PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série, tome 82(96) (2007), 55–78 DOI 102298/PIM0796055I

THEOREM PROVERS FOR SUBSTRUCTURAL LOGICS

Mirjana Isaković Ilić

Abstract. We describe theorem provers for some decidable propositional substructural logics.

1. Introduction

Sequent systems for intuitionistic and classical predicate logic, namely LJ and LK, were introduced by Gentzen in [**2**]. The basic expression of those systems is a 'sequent' $\Gamma \vdash \Delta$, where Γ and Δ are finite sequences of formulae. Γ is the antecedent and Δ is the succedent of $\Gamma \vdash \Delta$. If Δ is a single or an empty sequence, then $\Gamma \vdash \Delta$ is a single-conclusion sequent, otherwise, it is a multiple-conclusion sequent.

Each rule in LJ and LK is either an operational rule or a structural rule. *Operational rules* introduce new logical connectives to the left-, or to the right-hand side of the turnstile $\vdash$. On the other hand, *structural* rules refer to the structure of the sequents and they do not involve any logical symbol, of the language to which belong sequent-formulae, in their formulation. Gentzen's structural rules are:

Weakening:

$$\frac{\Gamma \vdash \Delta}{\alpha, \Gamma \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \alpha};$$

Contraction:

$$\frac{\alpha, \alpha, \Gamma \vdash \Delta}{\alpha, \Gamma \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta, \alpha, \alpha}{\Gamma \vdash \Delta, \alpha};$$

Permutation:

$$\frac{\Gamma_1, \alpha, \beta, \Gamma_2 \vdash \Delta}{\Gamma_1, \beta, \alpha, \Gamma_2 \vdash \Delta}, \quad \frac{\Gamma \vdash \Delta_1, \alpha, \beta, \Delta_2}{\Gamma \vdash \Delta_1, \beta, \alpha, \Delta_2};$$

Cut:

$$\frac{\Gamma_1 \vdash \Delta_1, \alpha \qquad \alpha, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}.$$

Substructural logics are logics whose sequent formulations can be obtained from Gentzen's system LK by rejecting or restricting some of Gentzen's structural rules.

2000 *Mathematics Subject Classification*: Primary 03F05; Secondary 03F52.
Key words and phrases: Substructural logics; Automated deduction.

We shall consider both, single- and multiple-conclusion sequent systems[1] for BCK, relevant[2], linear[3], Lambek and Lambek logic with weakening. Furthermore, we shall mention sequent systems for intuitionistic logic[4] and classical logic.

A sequent system for BCK logic can be obtained from the sequent system for classical logic by rejecting contraction, for relevant logic, by rejecting thinning and, for linear logic, by rejecting both thinning and contraction. Since in the presence of permutation, the position of a formula within Γ and within Δ in $\Gamma \vdash \Delta$ is irrelevant, we use *multisets* Γ and Δ, to define a sequent $\Gamma \vdash \Delta$, in the above systems.

A sequent system for Lambek logic is without any structural rules, other than cut.[5] We use *sequences* of formulae Γ and Δ to define $\Gamma \vdash \Delta$ in sequent systems for both Lambek logic and Lambek logic with weakening.

In sequent systems for intuitionistic and classical logics (i.e., in the presence of all Gentzen's structural rules), Γ and Δ in $\Gamma \vdash \Delta$ are *sets* of formulae.

In the absence of some (or all) of Gentzen's structural rules, classically equivalent connectives and constants split into nonequivalent dual pairs. Therefore, the rejection of contraction and/or weakening, produces two conjunctions: additive ($\wedge$) and multiplicative one ($\cdot$). (We may have two implications, also: multiplicative ($\rightarrow$) and additive ($\hookrightarrow$) one; however, in our sequent systems additive implication is definable, therefore it is not considered.) Only in multiple-conclusion sequent systems, we also have two disjunctions: additive ($\vee$) and multiplicative one ($+$). Furthermore, in the absence of permutation, we have two multiplicative implications: $\rightarrow$ and $\leftarrow$ (and accordingly, two definable negations: $\sim\alpha = \alpha \rightarrow 0$ and $\neg\alpha = 0 \leftarrow \alpha$).

In sequent systems without weakening, we have two units, 1 and $\top$ (1 replaces the empty collection of formulae on the left-hand side of $\vdash$) and two zeros, also, 0 and $\bot$ (0 replaces the empty collection of formulae on the right-hand side of $\vdash$). Algebraically, 1 and 0 behave like units for the multiplicatives $\cdot$ and $+$, respectively; $\top$ and $\bot$ behave like greatest and least element in a lattice.

Cut is the only rule of Gentzen-like systems, that allows a formula which appears in premisses to disappear in a conclusion. The formula that disappears (the formula α in the formulation of cut) is called the 'cut formula'. If the cut is an admissible rule in a sequent system (can be added to the system without increasing the stock of provable sequents) and if the application of contraction is controlled, a

[1]We shall say that a logic is a single-conclusion logic, if in its sequent formulation, starting with single-conclusion sequents, only single-conclusion sequents can be derived; a logic is a multiple-conclusion logic, if in its sequent formulation, multiple-conclusion sequents can be derived.

[2]By relevant logic we mean relevant logic without distribution.

[3]By linear logic we mean *pure* (i.e., without exponentials) linear logic.

[4]Intuitionistic logic may be considered as substructural logic, also: its sequent formulation may be obtained from LK by restricting thinning on the right, see [**4**]. Gentzen formulated a sequent system for intuitionistic logic, namely LJ, by restricting the language to single-conclusion sequents.

[5]Abrusci [**1**] investigated Lambek logic from the point of view of linear logic: Lambek logic is *pure* (i.e., without exponentials), *non-commutative* linear propositional logic.

decision procedure for deciding of a sequent, whether or not it is provable, can be formulated.

Therefore, the cut-elimination theorem is the fundamental theorem of sequent systems.

THEOREM 0 (Cut-Elimination Theorem). *Every derivation in a sequent system* GS*, can be transformed into a derivation in* GS*, with the same endsequent and in which no cut occurs.*

Unfortunately it does not hold for multiple-conclusion sequent system for both Lambek logic and Lambek logic with weakening. This means that there are sequents, which are derivable with, but not without cut. However, applications of cut in these two systems can be controlled, enabling the establishing of the decision procedure (see [**3**]).

We shall illustrate our tableau-based proof procedure for linear, BCK and relevant logic. First we give their sequent systems and prove soundness and completeness results with respect to corresponding algebraic structures (Section 2).

Tableau systems make the third section. In Section 4, we give the algorithm for automated theorem proving.

2. Sequent systems for linear, BCK and relevat logic

Let $\mathcal{L}$ be the language of the propositional calculus with the propositional constants $\bot$, $\top$, 1 and 0 and binary connectives $\rightarrow$, $\leftarrow$, $\cdot$, $+$, $\wedge$ and $\vee$. Formulae of $\mathcal{L}$ are defined inductively, as usual. We use α, β, $\gamma, \varphi, \ldots, \alpha_1, \ldots$ as schematic letters for formulae of $\mathcal{L}$. As usual, $\sim\alpha$ is defined as an abbreviation of $\alpha \rightarrow 0$ and $\neg\alpha$ as an abbreviation of $0 \leftarrow \alpha$. The vocabulary of $\mathcal{L}$ plus the comma and the turnstile $\vdash$, makes the vocabulary of the sequent language $\mathcal{G}$. Our sequent systems will be formulated in $\mathcal{G}$.

CBC is the multiple-conclusion sequent system for linear logic, with the following postulates (on the left- and on the right-hand side of $\vdash$ are multisets of formulae):

axioms:

$\alpha \vdash \alpha$ (Id)

$\vdash 1$ (1 r) $\qquad \Gamma \vdash \top, \Delta$ ($\top$)

$0 \vdash$ (0 l) $\qquad \Gamma, \bot \vdash \Delta$ ($\bot$)

structural rule (cut):

$$\frac{\Gamma_1 \vdash \Delta_1, \alpha, \Delta_2 \qquad \Gamma_2, \alpha, \Gamma_3 \vdash \Delta_3}{\Gamma_2, \Gamma_1, \Gamma_3 \vdash \Delta_1, \Delta_3, \Delta_2}\ (\text{cut})$$

rules for propositional constants and connectives:

$$\frac{\Gamma \vdash \Delta}{\Gamma, 1 \vdash \Delta}\ (1\ \text{l}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash 0, \Delta}\ (0\ \text{r})$$

$$\frac{\Gamma_1 \vdash \alpha, \Delta_1 \qquad \Gamma_2, \beta \vdash \Delta_2}{\Gamma_1, \Gamma_2, \alpha \rightarrow \beta \vdash \Delta_1, \Delta_2}\ (\rightarrow\ \text{l}) \qquad \frac{\Gamma, \alpha \vdash \beta, \Delta}{\Gamma \vdash \alpha \rightarrow \beta, \Delta}\ (\rightarrow\ \text{r})$$

$$\frac{\Gamma,\alpha,\beta\vdash\Delta}{\Gamma,\alpha\cdot\beta\vdash\Delta}\ (\cdot\ \mathrm{l}) \qquad \frac{\Gamma_1\vdash\alpha,\Delta_1 \qquad \Gamma_2\vdash\beta,\Delta_2}{\Gamma_1,\Gamma_2\vdash\alpha\cdot\beta,\Delta_1,\Delta_2}\ (\cdot\ \mathrm{r})$$

$$\frac{\Gamma_1,\alpha\vdash\Delta_1 \qquad \Gamma_2,\beta\vdash\Delta_2}{\Gamma_1,\Gamma_2,\alpha+\beta\vdash\Delta_1,\Delta_2}\ (+\ \mathrm{l}) \qquad \frac{\Gamma\vdash\alpha,\beta,\Delta}{\Gamma\vdash\alpha+\beta,\Delta}\ (+\ \mathrm{r})$$

$$\frac{\Gamma,\alpha\vdash\Delta}{\Gamma,\alpha\wedge\beta\vdash\Delta} \qquad \frac{\Gamma,\beta\vdash\Delta}{\Gamma,\alpha\wedge\beta\vdash\Delta}\ (\wedge\ \mathrm{l}) \qquad \frac{\Gamma\vdash\alpha,\Delta \qquad \Gamma\vdash\beta,\Delta}{\Gamma\vdash\alpha\wedge\beta,\Delta}\ (\wedge\ \mathrm{r})$$

$$\frac{\Gamma,\alpha\vdash\Delta \qquad \Gamma,\beta\vdash\Delta}{\Gamma,\alpha\vee\beta\vdash\Delta}\ (\vee\ \mathrm{l}) \qquad \frac{\Gamma\vdash\alpha,\Delta}{\Gamma\vdash\alpha\vee\beta,\Delta} \qquad \frac{\Gamma\vdash\beta,\Delta}{\Gamma\vdash\alpha\vee\beta,\Delta}\ (\vee\ \mathrm{r})$$

The postulates of CBC_{K^c}, which is the multiple-conclusion sequent system for BCK logic, is obtained when we add the structural rules of weakening, i.e., the rules from the set $K^c = \left\{ \frac{\Gamma\vdash\Delta}{\Gamma,\alpha\vdash\Delta}\ (k\ \mathrm{l}), \frac{\Gamma\vdash\Delta}{\Gamma\vdash\alpha,\Delta}\ (k\ \mathrm{r}) \right\}$ to CBC. However, in the presence of weakening, $1\vdash\top$, $\top\vdash 1$, $0\vdash\bot$ and $\bot\vdash 0$ are derivable sequents, therefore $\top$ and $\bot$, together with the axioms $(\top)$ and $(\bot)$, may be omitted in CBC_{K^c}.

The postulates of CBC_{W^c}, which is the multiple-conclusion sequent system for relevant logic, is obtained when we add the structural rules of contraction, i.e., the rules from the set $W^c = \left\{ \frac{\Gamma,\alpha,\alpha\vdash\Delta}{\Gamma,\alpha\vdash\Delta}\ (w\ \mathrm{l}), \frac{\Gamma\vdash\alpha,\alpha,\Delta}{\Gamma\vdash\alpha,\Delta}\ (w\ \mathrm{r}) \right\}$ to CBC.

Single-conclusion sequent systems for linear, BCK and relevant logic are obtained from the above sequent systems by the restriction to at most one formula in the succedent (note that these systems are without the connective $+$).

We shall prove the completeness and soundness results with respect to following algebras.

Definition 1. A structure $\mathbf{A} = \langle A, \to, \cdot, +, \wedge, \vee, 1, 0, \top, \bot\rangle$ is a CBC-algebra iff the set A is closed under the binary operation $\to$ and:

1. $\langle A, \wedge, \vee, \top, \bot\rangle$ is a lattice with the least element $\bot$ and the greatest element $\top$ for which $\top = \bot\to\bot$,
2. $\langle A, \cdot, 1\rangle$ and $\langle A, +, 0\rangle$ are commutative monoids with the identities 1 and 0, respectively,
3. $x\cdot(y\vee z) = x\cdot y\vee x\cdot z$, $\quad x+(y\wedge z) = x+y\wedge x+z$,
$(x\vee y)\cdot z = x\cdot z\vee y\cdot z$, $\quad (x\wedge y)+z = x+z\wedge y+z$, for every $x, y, z\in A$,
4. $y\cdot x \leqslant z+t$ iff $x\leqslant y\to z+t$, for every $x, y, z, t\in A$, where $a\leqslant b$ is an abbreviation for $a = a\wedge b$.

A CBC-algebra $\mathbf{A}$ satisfying $x\cdot y\leqslant x$, $y\cdot x\leqslant x$, $x\leqslant x+y$, and $y\leqslant x+y$, for every $x, y\in A$, is a CBC_{K^c}-algebra. A CBC-algebra $\mathbf{A}$ satisfying $x\leqslant x\cdot x$ and $x+x\leqslant x$, for every $x\in A$, is a CBC_{W^c}-algebra.

It is easy to show that $\leqslant$ is a partial order on A.

Algebraic models are defined as follows.

Definition 2. Let V be the set of propositional variables of $\mathcal{L}$ and let F be the set of formulae of $\mathcal{L}$. For a CBC-algebra $\mathbf{A} = \langle A, \to, \cdot, +, \wedge, \vee, 1, 0, \top, \bot\rangle$ and a basic valuation $v_0 : V\to A$, we define the valuation v, as follows:

1. for every $\alpha \in F$, we define $v(\alpha) \in A$, as follows:

$$
\begin{array}{cc}
v(p) = v_0(p), \quad p \in V & v(\alpha \to \beta) = v(\alpha) \to v(\beta) \\
v(\alpha \cdot \beta) = v(\alpha) \cdot v(\beta) & v(\alpha + \beta) = v(\alpha) + v(\beta) \\
v(\alpha \vee \beta) = v(\alpha) \vee v(\beta) & v(\alpha \wedge \beta) = v(\alpha) \wedge v(\beta) \\
v(1) = 1 & v(0) = 0 \\
v(\top) = \top & v(\bot) = \bot
\end{array}
$$

2. for every $\Gamma \vdash \Delta$, we define $v(\Gamma \vdash \Delta)$ as follows ($n \geqslant 1$, $m \geqslant 1$):

$$
\begin{array}{rcl}
v(\gamma_1, \ldots, \gamma_n \vdash \delta_1, \ldots, \delta_m) & \text{is} & v(\gamma_1 \cdot \ldots \cdot \gamma_n) \leqslant v(\delta_1 + \cdots + \delta_m) \\
v(\gamma_1, \ldots, \gamma_n \vdash) & \text{is} & v(\gamma_1 \cdot \ldots \cdot \gamma_n) \leqslant 0 \\
v(\vdash \delta_1, \ldots, \delta_m) & \text{is} & 1 \leqslant v(\delta_1 + \cdots + \delta_m) \\
v(\vdash) & \text{is} & 1 \leqslant 0.
\end{array}
$$

Definition 3. A CBC-model is $\langle \mathbf{A}, v \rangle$, where $\mathbf{A}$ is a CBC-algebra and v is a valuation.

By induction on the length of proof of $\Gamma \vdash \Delta$ in CBC, we can prove the following soundness lemma for CBC:

Lemma 1. *If $\Gamma \vdash \Delta$ is provable in* CBC*, then $v(\Gamma \vdash \Delta)$ holds in every* CBC-*model.*

Proof. Let $\langle \mathbf{A}, v \rangle$ be a CBC-model and let the length of proof of $\Gamma \vdash \Delta$ in CBC be 1. Then $\Gamma \vdash \Delta$ is an axiom. If it is of the form $\gamma_1, \ldots, \gamma_n, \bot \vdash \delta_1, \ldots, \delta_m$, where $n \geqslant 1$ and $m \geqslant 1$, and if $v(\gamma_i) = a_i$, for every $i \in \{1, \ldots, n\}$, and $v(\delta_j) = b_j$, for every $j \in \{1, \ldots, m\}$, then we wish to show that $a_1 \cdot \ldots \cdot a_n \cdot \bot \leqslant b_1 + \cdots + b_m$ holds in $\mathbf{A}$. However, for $a_1 \cdot \ldots \cdot a_n = a \in A$ and $b_1 + \cdots + b_m = b \in A$ we have $a \leqslant \bot \to \bot$, i.e., $\bot \cdot a \leqslant \bot$ and $a \cdot \bot \leqslant \bot$ (since $\cdot$ is commutative) and $\bot \leqslant b$, therefore $a \cdot \bot \leqslant b$ (since $\leqslant$ is transitive). We proceed analogously when $n = 0$ and/or $m = 0$ and also, when $\Gamma \vdash \Delta$ is any other axiom in CBC.

Suppose the length of proof of $\Gamma \vdash \Delta$ in CBC is greater than 1. Then $\Gamma \vdash \Delta$ is the lower sequent of an inference rule, in our proof. If $\Gamma \vdash \Delta$ is the lower sequent of the rule (1 l): $\dfrac{\Gamma_1 \vdash \Delta}{\Gamma_1, 1 \vdash \Delta}$ (1 l), then, for $\Gamma_1 = \gamma_1, \ldots, \gamma_n$, $n \geqslant 1$ and $\Delta = \delta_1, \ldots, \delta_m$, $m \geqslant 1$, and $v(\gamma_i) = a_i$, for every $i \in \{1, \ldots, n\}$, and $v(\delta_j) = b_j$, for every $j \in \{1, \ldots, m\}$, we wish to show that $a_1 \cdot \ldots \cdot a_n \cdot 1 \leqslant b_1 + \cdots + b_m$. However, by the induction hypothesis it follows that $a_1 \cdot \ldots \cdot a_n \leqslant b_1 + \cdots + b_m$. Now, since $\langle A, \cdot, 1 \rangle$ is a commutative monoid and $a_1 \cdot \ldots \cdot a_n = a \in A$ and $b_1 + \cdots + b_m = b \in A$, we have $a \cdot 1 \leqslant b$. We proceed analogously when $n = 0$ and/or $m = 0$ and also, when, in our proof, $\Gamma \vdash \Delta$ is the lower sequent of any other inference rule. $\square$

The completeness result for CBC will be proved with respect to the Lindenbaum algebra of CBC, as follows. Let $|\alpha| = \{\beta : \alpha \vdash \beta \text{ and } \beta \vdash \alpha \text{ are provable in CBC}\}$ and let $|\alpha| * |\beta| = |\alpha * \beta|$, for every $* \in \{\to, \cdot, +, \vee, \wedge\}$. Note that the set $|\alpha * \beta|$, where $* \in \{\to, \cdot, +, \vee, \wedge\}$, does not depend on the representatives of $|\alpha|$ and $|\beta|$. We define $|\alpha| \leqslant |\beta|$ as $|\alpha| = |\alpha| \wedge |\beta|$.

Let $\pounds C = \{|\alpha| : \alpha \in F\}$. The set $\pounds C$ is closed under the set operations $\to$, $\cdot$, $+$, $\vee$ and $\wedge$ ($|\alpha| * |\beta| = |\alpha * \beta| \in \pounds C$, for every $* \in \{\to, \cdot, +, \vee, \wedge\}$).

Lemma 2. *$\alpha \vdash \beta$ is provable in* CBC *iff $|\alpha| \leqslant |\beta|$ in $\pounds C$.*

Proof. $\alpha \vdash \beta$ iff

1.	$\alpha \vdash \alpha \wedge \beta$ and $\alpha \wedge \beta \vdash \alpha$	(applications of ($\wedge$ l), ($\wedge$ r) and cut)
2.	$\lvert\alpha\rvert = \lvert\alpha \wedge \beta\rvert$	(1)
3.	$\lvert\alpha \wedge \beta\rvert = \lvert\alpha\rvert \wedge \lvert\beta\rvert$	(definition)
4.	$\lvert\alpha\rvert = \lvert\alpha\rvert \wedge \lvert\beta\rvert$	(2 and 3)
5.	$\lvert\alpha\rvert \leqslant \lvert\beta\rvert$	(4, definition)

□

Definition 4. $\mathit{LindC} = \langle \pounds C, \to, \cdot, +, \vee, \wedge, |1|, |0|, |\top|, |\bot| \rangle$ is the Lindenbaum algebra of CBC.

Lemma 3. *The Lindenbaum algebra of* CBC *is a* CBC-*algebra.*

Proof. The set $\pounds C$ is closed under $\to$, $\cdot$, $+$, $\vee$ and $\wedge$, $|1| \in \pounds C$ and $|0| \in \pounds C$, therefore it is sufficient to prove that LindC satisfies the axioms of a CBC-algebra. For example:

$\lvert\alpha\rvert \leqslant \lvert\beta\rvert \to \lvert\gamma\rvert + \lvert\delta\rvert$	iff	1.	$\lvert\alpha\rvert \leqslant \lvert\beta \to \gamma + \delta\rvert$	(definition)
		2.	$\alpha \vdash \beta \to \gamma + \delta$	(1, Lemma 2)
		3.	$\beta \cdot \alpha \vdash \gamma + \delta$	(2, definition)
		4.	$\lvert\beta \cdot \alpha\rvert \leqslant \lvert\gamma + \delta\rvert$	(3, Lemma 2)
		5.	$\lvert\beta\rvert \cdot \lvert\alpha\rvert \leqslant \lvert\gamma\rvert + \lvert\delta\rvert$	(4, definition).

We proceed analogously for all other axioms of a CBC-algebra. □

Now, we can prove the completeness lemma for CBC.

Lemma 4. *If $v(\Gamma \vdash \Delta)$ is valid in every* CBC-*model, then $\Gamma \vdash \Delta$ is provable in* CBC.

Proof. Let $v(\Gamma \vdash \Delta)$ be valid in every CBC-model. Noting that a Lindenbaum algebra of CBC is a CBC-algebra, we define the mapping v as follows:

$$\begin{aligned} v(\alpha) &= |\alpha|, \quad \text{for every } \alpha \in F \\ v(\gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m) &\quad \text{is} \quad |\gamma_1 \cdot \ \dots \ \cdot \gamma_n| \leqslant |\delta_1 + \dots + \delta_m| \\ v(\gamma_1, \dots, \gamma_n \vdash) &\quad \text{is} \quad |\gamma_1 \cdot \ \dots \ \cdot \gamma_n| \leqslant |0| \\ v(\vdash \delta_1, \dots, \delta_m) &\quad \text{is} \quad |1| \leqslant |\delta_1 + \dots + \delta_m| \\ v(\vdash) &\quad \text{is} \quad |1| \leqslant |0|. \end{aligned}$$

The mapping v is well defined and it is a valuation, since it satisfies every condition from the definition of the valuation. Therefore, $\langle \mathit{LindC}, v \rangle$ is a CBC-model.

We show that if $v(\gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m)$, for $n \geqslant 1$ and $m \geqslant 1$, is valid in $\langle LindC, v\rangle$, then $\gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m$ is provable in CBC.

Suppose that $v(\gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m)$ is valid in $\langle LindC, v\rangle$. Then:

1.	$\lvert\gamma_1 \cdot \ldots \cdot \gamma_n\rvert \leqslant \lvert\delta_1 + \cdots + \delta_m\rvert$	(definition)
2.	$\lvert\gamma_1 \cdot \ldots \cdot \gamma_n\rvert = \lvert\gamma_1 \cdot \ldots \cdot \gamma_n\rvert \wedge \lvert\delta_1 + \cdots + \delta_m\rvert$	(1)
3.	$\lvert\gamma_1 \cdot \ldots \cdot \gamma_n\rvert = \lvert(\gamma_1 \cdot \ldots \cdot \gamma_n) \wedge (\delta_1 + \cdots + \delta_m)\rvert$	(2)
4.	$\gamma_1 \cdot \ldots \cdot \gamma_n \vdash (\gamma_1 \cdot \ldots \cdot \gamma_n) \wedge (\delta_1 + \cdots + \delta_m)$ and $(\gamma_1 \cdot \ldots \cdot \gamma_n) \wedge (\delta_1 + \cdots + \delta_m) \vdash \gamma_1 \cdot \ldots \cdot \gamma_n$	(3, Lemma 2)
5.	$\gamma_1 \cdot \ldots \cdot \gamma_n \vdash \delta_1 + \cdots + \delta_m$	(4)
6.	$\gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m$	(5, cut).

We proceed analogously when $v(\Gamma \vdash)$ is valid in $\langle LindC, v\rangle$, where Γ is a non-empty multiset, or when $v(\vdash \Delta)$ is valid in $\langle LindC, v\rangle$, where Δ is a non-empty multiset (note that $v(\vdash)$ is not valid in $\langle LindC, v\rangle$). □

We prove the soundness and completeness results for any other sequent system in a perfectly analogous way.

Cut is an admissible rule in CBC, CBC_{K^c}, CBC_{W^c} and in their single-conclusion variants. (In the absence of contraction, to show that every proof with a single cut, which is the last rule of the proof, can be transformed, in the same system, to a proof with the same endsequent without cuts, we proceed by induction on $\langle d, \rho\rangle$, lexicographically ordered, where d is the degree and ρ is the rank of our cut; in CBC_{W^c}, instead of cut, we eliminate mix, as above, similarly as in [**2**].) Therefore, in our systems, every sequent which is derivable with, is also derivable without cut. We shall use this fact in formulating corresponding tableau systems.

Permutation is only implicit rule in CBC, CBC_{K^c}, CBC_{W^c} and in their single-conclusion variants. Next, we shall define the system CBCK (and its single conclusion variant) which is equivalent to CBC_{K^c}, where weakening is implicit, too.

First we shall prove the following lemma.

LEMMA 5. *Every proof in* CBC_{K^c} *can be transformed into a proof with the same endsequent in* CBC_{K^c}*, where the upper sequent of every weakening is either an axiom or the lower sequent of another weakening.*

PROOF. Let S be the initial segment of a proof in CBC_{K^c}, with the single application of weakening, which is the last rule in S. By a straightforward induction on the length of S we can prove that S can be transformed into the proof with the same endsequent in CBC_{K^c}, where none application of weakening is immediately preceded either by a rule for a connective or by a rule for a constant. For example, if weakening is immediately preceded by a rule $(\rightarrow \mathrm{r})$ in S, we shall use the following reduction step:

$$\dfrac{\dfrac{\overset{\pi}{\alpha, \Gamma \vdash \beta, \Delta}}{\Gamma \vdash \alpha \rightarrow \beta, \Delta}\,(\rightarrow\ \mathrm{r})}{\Gamma \vdash \alpha \rightarrow \beta, \Delta, \varphi}\,(k\ \ \mathrm{r}) \quad \mapsto \quad \dfrac{\dfrac{\overset{\pi}{\alpha, \Gamma \vdash \beta, \Delta}}{\alpha, \Gamma \vdash \beta, \Delta, \varphi}\,(k\ \ \mathrm{r})}{\Gamma \vdash \alpha \rightarrow \beta, \Delta, \varphi}\,(\rightarrow\ \mathrm{r})$$

We proceed similarly when the upper sequent of weakening is the lower sequent of either any other rule for a connective or a rule for a constant in CBC_{K^c}. □

Definition 5. CBCK is the multiple-conclusion sequent system for BCK logic, whose postulates are the following axioms:

$$\Gamma, \alpha \vdash \alpha, \Delta \quad (\mathrm{Id})$$
$$\Gamma \vdash 1, \Delta \quad (1\ \mathrm{r})$$
$$\Gamma, 0 \vdash \Delta \quad (0\ \mathrm{l})$$

together with the rules for constants and the rules for connectives of the system CBC.

Directly from Lemma 5 and the Cut-Elimination Theorem for CBC_{K^c} it follows that CBC_{K^c} and CBCK are equivalent systems (and their single conclusion variants, also). Furthermore we note that every *upward derivation* of $\Gamma \vdash \Delta$ (where we start from $\Gamma \vdash \Delta$ and apply the rules of the sequent system from the bottom up) is finite and the total number of them is finite. Therefore CBCK, CBC and their single-conclusion variants, are decidable systems: $\Gamma \vdash \Delta$ is provable iff at least one of the attempted upward derivations is the proof. We should note that the same decision procedure can be applied to every cut-free, contractionless sequent system, à la Gentzen, which is either without permutation or the permutation is implicit in it.

Now we shall define the system CBCW, which is equivalent to CBC_{W^c}, where contraction is implicit (as well as permutation) and we shall show that CBCW is decidable. First, we shall limit applications of contraction (preserving soundness and completeness of CBC_{W^c}) according to Kripke's idea [**5**] to allow a contraction of the conclusion of a rule, only if the same sequent could not be obtained by contracting the premises only.

We shall use limCBC_{W^c} to denote the system CBC_{W^c}, where contraction is limited as above. To show that limCBC_{W^c} is decidable system, we shall use techniques from [**6**], where the Kripke's idea has been exploited in formulating the decision procedure for a single-conclusion sequent system for relevant logic.

It is easy to show that every sequent which is derivable in CBC_{W^c} is also derivable in limCBC_{W^c}, and vice versa. Therefore these two systems are equivalent.

It will be illustrative to see how some (upward) derivations in limCBC_{W^c} look like. For example, we construct the proof of $\alpha \to (\alpha \to \beta) \vdash \alpha \to \beta$ in limCBC_{W^c}, as follows:

$$\dfrac{\dfrac{\dfrac{\alpha \vdash \alpha \qquad \dfrac{\alpha \vdash \alpha \quad \beta \vdash \beta}{\alpha, \alpha \to \beta \vdash \beta}\,(\to \mathrm{l})}{\alpha, \alpha, \alpha \to (\alpha \to \beta) \vdash \beta}\,(\to \mathrm{l})}{\alpha, \alpha \to (\alpha \to \beta) \vdash \beta}\,(w\ \mathrm{l})}{\alpha \to (\alpha \to \beta) \vdash \alpha \to \beta}\,(\to \mathrm{r})$$

On the other hand, we can prove (using only a finite number of steps in every upward derivation) that $\alpha \cdot \alpha \vdash \alpha \wedge \alpha$ is not provable in limCBC_{W^c}.

$\alpha \cdot \alpha \vdash \alpha \wedge \alpha$ may be:

1. the conclusion of the rule $(\cdot\ \mathrm{l})$, of the form: $\dfrac{\alpha, \alpha \vdash \alpha \wedge \alpha}{\alpha \cdot \alpha \vdash \alpha \wedge \alpha}\,(\cdot\ \mathrm{l})$

2. the conclusion of the contraction of the form: $\dfrac{\dfrac{\alpha\cdot\alpha,\alpha,\alpha\vdash\alpha\wedge\alpha}{\alpha\cdot\alpha,\alpha\cdot\alpha\vdash\alpha\wedge\alpha}\,(\cdot\ \mathrm{l})}{\alpha\cdot\alpha\vdash\alpha\wedge\alpha}\,(w\ \mathrm{l})$

3. the conclusion of the rule $(\wedge\ \mathrm{r})$, of the form: $\dfrac{\alpha\cdot\alpha\vdash\alpha \qquad \alpha\cdot\alpha\vdash\alpha}{\alpha\cdot\alpha\vdash\alpha\wedge\alpha}\,(\wedge\ \mathrm{r})$

4. the conclusion of the contraction of the form: $\dfrac{\dfrac{\alpha\cdot\alpha\vdash\alpha,\alpha\wedge\alpha \qquad \alpha\cdot\alpha\vdash\alpha,\alpha\wedge\alpha}{\alpha\cdot\alpha\vdash\alpha\wedge\alpha,\alpha\wedge\alpha}\,(\wedge\ \mathrm{r})}{\alpha\cdot\alpha\vdash\alpha\wedge\alpha}\,(w\ \mathrm{r})$.

The derivation 1, may be continued as follows:

1.1. $\dfrac{\dfrac{\alpha,\alpha\vdash\alpha \qquad \alpha,\alpha\vdash\alpha}{\alpha,\alpha\vdash\alpha\wedge\alpha}\,(\wedge\ \mathrm{r})}{\alpha\cdot\alpha\vdash\alpha\wedge\alpha}\,(\cdot\ \mathrm{l})$ 1.2. $\dfrac{\dfrac{\dfrac{\alpha,\alpha\vdash\alpha,\alpha\wedge\alpha \qquad \alpha,\alpha\vdash\alpha,\alpha\wedge\alpha}{\alpha,\alpha\vdash\alpha\wedge\alpha,\alpha\wedge\alpha}\,(\wedge\ \mathrm{r})}{\alpha,\alpha\vdash\alpha\wedge\alpha}\,(w\ \mathrm{r})}{\alpha\cdot\alpha\vdash\alpha\wedge\alpha}\,(\cdot\ \mathrm{l})$

Since $\alpha,\alpha\vdash\alpha$ in 1.1, cannot be the lower sequent of either any rule for a constant or any rule for a connective in limCBC$_{W^c}$ (for $\alpha\neq 1$) and since it is not an axiom, the derivation at 1.1 is finished. $\alpha\cdot\alpha\vdash\alpha\wedge\alpha$ is not proved.

In the derivation 1.2, $\alpha,\alpha\vdash\alpha,\alpha\wedge\alpha$ can only be the lower sequent of the rule $(\wedge\ \mathrm{r})$, of the form: $\dfrac{\alpha,\alpha\vdash\alpha,\alpha \qquad \alpha,\alpha\vdash\alpha,\alpha}{\alpha,\alpha\vdash\alpha,\alpha\wedge\alpha}\,(\wedge\ \mathrm{r})$ (in limCBC$_{W^c}$ it cannot be the endsequent of the derivation: $\dfrac{\dfrac{\alpha,\alpha\vdash\alpha,\alpha,\alpha\wedge\alpha \quad \alpha,\alpha\vdash\alpha,\alpha,\alpha\wedge\alpha}{\alpha,\alpha\vdash\alpha,\alpha\wedge\alpha,\alpha\wedge\alpha}\,(\wedge\ \mathrm{r})}{\alpha,\alpha\vdash\alpha,\alpha\wedge\alpha}\,(w\ \mathrm{r})$), therefore the derivation 1.2 cannot be the endsegment of any proof.

We saw that every derivation in limCBC$_{W^c}$ which begins like 1, finishes (after the finite number of steps) without giving the proof. We proceed analogously in all other cases.

Now we shall show that every cut-free derivation in limCBC$_{W^c}$ finishes after the finite number of steps, and that there is only the finite number of them, i.e., we shall show that limCBC$_{W^c}$ is decidable.

We shall say that $\Gamma'\vdash\Delta'$ and $\Gamma\vdash\Delta$, where Γ', Δ', Γ and Δ are multisets of formulae, are *cognate* iff exactly the same formulae occur in Γ and Γ', and exactly the same formulae occur in Δ and Δ' (for example, $\alpha\vdash\beta$, $\alpha,\alpha\vdash\beta$ and $\alpha\vdash\beta,\beta$ are three different, cognate sequents). We shall call the class of all sequents cognate with a given sequent, *a cognation class.*

First we note that limCBC$_{W^c}$ has the *subformula property*, saying that for any provable sequent $\Gamma\vdash\Delta$, there is a proof of $\Gamma\vdash\Delta$ in limCBC$_{W^c}$ (a cut-free proof), such that all formulae occurring in this proof are subformulae of formulae from $\Gamma\vdash\Delta$. Therefore, for every provable sequent, there is a proof (at least one cut-free proof) where only the finite number of cognation classes occur. Now it is clear that if we want to show that every upward derivation in limCBC$_{W^c}$ is finite, it is enough to show that only the finite number of sequents from each cognation class, can be derived in every branch. Therefore, it is enough to prove the following lemma.

LEMMA 6. *Let $\sigma = S_0, S_1, \ldots$ be the sequence of cognate sequents. If σ is W-normal in the sense: for no $i<j$, S_i is the contraction of S_j, then σ is finite.*

Proof. Let $\sigma = S_0, S_1, \ldots$ be the W-normal sequence of cognate sequents, and let $\gamma_1, \gamma_2, \ldots, \gamma_s$ be all formulae which occur in σ. Let $\sigma' = S'_0, S'_1, \ldots$ be the sequence of sequents, which is obtained when we rename the formulae from σ, as follows: every appearance of γ_1, in the antecedent of a sequent, is substituted for α_1, and in the succedent of a sequent, for β_1; we proceed similarly for $\gamma_2, \ldots, \gamma_s$. We note that all formulae, which occur in σ' are $\alpha_1, \ldots \alpha_l, \beta_1, \ldots, \beta_d$, $l + d = n \leqslant 2s$.

σ' is the W-normal sequence of cognate sequents. We wish to prove that σ' is finite. We proceed by the induction on n (n is the number of different formulae occurring in σ'). Let $n = 1$. Then the claim is immediate, since the number of sequents, which belong to a W-normal sequence and which are built of repeated occurrences of only one formula, is finite.

Let $n > 1$ and let φ be any formula from σ'. Let $\sigma''(\varphi) = D_0, D_2, \ldots$ be the sequence of sequents from σ', which is defined as follows. Let S'_k be the first sequent from the left in σ' that contains φ. Then $D_0 = S'_k$; if D_m is defined and if $D_m = S'_j$, then D_{m+1} is the first sequent from the left in $S'_{j+1}, S'_{j+2}, \ldots$, in which the number of occurrences of φ is greater than or equal to the number of occurrences of φ in D_m. (For example, if σ' is the sequence of sequents: $\alpha_1, \alpha_1, \alpha_2 \vdash \beta_1, \beta_2, \beta_2, \quad \alpha_1, \alpha_2, \alpha_2 \vdash \beta_1, \beta_1, \beta_2, \quad \alpha_1, \alpha_2, \alpha_2 \vdash \beta_1, \beta_2, \beta_2$, then $\sigma''(\alpha_1) = \alpha_1, \alpha_1, \alpha_2 \vdash \beta_1, \beta_2, \beta_2$; $\sigma''(\alpha_2) = \sigma'$.)

We wish to prove that $\sigma''(\varphi)$ is finite. Let σ''_φ be the sequence of sequents which is obtained by deleting all occurrences of φ in $\sigma''(\varphi)$. Since σ''_φ is the W-normal sequence of cognate sequents (σ' is W-normal), by the induction hypothesis it follows that σ''_φ is finite. Therefore, $\sigma''(\varphi)$ is finite, and so is σ'. $\square$

We should note that this algorithm could not be applied to sequent systems with contraction in the absence of permutation, since the subformula property does not hold there (cut is not an admissible rule there).

Now we give the postulates of the system CBCW, whose derivations are defined from the bottom up. We shall use Θ^1 to denote the multiset of formulae from Θ, which appear exactly once in Θ.

Axioms of CBCW *are the axioms of the system* CBC.
Rules for propositional constants and connectives in CBCW *are:*

$$\frac{\Gamma \vdash \Delta}{\Gamma, 1 \vdash \Delta}\ (1\ \mathrm{l}) \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash 0, \Delta}\ (0\ \mathrm{r})$$

$$\frac{\Theta_1 \vdash \alpha, \Lambda_1 \qquad \Theta_2, \beta \vdash \Lambda_2}{\Theta, \alpha \to \beta \vdash \Lambda}\ (\to\ \mathrm{l}) \qquad\qquad \frac{\Gamma, \alpha \vdash \beta, \Omega_1}{\Gamma \vdash \alpha \to \beta, \Omega}\ (\to\ \mathrm{r})$$

$$\frac{\Omega_1, \alpha, \beta \vdash \Delta}{\Omega, \alpha \cdot \beta \vdash \Delta}\ (\cdot\ \mathrm{l}) \qquad\qquad \frac{\Lambda_1 \vdash \alpha, \Theta_1 \qquad \Lambda_2 \vdash \beta, \Theta_2}{\Lambda \vdash \alpha \cdot \beta, \Theta}\ (\cdot\ \mathrm{r})$$

$$\frac{\Theta_1, \alpha \vdash \Lambda_1 \qquad \Theta_2, \beta \vdash \Lambda_2}{\Theta, \alpha + \beta \vdash \Lambda}\ (+\ \mathrm{l}) \qquad\qquad \frac{\Gamma \vdash \alpha, \beta, \Omega_1}{\Gamma \vdash \alpha + \beta, \Omega}\ (+\ \mathrm{r})$$

$$\frac{\Omega_1, \alpha \vdash \Delta}{\Omega, \alpha \wedge \beta \vdash \Delta} \qquad \frac{\Omega_1, \beta \vdash \Delta}{\Omega, \alpha \wedge \beta \vdash \Delta}\ (\wedge\ \mathrm{l}) \qquad\qquad \frac{\Gamma \vdash \alpha, \Omega_1 \qquad \Gamma \vdash \beta, \Omega_1}{\Gamma \vdash \alpha \wedge \beta, \Omega}\ (\wedge\ \mathrm{r})$$

$$\frac{\Omega_1, \alpha \vdash \Delta \qquad \Omega_1, \beta \vdash \Delta}{\Omega, \alpha \vee \beta \vdash \Delta}\ (\vee\ \mathrm{l}) \qquad \frac{\Gamma \vdash \alpha, \Omega_1}{\Gamma \vdash \alpha \vee \beta, \Omega} \qquad \frac{\Gamma \vdash \beta, \Omega_1}{\Gamma \vdash \alpha \vee \beta, \Omega}\ (\vee\ \mathrm{r})$$

– $\Theta_1, \Theta_2 = \Theta$
– $\Theta_1 = \Sigma, \Theta'$, $\quad \Theta_2 = \Sigma, \Theta''$,
$\Theta = \Sigma, \Theta', \Theta''$, $\quad \Sigma \subseteq \Theta^1$
– $\Theta_1, \Theta_2 = \Theta, \alpha * \beta$,
provided that $\alpha * \beta$ doesn't appear in Θ
– $\Theta_1 = \Sigma, \Theta'$, $\quad \Theta_2 = \Sigma, \Theta''$,
$\Theta, \alpha * \beta = \Sigma, \Theta', \Theta''$, $\quad \Sigma \subseteq \Theta^1$,
provided that $\alpha * \beta$ doesn't appear in Θ
– $\Theta_1 = \alpha * \beta, \Sigma, \Theta'$, $\quad \Theta_2 = \alpha * \beta, \Sigma, \Theta''$,
$\Theta = \Sigma, \Theta', \Theta''$, $\quad \Sigma \subseteq \Theta^1$,
provided that $\alpha * \beta$ doesn't appear in Θ

– $\Lambda_1, \Lambda_2 = \Lambda$
– $\Lambda_1 = \Sigma, \Lambda'$, $\quad \Lambda_2 = \Sigma, \Lambda''$,
$\Lambda = \Sigma, \Lambda', \Lambda''$, $\quad \Sigma \subseteq \Lambda^1$
– $\Omega_1 = \Omega$
– $\Omega_1 = \Omega, \alpha * \beta$, provided that
$\alpha * \beta$ doesn't appear in Ω

$$* = \begin{cases} \rightarrow, & \text{in } (\rightarrow\ \mathrm{l}) \text{ and } (\rightarrow\ \mathrm{r}) \\ \cdot, & \text{in } (\cdot\ \mathrm{l}) \text{ and } (\cdot\ \mathrm{r}) \\ +, & \text{in } (+\ \mathrm{l}) \text{ and } (+\ \mathrm{r}) \\ \wedge, & \text{in } (\wedge\ \mathrm{l}) \text{ and } (\wedge\ \mathrm{r}) \\ \vee, & \text{in } (\vee\ \mathrm{l}) \text{ and } (\vee\ \mathrm{r}) \end{cases}$$

A derivation in CBCW is called *W-normal* if every branch of that derivation contains no $\Gamma \vdash \Delta$ below $\Gamma' \vdash \Delta'$, such that $\Gamma \vdash \Delta$ is a contraction of $\Gamma' \vdash \Delta'$.

Directly from the above consideration it follows that every W-normal derivation in CBCW is finite. Furthermore, the total number of them for any sequent $\Gamma \vdash \Delta$, is finite. Therefore, CBCW is decidable.

We should mention that in every upward derivation of any contractionless system, every (appearance of a) formula is allowed to be used *at most* once. From this point of view, contractionless logics can be looked upon as *resource sensitive* logics.

3. Tableaux for linear, BCK and relevant logic

For determining provability, we also formulate tableau systems for linear, BCK and relevant logic. The basic configuration of our tableau systems will be a finite tree whose nodes, which are not leaves, have either one or two successor nodes. Each tree will have exactly one root, which we call the *initial node of a tree* and at least one leaf, which we call the *end node* of a tree. The nodes of a tree which have exactly two successor nodes, will be called the *branching nodes*.

A *complete branch* of a tree is the sequence of tree-nodes whose initial node is the initial node of a tree, whose end node is the end node of a tree and whose every node (with the exception of exactly one: the end node of a branch) immediately precedes the tree-node which is the next node in the branch. Furthermore, we say that a sequence of tree-nodes is a *branch*, if it is the initial segment of some complete branch.

The *section* of a tree is the sequence of tree-nodes $N_1, \ldots, N_s$, such that:

1. $N_1, \ldots, N_s$ are consecutive nodes from a branch of a tree,
2. N_1 is either the root of the tree or an immediate successor node of a branching node,
3. N_s is either the end node of a tree or a branching node,
4. $N_1, \ldots, N_{s-1}$ are non-branching nodes.

Let τ be a tree. The *initial section* of τ, begins with the root of τ; its *end section*, ends with a leaf of τ.

Every section of a tree is associated with an o-*sequence*, which consists of (possible several) occurrences of 1 and/or 2, as follows:

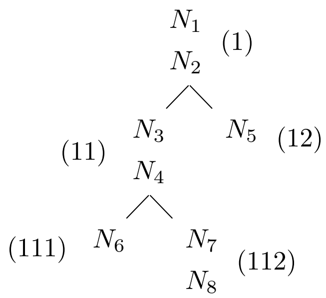
1. 1 is the o-sequence of the initial section of a tree;

2. if $o_1 o_2 \dots o_n$ is an o-sequence of a section, which is not the end section of a tree (the end node of our section is a branching node, then), then its left successor section is associated with $o_1 o_2 \dots o_n 1$ (obtained by concatenating $o_1 o_2 \dots o_n$ and 1) and its right successor section is associated with $o_1 o_2 \dots o_n 2$.

Let τ be a tree and let S be a section in τ. Since there exists only one branch in τ, whose end section is S, every branch in τ is uniquely determined by its end section. Therefore, with every branch of a tree, we associate the o-*sequence* of its end section. Furthermore, with every subtree of a tree, we associate the o-sequence of its initial section. An empty branch and an empty tree are associated with $\emptyset$.

We also define a *part* of a tree as follows. Let t be an o-sequence of a section in τ. Then a part t of τ is the tree which is obtained when the initial segment of a subtree t is substituted for the branch t.

For example, in the following tree (o-sequences of corresponding sections are given within parentheses):



the section 11 consists of the nodes N_3 and N_4, the branch 11 consists of N_1, N_2, N_3 and N_4, the subtree 11 consists of N_3, N_4, N_6, N_7 and N_8 and the part 11 consists of N_1, N_2, N_3, N_4, N_6, N_7 and N_8.

Let φ be a propositional formula. Then $s\,\varphi$, where $s \in \{+,-\}$ is a *signed formula*: $+\,\varphi$ is *positive* and $-\,\varphi$ is *negative.*

Tableaux will be defined as trees to whose nodes *t-formulae* have been assigned. A *t-formula* of a tableau τ is of the form $s\varphi z$, where $s\varphi$ is a signed formula and z is either an o-sequence of a section in τ or $\emptyset$, with the following meaning: the signed formula $s\varphi$ is not allowed to be developed in the part z of τ.

We shall use Σ, $\Sigma_1, \dots, \Pi, \dots, \Phi, \dots$, to denote finite sequences of t-formulae. A sequence Σ, Π is obtained by concatenating Σ and Π.

We say that a sequence of signed formulae $s_1\alpha_1, \dots, s_n\alpha_n$ *corresponds* to a sequence of t-formulae $F_1, \dots, F_n$, and vice versa, iff $F_i = s_i\alpha_i z_i$, for every $i \in \{1, \dots, n\}$. We shall use $S(\Sigma)$ to denote a sequence of signed formulae which corresponds to Σ.

Let S be a sequence of signed formulae. We shall use $\mathrm{Perm}(S)$ to denote a sequence of signed formulae which is obtained by permuting the elements of S.

Furthermore, we shall use $\mathrm{Perm}(\Sigma)$ to denote a sequence of t-formulae which is obtained by permuting the elements of Σ.

Tableau systems consist of tableau rules. Every tableau rule is either a *branch closure rule*, which is of the form $\frac{\Sigma}{-}$ (used) or an *expansion rule* (which is either a rule for a constant or a rule for a connective), which has one of the following forms: $\frac{\Sigma}{-}$ (used), when the rule generates no new nodes, or $\frac{\Sigma}{\Sigma_1}$ (used), when the rule generates some new nodes, enlarging the end section of some complete branch with Σ_1, or $\frac{\Sigma}{\Sigma_1|\Sigma_2}$ (used), when the rule generates some new nodes, branching some complete branch, such that Σ_1 belongs to its left successor section and Σ_2 belongs to its right successor section. The meaning of 'used' will be given later.

Now we define tableaux as follows.

Definition 6. Let TS be a tableau system and let $S = s_1\,\alpha_1, \ldots, s_n\,\alpha_n$ be a sequence of *signed formulae*. Let t be either an o-sequence or $\emptyset$. We define a *tableau of S* in TS as follows.

$$\begin{array}{c} s_1\,\alpha_1\,z_1 \\ \vdots \\ s_n\,\alpha_n\,z_n \end{array}$$

where $z_i \neq \mathrm{in}(t)$ for at least one $i \in \{1, \ldots, n\}$, is a tableau of S in TS. If $z_i = \emptyset$, for every $i \in \{1, \cdots, n\}$, then we call it the *initial tableau* of S.

If τ is a tableau of S in TS, then another tableau of S in TS is generated when a rule of TS is applied *at* the end node N of a complete branch in τ, *to* some node(s) from the same branch (to N or/and to some node(s) above N). We apply rules of TS until either a branch closure rule has been applied to every complete branch of our tableau, or until a tableau, where none of the rules from TS can be applied, has been derived. (It should be emphasized that after the application of a branch closure rule no further construction can take place in that branch.)

Instead of a 'node', above, we may also say a t-formula.

A node *at* which we apply a rule is called the *marked* node. Remember that we always apply a rule at the end node of a complete branch. We note that after the application of a rule which is of the form $\frac{\Sigma}{-}$ (used), no new nodes have been generated, therefore some nodes may be marked more than once.

Let t be a complete branch of a tableau τ. We shall say that t is *finished* in a tableau system TS, if none of the rules from TS can be applied to any node from t.

If a branch closure rule has been applied to some node(s) in a complete branch t, then t is *closed*. If every complete branch of a tableau τ is closed, then τ is closed.

Let t be an o-sequence of a section in a tableau τ. We shall say that a branch t (which is not necessary complete) in τ *closes* if:

1. t is a closed complete branch in τ, or
2. every complete branch of the part t of τ is closed, or

3. there is a derivation from τ to a tableau τ', such that every complete branch of the part t of τ' is closed.

A closed tableau of a sequence of signed formulae S in TS is the *proof* of S in TS. We shall say that a tableau τ *closes* in TS if either τ is closed in TS or there is a derivation in TS from τ to a closed tableau.

The leftmost complete branch of a tableau, which is not closed, will be called the *current* branch of a tableau derivation. In our tableau derivations, we shall always apply the rule at the end node of the current branch of a tableau.

A node (a t-formula) to which we apply a rule, is called the *principal node* (*the principal t-formula*) of that rule. When a rule is applied to a t-formula F from a complete branch t of a tableau τ, we say that the signed formula of F is *used* in the branch t. Since linear and BCK are resource sensitive logics, this signed formula is not allowed to be used again, either in the branch t, or in every subtree t, which could be generated in a further derivation (i.e., it cannot be used in the part t of τ and in the part t of any other tableau generated from τ in any further derivation). Therefore, when we apply a rule to a t-formula $F = s\,\varphi\,z$ from a complete branch t, we set z to t (and possible z_i of some other t-formulae $s_i\,\varphi_i\,z_i$ from the branch t), to denote that the signed formula of F cannot be used again in the part t of the current and any further tableau of that derivation. Settings are given in 'used', (see above) which is either empty or is of one of the following forms (t and g are o-sequences):

– t, which means: set z of the *principal* t-formula(e) to t,

– t^g, which means: set z_i of every t-formulae $s_i\,\varphi_i\,z_i$ from the branch g to t.

We shall say that a t-formula is *unused* in a tableau τ if it is of the form $s\,\varphi\,\emptyset$.

Let p be an o-sequence. We shall use in(p) to denote the initial subsequence of p (note that $\emptyset$ is not the initial subsequence of any o-sequence).

A t-formula $F = s\,\varphi\,z$, which belongs to the section g of a tableau τ, cannot be used anywhere in τ, if either $z = g$ or $z = g2\ldots2$. This is clear for $z = g$, since then F belongs to the part g of τ, where it cannot be used. On the other hand, z may be set to $g2\ldots2$ only when a branch $g2\ldots2$ is the current branch of our tableau derivation. Therefore, before the application of the rule, which sets z to $g2\ldots2$, F belongs only to the current branch $g2\ldots2$, of our tableau. After the application of that rule, it cannot be used there (and in the part $g2\ldots2$ of any tableau generated in a further derivation) any more. Therefore, $s\,\varphi\,z$, where $z = g$ or $z = g2\ldots2$, will be called *invisible* in τ (although invisible nodes can be deleted, we keep them to avoid empty sections; however in the program implementation, we delete those nodes and change o-sequences of some sections, if necessary). A t-formula which is not invisible in τ, is called a *visible* t-formula.

We shall use Σ^{vis} (Σ^{invis}) to denote the sequence of visible (invisible) t-formulae from Σ, and Σ_t to denote all t-formulae from a branch t of a tableau.

We now give the postulates of the tableau system for BCK logic, namely TCBCK:

A complete branch t closure rules, $z_1 \neq \mathrm{in}(t)$, $z_2 \neq \mathrm{in}(t)$, $z \neq \mathrm{in}(t)$:

$$(-1):\ \frac{\begin{array}{c}\Sigma\\ -\,1\,z\\ \Pi\end{array}}{-}\ (t^t) \qquad\qquad (+0):\ \frac{\begin{array}{c}\Sigma\\ +\,0\,z\\ \Pi\end{array}}{-}\ (t^t)$$

$$(-\top):\ \frac{\begin{array}{c}\Sigma\\ -\,\top\,z\\ \Pi\end{array}}{-}\ (t^t) \qquad\qquad (+\bot):\ \frac{\begin{array}{c}\Sigma\\ +\,\bot\,z\\ \Pi\end{array}}{-}\ (t^t)$$

$$(+\alpha,-\alpha):\ \frac{\begin{array}{c}\Sigma\\ s\,\alpha\,z_1\\ \Phi\\ \overline{s}\,\alpha\,z_2\\ \Pi\end{array}}{-}\ (t^t) \qquad\qquad \overline{s} = \begin{cases}+, & \text{if } s = -\\ -, & \text{if } s = +\end{cases}$$

Rules for constants, t is the current branch, $z \neq \mathrm{in}(t)$:

$$(+1):\ \frac{\begin{array}{c}\Sigma\\ +\,1\,z\\ \Pi\end{array}}{-}\ (t) \qquad\qquad (-0):\ \frac{\begin{array}{c}\Sigma\\ -\,0\,z\\ \Pi\end{array}}{-}\ (t)$$

Rules for connectives (t is the current branch, $z \neq \mathrm{in}(t)$; $t1$ ($t2$) is the o-sequence obtained by concatenating t and 1 (2); Φ_1 and Φ_2 are sequences of *unused* t-formulae, which are formulated as follows: for every visible t-formula $(s\,\varphi\,z)$ from the branch t, we formulate another *unused* t-formula $(s\,\varphi\,\emptyset)$ which belongs either to Φ_1 (i.e., to the branch $t1$), or to Φ_2 (i.e., to the branch $t2$)):

$$(+\rightarrow):\ \frac{\begin{array}{c}\Sigma\\ +\,\alpha\rightarrow\beta\,z\\ \Pi\end{array}}{\begin{array}{c|c}\Phi_1 & \Phi_2\\ -\,\alpha\,\emptyset & +\,\beta\,\emptyset\end{array}}\ (t^t) \qquad\qquad (-\rightarrow):\ \frac{\begin{array}{c}\Sigma\\ -\,\alpha\rightarrow\beta\,z\\ \Pi\end{array}}{\begin{array}{c}+\,\alpha\,\emptyset\\ -\,\beta\,\emptyset\end{array}}\ (t)$$

$$(+\cdot):\ \frac{\begin{array}{c}\Sigma\\ +\,\alpha\cdot\beta\,z\\ \Pi\end{array}}{\begin{array}{c}+\,\alpha\,\emptyset\\ +\,\beta\,\emptyset\end{array}}\ (t) \qquad\qquad (-\cdot):\ \frac{\begin{array}{c}\Sigma\\ -\,\alpha\cdot\beta\,z\\ \Pi\end{array}}{\begin{array}{c|c}\Phi_1 & \Phi_2\\ -\,\alpha\,\emptyset & -\,\beta\,\emptyset\end{array}}\ (t^t)$$

$$(++):\ \frac{\begin{array}{c}\Sigma\\ +\,\alpha+\beta\,z\\ \Pi\end{array}}{\begin{array}{c|c}\Phi_1 & \Phi_2\\ +\,\alpha\,\emptyset & +\,\beta\,\emptyset\end{array}}\ (t^t) \qquad\qquad (-+):\ \frac{\begin{array}{c}\Sigma\\ -\,\alpha+\beta\,z\\ \Pi\end{array}}{\begin{array}{c}-\,\alpha\,\emptyset\\ -\,\beta\,\emptyset\end{array}}\ (t)$$

$$(+\wedge):\ \frac{\begin{array}{c}\Sigma\\ +\,\alpha\wedge\beta\ z\\ \Pi\end{array}}{+\,\alpha\,\emptyset}\ (t),\qquad \frac{\begin{array}{c}\Sigma\\ +\,\alpha\wedge\beta\ z\\ \Pi\end{array}}{+\ \beta\,\emptyset}\ (t)\qquad\qquad (-\wedge):\ \frac{\begin{array}{c}\Sigma\\ -\,\alpha\wedge\beta\ z\\ \Pi\end{array}}{-\ \alpha\,\emptyset\ \big|-\ \beta\,\emptyset}\ (t)$$

$$(+\vee):\ \frac{\begin{array}{c}\Sigma\\ +\,\alpha\vee\beta\ z\\ \Pi\end{array}}{+\ \alpha\,\emptyset\ \big|+\ \beta\,\emptyset}\ (t)\qquad\qquad (-\vee):\ \frac{\begin{array}{c}\Sigma\\ -\,\alpha\vee\beta\ z\\ \Pi\end{array}}{-\,\alpha\,\emptyset}\ (t),\qquad \frac{\begin{array}{c}\Sigma\\ -\,\alpha\vee\beta\ z\\ \Pi\end{array}}{-\ \beta\,\emptyset}\ (t)$$

We note that after the application of a rule for a constant, no new nodes of a tableau have been generated, therefore a marked node of our tableau, at which we apply a rule for a constant is also a node at which we apply the next rule of our tableau derivation, i.e., this node is marked more than once.

We also note that the branch closure rules for 1 and 0 coincide with the rules for $\top$ and $\perp$, respectively. It is expected, since 1 and $\top$ and 0 and $\perp$ are equivalent in BCK logic. Therefore, we can exclude both $\top$ and $\perp$, together with the rules for $\top$ and $\perp$ from TCBCK. However in their presence, a tableau system for linear logic (where 1 differs from $\top$ and 0 from $\perp$), TCBC, can be defined: TCBC, has the postulates of TCBCK above, satisfying $((\Sigma,\Phi,\Pi)_t)^{\mathrm{vis}}$ in $(+\alpha,-\alpha)$ and $((\Sigma,\Pi)_t)^{\mathrm{vis}}$ in (-1) and $(+0)$ are empty.

Furthermore, we define a tableau system for relevant logic, TCBCW, from TCBC, as follows. The axioms and the rules for constants in TCBCW are the same as in TCBC. We change the rules for connectives in TCBC, to allow applications of contraction as discussed in the previous section. We shall use Σ^1 to denote the sequence of all t-formulae from Σ, whose corresponding signed formulae appear exactly once in $S(\Sigma)$.

We note that in the rules $(+\rightarrow)$, $(-\,\cdot)$ and $(+\,+)$, in TCBC (and TCBCK), Φ_1 and Φ_2 are such that:

1. $S((\Sigma,\Pi)^{\mathrm{vis}}) = \mathrm{Perm}(S(\Phi_1,\Phi_2))$.

In TCBCW, we also allow Φ_1 and Φ_2 to be as follows (we use F to denote the principal t-formula of the rule):

2. $S(\Phi_1) = S(\Omega,\Delta_1)$, $S(\Phi_2) = S(\Omega,\Delta_2)$, $\Omega \subseteq ((\Sigma,\Pi)^{\mathrm{vis}})^1$ and $\mathrm{Perm}(\Omega,\Delta_1,\Delta_2) = (\Sigma,\Pi)^{\mathrm{vis}}$.

3. $S(\Phi_1,\Phi_2) = \mathrm{Perm}(S(F,(\Sigma,\Pi)^{\mathrm{vis}}))$, provided that $S(F)$ does not appear in $S((\Sigma,\Pi)^{\mathrm{vis}})$.

4. $S(\Phi_1) = S(\Omega,\Delta_1)$, $S(\Phi_2) = S(\Omega,\Delta_2)$, $\Omega \subseteq ((\Sigma,\Pi)^{\mathrm{vis}})^1$ and $\mathrm{Perm}(\Omega,\Delta_1,\Delta_2) = \mathrm{Perm}(F,(\Sigma,\Pi)^{\mathrm{vis}})$, provided that $S(F)$ does not appear in $S((\Sigma,\Pi)^{\mathrm{vis}})$.

5. $S(\Phi_1) = S(F,\Omega,\Delta_1)$, $S(\Phi_2) = S(F,\Omega,\Delta_2)$, where $\Omega \subseteq ((\Sigma,\Pi)^{\mathrm{vis}})^1$ and $\mathrm{Perm}(\Omega,\Delta_1,\Delta_2) = (\Sigma,\Pi)^{\mathrm{vis}}$, provided that $S(F)$ does not appear in $S((\Sigma,\Pi)^{\mathrm{vis}})$.

The 'used' in the rules $(+\rightarrow)$, $(-\,\cdot)$ and $(+\,+)$ of TCBCW is the same as in TCBC.

In all other rules, we alow 'used' to be either (t) (as in TCBC) or empty, provided that $S(F)$ does not appear in $S((\Sigma,\Pi)^{\mathrm{vis}})$.

Our tableaux are very closely related to upward sequent derivations. In the proof of the following theorem, we shall see how the upward derivation of a sequent

$\gamma_1, \ldots, \gamma_n \vdash \delta_1, \ldots, \delta_m$ in CBCK can be transformed into the tableau derivation of the sequence $+\gamma_1, \ldots, +\gamma_n, -\delta_1, \ldots, -\delta_m$ in TCBCK, and vice versa.

We shall say that in a tableau τ, a node N_1 is the *predecessor* node of a node N_2 iff there is a branch in τ which contains both N_1 and N_2, such that N_1 is above N_2 (we also say that N_2 is the successor node to N_1, then).

Let τ be a tableau of S in TCBCK and let N be a marked node in τ, which is marked $k \geqslant 1$ times. Let t be a branch of τ (which is not necessarily complete) whose end node is N. Let $\Sigma = F_1, \ldots, F_m$ be the sequence of all t-formulae from t (F_m is assigned to N) and let $S(\Sigma) = f_1, \ldots, f_m$ (remember that $S(\Sigma)$ is the sequence of signed formulae which corresponds to Σ). Then, for every $j \in \{1, \ldots, k\}$, we define the sequence of signed formulae $\mathrm{Unused}_j(N)$, as follows: $\mathrm{Unused}_j(N)$ is the sequence of signed formulae from $f_1, \ldots, f_m$ which are unused in t until the moment we mark N for the j-th time.

For example:

$N_1:$	$-\ (\alpha \cdot \beta. \cdot \gamma) \cdot 1 \to \alpha \quad 1$	$\mathrm{Unused}_1(N_1) = -\,(\alpha \cdot \beta. \cdot \gamma) \cdot 1 \to \alpha$
$N_2:$	$+\ (\alpha \cdot \beta. \cdot \gamma) \cdot 1 \quad 1$	
$N_3:$	$-\ \alpha \quad 1$	$\mathrm{Unused}_1(N_3) = +\,(\alpha \cdot \beta. \cdot \gamma) \cdot 1, -\,\alpha$
$N_4:$	$+\ \alpha \cdot \beta. \cdot \gamma \quad 1$	
$N_5:$	$+\ 1 \quad 1$	$\mathrm{Unused}_1(N_5) = -\,\alpha, +\,\alpha \cdot \beta. \cdot \gamma, +\,1$
$N_6:$	$+\ \alpha \cdot \beta \quad 1$	
$N_7:$	$+\ \gamma \quad 1$	$\mathrm{Unused}_1(N_7) = -\,\alpha, +\,1, +\,\alpha \cdot \beta, +\,\gamma$
		$\mathrm{Unused}_2(N_7) = -\,\alpha, +\,\alpha \cdot \beta, +\,\gamma$
$N_8:$	$+\ \alpha \quad 1$	
$N_9:$	$+\ \beta \quad 1$	$\mathrm{Unused}_1(N_9) = -\,\alpha, +\,\gamma, +\,\alpha, +\,\beta$

At the end of our derivation, a branch closure rule has been applied to $-\,\alpha\,\emptyset$ and $+\,\alpha\,\emptyset$ at $+\,\beta\,\emptyset$.

Let S be a sequence of signed formulae. If $+\alpha_1, \ldots, +\alpha_m$ are all positive and if $-\beta_1, \ldots, -\beta_n$ are all negative signed formulae from S, then $\mathrm{Sequent}(S)$ (a *sequent of* S) is $\alpha_1, \ldots, \alpha_m \vdash \beta_1, \ldots, \beta_n$, where $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_n$ are multisets of formulae (each of them may be empty).

We now show that CBCK and TCBCK are equivalent systems. Namely, we shall prove the following theorem:

THEOREM 1. *The sequent* $\gamma_1, \ldots, \gamma_n \vdash \delta_1, \ldots, \delta_m$ *is provable in* CBCK *iff there is a proof of* $+\gamma_1, \ldots, +\gamma_n, -\delta_1, \ldots, -\delta_m$ *in* TCBCK.

For the proof we shall use the following lemma:

LEMMA 7. *Let* τ *be a closed tableau of* S *in* TCBCK *and let* N *be a marked node in* τ*. If* N *is marked* $m \geqslant 1$ *times in* τ*, then for every* $j \in \{1, \ldots, m\}$*, a sequent* $\mathrm{Sequent}(\mathrm{Unused}_j(N))$ *is provable in* CBCK.

PROOF. Let N be a marked node in a closed tableau τ of S in TCBCK. We proceed by induction on n, where n is the number of marked nodes which succeed N in τ. First we note that if N is marked $m > 1$ times, then rules for constants have been applied $m-1$ times to N, before a rule for a connective or a branch closure rule has been applied to N. Since every Sequent(Unused$_i(N)$) can be derived from Sequent(Unused$_{i+1}(N)$) by the application of either (1 l) or (0 r), for every $i \in \{1, \dots, m-1\}$, it is enough to prove that Sequent(Unused$_m(N)$) is provable in CBCK.

Let $n = 0$ (note that N is the end node of τ, then). If N is marked $m \geqslant 1$ times, then the m-th applied rule at N is a branch closure rule. We have the following cases. If the m-th applied rule at N is $(+\alpha, -\alpha)$, then Sequent(Unused$_m(N)$) is of the form $\Gamma, \alpha \vdash \alpha, \Delta$, i.e., it is the axiom of CBCK, therefore it is provable. We proceed analogously when any other branch closure rule has been applied at N.

Let $n > 0$ and let N be marked $m \geqslant 1$ times. Then we have the following cases. Let the m-th applied rule at N be $(+\ \rightarrow)$ and let it be applied to a node M to which $+\,\alpha \rightarrow \beta\, z$ is assigned. M is either N or the predecessor node of N. Then we wish to show that Sequent(Unused$_m(N)$) is provable in CBCK.

A node N_1, to which a t-formula $-\,\alpha\,\emptyset$ is assigned, immediately after the application of the rule $(+\ \rightarrow)$ at N to M, is our next marked node in the branch $t1$ (see the formulation of rule $(+\ \rightarrow)$ in TCBCK). Let the m_1-th, $m_1 \geqslant 1$, applied rule at N_1 be a rule which is not not a rule for a constant. Since N_1 is followed by less than n succeeding marked nodes, by the induction hypothesis Sequent(Unused$_{m_1}(N_1)$) is provable in CBCK. We proceed analogously to show that Sequent(Unused$_{m_2}(N_2)$) is provable in CBCK, where N_2 is a node, to which a t-formula $+\,\beta\,\emptyset$ is assigned immediately after the application of the rule $(+\ \rightarrow)$ at N to M and where $m_2 \geqslant 1$ is such that the m_2-th applied rule at N_2 is not a rule for a constant.

If Sequent(Unused$_{m_1}(N_1)$) $= \Gamma_1 \vdash \alpha, \Delta_1$ and if Sequent(Unused$_{m_2}(N_2)$) $= \Gamma_2, \beta \vdash \Delta_2$, then Sequent(Unused$_m(N)$) $= \Gamma, \alpha \rightarrow \beta \vdash \Delta$, where $\Gamma = \Gamma_1, \Gamma_2, \Sigma$, such that Σ is either an empty multiset or a multiset which consists of repeated occurrences of 1 only, and $\Delta = \Delta_1, \Delta_2, \Pi$, such that Π is either an empty multiset or a multiset which consists of repeated occurrences of 0 only. Now it is clear that our sequent Sequent(Unused$_m(N)$) is provable in CBCK:

$$
\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \Gamma_1 \vdash \alpha, \Delta_1\\ \hline \cdots\end{array}}{\Gamma_1, \Sigma_1 \vdash \alpha, \Delta_1, \Pi_1}\ \begin{array}{l}\text{possible applications}\\ \text{of the rules (1 l) and}\\ \text{(0 r)}\end{array} \qquad \dfrac{\begin{array}{c}\pi_2\\ \Gamma_2, \beta \vdash \Delta_2\\ \hline \cdots\end{array}}{\Gamma_2, \beta, \Sigma_2 \vdash \Delta_2, \Pi_2}\ \begin{array}{l}\text{possible applications}\\ \text{of the rules (1 l) and}\\ \text{(0 r)}\end{array}}{\Gamma_1, \Gamma_2, \Sigma, \alpha \rightarrow \beta \vdash \Delta_1, \Delta_2, \Pi}\ (\rightarrow\ \mathrm{l})
$$

where $\Sigma_1, \Sigma_2 = \Sigma$ and $\Pi_1, \Pi_2 = \Pi$.

We proceed analogously in all other cases. □

Proof of Theorem 1. Let D be the proof of $\gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m$ in CBCK ($\gamma_1, \dots, \gamma_n$ and $\delta_1, \dots, \delta_m$ are multisets of formulae). We formulate the initial tableau of $S = +\gamma_1, \dots, +\gamma_n, -\delta_1, \dots, -\delta_m$ in TCBCK:

$$\begin{array}{c} +\,\gamma_1\,\emptyset \\ \vdots \\ +\,\gamma_n\,\emptyset \\ -\,\delta_1\,\emptyset \\ \vdots \\ -\,\delta_m\,\emptyset \end{array}$$

By induction on n, where n is the length of D (n is the total number of sequents in D), we wish to show that this closes in TCBCK.

Let $n = 1$. Then our sequent is an axiom. We proceed as follows. If our sequent is an axiom $\Gamma, \alpha \vdash \alpha, \Delta$ (Id), then a tableau which begins with $+\alpha_1 z_1, \ldots,$ $+\alpha_{n-1} z_{n-1}, +\alpha z_n, -\alpha z_1', -\beta_2 z_2', \ldots, -\beta_m z_m'$, where $z_i \neq \mathrm{in}(t)$ for every $i = \{1, ..., n\}$ and $z_j' \neq \mathrm{in}(t)$ for every $j = \{1, \ldots, m\}$, for either $t = \emptyset$ or any o-sequence t, is closed by the application of the rule $(+\alpha, -\alpha)$ to $+\alpha z_n$ and $-\alpha z_1'$. We proceed analogously when our sequent is one of the axioms (1 r) or (0 l) ((⊤ r) or (⊥ l)).

Suppose $n > 1$. Then our sequent is the lower sequent of an inference rule in CBCK. We have the following cases.

Let D be of the following form:

$$\dfrac{\begin{array}{c}\pi_1 \\ \gamma_1, \ldots, \gamma_{i-1} \vdash \alpha, \delta_1, \ldots, \delta_j\end{array} \qquad \begin{array}{c}\pi_2 \\ \gamma_i, \ldots, \gamma_n, \beta \vdash \delta_{j+1}, \ldots, \delta_m\end{array}}{\gamma_1, \ldots, \gamma_n, \alpha \rightarrow \beta \vdash \delta_1, \ldots, \delta_m}\ (\rightarrow\ \mathrm{l})$$

Then we consider the tableau that begins with:

$$\begin{array}{c} \Sigma \\ +\,\alpha \rightarrow \beta\ z \\ \Pi \end{array}$$

such that $S((\Sigma, \Pi)_t^{\mathrm{vis}}) = \mathrm{Perm}(+\gamma_1, \ldots, +\gamma_n, -\delta_1, \ldots, -\delta_m)$ and $z \neq \mathrm{in}(t)$, for either $t = \emptyset$ or any o-sequence t. We wish to show that this tableau closes. We apply the rule $(+\rightarrow)$ to $+\,\alpha \rightarrow \beta\ z$:

$$\dfrac{\begin{array}{c} \Sigma \\ +\,\alpha \rightarrow \beta\ z \\ \Pi \end{array}}{\begin{array}{c|c} +\,\gamma_1\,\emptyset & +\,\gamma_i\,\emptyset \\ \vdots & \vdots \\ +\,\gamma_{i-1}\,\emptyset & +\,\gamma_n\,\emptyset \\ -\,\delta_1\,\emptyset & -\,\delta_{j+1}\,\emptyset \\ \vdots & \vdots \\ -\,\delta_j\,\emptyset & -\,\delta_m\,\emptyset \\ -\,\alpha\,\emptyset & +\,\beta\,\emptyset \end{array}}\ (t^t)$$

On inductive hypothesis, our tableau closes.

We proceed analogously when D ends with any other inference rule of CBCK.

Conversely, we wish to show that if there is a closed tableau τ of $S = +\gamma_1, \ldots,$ $\ldots, +\gamma_n, -\delta_1, \ldots, -\delta_m$, in TCBCK, then there is a proof of $\gamma_1, \ldots, \gamma_n \vdash \delta_1, \ldots, \delta_m$

in CBCK. We should note that the first marked node in our tableau is the node N_1 to which the t-formula $-\delta_m z$ is assigned ($z = \emptyset$ in the initial tableau of S). However, directly from Lemma 7 it follows that a sequent Sequent(Unused$_1(N_1)$) $= \gamma_1, \dots, \gamma_n \vdash \delta_1, \dots, \delta_m$ is provable in CBCK. □

A tableau system for classical logic, TCBCKW, consists of the axioms of TCBCK, together with the rules for constants and the rules for connectives of TCBCW. Tableau systems for single-conclusion logics are without the connective $+$ and they satisfy the following condition: at any moment of a derivation, every complete branch of a tableau contains at most one negative visible t-formula.

4. The general algorithm

We shall describe the algorithm which can be used for determining provability in any of our logics. We shall use it for designing two theorem provers: the first one, $\mathcal{P}_1$, searches for the proof in single- and multiple-conclusion Lambek logic and in single- and multiple conclusion Lambek logic with weakening; the other one, $\mathcal{P}_2$ searches for the proof in single- and multiple-conclusion linear, BCK and relevant logic, intuitionistic and classical logic. Here we shall describe $\mathcal{P}_2$, only.

We should mention that our prover is not the most efficient one: for every logic, it is possible to formulate more efficient proof search. However, the advantage of our approach is that, for a given input, we always get, as an output, the list of structural rules, whose presence is necessary in the proof, if the given input is a theorem of at least one of our logics (otherwise, the program exits either with the message about syntactic incorrectness of our input, or with the message that our input is a formula which is not a theorem in any of our logics).

The specific feature of our subtructural logics is the nondeterminism of their derivations. Namely, for tableaux τ and τ', in a tableau system TS, where τ' is obtained from τ by an application of a rule from TS, it is possible that τ closes and τ' doesn't (except when TS is a tableau system for either intuitionistic or classical logic). This means that if the application of a rule to a t-formula in the current branch leads to a closed tableau, then the application of a rule to some other t-formula in the current branch need not. Furthermore, even when a tableau of S closes in the presence of some structural rules or when it closes in a multiple-conclusion logic, the derivation is not finished: we still wish to know if there is a closed tableau of S, in the presence of some other structural rules or without any of them or if there is a closed tableau of S in some single-conclusion logic. Therefore, the use of backtracking is necessary, in designing the prover.

The structure that supports backtracking is `MOGUCI`. A node of the type `MOGUCI` contains information about the tableau at the moment when the derivation becomes nondeterministic (i.e., it contains the tableau at the moment we chose one possibility to continue the derivation, when at least two of them are available; furthermore, all structural rules that have been used in the derivation until then, together with the information whether the derivation has been in a single- or in a multiple-conclusion logic, form the *closure condition* which we save in the *closure fields*;

algorithm uses several closure fields; closure fields of a node of the type `MOGUCI` are some of them).

Our program ($\mathcal{P}_2$) always searches for a proof with the *minimal closure condition.* Our minimal logic, where a given formula can be a theorem is the single-conclusion linear logic. The *minimal closure condition* is defined as follows.

DEFINITION 7. Let C_1 and C_2 be two closure conditions. If C_1 gives the proof in a single-conclusion logic and if C_2 gives the proof in a multiple-conclusion logic, then C_1 is less than C_2. Furthermore, if both C_1 and C_2 give the proof either in a single- or in a multiple-conclusion logic, then C_1 is less than C_2 if either C_1 is with permutation only, or C_2 is with permutation, weakening and contraction.

Our prover uses the system TCBCKW. However, if there is a proof of $-\varphi$ in TCBCKW, where every complete branch is closed using the axioms of TCBCW (TCBC) only, and where at least one rule for a connective, which is not also the rule for a connective in TCBC, is used, then $-\varphi$ is the theorem in a multiple-conclusion logic, in the presence of permutation and contraction, only, i.e., in TCBCW. On the other hand, if there is a proof of $-\varphi$ in TCBCKW, where at least one branch is closed using an axiom of TCBCK, which is not the axiom of TCBC, also, and where every applied rule for a connective is from TCBC only, then $-\varphi$ is the theorem in a multiple-conclusion logic, in the presence of permutation and weakening only, i.e., in TCBCK. Furthermore, if there is a proof of $-\varphi$ in TCBCKW, where every complete branch is closed using the axioms of TCBCW (TCBC) only, and where every applied rule for a connective is from TCBC only, then $-\varphi$ is the theorem in a multiple-conclusion logic, in the presence of permutation only, i.e., in TCBC. Finally, if φ is without $+$ and if every complete branch generated in the proof has at most one visible negative t-formula, then our formula is provable in a single-conclusion logic with permutation and/or weakening and/or contraction, due to the above consideration.

Now we give the general algorithm. Our input gets the minus sign as prefix and if it is syntactically correct, the associated binary tree of a given signed formula $-\varphi$ is produced. From the root of this tree, the initial tableau is generated. A proof search procedure proceeds as follows.

1. If the current branch is closed, set the appropriate values in the closure fields. If the current branch is closed and finished, then go to 7, else go to 2.

2. If the current branch is finished, but not closed, or empty, then go to 4, else go to 3.

3. Pick up a t-formula to which we wish to apply a rule and form a new node of the list `moguci`, containing the information about the rest of t-formulae that could be picked up instead. If the rule can be applied to a picked up t-formula on several equally applicable possibilities, we chose one, and form another node of the list `moguci`, containing the information about the others. We apply the rule, then go to 1.

4. If the list `moguci` is empty, then exit with the message “the formula φ is not a theorem in any of corresponding logics”, else go to 5.

5. If all possibilities for deriving the tableau in the first node of the list `moguci`, are exhausted, then go to 6, else generate another new current state based on the information from that node and go to 1.

6. Delete the first node of the list `moguci`. If the tableau of the deleted node of the list `moguci` didn't close in any of earlier derivations, then go to 4, else, set the appropriate values in the closure fields and go to 7.

7. If the list `moguci` is empty, then go to 8, else go to 9.

8. If every complete branch of our tableau is closed, then exit with the message "the formula φ is a theorem", and give all structural rules needed in the proof, else generate another current state and go to 1.

9. If the tableau from the first node of the list `moguci`, closes, then go to 10, else take its leftmost open branch for the current branch and go to 1.

10. If all possibilities for deriving the tableau from the first node of the list `moguci` are exhausted, then delete that node from the list, set the appropriate values in the closure fields and go to 7, else form another current state based on the information from that node and go to 1.

We shall illustrate our proof search procedure with the following simple example. Namely, we shall prove the theorem: $(\alpha \to \beta) \to \alpha. \to \alpha$.

At the beginning the initial tableau of $-\,(\alpha \to \beta) \to \alpha. \to \alpha$ is produced:

$$\tau_1: \qquad -\,(\alpha \to \beta) \to \alpha. \to \alpha \quad \emptyset$$

Here, only one rule can be applied, the rule $(-\,\to)$ to $-\,(\alpha \to \beta) \to \alpha. \to \alpha \quad \emptyset$:

$$\tau_2: \qquad \begin{array}{l} -\,(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +\,(\alpha \to \beta) \to \alpha \quad \emptyset \\ -\,\alpha \quad \emptyset \end{array}$$

Here, we can apply only one rule again, the rule $(+\,\to)$ to $+\,(\alpha \to \beta) \to \alpha \quad \emptyset$. However it can be applied in several different ways. We chose one of them to derive a tableau τ_3 and we generate the first node of the list `moguci` to remember the other possibilities.

$$\tau_3: \qquad \begin{array}{l} -\,(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +\,(\alpha \to \beta) \to \alpha \quad 1 \\ -\,\alpha \quad 1 \\ \begin{array}{c|c} -\,\alpha \to \beta \quad \emptyset & \begin{array}{l} +\,\alpha \quad \emptyset \\ -\,\alpha \quad \emptyset \end{array} \end{array} \end{array}$$

After the application of the rule $(-\,\to)$ to $-\,\alpha \to \beta \quad \emptyset$, the tableau τ_4 is produced, whose every branch is finished, but the leftmost one is not closed, therefore it is not the proof of our formula:

$$\tau_4: \qquad \begin{array}{l} -\,(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +\,(\alpha \to \beta) \to \alpha \quad 1 \\ -\,\alpha \quad 1 \\ \begin{array}{l|l} \begin{array}{l} -\,\alpha \to \beta \quad 11 \\ +\,\alpha \quad \emptyset \\ -\,\beta \quad \emptyset \end{array} & \begin{array}{l} +\,\alpha \quad \emptyset \\ -\,\alpha \quad \emptyset \end{array} \end{array} \end{array}$$

Since the list `moguci` is not empty, we generate another tableau τ_3, keeping the remaining possibilities in the list `moguci`:

$$\tau_3: \quad \begin{array}{l} -(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +(\alpha \to \beta) \to \alpha \quad 1 \\ -\alpha \quad 1 \\ \begin{array}{l|l} \begin{array}{l} -\alpha \quad \emptyset \\ -\alpha \to \beta \quad \emptyset \end{array} & +\alpha \quad \emptyset \end{array} \end{array}$$

Then we apply the rule $(- \to)$ to generate τ_4:

$$\tau_4: \quad \begin{array}{l} -(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +(\alpha \to \beta) \to \alpha \quad 1 \\ -\alpha \quad 1 \\ \begin{array}{l|l} \begin{array}{l} -\alpha \quad \emptyset \\ -\alpha \to \beta \quad 11 \\ +\alpha \quad \emptyset \\ -\beta \quad \emptyset \end{array} & +\alpha \quad \emptyset \end{array} \end{array}$$

The current branch (the leftmost one) is closed in the multiple-conclusion logic with weakening. However the other one is finished, but it is not closed. Therefore our tableau, again, is not a proof of our formula. Once more, we pick the (last) information from the list `moguci` to generate another tableau τ_3:

$$\tau_3: \quad \begin{array}{l} -(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +(\alpha \to \beta) \to \alpha \quad 1 \\ -\alpha \quad 1 \\ \begin{array}{l|l} \begin{array}{l} -\alpha \quad \emptyset \\ -\alpha \to \beta \quad \emptyset \end{array} & \begin{array}{l} -\alpha \quad \emptyset \\ +\alpha \quad \emptyset \end{array} \end{array} \end{array}$$

We note that the above tableau is derived in the presence of contraction and we keep that information in the corresponding closure fields. Then again we apply the rule $(- \to)$ to $-\alpha \to \beta \quad \emptyset$ to generate the final tableau of our derivation:

$$\tau_4: \quad \begin{array}{l} -(\alpha \to \beta) \to \alpha. \to \alpha \quad 1 \\ +(\alpha \to \beta) \to \alpha \quad 1 \\ -\alpha \quad 1 \\ \begin{array}{l|l} \begin{array}{l} -\alpha \quad \emptyset \\ -\alpha \to \beta \quad 11 \\ +\alpha \quad \emptyset \\ -\beta \quad \emptyset \end{array} & \begin{array}{l} -\alpha \quad \emptyset \\ +\alpha \quad \emptyset \end{array} \end{array} \end{array}$$

The leftmost branch is closed in the multiple-conclusion logic with weakening; the other one is closed unconditionally. Since the list `moguci` is empty, our derivation is finished: our formula is a theorem in a multiple-conclusion logic with permutation, contraction and weakening (i.e., in classical logic, only).

Acknowledgements. I would like to express my gratitude to Professors Zoran Petrić and Miodrag Kapetanović for many useful suggestions. Furthermore, I am grateful to the referees for very helpful comments.

References

[1] V. M. Abrusci, *Phase semantics and sequent calculus for pure noncomutative classical linear propositional logic*, J. Symbolic Logic 56:4 (1991), 1403–1451.
[2] G. Gentzen, *Investigations into logical deduction*; in: M. E. Szabo (Ed.), *The Collected Papers of Gerhard Gentzen*, North-Holland, Amsterdam, 1969, pp. 68–131
[3] M. Isaković Ilić, *Cut elimination and decidability for classical Lambek logic*, Journal of Logic and Computation 2007; DOI:10.1093/logcom/exm063
[4] P. Schroeder-Heister and K. Došen (Eds.), *Substructural Logics*, Oxford, Oxford University Press, 1993.
[5] S. Kripke, *The problem of entailment*, abstract, J. Symbolic Logic 24 (1959), 234.
[6] G. Restall, *An intruduction to Substructural Logics*, Routledge, London, 2000.

Šumarski fakultet
Kneza Višeslava 1
11000 Beograd
Serbia
isakm@yubc.net