmd(X_1, X_2) = 0$ iff $X_1 = X_2$

(ii) $tmd(X_1, X_2) = tmd(X_2, X_1)$

(iii) $tmd(X_1, X_2) + tmd(X_2, X_3) \geq tmd(X_1, X_3)$

As for (i), from the definition of topo-games it follows that the amount of rounds that can be played is a positive quantity. Furthermore, the interpretation of $X_1 = X_2$ is that the spaces X_1, X_2 satisfy the same modal formulas. If Spoiler does not have a w.s. in $\lim_{n\to\infty} TG(X_1, X_2, n)$ then X_1, X_2 satisfy the same modal formulas. Thus, one correctly gets

$$tmd(X_1, X_2) = \lim_{n\to\infty} \frac{1}{n} = 0.$$

Equation (ii) is immediate by noting that, for all X_1, X_2, $TG(X_1, X_2, n) = TG(X_2, X_1, n)$.

As for (iii), the triangular property, consider any three models X_1, X_2, X_3 and the three games playable on them,

$$TG(X_1, X_2, n),\ TG(X_2, X_3, n),\ TG(X_1, X_3, n) \tag{A.1}$$

Two cases are possible. Either Spoiler does not have a winning strategy in all three games (A.1) for any amount of rounds, or he has a winning strategy in at least one of them.

If Spoiler does not have a winning strategy in all the games (A.1) for any number of rounds n, then Duplicator has a winning strategy in all games (A.1). Therefore, the three models satisfy the same modal formulas, $spw \to \infty$, and $tmd \to 0$. Trivially, the triangular property (iii) is satisfied.

Suppose Spoiler has a winning strategy in one of the games (A.1). Via Theorem 4.2 (adequacy), one can shift the reasoning from games to formulas: there exists a modal formula γ of multi-modal rank m such that $X_i \models \gamma$ and $X_j \models \neg\gamma$. Without loss of generality, one can think of γ as being in normal form:

$$\gamma = \bigvee \bigwedge [\neg] U(\varphi_{S4}) \tag{A.2}$$

This last step is granted by the fact that every formula φ of $\mathrm{S4}_u$ has an equivalent one in normal form whose modal rank is equivalent or smaller to that of φ.[5] Let γ^* be the formula with minimal multi-modal depth m^* with the property: $X_i \models \gamma^*$ and $X_j \models \neg\gamma^*$. Now, the other model X_k either satisfies γ^* or its negation. Without loss of generality, $X_k \models \gamma^*$ and therefore X_j and X_k are distinguished by a formula of depth m^*. Suppose X_j and X_k to be distinguished by a formula β of multi-modal rank $h < m^*$: $X_j \models \beta$ and $X_k \models \neg\beta$. By the minimality of m^*, one has that $X_i \models \beta$, and hence, X_i and X_k can be distinguished at depth h. As this argument is symmetric, it shows that either

- one model is at distance $\frac{1}{m*}$ from the other two models, which are at distance $\frac{1}{l}$ ($\leq \frac{1}{m*}$), or
- one model is at distance $\frac{1}{h}$ from the other two models, which are at distance $\frac{1}{m^*}$ ($\leq \frac{1}{h}$) one from the other.

It is a simple matter of algebraic manipulation to check that m^*, l and h, m^* (as in the two cases above), always satisfy the triangular inequality.

[5] In the proof, the availability of the normal form is not strictly necessary, but it gives a better impression of the behavior of the language, [3].

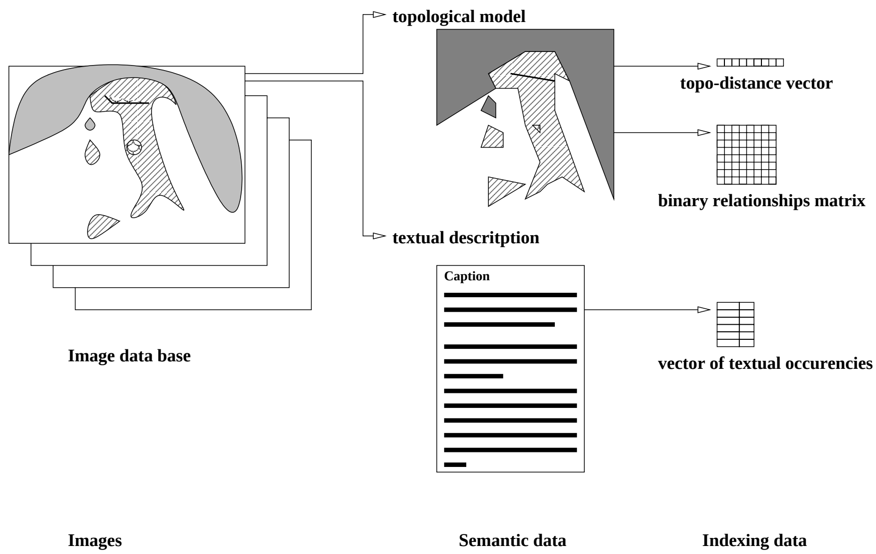


FIG. 4: The conceptual organization of IRIS together with the indexing data structures.

B The IRIS prototype

The ultimate step toward practice of the spatial framework presented in the paper is the actual implementation of the similarity measure in a prototype. The topo-distance is a building block of an image retrieval system, named IRIS Image RetrIeval based on Spatial relationships, coded in Java and enjoying a Swing interface (Figure 5).

The actual similarity measure is built in IRIS to both index and retrieve images on the basis of:

(i) The spatial intricacy of each region,

(ii) The binary spatial relationships between regions, and

(iii) The textual description accompanying the image.

Referring to Figure 4, one can get a glimpse of the conceptual organization of IRIS. A spatial model, as defined in Section 2, and a textual description (central portion of the figure) are associated with each image of the collection (on the left). Each topological model is represented by its topo-distance vector, as built by the algorithm in Section 5 and by a matrix of binary relationships holding between regions. Similarly, each textual description is indexed holding a representative textual vector of the text (right portion of the figure). In Figure 5, a screen-shot from IRIS after querying a database of about 50 images of men and cars is shown. On the top-right is the window for sketching queries. The top-center window serves to write textual queries and to attach information to the sketched regions. The bottom window shows the results of the query with the thumbnails of the retrieved images (left to right are the most similar). Finally, the window on the top-left controls the session.

We remark again the importance of moving from games to a distance measure and of identifying the topo-vectors for actually being able to implement the spatial framework. In particular, in IRIS once an image is place in the data-base the topo-vector for its related topological model is computed, thus off-line, and it is the only data structure actually used in the retrieval process. The representation is quite compact both if compared with the topological model and with the image itself. In addition, the availability of topo-vectors as indexing structures enables us to use a number of information retrieval optimizations, [20].

In IRIS, the similarity measure is built of three components:

$$similarity(I_q, I_j) = \frac{1}{k_n}(k_u^{\text{topo}} \cdot d_{\text{topo}}(I_q, I_j) + k_u^{\text{b}} \cdot d_{\text{b}}(I_q, I_j) + k_u^{\text{text}} \cdot d_{\text{text}}(I_q, I_j))$$

Fig. 5. The result of querying a database of men and cars.

where I_q is the query image (equipped with its topological model and textual description), I_j is the j-th image in the visual database, k_u^{topo}, k_u^{b}, and k_u^{text} are user defined factors to specify the relative importance of topological intricacy, binary relationships and text in the querying process, k_n is a normalizing factor, $d_{\text{topo}}(I_q, I_j)$ is the topo-distance between I_q and I_j, $d_{\text{b}}(I_q, I_j)$ and $d_{\text{text}}(I_q, I_j)$ are the distances for the binary spatial relationships and for the textual descriptions, respectively.

The entire Section 5 is concerned with the computation of $d_{\text{topo}}(I_q, I_j)$. The topo-distance component is simply:

$$d_{\text{topo}}(I_q, I_j) = \text{topo-distance}(\text{t-vec}(I_q), \text{t-vec}(I_j))$$

The second component $d_{\text{b}}(I_q, I_j)$ of the similarity measure accounts for the binary spatial relationships between objects. When an image is indexed, a matrix is built. This is a square matrix whose indices range over the regions present in the model. The generic entry $e_{i,j}$ of the matrix represents the spatial relationship between region i and region j and can be one of the following: disconnected, externally connected, overlap, equal, tangential part, non-tangential part, and the inverses of the last two (RCC8). Following [19], we define a topological distance using RCC8 in the following way. Any two relations are at distance n if there is a path of length n in the graph in Figure 6 connecting the two nodes representing the relations. Our distance is slightly different from that in [19] since we use a modification of its original graph, though the underlying idea is the same. In the similarity measure, one compares matrices $\text{b}(M_1, M_2)$:

$$d_{\text{b}}(I_q, I_j) = \text{b}(\text{b\_matrix}(I_q), \text{b\_matrix}(I_j))$$

where $\text{b\_matrix}(I_j)$ is the matrix of binary RCC8 relations associated with the regions identified in the j-th image.

The third and last component $d_{\text{text}}(I_q, I_j)$ of the similarity measure deals with textual annotation. The motivation comes from captions accompanying images in paper documents or present 'near' images in hyper-media documents. We employ quite standard textual information retrieval techniques, see for instance [20], and therefore omit further explanation of this part of the similarity

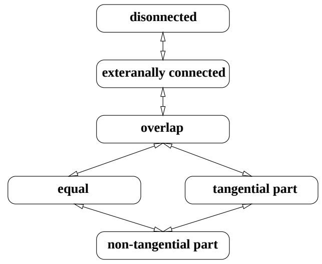


FIG. 6. The binary relationships graphs.

measure behalf for the standard definition of 'textual distance' between two image descriptions:

$$d_{\text{text}}(I_q, I_j) = (1 - \frac{\text{weighted\_occurrences}(\text{text\_vector}(I_q), \text{text\_vector}(I_j))}{\text{length}(\text{text\_vector}(I_q))})$$

where text_vector(I_j) is the list of meaningful words found in the description of the j-th image, weighted_occurences counts the number of instances of a word appearing in two textual vectors weighted by a factor indicating the indexing power of the word. A word is more powerful if it discriminates more, which in turns means that it occurs in less descriptions in the whole collection of image captions. The $d_{\text{text}}(I_q, I_j)$ follows a common way of defining a cosine distance among word vectors, see for instance [34].

Received 2001

Disjunctions and Specificity in Suppositional Defeasible Argumentation

GUSTAVO ADRIAN BODANZA, *Centro de Investigaciones de Lógica y Filosofía de la Ciencia, Universidad Nacional del Sur, 8000 Bahía Blanca, Argentina. E-mail: ccbodanz@criba.edu.ar*

Abstract

This work introduces a system of suppositional argumentation (SAS), trying to give a foundation for dealing intuitively with disjunctive information in a defeasible reasoning framework. Defeasible argumentation systems proposed in the field of Artificial Intelligence lack in general of such a capability. Our view is that suppositional reasoning is present in defeasible arguments involving disjunctions, just as in reasoning by cases in classical logic. Disjunctive information can express different plausible alternatives which consideration would improve the results of a debate. Here is studied in what extent an argument assuming such plausible alternatives can be considered relevant within the given context, and how those alternatives can be compared on basis of their explicative power. In consequence, a debate can be affected in several aspects, among which counter argumentation, defeat and justification have to be considered. Moreover, a comparison among arguments using specificity is adopted, obtaining that also defeasible contrapositive arguments are treated intuitively. Interesting properties of the system (consistency, a deduction theorem, reasoning by cases) are proved, and common sense rationality is tested with several benchmark problems.

Keywords: Defeasible argumentation, suppositional reasoning, disjunctive information, contraposition, specificity.

1 Introduction

An important effort to formalize defeasible reasoning has been made in the last fifty years through the study of defeasible argumentation systems, in the field of Artificial Intelligence (Poole [17], Loui [8], Simari & Loui [25], Prakken [20], Vreeswijk [31]), and recently also in the fields of legal argumentation (Kowalski & Toni [5]), Prakken & Sartor [21], Verheij [30], etc.) and negotiation (Parsons et al. [10], Tohmé [28], etc.). Argument systems evolved from philosophical ideas on non-deductive inferences, principally those of Toulmin [29], Kyburg [6] and Rescher [23]. These systems are not *logics* but *models* of defeasible argumentation that intend to capture one or more of the following aspects:

- how defeasible arguments are constructed;
- which arguments can be considered relevant with respect to a given context;
- how arguments can be compared;
- which arguments are defeated;
- which arguments are justified.

From the point of view of Artificial Intelligence, argument systems are aimed to find both the theoretical basis of defeasible argumentation and the way of translating this into sound computer programs. In our opinion, this dual purpose restricts something the comprehension of the deep philosophical foundations of defeasible argumentation. For example, much of the work in the literature restricts the language to Horn clauses, which facilitates implementation in detriment of expressive capabilities. This, moreover, is sometimes accompanied with restrictions on the inference of *prima facie* conclusions: being α a *prima facie* reason for β, β can be tentatively concluded only in case α is informed. This restriction clearly impedes suppositional reasoning. Consider the exclusion of reasoning by cases: if γ is a *prima facie* consequence of α as well as of β, then γ should be a *prima facie* consequence of $\alpha \vee \beta$; nevertheless, if neither α nor β are derived, γ can not be concluded. The point is that, even when this kind of restrictions may improve computability, the resulting programs represent idealizations that seem too strong for a rational agent. On the other hand, the philosophical basis for such idealizations are usually counterintuitive.

An outstanding approach can be found in John Pollock's system *OSCAR* [11, 12, 13, 14, 15]. This system allows suppositional argumentation through *prima facie* reasons that connect formulae of a non restricted first order language. Reasoning by cases, for instance, is possible in *OSCAR* since *dilemmas* are essentially suppositional forms of reasoning. We assume here that Pollock's approach is right in general. On the other hand, the introduction of suppositional arguments usually has drawbacks that have not been addressed in *OSCAR*. More precisely, since defeasible arguments have to interact dialectically with other arguments, it becomes necessary to determine whether a suppositional argument is relevant to the context of discussion or not. In this paper we shall see, in particular, that disjunctive information makes relevant some suppositional arguments, improving the rationality of the justifications in a context involving such information. A formalism for suppositional defeasible argumentation will be provided, which enables reasoning with disjunctions beyond the simple ability for constructing dilemmas.

Basically, the proposed system will use contextual disjunctive information to block credulous arguments. Consider, for example, an ideal detective investigating a murder. He has got evidence that the murder was committed by Bonnie or by Clyde, but he does not precisely know by which one of them; he knows that the law says that murderers are *prima facie* guilty, and that people is *prima facie* innocent (unless the contrary is proven). Within this context, should the detective believe that Bonnie *and* Clyde are both innocent (here we take 'innocent' and 'guilty' as opposed)?

Note that this problem covers all the five aspects of defeasible argumentation mentioned above:

- *Argument construction*: we need to deal with suppositional defeasible arguments, since the conclusion that both Bonnie and Clyde are innocent must be counterargued by an hypothetical analysis: Bonnie would not be innocent in the case that she was the murderer, which is a supposition; the same occurs with Clyde.
- *Relevance of suppositional arguments*: the supposition that Bonnie is the murderer, as well as the supposition that Clyde is the murderer, leaves place for a relevant argument that blocks the conclusion that both of them are innocent, because of the contextual disjunctive information that at least one of them is the murderer. Without this information (or any other) such supposition would not

leave place for a relevant argument.

- *Comparison among arguments*: a preference relation must be established among arguments to decide which ones are stronger than other ones. Consider, in our example, the argument assuming the possibility of Bonnie being the murderer and the argument concluding that both Bonnie and Clyde are innocent. It seems rational to intend to establish a preference between these arguments since both are equally relevant with respect to the contextual information. Instead, an argument assuming that, say, Lucky is the murderer might be preferred to that for Bonnie being guilty, but not in the present context where it is known that the murderer is either Bonnie or Clyde. On the other hand, preference may be established on several grounds. We shall show that *specificity*, a usual criterion in argument systems, can be applied naturally to suppositional argumentation.
- *Defeat among arguments*: preferences are to be used to determine defeat among arguments when conflicts arise. At this point, it will be important to note that in some cases a relevant suppositional argument can be defeated if evidence can be used against its suppositions, even when these are plausible. In our example, this would be the case if new evidence provides an alibi for Bonnie, which would reject the argument for Bonnie's guilt supposing she is the murderer.
- *Argument justification*: any argument, suppositional or not, may be justified at the end of the process unless there exists a justified contextually relevant argument against it. Of course, warranted conclusions (or *defensible*, as we shall call them) will be only those supported by non-suppositional arguments, that is, arguments for which any supposition introduced for establishing the proof has been discharged. In the example, the detective could have neither justified arguments for believing that both Bonnie and Clyde are innocent (since there exist relevant suppositional counterarguments) nor justified arguments about who is actually the murderer (since there are not justified non-suppositional arguments for this).

All these problems can be approached modularly, as exhibited by the different proposals for argument systems. For instance, Lin et al. [7] give a notion of argument, Dung [1] deals with the argument justification problem, Poole [17] focused on comparison among arguments through the study of specificity, and Loui [8] and Simari et al. [25] approached all the mentioned subjects. Given the nature of the problem we are going to deal with, our work will be mainly centered on the development of the two first items. The third, that of preference, will be based on Poole's specificity. Defeat will be defined on the basis of that preference criterion, and justification will be based on a Simari & Loui's reformulation of Pollock's notion of warrant.

1.1 A motivating example and organization of the paper

For our study we shall analyze the structure of an example given by Poole([19], p. 284) (a structure similar to that of the above example of Bonnie and Clyde). Suppose that we know two birds, Opus and Tweety, at least one of which is a penguin, but we do not actually know which of them is it. Accepting that birds tentatively fly and penguins tentatively do not fly (assuming that exceptions may exist to this), shall we believe that, say, Tweety, flies? On the other hand, shall we believe that it does not fly? It seems reasonable to think that we cannot give an affirmative answer to neither of these questions. The reason why we cannot claim that Tweety flies is that *in the case* that this bird is the penguin, then it would not fly; and the reason why we cannot claim that Tweety does not fly is that *in the case* that Opus is the penguin, not Tweety, then Opus would fly.

As this problem shows, we have to deal with arguments to assert, for example, that *assuming* or *supposing* that Tweety is a penguin then (tentatively) it would not fly. Arguments like this will be used to block credulous arguments like that concluding that both Tweety and Opus fly since they are birds. Suppositional argument structures will be defined within a suitable language (Section 2) for constructing a system similar to Pollock's *OSCAR* (Section 3). This will be sufficient to achieve defeasible reasoning by cases. Then we shall investigate examples like that of Poole where debates carried out with defeasible arguments are particularly affected by disjunctive information. In the debate about flight, for example, disjunctive information makes relevant the hypothesis that Tweety is a penguin: this relevance is based on the contextual fact that *at least one* between Tweety and Opus *is* a penguin. Following this intuition, we shall define a formal criterion to decide whether or not an argument is relevant on the basis of the contextual disjunctive information. For this we shall appeal to the minimal models of the formulas representing the contextual information (Section 5.1).

On the other hand, we shall study some circumstances under which relevant suppositional arguments can be objected. In particular, we shall refer to cases in which the hypotheses of relevant suppositional arguments are attacked by non-suppositional (*founded*, as we shall call them) arguments (Section 6). This notion together with Poole's *specificity* (Section 4) will give us a criterion for defeat. Defeat will complete the characterization of *justification*, which will be defined following the approach by Simari & Loui (Section 3.1).

2 Language for arguments

Let us introduce a formal language for *arguments*. Let $\mathcal{L}$ be a first-order language, and let Δ be a meta linguistic binary relation over sentences of $\mathcal{L}$. The corresponding ordered pairs are called *defeasible rules*[1]. A defeasible rule is symbolized by '$\alpha \succ\!\!- \beta$', which is intended to mean 'α is a good reason for supporting β'[2]. Free variables occurring in formulae of $\mathcal{L}$ will be interpreted as being universally quantified. Now we define

Definition 2.1 *An* argument *is a structure $\langle Def, Sup, \sigma\rangle$, where $Def \subseteq \Delta$ is a finite set of grounded instances of defeasible rules (i.e., defeasible rules in which formulae contain only individual constants), called the argument's Def*easible support*; $Sup \subset \mathcal{L}$ is a finite set, called the argument's Sup*positions*; and $\sigma \in \mathcal{L}$ is called the argument's* conclusion.

If $\langle Def, Sup, \sigma\rangle$ is an argument such that $Def = \emptyset$, we shall say that it is a *conclusive* (or *non-defeasible*) *argument* for σ; if $Sup = \emptyset$ then we shall say that it is a *founded* (or *non-suppositional*) *argument*.

---

[1] These are the same kind of entities appearing in the systems of Loui [8] and Simari & Loui [25].

[2] From now on we shall drop quotations when terms are clearly mentioned.

ARGUMENTS	CONCLUSIVE	DEFEASIBLE
FOUNDED	*-without defeasible rules* *-without suppositions*	*-with defeasible rules* *-without suppositions*
SUPPOSITIONAL	*-without defeasible rules* *-with suppositions*	*-with defeasible rules* *-with suppositions*

TABLE 1. A taxonomy of arguments.

Hence arguments may be conclusive or defeasible, founded or suppositional —cf. table 1.

We introduce the symbol '$\Rightarrow_\Delta$' to denote a relation between sets of arguments and arguments, called *argumentative consequence*. The expression '$Args{\Rightarrow_\Delta}arg$' denotes that the argument arg is an argumentative consequence of the set of arguments $Args$[3]. The relation is defined by the following rules:

Sup (supposition):
For all $Sup \subset \mathcal{L}$ and for all $\sigma \in Sup$, $\emptyset \Rightarrow_\Delta \langle \{\}, Sup, \sigma\rangle$

Deduction:
If $\{\theta_1, \ldots, \theta_n\} \vdash \sigma$[4] then $\{\langle Def_1, Sup_1, \theta_1\rangle\}, \ldots, \langle Def_n, Sup_n, \theta_n\rangle\} \Rightarrow_\Delta \langle Def_1 \cup \ldots \cup Def_n,\ Sup_1 \cup \ldots \cup Sup_n,\ \sigma\rangle$

DMP (defeasible modus ponens):
For any set of defeasible rules Δ and for every grounded instance of a defeasible rule $\theta {>\!\!-} \sigma \in \Delta$, $\{\langle Def, Sup, \theta\rangle\}{\Rightarrow_\Delta}\langle Def \cup \{\theta {>\!\!-} \sigma\}, Sup, \sigma\rangle$

Cond (conditionalization):
$\{\langle Def, Sup \cup \{\theta\}, \sigma\rangle\}{\Rightarrow_\Delta}\langle Def, Sup, (\theta \rightarrow \sigma)\rangle$

These rules are interpreted as valid steps in the construction of defeasible arguments. Any sequence of these valid steps from a set of arguments to a distinguished argument is called a *defeasible proof*. As it is evident in the previous list, we are particularly interested in including argumentative versions of all the valid rules of inference of classical logic. The rule **Cond** is special in that it allows to discharge suppositions. This implies that defeasible rules can be used to derive material implications, but those material implications could be supported only defeasibly.

Example 2.2 *From $\{\langle \{\}, \{\theta\}, \theta\rangle\}$ and the defeasible rule $\theta {>\!\!-} \sigma \in \Delta$, we derive $\langle \{\theta {>\!\!-} \sigma\}, \{\theta\}, \sigma\rangle$ by* **DMP***; then, by* **Cond** *we get $\langle \{\theta {>\!\!-} \sigma\}, \{\}, (\theta \rightarrow \sigma)\rangle$. So, in this argument the material implication $\theta \rightarrow \sigma$ has a defeasible support given by $\theta {>\!\!-} \sigma$.*

Definition 2.3
We say that an argument $\langle Def', Sup', \theta\rangle$ is a subargument *of $\langle Def, Sup, \sigma\rangle$ iff*

1. *$Def' \subseteq Def$, and*
2. *$Sup' \subseteq Sup$.*

[3]The symbol '$\Rightarrow_\Delta$' means that the applied defeasible rules used in the derivation are elements in Δ.
[4]'$vdash$' stands for the syntactical consequence relation in classical logic.

The notion of *subargument* will play a key role in determining defeat among arguments (this will be treated in Section 6).

Example 2.4 *For the above example 2.2, we have a defeasible proof in the following sequence:*

(1)	$\langle \{\}, \{\theta\}, \theta \rangle$	*(***Sup***)*
(2)	$\langle \{\theta \succ\!\!- \sigma\}, \{\theta\}, \sigma \rangle$	*(***DMP***, 1)*
(3)	$\langle \{\theta \succ\!\!- \sigma\}, \{\}, (\theta \rightarrow \sigma) \rangle$	*(***Cond***, 2)*

The argument in line (1) is a subargument of the argument in line (2), but neither of these is a subargument of that in line (3). Moreover, each one is a trivial subargument of itself.

It is easy to see that defeasible proofs can be established for dilemmas (in particular, for reasoning by cases) and contraposition, forms of reasoning which are rare beasts in the best known default formalisms.

In the next section we show how an ideal agent's defeasible knowledge can be systematized using this language, and how the interaction among arguments can be modeled to define the resulting justified (or "warranted") arguments.

3 *SAS*: suppositional argument systems

An agent's defeasible knowledge can be represented in a suppositional argument system. A *suppositional argument system* is a pair $SAS = \langle \mathcal{K}, \Delta \rangle$, where $\mathcal{K}$ is a finite and consistent set of formulae of $\mathcal{L}$, called *the context* of SAS, and every element of $\mathcal{K}$ stands for a basic belief of the agent (an axiom of the system). Δ is a finite set of defeasible rules. We define $[\mathcal{K}] = \{arg_i\}$, where each arg_i is the conclusive founded argument for ϕ_i, for each $\phi_i \in \mathcal{K}$; $[\mathcal{K}]$ expresses the "argument-axioms" of SAS. The set of all the "argument-theorems" of SAS is contained in the closure under $\Rightarrow_\Delta$ of $[\mathcal{K}]$, which will be denoted by $[[\mathcal{K}]]$. These argument-theorems represent all the arguments that the agent may consider in order to defend a defeasible inference; so, in considering whether an inference σ can be defended or not in her/his context of beliefs, the agent could find in $[[\mathcal{K}]]$ some arguments supporting σ and others refuting σ. The defensibility or not of σ will arise from the interaction of those opposite arguments, as we will see next.

3.1 Argument justification

The aim of this section is to anticipate a notion of justification. We will not discuss how to define a justification-status assignment to arguments, which is out of the scope of this paper[5]. Instead, we just shall adopt the notion as defined by Simari & Loui. These authors take the notion up from Pollock, whose idea is that arguments can be defeated in a sequence of levels, where all arguments are *in* at level-0, and an argument is *in* at level-$(n+1)$ if it is non-defeated by an argument *in* at level-n (cf. Pollock [11] and also [14] for a more refined definition that deals with self-defeating arguments). Simari and Loui reformulate the notion by considering whether *in* arguments are

[5] Readers who are interested in this theme can find in Dung [1] a very treatise.

active as *supporting* or *interfering* arguments, differentiating attacked arguments from defeated ones.

By the moment we appeal to the reader's intuition to understand the involved terms 'attack' and 'defeat'. That intuition will be enough by now, but the terms shall be formally defined later in Section 6. Let us say simply that a defeater is an attacker that is "stronger" than the defeated one, and that it is not transitive.

Definition 3.1 *(Adopted from Simari & Loui [25].) An argument is active at various levels as either a* supporting *or an* interfering *argument:*

1. *All arguments are* level-(0) *S-arguments (supporting arguments) and I-arguments (interfering arguments).*
2. *An argument is a* level-(n+1) *S-argument iff it is not attacked by a* level-(n) *I-argument.*
3. *An argument is a* level-(n+1) *I-argument iff it is not defeated by a* level-(n) *I-argument.*

Then the justified arguments are identified as follows:

Definition 3.2 *An argument* $\langle Def, Sup, \sigma\rangle \in [[\mathcal{K}]]$ *is* justified *iff there exists* m *such that for all* $n \geq m$, $\langle Def, Sup, \sigma\rangle$ *is a level-(n) S-argument.*

Definition 3.3 *A sentence* $\sigma \in \mathcal{L}$ *is* defensible *in* SAS *iff there exists a justified founded argument for* σ *in* SAS.

Let us consider some examples.

Example 3.4 *Consider our introductory motivating example. We have the context:*

$$\begin{aligned}\mathcal{K} \ = \{&penguin(x) \rightarrow bird(x),\\ &bird(Opus),\ bird(Tweety),\\ &penguin(Opus) \vee penguin(Tweety)\}\end{aligned}$$

and the following defeasible rules:

$$\begin{aligned}\Delta \ = \{&bird(x) \succ\!\!- flies(x),\\ &penguin(x) \succ\!\!- \neg flies(x)\}.\end{aligned}$$

Then we can obtain —among others— these arguments:

$$arg1 = \langle\{bird(Opus) \succ\!\!- flies(Opus)\},\ \{\},\ flies(Opus)\rangle,$$

$$arg2 = \langle\{bird(Tweety) \succ\!\!- flies(Tweety)\},\ \{\},\ flies(Tweety)\rangle,$$

$$arg3 = \langle\{penguin(Opus) \succ\!\!- \neg flies(Opus)\},\ \{penguin(Opus)\},\ \neg flies(Opus)\rangle,$$

$$\begin{aligned}arg4 = \langle&\{penguin(Tweety) \succ\!\!- \neg flies(Tweety)\},\ \{penguin(Tweety)\},\\ &\neg flies(Tweety)\rangle.\end{aligned}$$

Note that arguments $arg1$ and $arg2$ have no suppositions since their premises $bird(Opus)$ and $bird(Tweety)$, respectively, are contextual facts. The arguments $arg3$ and $arg4$, have $penguin(Opus)$ and $penguin(Tweety)$ as suppositions, respectively.

This example shows a first feature of the system: through suppositional argumentation we can bring the contextual disjunctive information into the debate to do something more than reasoning by cases. The second is how suppositional arguments using that information improve the rationality of the debate. Assuming that $arg3$ defeats $arg1$ and $arg4$ defeats $arg2$, then, following definition 3.1, arguments $arg3$ and $arg4$ are justified. So, no conclusion (neither affirmative nor negative) about the capacity of flight of Tweety and Opus would be defensible, since there are no justified founded arguments about that, neither pro nor con.

Another thing to note is the system's capability to use defeasible rules in contrapositive reasoning. In spite of the fact that some undesirable arguments could be constructed by means of contrapositive reasoning, these are innocuous. For example, assuming $bird(Opus)$ we can infer successively:

$\langle\{bird(Opus) \succ\!\!- flies(Opus)\}, \{bird(Opus)\}, flies(Opus)\rangle$ (by **DMP**)

$\langle\{bird(Opus) \succ\!\!- flies(Opus)\}, \{\}, bird(Opus) \rightarrow flies(Opus)\rangle$ (by **Cond**)

and then, together with $arg3$ and through *Deduction*:

$\langle\{penguin(Opus) \succ\!\!- \neg flies(Opus), bird(Opus) \succ\!\!- flies(Opus)\}, \{penguin(Opus)\}, \neg bird(Opus)\rangle$.

But this undesirable argument (undesirable because we cannot accept that Opus is not a bird assuming it is a penguin) should be defeated by a conclusive argument under the same supposition:

$\langle\{\}, \{penguin(Opus)\}, bird(Opus)\rangle$

as well as by the founded conclusive argument taking $bird(Opus)$ just as a fact:

$\langle\{\}, \{\}, bird(Opus)\rangle$.

On the other hand, contrapositive reasoning may be suitable in some contexts. Consider, for example, a context where sport cars tend not to be tourism cars, Fiats are usually tourism cars, Ferraris are usually sport cars, Fiat cars tend to be non-fast, Ferrari cars tend to be fast, and Peter's car is a Fiat or a Ferrari, but it is known to be a tourism car. The question is: is Peter's car fast or not? Note that the fact that Peter's car is a tourism car "confirms" the hypothesis that it is a Fiat while "refutes" the hypothesis that it is a Ferrari. So, the conclusion that it is non-fast appears to be more intuitive. The formal treatment of this example will be shown in Section 6, after we develop more theoretical aspects.

The next example shows the system's capability for reasoning by cases.

Example 3.5 *Consider a norm for no parking at some determined place in a street, with exception to ambulances and school buses. We can formulate the context:*

$\mathcal{K} = \{ambulance(x) \rightarrow vehicle(x),$
$\quad schoolbus(x) \rightarrow vehicle(x),$

$vehicle(a),$
$ambulance(a) \vee schoolbus(a)\}$;

and the defeasible rules:

$$\Delta = \{vehicle(x) \succ\!\!- \neg parking(x), \quad ambulance(x) \succ\!\!- parking(x), \quad \mathrm{schoolbus}(x) \succ\!\!- parking(x)\}$$

($parking(x)$ means 'x has parking permission'). Then we can get the following arguments:

$arg1 = \langle \{vehicle(a) \succ\!\!- \neg parking(a)\}, \{\}, \neg parking(a)\rangle$,

$arg2 = \langle \{ambulance(a) \succ\!\!- parking(a)\}, \{ambulance(a)\}, parking(a)\rangle$,

$arg3 = \langle \{schoolbus(a) \succ\!\!- parking(a)\}, \{schoolbus(a)\}, parking(a)\rangle$.

If $arg2$ defeats $arg1$ and $arg3$ defeats $arg1$[6], then arguments $arg2$ and $arg3$ are justified, while $arg1$ is not justified.

4 Specificity

Specificity can be adapted to SAS in a natural way, since suppositional reasoning is at the core idea of the notion. We will follow the main intuitions of Poole [17], and their reformulation by Simari & Loui [25].

Poole views arguments as explanations, more precisely, as explanations in the form of the Hempel's covering-law model. From this point of view, an argument is a set of general (legal), possibly probabilistic statements, which together with some particular information, the antecedent conditions, account for other statement, the *explanandum*. When the set of general statements includes some probabilistic (or *prima facie*, or default) statements, the argument can be compared with a rival in terms of specificity: the more specific argument gives the best explanation. The criterion says, informally, that an argument $argX$ is more specific than another $argY$, given any particular antecedent condition P, if the fact that the legal part of $argX$ together with P is sufficient to derive $argX$'s conclusion, implies that the legal part of $argY$ together with P is also sufficient to derive $argY$'s conclusion, but the inverse is not true. It is important to remark that what is compared is only the legal part of arguments, given *any* particular antecedent conditions.

For example, given the antecedent condition that a is a penguin, we may conclude that a does not fly, based on the general knowledge that penguins are birds that usually do not fly (assuming they do it in remotely strange cases). On the other hand, we may conclude that a flies based on the general knowledge that penguins are birds, and birds usually fly. But the first is the more specific argument, since *any* particular condition which together with 'penguins usually do not fly' enables to conclude 'a does not fly', also enables to conclude that 'a flies' together with 'birds usually fly'; nevertheless, some particular conditions (for example, that a is a bird)

[6] In fact, by means of *specificity* these conditions are true.

enable to conclude 'a flies' through 'birds usually fly', but not to conclude 'a does not fly' through 'penguins usually do not fly'.

In order to formalize this criterion in our symbolism, we need to differentiate the particular (ground) sentences from the general (non-ground) sentences in the context. We divide the context $\mathcal{K}$ of SAS in two subsets $\mathcal{K}_P$ and $\mathcal{K}_G$, containing the former and the later, respectively[7]. So, we can partition also the set $[\mathcal{K}]$ of arguments for contextual sentences in two subsets $[\mathcal{K}_P]$ and $[\mathcal{K}_G]$. Now we can formally express the notion of 'specificity' in SAS as follows.

Definition 4.1
An argument $\langle Def_1, Sup_1, \sigma\rangle$ *is* strictly more specific *than* $\langle Def_2, Sup_2, \theta\rangle$ *iff*

1. *for any set P of ground literals, and for some $Def_1' \subseteq Def_1$, if $\langle Def_1', P, \sigma\rangle \in [[\mathcal{K}_G]]$ then for some $Def_2' \subseteq Def_2$, $\langle Def_2', P, \theta\rangle \in [[\mathcal{K}_G]]$; and*
2. *there exists a set P of ground literals such that*
   (a) *for some $Def_2' \subseteq Def_2$, $\langle Def_2', P, \theta\rangle \in [[\mathcal{K}_G]]$; and*
   (b) *for all $Def_1' \subseteq Def_1$, $\langle Def_1', P, \sigma\rangle \notin [[\mathcal{K}_G]]$; and*
   (c) *$\langle \{\}, P, \theta\rangle \notin [[\mathcal{K}_G]]$ (non-triviality condition).*

Also the relations of *equi-specificity* and *at least as specific as* can be introduced in SAS with similar formulations to those given by Simari & Loui, but that is not important for our purpose.

Example 4.2 (Example 3.4 revisited). *The argument:*

$$arg3 = \langle\{penguin(Opus) \succ\!\!- \neg flies(Opus)\}, \{penguin(Opus)\}, \neg flies(Opus)\rangle$$

is strictly more specific than

$$arg1 = \langle\{bird(Opus) \succ\!\!- flies(Opus)\}, \{\}, flies(Opus)\rangle,$$

since

1. *supposing $penguin(Opus)$ we have:*

   $$\langle\{penguin(Opus) \succ\!\!- \neg flies(Opus)\}, \{penguin(Opus)\}, \neg flies(Opus)\rangle$$

   and

   $$\langle\{bird(Opus) \succ\!\!- flies(Opus)\}, \{penguin(Opus)\}, flies(Opus)\rangle$$

   (since $(penguin(x) \rightarrow bird(x)) \in \mathcal{K}_G$),

2(a) *supposing $bird(Opus)$ we obtain $\langle\{bird(Opus) \succ\!\!- flies(Opus)\}, \{bird(Opus)\}, flies(Opus)\rangle$, but*

[7] This is the same strategy used in the earlier formulations of 'specificity'.

(b) supposing $bird(Opus)$ we cannot obtain any argument for $\neg flies(Opus)$, and also

(c) $\langle \{\}, \{bird(Opus)\}, flies(Opus)\rangle$ cannot be derived.

In the same manner we could show that $arg4$ is strictly more specific than $arg2$.

As we have seen, Poole's specificity is adaptable to SAS with no essential modifications, since suppositional reasoning is at the very heart of the notion. The only restriction we have introduced is that the arbitrary suppositions in P must be ground literals (*i.e.*, ground atoms or their negation), which avoids some undesirable behaviors. Moreover, we consider that literals are more appropriate than arbitrary formulae for expressing the idea that the antecedent conditions of an explanation are particular circumstances. In fact, note that, for example, the formula $\alpha \rightarrow \beta$ does not express a particular circumstance but a set of possible circumstances, to wit, that β is a fact, or α and β are facts, or neither α nor β are facts.

The introduction of specificity in SAS yields important behaviors not seen in some related approaches, as in Simari & Loui's. In particular, in our system specificity allows to decide among defeasible arguments which use contrapositive reasoning. Consider the following framework (meta-variables are used for simplicity):

Example 4.3 (Specificity and contrapositive reasoning)

$\mathcal{K} = \{\neg\beta \wedge \neg\gamma\}$
$\Delta = \{\neg\alpha {>\!\!-} (\beta \vee \gamma), \alpha {>\!\!-} \beta\}$

We can get the arguments:

$arg1 = \langle\{\neg\alpha {>\!\!-} (\beta \vee \gamma)\}, \{\}, \alpha\rangle$

$arg2 = \langle\{\alpha {>\!\!-} \beta\}, \{\}, \neg\alpha\rangle$

($arg1$ can be constructed by assuming $\neg\alpha$ and deriving defeasibly $\beta \vee \gamma$, then applying conditionalization to derive a material conditional discharging the assumption, and next applying deduction (*modus tollens*) using $\neg\beta \wedge \neg\gamma$. $arg2$ can be constructed similarly by assuming α.) Note that $arg1$ is strictly more specific than $arg2$ since any assumption under which an argument for α can be constructed using the $arg1$'s defeasible support, can also be used to construct an argument for $\neg\alpha$ using the $arg2$'s defeasible support, but there is an assumption, to wit, $\{\neg\beta\}$, under which an argument for $\neg\alpha$ can be constructed using the $arg2$'s defeasible support, and it is not possible to use it to construct an argument for α using the $arg1$'s defeasible support.

It is remarkable that in some systems using specificity (*e.g.*, Loui's and Simari & Loui's) two defeasible structures like these would not be comparable. This is because of the postulated one-directionality of defeasible rules, which amounts to define specificity by considering the "activation" of the rules on basis of the availability of their antecedents. In systems like these, of course, contrapositive reasoning is not allowed, so that their authors might find counterintuitive the behaviors we take for rational.

5 Relevance of suppositional arguments with regard to disjunctive information

In this section we will study in what extent disjunctive information can give relevance to suppositional arguments. In our motivating example, as we have seen, some suppositional arguments come into the debate in order to interfere counterintuitive conclusions such as 'Tweety flies' and 'Opus flies'. Why those suppositional arguments are worthy of being taken on account for that interference? May *any* suppositional argument be used for such an interference? In order to look for an answer, let us analyze another example[8].

Example 5.1 *Let the context be* $\mathcal{K} = \{bird(Opus) \vee bat(Opus), penguin(x) \rightarrow bird(x)\}$ *(Opus is a bird or a bat; moreover, all penguins are birds) and let the defeasible rules be* $\Delta = \{penguin(x) \succ\!\!- \neg flies(x),\ bat(x) \succ\!\!- flies(x),\ bird(x) \succ\!\!- flies(x)\}$*. Then we can get, among others, the following arguments:*

$$arg1 = \langle \{penguin(Opus) \succ\!\!- \neg flies(Opus)\}, \{penguin(Opus)\}, \neg flies(Opus) \rangle$$

$$arg2 = \langle \{bird(Opus) \succ\!\!- flies(Opus)\}, \{bird(Opus)\}, flies(Opus) \rangle$$

$$arg3 = \langle \{bat(Opus) \succ\!\!- flies(Opus)\}, \{bat(Opus)\}, flies(Opus) \rangle$$

$$arg4 = \langle \{bird(Opus) \succ\!\!- flies(Opus), bat(Opus) \succ\!\!- flies(Opus)\}, \{\}, flies(Opus) \rangle$$

The suppositional argument $arg1$ says that, assuming that Opus is a penguin, we can conclude tentatively that it does not fly. $arg2$ and $arg3$ suppose respectively that Opus is a bird and Opus is a bat, then they are used to derive $arg4$, which concludes tentatively that Opus flies, discharging the suppositions. If we take Poole's specificity as the preference criterion, $arg1$ is preferred to $arg2$, so $arg2$ is defeated, and the same happens to $arg4$; hence, contrary to our intuition, the tentative conclusion that Opus flies would be overruled.

What is wrong with this? The wrong point is that we have no good reason for assuming that Opus is a penguin, hence $arg1$ is not worth of being considered. From a formal point of view, there is no structural difference neither between $arg1$ and $arg2$, nor between $arg1$ and $arg3$: all of them are formally right. The difference lies, instead, at the *information* level. The suppositions taken by $arg2$ and $arg3$ are both equally plausible alternatives, since we have as *data* that Opus *is* a bird *or* a bat; nevertheless, no contextual information makes plausible the supposition that Opus is a penguin, as $arg1$ assumes. We can extract a criterion from this remark, since contextual information —in particular, information in disjunctive form— can give relevance to some suppositional arguments but not to others: we may say that if a supposition is plausible —in the above sense— then an argument built on it is contextually relevant.

Obviously, we are considering a very partial aspect of relevance. The sense in which we use here the terms 'relevance' and 'contextual relevance' is only in relation to the informational problems discussed above. Moreover, when we characterize suppositional arguments as 'not relevant', we intend to mean that it is not relevant in this

[8] This example was provided by H. Prakken in a personal communication.

sense, and it is assumed to be irrelevant in every other sense[9]. Once this is clear, we can face the problem of formalizing our notion of contextual relevance.

5.1 Plausible alternatives and contextual relevance

As we have said in the above discussion, arguments like $arg1$ are not to be taken as contextually relevant because they are based on suppositions which are not backed up by the context. If, on the other hand, an argument's suppositions are informed as possible cases by a disjunction in the context (as occurs with $arg2$ and $arg3$), then we take them as plausible alternatives, and the argument is to be considered contextually relevant. We shall identify a subset of $[[\mathcal{K}]]$ whose elements are all the contextually relevant arguments on basis of the plausible alternatives they take as suppositions. We shall consider here the semantical level of the information, that is, we shall focus on *what* is informed by the context in spite of *how* it is expressed. In the next section, instead, we shall see how a contextually relevant argument can be constructed syntactically without semantical references.

To begin with, let us introduce some semantical entities. A *model* μ of a set of sentences $A \subset \mathcal{L}$ is an interpretation I assigning the truth value 'true' to every sentence in that set. The model μ can be viewed as a set containing the ground atoms which are true in I, *i.e.*, a Herbrand model. Given two Herbrand models μ_1 and μ_2 of A we say that μ_1 is *at least as preferred* as μ_2 iff $\mu_1 \subseteq \mu_2$. A Herbrand model μ_1 of a set $A \subset \mathcal{L}$ is *preferred* iff there is no other Herbrand model μ_2 of A which is as less as preferred as μ_1, *i.e.*, μ_1 is *minimal*.[10]. As usual, we say that α is true in μ iff $\mu \models \alpha$ (where '$\models$' stands for the (classical) semantical consequence). From now on, we shall talk about a Herbrand model whenever we say 'model', unless other case be specified.

We shall take the notion of preferred model to precise what we intend to mean with 'plausible alternative'. Consider the minimal models of $\mathcal{K}$ in example 5.1: there are two, one containing $bat(Opus)$, and the other containing $bird(Opus)$. Instead, $penguin(Opus)$ is not contained in any minimal model of $\mathcal{K}$(note that though $penguin(x) \rightarrow bird(x)$ is in $\mathcal{K}$, this is true whenever for every x it is true that $\neg penguin(x) \wedge \neg bird(x)$, hence the conditional is true in every model containing neither $penguin(x)$ nor $bird(x)$, for any x). This is the reason why we say that $bat(Opus)$ and $bird(Opus)$ are plausible alternatives, but not $penguin(Opus)$. In fact, we state that

Definition 5.2 *A ground atom α is a* plausible alternative *in a set of sentences S if and only if $\alpha \in \mu$ for some preferred model μ of S.*

Hence, the suppositions on which arguments $arg2$ and $arg3$ are based —$bird(Opus)$ and $bat(Opus)$, respectively— are plausible alternatives in $\mathcal{K}$, while the supposition on which argument $arg1$ is based ($penguin(Opus)$) is not. So, we consider arguments like $arg2$, $arg3$ and $arg4$ as contextually relevant arguments, and discard as such arguments like $arg1$. For the formal definition of 'contextual relevance' we have to

---

[9] A broad discussion about several conditions determining contextual relevance of suppositions can be found in D.Sperber & D. Wilson [27].

[10] Minimal (Herbrand) models semantics was the key in the study of the so called *closed world assumption*, *i.e.*, the assumption that not-α is believed if α is not believed —cf., for instance, Minker [9]. Nevertheless, our use of this semantics is not necessarily related to the CWA.

take into account that plausible alternatives can be just implicit in the suppositions of an argument. In this case, the argument will be contextually relevant if those implicit alternatives are sufficient for the argument to be constructed, using the same defeasible rules to arrive to the same conclusion. These considerations lead us to the next definition:

Definition 5.3 *An argument* $\langle Def, Sup, \sigma\rangle$ *is* contextually relevant *in* $\langle \mathcal{K}, \Delta\rangle$ *iff there exists a (possibly empty) set of atoms* S *such that* $\langle Def, S, \sigma\rangle \in [[\mathcal{K}]]$ *is an argument, and if* $\alpha \in S$ *then it is verified that:*

1. α *is a plausible alternative in* $\mathcal{K}$*, and*
2. $\langle \emptyset, Sup, \alpha\rangle \in [[\mathcal{K}]]$ *(*α *is a logical consequence of* Sup*).*

This definition states that every founded argument is contextually relevant (case $S = \emptyset$) and any argument based on suppositions that imply deductively some plausible alternatives in $\mathcal{K}$, is contextually relevant if those plausible alternatives are sufficient for constructing a contextually relevant argument, for the same conclusion and with the same rules.

Example 5.4 *(Continuation of example 5.1)* $arg4$ *is contextually relevant because it is founded;* $arg2$ *and* $arg3$ *are both contextually relevant because* $bird(Opus)$ *and* $bat(Opus)$ *are both plausible alternatives and* $\langle \emptyset, \{bird(Opus)\}, bird(Opus)\rangle$*,* $\langle \emptyset, \{bat(Opus)\}, bat(Opus)\rangle \in [[\mathcal{K}]]$*. On the other hand,* $arg1$ *is not contextually relevant since* $penguin(Opus)$ *is not a plausible alternative.*

5.2 Syntactical approach to case arguments

Arguments $arg2$ and $arg3$ in example 5.1 are based on suppositions which are components of some disjunction in the context. We may call arguments like these 'case arguments', since their suppositions (if they have some) are possible *cases* expressed by disjunctions. Case arguments give us the idea of what a contextually relevant suppositional argument is. But we have to take into account that non-disjunctive information in $\mathcal{K}$ can also be considered plausible, so case arguments should constitute a broader subclass of arguments than only those depending on information in disjunctive form. For example, $\neg\alpha \to \beta$ says that α happens or β happens, so it can be treated as being $\alpha \vee \beta$. Fortunately, we can get all the information in $\mathcal{K}$ in disjunctive form, transforming $\mathcal{K}$ into a set $\mathcal{K}^{DNF}$, where each formula in $\mathcal{K}$ is rewritten in its *disjunctive normal form*; so, if a formula as the above conditional is contained in $\mathcal{K}$ then the formula $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \beta) \vee (\alpha \wedge \neg\beta)$ is contained in $\mathcal{K}^{DNF}$. Note that this formula expresses all the possible *cases* in which $\neg\alpha \to \beta$ is true, and the context says that at least one of them is true. Hence, an argument assuming that some of these cases are true can be relevantly used for counterarguing an argument that ignores such a possibility.

We shall need some functions, each selecting a single component from each disjunction in $\mathcal{K}^{DNF}$, taking up all the literals occurring in that component.[11] Each selection takes up a set of alternatives informed by the context, and we are interested in the alternatives that *positively* may happen. For instance, for a disjunctive formula $(\neg\alpha \wedge \beta)$

[11] Cf. our use of this function with a similar one given by Geffner *et al.* [4]. See Section 8.

$\vee(\neg\alpha \wedge \neg\beta)$ $\vee(\alpha \wedge \beta)$ there will be some function selecting the component $(\neg\alpha \wedge \beta)$, where the set of alternatives offered is $\{\neg\alpha, \beta\}$, hence the only positive alternative selected by that function with regards to that formula is β. We shall interpret $\neg\alpha$ as saying that α is not an alternative, instead of that not-α is an alternative. Hence, β is the only alternative that occurs in $\{\neg\alpha, \beta\}$. As we shall see, this interpretation is captured by the Herbrand models semantics we used before.

Let us symbolize with f^+ the set of positive literals selected by f, and let us call it a *positive selection*. Then we can define a preference relation $\sqsubseteq$ among positive selections as follows: $f^+ \sqsubseteq g^+$ iff $f^+ \subseteq g^+$ and f has no contradictory literals (the example below shows this possibility). A positive selection f^+ is *minimal* iff for all g^+, if $g^+ \neq f^+$ then $g^+ \not\sqsubseteq f^+$. Minimal positive selections take up the minimal positive facts whose occurrence would confirm the contextual information, so we shall prefer those selections to non-minimal ones. (Non-minimal selections could take up redundant, "non-plausible", information.)

Example 5.5 *Let* $\mathcal{K} = \{\alpha, \neg\alpha \vee \beta\}$*. Then* $\mathcal{K}^{DNF} = \{\alpha, (\neg\alpha \wedge \beta) \vee(\neg\alpha \wedge \neg\beta) \vee(\alpha \wedge \beta)\}$*. Then we have three different positive selections:*

$f^+ = \{\alpha, \beta\}$ *(where* $f = \{\alpha, \neg\alpha, \beta\}$*)*
$g^+ = \{\alpha\}$ *(where* $g = \{\alpha, \neg\alpha, \neg\beta\}$*)*
$h^+ = \{\alpha, \beta\}$ *(where* $h = \{\alpha, \beta\}$*)*

Note that g^+ *is minimal (w.r.t. set inclusion) among the three, but* g *contains contradictory literals, as well as* f*. So, since* h *is the only contradiction-free selection,* h^+ *is the only minimal (w.r.t.* $\sqsubseteq$*) positive selection, thus the only preferred one.*

The correspondence between preferred positive selections and minimal models is established in the next results (such correspondence is expressed in set-theoretical terms, since both positive selections and (Herbrand) models are sets of atoms).

Lemma 5.6 *If* f^+ *is a preferred positive selection on* $\mathcal{K}^{DNF}$*, then there exists a preferred model* μ *of* $\mathcal{K}$ *such that* $\mu \subseteq f^+$*.*

Proof. For any selection f, it is obvious that $f \models \phi$, for all formulae $\phi \in \mathcal{K}$, and if ν is free of contradictory literals, then f^+ itself (which is a set of atoms) is a Herbrand model of $\mathcal{K}$. Now, suppose that f^+ is a preferred positive selection, so that f is free of contradictions. Hence, there exists some Herbrand model μ_i of $\mathcal{K}$ such that $f^+ \models \alpha$, for all atoms $\alpha \in \mu_i$, and since for some minimal model μ, $\mu \subseteq \mu_i$, then $\mu \subseteq f^+$. ∎

Lemma 5.7 *If* μ *is a minimal model of* $\mathcal{K}$*, then there exists a preferred positive selection* f^+ *on* $\mathcal{K}^{DNF}$ *such that* $f^+ \subseteq \mu$*.*

Proof. The lemma follows immediately from the two statements: (1) for every minimal model μ of $\mathcal{K}$, there exists a contradiction-free selection ν such that $f^+ \subseteq \mu$; and (2) f^+ is preferred.

Proof for (1): It is obvious that μ is a Herbrand model for each formula $\phi_i \in \mathcal{K}$. Let $(\delta_{i1} \vee \ldots \vee \delta_{ik})$ the DNF of ϕ_i (clearly, each δ_{ij} $(1 \leq j \leq k)$ is a conjunction of literals). So, for each formula ϕ_i, there exists at least one δ_{ij} for which μ is a Herbrand model. Let now δ be the set of all such δ_{ij}. Clearly, $\delta \equiv \nu$ for some selection f on $\mathcal{K}^{DNF}$. Then, μ is a Herbrand model for f, and since both f^+ and μ are sets of atoms, hence

$f^+ \subseteq \mu$. (Moreover, this implies that f is consistent, otherwise f^+ would contain some negative literal, which is impossible.)

Proof for (2): Suppose by the absurd that, in the above proof, f^+ is not preferred. Then (i) there exists some contradiction-free $g^+ \subset f^+$, and since μ is a Herbrand model of f (as demonstrated above), then (ii) μ is a Herbrand model of f^+, hence (iii) μ is a Herbrand model of g^+. Since f^+ and g^+ are sets of atoms, then $f^+ \subseteq \mu$ follows from (ii), which together with (i) and (iii) implies $g^+ \subset \mu$. But, since it is obvious that $g \models \phi$ for all formulae $\phi \in \mathcal{K}$, then g^+ is a Herbrand model of $\mathcal{K}$ which is lesser than μ, contradicting that μ is minimal. Hence, our hypothesis must be false, that is, f^+ must be preferred. ∎

Lemma 5.8 *For all set of atoms A, A is a preferred positive selection on $\mathcal{K}^{DNF}$ if and only if A is a preferred model of $\mathcal{K}$.*

Proof. Immediate from lemmas 5.6 and 5.7. ∎

Preferred positive selections enable us to give a syntactical account of the notion of 'case argument':

Definition 5.9 *An argument $\langle Def, Sup, \sigma\rangle \in [[\mathcal{K}]]$ is a* case argument *iff there exists a (possibly empty) set of atoms S such that $\langle Def, S, \sigma\rangle$ is an argument in $[[\mathcal{K}]]$, and if $\alpha \in S$ then it is verified that:*

1. *$\alpha \in f^+$ for some preferred positive selection f^+, and*
2. *$\langle \{\}, Sup, \alpha\rangle \in [[\mathcal{K}]]$ (α is a logical consequence of Sup).*

We can get the following result:

Proposition 5.10 *For any argument arg, arg is a case argument in SAS if and only if arg is contextually relevant in SAS.*

Proof. From lemma 5.8 there is an equivalence between condition 1 in definition 5.3 and condition 1 in definition 5.9, and clearly condition 2 in both definitions are the same. ∎

5.3 The complexity of finding $\mathcal{K}^{DNF}$, the selection functions and the minimal models

An important point is how complex is to find $\mathcal{K}^{DNF}$ having $\mathcal{K}$. As it is known, the DNF of a formula is obtained from the lines of the truth table in which that formula is true. Hence, the search space of the problem has 2^n elements, where n is the number of atoms occurring in the formula. So, the problem has the complexity of the satisfiability problem, which is known to be NP-complete[12].

With respect to the total number of selection functions generated by $\mathcal{K}^{DNF}$, let $\mathcal{K}^{DNF} = \{\phi_1, \ldots, \phi_n\}$ be such that each ϕ_i is the DNF of a corresponding formula in $\mathcal{K}$. Then the number of selections is $\prod k_i$, where k_i is the number of disjuncts occurring in ϕ_i. Since this is the size of the search space for preferred positive selections and, by lemma 5.8, preferred positive selections and minimal models are the same, we have that the search space for minimal models has $\prod k_i$ elements.

[12] On this subject see, for instance, Garey & Johnson [3].

In order to know the size of the whole problem, note that the greatest possible number of disjuncts of a formula $\phi_i \in \mathcal{K}^{DNF}$ is 2^n, where n is the number of atoms of ϕ_i. This is the case when ϕ_i is a tautology (because each line of its truth table yields a disjunct in the DNF). Hence, if every $\phi_i \in \mathcal{K}^{DNF}$ is a tautology, the total number of selection functions is $\prod 2^{n_i}$, where n_i is the number of atoms occurring in the formula ϕ_i.

6 How case arguments interact

6.1 A preference among alternatives

In this section we shall propose a preference among plausible alternatives. Our intuition is that, in some systems it is possible to find some plausible alternatives having more "explicative power" than others. What we intend to say with the "explicative power" of plausible alternatives, is that it could be the case that some facts —let us call 'fact' to any literal that is true in every model of $\mathcal{K}$— can be defeasibly explained (or predicted) by one alternative, but not by others; and, on the other hand, it could be the case that some fact refutes (*i.e.*, contradicts) defeasibly one alternative, but not others. For example, suppose we know that Richard is quaker *or* republican, but we know that he is militarist (our factual information); then the alternative where Richard is republican is more explicative than the alternative where he is quaker for two reasons: a) because republicans are usually militarist, hence the facts confirm that Richard is republican; and b) because quakers are usually not militarist, hence the facts refute that Richard is quaker. In order to formalize the notion we will first establish a weak preference and then a stronger one.

Definition 6.1 *We say that a plausible alternative α is* at least as explicative as *a plausible alternative β, in symbols, $\alpha \trianglerighteq \beta$, iff at least one of the following sentences is true:*

- *For some fact $\phi \in \mathcal{K}$, ϕ is defensible in $\langle \mathcal{K}_G \cup \{\alpha\}, \Delta\rangle$ but it is not defensible in $\langle \mathcal{K}_G \cup \{\beta\}, \Delta\rangle$.*
- *For some fact $\phi \in \mathcal{K}$, $\neg\phi$ is defensible in $\langle \mathcal{K}_G \cup \{\beta\}, \Delta\rangle$ but it is not defensible in $\langle \mathcal{K}_G \cup \{\alpha\}, \Delta\rangle$.*

The definition states a comparison between two hypothetical systems with the same general context, and where the only particular information in one of them is α while in the other one is β. Then α is at least as explicative as β iff for some fact ϕ, either ϕ is predicted or explained in α's system but not in β's system, or β's system predicts $\neg\phi$ which is refuted by the fact ϕ.

Lemma 6.2 The relation $\trianglerighteq$ is a weak order (reflexive, transitive and complete) over the set of all the alternatives of any system SAS.

A stronger preference relation is the following:

Definition 6.3 *We say that a plausible alternative α is* more explicative than *an alternative context β, in symbols, $\alpha \triangleright \beta$, iff $\alpha \trianglerighteq \beta$ and $\beta \not\trianglerighteq \alpha$.*

Now, since the relation $\triangleright$ determines chains of plausible alternatives, we have to take only the maximal elements of those chains as the preferred, most explicative alternatives.

Definition 6.4 *A plausible alternative* α *in SAS is* explicatively preferred *iff, for all* β*, if* β *is a plausible alternative in SAS then* $\beta \not\triangleright \alpha$*, i.e.,* α *is a maximal element w.r.t.* $\triangleright$.

Example 6.5 *In example 3.4, both* $penguin(Opus)$ *and* $penguin(Tweety)$ *are explicatively preferred, since the fact* $bird(Tweety)$ *is defensible in* $\langle \mathcal{K}_G \cup \{penguin(Tweety)\}, \Delta\rangle$ *but not in* $\langle \mathcal{K}_G \cup \{penguin(Opus)\}, \Delta\rangle$ *(i.e.,* $penguin(Tweety) \trianglerighteq penguin(Opus)$*), and the fact* $bird(Opus)$ *is defensible in* $\langle \mathcal{K}_G \cup \{penguin(Opus)\}, \Delta\rangle$ *but not in* $\langle \mathcal{K}_G \cup \{penguin(Tweety)\}, \Delta\rangle$ *(i.e.,* $penguin(Opus) \trianglerighteq penguin(Tweety)$*). Hence, neither alternative has more explicative power than the other.*

6.2 Defeat

Finally, we are in conditions of giving a precise meaning to 'defeat'. We consider that a defeater must attack and to be preferred (in a general sense) to some subargument of the defeated argument. In SAS, and following Simari & Loui, this can be implemented through specificity, but also taking on account the explicatively preferred alternatives. Hence the attack relation is defined as follows:

Definition 6.6
We say that an argument $\langle Def_1, Sup_1, \sigma\rangle$ attacks *an argument* $\langle Def_2, Sup_2, \theta\rangle$ *in* $\langle \mathcal{K}, \Delta\rangle$ *iff*

1. $Sup_1 \subseteq Sup_2$ *or* Sup_1 *contains only explicatively preferred alternatives, and*
2. $\mathcal{K} \cup \{\sigma, \theta\} \vdash \perp$.

Definition 6.7 *We say that an argument* $arg1$ defeats *an argument* $arg2$ *iff for some subargument* arg *of* $arg2$ *it is true that*

1. $arg1$ *attacks* arg*, and*
2. $arg1$ *is strictly more specific than* arg.

6.3 Some interesting examples

This section provides some interesting examples. The first is the following[13]. Suppose that a boy, John, believes that girls are romantic or pragmatic, that romantic girls tend to like roses, that pragmatic girls usually do not like roses, and that girls who like romantic movies tend to be romantic. Then he believes that Mary likes romantic movies and, of course, she is romantic or pragmatic. Suppose that John wishes to seduce Mary, it would be a good decision to get her roses for his purpose? Let us represent John's beliefs as follows:

Example 6.8

$$\mathcal{K} = \{romantic(Mary) \vee pragmatic(Mary), \neg lrmovies(Mary)\}$$

$$\begin{aligned}\Delta = \{&romantic(x) \succ\!\!- lroses(x),\\ &pragmatic(x) \succ\!\!- \neg lroses(x),\\ &\neg lrmovies(x) \succ\!\!- \neg romantic(x)\}\end{aligned}$$

[13] Provided by Rodrigo Moro in personal communication.

where the meaning of predicates is:
$pragmatic(x)$*: x is pragmatic;*
$romantic(x)$*: x is romantic;*
$lrmovies(x)$*: x likes romantic movies;*
$lroses(x)$*: x likes roses.*
The query is: Does Mary like roses?
The following arguments can be derived:

$$\begin{aligned}
arg1 = {} & \langle\{romantic(Mary) \succ\!\!\!- lroses(Mary)\}, \{romantic(Mary)\},\\
& lroses(Mary)\rangle\\
arg2 = {} & \langle\{pragmatic(Mary) \succ\!\!\!- \neg lroses(Mary)\}, \{pragmatic(Mary)\},\\
& \neg lroses(Mary)\rangle\\
arg3 = {} & \langle\{\neg lrmovies(Mary) \succ\!\!\!- \neg romantic(Mary)\}, \{romantic(Mary)\},\\
& lrmovies(Mary)\rangle\\
arg4 = {} & \langle\{\neg lrmovies(Mary) \succ\!\!\!- \neg romantic(Mary)\}, \{pragmatic(Mary)\},\\
& lrmovies(Mary)\rangle\\
arg5 = {} & \langle\{\neg lrmovies(Mary) \succ\!\!\!- \neg romantic(Mary),\\
& pragmatic(Mary) \succ\!\!\!- \neg lroses(Mary)\}, \{\}, \neg lroses(Mary)\rangle
\end{aligned}$$

Argument $arg5$ is justified in concluding tentatively that Mary does not like roses. Note that the alternative $pragmatic(Mary)$ is explicatively preferred, while *romantic*(*Mary*) is not, because $\neg lrmovies(Mary)$ confirms the first (see $arg4$) and refutes the later (see $arg3$).

The following example is similar to example 6.8, but here we consider cars. Our context contains the following beliefs: Fiats are Italian and tend not to be fast, even when Italian cars tend to be fast; Jaguars are de luxe and tend to be fast; de luxe cars tend not to be Italian; car c is known to be a Fiat or a Jaguar, and it is known to be de luxe. Is c fast or not? This benchmark problem is formally expressed as follows:

Example 6.9

$$\begin{aligned}
\mathcal{K} = {} & \{fiat(x) \rightarrow italian(x),\\
& jaguar(x) \rightarrow deluxe(x),\\
& fiat(c) \vee jaguar(c),\\
& deluxe(c)\}\\
\mathcal{R} = {} & \{fiat(x) \succ\!\!\!- \neg fast(x),\\
& jaguar(x) \succ\!\!\!- fast(x),\\
& deluxe(x) \succ\!\!\!- \neg italian(x),\\
& italian(x) \succ\!\!\!- fast(x)\}.
\end{aligned}$$

The only founded argument we can form to answer the query is for 'c is fast'. This argument, which uses all the information in $\mathcal{K}$, is:

$$arg1 = \langle\{deluxe(c) \succ\!\!\!- \neg italian(c), jaguar(c) \succ\!\!\!- fast(c)\}, \{\}, fast(c)\rangle$$

(since c is a de luxe car, it is believed not to be Italian; if c is believed not to be Italian, then it is believed not to be a Fiat; if c is believed not to be a Fiat, then it is believed to be a Jaguar, because c *is* a Fiat or a Jaguar; since Jaguars tend to be fast, then c is believed to be fast). Note that $deluxe(c)$ confirms $jaguar(c)$ and refutes $fiat(c)$. $jaguar(c)$ is explicatively preferred but $fiat(c)$ is not, so $fast(c)$ is warranted.

Our system also allows to argue with suppositional non-case arguments, that is, defeat may be established over arguments on purely hypothetical grounds. For example, assuming that d is a car within the above domain and context, we can argue hypothetically as follows:

$$arg2 = \langle \{fiat(d) \succ\!\!- \neg fast(d)\}, \{fiat(d)\}, \neg fast(d) \rangle$$

(assuming d is a Fiat, d is believed not to be fast); moreover

$$arg3 = \langle \{italian(d) \succ\!\!- fast(d)\}, \{fiat(d)\}, fast(d) \rangle$$

(assuming d is a Fiat, then it is Italian; hence d is believed to be fast). Even when $arg2$ and $arg3$ are not case arguments (hence non-contextually relevant, since $fiat(d)$ does not belong to any minimal model), by assuming d is a Fiat we can conclude that it is not fast, since $arg2$ defeats $arg3$ because of specificity.

Let us see one more example of how contrapositive reasoning is used on defeasible grounds. The example is that referred in Section 3.1, page 30, about Peter's car.

Example 6.10 *(Peter's car).*

$$\mathcal{K} = \{fiat(p) \vee ferrari(p),\\ tourism(p)\};$$

$$\Delta = \{fiat(x) \succ\!\!- tourism(x),\\ fiat(x) \succ\!\!- \neg fast(x),\\ ferrari(x) \succ\!\!- sport(x),\\ ferrari(x) \succ\!\!- fast(x),\\ sport(x) \succ\!\!- \neg tourism(x)\}.$$

The following arguments can be derived:

$$arg1 = \langle \{sport(p) \succ\!\!- \neg tourism(p),\ ferrari(p) \succ\!\!- sport(p)\}, \{\}, \neg ferrari(p) \rangle,$$

$$arg2 = \langle \{sport(p) \succ\!\!- \neg tourism(p),\ ferrari(p) \succ\!\!- sport(p)\}, \{\}, fiat(p) \rangle,$$

$$arg3 = \langle \{sport(p) \succ\!\!- \neg tourism(p),\ ferrari(p) \succ\!\!- sport(p),\ fiat(p) \succ\!\!- \neg fast(p)\}, \{\}, \neg fast(p) \rangle,$$

$$arg4 = \langle \{ferrari(p) \succ\!\!- fast(p)\}, \{ferrari(p)\}, fast(p) \rangle,$$

Since here $fiat(p)$ is explicatively preferred but not $ferrari(p)$ (on basis of $tourism(p)$), $arg4$ is refuted at its supposition by the contrapositive argument $arg1$. Hence, $arg1$ becomes justified, and in two more steps ($arg2$ and $arg3$), $\neg fast(p)$ is obtained, which turns out to be defended.

7 Properties of SAS

The following are properties that Pollock [12] proved for his system, and we will show that they are also true for SAS. In fact, both systems are similar in so far as we also take into account suppositional reasoning for the construction of arguments. Nevertheless, we made a different approach to defeat, getting very different behaviors (see the discussion in Section 8). Our results will show that the approach we made is conformed to such properties. The first and second properties we will prove, to wit, consistency and deductive closure of the set of defensible sentences, have a general interest. The deduction theorem, as we will formulate it, arises because of the validity of conditionalization, and its importance lies in the fact that non-suppositional systems cannot often justify a material conditional without justifying its consequent. Reasoning by cases is also demonstrated on basis of the capability to introduce suppositions. Another obvious but not less important property is non-monotonicity, which is shown in almost all the examples in this paper.

Theorem 7.1 *(Consistency) The set of all the defensible sentences in any system* $\langle \mathcal{K}, \Delta \rangle$ *is consistent.*

PROOF. Let S be the set of all the defensible sentences in $\langle \mathcal{K}, \Delta \rangle$. If $\sigma \in S$, then σ is supported by a justified founded argument arg. Suppose by the absurd that there exists $\sigma' \in S$ such that σ' is inconsistent with σ, but it is supported by a justified founded argument arg'. Then there exists an m such that for all $n > m$, arg and arg' are level-n S-arguments. This implies that there are no level-$(n-1)$ I-arguments attacking arg or arg' (so neither arg nor arg' are level-$(n-1)$ I-arguments, since they attack each other). Then arg and arg' are level-n I-arguments. But being both mutual attackers, this implies that neither arg nor arg' are level-$(n+1)$ S-arguments. Contradiction. Then σ' is not supported by a justified founded argument, or σ' and σ are not inconsistent together; both alternatives imply that the set of defensible sentences of the system is consistent. ∎

Theorem 7.2 *(Deductive closure) For any set* $S \subset \mathcal{L}$ *of defensible sentences in a system* $\langle \mathcal{K}, \Delta \rangle$*, if* $S \vdash \tau$ *then* τ *is defensible in* $\langle \mathcal{K}, \Delta \rangle$.

PROOF. Let $S = \{\sigma_1, \ldots, \sigma_n\}$ be any set of defensible sentences in $\langle \mathcal{K}, \Delta \rangle$, such that $S \vdash \tau$. By hypothesis we have that there exist some justified founded arguments $\langle Def_1, \{\}, \sigma_1 \rangle, \ldots, \langle Def_n, \{\}, \sigma_n \rangle$, and by **Deduction** $\langle Def_1 \cup \ldots \cup Def_n, \{\}, \tau \rangle \in [[\mathcal{K}]]$. Let's call this argument arg. Suppose by the absurd that arg is not justified in $\langle \mathcal{K}, \Delta \rangle$. That is, there is not m such that for all $n > m$, arg is a level-n S-argument. This implies that for any m there will ever be some $k > m$ such that there is a level-k I-argument arg' attacking arg. This leads to a dilemma: (a) the conclusion of arg' is inconsistent with τ; or (b) the conclusion of arg' is inconsistent with the conclusion of some proper subargument of arg. If case (a) is true, then the conclusion of arg' is inconsistent with S, which implies that some sentence in S is not defensible, and this is contradictory with the hypothesis. If case (b) is true then, since arg is an immediate conclusion of $\{\langle Def_1, \{\}, \sigma_1 \rangle, \ldots, \langle Def_n, \{\}, \sigma_n \rangle\}$, the conclusion of arg' is inconsistent with some subargument of $\langle Def_1, \{\}, \sigma_1 \rangle$ or... or $\langle Def_n, \{\}, \sigma_n \rangle$. This implies that some of those arguments are not justified and, again, some of the sentences $\sigma_1, \ldots, \sigma_n$ in S are not defensible, which contradicts the hypothesis. Hence, arg is justified in $\langle \mathcal{K}, \Delta \rangle$ and τ is defensible. ∎

Theorem 7.3 *(Deduction) If* $\langle Def, Sup \cup \{\tau\}, \sigma\rangle$ *is justified in a system* $\langle \mathcal{K}, \Delta\rangle$*, then* $\langle Def, Sup, (\tau \rightarrow \sigma)\rangle$ *is justified in* $\langle \mathcal{K}, \Delta\rangle$*.*

Proof. Let $\langle Def, Sup \cup \{\tau\}, \sigma\rangle$ be justified in the system $\langle \mathcal{K}, \Delta\rangle$. By **Cond** we have $\langle Def, Sup \cup \{\tau\}, \sigma\rangle \Vdash \langle Def, Sup, \tau \rightarrow \sigma\rangle$. By hypothesis, there is an m such that for all $n > m$, there are no level-n I-arguments attacking a subargument of $\langle Def, Sup \cup \{\tau\}, \sigma\rangle$, and since $\langle Def, Sup, (\tau \rightarrow \sigma)\rangle$ has the same defeasible support, no one of its subarguments is attacked by an I-argument. Thus, $\langle Def, Sup, (\tau \rightarrow \sigma)\rangle$ is justified. ∎

Theorem 7.4 *(Reasoning by cases) If* $(\tau \vee \upsilon)$ *is defensible and some arguments* $\langle Def_1 \cup \{\tau {>\!\!-} \sigma\}, \{\tau\}, \sigma\rangle$ *and* $\langle Def_2 \cup \{\upsilon {>\!\!-} \sigma\}, \{\upsilon\}, \sigma\rangle$ *are justified in a system* $\langle \mathcal{K}, \Delta\rangle$*, then* σ *is defensible in* $\langle \mathcal{K}, \Delta\rangle$*.*

Proof. Let $\langle Def_1 \cup \{\tau {>\!\!-} \sigma\}, \{\tau\}, \sigma\rangle$ and $\langle Def_2 \cup \{\upsilon {>\!\!-} \sigma\}, \{\upsilon\}, \sigma\rangle$ be justified in a system $\langle \mathcal{K}, \Delta\rangle$. Then, by the deduction theorem, $\langle Def_1 \cup \{\tau {>\!\!-} \sigma\}, \{\}, \tau \rightarrow \sigma\rangle$ and $\langle Def_2 \cup \{\upsilon {>\!\!-} \sigma\}, \{\}, \upsilon \rightarrow \sigma\rangle$ are also justified, hence $\tau \rightarrow \sigma$ and $\upsilon \rightarrow \sigma$ are defensible. Let now $(\tau \vee \upsilon)$ be defensible in $\langle \mathcal{K}, \Delta\rangle$. Then, by deductive closure σ is also defensible in $\langle \mathcal{K}, \Delta\rangle$. ∎

A final remark that may be important for some readers of this journal: *SAS* is often not complete for defensible sentences, and it is desirable that this be so. As in most of the formalisms of its sort, we intended to capture in *SAS* the idea that sometimes it is intuitive to remain sceptic about both one belief and its negation. A canonical example of this is the "republican-quaker diamond", where there are *prima facie* reasons to believe that quakers are pacifist and republicans are non-pacifist, and some individual n is known to be both republican and quaker; so, neither belief about n's pacifism nor her/his non-pacifism is defensible.

8 Comparison with related work

In this section we discuss similarities and differences between our system and related work. In the first place we discuss the two main sources of inspiration for our approach, to wit, the systems by Pollock (we shall refer to his 1990 version) and Simari & Loui [25]. Later we shall discuss the respective approaches by Delgrande et al. [2], Geffner et al. [4] and Shu [24] to the problem of reasoning by cases with default reasons.

Pollock

We take from Pollock's *OSCAR* the structure for suppositional arguments, and the notion of warranted sentences (through a reformulation by Simari-Loui of the idea of levels of arguments) for our definition of defensible sentences[14]. The logical properties that Pollock proves for his system are also true for *SAS* (see Section 7).

Among the differences, Pollock usually takes *prima facies* reasons as having a strength depending on probabilities (cf. the notion of "statistical syllogism" in Pollock [12], p. 80), while we are not interested in how evidence could justify our defeasible rules. We assume (following Simar-Louis, Poole, Reciter, and most of the people in

[14] In Pollock [16], the author gives up his notion of warrant based on levels, studying justification through a more abstract idea of defeat. The levels-approach results appropriate for our more concrete definition.

the non-monotonic reasoning community) that defeasible rules are all equally strong, and their acceptance quite depends on the agent's beliefs. We think that this is more appropriate for representing human beings' defeasible reasoning, since human beings seldom use probabilities for common sense reasoning.

Nevertheless, we consider that the most important difference lies in the treatment of the key benchmark problems we analyzed. The reason is our introduction of case arguments and its interaction, while in *OSCAR* there are not considerations to the relevance of suppositional arguments.

Simari and Loui

From Simari & Loui's system we adopted the incorporation of the defeasible support (*i.e.*, the set of defeasible rules used in a defeasible proof) into the argument structure, which is a key for the comparison among arguments. As their system, ours takes specificity as a preference criterion, and defeat is based on that preference. On the other hand, Simari & Loui's system is not suppositional and arguments support literals as their conclusion, but not arbitrary formulae. This is because the motivation involves a computational implementation of the formalism, while we just focus on logical and philosophical aspects of defeasible reasoning beyond computability. Simari & Loui's system is not designed for dealing with disjunctive information (which would lead to too complex and maybe intractable computations), so its behavior in presence of such an information is not intuitive.

Delgrande, Schaub and Jackson

The approach by Delgrande, Schaub and Jackson [2] has a similar behavior to ours (Poole's *Theorist* [18]), a system demonstrated equivalent to this we are reviewing, leads also to similar behaviors.) The approach is an alternative to Reiter's default logic [22], called *prerequisite-free default logic* (PfDL). It consists in a translation of every default $\frac{\alpha:\beta}{\gamma}$ into $\frac{:(\alpha\supset\gamma)\wedge\beta}{(\alpha\supset\gamma)}$. This translation obviates the need to prove the antecedent for the application of a default, so the default can be used as a material conditional if that conditional satisfies the usual consistency requirements. PfDL gets reasoning by cases and contrapositive reasoning. Using semi-normal defaults (*i.e.*, defaults where the formula by the right side of ':' entails logically the conclusion) contrapositive reasoning can be blocked. For example, the default theory (where S stands for 'student', A for 'adult', and E for 'employed'):

$$\left(\left\{\frac{:S(x)\supset A(x)}{S(x)\supset A(x)},\frac{:(A(x)\supset E(x))\wedge\neg S(x)}{(A(x)\supset E(x))},\frac{:S(x)\supset\neg E(x)}{S(x)\supset\neg E(x)}\right\},\{S(a)\}\right)$$

has one extension: $Th(\{A(a),S(a),\neg E(a)\})$ (cf. Delgrande et al., [2] *op. cit.*, p. 192). In the second default, the instantiation in a of $\neg S(x)$ blocks the derivation of $A(a)\supset E(a)$; hence, even when $\neg E(a)$ is derived by the third default, $\neg A(a)$ cannot be derived.

This situation cannot be accomplished using prerequisite-free *normal* defaults (*i.e.*, those of the form $\frac{:\alpha}{\alpha}$). In the above example, using the default $\frac{:A(x)\supset E(x)}{A(x)\supset E(x)}$ in place of the second one we get two additional extensions: $Th(\{A(a), S(a), E(a)\})$ and $Th(\{\neg A(a), S(a), \neg E(a)\})$, neither of which is intuitive (cf. *ibid.*, p. 191). A semi-normal default is required to specify that adults which are students are normally not employed. Confront this with our approach, where we do not need to specify exceptions into the defeasible rules. We get the same intuitive behavior by setting that scenario within the following (simpler) representation:

$$\langle \mathcal{K} = \{S(a)\}, \Delta = \{S(x) \mathrel{>\!\!\!-} A(x), A(x) \mathrel{>\!\!\!-} E(x), S(x) \mathrel{>\!\!\!-} \neg E(x)\}\rangle.$$

Note that $\neg E(a)$ is defensible because the argument

$$\langle \{S(a) \mathrel{>\!\!\!-} \neg E(a)\}, \{\}, \neg E(a)\rangle$$

is more specific than

$$\langle \{S(a) \mathrel{>\!\!\!-} A(a), A(a) \mathrel{>\!\!\!-} E(a)\}, \{\}, E(a)\rangle.$$

Moreover, $\neg A(a)$ is not defensible because the contrapositive argument

$$\langle \{S(a) \mathrel{>\!\!\!-} \neg E(a), A(a) \mathrel{>\!\!\!-} E(a)\}, \{\}, \neg A(a)\rangle$$

is defeated by the more specific argument

$$\langle \{S(a) \mathrel{>\!\!\!-} A(a)\}, \{\}, A(a)\rangle.$$

There exist obvious differences between default, extension-based approaches and argument-based systems. Beyond that, we argue that our approach offers a simpler mode of representation with regards to the pursued intuitions, since we do not have to use constraints to state non-applicability of rules. Our system simply yields arguments, and the interaction mechanisms determine what is defensible. On the other hand, it is not possible to express suppositional reasoning in Delgrande *et al.*'s [2] approach (neither in Poole's). A remaining difference is that no preference among disjuncts is considered there.

Geffner, Llopis and Méndez

Geffner, Llopis and Méndez [4] give the approach to disjunctive information in default reasoning to which our system is philosophically more related. They propose a semantics and a proof-theory for constrained default theories with disjunctions. A priority among defaults is established in the theory, which makes the first difference with our system where defeasible rules are not compared, but arguments are. The authors define structures $\langle I, \nu\rangle$, called 'vivid interpretations', where I is an interpretation of the default theory and ν is a selection function that selects an atom from each disjunction of the theory that has to be satisfied by I. The approach states a preference criterion over such structures, preferring those vivid interpretations which are minimal models of the theory (the facts and the conclusions of applicable defaults) together with the possible facts selected by ν. The proof-theory is implemented through a set of inference rules that allow reasoning by cases and contrapositive reasoning, and it is proved to be sound and complete in relation to the above semantics.

We use a selection function ν similar to that in the formalism by Geffner *et al.* [4], but we use it to state which suppositions are plausible, *i.e.*, make an argument contextually relevant. The authors emphasize a semantical description of the theory and how that can be captured by a sound and complete proof-procedure. On the other hand, we are mainly interested in the problem of argument interaction, as it is usual in the argument systems line of research. We agree with Geffner *et al.* [4], moreover, at the intuition that a case may be refuted by the evidence.

Shu

Hua Shu [24] proposes a system of distributed default reasoning, that is, a system based on default logic which satisfies the property of *distribution*: $C(T \cup X) \cap C(T \cup Y) \subseteq C(T \cup (X \vee Y))$ (where $C(\ )$ is a consequence operation, $T, X, Y \subset \mathcal{L}$, and $X \vee Y = \{(x \vee y) : x \in X \text{ and } y \in Y\}$). This property implies that reasoning by cases is valid. The idea is to interpret a disjunction $X \vee Y$ as a collection of *cases* called "information frame": $\{\{X\}, \{Y\}, \{X, Y\}\}$. Each case is a "coherent set" (no contradiction is contained) that gives a partial description of the information frame. Each description is extended by applying defaults. The intended conclusion of a reasoning by cases with defaults is contained in the intersection of all the resulting sets. Shu's approach satisfies semi-monotonicity and cumulativity with normal defaults, and also commitment to assumptions with non-normal defaults. These properties are not within the scope of our system.

As distinguished from our results, Shu's approach, as well as Reiter's (standard) default logics, cannot validate contrapositive reasoning. Maybe the author disagrees about the intuition of that form of reasoning, which is not discussed in his work.

9 Conclusion

Dealing with disjunctive information has been difficult since the beginning of the study of defeasible argumentation in Artificial Intelligence. One aspect of this difficulty is computational, but other is logical and philosophical. We have argued here that the way of solving the last is through suppositional reasoning. First we introduced a suppositional argument system, SAS, which can express arguments with disjunctive information, such as dilemmas or reasoning by cases. The main aspects of SAS are essentially similar to Pollock's $OSCAR$.

Using SAS we have studied how disjunctive information can improve the rationality of a debate, determining whether that information gives relevance to suppositional arguments or not. The solution was given by considering, from a semantical view, the minimal models of the context, which indicate which suppositions are plausible as alternatives, and determining which arguments are contextually relevant. From a syntactical view, the solution was to translate the context formulae in their disjunctive normal form, and establishing a preference among all the possible disjuncts; this allowed to define *case arguments*. As a result, an equivalence between contextually relevant arguments and case arguments was obtained (proposition 5.10).

Also a preference relation among plausible alternative was introduced through a notion of explicative power. Only explicatively preferred arguments can compete for justification in SAS, and that competence is established on the criterion of specificity. That was possible because specificity can be adapted to SAS in a natural way, being suppositional reasoning a proper part of the idea. We also emphasize on the emergent capability of the system for arguing contrapositively. This is usually problematic in defeasible argumentation, but we have captured behaviors that previously were seldom achieved in defeasible reasoning.

Acknowledgments

I am grateful to Fernando Tohmé for his very interest in my work and his valuable help and critics. Lots of thanks to him. Héctor Geffner and Henry Prakken are acknowledged for comments and suggestions on earlier ideas for this work (through discussions held while they were teaching in Bahía Blanca). I also wish to thank Jorge Roetti, Guillermo Simari and Juan Manuel Torres for their support. An anonymous referee is acknowledged for constructive criticisms that have notoriously improved this paper. This research was partially supported by Secretaría de Ciencia y Técnica, Universidad Nacional del Sur, Bahía Blanca, Argentina (Expte. SGCyT 1737-98).

References

[1] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, **77**: 321-357, 1995.

[2] J. Delgrande, T. Schaub and W. K. Jackson. Alternative approaches to default logic, *Artificial Intelligence*, **70**: 167-237, 1994.

[3] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of Np-completeness.* W. H. Freeman & Co., 1979.

[4] H. Geffner, J. Llopis and G. Méndez. Disjunctive facts in causal default reasoning. *Proc. of IBERAMIA-94*, Lisboa, 1994.

[5] R. Kowalski and F. Toni. Abstract argumentation. *Artificial Intelligence and Law*, **4** (3-4): 275-396, 1996.

[6] H. Kyburg. *The Logical Foundations of Statistical Inference*. Reidel, Dordrecht, 1974.

[7] F. Lin and Y. Shoham. Argument systems: a uniform basis for nonmonotonic reasoning. *Proc. of the First International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, 245-255, 1989.

[8] R. Loui. Defeat among arguments: a system of defeasible inference. *Computational Intelligence* **2**: 100-106, 1987.

[9] J. Minker. On indefinite data bases and the closed world assumption. *Proc. of the 6th Conference on Automated Deduction*, 292-308, 1982.

[10] S. Parsons, C. Sierra and N. Jennings. Agents that reason and negotiate by arguing. Manuscript, 1998 (to appear in *Journal of Logic and Computation*).

[11] J. Pollock. Defeasible reasoning. *Cognitive Science* **11**: 481-518, 1987.

[12] J. Pollock. *Nomic probability and the foundations of induction*. Oxford University Press, New York, 1990.

[13] J. Pollock. A theory of defeasible reasoning. *International Journal Of Intelligent Systems* **6**: 33-54, 1991 (a).

[14] J. Pollock. Self-defeating arguments. *Minds And Machines* **1**: 367-392, 1991 (b).

[15] J. Pollock. How to reason defeasibly. *Artificial Intelligence* **57**: 1-42, 1992.

[16] J. Pollock. Justification and defeat. *Artificial Intelligence* **67**: 377-407, 1994.

[17] D. Poole. On the comparison of theories: preferring the most specific explanation. *Proc. of the Ninth IJCAI*, Los Altos, 144-147, 1985.

[18] D. Poole. A logical framework for default reasoning. *Artificial Intelligence* **36** (1): 27-48, 1988.

[19] D. Poole. The effect of knowledge on belief: conditioning, specificity and the lottery paradox in default reasoning. *Artificial Intelligence* **49**: 281-307, 1991.

[20] H. Prakken. *Logical tools for modeling legal arguments*. Doctoral dissertation, Vrije Universiteit, Amsterdam, 1993.

[21] H. Prakken and G. Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law* **4**(3-4): 331-368, 1996.

[22] R. Reiter. A logic for default reasoning. *Artificial Intelligence* **13**: 81-132, 1980.

[23] N. Rescher. *Dialectics: a controversy-oriented approach to the theory of knowledge*. State University of New York Press, Albany, 1977.

[24] H. Shu. Distributed Default Reasoning. *Linköping Studies in Science and Technology*, Dissertation No. 41, 1996.

[25] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning. *Artificial Intelligence* **53**: 125-157, 1992.

[26] G. Simari and C. Delrieux. Combinando plausibilidad y razonamiento revisable. *Proc. of the 23 Jornadas Argentinas de Informática e Investigación Operativa*, SADIO, Buenos Aires, 99-110, 1994.

[27] D. Sperber and D. Wilson. *Relevance. Communication and cognition*. Basil Blackwell, Oxford, 1986.

[28] F. Tohmé. Negotiation and defeasible reasons for choice. In *Proceedings of the AAAI Stanford Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, 95-102, 1997.

[29] S. Toulmin. *The uses of argument*. Cambridge University Press, London, New York, 1958.

[30] B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. Presented at the *Computational Dialectics Workshop*, June 3-7 1996, Bonn. Published as report SKBS/B3.A/96-01. Full text available at the World-Wide Web at http://nathan.gmd.de/projects/zeno/fapr/programme.html.

[31] G. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence* **90**: 225-279, 1997.

Received 27 October 2000. Revised 10 January 2002

A General Framework for Pattern-Driven Modal Tableaux

LUIS FARIÑAS DEL CERRO and OLIVIER GASQUET
IRIT - Universit Paul Sabatier, 118 route de Narbonne, Toulouse Cedex 04, France. E-mail: {farinas, gasquet}@irit.fr.

Introduction

Following Kripke, tableau rules should be designed in order to propagate formulas in a tree *so that* it *simulates* the properties of a Kripke model, which is not simply a tree, but has additional features. E.g., a tree with the S4 rule "if $\Box A$ is present in some node then transport it into all successors" should behave *as if* the tree were transitive.

In the standard approach the propagation of formulas is only top-down; moreover, using only trees as underlying structures is too restrictive: it is difficult to design tableaux methods for some logics like those based on a density axiom ($\Diamond p \rightarrow \Diamond\Diamond p$) or on a confluence axiom ($\Diamond\Box p \rightarrow \Box\Diamond p$); this seems to indicate that trees are not a good basis for such properties.

We present here a new basis that is characterized by two ideas:

1. the propagation of formulas need not to be top-down,
2. the underlying structure need not to be a tree.

In [21, 15, 6, 20], the first idea has been investigated. Their work is intimately mixed with another feature: the so called single-step rules. Such rules allow to propagate formulas only from one node to one of its successors or predecessor in the tree. We generalize this notion to "pattern-driven rules": rules apply if some elementary pattern in the mathematical structure has been matched.

Besides, we will show that while trees are a good basis for tableaux for many usual modal logics (tableaux for systems K(D)(T)(4) are based on trees), they fail to support in a comprehensive way confluent or dense relations for example, and bimodal confluent systems too, i.e. systems which semantics is such that there may be several paths from the root of the structure (the tableau under computation) to some given world. This feature requires that the reasoning about the accessibility relation should be separated from the reasoning about formulas. At least two main approaches may be first considered:

- By the use of labels [21, 15, 6, 20, 12] where each possible world is associated with a label (its path from the root, the set of all path may be considered as a spanning tree of a graph) and an equationnal theory is needed in order to specify which paths denote the same world.
- By the explicite use of the accessibility relation, as explored in [1, 2]; this turns out to be a model-construction approach. But their work focuses on "grammar logics"

(multi-modal logics with axioms of the form $[a_1]\dots[a_n]p \rightarrow [b_1]\dots[b_m]p$) and mainly study their connections with formal languages, thus excluding properties like symmetry, euclideanness, confluence; moreover, it does not seem possible to extend decidability results and decision procedures for a property like density in their framework.

Our position (partially-explicit use of the accessibility relation), while being close to that of Baldoni et al. is different at least because as long as possible, we do not completely compute the accessibility relation: for example, we do not compute its transitive closure but simulate it by propagation of formulas. We compute it only for existential properties (one may find more examples in [3]) and handle properties they cannot.

We argue that rooted directed acyclic graphs (RDAG for short), which are DAGs having a distinguished node called the root, are better suited. They allow to naturally handle some properties that do not marry easily with tree structures (like confluence, density, and also, permutation in the multi-modal case), while other properties (like transitivity, symmetry, ...) can still be handled by the propagation of formulas. Moreover, we believe that discovering decision procedures for density and confluence was possible at least because of the very intuitive aspects of our framework.

This leads us to identify two kinds of tableaux rules:

1. propagation rules
2. structural rules.

The former are formulated as "if in some node of such pattern there is such formula, then propagate such formula (the same or another one)", while the latter are "if there is such pattern then add some new node(s) and edge(s)".

They respectively correspond to two different families of axioms (relational properties):

- Propagation rules correspond to axioms T, 4, B and 5 (properties of reflexivity, transitivity, symmetry and euclideanness, respectively);
- Structural rules correspond to axioms D, De and C (respectively properties of seriality, density and confluence). Also, in section 3, we will investigate some cases of bimodal logics with a permutation axiom.

What do we gain by this new perspective? It holds in a few words: simplicity, naturality and modularity, both in the definition of a tableau calculus for a given system and in its correctness proof. First, for the classical connectives as well as for $\Diamond$, rules and correctness proof are common to all systems. There only remains the case of structural rules, and of propagation rules for $\Box$ that are treated in a really simple, natural and modular way.

Generally speaking, a tableau is a structure (usually a tree, in our case it will be an RDAG) whose nodes are labelled by sets of formulas. The completeness proof of a tableau method is in two main steps: the construction of a model from this structure, and the verification that this model satisfies the formulas of the nodes (the so-called Fundamental Lemma).

The first step is usually done by adding new arrows to the structure, according to the particular property of the accessibility relation of the logic under concern. For example, for the system S4 the accessibility relation is reflexive and transitive. Hence, given a tree (the underlying structure for S4), we must close it under reflexivity and transitivity in order to make an S4-model of it. In other terms, we must characterize when two nodes are related in the resulting closure. Then we can say that for a given node x another node y will be accessible from x if there is an $n \geq 0$ and $x_0, \dots, x_i, x_{i+1}, \dots, x_n$ such that x_0 is x, x_n is y and x_{i+1} is a child of x_i in the original tree. From this characterization of the closure of the initial tree under the additional properties of the logic under concern, we can "read off" the rules to be designed. Thus the rules will ensure the correct propagation of formulas, the proof being very easy[1]. This gives naturality and simplicity. In addition, the rules that we have obtained fit closely to the intuition.
Modularity is achieved since we obtain tableaux calculi whose completeness proofs are neatly separated into three components:

1. the Relational Closure Lemma (lemma 1.5) where the properties of the closure of an RDAG under some relational properties are expressed in terms of the initial RDAG,

2. the Structural Lemma (lemma 1.7) where we check that the closure of an RDAG under some relational properties preserves some of its initial features (e.g. the transitive closure of a confluent RDAG yields a confluent relation, but not necessarily an RDAG),

3. the Box Lemma (lemma 1.11) where we check that whenever x and y are related in the closure, and $\Box A \in x$ then the set of associated rules ensures that A was transported into y.

The rest of the completeness proof is completely factorized. We also present a soundness result for the tableaux calculi we define.

Then we address the decidability issues by introducing the notion of a kernel. For each logic we are intested in, we identify those finite structures that can simulate infinite ones: e.g. it is well-known that kernels for KD4 are finite trees. as a result we obtain for KD4 plus confluence, and KD4 plus density, new semantical characterizations, and as a consequence, we obtain a PSPACE upper bound for the complexity of satisfiability for these logics. We will go back to this in section 2.

For backgrounds about tableaux for modal logic, the reader may look at [8], [12], [13], [15].

We will first concentrate on some basic monomodal and temporal logics, and investigate, in section 3, some cases of bimodal systems mixing temporal and modal concepts.

[1] The correctness proof mainly consists of results of relational calculus.

At the end, some remarks are given on Lotrec which is a generic theorem prover based on the notions presented in what follows and which allows to implement in a friendly way many of the modal and temporal logics used in applications.

1 Complete tableaux for monomodal logics

1.1 Modal logics and relational properties

A modal logic can be specified syntactically or semantically. We recall what the links between these presentations are.

The modal logics we consider are all obtained by extending the basic modal logic K by one or several of the well-known axioms T, B, 4, 5, D, De (axiom of density: $\Diamond p \to \Diamond\Diamond p$) and C (axiom of confluence: $\Diamond\Box p \to \Box\Diamond p$). Thus KDC4 denotes the modal logic obtained by adding the axioms D, C and 4 to the basic system K.

With each of these axioms can be associated a relational property of the accessibility relation of the Kripke models:

Axiom	Property	Notation
$\mathrm{T} = \Box p \to p$	reflexivity	*Ref*
$4 = \Box p \to \Box\Box p$	transitivity	*Tr*
$\mathrm{B} = \Diamond\Box p \to p$	symmetry	*Sym*
$5 = \Diamond\Box p \to \Box p$	euclideanness	*Eucl*

Group 1: Properties handled by propagation rules

Axiom	Property	Notation
$\mathrm{D} = \Box p \to \Diamond p$	seriality	*Ser*
$\mathrm{De} = \Diamond p \to \Diamond\Diamond p$	density	*Dens*
$\mathrm{C} = \Diamond\Box p \to \Box\Diamond p$	confluence	*Conf*

Group 2: Properties handled by structural rules

As a consequence of Sahlqvist's theorem [24], a system based on K plus any combination of these axioms is characterized by the Kripke models whose accessibility relation satisfies the corresponding properties. Thus, KD4 is characterized by Kripke models where the accessibility relation is both serial and transitive; for KT5 reflexivity and euclideanness are required (and, as a consequence, transitivity, seriality and symmetry).

From now on we will indistinctly denote a modal system by $\mathrm{KA}_1 \ldots \mathrm{A}_n$, where each A_i belongs to group 1 or 2, or by a set ρ of its accessibility relation properties; we will write $\rho = \rho_1 \cup \rho_2$ where ρ_1 is a maximal subset of properties of group 1 (maximal here means "including all those of group 1 which are a consequence of it": thus, symmetry and transitivity imply euclideanness: any set ρ_1 that contain *Sym* and *Tr* must also contains *Eucl*), and ρ_2 is a subset of properties of group 2. E.g. KCD4 will be denoted by $\{Ser, Tr, Conf\}$, KDeB4 by $\{Sym, Tr, Eucl, Dens\}$ (since euclideanness is a consequence of transitivity and symmetry).

Definition 1.1 Given a set ρ of relational properties among group 1 and 2, a ρ-model is a Kripke model whose accessibility relation satisfies ρ. A formula is ρ-satisfiable iff it is satisfiable in a ρ-model. It is ρ-valid iff it is valid in the class of all ρ-models, this will be denoted $\models_\rho A$. Thus A is a theorem of a system denoted by a set ρ of properties iff it is ρ-valid.

1.2 *Preliminaries and notations*

The tableau calculi we are going to present are based on RDAG (rooted directed acyclic graphs) having additional properties; let ρ be the set of these additional properties, we define:

Definition 1.2 A labelled ρ-RDAG is a triple $(\mathcal{N}, \Sigma, \Phi)$ where:

- $(\mathcal{N}, \Sigma)$ is a directed acyclic graph (DAG), i.e. a directed graph that contains no cycle, with a distinguished node called the *root* that can access every other node in the transitive closure of Σ,
- $(\mathcal{N}, \Sigma)$ satisfies all the properties of ρ,
- Φ is a function that associates additional information with each of the nodes: if x is a node, $\Phi(x)$ is a set of formulas.

By abuse of notation and for the sake of notational economy, we will make no distinction between the nodes and their associated sets of formulas; thus we will write $A \in x$ instead of $A \in \Phi(x)$. Also by abuse of notation, we will sometimes denote a ρ-RDAG $(\mathcal{N}, \Sigma)$ by the binary relation Σ. Thus we will make no distinction between labelled structures and structures.

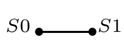
This notion also extend to graphs:

Definition 1.3 An RGRAPH is a graph that has a root, and a ρ-RGRAPH is a RGRAPH that satisfies all properties of ρ.

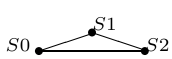
As usual, $\Sigma(x)$ will denote the set of nodes accessible from x by Σ: $\Sigma(x) = \{y \in \mathcal{N} : (x, y) \in \Sigma\}$. Also, Σ^n will denote the pairs (x, y) such that there is a path of length n between x and y. The diagonal relation: $\{(x, x) : x \in \mathcal{N}\}$ will be denoted by I and also by Σ^0.

For the sake of clarity, we will use a diagrammatic representation for RDAG. The figure below gives the intended meaning of those diagrammatic representations in which the edges are implicitely left-to-right directed[2]:

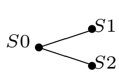
denotes a node S



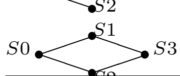
denotes $(S0, S1) \in \Sigma$



denotes $(S0, S1), (S0, S2), (S1, S2) \in \Sigma$



denotes $(S0, S1), (S0, S2) \in \Sigma$



denotes $(S0, S1), (S0, S2), (S1, S3), (S2, S3) \in \Sigma$

[2] Note that RDAG are of course antisymmetrical.

The last two diagrams do not involve any order between $S1$ and $S2$, e.g.

$$\begin{tikzcd} & S1 \\ S0 \arrow[ur, no head] \arrow[dr, no head] & \\ & S2 \end{tikzcd} \quad \text{can be represented as well by} \quad \begin{tikzcd} & S2 \\ S0 \arrow[ur, no head] \arrow[dr, no head] & \\ & S1 \end{tikzcd}$$

1.3 Closure of RGRAPH

We define the following closure operation on RGRAPH:

Definition 1.4 Let Σ be an RGRAPH over a set $\mathcal{N}$ and ρ a set of relational properties of group 1; the ρ-closure of Σ (denoted by Σ^{ρ}) is the least RGRAPH that contains Σ and which satisfies every property of ρ.

This ρ-closure always exists if the properties are among $\{Ref, Tr, Sym, Eucl\}$. A very important point is that for properties of group 1, the closure can be expressed in terms of the initial RGRAPH. E.g. the transitive closure of an RGRAPH Σ is defined by: $(x, y) \in \Sigma^{Tr}$ iff $\exists n \geq 1$ such that $(x, y) \in \Sigma^{n}$ (c.f. def A.1). Note that we do not consider here properties of group 2: it makes no sense to talk about closure under a property of group 2. This is the reason why they are handled in a different way: no propagation rule can simulate them.

Lemma 1.5 (Relational Closure Lemma)
Let Σ be an RDAG over a set $\mathcal{N}$ of nodes:

- $(x, y) \in \Sigma^{Ref}$ iff $(x, y) \in \Sigma$ or $x = y$.
- $(x, y) \in \Sigma^{Sym}$ iff $(x, y) \in \Sigma$ or $(y, x) \in \Sigma$.
- $(x, y) \in \Sigma^{Tr}$ iff $\exists n \geq 1$ such that $(x, y) \in \Sigma^{n}$.
- $(x, y) \in \Sigma^{Eucl}$ iff $(x, y) \in \Sigma$ or $\exists u \in \mathcal{N}\ \exists n \geq 1\ \exists m \geq 1$ such that $(u, x) \in \Sigma^{n}$ and $(u, y) \in \Sigma^{m}$.
- $(x, y) \in \Sigma^{Ref,Sym}$ iff $(x, y) \in \Sigma$ or $x = y$ or $(y, x) \in \Sigma$.
- $(x, y) \in \Sigma^{Ref,Tr}$ iff $\exists n \geq 0$ such that $(x, y) \in \Sigma^{n}$.
- $(x, y) \in \Sigma^{Ref,Eucl}$ iff $\exists n \geq 0\ \exists x_0 = x, \ldots x_i, x_{i+1}, \ldots, x_n = y : (x_i, x_{i+1}) \in \Sigma$ or $(x_{i+1}, x_i) \in \Sigma$.
- $(x, y) \in \Sigma^{Sym,Tr}$ iff $\exists n \geq 1\ \exists x_0 = x, \ldots x_i, x_{i+1}, \ldots, x_n : (x_i, x_{i+1}) \in \Sigma$ or $(x_{i+1}, x_i) \in \Sigma$.
- $(x, y) \in \Sigma^{Tr,Eucl}$ iff $\exists u \in \mathcal{N}\ \exists n \geq 0\ \exists m \geq 1$ such that $(u, x) \in \Sigma^{n}$ and $(u, y) \in \Sigma^{m}$.

PROOF. Straightforward consequence of the lemmas B.1 and B.3 of the appendix. ∎

Lemma 1.6 The remaining cases are reducible to those of the previous lemma:

- $\Sigma^{Sym,Eucl} = \Sigma^{Sym,Tr,Eucl} = \Sigma^{Sym,Tr}$
- $\Sigma^{Ref,Sym,Tr} = \Sigma^{Ref,Tr,Eucl} = \Sigma^{Ref,Sym,Eucl} = \Sigma^{Ref,Sym,Tr,Eucl} = \Sigma^{Ref,Eucl}$

PROOF. Straightforward. ∎

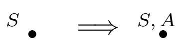
The above lemma will be a powerful tool for proving completeness: it will allow to define a model for a formula from an open tableau. But this is not the whole story. As we previously said, some properties are handled structurally; roughly speaking seriality, density and confluence are treated by the underlying "kind" of RDAG of the tableaux. When in the completeness proof we must close the RDAG under one or several properties of group 1 (note that after this closure operation, the initial RDAG is no longer an RDAG but an RGRAPH), we must also check that its structural properties are preserved after this closure (i.e. that it is still of the same "kind"). E.g. we must prove that the transitive closure of a confluent RDAG is still confluent. This is the aim of the lemma below:

Lemma 1.7 (Structural Lemma) Let ρ_2 be a subset of group 2, ρ_1 a subset of group 1 and let Σ be a ρ_2-RGRAPH over a set $\mathcal{N}$ of nodes. Then Σ^{ρ_1} is also a ρ_2-RGRAPH and hence is a $(\rho_1 \cup \rho_2)$-RGRAPH.

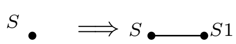
PROOF. See appendix B. ∎

1.4 Rewriting RDAG

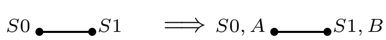
Usually, tableaux calculi consist in rewriting a structure by using some appropriate set of rewriting rules (or simply rules). But before presenting our rules, we want to propose some visual conventions; as usual, S, A denotes $S \cup \{A\}$:



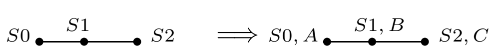
rewrite the node S into the node $S \cup \{A\}$, i.e. add the formula A to the node S,



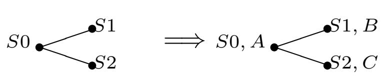
add the new node $S1$ to the successors of the node S,



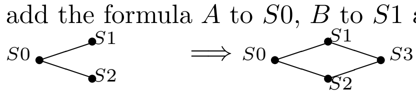
add the formula A to the node $S0$ and B to $S1$,



add the formula A to $S0$, B to $S1$ and C to $S2$,



add the formula A to $S0$, B to $S1$ and C to $S2$.



add the new node $S3$ as a common successor of the node $S1$ and $S2$,

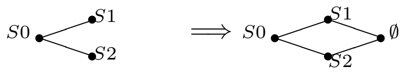
$S0 \bullet\!-\!\!-\!\!-\!\bullet S1 \quad\Longrightarrow\quad S0 \bullet\!-\!\!-\!\overset{S2}{\bullet}\!-\!\!-\!\bullet S1$
add the new node $S2$ between $S0$ and $S1$.

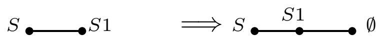
This presentation allows to implicitly take into account constraints on the applicability of rules: e.g. a rule such as
$S0 \bullet\!-\!\!-\!\overset{S1,\Box A}{\bullet}\!-\!\!-\!\bullet S2 \quad\Longrightarrow\quad S0 \bullet\!-\!\!-\!\overset{S1,\Box A}{\bullet}\!-\!\!-\!\bullet S2,\Box A$ reads "add $\Box A$ to any successor of $S1$ if $S1$ has a predecessor and contains $\Box A$".

1.5 Rules

Here are the rules we need:

- Classical and $\Diamond$ rules:
  - Rule $\perp$: $\overset{A,\neg A,S}{\bullet} \Longrightarrow \overset{A,\neg A,\perp,S}{\bullet}$
  - Rule $\neg$: $\overset{\neg\neg A,S}{\bullet} \Longrightarrow \overset{\neg\neg A,A,S}{\bullet}$
  - Rule $\wedge$: $\overset{A\wedge B,S}{\bullet} \Longrightarrow \overset{A\wedge B,A,B,S}{\bullet}$
  - Rule $\vee$: $\overset{\neg(A\wedge B),S}{\bullet} \Longrightarrow \overset{\neg(A\wedge B),C,S}{\bullet}$
    where C is one of $\neg A$ and $\neg B$
  - Rule $\Diamond$: $\overset{\Diamond A,S}{\bullet} \Longrightarrow \Diamond A,S \bullet\!-\!\!-\!\!-\!\bullet A$
- Propagation rules:
  - Rule K: $\Box A,S \bullet\!-\!\!-\!\!-\!\bullet S1 \Longrightarrow \Box A,S \bullet\!-\!\!-\!\!-\!\bullet A,S1$
  - Rule T: $\overset{\Box A,S}{\bullet} \Longrightarrow \overset{\Box A,A,S}{\bullet}$
  - Rule 4: $S,\Box A \bullet\!-\!\!-\!\!-\!\bullet S1 \Longrightarrow S,\Box A \bullet\!-\!\!-\!\!-\!\bullet S1,\Box A$
  - Rule B: $S \bullet\!-\!\!-\!\!-\!\bullet S1,\Box A \Longrightarrow S,A \bullet\!-\!\!-\!\!-\!\bullet S1,\Box A$
  - Rule $5_{\rightarrow}$: $S \bullet\!\!<\!\begin{array}{l}\bullet S1,\Box A\\ \bullet S2\end{array} \Longrightarrow S \bullet\!\!<\!\begin{array}{l}\bullet S1,\Box A\\ \bullet S2,\Box A\end{array}$
  - Rule $5_{\uparrow}$: $S \bullet\!-\!\!-\!\!-\!\bullet S1,\Box A \Longrightarrow S,\Box A \bullet\!-\!\!-\!\!-\!\bullet S1,\Box A$
  - Rule $5_{\downarrow}$: $S \bullet\!-\!\!-\!\overset{S1,\Box A}{\bullet}\!-\!\!-\!\bullet S2 \Longrightarrow S \bullet\!-\!\!-\!\overset{S1,\Box A}{\bullet}\!-\!\!-\!\bullet S2,\Box A$
- Structural rules:
  - Rule D: $\overset{S}{\bullet} \Longrightarrow S \bullet\!-\!\!-\!\!-\!\bullet \emptyset$

– Rule C: 

– Rule $C*$: 

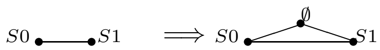
– Rule De: 

Tableau rules

Of course, it may be discussed why not to consider rule $\Diamond$ as a structural rule. This decision is quite arbitrary, we consider that structural rules are those describing an existential property of the accessibility relation, while rule $\Diamond$ correspond to the expression *in the object language* of the existence of objects. Thus, we will sometimes consider rule $\Diamond$ as a structural rule for convenience.

1.6 Tableau calculi

In order to define a tableau calculus for a system denoted by $\rho_1 \cup \rho_2$, we must associate a set of rules with it. All the tableaux calculi we are going to define contain: the classical rules and the rule $\Diamond$ plus the rule K (as these rules are common to all tableaux calculi, we will henceforth omit them) plus none or some structural and propagation rules.

A tableau calculus for a system denoted by $(\rho_1 \cup \rho_2)$ is obtained by taking (in addition to classical, $\Diamond$ and K rules) the rules corresponding to properties of $(\rho_1 \cup \rho_2)$; this correspondance is given in the figure below.

	Properties	Rules	
Group 1	*Ref*	T	Propagation
	Sym	B	Rules
	Tr	4	
	Eucl	$5_\uparrow$ $5_\downarrow$ $5_\rightarrow$	
Group 2	*Ser*	D	Structural
	Dens	De	Rules
	Conf	C C*	

We define what we call *naive tableaux*, i.e. tableaux computed with no strategy, particularly strategies ensuring termination. We will see such strategies in section 2.

Definition 1.8 A (naive) $(\rho_1 \cup \rho_2)$-tableau for a formula A is the limit of a sequence $\Upsilon_0, \ldots, \Upsilon_i, \Upsilon_{i+1}, \ldots$ where $\Upsilon_i = (\mathcal{N}_i, \Sigma_i, \Phi_i)$. It is the limit in the sense that $\mathcal{N} = \bigcup_{i\geq 0} \mathcal{N}_i, \Sigma = \bigcup_{i\geq 0} \Sigma_i, \Phi(x \in \mathcal{N}) = \bigcup_{i\geq 0, x\in\mathcal{N}_i} \Phi_i(x)$.
In addition we must have:

- Υ_0 is an RDAG consisting of only one node whose associated set of formulas is $\{A\}$,

- Υ_{i+1} is obtained from Υ_i by applying either a classical rule, or the $\Diamond$ rule, or the rule K, or a rule of $(\rho_1 \cup \rho_2)$

- and in which every applicable rule has been applied once.

Definition 1.9 A tableau is closed if some node in it contains $\perp$; it is open otherwise. A formula is $\rho_1 \cup \rho_2$-closed iff all its $(\rho_1 \cup \rho_2)$-tableaux are closed [3].

Of course, as implicitely stated above, we make the usual assumption of *fairness*: if at some iteration i, some rule is applicable then for some $j \geq i$ the rule has been applied (Note that any applicable rule remains so until it has been applied). Thus, saying that completeness and soundness rely on the fairness assumption consists in saying that *they only hold for fair algorithms*. For example, an algorithm that would apply rule (D) first (hence with a priority greater than that of other rules) would not be fair since rule (D) may be applied infinitely giving no chance to apply any other rule.

1.7 Completeness

In this subsection we prove the completeness of our tableaux calculi. We show how, from a given open $(\rho_1 \cup \rho_2)$-tableau for A we can construct a $(\rho_1 \cup \rho_2)$-model for A.

Let Υ be an open $(\rho_1 \cup \rho_2)$-tableau for A. Υ is a ρ_2-RDAG where $\Upsilon = (\mathcal{N}, \Sigma, \Phi)$ with root r, since structural rules corresponding to ρ_2 ensure that Υ satisfies ρ_2.

Now let $\mu = (W, R, \tau)$ be the Kripke model defined as follows:

Definition 1.10

- $W = \mathcal{N}$
- R is the ρ_1-closure of Σ, i.e. $R = \Sigma^{\rho_1}$
- for all $w \in W$, $w \in \tau(p)$ iff $p \in w$ (in fact iff $p \in \Phi(w)$).

By construction, μ satisfies properties of ρ_1 and, by the Structural Lemma (lemma 1.7), it also satisfies the properties of ρ_2; hence it is a $(\rho_1 \cup \rho_2)$-model. What remains is to prove that it satisfies the formula A. We first establish the following important lemma:

Lemma 1.11 (Box Lemma) Let $\Upsilon = (\mathcal{N}, \Sigma, \Phi)$ be a $(\rho_1 \cup \rho_2)$-tableau with root r. Let x, y be such that $(x, y) \in \Sigma^{\rho_1}$ and $\Box A \in x$; then $A \in y$.

Proof. There are nine cases, according to ρ_1; we only prove the lemma for some of the most complex cases (all involving euclideanness):

- $\rho_1 = \{Eucl\}$: if $(x, y) \in \Sigma^{\rho_1}$ then by the Relational Closure lemma, we have either $(x, y) \in \Sigma$ and then $A \in y$ (by rule K), or $\exists u \in \mathcal{N}\ \exists n \geq 1\ \exists m \geq 1$ such that $(u, x) \in \Sigma^n$ and $(u, y) \in \Sigma^m$. Hence we both have

[3] Due to the rule $\vee$, a formula may have several distinct tableaux.

$\exists x_0 = x, \ldots, x_i, x_{i+1}, \ldots, x_n = u\colon (x_{i+1}, x_i) \in \Sigma$; then $\Box A \in x_i$ for $0 \le i \le n$ (by rule $5_\uparrow$ n times), in particular: $\Box A \in x_{n-1}$ and $\Box A \in x_n = u$.

and $\exists y_0 = u, \ldots, y_i, y_{i+1}, \ldots, y_m = y\colon (y_i, y_{i+1}) \in \Sigma$; hence $\Box A \in y_1$ (by rule $5_\rightarrow$ since $\Box A \in x_{n-1}$) from which we get $\Box A \in y_i$ for $1 \le i \le m$ (by rule $5_\downarrow$ $m-1$ times) and since also $\Box A \in x_n = u = y_0$, we have $\Box A \in y_i$ for $0 \le i \le m$. Hence $A \in y_i$ for $1 \le i \le m$ (by rule K), in particular $A \in y$.

- $\rho_1 = \{Tr, Eucl\}$: if $(x, y) \in \Sigma^{\rho_1}$ then by the Relational Closure lemma, we have $\exists u \in \mathcal{N}\ \exists n \ge 0\ \exists m \ge 1$ such that $(u, x) \in \Sigma^n$ and $(u, y) \in \Sigma^m$. This implies that: $\exists n \ge 0\ \exists x_0 = x, \ldots, x_i, x_{i+1}, \ldots, x_n = u\colon (x_{i+1}, x_i) \in \Sigma$; then $\Box A \in x_0$ implies $\Box A \in u$ (by rule $5_\uparrow$, n times)

and $\exists m \ge 0\ \exists y_0 = u, \ldots, y_i, y_{i+1}, \ldots, y_{m+1} = y\colon (x_i, x_{i+1}) \in \Sigma$; hence $\Box A \in u$ implies $\Box A \in y_m$ (by rule 4, m times) and $A \in y$ (by rule K).

- $\rho_1 = \{Sym, Tr, Eucl\}$: if $(x, y) \in \Sigma^{\rho_1}$ then by the Relational Closure lemma, we have $\exists n \ge 1\ \exists x_0 = x, \ldots, x_i, x_{i+1}, \ldots, x_n = y\colon (x_i, x_{i+1}) \in \Sigma$ or $(x_{i+1}, x_i) \in \Sigma$; but $\Box A \in x_0$ and $\Box A \in x_i \Rightarrow \Box A \in x_{i+1}$ (by rule 4 or $5_\uparrow$, according to whether $(x_i, x_{i+1}) \in \Sigma$ or $(x_{i+1}, x_i) \in \Sigma$). Thus $\Box A \in x_i$ for $0 \le i \le n$ and hence $A \in x_i$ for $0 \le i \le n+1$ (by rule K or B). Thus $A \in y$.

∎

The following fundamental lemma brings us to the desired conclusion:

Lemma 1.12 (Fundamental Lemma) Let Υ be an open $(\rho_1 \cup \rho_2)$-tableau for A, let μ be the $(\rho_1 \cup \rho_2)$-model defined as in definition 1.10 w.r.t. Υ and let $B \in$ Subformulas(A) then: (i) if $B \in x$ then $\mu, x \models B$.

Proof. (By induction on the structure of B: W.l.o.g we can suppose that B is written with only $\neg$, $\wedge$, $\perp$ and $\Box$).

Induction initialization: let B be an atom; then (i) holds by definition of τ.

Induction step[4]:

- B cannot be $\perp$, otherwise Υ would be closed.

- Let B be $\neg\neg C$.
  $\neg\neg C \in x$
  $\Rightarrow C \in x$ (by rule $\neg$)
  $\Rightarrow \mu, x \models C$ (by IH)
  $\Rightarrow \mu, x \models \neg\neg C$.

- Let B be $(C \wedge D)$.
  $(C \wedge D) \in x$
  $\Rightarrow C \in x$ and $D \in x$ (by rule $\wedge$)

[4] In this proof, when we say "by rule R" we mean "by rule R and by the fairness assumption that rule R has been applied".

$\Rightarrow \mu, x \models C$ and $\mu, x \models D$ (by IH)
$\Rightarrow \mu, x \models (C \wedge D)$.

- Let B be $\neg(C \wedge D)$.
  $\neg(C \wedge D) \in x$
  $\Rightarrow \neg C \in x$ or $\neg D \in x$ (by rule $\vee$)
  $\Rightarrow \mu, x \models \neg C$ or $\mu, x \models \neg D$ (by IH)
  $\Rightarrow \mu, x \models \neg(C \wedge D)$.

- Let B be $\neg\Box C$
  $\neg\Box C \in x$
  $\Rightarrow$ there exists y such that $(x, y) \in \Sigma$ and $\neg C \in y$ (by rule $\Diamond$)
  $\Rightarrow$ there exists y such that $(x, y) \in R$, and $\mu, y \models \neg C$ (by IH and definition of R)
  $\Rightarrow \mu, x \models \neg\Box C$.

- Let B be $\Box C$ and suppose $(x, y) \in R = \Sigma^{\rho_1}$; then by the Box Lemma (1.11), $C \in y$. Then by IH, it comes $\mu, y \models C$. Hence, $\mu, x \models \Box C$.

∎

As a direct consequence of the previous lemma, we have:

Corollary 1.13 If A has a fair open $(\rho_1 \cup \rho_2)$-tableau then A is $(\rho_1 \cup \rho_2)$-satisfiable. Hence our tableaux calculi are complete under the fairness assumption.

1.8 Soundness

In this subsection, we prove the soundness of our tableaux calculi: if a formula A is $(\rho_1 \cup \rho_2)$-closed then A is $(\rho_1 \cup \rho_2)$-unsatisfiable. The technique we use for proving the soundness of our tableaux is simple. We prove that all rules preserve the "satisfiability" of the pattern involved in its application. In our sense, a pattern is $(\rho_1 \cup \rho_2)$-satisfiable iff there exists a $(\rho_1 \cup \rho_2)$-model that contains it and satisfies its formulas. We formally develop this below.

Definition 1.14 Let $\Upsilon = (\mathcal{N}, \Sigma, \Phi)$ be a labelled $(\rho_1 \cup \rho_2)$-RDAG and $\mu = (W, R, \tau)$ be a $(\rho_1 \cup \rho_2)$-model; let h be a function such that $h(\mathcal{N}) \subseteq W$ and $\forall n_1, n_2 \in \mathcal{N}\colon (n_1, n_2) \in \Sigma \Rightarrow (h(n_1), h(n_2)) \in R$.

- h is called an *embedding* from Υ to μ (or h matches Υ to μ);
- μ satisfies Υ via h iff $\forall n \in \mathcal{N}\colon A \in \Phi(n) \Rightarrow \mu, h(n) \models A$;
- μ satisfies Υ iff there exists an embedding h from Υ to μ such that μ satisfies Υ via h.

Lemma 1.15 Let $\Upsilon \Longrightarrow \Upsilon'$ be a rule of some set ρ (resp. $\Upsilon \Longrightarrow \Upsilon'$ or Υ'' for rule $\vee$); then if some ρ-model μ satisfies Υ then it satisfies Υ' (resp. then it satisfies Υ' or Υ'').

PROOF. If we suppose that μ satisfies Υ via some embedding h we just have to exhibit an embedding h' such that μ satisfies Υ' via h' (resp. such that μ satisfies Υ' or Υ'' via h'). This is done by analysing every rule. We only do it for the $\Diamond$ rule, for one structural rule and for one propagation rule. For classical rules, it is immediate: just take $h' = h$.

- Rule $\Diamond$: $\Upsilon = (\mathcal{N} = \{n_0\}, \Sigma = \emptyset, \Phi = \{(n_0, \Diamond A)\})$ rewrites into $\Upsilon' = (\mathcal{N} \cup \{n_1\}, \Sigma \cup \{(n_0, n_1)\}, \Phi \cup \{(n_1, A)\})$.
  If μ satisfies Υ via h then $\mu, h(n_0) \models \Diamond A$, hence $\exists y \in R(h(n_0))\colon \mu, y \models A$; let y_1 be such a y, and define $h'(n_1) = y_1$ and $h'(n_0) = h(n_0)$. μ satisfies Υ' via h', since $(h'(n_0), h'(n_1)) \in R$ and $\mu, h'(n_1) \models A$.

- Rule De: $\Upsilon = (\mathcal{N} = \{n_0, n_1\}, \Sigma = \{(n_0, n_1)\}, \Phi = \{(n_0, S_0), (n_1, S_1)\})$ rewrites into $\Upsilon' = (\mathcal{N} \cup \{n_2\}, \Sigma \cup \{(n_0, n_2), (n_2, n_1)\}, \Phi \cup \{(n_2, \emptyset)\})$.
  If μ satisfies Υ via h then $(h(n_0), h(n_1)) \in R$, and since R is dense $\exists z\colon (h(n_0), z) \in R$ and $(z, h(n_1)) \in R$. Let z_2 be such a z and define $h'(n_2) = z_2$ and $h'(n) = h(n)$ for $n \neq n_2$. μ satisfies Υ' via h', since $(h'(n_0), h'(n_2)) \in R$ and $(h'(n_2), h'(n_1)) \in R$, and $\Phi(n_2) = \emptyset$.

  For propagation rules, we just have to prove that we are done by taking $h' = h$.

- Rule $5_{\rightarrow}$: $\Upsilon = (\mathcal{N} = \{n_0, n_1, n_2\}, \Sigma = \{(n_0, n_1), (n_0, n_2)\}, \Phi = \{(n_0, S_0), (n_1, S_1 \cup \{\Box A\}), (n_2, S_2)\})$ rewrites into $\Upsilon' = (\mathcal{N}, \Sigma, \Phi \cup \{(n_2, \Box A)\})$.
  If μ satisfies Υ via h then $\mu, h(n_1) \models \Box A$;
  Also, since R is euclidean, we have:
  $\mu, h(n_0) \models \Box(\Box A \rightarrow \Box\Box A)$ (valid formula of euclidean models)
  $\Rightarrow \mu, h(n_1) \models \Box A \rightarrow \Box\Box A$ (since $(h(n_0), h(n_1)) \in R$)
  $\Rightarrow \mu, h(n_1) \models \Box\Box A$ (since $\mu, h(n_1) \models \Box A$)
  $\Rightarrow \mu, h(n_0) \models \Diamond\Box\Box A$ (since $(h(n_0), h(n_1)) \in R$)
  $\Rightarrow \mu, h(n_0) \models \Box\Box A$ ($\Diamond\Box\Box A \rightarrow \Box\Box A$ is valid in euclidean models)
  $\Rightarrow \mu, h(n_2) \models \Box A$ (since $(h(n_0), h(n_2)) \in R$).

∎

Corollary 1.16 If A is $(\rho_1 \cup \rho_2)$-satisfiable then it has an open $(\rho_1 \cup \rho_2)$-tableau. Hence our tableaux calculi are sound.

PROOF. If A is $(\rho_1 \cup \rho_2)$-satisfiable by some world x of some $(\rho_1 \cup \rho_2)$-model μ, then its starting labelled RDAG: $(\{n_0\}, \emptyset, \{(n_0, A)\})$ is satisfied by μ (via the embedding $h\colon n_0 \mapsto x$). Hence, at least one of its $(\rho_1 \cup \rho_2)$-tableaux must be open since no closed tableau is satisfiable by μ. ∎

1 Extensions to other propertiesExtensions to other properties Our work extends easily to other properties of group 1 (almost-reflexivity: $\forall x(\exists u\colon (u, x) \in R \Rightarrow (x, x) \in R)$, almost-transitivity: $\forall x, y, z, u\ ((x, y) \in R \wedge (y, z) \in R \wedge (z, u) \in R) \Rightarrow (y, u) \in R$, ...). First complete the Relational Closure lemma (1.5) and then check that the closure under this new property of a ρ_2-RGRAPH is still a ρ_2-RGRAPH (Structural lemma). Then design one or several rules for this property e.g. for almost-reflexivity, the natural rule such as:

$S0 \bullet\!\!-\!\!-\!\!\bullet S1, \Box A \quad \Longrightarrow \quad S0 \bullet\!\!-\!\!-\!\!\bullet S1, \Box A, A$
(it is obviously sound). Then prove that this/these rule(s) allow(s) to correctly propagate formulas (Box lemma) with the help of the Relational Closure lemma.
For new properties of group 2 (like 3-density: $(x, y) \in R \Rightarrow \exists u, v\colon (x, u) \in R \wedge (u, v) \in R \wedge (v, y) \in R$), one must first define the underlying structure (here 3-dense RGRAPH) and extend the Structural lemma (if possible). Then designing a corresponding sound structural rule is straightforward, and completeness is for free.

2 Terminating tableaux for K4.C and K4.De

As they are defined, naive tableaux may run infinitely. As an example, a $\{Ser, Tr\}$-tableau for $\Box\Diamond p$ runs infinitely because of rules (4) and ($\Diamond$) that apply infinitely. In this case, of course, the nodes generated all contain the same formulas and thus the tableau loops.

2.1 The kernel approach

The basic idea once completeness has been obtained to get a decision procedure is to find for each logic a so-called family of "kernels": a kernel is simply a finite structure able to simulate the infinite tableaux obtained with a naive algorithm that would just implement tableaux as presented above. In this sense, it is well-known that kernels for S4 (i.e. KT4) are finite trees (since no rule makes two sibling nodes having a common successor). We will show below that kernels for KD4.C are finite sequences of finite lattices, and we will show a similar result for KD4.De.
We will focus on two characteristic systems, namely KD4.C and KD4.De, it is straightforward to lift the present results to systems without axiom D, or with axiom T (that subsumes axiom De), and with both axioms C and De. But first, we will apply it to KD4: nothing new will be stated, but this will help in understanding the sequel.

In what follows we will use some conventions and notations that are presented here:

1. Given a tableau $Y = (N, \Sigma, \Phi)$ and a rule (S) (among those that lead to add a new node: (D), ($\Diamond$), (De), (C),…) we will denote by N^{S} the set of nodes created by applying this rule at some iteration, and dually, Σ^{S} the set of edges created by applying this rule;
2. For sake of simplicity, we decided to use bold-face symbols for those which concern kernels, while naive tableau will be denoted by non-bold symbols (thus Y denotes a naive tableau, while $\mathbf{Y}$ denotes a kernel).

2.2 Kernels for KD4

The set of rules that will be used is: all classical rules, and rules (D), ($\Diamond$), (K) and (4). Kernels for KD4 are simply naive tableaux provided with a strategy that allows to conclude that some tableau is open after only finitely many steps; this proves that kernels for KD4 are finite trees. Let A be the starting formula, we get the following non-deterministic (w.r.t. the order of application of the rules) algorithm that computes a sequence $(\mathbf{Y})_i$ of RDAG:
Starting from $\mathbf{Y}_0 = (\mathbf{N}_0, \mathbf{\Sigma}_0, \mathbf{\Phi}_0)$ where $\mathbf{N}_0$ only consists of one node $\mathbf{r}$ (the root), $\mathbf{\Sigma}_0$

is empty and $\mathbf{\Phi}_0$ associate the formula A with $\mathbf{r}$. Then compute $\mathbf{Y}_{i+1} = (\mathbf{N}_{i+1}, \mathbf{\Sigma}_{i+1}, \mathbf{\Phi}_{i+1})$ from $\mathbf{Y}_i = (\mathbf{N}_i, \mathbf{\Sigma}_i, \mathbf{\Phi}_i)$ by applying successively each of the following steps:

1. Loop step: consider all nodes $x \in \mathbf{N}_i$ such that $\exists y \in \mathbf{N}_i$ and y is an ancestor of x (i.e. $(y, x) \in \mathbf{\Sigma}_i^{Tr}$) and $\mathbf{\Phi}_i(x) \subseteq \mathbf{\Phi}_i(y)$ and set them as *loop nodes*; (nodes that are "contained" in one already present in the tree need not to be further developed)
2. Classical step: apply classical rules $((\perp), (\neg), (\vee), (\wedge))$ on all nodes as much as possible (also known as classical saturation: as usual, in a given node, a given formula will be treated only once);
3. Structural step: apply rules (D) and $(\Diamond)$ on each *non loop node* where they have not been applied yet (As usual, rule (D) must be applied only once on each node while rule $(\Diamond)$ must be applied once for each formula $\Diamond B$ of each node.);
4. Propagation step: apply rules (K) and (4) as much as possible.

The above algorithm must be applied until for some i, either $\mathbf{Y}_i$ is closed or $\mathbf{Y}_{i+1} = \mathbf{Y}_i$ (i.e. there are loop nodes on each branch)[5]. As in the case of naive tableaux, the tableau for A is said closed if all possible kernels for A are closed, it is open otherwise.

Theorem 2.1 The strategy given above is sound for KD4.

Proof. Straightforward since the resulting algorithm is a fair restriction of the naive one which is sound. ∎

For proving the completeness of this strategy, we need first to establish the following lemma[6]:

Lemma 2.2 Let $\mathbf{Y} =< \mathbf{N}, \mathbf{\Sigma}, \mathbf{\Phi} >$ be a kernel of root $\mathbf{r}$ for A (obtained by the above strategy) then there exists an naive tableau Y for A with $Y =< N, \Sigma, \Phi >$ of root r and such that:

$$(\heartsuit)\ \forall x \in N : \exists u \in \mathbf{N}\ [\Phi(x) \subseteq \mathbf{\Phi}(u)\ \&\ \forall y \in N : \exists v \in \mathbf{N} : ((x, y) \in \Sigma \Rightarrow ((u, v) \in \mathbf{\Sigma}\ \&\ \Phi(y) \subseteq \mathbf{\Phi}(v))]$$

Proof. It is done by induction. Let

$$(\heartsuit_i)\ \forall x \in N_i : \exists u \in \mathbf{N}\ [\Phi_i(x) \subseteq \mathbf{\Phi}(u)\ \&\ \forall y \in N_i : \exists v \in \mathbf{N} : (x, y) \in \Sigma_i \Rightarrow: ((u, v) \in \mathbf{\Sigma}\ \&\ \Phi_i(y) \subseteq \mathbf{\Phi}(v))]$$

Induction base: True since $\Phi_0(r) = \{A\} \subseteq \mathbf{\Phi}(\mathbf{r})$, and $\Sigma_0 = \emptyset$.
Induction step: The induction hypothesis is $(\heartsuit_i)$, we examine the rule that may lead from Y_i to Y_{i+1}
We only treat rules $\perp$ and $\vee$, the other classical rules are simpler.

Rule $(\perp)$: let us suppose that rule $\perp$ may be applied to some node x of N_i, then for some formula B, $\{B, \neg B\} \subseteq \Phi_i(x)$ hence by IH: $\exists u : \{B, \neg B\} \subseteq \mathbf{\Phi}(u)$; thus $\perp \in \mathbf{\Phi}(u)$, by rule $(\perp)$.

[5] Since there are no backwards rules, rules (K) and (4) must be applied only in order to propagate formulas in new nodes introduced at step 3.
[6] The whole completeness proof is standard (the reader may find details in [12] for example).

Rule ($\vee$): let $\Phi_i(x) = S, A \vee B$ then by IH $S \subseteq \mathbf{\Phi}(u)$ and $A \vee B \in \mathbf{\Phi}(u)$, hence by rule ($\vee$) $C \in \mathbf{\Phi}(u)$ (for some $C \in \{A, B\}$); then we set[7] $\Phi_{i+1}(x) = \Phi_i(x) \cup \{C\}$ and ($\heartsuit_{i+1}$) holds.

Rule ($\Diamond$): let $\Phi_i(x) = S, \Diamond A$ and suppose that applying rule ($\Diamond$) leads to introduce y with $\Phi_{i+1}(y) = A$ and $(x, y) \in \Sigma_{i+1}$ then by IH: $\exists u \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u) \Rightarrow$ $(\exists u' \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u')$ and u' is not a loop node)
(since either u is *not a loop node* or it is a loop node but then there must be another node u' such that $\Phi(u) \subseteq \Phi(u')) \Rightarrow S \subseteq \mathbf{\Phi}(u')$ and $\Diamond A \in \mathbf{\Phi}(u')$
$\Rightarrow$(by rule ($\Diamond$)) $\exists v' \in \mathbf{N} : (u', v') \in \mathbf{\Sigma}$ & $A \in \mathbf{\Phi}(v') \Rightarrow (\heartsuit_{i+1})$

Rule (D): As for rule ($\Diamond$) but with $\Phi_{i+1}(y) = \emptyset$

Rule (K) –resp. (4): let $(x, y) \in \Sigma_i$ & $\Box A \in \Phi_i(x)$; by IH $\exists u, v \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u)$ & $\Phi_i(y) \subseteq \mathbf{\Phi}(v)$ & $(u, v) \in \mathbf{\Sigma}$, since $\Box A \in \mathbf{\Phi}(u)$, by rule (K) –resp.(4)– $A \in \mathbf{\Phi}(v)$ –resp. $\Box A \in \mathbf{\Phi}(v)$–[8].

∎

Theorem 2.3 The strategy given above is complete for KD4.

Proof. The lemma 2.2 implies straightforwardly $\forall x \in N : \exists u \in \mathbf{N} : \Phi(x) \subseteq \mathbf{\Phi}(u)$. Hence if $\mathbf{Y}$ is open then so is Y. ∎

Now we come to the termination argument which is standard:

Theorem 2.4 The strategy given above is terminating.

Proof. Since there are finitely many subsets of the set of subformulas of the initial formula A, and since each rule only introduces such subformulas, at some iteration each branch in an open kernel should have loop nodes. ∎

Remark 2.5 The above argument is very rough and leads us to conclude that satisfiability for KD4 is exponential w.r.t. the input formula (since we gave an algorithm that runs in exponential time using exponential space). This may be improved: it is well-known that the complexity of satisfiability for KD4 is in PSPACE (see [18] for details), this is because any branch produced by the algorithm is in fact of polynomial length (i.e. it contains a polynomial number of nodes), each node containing polynomially many subformulas. Then, since the computation may be performed one branch after the other (depth-first computation) only polynomial space is needed. In other words, though a tableau may be exponentially large, it does not need to be exponentially deep. In our case, we would have to modify our algorithm in order to develop only one branch at a time. This may be done by applying rule (D) or ($\Diamond$) only once and on only one node at a time.

2.3 Kernels for KD4.C

In this subsection, we give a terminating non-deterministic tableau calculus for the system KD4.C, that can be straightforwardly modified in order to apply to K4.C and

[7] Note that we are proving that *there exists* such a Y_{i+1}; this amounts, in the case of rule $\vee$ to proving that there exists a choice among A and B such that ($\heartsuit_{i+1}$) holds.

[8] Note that because of point 4 of the algorithm, we are sure that propagation rules are applied at some iteration of the tableau computation.

KT4.C.

For this we define a strategy to be applied on naive tableaux as defined previously; the set of rules that is needed is: all classical rules, and rules (D), ($\Diamond$), (K), (4) and (C) (rule (C*) is superfluous since it is subsumed by the rule (D)).
This strategy mainly consists of the following:

1. Compute a KD4-kernel (using only classical rules, and rules (D), ($\Diamond$), (K) and (4)). This provides a finite tree (either closed or looping on each branch).
2. Create a successor common to each loop node[9] (we will call this node the **anti-root**) and propagate formulas (rules (K) and (4)) into it. Then go back to step 1, with the anti-root as the starting node (and as such, as the new root).

Stop the computation when: the tableau closes at any step, or if **looping** anti-roots are generated.

The algorithm runs as follows:
Starting from $\mathbf{Y}_0 = (\mathbf{N}_0, \mathbf{\Sigma}_0, \mathbf{\Phi}_0)$ where $\mathbf{N}_0$ only consists of only one node $\mathbf{r}^0$ (the root), $\mathbf{\Sigma}_0$ is empty and $\mathbf{\Phi}_0$ associate the formula A with $\mathbf{r}^0$.

1. Compute $\mathbf{Y}_{i+1} = (\mathbf{N}_{i+1}, \mathbf{\Sigma}_{i+1}, \mathbf{\Phi}_{i+1})$ from $\mathbf{Y}_i = (\mathbf{N}_i, \mathbf{\Sigma}_i, \mathbf{\Phi}_i)$ by applying the strategy for KD4 only (i.e. by using only classical rules, and rules (D), ($\Diamond$), (K) and (4)). In $\mathbf{Y}_{i+1}$, each branch loops (i.e. each of its leaves is a loop node), or else, it is closed.

   Let us denote by $Loop_{i+1}$ the set of nodes of $\mathbf{N}_{i+1}$ that are loop nodes
2. Compute $\mathbf{Y}_{i+2} = (\mathbf{N}_{i+2}, \mathbf{\Sigma}_{i+2}, \mathbf{\Phi}_{i+2})$ from $\mathbf{Y}_{i+1} = (\mathbf{N}_{i+1}, \mathbf{\Sigma}_{i+1}, \mathbf{\Phi}_{i+1})$ by:
   - $\mathbf{N}_{i+2} = \mathbf{N}_{i+1} \cup \{\mathbf{r}^{i+2}\}$, where $\mathbf{r}^{i+2}$ is a new node,
   - $\mathbf{\Sigma}_{i+2} = \mathbf{\Sigma}_{i+1} \cup \{(x, \mathbf{r}^{i+2}) \colon x \in Loop_{i+1}\}$
   - $\mathbf{\Phi}_{i+2}(\mathbf{r}^{i+2}) = \bigcup_{x \in Loop_{i+1}} (\mathbf{\Phi}_{i+1}(x))^{\Box}$;
     where $S^{\Box}$ denotes the set $\{A, \Box A \colon \Box A \in S\}$

The above algorithm must be applied until for some i and some l, either $\mathbf{Y}_i$ is closed or $\mathbf{\Phi}_i(\mathbf{r}^l) \subseteq \mathbf{\Phi}_i(\mathbf{r}^{k<l})$, where $\mathbf{r}^l$ denotes the last anti-root generated. This strategy is graphically represented in figure 1.
The key feature of our algorithm is that it is sufficient to compute only a unique common successors for each of the nodes of the KD4 kernel, and not one for each pair of nodes that have a common direct ancestor. In some sense, the anti-roots sum up all the information of the intermediary nodes (those we would get by naively applying rule C). It should be clear that this property only holds because of transitivity.

Theorem 2.6 The strategy given above is sound for KD4.C.

Proof. Straightforward since the resulting algorithm is a fair restriction of the naive one which is sound. ∎

Now, for proving the completeness of the algorithm, the main tool is the following lemma, which plays the same role for KD4.C as the lemma 2.2 for KD4(this lemma assumes that the algorithm always terminates. This will be proved in lemma 2.9):

[9] See previous section

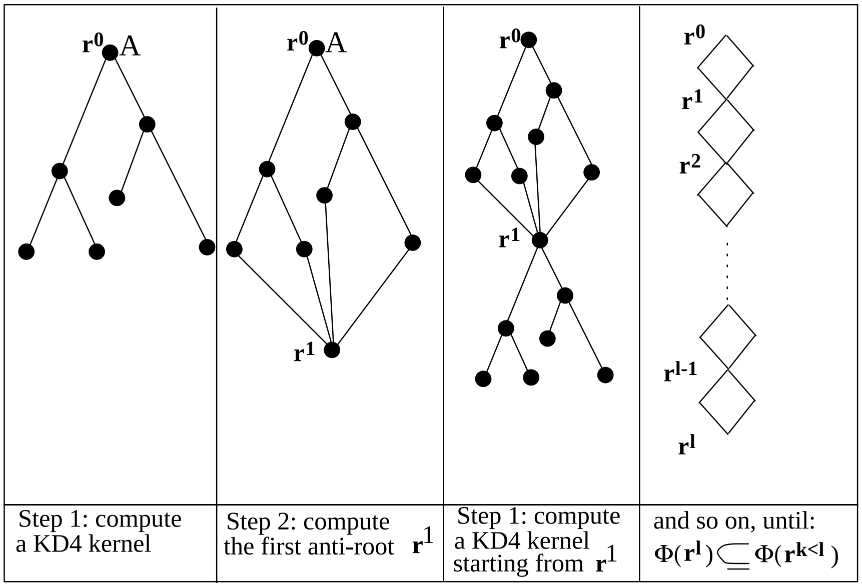


FIG. 1. The strategy for KD4.C

Lemma 2.7 Let $\mathbf{Y} =< \mathbf{N}, \mathbf{\Sigma}, \mathbf{\Phi} >$ be a kernel for A, last of the sequence $\mathbf{Y}_0, \dots, \mathbf{Y}_l$ (whose anti-root is $\mathbf{r}^l$) obtained by the strategy given, then there exists a naive tableau $Y =< N, \Sigma, \Phi >$ for A such that:

(♠) $\forall x \in N : \exists u \in \mathbf{N} : \Phi(x) \subseteq \mathbf{\Phi}(u)$ and
$\forall y($
if $((x,y) \in \Sigma^{\diamond}$ then $\exists v \in \mathbf{N}$ s.th. $(u,v) \in \mathbf{\Sigma}$ & $\Phi(y) \subseteq \mathbf{\Phi}(v))$ and
if $((x,y) \in \Sigma^{\mathrm{C}}$ then $\Phi(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)))$

NB: $\Sigma^{\diamond}$ and Σ^{C} are defined page 64.

PROOF. It is done by induction. Let

($♠_i$) $\forall x \in N_i : \exists u \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u)$ and
$\forall y($
if $((x,y) \in \Sigma_i^{\diamond}$ then $\exists v \in \mathbf{N}$ s.th. $(u,v) \in \mathbf{\Sigma}$ & $\Phi_i(y) \subseteq \mathbf{\Phi}(v))$ and
if $((x,y) \in \Sigma_i^{\mathrm{C}}$ then $\Phi_i(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)))$.

Induction base: True since $\Phi_0(x) = \{A\} \subseteq \mathbf{\Phi}(\mathbf{r}^0)$, and $\Sigma_0^{\diamond} = \Sigma_0^{\mathrm{C}} = \emptyset$.
Induction step: The induction hypothesis is ($♠_i$), we examine the rule that may lead from Y_i to Y_{i+1}:

- Classical rules: As for KD4, it is straightforward to show that Y_{i+1} may be defined such that ($♠_{i+1}$) holds.
- Rule $(\diamond)$: In this case,
  $N_{i+1} = N_i \cup \{y\}$ where y is a new node,

$\Sigma^{\Diamond}_{i+1} = \Sigma^{\Diamond}_i \cup \{(x,y)\}$ (rule ($\Diamond$) is applied on node x),
$\Sigma^{\mathrm{C}}_{i+1} = \Sigma^{\mathrm{C}}$,
$\Phi_{i+1}(y) = \{A\}$, (since $\Diamond A \in \Phi_i(x)$)

Then, by IH: $\exists u \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u)$
$\Rightarrow \exists u' \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u')$ and u' is not a loop node (since either u is *not a loop node* or it is a loop node but then there must be another node u' such that $\Phi(u) \subseteq \Phi(u')$)
$\Rightarrow$ (by rule ($\Diamond$)) $\exists v' \in \mathbf{N} : (u', v') \in \mathbf{\Sigma}$ & $A \in \mathbf{\Phi}(v') \Rightarrow (\spadesuit_{i+1})$.

- Rule (D): As for rule ($\Diamond$) but with $\Phi_{i+1}(y) = \emptyset$
- Rule (4) and (K): In this item, $\mathbf{r}^l$ denotes the last anti-root (which has no successor) Let $(x,y) \in \Sigma_i$ and $\Phi_i(x) = S, \Box A$ and $\Phi_i(y) = S'$. There are two cases according to whether $(x,y) \in \Sigma^{\Diamond}_i$ or $(x,y) \in \Sigma^{\mathrm{C}}_i$.
  1. $(x,y) \in \Sigma^{\Diamond}_i$:
     it is enough to prove that $\exists u, v \in \mathbf{N} : (u,v) \in \mathbf{\Sigma}$ & $\Phi_{i+1}(x) \subseteq \mathbf{\Phi}(u)$ & $\Phi_{i+1}(y) \subseteq \mathbf{\Phi}(v)$
     By IH, $\exists u, v : (u,v) \in \mathbf{\Sigma}$ & $\Phi_i(x) \subseteq \mathbf{\Phi}(u)$ & $\Phi_i(y) \subseteq \mathbf{\Phi}(v)$, hence $\Box A \in \mathbf{\Phi}(u)$
     $\Rightarrow$ (by rule (4) and (K)) $A, \Box A \in \mathbf{\Phi}(v)$
     $\Rightarrow \Phi_{i+1}(x) = \Phi_i(x) \subseteq \mathbf{\Phi}(u)$ & $\Phi_{i+1}(y) = \Phi_i(y) \cup \{A, \Box A\} \subseteq \mathbf{\Phi}(v)$;
  2. $(x,y) \in \Sigma^{\mathrm{C}}_i$:
     it is enough to prove that $\Phi_{i+1}(x) \subseteq \mathbf{\Phi}(u)$ & $\Phi_{i+1}(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)$, and there are two subcases:
     - Either $u \neq \mathbf{r}^l$: then by IH, $\Box A \in \mathbf{\Phi}(u)$ & $\Phi_i(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)$
       $\Rightarrow A, \Box A \in \mathbf{\Phi}(\mathbf{r}^l)$ & $\Phi_i(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)$, by straightforward application of rule (4) and (K), and the definition of the sets $\mathbf{\Phi}(\mathbf{r}^i)$.
       $\Rightarrow \Phi_{i+1}(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)$
     - Or $u = \mathbf{r}^l$: and then by IH, $\Box A \in \mathbf{\Phi}(u) = \mathbf{\Phi}(\mathbf{r}^l)$. Since $\mathbf{r}^l$ is the last anti-root, we have: $\mathbf{\Phi}(\mathbf{r}^l) \subseteq \mathbf{\Phi}(\mathbf{r}^{k<l}) \Rightarrow \Box A \in \mathbf{\Phi}(\mathbf{r}^k) \Rightarrow A, \Box A \in \mathbf{\Phi}(\mathbf{r}^l)$ (again by application of rule (4) and (K), and the definition of the sets $\mathbf{\Phi}(\mathbf{r}^i)$)
       $\Rightarrow \Phi_{i+1}(y) \subseteq \mathbf{\Phi}(\mathbf{r}^l)$.
- Rule (C): This case is trivial since with $N_{i+1} = N_i \cup \{y\}$ (y is the new node) and $\Phi_{i+1}(y) = \emptyset \subseteq \mathbf{\Phi}(\mathbf{r}^l)$ we have $(\spadesuit_{i+1})$.

∎

We can now state:

Theorem 2.8 The strategy given above is complete for KD4.C.

Proof. Similar to that of theorem 2.3. ∎

Concerning the termination of the corresponding algorithm, we have:

Theorem 2.9 The strategy given above is terminating.

Proof. Our algorithm consists of two nested while-loops: for the inner one, the argument is as for KD4, concerning the outer one, the argument is nevertheless the same, there are finitely many distinct sets $\mathbf{\Phi}(\mathbf{r}^l)$, hence some loop must appear. ∎

Again, this only gives a bad upper bound for the complexity of satisfiability for KD4.C: it is at least in EXPTIME. We improve this decidability result:

Theorem 2.10 The complexity of satisfiability for KD4.C is in PSPACE.

Proof. Since there exist PSPACE algorithms for KD4, it is clear that the inner while-loop of our algorithm may use only polynomial space (but exponential time) if we adopt any PSPACE algorithms for KD4. Moreover, the next anti-root (provided the current tableau does not close) may be computed by updating it each time we obtain a loop node in the inner while-loop (we do not need to compute all the loop nodes first). Concerning the outer while-loop of our algorithm, we only need to memorize the branch consisting of the anti-roots that have been generated: then Ladner's argument still apply and only polynomially many anti-roots may be generated before being included in some previous one. ∎

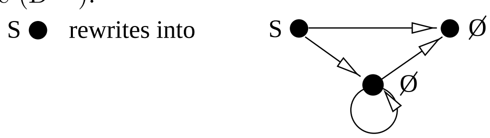
2.4 Kernels for KD4.De

In this subsection, we give a terminating tableau calculus for the system KD4.De, that can be straightforwardly modified in order to apply to K4.De (Note that the system KT4.De is the same as KT4 i.e. S4).

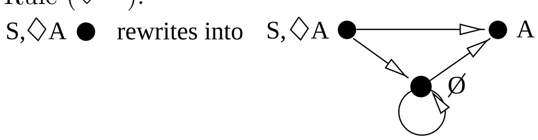
The algorithm consists in using a modified rule for the treatment of density, together *with the loop step of KD4* (cf. p. 65).

Consider the new following rules:

Rule (D^{De}):



Rule ($\Diamond^{De}$):



These two rules introduce what we call here *reflexive nodes* w.r.t. the other nodes that we call *non reflexive nodes*. These two new rules only apply on non reflexive nodes (of course they apply only once), while usual rules (D) and ($\Diamond$) apply on reflexive nodes. Thus the set of rules for KD4.De is: all classical rules, and rules (K), (4), rules ($\Diamond^{De}$) and (D^{De}) for non reflexive nodes, and rules (D) and ($\Diamond$) for reflexive nodes.
We will prove below that this set of rules is both complete and sound: as a consequence, in order to handle density (together with transitivity) one just need to consider models with only one intermediary world having a reflexive edge. In other words, the sequel will also prove that KD4.De is characterized by the Kripke frames (W_1, W_2, R) where: $\forall z \in W_2 : (z, z) \in R$ and $\forall x, y \in W_1 : (x, y) \in R \Rightarrow \exists z \in W_2 : ((x, z) \in R \,\&\, (z, y) \in R)$, that is to say, frames where there are two kinds of worlds (non reflexive ones and reflexive ones) such that the accessibility relation is reflexive on reflexive ones, and there is always a reflexive node between two non reflexive ones.

The algorithm we propose is the same as that for KD4, but using rules ($\Diamond$) and (D), or ($\Diamond^{De}$) and (D^{De}) according to the type of node (reflexive or not). Also, a node in this algorithm is not considered as its own ancestor (otherwise the loop step would apply immediately!).
It must be applied until for some i, either $\mathbf{Y}_i$ is closed or $\mathbf{Y}_{i+1} = \mathbf{Y}_i$ (i.e. there are loop nodes on each branch).

For convenience, we introduce here the notion of *constrained naive tableaux*, which are naive tableaux such that each application of rules (D) or ($\Diamond$) is followed by an application of rule (De): this amounts to introduce the two rules (D+De) and ($\Diamond$+De) in place of (D), ($\Diamond$) and (De). Of course, a constrained naive tableau is a naive tableau.

For proving completeness, we rely (as in the case of KD4) on the following lemma:

Lemma 2.11 Let $\mathbf{Y} =< \mathbf{N}, \mathbf{\Sigma}, \mathbf{\Phi} >$ be a kernel for A (obtained by the above strategy) then there exists a constrained naive tableau Y for A with $Y =< N, \Sigma, \Phi >$ and such that:

($\heartsuit$) $\forall x \in N : \exists u \in \mathbf{N}$
$[\Phi(x) \subseteq \mathbf{\Phi}(u)$ and
$\forall y \in N : \exists v \in \mathbf{N}$ s.th.
if $((x, y) \in \Sigma$ then $((u, v) \in \mathbf{\Sigma}$ & $\Phi(y) \subseteq \mathbf{\Phi}(v))]$

Proof. It is again done by induction. Let

($\heartsuit_i$) $\forall x \in N_i : \exists u \in \mathbf{N}$
$[\Phi_i(x) \subseteq \mathbf{\Phi}(u)$ and
$\forall y \in N_i : \exists v \in \mathbf{N}$ s.th.
if $(x, y) \in \Sigma_i$ then $((u, v) \in \mathbf{\Sigma}$ & $\Phi_i(y) \subseteq \mathbf{\Phi}(v))]$

Induction base: True since $\Sigma_0 = \emptyset$.
Induction step: The induction hypothesis is ($\heartsuit_i$), we examine the rule that may lead from Y_i to Y_{i+1}: concerning the classical rules, rules (K) and (4), the proof is similar to that of KD4, we only consider the case of rule $(D + De)$ (the case of rule $(D + \Diamond)$ is similar):

Rule $(D{+}De)$: Let $x \in N_i$ and y, z be two new nodes such that: $N_{i+1} = N_i \cup \{y, z\}$, $\Sigma^{\Diamond}_{i+1} = \Sigma^{\Diamond}_i \cup \{x, y\}$, $\Sigma^{\mathrm{De}}_{i+1} = \Sigma^{\mathrm{De}}_i \cup \{(x, z), (z, y), (z, z)\}$, $\Phi_{i+1}(x) = \Phi_i(x)$ and $\Phi_{i+1}(y) = \Phi_{i+1}(z) = \emptyset$, this corresponds to applying rule (D) and then rule (De).
We just have to prove that ($\heartsuit_{i+1}$) is true for (x, y), (x, z) and (z, y) respectively, i.e. we must give three pairs of nodes of $\mathbf{\Sigma}$ that satisfy ($\heartsuit_{i+1}$)
By IH, $\exists u \in \mathbf{N} : \Phi_i(x) \subseteq \mathbf{\Phi}(u)$
– If u is not a reflexive node: $\Rightarrow$ (by rule (D^{De})) $\exists v, w \in \mathbf{N} : \Phi_{i+1}(x) \subseteq \mathbf{\Phi}(u)$(since x is unchanged) and $\Phi_{i+1}(y) = \emptyset \subseteq \mathbf{\Phi}(v)$ and $\Phi_{i+1}(z) = \emptyset \subseteq \mathbf{\Phi}(w)$; moreover we have: $(u, v), (u, w), (v, w), (w, w) \in \mathbf{\Sigma}$
We are done with the pairs (u, v), (u, w) and (v, w) respectively.
– If u is a reflexive node: $\Rightarrow$ (by rule (D)) $\exists v \in \mathbf{N} : \Phi_{i+1}(x) \subseteq \mathbf{\Phi}(u)$(since x is unchanged) and $\Phi_{i+1}(y) = \emptyset \subseteq \mathbf{\Phi}(v)$; moreover we have: $(u, u), (u, v) \in \mathbf{\Sigma}$
We are done with the pairs (u, v), (u, u) and (u, v) respectively.

■

Theorem 2.12 The strategy given above is complete for KD4.De.

Proof. Direct consequence of the lemma 2.11. ■

It remains to prove the soundness of the algorithm: contrarily to the case of KD4.C and KD4, soundness must be proven since it is not a restriction of the naive one (but it is fair). To this aim, we will prove below that if there exists an open tableau for A, then there exists an open kernel for A.

In order to establish the proof of this we need some additionnal considerations. Given a naive tableau $Y =< N, \Sigma >$ we define the sets $\texttt{bet}(x, y)$ to be the set of nodes of Y which are between x and y by:

Definition 2.13 Let $Y =< N, \Sigma >$ be a naive tableau, for all x, $y \in \Sigma$ we set: $\texttt{bet}(x, y) = \{z \in N\colon (x, z) \in \Sigma^+ \;\&\; (z, y) \in \Sigma^+\}$ i.e. the set of nodes which are descendants of x and ancestors of y.

We also need the following fact:

Fact 2.14 Let $Y =< N, \Sigma, \Phi >$ be a naive tableau for A (open or closed), then $\forall x, y \in N^\diamond$ if $(x, y) \in \Sigma^\diamond$ then $\exists u, v \in \texttt{bet}(x, y)$ s.th. $(u, v) \in \Sigma^+ \;\&\; \Phi(u) = \Phi(v)$

Proof. Direct consequence of the facts that there is a finite number of subsets of subformulas of A and that there are infinitely many nodes between such x and y. ■

Remark 2.15 Let us now consider infinite kernels (i.e. structures obtained by applying the above algorithm without the loop step - cf. p. 65): it should be clear that if there exists a open infinite kernel (for A) then there exists a finite one (note that nodes are finitely branching): we just have to cut the infinite branches at the loop nodes by applying the loop step. Thus we only have to show that from the open naive tableau Y, we can build an open infinite kernel $\mathbf{Y}$.

As in preceeding subsection we state and prove an intermediary lemma:

Lemma 2.16 Let $Y =< N, \Sigma, \Phi >$ be a naive tableau for A of root r then there exists an infinite kernel for A of root rr and denoted by $\mathbf{Y} =< \mathbf{N}, \mathbf{\Sigma}, \mathbf{\Phi} >$ such that:

($\clubsuit$) $\forall x \in \mathbf{N} : \exists u \in N$
$\mathbf{\Phi}(x) \subseteq \Phi(u)$ and
$[\forall y \in \mathbf{N} : \exists v \in N$ s.th.
if $((x, y) \in \mathbf{\Sigma}$
then $((u, v) \in \Sigma^+ \;\&\; \mathbf{\Phi}(y) \subseteq \Phi(v)) \;\&\; (x = y \Rightarrow \Phi(u) = \Phi(v))]$

Proof. This is again proved by induction, let:

($\clubsuit_i$) $\forall x \in \mathbf{N}_i : \exists u \in N$
$\mathbf{\Phi}_i(x) \subseteq \Phi(u)$ and
$\forall y \in \mathbf{N}_i : \exists v \in N$ s.th.
if $((x, y) \in \mathbf{\Sigma}_i$
then $((u, v) \in \Sigma^+ \;\&\; \mathbf{\Phi}_i(y) \subseteq \Phi(v)) \;\&\; (x = y \Rightarrow \Phi(u) = \Phi(v))$

Induction base: True since $\mathbf{N}_0 = \emptyset$ and $\mathbf{\Phi}_0(rr) = \emptyset \subseteq \Phi(r)$. Induction step: The induction hypothesis is ($\clubsuit_i$), we examine the rule that may leads from $\mathbf{Y}_i$ to $\mathbf{Y}_{i+1}$ We only treat the non-classical ones, classical cases are similar to those of previous proofs.
In order to simplify the notation, we will abbreviate the conjunction
$\mathbf{\Phi}_i(x) \subseteq \Phi(u) \,\&\, \mathbf{\Phi}_i(y) \subseteq \Phi(v)$ by $(x, y)_i \subseteq (u, v)$.

Rule $(\Diamond^{De})$– rule (D^{De}) is similar–: suppose rule $(\Diamond^{De})$ is applied on node x , thus (with the two new nodes y and z) we have: $\mathbf{N}_{i+1} = \mathbf{N}_i \cup \{y, z\}$, $\mathbf{\Sigma}_{i+1} = \mathbf{\Sigma}_i \cup \{(x, y), (x, z), (z, z), (z, y)\}$, $\mathbf{\Phi}_{i+1}(z) = \emptyset$ and $\mathbf{\Phi}_{i+1}(y) = \{A\}$:
then by IH, $\exists u \in N : \mathbf{\Phi}_i(x) \subseteq \Phi(u)$
$\Rightarrow$ (by rule $(\Diamond)$) $\exists v\colon (u, v) \in \Sigma \,\&\, \mathbf{\Phi}_{i+1}(y) = \{A\} \subseteq \Phi(v)$
$\Rightarrow$ (by rule (De) infinitely many times) $\exists w_0, w_1, \ldots\colon (u, w_i) \in \Sigma \,\&\, (w_0, v) \in \Sigma \,\&\, (w_{i+1}, w_i) \in \Sigma$
$\Rightarrow \exists w_k, w_l\colon (u, w_k) \in \Sigma^+ \,\&\, (w_k, w_l) \in \Sigma^+ \,\&\, (w_l, v) \in \Sigma^+ \,\&\, \Phi(w_k) = \Phi(w_l)$ (by the fact 2.14)
Then, we have: $(x, y)_{i+1} \subseteq (u, v)$, $(x, z)_{i+1} \subseteq (u, w_k) \subseteq (u, w_l) \,\&\, \Phi(w_k) = \Phi(w_l)$ and $(z, y)_{i+1} \subseteq (w_l, v)$

Rule (4) and (K): Let $\Phi_i(x) = S, \Box A \,\&\, \Phi_i(y) = S'$ and $(x, y) \in \Sigma_i$. There are two cases according to whether $x = y$ (reflexive node) or not.

$x \neq y$: By IH, $(x, y)_i \subseteq (u, v)$ and $(u, v) \in \Sigma^+$ hence $\Box A \in \Phi(u)$ and by rules (K) and (4), $\Box A, A \in \Phi(v)$;

$x = y$: By IH, $(x, x)_i \subseteq (u, v)$ and $(u, v) \in \Sigma^+$ hence $\Box A \in \Phi(u)$ and $\Box A \in \Phi(v)$, thus by rules (4) and (K) we have $A \in \Phi(v)$ and then $A \in \Phi(u)$.

∎

Lemma 2.17 Let $Y =< N, \Sigma, \Phi >$ be an open naive tableau for A, then there exists an open kernel for A.

Proof. Direct consequence of lemma 2.16 and by the remark about the infinite kernels (cf. page 72). ∎

We can now state:

Theorem 2.18 The strategy given above is sound for KD4.De.

Proof. Immediate consequence of lemma 2.17. ∎

And concerning the termination of the corresponding algorithm, we have:

Theorem 2.19 The strategy given above is terminating.

Proof. The argument is now exactly the same as for KD4. ∎

Again, this only gives a bad upper bound for the complexity of satisfiability for KD4.De: it is in EXPTIME. We improve this decidability result:

Theorem 2.20 The complexity of satisfiability for KD4.De is in PSPACE.

Proof. The proof is the same as for KD4, we just have to modify the notion of branch: here a branch is the longest path from the root to a leaf (thus including reflexive nodes). ∎

3 Complete tableaux for some bimodal logics with permutation and/or confluence

3.1 Modal logics and relational properties

The language of our bimodal logic have $\Box_{\mathtt{a}}$, $\Box_{\mathtt{b}}$,... and $\Diamond_{\mathtt{a}}$, $\Diamond_{\mathtt{b}}$,... as additionnal connectives w.r.t. classical logic. As usual, $\Diamond_{\mathtt{a}}$ and $\Diamond_{\mathtt{b}}$ abbreviate $\neg\Box_{\mathtt{a}}\neg$ and $\neg\Box_{\mathtt{b}}\neg$. Thus we will use the same axioms as above but with indexes a or b.

We investigate some systems based on K($\mathtt{a}$,$\mathtt{b}$) plus some of the axioms below. Among them, axioms involving both modalities are known as *interaction axioms*.
As in the monomodal case, with each of these axioms can be associated a relational property of the accessibility relations of the Kripke models (which are of the form $(W, R_{\mathtt{a}}, R_{\mathtt{b}}, m)$ where $R_{\mathtt{a}}$ and $R_{\mathtt{b}}$ are binary relations over W):

Axiom	Property	Notation
$\mathrm{T}_a = \Box_{\mathtt{a}} p \to p$	reflexivity	Ref_a
$4_a = \Box_{\mathtt{a}} p \to \Box_{\mathtt{a}}\Box_{\mathtt{a}} p$	transitivity	Tr_a
$\mathrm{T}_b = \Box_{\mathtt{b}} p \to p$	reflexivity	Ref_b
$4_b = \Box_{\mathtt{b}} p \to \Box_{\mathtt{b}}\Box_{\mathtt{b}} p$	transitivity	Tr_b

Group 3: Properties handled by propagation rules

Axiom	Property	Notation
$\mathrm{D}_a = \Box_{\mathtt{a}} p \to \Diamond_{\mathtt{a}} p$	seriality	Ser_a
$\mathrm{De}_a = \Diamond_{\mathtt{a}} p \to \Diamond_{\mathtt{a}}\Diamond_{\mathtt{a}} p$	density	Dens_a
$\mathrm{C}_a = \Diamond_{\mathtt{a}}\Box_{\mathtt{a}} p \to \Box_{\mathtt{a}}\Diamond_{\mathtt{a}} p$	confluence	Conf_a
$\mathrm{D}_b = \Box_{\mathtt{b}} p \to \Diamond_{\mathtt{b}} p$	seriality	Ser_b
$\mathrm{De}_b = \Diamond_{\mathtt{b}} p \to \Diamond_{\mathtt{b}}\Diamond_{\mathtt{b}} p$	density	Dens_b
$\mathrm{C}_b = \Diamond_{\mathtt{b}}\Box_{\mathtt{b}} p \to \Box_{\mathtt{b}}\Diamond_{\mathtt{b}} p$	confluence	Conf_b
$\mathrm{Per} = \Box_{\mathtt{a}}\Box_{\mathtt{b}} p \leftrightarrow \Box_{\mathtt{b}}\Box_{\mathtt{a}} p$	permutation	Per
$\mathrm{C}_{ab} = \Diamond_{\mathtt{b}}\Box_{\mathtt{a}} p \to \Box_{\mathtt{a}}\Diamond_{\mathtt{b}} p$	ab-confluence	Conf_{ab}

Group 4: Properties handled by structural rules

Where properties Per and Conf_{ab} are defined by:

- Per:
  $\forall x, y, z : ((x, y) \in R_{\mathtt{b}} \& (y, z) \in R_{\mathtt{a}}) \to (\exists u \colon (x, u) \in R_{\mathtt{a}} \& (u, z) \in R_{\mathtt{b}}))$, and
  $\forall x, y, z : ((x, y) \in R_{\mathtt{a}} \& (y, z) \in R_{\mathtt{b}}) \to (\exists u \colon (x, u) \in R_{\mathtt{b}} \& (u, z) \in R_{\mathtt{a}}))$

  or $R_{\mathtt{b}} \circ R_{\mathtt{a}} = R_{\mathtt{a}} \circ R_{\mathtt{b}}$
- Conf_{ab}:
  $\forall x, y, z : ((x, y) \in R_{\mathtt{b}} \& (x, z) \in R_{\mathtt{a}}) \to (\exists u \colon (y, u) \in R_{\mathtt{a}} \& (z, u) \in R_{\mathtt{b}}))$,

  or $\overline{R_{\mathtt{b}}} \circ R_{\mathtt{a}} = R_{\mathtt{a}} \circ \overline{R_{\mathtt{b}}}$

As a consequence of Sahlqvist's theorem [24], a system based on K($\mathtt{a}$,$\mathtt{b}$)+Per plus any combination of these axioms is characterized by the Kripke models $(W, R_{\mathtt{a}}, R_{\mathtt{b}})$

whose accessibility relation satisfies the corresponding properties. Decision procedures for systems based on K(a,b) plus *non-interaction* axioms are treated in [5], and those based on K(a,b) plus *interaction* axioms of the form $\Diamond_{\mathtt{b}} p \to \Diamond_{\mathtt{a}} p$ in [7]. The system K(a,b)+Per+$Conf_{ab}$ is also known as the weakest product modal logic K× K investigated in [14].

3.2 Preliminaries and notations

Definition 3.1 A labelled ρ-RDAG is a triple $(\mathcal{N}, \Sigma, \Phi)$ as in the monomodal case but where Σ is partitionned into $\Sigma^{\mathtt{a}}$ and $\Sigma^{\mathtt{b}}$.

3.3 Rules

Some rules are the same as for the monomodal case but with indexes, e.g.

- Rule $(\Diamond_{\mathtt{a}})$: $\Diamond_{\mathtt{a}} A, S \bullet \Longrightarrow \Diamond_{\mathtt{a}} A, S \bullet\!\xrightarrow{\ \mathtt{a}\ }\!\bullet A$
- Rule $(\Box_{\mathtt{a}})$: $\Box_{\mathtt{a}} A, S \bullet\!\xrightarrow{\ \mathtt{a}\ }\!\bullet S1 \Longrightarrow \Box_{\mathtt{a}} A, S \bullet\!\xrightarrow{\ \mathtt{a}\ }\!\bullet A, S1$
- Rule $(4_{\mathtt{a}})$: $\Box_{\mathtt{a}} A, S \bullet\!\xrightarrow{\ \mathtt{a}\ }\!\bullet S1 \Longrightarrow \Box_{\mathtt{a}} A, S \bullet\!\xrightarrow{\ \mathtt{a}\ }\!\bullet \Box_{\mathtt{a}} A, S1$
- Rule (D_a): $S \bullet \Longrightarrow S \bullet\!\xrightarrow{\ \mathtt{a}\ }\!\bullet \emptyset$

- And as well for rules $(\Diamond_{\mathtt{b}})$, $(\Box_{\mathtt{b}})$, $(4_{\mathtt{b}})$ and (D_b).

In addition, in order to handle the permutation properties, we will use the following structural rules:

- Rule Per_{ba}:
$$\begin{tikzcd}[column sep=small] & S1 \arrow[dr, "\mathtt{a}"] & \\ S0 \arrow[ur, "\mathtt{b}"] & & S2 \end{tikzcd} \Longrightarrow \begin{tikzcd}[column sep=small] & S1 \arrow[dr, "\mathtt{a}"] & \\ S0 \arrow[ur, "\mathtt{b}"] \arrow[dr, "\mathtt{a}"'] & & S2 \\ & \emptyset \arrow[ur, "\mathtt{b}"'] & \end{tikzcd}$$

- Rule Per_{ab}:
$$\begin{tikzcd}[column sep=small] & S1 \arrow[dr, "\mathtt{b}"] & \\ S0 \arrow[ur, "\mathtt{a}"] & & S2 \end{tikzcd} \Longrightarrow \begin{tikzcd}[column sep=small] & S1 \arrow[dr, "\mathtt{b}"] & \\ S0 \arrow[ur, "\mathtt{a}"] \arrow[dr, "\mathtt{b}"'] & & S2 \\ & \emptyset \arrow[ur, "\mathtt{a}"'] & \end{tikzcd}$$

Rule $Conf_{ab}$ is the ab-version of rule Cf.

In order to define a tableau calculus for a logical system, we must associate a set of rules with it. Tableaux calculi we are going to define contain:

- Classical rules
- Rules $\Diamond_{\mathtt{a}}$ and $\Diamond_{\mathtt{b}}$
- Rule $\Box_{\mathtt{a}}$ and $\Box_{\mathtt{b}}$
- Rules corresponding to the axioms of the system.

3.4 Complete tableaux

Of course, in these cases, our Relational Closure lemma (1.5) still holds, and our Fundamental lemma (1.12) holds as long as our Box lemma (1.11) holds too. Thus completeness of tableaux calculi will hold if both the Box lemma and the Structural lemma (1.7) hold. It is straightforward with these properties (as well as for symmetry and/or euclideanness) to check that the Box lemma indeed holds:

Lemma 3.2 Let $\Upsilon = (\mathcal{N}, \Sigma^{\mathsf{a}}, \Sigma^{\mathsf{b}}, \Phi)$ be a $(\rho_1 \cup \rho_2)$-tableau with root r. Let x, y be such that $(x, y) \in (\Sigma^{\mathsf{a}})^{\rho_1}$ and $\Box_{\mathsf{a}} A \in x$; then $A \in y$. The same holds for $(x, y) \in (\Sigma^{\mathsf{b}})^{\rho_1}$ and $\Box_{\mathsf{b}} A \in x$.

Proof. This is trivial since in the case of the properties we consider, we have $(\Sigma^{\mathsf{a}} \cup \Sigma^{\mathsf{b}})^{\rho_1} = (\Sigma^{\mathsf{a}})^{\rho_1} \cup (\Sigma^{\mathsf{b}})^{\rho_1}$. Other difficulties would arise if one consider closure properties that involve both relations, e.g. euclideanness of $\Sigma^{\mathsf{a}} \cup \Sigma^{\mathsf{a}}$, symmetry of $\Sigma^{\mathsf{a}} \cap \Sigma^{\mathsf{b}}$, ...

∎

What about the Structural lemma (3)? It holds in fact as it is in the case of closure under reflexivity and transitivity (and the proof is the same), but needs some additionnal requirements to hold for closure under symmetry or euclideanness: for example, in the case of closure under symmetry, if $\Sigma = \Sigma^{\mathsf{a}} \cup \Sigma^{\mathsf{b}}$ satisfies $\Sigma^{\mathsf{b}} \circ \Sigma^{\mathsf{a}} = \Sigma^{\mathsf{a}} \circ \Sigma^{\mathsf{b}}$ then Σ^{ρ_1} does not and some structural rules need to be added !

Lemma 3.3 Let ρ_2 be a subset of properties of group 4, let ρ_1 be a subset of group 3 and let $\Sigma = \Sigma^{\mathsf{a}} \cup \Sigma^{\mathsf{b}}$ be a ρ_2-RGRAPH over a set $\mathcal{N}$ of nodes. Then Σ^{ρ_1} is also a ρ_2-RGRAPH and hence is a $(\rho_1 \cup \rho_2)$-RGRAPH.

Proof. We only treat one example: $\mathit{Conf}_{ab} \in \rho_2$ and $\mathit{Tr}_a \in \rho_1$.
By hypothesis, we have $\overline{\Sigma^{\mathsf{b}}} \circ \Sigma^{\mathsf{a}} = \Sigma^{\mathsf{a}} \circ \overline{\Sigma^{\mathsf{b}}}$, and we must prove that: $\overline{\Sigma^{\mathsf{b}+}} \circ \Sigma^{\mathsf{a}} \subseteq \Sigma^{\mathsf{a}} \circ \overline{\Sigma^{\mathsf{b}+}}$ since $\underline{\Sigma^{\mathit{Tr}_a}} = \Sigma^{\mathsf{b}+} \cup \Sigma^{\mathsf{a}}$.
But, $\overline{\Sigma^{\mathsf{b}+}} \circ \Sigma^{\mathsf{a}} = \overline{\Sigma^{\mathsf{b}}}^{+} \circ \Sigma^{\mathsf{a}} = \bigcup_{i \geq 1} \overline{\Sigma^{\mathsf{b}}}^{i} \circ \Sigma^{\mathsf{a}} \subseteq \bigcup_{i \geq 1} \Sigma^{\mathsf{a}} \circ \overline{\Sigma^{\mathsf{b}}}^{i} = \Sigma^{\mathsf{a}} \circ \overline{\Sigma^{\mathsf{b}+}}$ (by straightforward induction). The proof is similar for Per, and is the same as in the monomodal case for the structural properties D_a, De_a, C_a, and their b-versions. For other properties of group 3 (than Tr_a) the proof is the same (Tr_b) or simpler (Ref_a, Ref_b). ∎

Thus, for bimodal logics of permutation containing in addition any axiom listed in groups 3 and 4 above, we obtain sound and complete naive tableaux very easily (proof of soundness is straightforward).
About termination, as they are, naive tableaux terminates for systems not containing Tr_a nor Tr_b because of the decrease of the modal degree of formulas contained in nodes, but with probably very high complexity (it is conjectured in [19] that even the simple $\mathrm{K}(\mathsf{a},\mathsf{b})+\mathit{Per}+\mathit{Conf}_{ab}$ may be of non-elementary complexity). At the moment, it is not known whether $\mathrm{K}(\mathsf{a},\mathsf{b})+\mathit{Per}+\mathit{Conf}_{ab}+\mathit{Tr}_a+\mathit{Tr}_b$ is decidable ([19]).

4 Lotrec: a generic theorem prover

The above generic approach has been a theoretical basis for the development of the generic modal theorem prover Lotrec at the Institut de Recherche en Informatique de Toulouse. It is described in [10] and may be downloaded at

http://www.irit.fr/ACTIVITES/LILaC/Lotrec.
It has been designed in order to answer the need for designing provers for new modal logics. In general, most implementors of provers for modal logics put the emphasis on performance, and thus have restricted their prover to few *fixed* logics, hacking logics and strategies in their systems.

The choice of one logic over another among possibly infinitely many modal logics is driven by modeling needs and computational constraints of one's applications. A logic about actions and time is likely to have different semantical and computational properties from a logic about database schemata. Even with the same logic, different search strategies may be needed for different applications.

To answer the needs of users wishing to *experiment and model with different logics or strategies* there is a need for a *generic theorem prover for modal logics* playing the same role as Isabelle [23] or PVS [22] for higher order logics, while being less complex. If the user is not the same person as the programmer of the prover, one needs flexibility and portability of the implementation, high-level languages for tableau rules and strategy definition and user-friendly interfaces.

Lotrec is such a generic tableau prover. It aims at covering all logics having possible worlds semantics. Lotrec has been implemented by D. Fauthoux [11]. It is written in Java. All entities are modeled as objects, in particular the tableaux, the nodes and links of a tableau, the tableau rules, and the strategy. Within such an object-based programming language, Lotrec raises Java's event-based architecture to a declarative approach, in order to be able to manipulate and manage the computation in an easy but strict way.

In Lotrec, tableaux are generalized to *graphs* in order to enable complex modal logics such as that of confluence, or multimodal logics with complex interactions between modalities.

References

[1] M. Baldoni. Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension. PhD thesis, Univ. Torino, Italy, 1998.

[2] M. Baldoni, L. Giordano, and A. Martelli. A Tableau Calculus for Multimodal Logics and some (Un)Decidability Results. In H. de Swart (Ed) *T*ABLEAUX'98, LNAI 1397,Springer-Verlag, 1998.

[3] M. A. Castilho, L. Fariñas del Cerro, O. Gasquet, and A. Herzig. Modal tableaux with propagation rules and structural rules. *F*undamenta Informaticae, 32(3/4):281-297, 1997.

[4] L. Catach, Les logiques multi-modales. Ph.D. thesis, Université Paris VI, France, 1989

[5] L. Catach, TABLEAUX: A General Theorem Prover for Modal Logics. *J*ournal of Automated Reasoning, 7:489-510, 1991.

[6] G. De Giacomo, F. Massacci. Tableau and Algorithm for Propositional Dynamic Logic with Converse. In M. A. McRobbie, J. K. Slaney editor, *Proc. CADE-13, LNAI 1104, Springer, 1996.*

[7] S. Demri Complexity of simple dependent bimodal logics. In *Proc. Int. Conf. Automated Reasoning with Analytic Tableaux and Related Methods* (TABLEAUX'2000), LNCS 1847, Springer, 2000.

[8] H. C. M. de Swart. Gentzen-type systems for C, K and several extensions of C and K; constructive completeness proofs and effective decision procedure for these systems. *Logique et Analyse*, 1980.

[9] L. Fariñas del Cerro, and O. Gasquet. Tableaux Based Decision Procedures for Modal Logics of Confluence and Density. *F*undamenta Informaticae, 41(1):1-17, 2000.

[10] L. Fariñas del Cerro, D. Fauthoux, O. Gasquet, A. Herzig, D. Longin and F. Massacci. Lotrec:

The Generic Tableau Prover for Modal and Description Logics. In: International Joint Conference on Automated Reasoning, System Description, Siena, Italy, 2001.

[11] D. Fauthoux. Lotrec, un outil javanais de traitement formel sur les graphes. Tech. rep., IRIT, June 2000. Master Thesis (rapport de D.E.A).

[12] M. Fitting. *Proof methods for modal and intuitionistic logics*. D. Reidel, Dordrecht, 1983.

[13] M. Fitting. Basic modal logic. In D. Gabbay et al., editor, *Handbook of Logic in Artificial Intelligence and Logic Programming: Logical Foundations*, vol. 1, Oxford University Press, 1993.

[14] D. M. Gabbay, V. Shehtman. Products of modal logics, Part 1. Logic Journal of the IGPL, 6:73-146, 1998.

[15] R. Goré. Methods for Modal and Temporal Logics. In D. Gabbay et al., editor, *Handbook of Tableau Methods*, Kluwer Dordrecht, 1998.

[16] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic (24)*, 1959.

[17] S. Kripke. Semantical analysis of modal logic I: Normal modal propositional calculi. *Zeitschrift fr Mathematische Logik und Grundlagen der Mathematik (9)*, 1963.

[18] R. Ladner. The computational complexity of provability in systems of modal logic. *SIAM Journal on Computing*, 6:466-480, 1977.

[19] M. Marx,S. Mikulas An elementary constrution for a non-elementary procedure Studia Logica,72(2), 2002 (to appear).

[20] F. Massacci. Strongly analytic tableaux for normal modal logics. In Alan Bundy editor, *Proc. CADE-12, LNAI 814, Springer, 1994*.

[21] F. Massacci. Single Step Tableaux for Modal Logics: Computational properties, complexity and methodology. *Journal of Automated Reasoning*, 24(3): 319-364, April 2000.

[22] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In *Proc. of CADE'92*, LNAI, pp. 748–752, 1992.

[23] L. C. Paulson. Isabelle: A Generic Theorem Prover. LNCS, Springer-Verlag, 1994.

[24] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logics. In S.Kanger editor, *Proc. 3rd Scandinavian Logic Symposium 1973, Studies in Logic 82* (1975), North-Holland, 1973

A Some properties about binary relations

In these appendix, we will make use of *relations* (binary relations) and *rooted relations* instead of graphs and rooted graphs. The set of all relations over a given set will be denoted by $\mathcal{R}$ while that of rooted relations will be denoted by $\mathcal{RR}$.

Definition A.1 Let R be a relation over a set $\mathcal{N}$: $R(x)$ will denote the set of nodes accessible from x by R: $R(x) = \{y \in \mathcal{N} : (x, y) \in R\}$, $\overline{R}$ will denote its inverse, R^+ will denote its transitive closure and R^* its transitive and reflexive closure. Also, R^n will denote the pairs (x, y) such that there is a path of length n between x and y. The diagonal relation: $\{(x, x) : x \in \mathcal{N}\}$ will be denoted by I and also by R^0. The composition of two relations R and S (which is defined as $\{(x, y) : \exists z(x, z) \in R$ and $(z, y) \in S\}$ will be denoted by $(R \circ S)$. The total relation $\mathcal{N}^2$ is denoted by $\mathcal{U}$. The empty relation is denoted by $\mathcal{O}$.

Property A.2 (About $\mathcal{R}$) Let R, S and $T \in \mathcal{R}$, let ρ be a subset of group 1:

1. $R^+ = \bigcup_{i \geq 1} R^i$

2. $\overline{\overline{R}} = R$

3. $\overline{R \cup S} = \overline{R} \cup \overline{S}$

4. $\overline{R \circ S} = \overline{S} \circ \overline{R}$

5. $\overline{R^n} = \overline{R}^n$ (for $n \geq 0$)

6. $\overline{R^+} = \overline{R}^+$

7. $\overline{R^*} = \overline{R}^*$

8. $(R \cup I)^+ = R^*$

9. $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$

10. $T \circ (R \cup S) = (T \circ R) \cup (T \circ S)$

11. $I^+ = I^* = \overline{I} = I$

12. If $R \neq \mathcal{O}$ then $R \circ \overline{R} \neq \mathcal{O}$

13. If $R \neq \mathcal{O}$ then $\mathcal{U} \circ R \circ \mathcal{U} = \mathcal{U}$

14. $(R^n)^+ \subseteq (R^+)^n$ (for $n \geq 0$)

15. $R \subseteq R^\rho$ (growth)

16. $R \subseteq S \Rightarrow R^\rho \subseteq S^\rho$ (monotonicity)

17. If $P \in \rho$ then $(R^\rho)^P = R^\rho$ (idempotence); and of course: $(R^\rho)^\rho = R^\rho$

18. R is reflexive iff $I \subseteq R$

19. R is symmetrical iff $\overline{R} \subseteq R$

20. R is transitive iff $R^2 \subseteq R$

21. R is euclidean iff $(\overline{R} \circ R) \subseteq R$, or iff $(\overline{R} \circ R) \subseteq \overline{R}$

22. R is dense iff $R \subseteq R^2$

23. R is serial iff $I \subseteq (R \circ \overline{R})$

24. R is confluent iff $(\overline{R} \circ R) \subseteq (R \circ \overline{R})$

25. R is rooted iff $(\overline{R}^* \circ R^*) = \mathcal{U}$

26. R is connected iff $(\overline{R} \cup R)^* = \mathcal{U}$, and rooted implies connected.

Proof. All are well-known or obvious properties except maybe 14 for which it suffices to prove that $(R^2)^+ \subseteq (R^+)^2$:
$(R^+)^2 = (\bigcup_{i\geq 1} R^i)^2 = (\bigcup_{i\geq 1} R^i) \circ (\bigcup_{i\geq 1} R^i) = \bigcup_{i\geq 1} \bigcup_{j\geq 1} (R^i \circ R^j)$
$= \bigcup_{i\geq 1} \bigcup_{j\geq 1} (R^{i+j}) = \bigcup_{i\geq 2} (R^i)$,
and, $(R^2)^+ = \bigcup_{i\geq 1} R^{2i} \subseteq \bigcup_{i\geq 2} (R^i)$. ∎

Property A.3 (About $\mathcal{RR}$) Let $R \in \mathcal{RR}$:
1. Let ρ be a subset of group 1 then R^ρ is also in $\mathcal{RR}$.

2. $(\overline{R}^+ \circ R^+ \circ \overline{R}^+ \circ R^+) = (\overline{R}^+ \circ R^+)$

3. If $(\overline{R}\circ R) \subseteq (R\circ\overline{R})$ then $(\overline{R}^{+}\circ R^{+}) \subseteq (R^{+}\circ\overline{R}^{+})$

4. $(\overline{R}^{*}\circ R^{+})^{+} = (\overline{R}^{*}\circ R^{+})$

PROOF. 1. Trivial since the root r of R is still a root in R^{ρ}.

2. If $R = \mathcal{O}$ then 2 holds trivially, else we have:
$(\overline{R}^{+}\circ R^{+}\circ\overline{R}^{+}\circ R^{+}) = (\overline{R}\circ\overline{R}^{*}\circ R^{*}\circ R\circ\overline{R}\circ\overline{R}^{*}\circ R^{*}\circ R) = (\overline{R}\circ\mathcal{U}\circ R\circ\overline{R}\circ\mathcal{U}\circ R) = (\overline{R}\circ\mathcal{U}\circ R)$; (since $R \neq \mathcal{O} \Rightarrow R\circ\overline{R} \neq \mathcal{O}) = (\overline{R}^{+}\circ R^{+})$

3. We show that $(\overline{R}\circ R) \subseteq (R\circ\overline{R}) \Rightarrow \forall k, l \geq 1\colon (\overline{R}^{k}\circ R^{l}) \subseteq (R^{l}\circ\overline{R}^{k})$ by induction on $k + l$.
Induction base:
if $k + l = 2$, the property holds by hypothesis.
Induction step:
if $k > 1$ then $(\overline{R}^{k}\circ R^{l}) = (\overline{R}\circ\overline{R}^{k-1}\circ R^{l}) \subseteq (\overline{R}\circ R^{l}\circ\overline{R}^{k-1})$(by IH) $\subseteq (R^{l}\circ\overline{R}\circ\overline{R}^{k-1})$ (by IH) $\subseteq (R^{l}\circ\overline{R}^{k})$.
else if $k = 1$ and $l > 1$ then
$(\overline{R}^{k}\circ R^{l}) = (\overline{R}^{k}\circ R\circ R^{l-1}) \subseteq (R\circ\overline{R}^{k}\circ R^{l-1})$(by IH)
$\subseteq (R\circ R^{l-1}\circ\overline{R}^{k})$(by IH) $\subseteq (R^{l}\circ\overline{R}^{k})$.

4. It suffices to show that $(\overline{R}^{*}\circ R^{+})^{2} = (\overline{R}^{*}\circ R^{+})$:
If $R = \mathcal{O}$ then it holds trivially, else we have: $(\overline{R}^{*}\circ R^{+})^{2} = (\overline{R}^{*}\circ R^{*}\circ R)^{2} = (\mathcal{U}\circ R)^{2} = (\mathcal{U}\circ R\circ\mathcal{U}\circ R) = (\mathcal{U}\circ R) = (\overline{R}^{*}\circ R^{+})$.

∎

B Properties of closure operations

Lemma 1.5 (Relational Closure Lemma)
This lemma stated on page 56 is a straightforward consequence of the lemmas B.1 and B.3 below.

Lemma B.1 (Closure under one property) Let $R \in \mathcal{RR}$:
1. $R^{Ref} = R \cup I$

2. $R^{Sym} = R \cup \overline{R}$

3. $R^{Tr} = R^+$

4. $R^{Eucl} = R \cup (\overline{R}^+ \circ R^+)$

Proof. Only 4. is not obvious and well-known (it uses the fact that R has a root). We prove it by showing:
i) $R \cup (\overline{R}^+ \circ R^+) \subseteq R^{Eucl}$

ii) $R \cup (\overline{R}^+ \circ R^+)$ is euclidean
and we will get the conclusion since R^{Eucl} is the least superset of R being euclidean and, as such, it contains any other euclidean superset of R.
i) First we prove by induction on $i+j$ that $\forall i,j\colon (\overline{R}^i \circ R^j) \subseteq (\overline{R^{Eucl}} \circ R^{Eucl})$.
Induction base:
$i+j=2$, i.e. $i=j=1$: $(\overline{R} \circ R) \subseteq (\overline{R^{Eucl}} \circ R^{Eucl})$ (since $R \subseteq R^{Eucl}$ and hence $\overline{R} \subseteq \overline{R^{Eucl}}$).
Induction step:
if $j > 1$ then
$(\overline{R}^i \circ R^j) = (\overline{R}^i \circ R^{j-1} \circ R) \subseteq (\overline{R^{Eucl}} \circ R^{Eucl} \circ R)$ (by IH) $\subseteq (\overline{R^{Eucl}} \circ R)$
$\subseteq (\overline{R^{Eucl}} \circ R^{Eucl})$ (by growth).
else
if $j = 1$ and $i > 1$ then
$(\overline{R}^i \circ R^j) = (\overline{R} \circ \overline{R}^{i-1} \circ R^j) \subseteq (\overline{R} \circ \overline{R^{Eucl}} \circ R^{Eucl})$ (by IH) $\subseteq (\overline{R} \circ R^{Eucl})$
$\subseteq (\overline{R^{Eucl}} \circ R^{Eucl})$.

Now, since $(\overline{R}^+ \circ R^+) = (\bigcup_{i \geq 1} \overline{R}^i) \circ (\bigcup_{j \geq 1} R^j) = \bigcup_{i,j \geq 1} (\overline{R}^i \circ R^j)$
$\subseteq \bigcup_{i,j \geq 1} (\overline{R^{Eucl}} \circ R^{Eucl}) = (\overline{R^{Eucl}} \circ R^{Eucl}) \subseteq R^{Eucl}$:
we obtain $R \cup (\overline{R}^+ \circ R^+) \subseteq R \cup R^{Eucl} \subseteq R^{Eucl}$.

ii) We show that indeed $R \cup (\overline{R}^+ \circ R^+)$ is euclidean by using lemma A.2:
$\overline{(R \cup (\overline{R}^+ \circ R^+))} \circ (R \cup (\overline{R}^+ \circ R^+)) = (\overline{R} \cup \overline{(\overline{R}^+ \circ R^+)}) \circ (R \cup (\overline{R}^+ \circ R^+))$
$= (\overline{R} \cup (\overline{R^+} \circ \overline{\overline{R}^+})) \circ (R \cup (\overline{R}^+ \circ R^+)) = (\overline{R} \cup (\overline{R}^+ \circ R^+)) \circ (R \cup (\overline{R}^+ \circ R^+))$
$= (\overline{R} \circ R) \cup (\overline{R} \circ \overline{R}^+ \circ R^+) \cup (\overline{R}^+ \circ R^+ \circ R) \cup (\overline{R}^+ \circ R^+ \circ \overline{R}^+ \circ R^+)$
$\subseteq (\overline{R}^+ \circ R^+) \cup (\overline{R}^+ \circ R^+ \circ \overline{R}^+ \circ R^+)$
(since $(\overline{R} \circ R)$, $(\overline{R} \circ \overline{R}^+ \circ R^+)$ and $(\overline{R}^+ \circ R^+ \circ R) \subseteq (\overline{R}^+ \circ R^+)$)
$\subseteq (\overline{R}^+ \circ R^+)$ (Abour $\mathcal{RR}$: 2) $\subseteq R \cup (\overline{R}^+ \circ R^+)$.

∎

Thanks to the previous lemma, we know how to compute the closure of an $\mathcal{RR}$ under one property of group 1, but how to do it for several properties ? The following lemma will provide us with a tool for this computation. It states that if some fix-point is reached by performing alternatively the closures under each of the properties of some subset ρ of group 1, then this fix-point *is* the closure under ρ. Before, we recall that if $\rho = \{P_1, \ldots, P_n\}$ is a set of properties, a relation S is said to be the ρ-closure of some relation R (i.e. $S = R^\rho$) if and only if S is the least relation containing R and closed under each P_i.

Lemma B.2 Let $\rho = \{P_1, \ldots, P_n\}$ be a subset of group 1, and $R \in \mathcal{RR}$. Let $R_0 = R$ and $R_{i+1} = (\ldots(R_i^{P_1})\ldots)^{P_n}$; then if there exists m such that $R_{m+1} = R_m$ then $R_m = R^\rho$.

Proof. We have by growth:
$R_m \subseteq R_m^{P_1} \subseteq (R_m^{P_1})^{P_2} \subseteq \ldots \subseteq (\ldots((R_m^{P_1})^{P_2})\ldots)^{P_n} = R_{m+1}$. Now, since $R_m = R_{m+1}$ it comes: $R_m = R_m^{P_i}$, for $1 \leq i \leq n$ (otherwise growth would be falsified) and thus R_m is closed under each P_i $(1 \leq i \leq n)$. Hence R_m is closed under ρ. To conclude, take note that R^ρ is the least superset of R closed under ρ and as such is contained in R_m which, in its turn, is contained in R^ρ since $R_0 \subseteq R^\rho$ (by growth) and $R_i \subseteq R^\rho \Rightarrow R_{i+1} \subseteq (\ldots(((R^\rho)^{P_1})^{P_2})\ldots)^{P_n} = R^\rho$ (by idempotence). ∎

Lemma B.3 (Closure under several properties) Let R be any $\mathcal{RR}$:
1. $R^{Ref,Sym} = R \cup \overline{R} \cup I$

2. $R^{Ref,Tr} = (R \cup I)^+$

3. $R^{Ref,Sym,Tr} = (R \cup \overline{R} \cup I)^+$

4. $R^{Sym,Tr} = (R \cup \overline{R})^+$

5. $R^{Tr,Eucl} = (\overline{R}^* \circ R^+)$

Due to lemma 1.6, the other cases reduce to one of the previous.

Proof. We indicate a closure by some property ρ by $\stackrel{\rho}{\Longrightarrow}$:
1. Case of $R^{Ref,Sym}$: $R \stackrel{Ref}{\Longrightarrow} R \cup I \stackrel{Sym}{\Longrightarrow} R \cup I \cup \overline{R \cup I} = R \cup \overline{R} \cup I \stackrel{Ref}{\Longrightarrow} R \cup \overline{R} \cup I$. A fix-point has been obtained.

2. Case of $R^{Ref,Tr}$: $R \stackrel{Ref}{\Longrightarrow} R \cup I \stackrel{Tr}{\Longrightarrow} = (R \cup I)^+ \stackrel{Ref}{\Longrightarrow} (R \cup I)^+ \cup I = (R \cup I)^+ = R^*$.

3. Case of $R^{Ref,Sym,Tr}$: $R \stackrel{Ref}{\Longrightarrow}$ cf. case 1 $\stackrel{Sym}{\Longrightarrow} R \cup \overline{R} \cup I \stackrel{Tr}{\Longrightarrow} (R \cup \overline{R} \cup I)^+$
$\stackrel{Ref}{\Longrightarrow} (R \cup \overline{R} \cup I)^+ \cup I = (R \cup \overline{R} \cup I)^+ = (R \cup \overline{R})^* = \mathcal{U} \stackrel{Sym}{\Longrightarrow} \mathcal{U} \cup \overline{\mathcal{U}} = \mathcal{U} = (\overline{R}^* \circ R^*)$.

4. Case of $R^{Sym,Tr}$: $R \stackrel{Sym}{\Longrightarrow} (R \cup \overline{R}) \stackrel{Tr}{\Longrightarrow} (R \cup \overline{R})^+ \stackrel{Sym}{\Longrightarrow} (R \cup \overline{R})^+ \cup \overline{(R \cup \overline{R})^+}$
$= (R \cup \overline{R})^+ \cup \overline{(R \cup \overline{R})}^+ = (R \cup \overline{R})^+ \cup (\overline{R} \cup R)^+ = (R \cup \overline{R})^+$.

5. Case of $R^{Tr,Eucl}$: $R \stackrel{Tr}{\Longrightarrow} R^+ \stackrel{Eucl}{\Longrightarrow} = R^+ \cup ((\overline{R^+})^+ \circ (R^+)^+) = R^+ \cup (\overline{R}^+ \circ R^+)$
$= (\overline{R}^* \circ R^+) \stackrel{Tr}{\Longrightarrow} (\overline{R}^* \circ R^+)^+ = (\overline{R}^* \circ R^+)$ (About $\mathcal{RR}$: 4).

∎

We need to prove now the stability of group 2 with respect to closure under several properties of group 1. We first prove the following lemma concerning this stability with respect to closure under one property of group 1, and then (lemma 1.7) shows that the same holds for several properties.

Lemma B.4 Let ρ_2 be a subset of group 2, ρ_1 a property of group 1, let $R \in \mathcal{RR}$ satisfying ρ_2 then R^{ρ_1} is in $\mathcal{RR}$ and satisfies ρ_2; hence it satisfies $\rho_1 \cup \rho_2$.

Proof. The proof is case-based:
Case $\rho_2 \ni Ser$: Immediate since $R \subseteq R^{\rho_1}$ (by monotonicity).

Case $\rho_2 \ni Dens$: we must show that $R^{\rho_1} \subseteq (R^{\rho_1})^2$:
– If $\rho_1 = Ref$: Trivial since reflexivity implies density;
– If $\rho_1 = Sym$:
$(R^{\rho_1})^2 = R^2 \cup (R \circ \overline{R}) \cup (\overline{R} \circ R) \cup \overline{R}^2 \supseteq R^2 \cup \overline{R}^2 \supseteq R \cup \overline{R}$,
hence $(R^{\rho_1})^2 \supseteq R \cup \overline{R} = R^{\rho_1}$;

– If $\rho_1 = Tr$:
$(R^{\rho_1})^2 = (R^+)^2 \supseteq (R^2)^+$ (About $\mathcal{R}$: 14) $\supseteq R^+ = R^{\rho_1}$;
– If $\rho_1 = Eucl$: Trivial since euclideanness implies density;

Case $\rho_2 \ni Conf$: we must show that $(\overline{R^{\rho_1}} \circ R^{\rho_1}) \subseteq (R^{\rho_1} \circ \overline{R^{\rho_1}})$:
– If $\rho_1 = Ref$:
$(\overline{R^{\rho_1}} \circ R^{\rho_1}) = (\overline{(R \cup I)} \circ (R \cup I)) = (\overline{R} \cup I) \circ (R \cup I) = (\overline{R} \circ R) \cup R \cup \overline{R} \cup I$
$\subseteq (R \circ \overline{R}) \cup R \cup \overline{R} \cup I$ (since R is confluent)
On the other hand, $(R^{\rho_1} \circ \overline{R^{\rho_1}}) = (R \circ \overline{R}) \cup R \cup \overline{R} \cup I$
hence $(\overline{R^{\rho_1}} \circ R^{\rho_1}) \subseteq (R^{\rho_1} \circ \overline{R^{\rho_1}})$;
– If $\rho_1 = Sym$: Trivial since symmetry implies confluence;
– If $\rho_1 = Tr$:
$(\overline{R^{\rho_1}} \circ R^{\rho_1}) = (\overline{R}^+ \circ R^+) \subseteq (R^+ \circ \overline{R}^+)$ (About $\mathcal{RR}$: 3) $= (R^{\rho_1} \circ \overline{R^{\rho_1}})$
– If $\rho_1 = Eucl$: Trivial since euclideanness implies confluence.

∎

Lemma B.5 Let ρ_2 be a subset of group 4 containing Per, let ρ_1 be a property of group 3 and let $\Sigma = \Sigma^{\mathsf{a}} \cup \Sigma^{\mathsf{b}}$ be a ρ_2-RGRAPH over a set $\mathcal{N}$ of nodes. Then Σ^{ρ_1} is also a ρ_2-RGRAPH and hence is a $(\rho_1 \cup \rho_2)$-RGRAPH.

PROOF. For properties among $\{Ser_a, Ser_b, Dens_a, Dens_b, Conf_a, Conf_b\}$ the proof is the same as in lemma B.4. It remains the case of Per(we only examine the cases of Ref_a and Tr_a):
- If $\rho_1 = Ref_a$: $(\Sigma^{\mathsf{a}})^{\rho_1} \circ (\Sigma^{\mathsf{b}})^{\rho_1} = (\Sigma^{\mathsf{a}} \cup I) \circ \Sigma^{\mathsf{b}} = (\Sigma^{\mathsf{a}} \circ \Sigma^{\mathsf{b}}) \cup (I \circ \Sigma^{\mathsf{b}}) = (\Sigma^{\mathsf{b}} \circ \Sigma^{\mathsf{a}}) \cup \Sigma^{\mathsf{b}} = (\Sigma^{\mathsf{b}})^{\rho_1} \circ (\Sigma^{\mathsf{a}})^{\rho_1}$;
- If $\rho_1 = Tr_a$: $(\Sigma^{\mathsf{a}})^{\rho_1} \circ (\Sigma^{\mathsf{b}})^{\rho_1} = (\Sigma^{\mathsf{a}})^+ \circ \Sigma^{\mathsf{b}} = (\bigcup_{n \geq 1} (\Sigma^{\mathsf{a}})^n) \circ \Sigma^{\mathsf{b}} =$
$\bigcup_{n \geq 1} ((\Sigma^{\mathsf{a}})^n) \circ \Sigma^{\mathsf{b}}) = \bigcup_{n \geq 1} (\Sigma^{\mathsf{b}} \circ (\Sigma^{\mathsf{a}})^n) = (\Sigma^{\mathsf{b}})^{\rho_1} \circ (\Sigma^{\mathsf{a}})^{\rho_1}$

∎

Lemma 1.7 (Structural Lemma)
Let ρ_2 be a subset of group 2, ρ_1 a subset of group 1, let $R \in \mathcal{RR}$ satisfying ρ_2 then R^{ρ_1} is in $\mathcal{RR}$ and satisfies ρ_2; hence it satisfies $\rho_1 \cup \rho_2$. This lemma extends straightforwardly to the bimodal case w.r.t groups 3 and 4.

PROOF. If ρ_1 is empty it is trivial. Now suppose (IH1): the lemma is true for some ρ_1; let P be a property of group 1; we must prove (C): the lemma holds for $\rho_1 \cup \{P\}$. But $R^{\rho_1 \cup \{P\}}$ is the fixpoint of the sequence $(((...((R^{\rho_1})^P)^{\rho_1})^P...)^{\rho_1})^P$ that will be denoted by $((R^{\rho_1})^P)_{n \text{ times}}$ where n is the number of closure operations to be done before to reach the fixpoint. If $n = 0$ we trivially have (C). Now suppose (IH2): (C) holds for N, we must prove that it holds for $N+1$. We have: $((R^{\rho_1})^P)_{N+1 \text{ times}} = ((((R^{\rho_1})^P)_{N \text{ times}})^{\rho_1})^P$. By (IH2), $((R^{\rho_1})^P)_{N \text{ times}}$ satisfies ρ_2, then by (IH1) $(((R^{\rho_1})^P)_{N \text{ times}})^{\rho_1}$ also satisfies ρ_2 and by lemma B.4 $((((R^{\rho_1})^P)_{N \text{ times}})^{\rho_1})^P = ((R^{\rho_1})^P)_{N+1 \text{ times}}$ satisfies ρ_2 too.

∎

An Open Research Problem: Strong Completeness of R. Kowalski's Connection Graph Proof Procedure

Jörg Siekmann *Universität des Saarlandes, Stuhlsatzenhausweg, D-66123 Saarbrücken, Germany. E-mail: siekmann@dfki.de*

Graham Wrightson *Department of Computer Science and Software Engineering, The University of Newcastle, NSW 2308, Australia. E-mail: graham@cs.newcastle.edu.au*

Abstract

The connection graph proof procedure (or clause graph resolution as it is more commonly called today) is a theorem proving technique due to Robert Kowalski. It is a negative test calculus (a refutation procedure) based on resolution. Due to an intricate deletion mechanism that generalises the well-known purity principle, it substantially refines the usual notions of resolution-based systems and leads to a largely reduced search space. The dynamic nature of the clause graph upon which this refutation procedure is based, poses novel meta-logical problems previously unencountered in logical deduction systems. Ever since its invention in 1975 the soundness, confluence and (strong) completeness of the procedure have been in doubt in spite of many partial results. This paper provides an introduction to the problem as well as an overview of the main results that have been obtained in the last twenty-five years.

1 Introduction to Clause Graph Resolution

We assume the reader to be familiar with the basic notions of resolution-based theorem proving (see, for example, Alan Robinson [39], Chang, C.-L. and Lee, R.C.-T. [16] or Don Loveland [29]). Clause graphs introduced a new ingenious development into the field, the central idea of which is the following: In standard resolution two resolvable literals must first be found in the set of sets of literals before a resolution step can be performed, where a set of literals represents a clause (i.e. a disjunction of these literals) and a statement to be refuted is represented as a set of clauses. Various techniques were developed to carry out this search. However, Robert Kowalski [27] proposed an enhancement to the basic data structure in order to make possible resolution steps explicit, which — as it turned out in subsequent years — not only simplified the search, but also introduced new and unexpected logical problems. This enhancement was gained by the use of so-called links between complementary literals, thus turning the *set notation* into a *graph-like structure*. The new approach allowed in particular for the removal of a link after the corresponding resolution step and a clause that contains a literal which is no longer connected by a link may be removed also (generalised purity principle). An important side effect was that this link removal had the potential to cause the disappearance of even more clauses from the current set of clauses (avalanche effect).

Although this effect could reduce the search space drastically it also had a significant impact on the underlying logical foundations. To quote Norbert Eisinger from his monograph on Kowalski's clause graphs [21]:

> "Let S and S' be the current set of formulae before and after a deduction step $S \vdash S'$. A step of a classical calculus and a resolution step both simply add a formula following from S. Thus, each interpreted as the conjunction of its members, S and S' are always equivalent. For clause graph resolution, however, S may contain formulae missing in S', and the removed formulae are not necessarily consequences of those still present in S'. While this does not affect the forward implication, S does in general no longer ensue from S'. In other words, it is possible for S' to possess more models than S. But, when S is unsatisfiable, so must be S', i.e. S' must not have more models than S, if soundness, unsatisfiability and hence refutability, is to be preserved."

This basic problem underlying all investigations of the properties of the clause graph procedure will be made more explicit in the following.

2 Clause Graph Resolution: The Problem

The standard resolution principle, called *set resolution* in the following, assumes the axioms and the negated theorem to be represented as a *set of clauses*. In contrast, the clause graph proof procedure represents the initial set of clauses as a *graph* by drawing a link between pairs of literal occurrences to denote that some relation holds between these two literals. If this relation is "*complementarity*" (it may denote other relations as well, see e.g. Christoph Walther [52], but this is the standard case and the basic point of interest in this paper) of the two literals, i.e. resolvability of the respective clauses, then an initial clause graph for the set

$$\begin{array}{rcl} S & = & \{\{ \quad -P(z,c,z), -P(z,d,z)\}, \{P(a,x,a), -P(a,b,c)\}, \\ & & \quad \{P(a,w,c), P(w,y,w)\}, \{P(u,d,u), -P(b,u,d), P(u,b,b)\}, \\ & & \quad \{-P(a,b,b)\}, \{-P(c,b,c), P(v,a,d), P(a,v,b)\}\} \end{array}$$

is the graph in Figure 1. Here P is a ternary predicate symbol, letters from the beginning of the alphabet $a, b, c, \ldots$ denote constants, letters from the end of the alphabet $x, y, z, v, \ldots$ denote variables and $-P(\ldots)$ denotes the negation of $P(\ldots)$.

Example 2.1

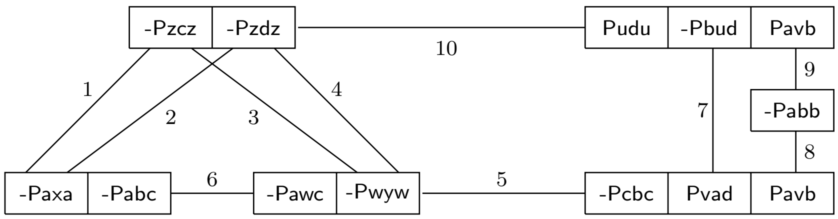


FIG. 1.

An appropriate most general unifier is associated with each link (not shown in the example of Figure 1). We use the now standard notation that adjacent boxes denote a clause, i.e. the conjunction of the literals in the boxes.

So far such a clause graph is just a data structure without commitment to a particular proof procedure and in fact there have been many proposals to base an automated deduction procedure on some graph-like notion (e.g. Andrews [2], Andrews [3], Bibel [7], Bibel [8], Chang and Slagle [17], Kowalski [27], Shostak [40, 41], Sickel [42], Stickel [51], Yates and Raphael and Hart [58], Omodeo [37], Yarmush [57], Murray and Rosenthal [31, 32]).

Kowalski's procedure uses a graph-like data structure as well, but its impact is more fundamental since it operates now as follows: suppose we want to perform the resolution step represented by link 6 in Figure 1 based on the unifier $\sigma = \{w \to b\}$. Renaming the variables appropriately we obtain the resolvent $\{P(a, x', a), P(b, y', b)\}$ which is inserted into the graph and if now all additional links are set this yields the graph:

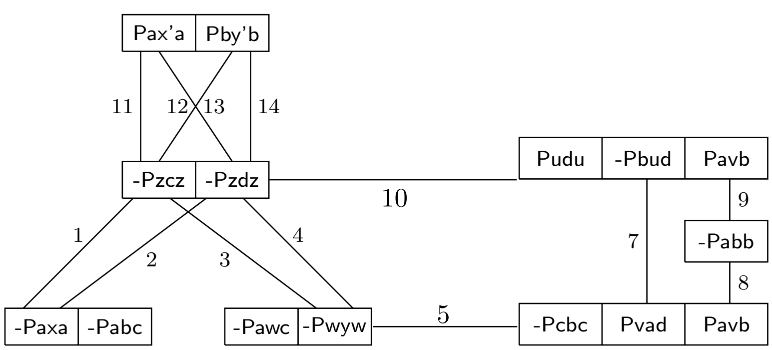


FIG. 2.

Now there are three essential operations:

1. The new links don't have to be recomputed by comparing every pair of literals again for complementarity, but this information can instead be inherited from the given link structure.

2. The link resolved upon is deleted to mark the fact that this resolution step has already been performed,

3. Clauses that contain a literal with no link connecting it to the rest of the graph may be deleted (generalised purity principle).

While the first point is the essential ingredient for the computational attractiveness of the clause graph procedure, the second and third points show the ambivalence between gross logical and computational advantages versus severe and novel theoretical problems. Let us turn to the above example again. After resolution upon link 6 we obtain the graph in Figure 2 above. Now since link 6 has been resolved upon we have it deleted it according to rule (2). But now the two literals involved become pure and hence the two clauses can be deleted as well leading to the following graph:

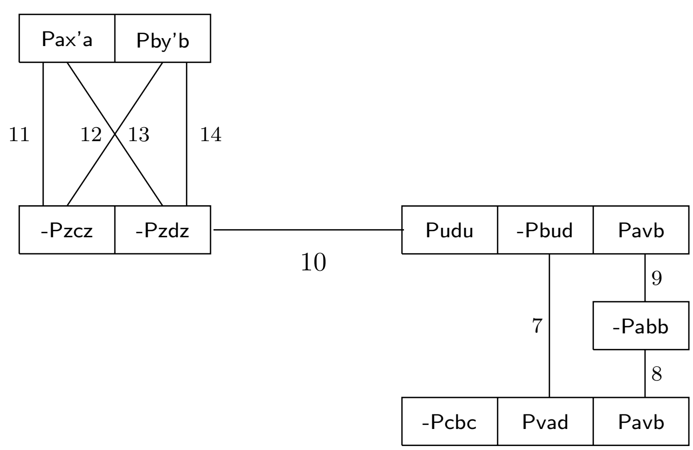


FIG. 3.

But now the literal $-P(c, b, c)$ in the bottom clause becomes pure as well and hence we have the graph:

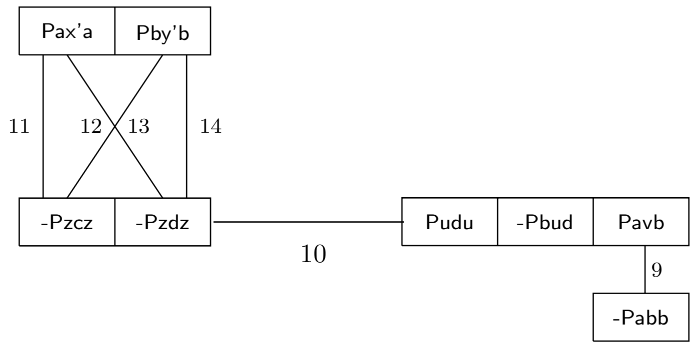


FIG. 4.

This removal causes the only literal $-P(a, b, b)$ in the bottom clause to become pure and hence, after a single resolution step followed by all these purity deletions, we arrive at the final graph:

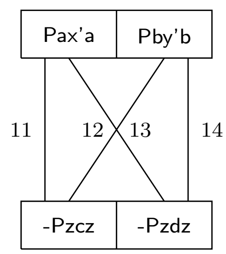


FIG. 5.

It is this strong feature that reduces redundancy in the complementary set of clauses, that marks the fascination for this proof procedure (see Ohlbach [33, 34], Bläsius [11, 12], Eisinger *et al.* [23], Ohlbach and Siekmann [36], Bläsius *et al.* [14], Eisinger [19], Eisinger, Siekmann and Unvericht [22], Ohlbach [35], Ramesh *et al.* [38], Murray and Rosenthal [32], Siekmann and Wrightson [45]). It can sometimes even reduce the initial redundant set to its essential contradictory subset (subgraph). But this also marks its problematical theoretical status: *how do we know that we have not deleted too many clauses?* Skipping the details of an exact definition of the various inheritance mechanisms (see e.g. Eisinger [21] for details) the following example demonstrates the problem.

Suppose we have the refutable set $S = \{\{P(a), P(a)\}, \{-Pa\}\}$ and its initial graph as in Figure 6, where PUR means purity deletion and MER stands for merging two literals (Andrews [1]), whilst RES stands for resolution.

Example 2.2

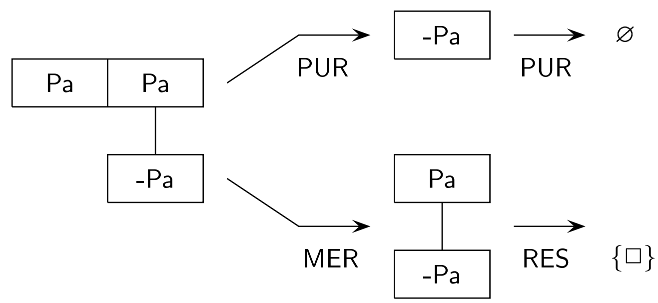


FIG. 6.

Thus in two steps we would arrive either at the empty set ∅, which stands for satisfiability, or in the lower derivation we arrive at the empty clause {□}, which stands for unsatisfiability.

This example would seem to show that the procedure:

(i) is not confluent, as defined below

(ii) is not sound (correct), and

(iii) is not refutation complete (at least not in the strong sense as defined below),

and hence would be useless for all practical purposes.

But here we can spot the flaw immediately: the process did not start with the full initial graph, where all possible links are set. If, instead, all possible links are drawn in the initial graph, the example in Figure 6 fails to be a counterexample. On the other hand, after a few initial steps we always have a graph with some links deleted, for example because they have been resolved upon. So how can we be sure that the same disastrous phenomenon, as in the above example, will not occur again later on in the derivation?

These problems have been called the *confluence*, the *soundness* and the *(strong) completeness* problem of the clause graph procedure and it can be shown that for the original formulation of the procedure in Kowalski [27] (with full subsumption and tautology removal) all these three essential properties unfortunately do not hold in general. However, for suitable remedies (of subsumption and tautology removal) the first two properties hold, whereas the third property has been open ever since.

3 Properties and Results for the Clause Graph Proof Procedure

In order to capture the strange and novel properties of logical graphs let us fix the following notions: A *clause graph of a set of clauses* S consists of a set of nodes labelled by the literal occurrences in S and a set of links that connect complementary literals. There are various possibilities to make this notion precise (e.g. Siekmann and Stephan [43, 44], Brown [15], Eisinger [18] and [21], Bibel [5], Smolka [48, 49, 50] Bibel and Eder [10], Hähnle *et al.* [25], Murray and Rosenthal [31]).

Let INIT(S) be the full initial clause graph for S with all possible links set. This is called a full connection graph in Bibel and Eder [10], a total graph in Eisinger [21] and in Siekmann, Stephan [43] and a complete graph in Brown [15].

Definition 3.1 Clause graph resolution is called

refutation sound	if $\mathrm{INIT}(S) \xrightarrow{*} \{\Box\}$ then S is unsatisfiable;
refutation complete	if S is unsatisfiable then there exists a derivation $\mathrm{INIT}(S) \xrightarrow{*} \{\Box\}$;
refutation confluent	if S is unsatisfiable, and, if $\mathrm{INIT}(S) \xrightarrow{*} G_1$ and $\mathrm{INIT}(S) \xrightarrow{*} G_2$ then there exists $G_1 \xrightarrow{*} G'$ and $G_2 \xrightarrow{*} G'$ for some G';
affirmation sound	if $\mathrm{INIT}(S) \xrightarrow{*} \varnothing$ then S is satisfiable;
affirmation complete	if S is satisfiable then there exists a derivation $\mathrm{INIT}(S) \xrightarrow{*} \varnothing$;
affirmation confluent	if S is satisfiable, and, if $\mathrm{INIT}(S) \xrightarrow{*} G_1$ and $\mathrm{INIT}(S) \xrightarrow{*} G_2$ then there exists $G_1 \xrightarrow{*} G'$ and $G_2 \xrightarrow{*} G'$, for some G'.

The state of knowledge about the clause graph proof procedure at the end of the 1980's can be summarised by the following major theorems. There are some subtleties involved when subsumption and tautology removal are involved (see Eisinger [21] for a thorough exposition; the discovery of the problems with subsumption and tautology removal and an appropriate remedy for these problems is due to Wolfgang Bibel).

Theorem 3.2 (Bibel, Brown, Eisinger, Siekmann, Stephan) Clause graph resolution is refutation sound.

Theorem 3.3 (Bibel) Clause graph resolution is refutation complete.

Theorem 3.4 (Eisinger, Smolka, Siekmann, Stephan) Clause graph resolution is refutation confluent.

Theorem 3.5 (Eisinger) Clause graph resolution is affirmation sound.

Theorem 3.6 (Eisinger) Clause graph resolution is not affirmation confluent.

Theorem 3.7 (Smolka) For the unit refutable class, clause graph resolution with an unrestricted tautology rule is refutation complete, refutation confluent, affirmation sound, (and strongly complete).

The important notion of strong completeness is introduced below.

Theorem 3.8 (Eisinger) Clause graph resolution with an unrestricted tautology rule is refutation complete, but neither refutation confluent nor affirmation sound.

As important and essential as the above-mentioned results may be, they are not enough for the practical usefulness of the clause graph procedure: the principal requirement for a proof procedure is not only to know that there exists a refutation, but even more importantly that the procedure can actually find it after a finite number of steps. These two notions, called *refutation completeness* and *strong refutation completeness* in the following, essentially coincide for *set resolution* but unfortunately they do not do so for the *clause graph procedure.*

This can be demonstrated by the example, in Figure 7, where we start with the graph G_0 and derive G_1 from G_0 by resolution upon the link marked ☞. The last graph G_2 contains a subgraph that is isomorphic to the first, hence the corresponding inference steps can be repeated over and over again and the procedure will not terminate with the empty clause. Note that a refutation, i.e. the derivation of the empty clause, could have been obtained by resolving upon the leftmost link between P and $-P$.

Example 3.9 (adapted from Eisinger [21])

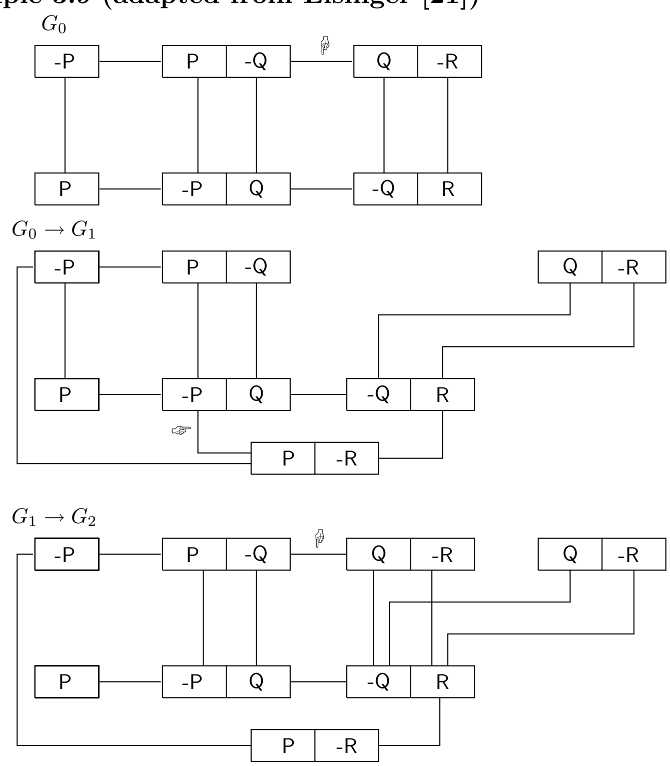


FIG. 7.

Examples of this nature gave rise to the strong completeness conjecture, which in spite of numerous attacks has remained an open problem now for over twenty years:

> *How can we ensure for an unsatisfiable graph that the derivation stops after finitely many steps with a graph that contains the empty clause?*

If this crucial property cannot be ascertained, the whole procedure would be rendered useless for all practical purposes, as we would have to backtrack to some earlier state in the derivation, and hence would have to store all intermediate graphs.

The theoretical problems and strange counter intuitive facts that arise from the (graphical) representation were first discovered by Jörg Siekmann and Werner Stephan and reported independently in Siekmann and Stephan [43, 44] and by Frank Brown in [15]. They suggested a remedy to the problem: the obvious flaw in the above example can be attributed to the fact that the proof procedure never selects the essential link for the refutation (the link between $-P$ and P).

This, of course, is a property which a control strategy should have, i.e. it should be fair in the sense that every link is eventually selected. However this is a subtle property in the dynamic context of the clause graph procedure as we shall see in the following.

Control Strategies

In order to capture the strange metalogical properties of the clause graph procedure, Siekmann and Stephan [43, 44] introduced two essential notions in order to capture the above-mentioned awkward phenomenon. These two notions have been the essence of all subsequent investigations:

(i) the notion of a *kernel*. This is now sometimes called the minimal refutable subgraph of a graph, e.g. in Bibel and Eder [10];

(ii) several notions of *covering*, called fairness in Bibel and Eder [10], exhaustiveness in Brown [15], fairness-one and fairness-two in Eisinger [21] and covering-one, two and three in Siekmann and Stephan [43].

Let us have a look at these notions in turn, using the more recent and advanced notation of Eisinger [21].

Why is it not enough to simply prove refutation completeness as in the case of clause set resolution? Ordinary refutation completeness ensures that if the initial set of clauses is unsatisfiable, then there exists a refutation, i.e. a finite derivation of the empty clause. Of course, there is a control strategy for which this would be sufficient for clause graph resolution as well, namely an exhaustive enumeration of all possible graphs, as in Figure 8, where we assume that the initial graph G_0 has n links. However such a strategy is computationally infeasible and far too expensive and would make the whole approach useless.

We know by Theorem 3.3 that the clause graph procedure is refutation complete, i.e. that there exists a subgraph from which the derivation can be obtained. Could we not use this information from a potential derivation we know to exist in order to guide the procedure in general?

Many strategies for clause graphs are in fact based on this very idea (Andrews [3], Antoniou and Ohlbach [4], Bibel [6, 8], Chang and Slagle [17], Sickel [42]). However,

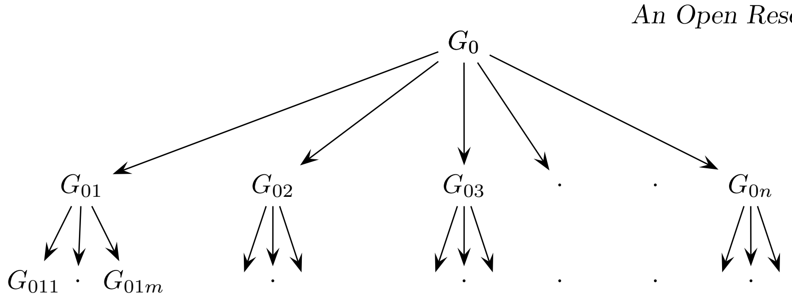


FIG. 8.

in general, finding the appropriate subgraph essentially amounts to finding a proof in the first place and we might as well use a standard resolution-based proof procedure to find the derivation and then use this information to guide the clause graph procedure.

So let us just assume in the abstract that every full (i.e. a graph where every possible link is set) and unsatisfiable graph contains a subgraph, called a *kernel* (the shaded area in Figure 9), from which an actual refutation can be found in a finite number of steps.

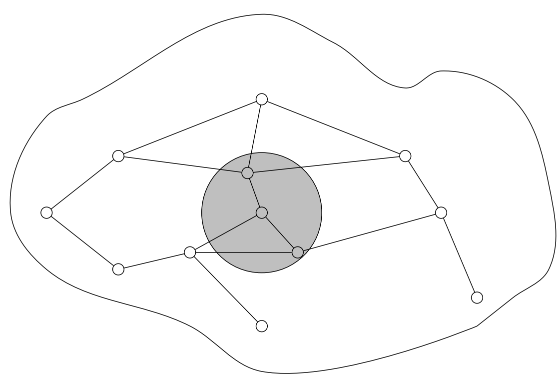


FIG. 9.

We know from Theorem 3.3 above and from the results in Siekmann and Stephan [43, 44] that every resolution step upon a link within the kernel eventually leads to the empty clause and thus to the desired refutation. If we can ensure that:

1. resolution steps involving links outside of the kernel do not destroy the kernel, and
2. every link in the kernel is eventually selected,

then we are done. This has been the line of attack ever since. Unfortunately the second condition turned out to be more subtle and rather difficult to establish. So far no satisfactory solution to this problem has been found.

So let us look at these concepts a little closer.

Definition 3.10 A *filter* for an inference system is a unary predicate F on the set of finite sequences of states. The notation $S_0 \xrightarrow{*} S_n$ *with* F stands for a derivation $S_0 \xrightarrow{*} S_n$ where $\mathtt{F}(S_0 \ldots S_n)$ holds. For an infinite derivation, $S_0 \to \ldots \to S_n \to \ldots$ *with* F means that $\mathtt{F}(S_0 \ldots S_n \ldots)$ holds for each n.

This notion is due to Gert Smolka in [49] and Norbert Eisinger in [21] and it is now used in several monographs on deduction systems (see e.g. K. Bläsius and H. J. Bürckert [13]). Typical examples for a filter are the usual restriction and ordering strategies in automated theorem proving, such as set-of-support by Wos and Robinson and Carson [54], linear refutation by Loveland [28], merge resolution by Andrews [1], unit resolution by Wos [53], or see Kowalski [26].

Definition 3.11 A filter F for clause graph resolution is called

refutation sound:	$\mathrm{INIT}(S) \xrightarrow{*} \{\square\}$ *with* F *then* S is unsatisfiable;
refutation complete:	if S is unsatisfiable then there exists $\mathrm{INIT}(S) \xrightarrow{*} \{\square\}$ *with* F;
refutation confluent:	Let S be unsatisfiable, For $\mathrm{INIT}(S) \xrightarrow{*} G_1$ *with* F and $\mathrm{INIT}(S) \xrightarrow{*} G_2$ *with* F then there exists $G_1 \xrightarrow{*} G'$ *with* F and $G_2 \xrightarrow{*} G'$ *with* F, for some G';
strong refutation completeness:	for an unsatisfiable S there does not exist an infinite derivation $\mathrm{INIT}(S) \to G_1 \to G_1 \to \ldots \to G_n \to \ldots$ *with* F.

Note that $\to$ *with* F need not be transitive, hence the special form of confluence, also note that the procedure terminates with $\{\square\}$ or with $\varnothing$.

The most important and still open question is now: can we find a general property for a filter that turns the clause graph proof procedure into a strongly complete system? Obviously the filter has to make sure that every link (in particular every link in some fixed kernel) is eventually selected for resolution and not infinitely postponed.

Definition 3.12 A filter F for clause graph resolution is called *covering*, if the following holds: Let G_0 be an initial graph, let $G_0 \xrightarrow{*} G_n$ *with* F be a derivation, and let λ be a link in G_n. Then there is a finite number $n(\lambda)$, such that for any derivation $G_0 \xrightarrow{*} G_n \xrightarrow{*} G$ *with* F extending the given one by at least $n(\lambda)$ steps, λ is not in G.

This is the weakest notion, called "coveringthree" in Siekmann and Stephan [43], exhaustiveness in Brown [15] and fairness in Bibel and Eder [10]. It is well-known and was already observed in Siekmann and Stephan [43] that the strong completeness conjecture is false for this notion of covering.

The problem is that a link can disappear without being resolved upon, namely by purity deletion, as the examples from the beginning demonstrate. Even the original links in the kernel can be deleted without being resolved upon, but may reappear after the copying process.

For this reason stronger notions of fairness are required: apparently even essential links can disappear without being resolved upon and reappear later due to the copying process. Hence we have to make absolutely sure that every link in the kernel is eventually resolved upon. To this end imagine that each initial link bears a distinct colour and that each descendant of a coloured link inherits the ancestor's colour:

Definition 3.13 An ordering filter `F` for clause graph resolution is called *coveringtwo*, if it is a covering and at least one link of each colour must have been resolved upon after at most finitely many steps.

At first sight this definition now seems to capture the essence, but how do we know that the "right" descendant (as there may be more than one) of the coloured ancestor has been operated upon? Hence the strongest definition of fairness for a filter:

Definition 3.14 A filter `F` for clause graph resolution is called *coveringone*, if each colour must have disappeared after at most finitely many steps.

While the strong completeness conjecture can be shown in the positive for the latter notion of covering (see Siekmann and Stephan [44]), hardly any of the practical and standard filters actually fulfill this property (except for some obvious and exotic cases).

So the strong completeness conjecture boils down to finding:

1. a proof or a refutation that a covering filter is strongly complete, for the appropriate notions of coveringone, -two, and -three, and
2. strong completeness results for subclasses of the full first-order predicate calculus, or
3. an alternative notion of covering for which strong completeness can be shown.

The first two problems were settled by Norbert Eisinger and Gerd Smolka.

Theorem 3.15 (Smolka) For the unit refutable class the strong completeness conjecture is true, i.e. the conjunction of a covering filter with any refutation complete and refutation confluent restriction filter is refutation complete, refutation confluent, and Noetherian, i.e. it terminates.

This theorem, whose essential contribution is due to Gerd Smolka [48] accounts for the optimism at the time. After all the unit refutable class of clauses (Horn clauses) turned out to be very important for many practical purposes, includng logic programming, and the theorem shows that all the essential properties of a useful proof procedure now hold for the clause graph procedure. Based on an ingenious construction, Norbert Eisinger showed however the following devastating result which we will look at again in more detail in Section 4.

Theorem 3.16 (Eisinger) In general the strong completeness conjecture is false, even for a restriction filter based on the coveringtwo definition.

This theorem destroyed once and for all the hope of finding a solution to the problem based on the notion of fairness, as it shows that even for the strongest possible form of fairness, strong completeness cannot be obtained.

So attention turned to the third of the above options, namely of finding alternative notions of a filter for which strong completeness can be shown. Early results are in Wrightson [56], Eisinger [21] and more recent results are Hähnle *et al.* [25], Meagher and Hext [30].

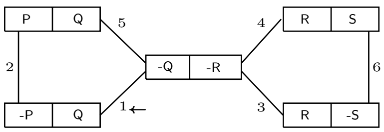
Let us now look at the proof of Theorem 3.16 in more detail.

4 The Eisinger Example

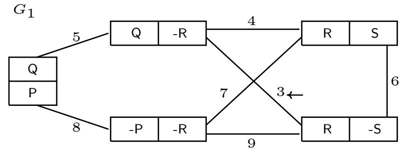
This example is taken from Eisinger [21], p. 158, Example 7.4_7. It shows a cyclic coveringtwo derivation, i.e. it shows that the clause graph proof procedure does not terminate even for the strong notion of a coveringtwo filter, hence in particular not for the notion of coveringthree either.

Let $S = \{PQ, -PQ, -Q-R, RS, R-S\}$ and $\text{INIT}(S) = G_0$.

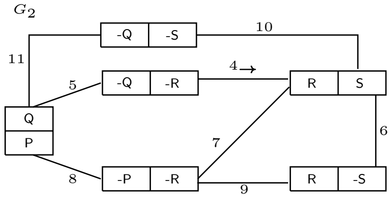
G_0



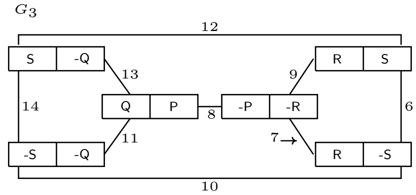
G_1



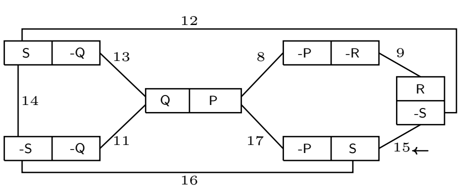
G_2



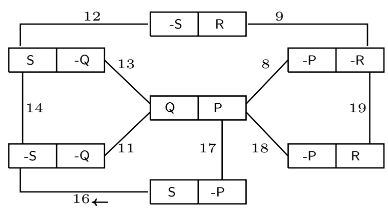
G_3



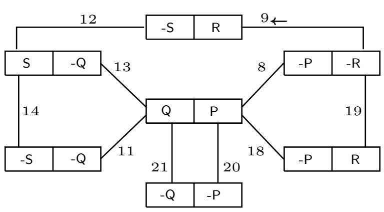
G_4



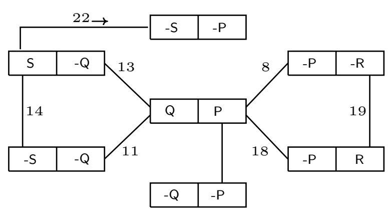
G_5



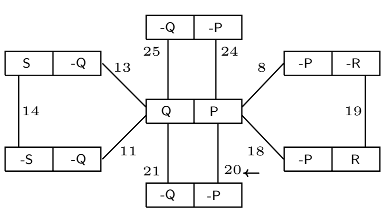
G_6



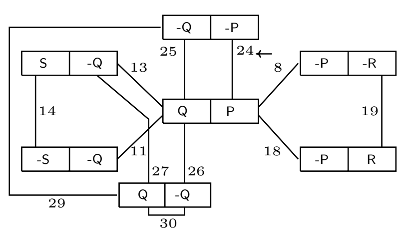
G_7



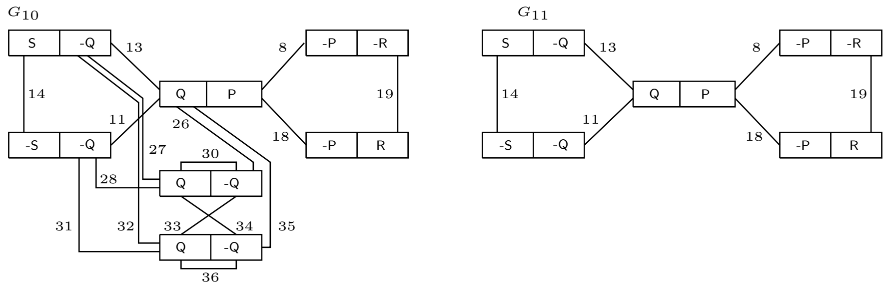
G_8



G_9



G_8 includes two copies of $-Q-P$, one of which might be removed by subsumption. To make sure that the phenomenon is not just a variation of the notorious subsumption problem described earlier in his monograph, Norbert Eisinger does not subsume, but performs the corresponding resolution steps for both clause nodes in succession.



G_{10} contains two tautologies and all links which are possible among its clause nodes. In other words, it is the initial clause graph of $\{S-Q, -S-Q, QP, -P-R, -PR, Q-Q, Q-Q\}$. So far only resolution steps and purity removals were performed; now apply two tautology removals to obtain G_{11}.

G_{11} has the same structure as G_0, from which it can be obtained by applying the literal permutation $\pi : \pm Q \mapsto \mp Q, \pm P \mapsto \pm S \mapsto \mp R \mapsto \pm P$. Since $\pi^6 = \mathrm{id}$, five more "rounds" with the analogous sequence of inference steps will reproduce G_0 as G_{66}, thus after sixty-six steps we arrive at a graph isomorphic to G_0.

The only object of G_0 still present in G_{11} is the clause node labelled PQ. In particular, all initial links disappeared during the derivation. Hence G_0 and G_{66} have no object in common, which implies that the derivation is covering. The following classes of link numbers represent the "colours" introduced for the coveringtwo concept in Definition 3.13; the numbers of links resolved upon are asterisked: $\{1*\}, \{2, 8, 17, 18, 20*, 23, 24*\}, \{3*, 9*, 19\}, \{4*, 7*\}, \{5, 11, 13, 21, 25, 26, \ldots, 36\}$, $\{6, 10, 12, 14, 15*, 16*, 22*\}$. Only the colour $\{5, 11, \ldots, 36\}$ was never selected for resolution during the first round, and it just so happens that the second round starts with a resolution on link 11, which bears the critical colour. Hence the derivation also belongs to the coveringtwo class.

This seminal example was discovered in the autumn of 1986 and has since been published and quoted many times. It has once and for all destroyed all hope of a positive result for the strong completeness conjecture based only on the notion of covering or fairness.

The consequence of this negative result has been compared to the most unfortunate fact that the halting problem of a Turing machine is unsolvable. The (weak) analogy is in the following sense: all the work on deduction systems rests upon the basic result that the predicate calculus is semidecidable, i.e. if the theorem to be shown is in fact valid then this can be shown after a finite number of steps, provided the uniform proof procedure carries out *every* possible inference step.

Yet, here we have a uniform proof procedure — clause graph resolution — which by any intuitive notion of fairness ("carries out every possible inference step eventually") runs forever even on a valid theorem — hence is not even semidecidable.

In summary:

> *The open problem is to find a filter that captures the essence of fairness on the kernel which is practically useful*[1] *— and then to show the strong completeness*

[1] This is important, as there are strategies which are known to be complete (for example to take a standard resolution

property holds for this new notion of a filter.

The open problem is not to invent an appropriate termination condition (even as brilliant as the decomposition criteria of Bibel and Eder [10][2] as the proof procedure will not terminate even for the strongest known notion of covering (fairness) — and this is exactly why the problem is still interesting even when the day is gone.

5 Lifting

All of the previous results and counterexamples apply to the propositional case or ground level as it is called in the literature on deduction systems.

The question is, if and how these ground results can be lifted to the general case of the predicate calculus.

While lifting is not necessarily the wrong approach for the connection graph, the proof techniques known so far are too weak: the problem is more subtle and requires much stronger machinery for the actual lifting.

The standard argument is as follows: first the result is established for the ground case, and there is now a battery of proof techniques[3] known in order to do so. After that the result is "lifted" to the general case in the following sense: Let S be an unsatisfiable set of clauses, then by Herbrand's theorem we know that there exists a finite truth-functionally contradictory set S' of ground instances of S. Now since we have the completeness result for this propositional case we know there exists a (resolution style) derivation. Taking this derivation, we observe that all the clauses involved are just instances of the clauses at the general level and hence "lifting" this derivation amounts to exhibiting a mirror image of this derivation at the general level, as the following figures shows:

$$\begin{array}{ccc} S & \vdash & \{\Box\} \\ \Downarrow & & \Uparrow \\ S' & \vdash_{\text{ground}} & \{\Box\} \end{array}$$

This proof technique is due to Alan Robinson [39].

Unfortunately this is not enough for the clause graph procedure, as we have the additional graph-like structure: not only has the ground proof to be lifted to the general level as usual, it has also to be shown that an isomorphic (or otherwise sufficient) *graph structure* can be mirrored from the ground level graph INIT(S') to the graph at the general level INIT(S), such that the derivation can actually be carried out within this graph structure as well:

---

theorem prover to find a proof and then use this information for clause-graph resolution). Hence these strategies are either based on some strange notion, or else on some too specific property.

[2] The weak notion of fairness as defined by W. Bibel and E. Eder [10] can easily be refuted by much simpler examples (see e.g. Siekmann and Stephan [43]) and Norbert Eisinger's construction above refutes a much stronger conjecture. The proof in the Bibel and Eder paper not only contains an excusable technical error, which we all are unfortunately prone to (the flaw is on page 336, line 29, where they assume that the fairness condition forces the procedure to resolve upon every link in the minimal complementary submatrix, here called the kernel), but unfortunately misses the very nature of the open problem (see also Siekmann and Wrightson [47]).

[3] Such as induction on the excess-literal-number, which is due to W. Bledsoe (see Loveland [29]).

$$
\begin{array}{ccc}
\mathrm{INIT}(S) & \vdash & \{G(\Box)\} \\
\Downarrow & & \Uparrow \\
\mathrm{INIT}(S') & \vdash_{\overline{\text{ground}}} & \{G'(\Box)\}
\end{array}
$$

where $G(\Box)$ is a clause graph that contains the empty clause $\Box$.

This turned out to be more difficult than expected in the late 1970's, when most of this work got started. However by the end of the 1980's it was well-known that standard lifting techniques fail: the non-standard graph-oriented lifting results in Siekmann and Stephan [44] turned out to be false. Similarly the lifting results in Bibel [8] and in Bibel and Eder [10], theorem 5.4 are also false.

To quote from Norbert Eisinger's monograph ([21], p. 125) on clause graphs

> "Unfortunately the idea (of lifting a refutation) fails for an intricate difficulty which is *the* central problem in lifting graph theoretic properties. A resolution step on a link in G (the general case) requires elimination of all links in G' (the ground refutation) that are mapped to the link in G. ... Such a side effect can forestall the derivation of the successor."

This phenomenon seems to touch upon a new and fundamental problem, namely, the lifting technique has to take the topological structure of the two graphs (the ground graph and the general clause graph) into account as well, and several additional graph-theoretical arguments are asked for.

The ground case part essentially develops a strategy which from any ground initial state leads to a final state. In the clause graph resolution system any such strategy has to willy-nilly distinguish between "good" steps and "bad" steps from each ground state, because there are ground case examples where an inappropriate choice of inference steps leads to infinite derivations that do not reach a final state. Eliminating or reducing the number of links with a given atom are sample criteria for "good" steps in different strategies. The lifting part then exploits the fact that it suffices to consider the conjunction of finitely many ground instances of a given first order formula, and show how to lift the steps of a derivation existing for the ground formula to the first order level. Clause graph resolution faces the problem that a single resolution step on the general level couples different ground level steps together in a way that may be incompatible with a given ground case strategy, because "bad" steps have to be performed as a side effect of "good" steps.

That this is not always straightforward and may fail in general is shown by several (rather complex) examples (pp.123–130 in Eisinger [21]), which we shall omit here. The interested reader may consult the monograph itself, which still represents most of what is known about the theoretical properties of clause graphs today.

To be sure, there is a very simple way to solve this problem: just add to the inference system an unrestricted copy rule and use it to insert sufficiently many variants.

However to introduce an unrestricted copy rule, as, for example, implicitly assumed in the Bibel [8] monograph, completely destroys the practical advantages of the clause graph procedure. It is precisely the advantage of the strong redundancy removal which motivated so many practical systems to employ this rather complicated machinery (see e.g. Ohlbach and Siekmann [36]). Otherwise we may just use ordinary resolution instead.

We feel that maybe the lifting technique should be abandoned altogether for clause graph refutation systems: the burden of mapping the appropriate graph structure (and taking its dynamically changing nature into account) seems to outweigh its advantages and a direct proof at the most general level with an appropriate technique appears far more promising. But only the future will tell.

6 Conclusion

The last twenty-five years have seen many attempts and partial results about so far unencountered theoretical problems that marred this new proof procedure, but it is probably no unfair generalisation to say, that almost every paper (including ours) on the problems has had technical flaws or major errors and the main problem — strong completeness — has been open ever since 1975 when clause graph resolution was first introduced to the scholarly community.

Why is that so?

One reason may be methodological. Clause graph resolution is formulated within three different conceptual frameworks: the usual clausal logic, the graphtheoretic properties and finally the algorithmic aspects, which account for its nonmonotonic nature. So far most of the methodological effort has been spent on the graphtheoretical notions (see e.g. Eisinger [21]) in order to obtain a firm theoretical basis. The hope being that once these graphtheoretical properties have a sound mathematical foundation, the rest will follow suit. But this may have been a misconception: it is — after all — the *metalogical* properties of the proof procedure we are after and hence the time may have come to question the whole approach.

In (Gabbay, Siekmann [24]) we try to turn the situation back from its (graph-theroetical) head to standing on its (logical) feet, by showing a logical encoding of the proof procedure without explicit reference to graphtheoretical properties.

Mathematics, it is said, advances through conjectures and refutations and this is a social process often carried out over more than one generation. Theoretical computer science and artificial intelligence apparently are no exceptions to this general rule.

Acknowledgements

This paper has been considerably improved by critical comments and suggestions from the anonymous referees and from Norbert Eisinger, Christoph Walther and Dov Gabbay.

References

[1] Andrews, P. B.: Resolution with Merging. *J. ACM* **15** (1968) 367–381.

[2] Andrews, P. B.: Refutations by Matings. *IEEE Trans. Comp. C-25*, (1976) **8**, 801–807.

[3] Andrews, P.B.: Theorem Proving via General Matings. *J. ACM* **28** (1981) 193–214.

[4] Antoniou, G., Ohlbach, H.J.: Terminator. *Proceedings 8th IJCAI*, Karlsruhe, (1983) 916–919.

[5] Bibel, W.: A Strong Completeness Result for the Connection Graph Proof Procedure. Bericht ATP-3-IV-80, Institut für Informatik, Technische Universität, München (1980).

[6] Bibel, W.: On the completeness of connection graph resolution. In *German Workshop on Artificial Intelligence*. J.Siekmann, ed. Informatik Fachberichte 47, Springer, Berlin, Germany (1981) pp.246–247.

[7] Bibel, W.: On matrices with connections. *J.ACM*, **28** (1981) 633–645.

[8] Bibel, W.: *Automated Theorem Proving*. (1982) Vieweg. Wiesbaden.

[9] Bibel, W.: Matings in matrices. *Commun. ACM*, **26**, (1983) 844–852.

[10] Bibel, W., Eder, E.: Decomposition of tautologies into regular formula and strong completeness of connection-graph resolution *J. ACM* **44** (1997) 320–344.

[11] Bläsius, K. H.: Construction of equality graphs. SEKI report SR-86-01 (1986) Univ. Karlsruhe, Germany.

[12] Bläsius, K. H.: Equality reasoning based on graphs. SEKI report SR-87-01 (1987) Univ. Karlsruhe, Germany.

[13] Bläsius, K. H., Bürckert, H. J.: *Deduktions Systeme*, (1992) Oldenbourg Verlag. Also in English: Ellis Horwood, 1989.

[14] Bläsius, K. H., Eisinger, N., Siekmann, J., Smolka, G., Herald A., Walter, C. The Markgraf Karl refutation procedure. *Proc 7th IJCAI*, Vancouver (1981).

[15] Brown, F. Notes on Chains and Connection Graphs. Personal Notes, Dept. of Computation and Logic, University of Edinburgh (1976).

[16] Chang, C.-L., Lee, R.C.-T.: *Symbolic Logic and Mechanical Theorem Proving*, Academic Press (1973).

[17] Chang, C.-L., Slagle, J.R.: Using Rewriting Rules for Connection Graphs to Prove Theorems. *Artificial Intelligence* **12** (1979) 159–178.

[18] Eisinger, N.: What you always wanted to know about clause graph resolution. In *Proc of 8th Conf. on Automated Deduction* Oxford (1986) LNCS 230, Springer.

[19] Eisinger, N.: Subsumption for connection graphs. *Proc 7th IGCAI*, Vancouver (1981).

[20] Eisinger, N.: Completeness, Confluence, and Related Properties of Clause Graph Resolution. Ph.D. dissertation, Universität Kaiserslautern (1988).

[21] Eisinger, N.: *Completeness, Confluence, and Related Properties of Clause Graph Resolution*. Pitman, London, Morgan Kaufmann Publishers,Inc., San Mateo,California (1991).

[22] Eisinger, N., Siekmann, J., Unvericht, E.: The Markgraf Karl refutation procedure. *Proc of Conf on Automated Deduction*, Austin, Texas (1979).

[23] Eisinger, N., Ohlbach, H. J., Präcklein, A.: Reduction Rules for Resolution Based Systems *Artificial Intelligence* **50,2** (1991) 141–181.

[24] Gabbay, D., Siekmann, J.: Logical encoding of the clause graph proof procedure, 2002, forthcoming.

[25] Hähnle, R., Murray, N. V., Rosenthal, E.: Ordered resolution versus connection graph resolution. In: R. Goré, A. Leitsch, T. Nipkow *Automated Reasoning, Proc of IJCAR 2001* (2001) LNAI 2083, Springer.

[26] Kowalski, R.: Search Strategies for Theorem Proving. *Machine Intelligence* (B.Meltzer and D.Michie, eds.), **5** Edinburgh University Press, Edinburgh, (1970) 181–201.

[27] Kowalski, R.: . A proof procedure using connection graphs. *J.ACM* **22** (1975) 572–595.

[28] Loveland, D. W.: A Linear Format for Resolution. *Proc. of Symp. on Automatic Demonstration.* Lecture Notes in Math 125, Springer Verlag, Berlin, (1970) 147–162. Also in Siekmann and Wrightson [46], 377–398.

[29] Loveland, D. W.: *Automated Theorem Proving: A Logical Basis* North- Holland, New York (1978).

[30] Meagher D., Hext, J.: Link deletion in resolution theorem proving (1998) unpublished manuscript.

[31] Murray, N. V., Rosenthal, E.: Path resolution with link deletion. *Proc. of 9th IJCAII* Los Angeles (1985).

[32] Murray, N. V., Rosenthal, E.: Dissolution: making paths vanish. *J. ACM* **40** (1993).

[33] Ohlbach, H. J.: Ein regelbasiertes Klauselgraph Beweisverfahren. *Proc. of German Conference on AI, GWAI-83* (1983) Springer Verlag IFB vol 76.

[34] Ohlbach, H. J.: Theory unification in abstract clause graphs. *Proc. of German Conf. on AI GWAI-85* (1985) Springer Verlag IFB vol 118.

[35] Ohlbach, H. J.: Link inheritance in abstract clause graphs *J. Autom. Reasoning* **3** (1987).

[36] Ohlbach, H. J., Siekmann, J.: The Markgraf Karl refutation procedure. In: J. L. Lassez, G. Plotkin, *Computational Logic* (1991) MIT Press, Cambridge MA.

[37] Omodeo, E. G.: The linked conjunct method for automatic deduction and related search techniques. *Computers and Mathematics with Applications* **8** (1982) 185–203.

[38] Ramesh, A., Beckert, B., Hähnle, R., Murray, N. V.: Fast subsumption checks using anti-links *J. Autom. Reasoning* **18** (1997) 47–83.

[39] Robinson, J.A.: A machine-oriented logic based on the resolution principle. *J.ACM* **12** (1965) 23–41.

[40] Shostak, R.E.: Refutation Graphs. *J. Artificial Intelligence* **7**, (1976), 51–64.

[41] Shostak, R.E.: A Graph-Theoretic View of Resolution Theorem-Proving. Report SRI International, Menlo Park (1979).

[42] Sickel, S.: A Search Technique for Clause Interconnectivity Graphs. *IEEE Trans. Comp.* **C-25** (1976) 823–835.

[43] Siekmann, J. H., Stephan, W.: Completeness and Soundness of the Connection Graph Proof Procedure. Bericht 7/76, Fakultät Informatik, Universität Karlsruhe (1976). Also in *Proceedings of AISB/GI Conference on Artificial Intelligence*, Hamburg (1978).

[44] Siekmann, J. H., Stephan, W.: Completeness and Consistency of the Connection Graph Proof Procedure. Interner Bericht Institut I, Fakultät Informatik, Universität Karlsruhe (1980).

[45] Siekmann, J. H., Wrightson, G.: Paramodulated connection graphs *Acta Informatica* **13** (1980).

[46] Siekmann, J. H., Wrightson, G.: *Automation of Reasoning.* Springer- Verlag, Berlin, Heidelberg, New York. Vol 1 and vol 2 (1983).

[47] Siekmann, J. H., Wrightson, G.: Erratum: A counterexample to W. Bibel's and E. Eder's strong completeness result for connection graph resolution. *J. ACM* **48** (2001) 145.

[48] Smolka, G.: Completeness of the connection graph proof procedure for unit refutable clause sets. In *Proceedings of GWAI-82*. Informatik Fachberichte, vol. 58. Springer-Verlag, Berlin, Germany (1982) 191-204.

[49] Smolka, G.: Einige Ergebnisse zur Vollständigkeit der Beweisprozedur von Kowalski. Diplomarbeit, Fakultät Informatik, Universität Karlsruhe (1982).

[50] Smolka, G.: Completeness and confluence properties of Kowalksi's clause graph calculus (1982) SEKI report SR-82-03, University of Karlsruhe, Germany.

[51] Stickel, M.: A Non-Clausal Connection-Graph Resolution Theorem-Proving Program. *Proceedings AAAI-82*, Pittsburgh (1982) 229–233.

[52] Walthe, Chr.: Elimination of redundant links in extended connection graphs. *Proc of German Workshop on AI, GWAI-81* (1981) Springer Verlag, Fachberichte vol 47.

[53] Wos, L.T., Carson, D.F., Robinson, G.A.: The Unit Preference Strategy in Theorem Proving. *AFIPS Conf. Proc. 26*, (1964) Spartan Books, Washington.
Also in Siekmann and Wrightson [46], 387–396.

[54] Wos, L.T., Robinson, G.A., Carson, D.F.: Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. *J.ACM* **12**, (1965) 536–541. Also in Siekmann and Wrightson [46], 484–492.

[55] Wos, L. T, *et al.*: *Automated Reasoning: Introduction and Applications* (1984) Englewood Cliffs, new Jersey, Prentice-Hall.

[56] Wrightson, G.: A pragmatic strategy for clause graphs or the strong completeness of connection graphs. Report 98-3, Dept Comp. Sci., Univ of Newcastle, Australia (1989).

[57] Yarmush, D. L.: The linked conjunct and other algorithms for mechanical theorem-proving. Technical Report IMM 412, Courant Institute of Mathematical Sciences, New York University (1976).

[58] Yates, R. A., Raphael, B., Hart, T. P.: Resolution Graphs. *Artificial Intelligence* **1** (1970) 257–289.

Received January 2002

Interest Group in Pure and Applied Logics (IGPL)

The Interest Group in Pure and Applied Logics (IGPL) is sponsored by The European Association for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, mathematical, computational, philosophical, etc.) from all over the world (currently, more than 50 countries). Our main activity is that of a research and information clearing house.

Our activities include:

- Exchanging information about research problems, references and common interest among group members, and among different communities in pure and applied logic.
- Helping to obtain photocopies of papers to colleagues (under the appropriate copyright restrictions), especially where there may be difficulties of access.
- Supplying review copies of books through the journals on which some of us are editors.
- Helping to organise exchange visits and workshops among members.
- Advising on papers for publication.
- Editing and distributing a Newsletter and a Journal (the first scientific journal on logic which is FULLY electronic: submission, refereeing, revising, typesetting, publishing, distribution; first issue: July 1993): the Logic Journal of the Interest Group on Pure and Applied Logics. (For more information on the Logic Journal of the IGPL, see the Web homepage: http://www.jigpal.oupjournals.org)
- Keeping a public archive of papers, abstracts, etc., accessible via ftp.
- Wherever possible, obtaining reductions on group (6 or more) purchases of logic books from publishers.

If you are interested, please send your details (name, postal address, phone, fax, e-mail address, research interests) to:

IGPL Headquarters
c/o Prof. Dov Gabbay
King's College, Dept of Computer Science
Strand
London WC2R 2LS
United Kingdom
e-mail: dg@dcs.kcl.ac.uk

For the organisation, Dov Gabbay, Ruy de Queiroz and Hans Jürgen Ohlbach