Logique & Analyse **150-151-152** (1995), 329-346

ALGORITHMS FOR RELEVANT LOGIC
Leo Apostel in memoriam

Paul GOCHET,* Pascal GRIBOMONT and Didier ROSSETTO

Abstract.
The classical analytic tableau method has been extended successfully to modal logics (see e.g. [3, 4, 5]) and also to relevant and paraconsistent logics [1, 2]. The classical connection method has been extended to modal and intuitionistic logics [10], and the purpose of this paper is to investigate whether a similar adaptation to relevant logic is possible. A hybrid method is developed for B^+ with a specific solution to the "multiplicity problem", as in the technique of modal semantic diagrams introduced in [7]. Proofs of soundness and completeness are also given.

1. *Introduction*

The sequent calculi and tableau methods suffer from three kinds of redundancies: duplication of redundant information, the consideration of reductions that do not advance the search toward finding a proof, and the need to distinguish derivations that differ in the order in which sequent rules are applied [10, p.82]. The connection method has proved computationally more efficient for classical, modal and intuitionistic logics; it may therefore be useful to extend it to other non-classical logics. This is especially true for logics used in artificial intelligence, for which efficient theorem proving techniques are needed. Bloesch has conjectured that such an extension is feasible in several cases [1, p.24]:

* We are grateful to the research team of the Automated Reasoning Project of the Australian National University (Canberra) for their invaluable help. We are very grateful to Dr. Bloesch for granting us the permission to quote his Ph.D. thesis. We received substantial help from Dr. Goré and Dr. Lugardon.We also thank Dr. van der Does and Dr. Herzig. This work was supported by a grant of the F.N.R.S. (project 8.4536.94) and an earlier version was discussed at the Centenary Conference of the Lvov-Warsaw School of Logic (Nov. 1995). We express our gratitude to our sponsors and our hosts, and to Dr. Gailly and Mrs Gailly Goffaux for several corrections. We owe the topic of this paper to the late Prof. Sylvan.

> "It is not clear how to create connection method style proof systems for many non classical logics. In particular the relevant and paraconsistent logics seem particularly difficult since the law of noncontradiction does not hold in most paraconsistent logics and many relevant logics. By relying heavily on the properties of classical logic, the connection method gains great efficiency but at the cost of poor flexibility. It does however seem fair to conjecture that techniques, such as using truth signs to represent exclusive semantic classes, developed in later chapters, could be applied to the connection method."

The main purpose of this paper is to investigate the connection method style for B^+, the most basic system of relevant logic, with the simplified semantics introduced in [8, 9]. The connection method presented here draws from two sources: Wallen's connection method for modal logic K and Bloesch's tableau method for relevant and paraconsistent logics. Both methods will have to be modified to fit our purpose. B^+ is known to be decidable. We present a decision procedure which automatically supplies finite models when applied to a formula which happens to be satisfiable.

This paper goes on as follows: the axiomatic system B^+ is recalled, Bloesch's Tableau Method for B^+ is briefly presented, Wallen's Connection Method for modal logic is adapted to B^+, and the soundness and the completeness of the extension are proven.

2. *The axiomatic method for relevant logic B^+*

2.1 *Axioms*

1. $A \to A$
2. $A \to (A \vee B), B \to (A \vee B)$
3. $(A \wedge B) \to A, (A \wedge B) \to B$
4. $(A \wedge (B \vee C)) \to ((A \wedge B) \vee C)$
5. $((A \to B) \wedge (A \to C)) \to (A \to (B \wedge C))$
6. $((A \to B) \wedge (B \to C)) \to ((A \vee B) \to C)$

2.2 *Rules*

1. $\dfrac{A, A \to B}{B}$ (modus ponens), and its disjunctive form.
2. $\dfrac{A, B}{A \wedge B}$ (adjunction) and its disjunctive form.
3. $\dfrac{A \to B, C \to D}{(B \to C) \to (A \to D)}$ (affixing rule) and its disjunctive form.

Comment. If $\frac{A_1, \ldots, A_n}{B}$ is a rule then its disjunctive form is the rule $\frac{C \vee A_1, \ldots, C \vee A_n}{C \vee B}$.

2.3 *Example*
Let us prove that formula 1 is a theorem of B^+

$$(p \to q) \to ((s \to q) \vee ((r \wedge p) \to q)) \tag{1}$$

Proof

1.	$(r \wedge p) \to p$	(axiom 3)
2.	$q \to q$	(axiom 1)
3.	$(p \to q) \to ((r \wedge p) \to q)$	(rule 3 (1, 2))
4.	$(p \to q) \to (p \to q)$	(axiom 1)
5.	$((r \wedge p) \to q) \to ((s \to q) \vee ((r \wedge p) \to q))$	(axiom 2)
6.	$((p \to q) \to ((r \wedge p) \to q)) \to ((p \to q) \to ((s \to q) \vee ((r \wedge p) \to q)))$	(rule 3 (4, 5))
7.	$(p \to q) \to ((s \to q) \vee ((r \wedge p) \to q))$	(rule 1 (3, 6))

This theorem will be proven again by the method presented here.

3. *Semantic tableau method for relevant logic B^+*

The principle of the semantic tableau method is rather straightforward. It is a systematic search for a model that falsifies the formula being checked for validity. If such a model does not exist, the formula is valid.

The formula φ to be proven is assumed first to be *false* inworld w_0 (we write $\mathbf{F}_{w_0}\varphi$). Next it is reduced by applying rules which remove the connectives stepwise starting with the less deeply nested connective until each branch of the tableau becomes closed or gives rise to a model of $\neg\varphi$. If all the branches of the tree are closed (i.e. there is no model that falsifies φ), $\mathbf{F}_{w_0}\varphi$ cannot be forced and φ is valid. This process is not deterministic. Generally several tableaux can be produced depending on which order we chose when we apply the reduction rules.

The reduction rules are based on the truth conditions. For example, $A \wedge B$ is *true* iff A and B are *true*. The truth conditions of the relevant conditional bear some resemblance with those of the classical conditional. Indeed, when the relevant conditional $A \to B$ is *false* the antecedent A is *true* and the consequent B is *false,* and when the relevant conditional is *true*, the antecedent is *false* or the consequent is *true*. However, as the rele-

vant conditional is intensional, something more is needed to capture its meaning. Just like modal formulas, the intensional formula $A \rightarrow B$ is evaluated at a world w and an accessibility relation relates the world w to the worlds w' and w'' at which A and B respectively are evaluated. The interpretation of relevant implication involves three worlds, instead of two for modal operators, so the relevant accessibility relation R is ternary.[1] The contrast between $\mathbf{F}_w A \rightarrow B$ and $\mathbf{T}_w A \rightarrow B$ matches the contrast between $\mathbf{F}_w \Box A$ and $\mathbf{T}_w \Box A$. Formula $A \rightarrow B$ is *false* at w iff w' and w'' exist such that $R_{ww'w''}$, A is *true* at w' and B is false at w'', just as $\Box A$ is *false* at w iff there exists a world w' such that $R_{ww'}$ and A is *false* at w'. Formula $A \rightarrow B$ is *true* iff for all w' and w'' such that $R_{ww'w''}$, A is *false* at w' or B is *true* at w''.

α	α_1	α_2
$\mathbf{T}_{w_i} x \wedge y$	$\mathbf{T}_{w_i} x$	$\mathbf{T}_{w_i} y$
$\mathbf{F}_{w_i} x \vee y$	$\mathbf{F}_{w_i} x$	$\mathbf{F}_{w_i} y$

β	β_1	β_2
$\mathbf{F}_{w_i} x \wedge y$	$\mathbf{F}_{w_i} x$	$\mathbf{F}_{w_i} y$
$\mathbf{T}_{w_i} x \vee y$	$\mathbf{T}_{w_i} x$	$\mathbf{T}_{w_i} y$

$\pi\alpha$	$\pi\alpha_1$	$\pi\alpha_2$
$\mathbf{F}_{w_i} x \rightarrow y$	$\mathbf{T}_{w_j} x$	$\mathbf{F}_{w_k} y$

w_j and w_k (j=k iff i=0) are new worlds and $R_{w_i w_j w_k}$ is added

$\nu\beta$	$\nu\beta_1$	$\nu\beta_2$
$\mathbf{T}_{w_i} x \rightarrow y$	$\mathbf{F}_{w_j} x$	$\mathbf{T}_{w_k} y$

w_j and w_k are such that $R_{w_i w_j w_k}$ was previously added

Fig. 1.: Reduction rules for B^+

Taking advantage of this semantics, Bloesch [1, p.88] gives reduction rules for $\mathbf{F}_w A \rightarrow B$ and $\mathbf{T}_w A \rightarrow B$.Whenever a formula such as $\mathbf{F}_w A \rightarrow B$ appears on a branch, we record, for that branch, that $R_{ww'w''}$ and we add the formulas $\mathbf{T}_{w'} A$ and $\mathbf{F}_{w''} B$ to the branch, where w' and $w''$are new worlds, *i.e.* they do not appear on the branch, and $w' = w''$ iff $w = w_0$ (*i.e.* the base world). Whenever a formula such as $\mathbf{T}_w A \rightarrow B$ appears on a branch, if we have $R_{ww'w''}$ on that branch, we may create two sub-branches with the formulas $\mathbf{F}_{w'} A$ on one and $\mathbf{T}_{w''} B$ on the other.

Since the tableau rule for $\mathbf{F}_{\rightarrow}$ is a hybrid rule which bears similarity both to π-rule (modal rule dealing with $T_{\Diamond}$) and to α-rule, it is appropriate to introduce the category of "$\pi\alpha$-rule". Analogously a tableau rule for $\mathbf{T}_{\rightarrow}$ is a hybrid rule which falls under the heading of "$\nu\beta$-rule" (modal ν-rule deals with $T_{\Box}$).

[1] A frame in modal logic is a graph; a frame in relevant logic is a hypergraph.

The tableau rules based on the semantics of the connectives fall therefore in four types: α, β, $\pi\alpha$ and $\nu\beta$ (figure 1).

1. *Prolongation rules* (or α-rules).Each α-node gives rise to a prolongation of the current branch. Two successive nodes are added, one of which is labelled with α_1 and the other is labelled with α_2.
2. *Branching rules* (or β-rules).Each β-node splits the branch into two sub-branches one of which bears a node labelled with β_1 and the other a node labelled with β_2.
3. *Relevant prolongation rule* (or $\pi\alpha$-rule). Each $\pi\alpha$-node gives rise to a prolongation of the branch. Two successive nodes are added, one of which is labelled with $\pi\alpha_1$ and the other is labelled with $\pi\alpha_2$.
4. *Relevant branching rule* (or $\nu\beta$-rule). Each $\nu\beta$-node splits the branch into two sub-branches one of which bears a node labelled with $\nu\beta_1$ and the other a node labelled with $\nu\beta_2$. The $\nu\beta$-rule can be used μ times; μ, also called the multiplicity of the $\nu\beta$-formula, is the number of pairs of worlds related to w_i previously introduced by the application of a $\pi\alpha$-rule ($R_{w_i w_j w_k}$ was previously added to the branch).

Comment. The parameter μ associated with a $\nu\beta$-formula has a definite value only at the end of the derivation. Here is the feature that will raise the *multiplicity problem* in the connection method.

4. *Connection method extended to relevant logic*

The principle of the connection method is the same as that of the tableau method: a formula is said to be proven by a connection proof whenever its attempted refutation fails. The connection method proceeds in three stages: a syntactic tree, an indexed formula tree and a path tree are successively built.

4.1 *The syntactic tree*

The syntactic tree or formation tree displays the structure of the formula. For example, figure 2 shows the syntactic tree of formula 1.The nodes are numbered by traversing the tree depthfirst, from left to right.

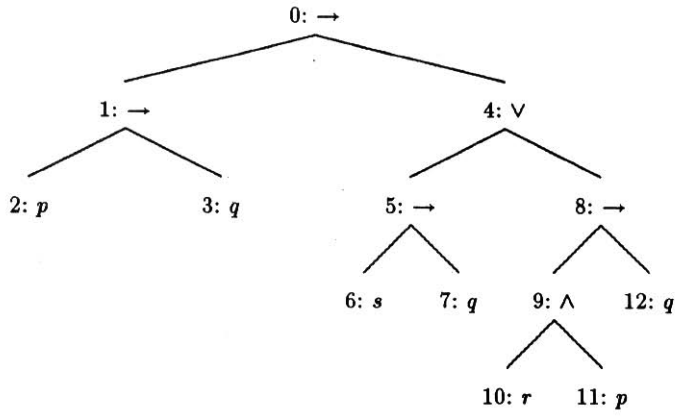

Fig. 2.: Syntactic Tree

4.2 *The indexed formula tree*

In tableau proof trees, a derivation is done by removing connectives in accordance with logical rules of elimination which we call *reduction steps.* These steps are *graphically* depicted (*e.g.* the elimination of an asserted conjunction gives rise to a prolongation of the branch, the elimination of a denied conjunction gives rise to the splitting of the branch, ...). The sign of the initial formula tested for inconsistency is **F** since a refutation proof is intended. The sign of the other formulas is determined by the sign of the input formula and by the reduction rule applied to it (*e.g.* the denial of a formula whose main connective is a conjunction leads to the disjunction of the denials of each conjunct).

In the connection method, these pieces of information are encoded in the indexed formula tree.The indexed formula tree associated with a signed formula φ can be represented as an array (figure 3} shows the indexed formula tree of formula 1). Each line of this array is a record representing a signed subformula of φ obtained when applying recursively the reduction rules (figure 1) to φ.

k	$pol(k)$	$lab(k)$	$Pt(k)$	$St(k)$	$w(k)$	$h(k)$
a_0	**F**	$(p \to q) \to ((s \to q) \vee ((r \wedge p) \to q))$	$\pi\alpha$	–	w_0	0
a_1	**T**	$p \to q$	$\nu\beta$	$\pi\alpha_1$	w_1	1
a_2^1	**F**	p	–	$\nu\beta_1$	w_2	4
a_3^1	**T**	q	–	$\nu\beta_2$	w_3	4
a_2^2	**F**	p	–	$\nu\beta_1$	w_4	7
a_3^2	**T**	q	–	$\nu\beta_2$	w_5	7
a_4	**F**	$(s \to q) \vee ((r \wedge p) \to q)$	α	$\pi\alpha_2$	w_1	1
a_5	**F**	$s \to q$	$\pi\alpha$	α_1	w_1	2
a_6	**T**	s	–	$\pi\alpha_1$	w_2	3
a_7	**F**	q	–	$\pi\alpha_2$	w_3	3
a_8	**F**	$(r \wedge p) \to q$	$\pi\alpha$	α_2	w_1	2
a_9	**T**	$r \wedge p$	α	$\pi\alpha_1$	w_4	5
a_{10}	**T**	r	–	α_1	w_4	6
a_{11}	**T**	p	–	α_2	w_4	6
a_{12}	**F**	q	–	$\pi\alpha_2$	w_5	5

Fig. 3.: Indexed Formula Tree

The indexed formula tree has seven columns. The first column is a key identifying the line;[2] the key will be a_i, where i is the index of the subformula in the syntactic tree of φ.Some formulas have to be repeated (see later); superscripts are used to distinguish the multiple occurrences of these formulas, and are transmitted to their children. The second column, named $pol(k)$ (where k is the key), records the polarity of the formula (**T** or **F**). The third column, named $lab(k)$, records the label, *i.e.* the signed subformula itself. Each non atomic signed formula has a type, which can be α, β, $\pi\alpha$ or $\nu\beta$, and also two components, called *children*, as we saw in figure 1. Each child has its own type, called the primary type; its secondary type is the (primary) type of the parent formula, with the subscript 1 or 2, as indicated in figure 1.The fourth column, named $Pt(k)$, records the (primary) type of the subformula; an atomic formula has no primary type (- in the fourth column). The fifth column, named $St(k)$, records the secondary type of the subformula. The main formula φ has no secondary type (- in the fifth column). The columns "$Pt(k)$"and "$St(k)$" are introduced merely to support the understanding of the indexed formula tree. In relevant logic, as in modal logic, the polarity is ascribed to a formula relatively to a given world. The

[2] The path tree will deal with these indices to represent the (sub)formulas in an efficient way.

sixth column, named $w(k)$, records this piece of information; worlds are given names from the unlimited sequence w_0, w_1,w_2, ... Indeed two identical formulas of opposite signs such as **T***A* and **F***A* will combine together to yield a contradiction only if they are evaluated at the same world. The last column, named $h(k)$, records the history: it contains the number of the step during which the line is created. Column 7 is also introduced to support the understanding of the indexed formula tree.

We now give the iterative procedure to construct the indexed formula tree. Throughout the execution, the procedure maintains for each line a binary variable, whose value can be "active" or "passive". Furthermore, for each line of type $\nu\beta$, a set Ω of ordered pairs of worlds is maintained.

Each step of the procedure selects an active line ℓ and generates two or more new lines ℓ_1, ℓ_2,, ... In every case, the label of a new line is a child of the label of the parent line, according to the type-dependent reduction rules given in figure 1. After the generation, line ℓ becomes passive, whereas lines ℓ_1, ℓ_2,, ... are active if their label is non atomic and passive otherwise. The set Ω associated with a new $\nu\beta$-line is empty.

Initial step 0. *Create an initial line.*

- The initial line has key a_0; its label is the given signed formula, to which we ascribe the initial world w_0. The history is 0. The line is made active if its (primary) type is α, β or $\pi\alpha$, and passive otherwise.

Iterative step $n > 0$. *Select an active line of key k.*

- If $Pt(k)=\alpha$, two lines are added to the indexed formula tree. Their labels respectively are the α_1-child and the α_2-child of *lab*(*k*,) determined by the α-reduction rule; their indices are extracted from the syntactic tree; their world and history are $w(k)$ and n, respectively.
- If $Pt(k)=\beta$, two lines are added to the indexed formula tree. Their labels respectively are the β_1-child and the β_2,-child of *lab*(*k*), determined by the β-reduction rule; their indices are extracted from the syntactic tree and their world is $w(k)$. The history of both new lines is n.
- If $Pt(k)=\pi\alpha$, two lines are added to the indexed formula tree. Their labels respectively are the $\pi\alpha_1$-child and the $\pi\alpha_2$-child of *lab*(*k*) ,determined by the $\pi\alpha$-reduction rule; their indices are extracted from the syntactic tree.The worlds w_j and w_k associated with these two formulas must not have been used before, say the first elements of the world sequence which differ from all worlds associated with existing lines. The history of both new lines is n.Furthermore, each existing $\nu\beta$-line k' (*i.e.* $Pt(k')=\nu\beta$) such that $w(k') = w(k)$ is made active again.
- If $Pt(k)=\nu\beta$, t he set E of all passive existing $\pi\alpha$-lines k' (*i.e.* $Pt(k')=\pi\alpha$) such that $w(k')=w(k)$ is determined. For each k' in E we do the following. As k' is passive, it has two children, say k'' and k''', whose associated

worlds are $w(k'')$ and $w(k''')$. If $(w(k''), w(k'''))$ is not an element of the set $\Omega(k)$ associated with line k, two new lines, say ℓ and ℓ', are added to the indexed formula tree. Signed subformula $lab(\ell)$ and $lab(\ell')$ are the children of $lab(k)$, as determined by the $\nu\beta$-reduction rule; the indices ℓ and ℓ' are extracted from the syntactic tree. Remember that a superscript is used to distinguish the different pairs of children of the parent formula. The associated worlds $w(\ell)$ and $w(\ell')$ are $w(k'')$ and $w(k''')$ respectively, and the ordered pair $(w(k''), w(k'''))$ is added to $\Omega(k)$. The history of each new line is n.

Comment. It should be emphasized that a $\nu\beta$-line k may switch several times from active to passive and conversely during the execution. Besides, each step about the $\nu\beta$-line k may introduce any number of new ordered pairs of children for k, up to the size of the set E. Last, the worlds associated with the children are not inherited from the parent line k, but from other lines, of secondary type $\pi\alpha_i$ $(i=1,2)$.

4.3 *The path tree*

The path tree (it is actually an acyclic graph, see [6] for an example) has the same role as a tableau proof tree, but its construction is computationally more efficient. The path tree comes from the following recursive definition of a path.

Basis

- If a_0 is the root of the indexed formula tree, a_0 is a path.

Recursion

- If S is a path containing the node α,[3]
  $(S \setminus \{\alpha\}) \cup \{\alpha_1\} \cup \{\alpha_2\}$ is a path.
- If S is a path containing the node β,
  $(S \setminus \{\beta\}) \cup \{\beta_1\}$ and $(S \setminus \{\beta\}) \cup \{\beta_2\}$ are paths.
- If S is a path containing the node $\pi\alpha$,
  $(S \setminus \{\pi\alpha\}) \cup \{\pi\alpha_1\} \cup \{\pi\alpha_2\}$ is a path.
- If S is a path containing the node $\nu\beta$,
  $S \setminus \{\nu\beta\}, \cup \{t_1, \ldots, t_\mu\}$, where $t_i (1 \leq i \leq \mu)$ is either $\nu\beta_1^i$ or $\nu\beta_2^i$, are paths.

Comments. There are 2^μ such paths, where μ is the multiplicity associated with $\nu\beta$. The μ ordered pairs $(\nu\beta_1, \nu\beta_2)$ used here are the μ ordered pairs (among the ordered pairs $(\nu\beta_1, \nu\beta_2)$ got during the building of the indexed formula tree) which are *frame compatible* with the nodes of the

[3] *i.e* node of the indexed formula tree whose label is a signed formula of primary type a.

current path, i.e. $\nu\beta_1$ and $\nu\beta_2$ are associated with worlds already involved in the current path.[4]

Each step in the construction of the path tree consists of the application of a rule (α, β, $\pi\alpha$, *or* $\nu\beta$) to a formula. These applications generate new paths. It should be emphasized here that as soon as a the successors of a path have been determined, this path can be erased. In fact it is erased when the connection method is implemented on acomputer; the path tree never resides wholly in the computer memory. Only its leaves, i.e., the atomic paths, are saved and determine models of $\neg\varphi$ if they do not contain connections. The atomic paths contain only atomic formulas and (vacuously true) $\nu\beta$-formulas.

The formula tested is a theorem if and only if each leaf of the path tree contains a *connection*, that is, two signed formulas $\mathbf{T}_{w_i}A$ and $\mathbf{F}_{w_j}B$, with $w_i = w_j$ and A identical to B.

To ensure soundness, the path tree has to respect a *reduction ordering* which combines two orderings: the *subformula ordering* and the *modal ordering*. The reduction ordering (denoted $\lhd$) is the transitive closure of the union of the subformula ordering (denoted $\lessdot$) and the modal ordering (denoted $\sqsubset_M$), so $\lhd = (\lessdot \cup \sqsubset_M)^*$[10].

Therefore the recursion has to be applied in the following way.

1. *Respecting the order of the world indices*: consider nodes related to world w_i only when all nodes attached to world w_j with $j < i$ have been considered.
2. *Respecting the necessity order*: consider $\nu\beta$-nodes related to world w_i only when all other nodes related to world w_i have been considered.

This is automatically done if the recursion is applied in the order recorded in the seventh column $h(k)$ of the indexed formula tree.

The reason behind the frame compatibility restriction is this: allowing the introduction in a path of formulas that are not frame compatible with the other formulas of the path would amount to allowing a move from one subbranch to another in a tableau proof tree, which would clearly be unsound.

Let us observe that if we consider the semantic tableau style of proof, a $\nu\beta$-formula could introduce μ successive branching that lead to 2^μ splits of the branch. In the path tree, all the 2^μ paths generated by its instantiation are got in one step. Indeed, in tableau trees we do the branching and instantiation *separately*. On the contrary here we combine branching with modal instantiation when applying $\nu\beta$-rules. Our policy produces a reduc-

[4] A *syntactic criterion* to check frame compatibility between two formulas is the following: formula φ is frame compatible with formula ψ if and only if the primary type of their common ancestor in the syntactic tree is neither β nor $\nu\beta$

tion of the number of nodes since 2^{μ} nodes are needed instead of $\sum_{i=1}^{\mu} 2^i = 2^{\mu+1} - 2$.

When building the path tree, the same operation applies for rules of any category: we replace the parent formula by its children. Here again our technique diverges from the standard one used in modal logic, where whenever a necessity rule is applied, a child is added but the parent formula is maintained; if the n^{th} instantiation has failed to produce a contradiction, a new instantiation takes place; multiplicity is demand driven [10].

Demand driven policy secures completeness. However if the formula tested fails to be unsatisfiable, a loop may occur. The policy advocated in this paper rules out the risk of non termination as far as B^+ is concerned. Moreover it automatically supplies finite models whenever the formula tested for inconsistency happens to be satisfiable. Our treatment of the multiplicity problem establishes a tight correspondence between the $\nu\beta$-reduction rule and the semantics of $\nu\beta$-formulas.

The path tree corresponding to formula 1 is represented in figure 4. To help the reader to understand the path tree, the node developed is underlined once and the indices standing for connections are underlined twice. There are four atomic paths, each of them containing a connection, so no model exists for signedformula $\mathbf{F}_{w_0}\varphi$, and φ is a B^+-valid.

We see that the four atomic paths all contain a connection, therefore no frame that falsifies formula 1 exists, *i.e.* this formula is B^+-valid.

$$\{a_1\}$$
$$\{a_1, \underline{a_4}\}$$
$$\{a_1, \underline{a_5}, a_8\}$$
$$\{a_1, a_6, a_7, \underline{a_8}\}$$
$$\{\underline{a_1}, a_6, a_7, a_9, a_{12}\}$$

$$\{a_2^1, a_2^2, a_6, a_7, \underline{a_9}, a_{12}\} \quad \{a_3^1, a_2^2, a_6, a_7, \underline{a_9}, a_{12}\} \quad \{a_2^1, a_3^2, a_6, a_7, \underline{a_9}, a_{12}\} \quad \{a_3^1, a_3^2, a_6, a_7, \underline{a_9}, a_{12}\}$$

$$\{a_2^1, \underline{\underline{a_2^2}}, a_6, a_7, a_{10}, \underline{\underline{a_{11}}}, a_{12}\} \quad \{a_3^1, \underline{\underline{a_2^2}}, a_6, a_7, a_{10}, \underline{\underline{a_{11}}}, a_{12}\} \quad \{a_2^1, \underline{\underline{a_3^2}}, a_6, a_7, a_{10}, a_{11}, \underline{\underline{a_{12}}}\} \quad \{\underline{\underline{a_3^1}}, \underline{\underline{a_3^2}}, a_6, \underline{a_7}, a_{10}, a_{11}, a_{12}\}$$

Fig. 4.: Path Tree

5. *Soundness and completeness*

In this section we prove that our method is a sound and complete decision procedure for relevant logic B^+.

5.1 *Complexity*

If A is a (signed or not) formula, its *degree* $d(A)$ is the number of internal nodes of the syntactic tree associated with A, *i.e.* the total number of connectives contained in A. In other words, the degree of an atomic formula is 0; $d(\neg A) = 1 + d(A)$ and if $\circ$ is a binary operator, $d(A \circ A') = 1 + d(A) + d(A')$. The notion of degree also applies to signed formulas: $d(\mathbf{T}A) = d(\mathbf{F}A) = d(A)$.

If $\mathscr{E}$ is a set of formulas, its *complexity* $c(\mathscr{E})$ is the sum of the degrees of its elements.

Every finite set of formulas has a finite complexity. Furthermore, with usual notation, we have the following inequalities:

- if $\mathscr{E}_1 \subseteq \mathscr{E}_2$, then $c(\mathscr{E}_1) \leq c(\mathscr{E}_2)$.
- $c(\mathscr{E} \cup \{\alpha\}) > c((\mathscr{E} \setminus \{\alpha\}) \cup \{\alpha_1, \alpha_2\})$,
  since $d(\alpha) > d(\alpha_1) + d(\alpha_2)$.
- $c(\mathscr{E} \cup \{\beta\}) > \max(c((\mathscr{E} \setminus \{\beta\}) \cup \{\beta_1\}),\ c((\mathscr{E} \setminus \{\beta\}) \cup \{\beta_2\}))$,
  since $d(\beta) > d(\beta_1)$ and $d(\beta) > d(\beta_2)$.
- $c(\mathscr{E} \cup \{\pi\alpha\}) > c((\mathscr{E} \setminus \{\pi\alpha\}) \cup \{\pi\alpha_1, \pi\alpha_2\})$,
  since $d(\pi\alpha) > d(\pi\alpha_1) + d(\pi\alpha_2)$.
- $c(\mathscr{E} \cup \{\nu\beta\}) > \max(c((\mathscr{E} \setminus \{\nu\beta\}) \cup \{\nu\beta_1\}),\ c((\mathscr{E} \setminus \{\nu\beta\}) \cup \{\nu\beta_2\}))$,
  since $d(\nu\beta) > d(\nu\beta_1)$ and $d(\nu\beta) > d(\nu\beta_2)$.

5.2 *Auxiliary structure*

As a preliminary result, we need to know that the execution of the construction procedure for the indexed formula tree always terminates. We first observe that every step induces the addition of finitely many new lines (or nodes). This is trivial, even for $\nu\beta$-steps, since the number of lines added in a $\nu\beta$-step cannot exceed $2n$ where n is the total number of lines introduced before this step. As a consequence, termination could be prevented only if an infinite number of steps could take place.

In order to prove that every execution involves finitely many steps, we will suppose that each step of the construction procedure updates not only the indexed formula tree, but also an auxiliary structure defined below; it will then be sufficient to prove that this structure can be updated only finitely many times. The auxiliary structure is a set of hypertrees; the nodes of the hypertrees are worlds (so each hypertree is a frame). Each node w_i is labelled with a set $\ell(w_i)$ of signed formulas.

We describe now the update induced on the auxiliary structure by each step of the construction procedure, for a signed formula φ.

Initial step 0. (Creation of the initial line.)

- The structure initially contains only one frame; this frame consists of a single node w_0, whose label contains a single element, which is the initial signed formula: $\ell(w_0) = \{\varphi\}$.

Iterative step n > 0. (Addition of children of line k.)

In every case, frames that contain world-node $w(k)$ with signed formula "$pol(k) : lab(k)$" in $\ell(w(k))$ already exist in the auxiliary structure.

- If $Pt(k)=\alpha$, in each frame such that the signed formula $\alpha = pol(k) : lab(k)$ is a member of $\ell(w(k))$, we update $\ell(w(k))$, by replacing its element α by the corresponding elements α_1 and α_2.
- If $Pt(k)=\beta$, each frame of the auxiliary structure that has a node named $w(k)$ with the signed formula $\beta = pol(k) : lab(k)$ in its label $\ell(w(k))$, is updated as follows. First, the frame is replaced by two identical frames; second, in the label $\ell(w(k))$, the element β is replaced by the corresponding element β_1 in the first frame and β_2 in the second one.
- If $Pt(k)=\pi\alpha$, each frame of the auxiliary structure that has a node named $w(k)$ with the signed formula $\pi\alpha = pol(k) : lab(k)$ in its label $\ell(w(k))$ is updated as follows. Two nodes w and w' and the hyperarrow $(w(k), w, w')$ are added to the frame, where w and w' are the new worlds associated with the children of line k. The new nodes are labelled respectively $\ell(w) = \{\pi\alpha_1\}$ and $\ell(w')=\{\pi\alpha_2\}$.
- If $Pt(k)=\nu\beta$, each frame that contains node $w(k)$, with $\nu\beta \in \ell(w(k))$ and that has a hyperarrow $(w(k), w, w')$ for some w and w' is updated as follows, if it has not been done yet. First, the frame is replaced by two identical frames; second, the children signed formulas $\nu\beta_1$ and $\nu\beta_2$ are added to $\ell(w)$ in the first frame and to $\ell(w')$ in the second frame respectively.

We first observe that duplication of frames is induced by β and $\nu\beta$-reductions. However, a branching formula can be a subformula of another branching formula, so β-reductions and $\nu\beta$-reductions can be reused several times. As a result, the number of frame duplication is bounded by $d(\varphi)!$ and the total number of frames in an auxiliary structure cannot exceed $2^{d(\varphi)!}$.

Furthermore, each frame of the auxiliary structure is finite. First, such a frame is a finitary hypertree since the number of successors of any node of the frame is bounded by the number of $\pi\alpha$-subformulas contained in the initial formula, and therefore by its degree $d(\varphi)$. Second, the length of a hyperbranch cannot exceed $d(\varphi) = c(\{\varphi\})$, since the complexity of a successor node is strictly less than the complexity of the parent node. An upper bound for the number of nodes in a frame is $d(\varphi)^{d(\varphi)+1}$, and an upper bound for the total number of nodes in the whole structure is therefore $\Sigma_\varphi =_{def} 2^{d(\varphi)!} d(\varphi)^{d(\varphi)+1}$ (this is also an upper bound for the number of hyperarrows). Tighter bounds can be found, but we do not need them here.

Comment. The notion of auxiliary structure applies not only to a signed formula, but also to a finite (conjunctive) set of signed formulas.

5.3 *Termination proofs*

We can now give an upper bound for the length of any execution of the construction algorithm for the indexed formula tree. Each step induces at least one of the following operations:

1. Extension of the auxiliary structure (β-step, $\pi\alpha$-step, useful $\nu\beta$-step) (cannot be performed more than Σ_φ times);
2. Addition of a formula to a node label (useful $\nu\beta$-step,$\pi\alpha$-step) (cannot be performed more than $d(\varphi)\times\Sigma_\varphi$ times);
3. Reduction of an element of a node label (α-step, β-step) (cannot be performed more than $d(\varphi)\times\Sigma_\varphi$ times).

Comment. A $\nu\beta$-step about line k will do nothing at all if no new ordered pair of worlds accessible from $w(k)$ has been created since the last activation; however, as $\nu\beta$-lines are "reactivated" by $\pi\alpha$-steps, useless steps can occur only finitely many times.

This completes the termination proof for the indexed formula tree algorithm.

In order to prove the termination of the path tree algorithm, we observe that each step adds a finite number of paths to the path tree, which is a finitary tree. Due to König's lemma, it is sufficient to establish that a new path is always strictly less complex than its parent. However, this is not true for the notion of (multi)-set complexity introduced above; indeed,

$$c((S\setminus\{\nu\beta\})\cup\{\nu\beta_{i_1}^1,\ldots,\nu\beta_{i_\mu}^\mu\})<c(S),\text{ where } i_j\in\{1,2\}\ (1\le j\le\mu),$$

cannot be guaranteed.

The problem is, we know the multiplicity μ to be finite, but nothing else. To solve this, we define another well-founded ordering on paths. First, to each path S we associate $m(S)$, a multiset of natural numbers, which are the degrees of the elements of the path. Second, we show that, for some relation $\sqsubset$, the domain of multisets is well-founded. Third, we show that, if S' is produced by the algorithm from the path S, then $m(S')\sqsubset m(S)$.

A multiset of natural numbers can be represented as a decreasing sequence of numbers; the lexical ordering on these sequences is defined as follows:

$$(a_1,\ldots,a_n)\sqsubset(b_1,\ldots,b_m)$$

if there is a number $r\in\mathbf{N}$ such that $r\le n, r\le m, a_i=b_i$ for $i=1,\ldots,r$ and either $r=n$ and $r<m$, or $r<n$, $r<m$ and $a_{r+1}<b_{r+1}$.

As (N,<) is a well-ordered set, so is (N^*, $\sqsubset$). Besides, the lexical ordering $\sqsubset$ induces a (partial) ordering (also noted $\sqsubset$) on the set of paths:

$$S \sqsubset S' =_{def} m(S) \sqsubset m(S')$$

and this ordering is well-founded.

Last, we observe that each reduction step for a path consists in replacing an element of the path by finitely many elements of lower degree; this always leads to a $\sqsubset$-smaller path.

5.4 *Hypertree-models*

The auxiliary structure is in fact a set of frames, and each frame is an interpretation of the initial signed formula φ, called a *hypertree-frame*; it is a *hypertree-model* for φ if φ holds at the initial world w_0. A hypertree-frame is *consistent* if no world label contains a pair of opposite formulas. We first prove the following lemma.

Lemma I. A (hypertree-)frame $\mathcal{S}$ is consistent if and only if $\mathcal{S}, w_i \models \psi$, for all worlds $w_i \in \mathcal{S}$ and for all signed formulas $\psi \in \ell(w_i)$.

Proof. The "if" part is trivial: if $\mathcal{S}, w_i \models \psi$ and $\mathcal{S}, w_i \models \xi$ then $\{\psi, \xi\}$ cannot be a pair of opposite signed formulas. For the "only if" part, we proceed by induction on the height of the nodes in the frame. A leaf w (a node without successor) has height $h(w) = 0$, and if w is an internal node with successors $\{(w_1, w_1'), \ldots, (w_k, w_k')\}$, then $h(w) = 1 + \max \{(h(w_1) + h(w_1')), \ldots, (h(w_k) + h(w_k'))\}$[5]

Base case. If the world w has no successor in cal $\mathcal{S}$, the label $\ell(w)$ contains only atoms and $\nu\beta$-formulas. The $\nu\beta$-formulas are vacuously true, and the atoms are forced to be true at w, which is possible since there is no opposite pair.

Induction case. Let w be a node of height $n>0$. As for the base case, all atomic formulas can be forced at w. Non atomic formulas are $\nu\beta$ or $\pi\alpha$. Let $\{(w_1, w_1'), \ldots, (w_k, w_k')\}$ be the (non empty) set of successors. All w_i and w_i' have a height less than n, so the induction hypothesis applies to them. Let $\nu\beta \in \ell(w)$; due to the construction process of the frame, either $\nu\beta_1^i \in \ell w_i$ or $\nu\beta_2^i \in \ell w_i'$ for all $i = 1, \ldots, k$, so by the induction hypothesis, either $\nu\beta_1^i$ is forced at w_i or $\nu\beta_2^i$ is forced at w_i' for all $i = 1, \ldots, k$, and so $\nu\beta$ is forced at w. Similarly, let $\pi\alpha \in \ell(w)$; due to the construction process of the frame, $\pi\alpha_1 \in \ell(w_i)$ and $\pi\alpha_2 \in \ell(w_i')$ for some $i \in 1, \ldots, k$, so by the induction hy-

[5] Node w has successor (w', w'') if (w, w', w'') is a hyperarrow.

pothesis, $\pi\alpha_1$ is forced at w_i and $\pi\alpha_2$ is forced at w_i' for some i, and so $\pi\alpha$ is forced at w.

Lemma II. If Φ is a satisfiable finite set of formulas, then at least one of the hypertree-frame of the Φ-auxiliary structure is a (hypertree-)model of Φ.

Proof. We proceed by induction on the complexity of Φ.

Base case. If the complexity of Φ is 0, Φ contains only atomic signed formulas, and its hypertree-frame contains the single world w_0 where each φ in Φ is forced; this is possible if and only if Φ is satisfiable. This frame clearly is a model of Φ.
Induction case. Φ contains at least one non-atomic signed formula.

If Φ contains an α-formula, then $(\Phi \setminus \{\alpha\}) \cup \{\alpha_1, \alpha_2\}$ has lower complexity than Φ and is also satisfiable; therefore one of the hypertree-frames of $(\Phi \setminus \{\alpha\}) \cup \{\alpha_1, \alpha_2\}$ is a model of $(\Phi \setminus \{\alpha\}) \cup \{\alpha_1, \alpha_2\}$ and also of~Φ.

If Φ contains a β-formula, then $(\Phi \setminus \{\beta\}) \cup \{\beta_1\}$ or $(\Phi \setminus \{\beta\}) \cup \{\beta_2\}$ that has lower complexity than Φ, is also satisfiable and has a hypertree-model, which is also a model of Φ.

If the set Φ contains only $\pi\alpha$-formulas and $\nu\beta$-formulas, let $\Phi = \mathcal{P} \cup \mathcal{N}$ with $\mathcal{P} = \{\pi\alpha^1, \ldots, \pi\alpha^n\}$ and $\mathcal{N} = \{\nu\beta^1, \ldots, \nu\beta^m\}$ Furthermore, let $\mathcal{N}_{12} = \{\{\nu\beta^1_{i_1}, \ldots, \nu\beta^m_{i_m}\}: i_1, \ldots, i_m \in \{1, 2\}$ and$\} \nu\beta \in \mathcal{N}\}$. If $X \in \mathcal{N}_{12}$, let X_1 be the set of $\nu\beta_1$-elements of X and X_2 the set of $\nu\beta_2$-elements of X. If Φ is satisfiable, then for each ordered pair $(\pi\alpha^i_1, \pi\alpha^i_2)$ such that $\pi\alpha^i \in \mathcal{P}$, there exists an element $X \in \mathcal{N}_{12}$ such that the sets $\Phi^i_1 = \{\pi\alpha^i_1\} \cup X_1$ and $\Phi^i_2 = \{\pi\alpha^i_2\} \cup X_2$ are also satisfiable with a lower complexity, and therefore have a hypertree-model. The corresponding hypertree-model of Φ is obtained as follows: the root is a world w_0, with $\ell(w_0,) = \Phi$, and there are n outgoing hyperarrows, leading to the $2n$ hypertree-models of the $\Phi^i_{1,2}, i = 1, \ldots, n$. (Renaming of worlds is used to avoid name clashes.)

Comment. The case $n = 0$ is not ruled out; a set of $\nu\beta$-formulas is always B^+-satisfiable, and every one-world frame is a model.

Theorem 1. Signed formula φ has a model if and only if some hypertree-frame associated with φ is a hypertree-model of φ.

It is an immediate consequence of lemmas I and II.

5.5 *Path tree, soundness and completeness*

Lemma III. The path tree associated with a finite (conjunctive) set of signed formulas is finite.

Proof. See §5.3, where the termination of the construction algorithm for the path tree has been proven.

A *line* in a hypertree-frame is a sequence $(\varphi_0, \ldots, \varphi_n)$ of signed formulas such that φ_0 is the initial formula, φ_i is a child of φ_{i-1} and φ_n has no child.

Lemma IV. If S is a path, there exists a hypertree-frame such that S contains exactly one member of every line.

Proof. This is true for the root of the path tree, and if it is true for some path, it is also true for every successor-path.

Lemma V. There is a correspondence between (hypertree-)frames and atomic paths associated with a finite set Φ of formulas; each atomic path is the set of signed atomic formulas of a frame, and the set of signed atomic formulas of each frame is an atomic path.

Proof. By induction on the degree of φ.

Theorem 2. Signed formula φ has a model if and only if (at least) one of its atomic path does not contain an opposite pair.

Proof. Signed formula φ has a model if and only if it has a hypertree-model (theorem 1). The corresponding atomic path (lemma V) is consistent and does not contain an opposite pair.

Corollary. The method is sound and complete.

Conclusion. We are half-way to an extension of the connection method to the decidable relevant logic B^+. The method introduced here inherits most of its properties from the tableau method [2]: soundness, completeness and termination. The next step is to obtain a true connection method, and to investigate its properties by using a matrix-characterization of B^+. Extensions to more powerful systems of relevant logic should also be possible.

REFERENCES

1. A. Bloesch, *Signed Tableaux - A Basis for Automated Theorem Proving in non Classical Logics,* PHD thesis, 1993, University of Queensland
2. A. Bloesch, A Tableau Style Proof System for Two Paraconsistent Logics, *Notre Dame Journal of Formal Logic*, Volume 34, Number 2, pp. 295-301, 1993
3. M. Fitting, *Proof Methods for Modal and Intuitionistic Logics*, Dordrecht Reidel Publishing Company, 1983
4. M. Fitting, Basic Modal Logic, *Handbook of Logic in Artificial Intelligence and Logic Programming* (vol1), Ed. by D. M. Gabbay, C. J. Hogger and J. A. Robinson, Oxford Clarendon Press 1993
5. R. Goré, *Cut-free Sequent and Tableau Systems for Propositional Normal Logic*, Technical report, Cambridge 1992
6. E.P. Gribomont and D. Rossetto, CAVEAT: technique and tool for Computer Aided Verification And Transformation, *Lecture Notes in Computer Science* 939, *Computer Aided Verification*, Springer 1995
7. G.E. Hughes and M.J. Cresswell, *An Introduction to Modal Logic*, London Methuen and Co Ltd 1968, 1972
8. G. Priest and R. Sylvan, Simplified Semantics for Basic Relevant Logics, *Journal of Philosophical Logic* 21: 217-232, 1992
9. G. Restall, Simplified Semantics for Relevant Logics (and some of their rivals), *Journal of Philosophical Logic* 22: 481-511, 1993
10. L. Wallen, Automated Proof Search in Non-classical Logics, *Cambridge MIT Press*, 1990