A Decidable Tableau Calculus for MLSS

Domenico Cantone

Dipartimento di Matematica e Informatica
Università di Catania

Logica Computazionale, A.A. 2006/07

Outline

1. **A Decidable Tableau Calculus for MLSS**
   - Syntax and semantics of MLSS
   - A tableau calculus for MLSS
   - Soundness of the tableaux calculus $\mathfrak{T}_{MLSS}$
   - Completeness of $\mathfrak{T}_{MLSS}$
     - A saturation process for $\mathfrak{T}_{MLSS}$-tableaux
     - Realizations of graphs by set-labelings
     - Satisfiability of open and s-restricted $\mathfrak{T}_{MLSS}$-tableaux
     - A decision procedure for MLSS

Outline

1. **A Decidable Tableau Calculus for MLSS**
   - Syntax and semantics of MLSS
   - A tableau calculus for MLSS
   - Soundness of the tableaux calculus $\mathfrak{T}_{MLSS}$
   - Completeness of $\mathfrak{T}_{MLSS}$
     - A saturation process for $\mathfrak{T}_{MLSS}$-tableaux
     - Realizations of graphs by set-labelings
     - Satisfiability of open and s-restricted $\mathfrak{T}_{MLSS}$-tableaux
     - A decision procedure for MLSS

We present a fast tableau-based decision procedure for the ground set-theoretic fragment *Multi-Level Syllogistic with Singleton* (in short **MLSS**), a quantifier-free language with the basic *Boolean set operators* and the *singleton operator*.

The language of MLSS

The language of the fragment **MLSS** of set theory consists of

- a denumerable infinity of uninterpreted set constants $c_0, c_1, \ldots$;
- the interpreted set constant $\emptyset$ (empty set),
- the operator symbols $\cup$ (union), $\cap$ (intersection), $\setminus$ (set difference) and $\{\_\}$-rule (singleton),
- the predicate symbols $\in$ (membership) and $=$ (equality), and
- propositional connectives.

Terms of MLSS

Set terms of **MLSS** are defined in the standard recursive way, namely

- any set constant is an **MLSS**-term;
- if t_1 and t_2 are **MLSS**-terms, so are $t_1 \cup t_2$, $t_1 \cap t_2$, $t_1 \setminus t_2$, and $\{t_1\}$.

Sentences of MLSS

Finally, **MLSS**-sentences are just propositional combinations of atoms of the form $t_1 \in t_2$ and $t_1 = t_2$, with t_1 and t_2 being any set terms.

The intended semantics of **MLSS** over the standard von Neumann universe $\mathbb{V}$ is defined as follows.

- A *(standard) assignment* over a collection V of variables is any function M from V into $\mathbb{V}$.

Clearly, any set-theoretic formula φ all of whose variables belong to V becomes either *true* or *false* when each free occurrence x gets replaced by Mx within φ and set-theoretic operators and relators are interpreted according to their standard meaning.

- An assignment M which makes φ true is said to be a *model* of φ.
- A formula φ is said to be *satisfiable* if it has a model.

Remarks

Notice that the relator $\subseteq$ and the enumerative set-former operator $\{\_, \_, \ldots, \_\}$ are easily expressible in **MLSS**. Indeed, $s \subseteq t$ is equivalent to $s \cup t = t$ and the term $\{t_1, t_2, \ldots, t_k\}$ can be rewritten as $\{t_1\} \cup \ldots \cup \{t_k\}$.

We will make use of the abbreviations $s \notin t$ and $s \neq t$ to denote the literals $\neg(s \in t)$ and $\neg(s = t)$, respectively. □

The decision problem

The *(satisfiability) decision problem* for a collection $\mathcal{C}$ of set-theoretic formulas is the problem of establishing whether or not φ has a model, for any given formula φ in $\mathcal{C}$.

To solve this problem positively, one must design an algorithm that can test any φ in $\mathcal{C}$ for satisfiability. Of course, for a specific $\mathcal{C}$, such an algorithm may not exist.

We present a tableau calculus for **MLSS**, denoted $\mathfrak{T}_{\text{MLSS}}$, which is based upon the system **KE**.

The *linear* rules of $\mathfrak{T}_{\text{MLSS}}$, namely those rules which do not cause branch splittings, are listed below. Note that the index i in the $\cup$- and $\cap$-rules can assume the values **1** and **2**, whereas ℓ in the $=$-rules stands for any **MLSS** literal.

A tableau calculus for MLSS

prop rules:	$\dfrac{\alpha}{\alpha_1}$	$\dfrac{\alpha}{\alpha_2}$	$\dfrac{\beta \quad \beta_1^c}{\beta_2}$	$\dfrac{\beta \quad \beta_2^c}{\beta_1}$
$\cup$-rules:	$\dfrac{s \in t_1 \cup t_2 \quad s \notin t_i}{s \in t_{3-i}}$	$\dfrac{s \notin t_1 \cup t_2}{s \notin t_i}$	$\dfrac{s \in t_i}{s \in t_1 \cup t_2}$	$\dfrac{s \notin t_1 \quad s \notin t_2}{s \notin t_1 \cup t_2}$
$\cap$-rules:	$\dfrac{s \notin t_1 \cap t_2 \quad s \in t_i}{s \notin t_{3-i}}$	$\dfrac{s \in t_1 \cap t_2}{s \in t_i}$	$\dfrac{s \notin t_i}{s \notin t_1 \cap t_2}$	$\dfrac{s \in t_1 \quad s \in t_2}{s \in t_1 \cap t_2}$

Table: $\mathfrak{T}_{\mathsf{MLSS}}$-linear rules (part a)

$\setminus$-rules:	$\dfrac{s \notin t_1 \setminus t_2 \quad s \in t_1}{s \in t_2}$	$\dfrac{s \in t_1 \setminus t_2}{s \in t_1}$	$\dfrac{s \notin t_1}{s \notin t_1 \setminus t_2}$	$\dfrac{s \in t_1 \quad s \notin t_2}{s \in t_1 \setminus t_2}$
	$\dfrac{s \notin t_1 \setminus t_2 \quad s \notin t_2}{s \notin t_1}$	$\dfrac{s \in t_1 \setminus t_2}{s \notin t_2}$	$\dfrac{s \in t_2}{s \notin t_1 \setminus t_2}$	
$\{\_\}$-rules:	$\dfrac{s \in \{t_1\}}{s = t_1}$	$\dfrac{s \notin \{t_1\}}{s \neq t_1}$	$\dfrac{}{t_1 \in \{t_1\}}$	
$=$-rules:	$\dfrac{t_1 = t_2 \quad \ell}{\ell^{t_1}_{t_2}}$	$\dfrac{t_1 = t_2 \quad \ell}{\ell^{t_2}_{t_1}}$	$\dfrac{s \in t \quad s' \notin t}{s \neq s'}$	

Table: $\mathfrak{T}_{\mathbf{MLSS}}$-linear rules (part b)

The *branching* rules of $\mathfrak{T}_{\mathbf{MLSS}}$ are listed next. Note that rules **(β1)**, **(β2)**, and **($\in$)** are cut rules, whereas rule **(*ext*)** is not. In particular, in rule **(*ext*)**, the symbol ***c*** denotes a *new* set constant, one not already occurring in the branch to which the rule is applied.

$$\frac{\beta}{\beta_1 \mid \beta_1^c}\ (\beta 1) \qquad \frac{\beta}{\beta_2 \mid \beta_2^c}\ (\beta 2)$$

$$\frac{}{s \in t \mid s \notin t}\ (\in) \qquad \frac{t_1 \neq t_2}{\begin{array}{c|c} c \in t_1 & c \notin t_1 \\ c \notin t_2 & c \in t_2 \end{array}}\ (ext)$$

Table: $\mathfrak{T}_{\mathsf{MLSS}}$-branching rules

Remark

In rule **(*ext*)**, c stands for a *new* uninterpreted set constant.

Next we define how to construct $\mathfrak{T}_{\mathsf{MLSS}}$-tableaux.

Definition

Let φ be an **MLSS**-sentence. The *initial* $\mathfrak{T}_{\mathsf{MLSS}}$*-tableau* for φ is the single-node tree whose root is labeled by φ.
An $\mathfrak{T}_{\mathsf{MLSS}}$-*tableau* for φ is a tableau labeled with **MLSS**-sentences, which can be constructed from the initial tableau for φ by a finite number of applications of the rules of $\mathfrak{T}_{\mathsf{MLSS}}$. □

Closure conditions must take into account also the semantics of set theory, as in the following definition.

Definition

A branch of a $\mathfrak{T}_{\mathsf{MLSS}}$-tableau is *closed* if it contains either

- two complementary sentences ψ and $\neg\psi$, or
- a finite membership cycle of the form $t_0 \in t_1 \in \ldots \in t_n \in t_0$, or
- a literal of the form $t \neq t$, or
- a literal of the form $t \in \emptyset$.

A tableau is *closed* if all its branches are closed. □

Tableau proofs and refutations are defined in the standard way.

Definition

A $\mathfrak{T}_{\mathsf{MLSS}}$-*tableau proof* for an $\mathbf{MLSS}$-sentence φ is any closed $\mathfrak{T}_{\mathsf{MLSS}}$-tableau for $\neg\varphi$.

A $\mathfrak{T}_{\mathsf{MLSS}}$-*tableau refutation* for φ is any closed $\mathfrak{T}_{\mathsf{MLSS}}$-tableau for φ. □

Our next task is to show that the tableau calculus $\mathfrak{T}_{\mathsf{MLSS}}$ captures the semantics of $\mathbf{MLSS}$ exactly, namely it is both *sound* and *complete*. In fact, completeness will be proved under some restrictions which will render the calculus suitable for effective use as a decision procedure for $\mathbf{MLSS}$.

Soundness of the tableau calculus $\mathfrak{T}_{\mathsf{MLSS}}$ can easily be proved by showing that

(a) at least one of the extensions of any satisfiable branch is satisfiable, and

(b) no closed branch is satisfiable,

where a branch ϑ of a $\mathfrak{T}_{\mathsf{MLSS}}$-tableau is said to be *satisfiable* if there exists a set model which makes true all sentences in ϑ.

Property (b) follows by observing that *all closure conditions are unsatisfiable in the standard von Neumann universe*.

Concerning (a), for the sake of simplicity we will only consider the case in which a satisfiable branch ϑ is extended by an application of the branching rule **(*ext*)**, with the literal $t_1 \neq t_2$ as premiss, using the set constant c (not occurring in ϑ).

Let ϑ_1 and ϑ_2 be the two branches which extend ϑ and let M be a set model satisfying ϑ. Since $Mt_1 \neq Mt_2$, by extensionality there exists a set **e** which belongs to the symmetric difference of Mt_1 and Mt_2.
Let M' be the assignment such that $M'c = \mathbf{e}$ and otherwise takes the same values as M. Since the set constant c does not occur in any sentence in ϑ, it is plain that M' must satisfy either ϑ_1 or ϑ_2.

Other cases can be treated similarly. Thus we have:

Theorem (Soundness)

The tableau calculus $\mathfrak{T}_{MLSS}$ for MLSS is sound. □

Exercise

Complete the proof of soundness of the tableau calculus $\mathfrak{T}_{MLSS}$.

Some of the rules of the tableau calculus $\mathfrak{T}_{\mathbf{MLSS}}$, if not suitably restricted, can cause a combinatorial explosion of the number of branches in an attempt to find a closed tableau for a given **MLSS**-sentence.
This is the case, for instance, for the linear rules

$$\frac{s \in t_i}{s \in t_1 \cup t_2} \qquad \frac{s \notin t_i}{s \notin t_1 \cap t_2} \qquad \frac{s \notin t_1}{s \notin t_1 \setminus t_2} \qquad \frac{}{t_1 \in \{t_1\}}$$

which can cause the introduction of an unbounded number of new terms.

Hence, some restrictions need to be imposed on rule applicability.
The following is a quite strong restriction which completely takes care of the first problem outlined above:

R1. during the construction of a $\mathfrak{T}_{\mathbf{MLSS}}$-tableau **T** for an **MLSS**-sentence φ, *no* new compound set term can be introduced in **T** by any linear or branching rule.

To further optimize our tableau system $\mathcal{T}_{\mathbf{MLSS}}$, we will impose some restrictions on the applicability of the branching rules **(∈)** and **(*ext*)**.

It is convenient to introduce the following definition:

Definition

Let $\mathbf{T}$ be an $\mathcal{T}_{\mathbf{MLSS}}$-tableau for φ. An *unfulfilled item* of a branch ϑ of $\mathbf{T}$ is either

- a sentence β in ϑ such that *none* of its components β_1 and β_2 occurs in ϑ;
  or
- a literal $t_1 \neq t_2$ in ϑ such that
  (b1) t_1 and t_2 occur in φ; and
  (b2) there exists no term s such that the branch ϑ contains both $s \in t_i$ and $s \notin t_{3-i}$, for some $i \in \{1, 2\}$; ./.

Definition (cntd)

or

- an ordered pair (s, t) of terms occurring in ϑ such that
  - **(c1)** neither $s \in t$ nor $s \notin t$ is in ϑ; and
  - **(c2)** for some terms t' and t'' in φ, either
    - **(c2.i)** ϑ contains the literal $s \in t \cup t'$, or
    - **(c2.ii)** t has the form $t' \setminus t''$ and $s \in t'$ occurs in ϑ, or
    - **(c2.iii)** t has the form $t' \cap t''$ and $s \in t'$ occurs in ϑ.

A branch ϑ is *fulfilled* if it has no unfulfilled item.
A tableau is *fulfilled* if all its branches are fulfilled. □

$$\frac{s \in t \cup t'}{s \in t \mid s \notin t}\ (c2.i) \qquad \frac{s \in t'}{s \in t' \setminus t'' \mid s \notin t' \setminus t''}\ (c2.ii)$$

$$\frac{s \in t'}{s \in t' \cap t'' \mid s \notin t' \cap t''}\ (c2.iii)$$

The idea is that only branching rules which cause a previously unfulfilled item to become fulfilled are allowed. More precisely, we impose the following further restriction:

R2. during the construction of a $\mathfrak{T}_{\mathbf{MLSS}}$-tableau **T** for an **MLSS**-sentence φ, the branching rules can be used to extend a branch ϑ of **T** only if they are applied to *unfulfilled* items of ϑ, where, by convention, when the cut rule $(\in)$ is applied with the sentences $s \in t$ and $s \notin t$, we say that it has been applied to the ordered pair (s, t).

A further optimization can be achieved by imposing the following restriction on the first two $=$-rules:

R3. during the construction of a $\mathfrak{T}_{\mathbf{MLSS}}$-tableau **T** for an **MLSS**-sentence φ, in the first two $=$-rules the substituted term is restricted to being a top-level term of the literal ℓ.

Thus, for instance, restriction **R3** allows the substitution only of the terms t_1 and t_2 in the literal $t_1 \notin t_2$ by means of a $=$-rule.

Definition

A $\mathfrak{T}_{\mathbf{MLSS}}$-tableau is *restricted* if, during its construction, the above restrictions **R1** through **R3** have been observed. □

Example

1: $\neg(\{c_1\} = c_1 \cup c_2 \rightarrow (c_1 = \emptyset \wedge c_2 = \{c_1\}))$

2: $\{c_1\} = c_1 \cup c_2$

3: $\neg(c_1 = \emptyset \wedge c_2 = \{c_1\})$

4: $c_1 \in \{c_1\}$

5: $c_1 \in c_1 \cup c_2$

6: $c_1 \in c_1$	7: $c_1 \notin c_1$
$\bot$	8: $c_1 \in c_2$
	$\vdots$

Table 1: A $\mathfrak{T}_{\mathbf{MLSS}}$-tableau proof of
$\{c_1\} = c_1 \cup c_2 \rightarrow (c_1 = \emptyset \wedge c_2 = \{c_1\})$

Example (cntd)

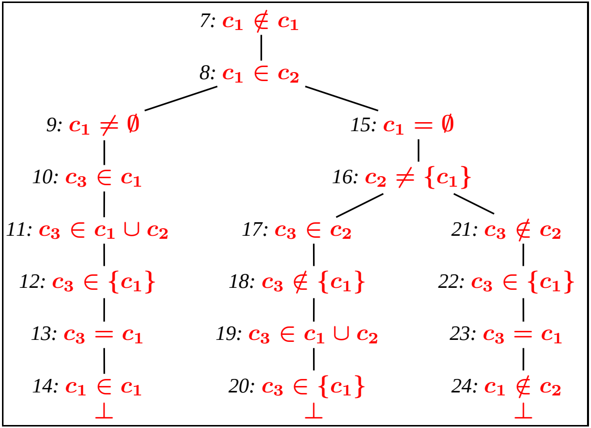

Table 1: A $\mathfrak{T}_{\mathbf{MLSS}}$-tableau proof of $\{c_1\} = c_1 \cup c_2 \rightarrow (c_1 = \emptyset \wedge c_2 = \{c_1\})$ (cntd)

Definition

A branch of a $\mathfrak{T}_{\mathbf{MLSS}}$-tableau is

- *linearly saturated*, if no new sentence can be added to it by any application of a linear rule complying with restrictions **R1** and **R3**;
- *saturated*, if it is linearly saturated and it does not contain any unfulfilled item.

Likewise, a $\mathfrak{T}_{\mathbf{MLSS}}$-tableau is

- *linearly saturated*, if all its branches are linearly saturated;
- *saturated*, if all its branches are saturated.

A $\mathfrak{T}_{\mathbf{MLSS}}$-tableau is *s-restricted* if it is restricted *and* saturated. □

Completeness of the tableau calculus $\mathfrak{T}_{\mathsf{MLSS}}$ will be proved

- by exhibiting a saturation process which, given an **MLSS**-sentence φ, constructs a restricted $\mathfrak{T}_{\mathsf{MLSS}}$-tableau for φ which is either closed or saturated; and
- by showing that any open and s-restricted $\mathfrak{T}_{\mathsf{MLSS}}$-tableau is satisfiable, namely it has at least one satisfiable branch.

Let us consider the following procedure for the construction of restricted $\mathfrak{T}_{\mathbf{MLSS}}$-tableaux, based on the procedure KE-Saturate

procedure $\mathfrak{T}$-Saturate(φ);
- let **T** be the initial tableau for φ;

repeat
- linearly saturate **T**;

if **T** has an unfulfilled branch ϑ **then**
- select an unfulfilled item χ of ϑ;
- apply the appropriate branching rule for χ on ϑ;

end if;
until **T** is either closed or saturated;
return T;
end $\mathfrak{T}$-Saturate;

Remark

We will assume that the linear saturation phase which takes place just at the beginning of the **repeat-until** block is *regular*, i.e., it never introduces on a branch a literal which already occurs in it.

Termination

- To prove that procedure $\mathfrak{T}$-Saturate terminates, by König's lemma it is enough to show that at any time during its execution for the construction of a tableau **T** for φ, each branch of **T** has bounded length.
- Accordingly, let ϑ be a branch of **T** and let T_φ and T_ϑ be the collections of set terms occurring in φ and ϑ, respectively.
- In view of restriction **R1**, the only new terms introduced in ϑ are the set constants added by the branching rule **(*ext*)**.
- Since, by restriction **R2**, rule **(*ext*)** can be applied on ϑ at most $|T_\varphi|^2$ times, it follows that $|T_\vartheta| \leq |T_\varphi| + |T_\varphi|^2$.
- Hence, the number of literals in ϑ and, in turn the length of ϑ, can easily be bounded in terms of $|T_\varphi|$. □

Let $G = \langle N, \widehat{\in} \rangle$ be a *directed acyclic* graph (*dag*, for short), let $\{V, T\}$ be a partition of N, and let $\{u_t : t \in T\}$ be a family of sets indexed over T.

Edges can form *any* acyclic dyadic relation on N, but we denote their collection as $\widehat{\in}$ to suggest that they will be interpreted as *set-membership constraints*.

Since the graph G is *acyclic*, the following definitions are well posed:

Definition (REALIZATIONS)

The REALIZATION of $G = \langle N, \widehat{\in} \rangle$ relative to $\{u_t : t \in T\}$ and to V, T is the assignment R recursively defined over $N = V \cup T$ as follows:

$$
\begin{aligned}
Rx &=_{\text{Def}} \{Rz : z \in V \cup T \mid z \widehat{\in} x\}, && \text{for } x \text{ in } V;\\
Rt &=_{\text{Def}} \{Rz : z \in V \cup T \mid z \widehat{\in} t\} \cup \{u_t\}, && \text{for } t \text{ in } T.
\end{aligned}
$$

The HEIGHT of a v in N is

$$
height(v) =_{\text{Def}} \begin{cases} 0 & \text{if } y \,\widehat{\notin}\, v \text{ for any } y \text{ in } N, \\ \max\{height(y) + 1 : y \in N \mid y \widehat{\in} v\} & \text{otherwise.} \end{cases}
$$

Lemma (Realizations' Lemma)

Let $G = \langle V \cup T, \widehat{\in} \rangle$ *be a dag with* $V \cap T = \emptyset$. *Also, let* $\{u_t : t \in T\}$ *be a given family of* $\mathcal{U}$*-sets, and let* R *be the* realization *of* G *relative to* $\{u_t : t \in T\}$ *and to* V, T.
Assume that

- **(a)** $u_t \neq u_d$, *for all distinct* t, d *in* T;
- **(b)** $u_t \neq Rv$, *for all* t *in* T *and all* v *in* $V \cup T$.

Then:

- **(i)** $Rt \neq Rt'$, *for all distinct* $t, t' \in T$;
- **(ii)** *if* $Rx = Ry$, *then* $height(x) = height(y)$, *for* x, y *in* $V \cup T$;
- **(iii)** *if* $Rx \in Ry$, *then* $height(x) < height(y)$, *for* x, y *in* $V \cup T$.

Proof of the Realizations' Lemma

(i) follows immediately from (a) and (b) above.

Concerning *(ii)*, we can proceed by induction on

$$\max\{height(x), height(y)\}\,.$$

If $\max\{height(x), height(y)\} = 0$, then *(ii)* is trivially true.
As for the inductive step, notice that if $Rx = Ry$ then for each $w \widehat{\in} x$ there exists a $v \widehat{\in} y$ such that $Rw = Rv$, and symmetrically.
Therefore,

$$\begin{aligned} height(x) &= \max\{height(w) + 1 : w \widehat{\in} x\} \\ &= \max\{height(v) + 1 : v \widehat{\in} y\} \quad = \quad height(y)\,. \end{aligned}$$

./.

Proof of the Realizations' Lemma (cntd)

As for *(iii)*, let $Rx \in Ry$. Then $Rx = Rz$, for some $z \widehat{\in} y$.
Hence, $height(x) = height(z) < height(y)$. □

Let T be an open and s-restricted $\mathfrak{T}_{\mathsf{MLSS}}$-tableau for φ and let ϑ be an open branch of T. It is convenient to associate with ϑ and φ the following objects:

- T_φ: the collection of all terms occurring in φ;
- C_ϑ: the collection of the new set constants added to ϑ, namely those constants not occurring in φ;
- C'_ϑ: the collection of the set constants c in C_ϑ such that for no term $t \in T_\varphi$ either $c = t$ or $t = c$ occurs in ϑ;
- T'_ϑ: the set $T_\varphi \cup (C_\vartheta \setminus C'_\vartheta)$;
- G_ϑ: the directed acyclic graph $\langle C'_\vartheta \cup T'_\vartheta, \widehat{\in} \rangle$, where $s \widehat{\in} t$ iff the literal $s \in t$ occurs in ϑ; notice that the acyclicity of G_ϑ follows from the fact that the branch ϑ is open and therefore it cannot contain any membership cycle;

./.

R_ϑ: a *realization* of G_ϑ relative to C'_ϑ, T'_ϑ and to *pairwise distinct* sets $\mathbf{u}_c$, for $c \in C'_\vartheta$, each having cardinality *no less* than $|C'_\vartheta \cup T'_\vartheta|$, defined by

$$R_\vartheta t =_{\text{Def}} \{R_\vartheta s : s \in T'_\vartheta \cup C'_\vartheta \mid s \widehat{\in} t\}, \quad \text{for } t \text{ in } T'_\vartheta$$

$$R_\vartheta c =_{\text{Def}} \{R_\vartheta s : s \in T'_\vartheta \cup C'_\vartheta \mid s \widehat{\in} c\} \cup \{\mathbf{u}_c\}, \quad \text{for } c \text{ in } C'_\vartheta$$

so that $\mathbf{u}_c \neq R_\vartheta t$, for every $c \in C'_\vartheta$ and $t \in T'_\vartheta \cup C'_\vartheta$.

The following lemma can be proved by induction on the length of ϑ.

Lemma

If $c \in C'_\vartheta$ then there can be no term t in $T_\varphi \cup C_\vartheta$ such that either $c = t$, or $t = c$, or $t \in c$ occurs in ϑ. □

Exercise

Prove the preceding lemma.

In order to show that the realization R_ϑ satisfies ϑ, we begin by proving that it models correctly all literals in ϑ, at least in the case in which compound terms are not interpreted, i.e. they are treated as if they were just “complex names” for constant symbols.

Lemma

The following assertions hold:

(i) *if* $s \in t$ *occurs in* ϑ, *then* $R_\vartheta s \in R_\vartheta t$;

(ii) *if* $s \notin t$ *occurs in* ϑ, *then* $R_\vartheta s \notin R_\vartheta t$;

(iii) *if* $t_1 = t_2$ *occurs in* ϑ, *then* $R_\vartheta t_1 = R_\vartheta t_2$;

(iv) *if* $t_1 \neq t_2$ *occurs in* ϑ, *then* $R_\vartheta t_1 \neq R_\vartheta t_2$.

Proof of (i) and (iii)

- Assertion (i) follows directly from the definition of R_ϑ.

  It is more convenient to prove (iii) and (iv) before (ii).

- Concerning (iii), let $t_1 = t_2$ be in ϑ. By the preceding lemma, $t_1, t_2 \in T'_\vartheta$. If, by contradiction, $R_\vartheta t_1 \neq R_\vartheta t_2$, then without loss of generality we can assume that there exists a term s such that the literal $s \in t_1$ is in ϑ and $R_\vartheta s \notin R_\vartheta t_2$. But ϑ is linearly saturated, thus it must also contain the literal $s \in t_2$, so that $R_\vartheta s \in R_\vartheta t_2$, a contradiction.

Proof of (iv)

- To show that (iv) holds, let $t_1 \neq t_2$ be in ϑ. Let us first assume that $t_1 \in C'_\vartheta$. Then, $\mathbf{u}_{t_1} \in R_\vartheta t_1$. Since ϑ is open, it follows that the terms t_1 and t_2 are distinct, so that $\mathbf{u}_{t_1} \notin R_\vartheta t_2$, which in turn yields $R_\vartheta t_1 \neq R_\vartheta t_2$. Analogous conclusion can be reached in the case in which $t_2 \in C'_\vartheta$.
  To conclude the proof of (iv), it remains to show that the set

  $$\Delta_\vartheta =_{\mathrm{Def}} \{(\tau_1, \tau_2) : \tau_1, \tau_2 \in T'_\vartheta \mid \tau_1 \neq \tau_2 \text{ is in } \vartheta \text{ and } R_\vartheta t_1 = R_\vartheta t_2\}$$

  is empty.
  Let us assume, by way of contradiction, that $\Delta_\vartheta \neq \emptyset$ and put

  $$h_\vartheta(\tau_1, \tau_2) =_{\mathrm{Def}} \min\{\mathit{height}(\tau_1), \mathit{height}(\tau_2)\},$$

  for $\tau_1, \tau_2 \in T'_\vartheta$. ./.

Proof of (iv) - cntd

- Let $(t_1, t_2) \in \Delta_\vartheta$ be such that

$$h_\vartheta(t_1, t_2) = \min_{(\tau_1,\tau_2)\in\Delta_\vartheta} h(\tau_1, \tau_2)\,.$$

It is easy to see that since the tableau $\mathbf{T}$ is restricted, there must exist two terms $t'_1, t'_2 \in T_\varphi$ such that $R_\vartheta t'_i = R_\vartheta t_i$, for $i = 1, 2$, and the literal $t'_1 \neq t'_2$ occurs in ϑ. Hence, $(t'_1, t'_2) \in \Delta_\vartheta$ and, by (ii) of the *Realizations' Lemma*, $h_\vartheta(t'_1, t'_2) = h_\vartheta(t_1, t_2)$.
Due to the fact that the literal $t'_1 \neq t'_2$ is a fulfilled item of ϑ, we can assume without loss of generality that there exists a term s_2 such that both the literals $s_2 \in t'_2$ and $s_2 \notin t'_1$ occur in ϑ. Hence, $R_\vartheta s_2 \in R_\vartheta t'_2 = R_\vartheta t'_1$, so that, since ϑ is open, there must exist a term s_1 such that the literal $s_1 \in t'_1$ is in ϑ and $R_\vartheta s_1 = R_\vartheta s_2$. ./.

Proof of (iv) - cntd

- As ϑ is linearly saturated and open, it must contain the literal $s_1 \neq s_2$ and we must also have that $s_1, s_2 \in T'_\vartheta$. Thus, $(s_1, s_2) \in \Delta_\vartheta$, which is a contradiction, as by (iii) of the *Realizations' Lemma* we have $h_\vartheta(s_1, s_2) < h_\vartheta(t_1, t_2)$.

Proof of (ii)

- Finally, let us show that (ii) holds. Let the literal $t_1 \notin t_2$ be in ϑ, but assume by contradiction that $R_\vartheta t_1 \in R_\vartheta t_2$. Hence, there must exist a term s such that the literal $s \in t_2$ is in ϑ and $R_\vartheta s = R_\vartheta t_1$. Owing to the fact that ϑ is linearly saturated, ϑ must also contain the literal $s \neq t_1$, so that, by (iv) above, $R_\vartheta s \neq R_\vartheta t_1$, which is a contradiction. □

Next we will show that even set operators are correctly modeled by R_ϑ.

Lemma

Every compound set term s occurring in ϑ is modeled correctly by the realization R_ϑ, namely if s has the form $t_1 \star t_2$, with $\star \in \{\cup, \cap, \setminus\}$, then $R_\vartheta s = R_\vartheta t_1 \star R_\vartheta t_2$, whereas if s has the form $\{t\}$, then $R_\vartheta s = \{R_\vartheta t\}$.
In addition, $R_\vartheta \emptyset = \emptyset$, provided that the constant $\emptyset$ occurs in ϑ.

Proof

Plainly, since ϑ is open, no sentence of the form $s \in \emptyset$ can occur in ϑ, so that $R_\vartheta \emptyset = \emptyset$, provided that the constant $\emptyset$ occurs in ϑ.
Concerning compound terms t, we proceed by induction on the number of set operators occurring in t. Accordingly, let us assume that the realization R_ϑ models correctly all terms in ϑ with fewer than k set operators, with $k \geq 1$, and let t be a term in ϑ containing exactly k operators.

We will consider only the case in which t has the form $t_1 \cap t_2$ and leave the remaining cases to the reader as an exercise. To begin with, notice that, by restrictions **R1** and **R3**, it follows that $t_1, t_2 \in T_\varphi$. ./.

Proof (cntd)

Let $\mathbf{e} \in R_\vartheta(t_1 \cap t_2)$. Then there exists a term s such that $R_\vartheta s = \mathbf{e}$ and the literal $s \in t_1 \cap t_2$ occurs in ϑ. Since ϑ is saturated, both literals $s \in t_1$ and $s \in t_2$ must also occur in ϑ, so that, by property *(i)* of the preceding lemma, $\mathbf{e} = R_\vartheta s \in R_\vartheta t_1 \cap R_\vartheta t_2$, which in turn implies $R_\vartheta(t_1 \cap t_2) \subseteq R_\vartheta t_1 \cap R_\vartheta t_2$. ✓.

Proof (cntd)

Conversely, let $e \in R_\vartheta t_1 \cap R_\vartheta t_2$. Then, there must exist two terms s_1 and s_2 such that $R_\vartheta s_1 = R_\vartheta s_2 = e$ and the literals $s_1 \in t_1$, $s_2 \in t_2$ occur in ϑ. By saturation, either $s_1 \in t_1 \cap t_2$ or $s_1 \notin t_1 \cap t_2$ must occur in ϑ. In the latter case, $s_1 \notin t_2$, and therefore also $s_2 \neq s_1$, must occur in ϑ, so that, by property (iv) of the preceding lemma, we have $R_\vartheta s_2 \neq R_\vartheta s_1$, a contradiction. Hence, the literal $s_1 \in t_1 \cap t_2$ must occur in ϑ, so that $e = R_\vartheta s_1 \in R_\vartheta(t_1 \cap t_2)$. Thus, $R_\vartheta t_1 \cap R_\vartheta t_2 \subseteq R_\vartheta(t_1 \cap t_2)$ which, together with the inverse inclusion established earlier, yields $R_\vartheta(t_1 \cap t_2) = R_\vartheta t_1 \cap R_\vartheta t_2$. □

Exercise

Complete the proof of the lemma.

So far we have shown that the realization $\boldsymbol{R_\vartheta}$ is a set model for all literals in ϑ. By induction, it can easily be shown that compound sentences in ϑ are also satisfied by $\boldsymbol{R_\vartheta}$, yielding:

Lemma

Let T be an open and s-restricted $\mathfrak{T}_{MLSS}$-tableau for an $MLSS$ sentence. Then any open branch of T is satisfiable. □

Exercise

Complete the proof of the lemma.

Thus, in view of the saturation process $\mathfrak{T}$-Saturate described earlier, we have

Theorem (Completeness)

The tableau calculus $\mathfrak{T}_{MLSS}$ for $MLSS$ is complete. □

The results discussed above imply immediately that the following is a decision procedure for **MLSS**:

procedure **MLSS**-decision test(φ);
- let **T** be the initial tableau for $\neg\varphi$;

T := $\mathfrak{T}$-Saturate(**T**);
if **T** is closed **then**
return "**T** is a $\mathfrak{T}_{MLSS}$-tableau proof for φ";
else
- let ϑ be an open branch of **T** and let R_ϑ be a realization associated with ϑ;

return "R_ϑ is a set model which falsifies φ";
end if;
end **MLSS**-decision test;

Remark

Notice that the $\mathfrak{T}_{\mathbf{MLSS}}$-tableau proof in our preceding example could be the possible output of the **MLSS**-decision test when applied to the **MLSS**-sentence
$\{c_1\} = c_1 \cup c_2 \rightarrow (c_1 = \emptyset \wedge c_2 = \{c_1\})$. □

In conclusion we have:

Theorem (Decidability of MLSS)

*The collection of **MLSS**-sentences has a solvable decision problem.* □