# ML systems: A Proof Theory for Contexts

Luciano Serafini (serafini@itc.it)
*ITC–IRST, Centro per la Ricerca Scientifica e Tecnologica, via Sommarive 18, 38050 Povo, Trento, Italy*

Fausto Giunchiglia (fausto@itc.it)
*DISA – Dipartimento di Informatica e Studia Aziendali, University of Trento, Via Inama 5, 38100 Trento, Italy*

June 5, 2000

**Abstract.** In the last decade the concept of context has been extensively exploited in many research areas, e.g., distributed artificial intelligence, multi agent systems, distributed databases, information integration, cognitive science, and epistemology. Three alternative approaches to the formalization of the notion of context have been proposed: Giunchiglia and Serafini's Multi Language Systems (ML systems), McCarthy's modal logics of contexts, and Gabbay's Labelled Deductive Systems. Previous papers have argued in favor of ML systems with respect to the other approaches. Our aim in this paper is to support these arguments from a theoretical perspective. We provide a very general definition of ML systems, which covers all the ML systems used in the literature, and we develop a proof theory for an important subclass of them: the MR systems. We prove various important results; among other things, we prove a normal form theorem, the sub-formula property, and the decidability of an important instance of the class of the MR systems. The paper concludes with a detailed comparison among the alternative approaches.

**Keywords:** Contextual Reasoning, Distributed Information-Oriented Theories, Modal Logics, Multi Context systems, Normal Form, Proof Theory

## 1. Introduction

We usually talk of distributed knowledge representation meaning that knowledge is composed of a set of heterogeneous subsystems, each representing a certain subset of the whole knowledge. Similarly, we talk of distributed reasoning meaning that reasoning is the result of the combination of partial reasoning processes each using only a subset of the global knowledge. One important paradigm which has been proposed for the formalization of distributed knowledge and reasoning is based on the notion of *context*. Contexts were introduced by R. Weyhrauch in (Weyhrauch, 1980) and subsequently developed by J. McCarthy (McCarthy and Buvač, 1998) and F. Giunchiglia (Giunchiglia, 1993) (see (Sharma, 1995) but also (Bouquet et al., 1999) for a survey of the notion of context).

According to (Giunchiglia, 1993), the notion of context formalizes the idea of localization of knowledge and reasoning. Intuitively speaking, a context is a set of facts (expressed in a suitable language, usually different for each different set of facts) used locally to prove a given goal, plus the inference routines used to reason about them (which can be different for different sets of facts). A context encodes a perspective about the world. It is a *partial perspective* as the complete description of the world is given by the set of all the contexts. It is an *approximate perspective* (in the sense described in (McCarthy, 1979)) as we never describe the world in full detail. Finally, different contexts, in general, are not independent of one another as the different perspectives are about the same world, and, as a consequence, the facts in a context are related to the facts in other contexts.

The work in (Giunchiglia and Serafini, 1994) provides a logic, called Multi Language Systems (ML Systems), formalizing the principles of reasoning with contexts informally described in (Giunchiglia, 1993). In ML systems, contexts are formalized using multiple distinct languages, each language being associated with its own theory (a set of formulas closed under a set of inference rules). Relations among different contexts are formalized using *bridge rules*, namely inference rules with premises and consequences in distinct languages. ML systems have been applied in many research areas. In (Giunchiglia, 1993; Benerecetti et al., 2000) they have been used to formalize general reasoning with contexts. In (Giunchiglia and Serafini, 1994) ML systems are shown to be an alternative from modal logics for the formalization of meta reasoning and propositional attitudes. In (Benerecetti et al., 1998a; Giunchiglia et al., 1993; Giunchiglia and Giunchiglia, 2000; Fisher and Ghidini, 1999), ML systems are used for the formalization of reasoning about beliefs. In (Bouquet and Giunchiglia, 1995) ML systems are used to formalize context based commonsense reasoning. (Dichev, 1993) provides an algebra of contexts formalized via ML systems. In (Cimatti and Serafini, 1996) and (Benerecetti et al., 1998b) flexible reasoning about static and dynamic aspects of multi agent systems are formalized via ML systems. In (Mylopoulos and Motschnig-Pitrik, 1995; Serafini and Ghidini, 2000; Subrahmanian, 1994), ML systems have been used as a formal basis for the specification distributed and heterogeneous multiple databases and for the partitioning of information bases. In (Noriega and Sierra, 1996; Parsons et al., 1998), ML systems have been used for modelling dialog and argumentation in electronic commerce. In (Giunchiglia and Bouquet, 1998), ML systems have been used for the formalization of mental representation in cognitive science. Finally in (Penco, 1999) ML systems has been suggested as a formalism for studying a subjective view of contexts as an alternative then objective contexts.

The goal of this paper is to make a first step towards the development of a proof theory for ML systems. We concentrate on a particular subclass of ML systems, called MR systems. Inside this class we single out a simple, but significant, MR system called MK and develop a proof theory for it. The main results are a *weak normal form theorem* (Theorem 32) and a *strong normal form theorem* (Theorem 40). The weak normal form characterizes the set of contexts in which one has to deduce a formula from a set of assumptions. An immediate consequence of the weak normal form is the *consistency of MK* (Corollary 33). The strong normal form result describes a standard shape of the deductions in MK, and it provides a complete strategy for inference rule applications. The main consequence of the strong normal form is the *subformula property* (Theorem 51), which, in turns, entails decidability of MK. MR systems, and these results are important for at least two reasons. First, MR systems constitute a large class, which encompasses most of the ML systems used in the literature. Second, as it has been shown in (Giunchiglia and Serafini, 1994), normal modal logics can be embedded in MR systems. This implies that the proof theoretical results provided in this paper are relevant also for modal logics. In the related work section we indeed show how MK constitutes a more natural alternative then other calculi for many of the modal logics proposed in the literature.

The paper is structured as follows. We start by introducing the basic definitions of ML systems (Section 2. In Section 3 we define the class of MR systems. In Section 4 we prove the main results of the paper, namely, the weak and strong normal form, and the sub-formula property. We conclude the paper with a comparison of ML systems with other similar formalisms, and in particular with Gabbay's Labelled

Deductive Systems (Gabbay, 1994), McCarthy's Propositional Logics of Contexts (McCarthy, 1993; Buvac and Mason, 1993), and Masini's 2-Sequent Calculi (Masini, 1992).

## 2. ML Systems

The definition of ML system is given in two steps. First we provide an abstract representation in terms of languages (set of well formed formulas) and consequence relation (relations between formulas). Then, we move to a concrete representation, where the consequence relation is represented by a Natural Deduction calculus.

Abstractly, a single language formal system is composed by a language and a consequence relation on the formulas of the language. Similar components define an ML system. Let $I$ be a not empty set of *indices.* $\{L_i\}_{i\in I}$ (hereafter $\{L_i\}$) is a family of logical languages. Intuitively $I$ is a set of labels for the contexts in which the whole knowledge is partitioned, and $L_i$ is the language adopted to express the facts in the $i$-th context.

Two occurrences of the same formula in two distinct contexts may have different meanings. To distinguish the formula $A$ occurring in the context $i$ from the occurrences of $A$ in the other contexts, we write $i : A$. We say that $i : A$ is a wff and that $A$ is an $L_i$-wff. If $G$ is a set of $L_i$-wffs, $i : G$ denotes the set of wffs $\{i : A | A \in G\}$. $L_I$ denotes the set of all wffs, namely $\cup_{i\in I}(i : L_i)$. Similar notations have been introduced in various approaches; see for instance (Gabbay, 1994; Subrahmanian, 1994; Dinsmore, 1991; Masini, 1992; McCarthy and Buvač, 1998). In ML systems, indexes are not part of the languages. They are, rather, a "metanotation" useful proof-theoretically to keep track of the locality of the reasoning. See the related work section for a more detailed discussion.

We define the consequence relation for ML systems as follows. We start from Avron's general definition of consequence relation for a logic with one language (as described in (Avron, 1987)) and extend it to a *logic on multiple languages.*

DEFINITION 1. A *Multi-Language Consequence Relation (ML–C.R.)* on a family of languages $\{L_i\}$ is a relation, denoted by $\vdash$, between sets of wffs and wffs (namely, $\vdash \subseteq 2^{L_I} \times L_I$) such that for all $i$:

| | |
|---|---|
| *ML–RX(i)* | $\Gamma, i : A \vdash i : A$ |
| *ML–CUT(i)* | if $\Gamma \vdash i : A$, and $\Gamma, i : A \vdash i : B$, then $\Gamma \vdash i : B$ |

DEFINITION 2. An *abstract ML system* is a tuple $\langle \{L_i\}, \vdash \rangle$, where $\vdash$ is a ML–C.R. on $\{L_i\}$.

We adopt a *concrete representation* of an ML–C.R.. based on Natural Deduction (ND) (Prawitz, 1965). The deduction machinery of an ML system is, therefore, composed of a set of ND inference rules on indexed formulas, called *Multi Language inference rules* (ML inference rules). ML inference rules are specified as follows:

$$\frac{\begin{array}{c}[\Gamma_1]\\ i_1 : A_1\end{array} \quad \cdots \quad \begin{array}{c}[\Gamma_n]\\ i_n : A_n\end{array}}{i : A}\ \text{I}$$

$A$ and $A_k$ are formula schemata, and $\Gamma_k$ is a finite set of indexed formula schemata. If empty, $\Gamma_k$ is omitted. $i_k : A_k$ is called the $k$-th premise of I; $i : A$ is the consequence (or conclusion) of I; each $\Gamma_k$ is the set of assumptions of the $k$-th premise discharged by I. An ML inference rule can be associated with a restriction (or applicability condition), namely a criterion which states the conditions of its applicability. We call *unrestricted* the inference rules with no applicability condition. The reader can find the formal definitions about ML inference rules in Appendix A.

DEFINITION 3. A concrete ML System is a triple $\langle\{L_i\}, \{\Omega_i\}, \Delta\rangle$ where $\{L_i\}$ is a *family of logical Languages*, $\{\Omega_i \subseteq L_i\}$ is a *family of set of axioms* and $\Delta$ is a set of ML inference rules.[1]

We can distinguish two types of ML inference rules: rules with premises, conclusion, discharged assumptions and restriction in the same language $L_i$, called *$i$-rules*, and rules with premises, conclusion, discharged assumptions or restriction in more than one language, called *bridge rules*. $i$-rules represent the part of a ML–C.R. which involves a single language. They can be viewed as the translation of the single language inference rules into the ML format. For instance the $i$-rules defined in (1) are the ML versions in $L_i$ of the ND inference rules $\supset$E, $\supset$I and $\bot_c$ defined in (Prawitz, 1965).

$$\frac{i : A \quad i : A \supset B}{i : B}\ \supset\mathrm{E}_i \qquad \frac{\begin{array}{c}[i : A]\\ i : B\end{array}}{i : A \supset B}\ \supset\mathrm{I}_i \qquad \frac{\begin{array}{c}[i : \neg A]\\ i : \bot\end{array}}{i : A}\ \bot_i \tag{1}$$

Bridge rules, instead, represent the inter-language properties of an ML–C.R.. Examples of bridge rules are:

$$\frac{i : A}{j : A}\ i \subseteq j \qquad \frac{i : A \quad j : A \supset B}{k : B}\ \supset\mathrm{E}_{i,j\to k} \qquad \frac{\begin{array}{c}[i : A]\\ j : B\end{array}}{j : A \supset B}\ \supset\mathrm{I}_{i\to j} \tag{2}$$

[1] In the rest of the paper, we write ML system meaning concrete ML system, and inference rule meaning ML inference rule.

$i \subseteq j$ and $\supset \mathrm{E}_{i,j \to k}$ are bridge rules as their premises and conclusion are in different languages. $\supset \mathrm{I}_{i \to j}$ is a bridge rule as it discharges an assumption in a language which is different from that of the premise and the conclusion.

Examples of simple and complex ML systems can be found in all the papers cited in the introduction. In the following we propose two examples with the purpose of explaining the basic underlying intuitions.

EXAMPLE 4. Let $\mathrm{ML}_3$ be an ML system defined on the set of indexes $I = \{1, 2, 3\}$, where: $L_1$, $L_2$ and $L_3$ are propositional languages such that $L_1 \cup L_2 \subseteq L_3$, and $\Delta$ contains the $i$-rules defined in (1) for $i = 1, 2, 3$, and the bridge rules $\supset \mathrm{E}_{1,2 \to 3}$ and $\supset \mathrm{E}_{2,1 \to 3}$, as defined in (2). $\mathrm{ML}_3$ can be seen as modeling a situation with three agents 1, 2, and 3. For each agent $i$, the formula $i : A$ represents the fact that $i$ knows that $A$ is true. The set of $i$-rules represent $i$'s reasoning capability. The bridge rules $\supset \mathrm{E}_{1,2 \to 3}$ and $\supset \mathrm{E}_{2,1 \to 3}$ represent the fact that 1 and 2 use their mutual knowledge and "communicate" the result to 3. Notice that 1 cannot communicate anything to 3 without agreeing with 2 and vice-versa.

EXAMPLE 5. Let $\mathrm{ML}'_3$ be an ML system obtained from $\mathrm{ML}_3$ by replacing its bridge rules with $1 \subseteq 3$ and $2 \subseteq 3$ as defined in (2). $1 \subseteq 3$ and $2 \subseteq 3$ can be taken to represent the fact that two agents 1 and 2 communicate their knowledge to 3 independently of each other. The result is that the knowledge of 3 contains that of 1 and 2.

EXAMPLE 6. Let $\mathrm{ML}''_3$ be obtained by adding to $\mathrm{ML}'_3$ the bridge rules $\supset \mathrm{I}_{1 \to 3}$ and $\supset \mathrm{I}_{2 \to 3}$ In this situation, not only do 1 and 2 autonomously communicate their knowledge to 3, but 3 can also perform hypothetical reasoning based on the knowledge of 1 and 2. The intuition underlying the bridge rule $\supset \mathrm{I}_{1 \to 3}$ is that 3 can conclude $A \supset B$ if it is able to infer $B$ under the hypothesis that 1 knows $A$. This can be done independently from the actual knowledge of 1 about $A$.

Figure 1 provides a graphical representation of the ML systems of Example 4 and Example 5.

We now define deductions in an ML system MS.

DEFINITION 7. A *formula-tree* in MS recursively defined as follows:

1. $i : A$ is a formula-tree;
2. if $\Pi_1, \ldots, \Pi_n$ are formula-trees then $\frac{\Pi_1, \ldots, \Pi_n}{i:A}$ is a formula-tree;

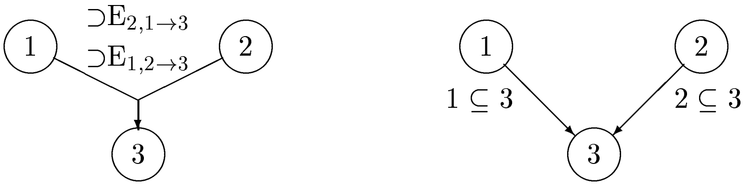


*Figure 1.* $\mathrm{ML}_3$ and $\mathrm{ML}'_3$

3. nothing else is a formula-tree.

The *top formulae* of a formula-tree $\Pi$ are the leaves of $\Pi$ and the *bottom formula* of $\Pi$ is the root of $\Pi$.

We write $\begin{array}{c}\Pi\\ i : A\end{array}$ to mean a formula-tree with bottom formula $i : A$. *Deductions* are formula-trees built by applying a finite number of inference rules to a finite number of assumptions and axioms, possibly belonging to distinct languages.

DEFINITION 8. A formula-tree is a *deduction in MS* of $i : A$ *depending on* a set of formulas, according to the following rules:

1. if $i : A$ is an axiom of MS, i.e. $A \in \Omega_i$, then $i : A$ is a deduction in MS of $i : A$ depending on the empty set;

2. if $i : A$ is not an axiom of MS, i.e. $A \notin \Omega_i$, then $i : A$ is a deduction in MS of $i : A$ depending on $\{i : A\}$;

3. if $\Pi_k$ is a deduction of $i_k : A_k$ depending on $\Gamma_k$ for any $(1 \leq k \leq n)$, then
$$\frac{\begin{array}{c}\Pi_1\\ i_1 : A_1\end{array} \quad \ldots \quad \begin{array}{c}\Pi_n\\ i_1 : A_1\end{array}}{i : A}\,\mathrm{I}$$
is a deduction in MS of $i : A$ depending on $\Gamma$ where:
$$\frac{\begin{array}{c}[\Gamma'_1]\\ i_1 : A_1\end{array} \quad \ldots \quad \begin{array}{c}[\Gamma'_n]\\ i_n : A_n\end{array}}{i : A}\,\mathrm{I}$$
is an instance of the inference rule I; I is applicable; $\Gamma = \cup_{1 \leq k \leq n}(\Gamma_k \setminus \Gamma'_k)$.

An ML system MS defines an ML–C.R., denoted by $\vdash_{\mathrm{MS}}$, defined in terms its deductions.

DEFINITION 9. A deduction $\Pi$ is a *deduction* in MS *of* $i : A$ *from* $\Gamma$, if and only if $\Pi$ is a deduction in MS of $i : A$ depending on $\Gamma$ or some subset of $\Gamma$.

$i : A$ is *derivable from* $\Gamma$ in MS, written as $\Gamma \vdash_{\mathrm{MS}} i : A$ if and only if there is a deduction in MS of $i : A$ from $\Gamma$.

A deduction of $i : A$ from the empty set is a *proof* of $i : A$. $i : A$ is *provable in* (*a theorem of*) MS, abbreviated as $\vdash_{\mathrm{MS}} i : A$, if and only if there exists a proof in MS of $i : A$.

With the above definitions it is easy to see that the relation $\vdash_{\mathrm{MS}}$ is an ML–C.R. In the following we talk about *i-theory*, or simply *theory*, meaning the set of theorems with index $i$ which are provable in an ML system.

DEFINITION 10. A *thread* of a deduction $\Pi$ is a sequence $\delta = i_1 : A_1, \ldots i_n : A_n$ such that:

1. $i_1 : A_1$ is a top formula of $\Pi$;
2. $i_n : A_n$ is the bottom formula of $\Pi$;
3. for every $1 \leq k < n$, $i_k : A_k$ and $i_{k+1} : A_{k+1}$ are a premise and the consequence of an application of an inference rule.

DEFINITION 11. $i : A$ is an *assumption* of a deduction $\Pi$ in MS, if and only if, it is a top formula of $\Pi$ and it is not an axiom of MS. An assumption $i : A$ is *(un)discharged in a deduction* according to the following rules:

1. $i : A$ is undischarged in $i : A$;
2. if $\Pi$ is of the form

$$\frac{\begin{array}{ccc}\Pi_1 & & \Pi_n \\ j_1 : B_1 & \ldots & j_n : B_n\end{array}}{j : B}\ \textsc{i}$$

then, if $i : A$ is discharged in $\Pi_k$, then it is discharged in $\Pi$. If $i : A$ is undischarged in $\Pi_k$, and $i : A$ is an assumption of the $k$-th premise discharged by $\textsc{i}$, then $i : A$ is discharged in $\Pi$ at $j : B$; otherwise $i : A$ is undischarged in $\Pi$.

EXAMPLE 12. (Examples 4 and 5 – continued) The following is a simple example of deduction in $\mathrm{ML}_3$.

$$\frac{1 : A \quad \dfrac{2 : A}{2 : A \supset A}\supset\mathrm{I}_2}{3 : A}\supset\mathrm{E}_{1,2\rightarrow 3} \tag{3}$$

(3) is a deduction of $3 : A$ from $1 : A$. $1 : A$ is an undischarged assumption. The assumption $2 : A$ is discharged by the application of $\supset I_2$. The threads of (3) are

$$\begin{aligned}\tau_1 &= 1 : A, 3 : A \\ \tau_2 &= 2 : A, 2 : A \supset A, 3 : A\end{aligned}$$

(3) shows that $1 : A \vdash_{ML_3} 3 : A$ under the hypothesis that $A$ is an $L_2$-wff. This deduction can be interpreted as the communication of 1 to 3 of the fact $A$, with the "permission" of 2. The "permission" is given by agent 2, only if it is "aware" of $A$, namely if $A$ belongs to the language of 2.

Notice that in general $ML_3$ and $ML'_3$ are not equivalent, namely they don't define the same ML–C.R. This is the case, for instance when, $L_1$ contains a formula, say $A$, which is not in $L_2$. In this case, indeed, $1 : A \vdash_{ML'_3} 3 : A$, but $1 : A \not\vdash_{ML_3} 3 : A$.

Conversely $ML'_3$ is stronger than $ML_3$, meaning that the ML–C.R. defined by the former contains the one defined by the latter. Notice indeed that $\supset E_{1,2\rightarrow 3}$ and $\supset E_{2,1\rightarrow 3}$ are derived inference rules in $ML'_3$. Consider, for instance, the following derivation in $ML'_3$

$$\dfrac{\dfrac{1 : A}{3 : A}\,1 \subseteq 3 \quad \dfrac{2 : A \supset B}{3 : A \supset B}\,2 \subseteq 3}{3 : B}\,\supset E_3$$

The above derivation states that $1 : A, 2 : A \supset B \vdash_{ML'_3} 3 : B$, which corresponds to the rule $\supset E_{1,2\rightarrow 3}$. In the limit case in which $L_1 \cap L_2 = \emptyset$, $\supset E_{1,2\rightarrow 3}$ and $\supset E_{2,1\rightarrow 3}$ are never applicable, as there is no wff $A \in L_1$ such that $A \supset B \in L_2$ and vice versa. If, vice-versa, $L_1 = L_2$, then $ML_3$ is equivalent to $ML'_3$.

This last example highlights the fact that the relation between languages in an ML system plays a crucial role with many interesting applications. For example, in the formalization of belief, we are often interested in non-omniscient agents whose reasoning capabilities are bounded by their languages. In a single language approach, a "boundary" in the language must be explicitly formalized by means of some meta-theoretic predicate or modal operator, whose meaning is given by a particular axiomatization (see for example the "*awareness*" operator introduced by Fagin and Halpern in (Fagin and Halpern, 1988)). In the ML approach this kind of limitation can be directly induced by the choice of the languages. An in depth discussion and several examples of ML systems for non-omniscient believers can be found in (Giunchiglia et al., 1993).

## 3. MR systems

Let us focus on the special class of ML systems called MR systems. After the basic definitions, in Subsection 3.2 we provide several instances of MR systems with the purpose of showing their relation with modal logics, but also how they can be (and have actually been) exploited for the formalization of relevant issues in distributed knowledge representation.

### 3.1. The Basic Definition

Informally MR systems formalize the situation where contexts are organized in a hierarchy. The hierarchical structure of contexts is formalized by an acyclic directional graph $\langle I, \prec \rangle$. Intuitively $i \prec j$ means that contexts $j$ contains facts about the facts of context $i$. This implies that if $i \prec j$, the language $L_j$ is a metalanguage of $L_j$. Bridge rules are admitted only between adjacent contexts (i.e., contexts labelled with $i$ and $j$, with $i \prec j$), and they are of the following simple form:

$$\frac{i : A}{j : \bullet_i(\text{``}A\text{''})}\ R_{\mathrm{up}.i} \qquad\qquad \frac{j : \bullet_i(\text{``}A\text{''})}{i : A}\ R_{\mathrm{dn}.i} \tag{4}$$

with any computable applicability restriction. In (4), $\bullet_i(\text{``}A\text{''})$ is an atomic, ground formula of $L_j$. We require that, if $A$ and $B$ are distinct formulas in $L_i$, then $\bullet_i(\text{``}A\text{''})$ and $\bullet_i(\text{``}B\text{''})$ are distinct formulas in $L_j$. Notice that, in $L_j$ there are infinite atomic formulas of the form $\bullet_i(\text{``}A\text{''})$, one for each formula of $L_i$.

The bridge rules of the form (4) are called *reflection rules*. $R_{\mathrm{up.}}$ is called reflection up, $R_{\mathrm{dn.}}$ reflection down. Reading "$\bullet_i$" as "provable in $i$", $R_{\mathrm{dn.}}$ and $R_{\mathrm{up.}}$ can be thought as statements of the soundness and completeness of the metatheory $j$ with respect to provability in $i$.

DEFINITION 13. An ML system MS $= \langle \{L_i\}, \{\Omega_i\}, \Delta \rangle$, is an MR system if and only if:

1. $I$ is a not empty at most enumerable set of indexes;
2. a recursive binary relation $\prec$ is defined on $I$, such that its transitive closure $\prec^+$ is not reflexive;
3. for each $i \prec j$, and for each $L_i$-wff $A$, $L_j$ contains an atomic ground formula $\bullet_i(\text{``}A\text{''})$, which is recursively computable from $A$, and for any $L_i$-wff $B$ distinct from $A$, $\bullet_i(\text{``}A\text{''})$ is distinct from $\bullet_i(\text{``}A\text{''})$;
4. the bridge rules of $\Delta$ are of the form (4), with $i \prec j$.

Condition 1 of Definition 13 ensures that the cardinality of the set of languages is at most enumerable; this restriction is justified by the fact that we want to avoid an uncountable set of formulas $\bullet_i(``A")$ (see condition 3). The first part of condition 2 guarantees that the applicability of $R_{\mathrm{up}.i}$ and $R_{\mathrm{dn}.i}$ are decidable; the second part of the same condition ensures that $\prec$ induces an acyclic graph on $I$, which prevents theories from being self-referential. 3 is the minimal expressivity condition under which $L_j$ can be used as metalanguage for a theory in $L_i$ (see (Giunchiglia et al., 1992) for more details). Condition 4 ensures that, if $i \not\prec j$ and $j \not\prec i$ then there are no bridge rules between $i$ and $j$, and that the only bridge rules are reflection rules. Reflection rules can have any form of restriction on their applicability. One such example is:

RESTRICTION ON THE PREMISE *a bridge rule is applicable only if there are no undischarged assumptions with index equal to that of the premises of the rule itself.*

In the following we will call *restricted* $R_{\mathrm{up.}}$ the reflection rule $R_{\mathrm{up.}}$ with the above restriction on the premise.

### 3.2. Some important MR systems

MPK is the basic propositional system for the formalization of theorem proving with metatheories (Giunchiglia and Traverso, 1996). In metatheoretic reasoning, one usually starts with the object theory and then defines its metatheory, its metametatheory and so on. Analogously, in MPK (see Figure 2), the bottom theory, labeled with 0, is any object theory in a propositional language $L$, the theory labeled with 1 is its simplest metatheory, the theory labeled with 2 is the simplest metatheory of that labeled with 1, i.e. the metametatheory of the bottom theory, and so on. The *simplest* metatheory of an object theory with language $L$, is a theory with the minimal linguistic and axiomatic requirements. The language of such a metatheory is $T(``L")$, namely the propositional language whose only atomic wffs are $\{T(``A") : A$ is an $L$-wff$\}$, its set of axioms is the empty set.

DEFINITION 14. The MR system MPK is defined as follows:

1. $I$ is the set of positive integers (with 0), and $i \prec i+1$;
2. $L_0 = L$, and $L_{i+1}$ is the propositional language containing the atomic formulas $T(``A")$, with $A$ any $L_i$-wff;
3. $\Omega_i = \emptyset$;

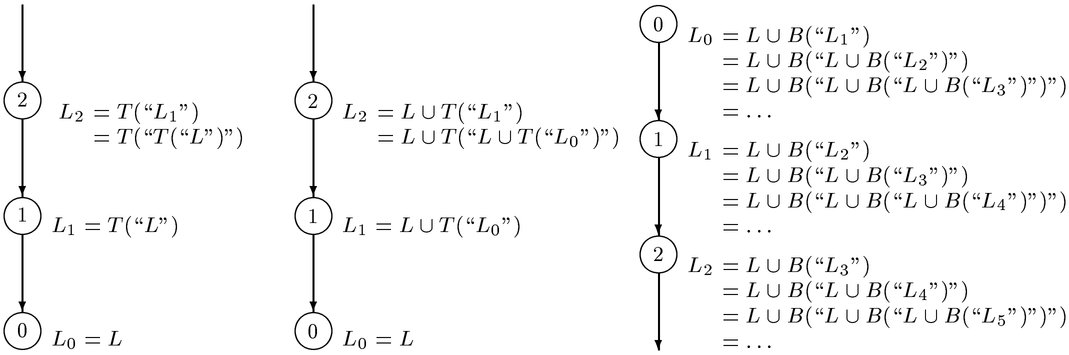


*Figure 2.* Structures and languages of MPK, MK, and MBK.

4. $\Delta$ contains the ML version of the propositional classical ND rules in in $L_i$, $R_{\mathrm{dn}.i}$ from $i+1$ to $i$, and restricted $R_{\mathrm{up}.i}$ from $i$ to $i+1$.

MK is the MR system studied in detail in this paper. Analogously to MPK, MK is an ML system for the formalization of metatheoretic reasoning (Giunchiglia and Serafini, 1994; Giunchiglia and Traverso, 1996). Indeed MK is defined as MPK except for the fact that the language of each metatheory contains the language of the object theory. Hence in MK, if the language of the theory $i$-th is $L$, then the language of its metatheory theory is $L \cup T(\text{“}L\text{”})$ (see Figure 2).

DEFINITION 15. *MK* is defined by replacing clause 2 of Definition 14 of MPK with

2. $L_0 = L$, and $L_{i+1}$ is the propositional language containing $L_i$ and the atomic formulas $T(\text{“}A\text{”})$ with $A$ any $L_i$-wff.

In MK the predicate “$T$” has the same properties as the operator $\Box$ in modal $K$ (see below for more details). Certain results, like the equivalence with modal logics, are stated only under this hypothesis (see (Giunchiglia and Serafini, 1994)). (Giunchiglia et al., 1992) presents a version of MK with $L_i \not\subseteq \mathrm{L}_{i+1}$.

MBK is the basic ML system for the representation of the beliefs of a single agent. (Giunchiglia and Serafini, 1994; Giunchiglia et al., 1993; Giunchiglia and Giunchiglia, 2000; Giunchiglia and Ghidini, 1998). The idea underlying MBK is that there is an agent $a$ who has beliefs about the world and beliefs about its own beliefs. Given a proposition $A$ about the state of the world, $B(\text{“}A\text{”})$ means that $A$ itself is believed by $a$ or, in other words, that $A$ holds in $a$'s view of the world. Similarly $B(\text{“}B(\text{“}A\text{”})\text{”})$ means that $B(\text{“}A\text{”})$ is believed by $a$, namely that $A$ holds in $a$'s view of its beliefs about the world, and so on. In MBK each view is modeled as a theory with its own language. The basic structure of MBK is reported in Figure 2.

DEFINITION 16. The MR system MBK is defined as follows:

1. $I$ is the set of positive integers (with 0), and $i+1 \prec i$;
2. each $L_i$ is the propositional languge containing $L$ and the atomic formulas $B(\text{“}A\text{”})$ with $A$ any $L_{i+1}$-wff;
3. $\Omega_i = \emptyset$;
4. $\Delta$ contains the ML version of the propositional classical ND rules in $L_i$, $R_{\mathrm{dn}.i+1}$ from $i$ to $i+1$, and restricted $R_{\mathrm{up}.i+1}$ from $i+1$ to $i$.

Analogously to what happens for MK, in MBK the predicate “$B$” has the same properties as $\Box$ in modal K.

MB4, MB5, MB45 are extensions of MBK, where the predicate $B$ has the same properties of $\Box$ in the modal systems K4, K5, and K45 respectively (Giunchiglia and Serafini, 1994).

DEFINITION 17. MB4, MB5, and MB45 are defined by extending clauses 1 and 4 of Definition 16 of MBK as follows:

1. for MB4, MB5, MB45 add $i+2 \prec i$;

4 for MB4 add restricted $R_{\mathrm{up}.i+2}$ from $i+2$ to $i$;

4 for MB5 add $R_{\mathrm{dn}.i+2}$ from $i$ to $i+2$;

4 for MB45 add $R_{\mathrm{dn}.i+2}$ from $i$ to $i+2$ and restricted $R_{\mathrm{up}.i+2}$ from $i+2$ to $i$.

MBK(n) is the basic MR system for the representation of multiagent beliefs (Giunchiglia et al., 1993; Giunchiglia and Serafini, 1994; Cimatti and Serafini, 1996; Benerecetti et al., 1999). The idea underlying the formalization of multiagent belief is that there is a set of agents $\mathcal{A} = \{a_1, a_2 \ldots a_n\}$, who have beliefs about the world, beliefs about their own beliefs, and beliefs about the other agents' beliefs. The basic structure of MBK(n) is reported in Figure 3.

DEFINITION 18. Let $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be a set of $n$ symbols and $\mathcal{A}^*$ the set of finite (possible empty) sequences of symbols in $\mathcal{A}$. Let $\epsilon \in \mathcal{A}^*$ denote the empty sequence. The MR system MBK(n) on the propositional language $L$ is defined as follows:

1. $I$ is the set $\mathcal{A}^*$, and $\overline{a}a \prec \overline{a}$, for each $\overline{a} \in \mathcal{A}^*$ and $a \in \mathcal{A}$;
2. $L_{\overline{a}}$ is the propositional language containing $L$ and the atomic formulas $B_a(\text{“}A\text{”})$ with $A$ any $L_{\overline{a}a}$-wff;
3. $\Omega_{\overline{a}} = \emptyset$;
4. $\Delta$ contains the ML version of the propositional classical ND rules in $L_{\overline{a}}$, $R_{\mathrm{dn}.\overline{a}a}$, from $\overline{a}$ to $\overline{a}a$ and restricted $R_{\mathrm{up}.\overline{a}a}$ from $\overline{a}a$ to $\overline{a}$.

MBK(n) formalizes situations where the state of the world does not change, and where the agents don't change their beliefs about such a state. A logic which deals with dynamic world and dynamic beliefs about the world is MATL (Multi Agent Temporal Logic), introduced in (Benerecetti et al., 1998b). MATL allows us to model situations where the state of the world changes, and where agents can change their beliefs (but also other propositional attitudes, such as desire and intentions) about the evolution of the world. Restricting to a single propositional attitude (e.g., belief), MATL has the same structure as MBK(n) (see Figure 3) with the only difference that the language of each view $\overline{a} \in \mathcal{A}^*$ is the temporal language CTL (Clarke et al., 1994).

DEFINITION 19. *MATL* is defined by replacing clauses 2, 3, and 4 of Definition 18 of MBK(n) with the following clauses:

2. $L_{\overline{a}}$ is the CTL language containing $L$ and the atomic formulas $B_a(\text{“}A\text{”})$ with $A$ any $L_{\overline{a}a}$-wff;
3. $\Omega_{\overline{a}}$ is the set of CTL axioms for $L_{\overline{a}}$;
4. $\Delta$ contains the ML version of a propositional ND rules in $L_{\overline{a}}$, $R_{\mathrm{dn}.\overline{a}a}$, from $\overline{a}$ to $\overline{a}a$ and restricted $R_{\mathrm{up}.\overline{a}a}$ from $\overline{a}a$ to $\overline{a}$.

### 3.3. The MR system MK

MK, as defined in Definition 15, is constituted of a hierarchy of theories labelled with natural numbers. In MK we replace the formulas $\bullet_i(\text{“}A\text{”})$ with $T(\text{“}A\text{”})$, to emphasize the fact that $T(\text{“}A\text{”})$ should be read “$A$ is a theorem”. Notice that the reference to the theory in which $A$ is a theorem is not specified in the formula $T(\text{“}A\text{”})$. Indeed, the intuitive meaning of $T(\text{“}A\text{”})$ depends on the context where it is stated. If $T(\text{“}A\text{”})$ is stated in 1 (i.e. $1 : T(\text{“}A\text{”})$, then it should be read as “$A$ is theorem of 0”, while, if it is stated in 4 (i.e. $4 : T(\text{“}A\text{”})$), it should be read as “$A$ is a theorem of 3”. In general the predicate $T$ refers to theoremhood in the theory below. The set of inference rules of MK, reported in

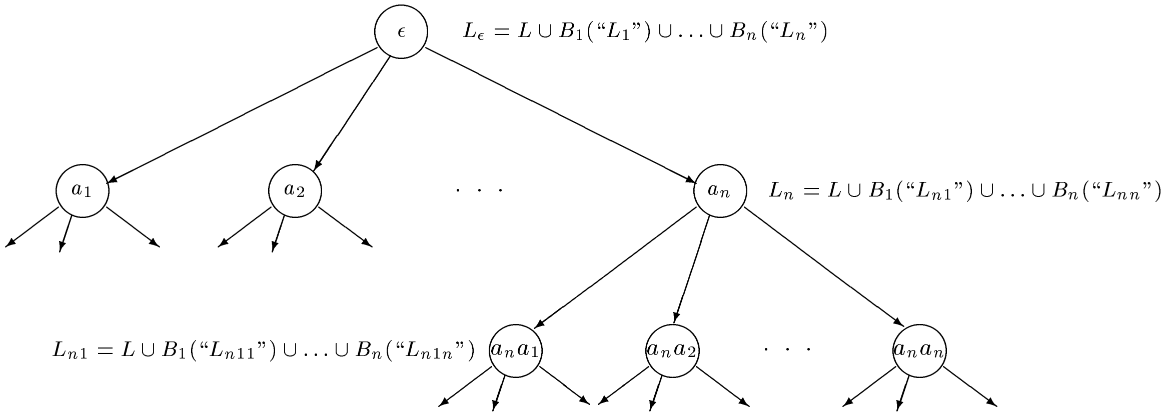


*Figure 3.* Structure and languages of MBK(n)

Figure 4, contains the ML version in $L_i$ of the classical propositional ND inference rules, and the bridge rules $R_{\mathrm{dn}.i}$ from $i+1$ to $i$, and restricted $R_{\mathrm{up}.i}$ from $i$ to $i+1$.

There are several reasons why we choose MK as a case study. First, it is simple and has a clear interpretation in terms of metatheoretic concepts (see (Giunchiglia et al., 1992)). Its simple structure allows us to minimize the technical overhead in the development of the proof theory. A second reason the structure of MK is easily mappable in other MR systems. For instance MPK can be seen as MK with restrictions on the languages; MBK can be seen as an upside-down version of MK; MBK(n) can be seen as an infinite set of upside-down versions of MK, corresponding to the infinite branches of the tree shown in Figure 3. MB4, MB5, and MB45, can be seen as an upside-down version of MK with some extra relation between theories $i$ and $i+2$. Finally MATL is an upside-down version of MK, where each theory is CTL rather than propositional. A third reason is that normal modal logics have been proved to be embeddable in MK and its extensions (see (Giunchiglia and Serafini, 1994)). A proof theory for MK is therefore relevant also to modal logics, and in particular, to modal K. We will show, indeed, that MK enjoys important properties such as the subformula property, which do not hold in most other calculi for modal logics. These results are reported in Section 4. We give below some preliminary results which give the “feeling” on how MK behaves.

$$\frac{\begin{array}{c}[i : A]\\ i : B\end{array}}{i : A \supset B}\ \supset\mathrm{I}_i \qquad \frac{i : A \quad i : A \supset B}{i : B}\ \supset\mathrm{E}_i$$

$$\frac{i : A \quad i : B}{i : A \wedge B}\ \wedge\mathrm{I}_i \qquad \frac{i : A \wedge B}{i : A}\ \frac{i : A \wedge B}{i : B}\ \wedge\mathrm{E}_i$$

$$\frac{i : A}{i : A \vee B}\ \frac{i : B}{i : A \vee B}\ \vee\mathrm{I}_i \qquad \frac{i : A \vee B \quad \begin{array}{c}[i : A]\\ i : C\end{array} \quad \begin{array}{c}[i : B]\\ i : C\end{array}}{i : C}\ \vee\mathrm{E}_i$$

$$\frac{\begin{array}{c}[i : \neg A]\\ i : \bot\end{array}}{i : A}\ \bot_i$$

$$\frac{i : A}{i+1 : T(\text{“}A\text{”})}\ R_{\mathrm{up}.i} \qquad \frac{i+1 : T(\text{“}A\text{”})}{i : A}\ R_{\mathrm{dn}.i}$$

RESTRICTION: $\bot_i$ can be applied if $A$ is not of the form $\bot$. $R_{\mathrm{up}.i}$ can be applied only if $i : A$ does not depend from any assumptions with index equal to $i$.

*Figure 4.* Inference rules of MK

Concerning the relation between MK and modal K, we have that, for any $L_i$ formula $A$:

$$\vdash_{\mathrm{MK}} i+1 : T(\text{“}A \supset B\text{”}) \supset (T(\text{“}A\text{”}) \supset T(\text{“}B\text{”})) \tag{5}$$

$$\vdash_{\mathrm{MK}} i : A \ \text{ implies } \ \vdash_{\mathrm{MK}} i+1 : T(\text{“}A\text{”}) \tag{6}$$

Notice that (5) is the MK version of the modal axiom K ($\Box(A \supset B) \supset \Box A \supset \Box B$), and (6) corresponds to the necessitation rule ($\vdash A \Rightarrow \vdash \Box A$). The proofs of other theorems of MK corresponding to K-valid formulas are reported in Appendix B. An example of a proof of (5) in MK is the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A\text{”})}{i : A}\ R_{\mathrm{dn}.i} \quad \dfrac{i+1 : T(\text{“}A \supset B\text{”})}{i : A \supset B}\ R_{\mathrm{dn}.i}}{i : B}\ \supset\mathrm{E}_i}{i+1 : T(\text{“}B\text{”})}\ R_{\mathrm{up}.i}}{i+1 : T(\text{“}A\text{”}) \supset T(\text{“}B\text{”})}\ \supset\mathrm{I}_{i+1}}{i+1 : T(\text{“}A \supset B\text{”}) \supset (T(\text{“}A\text{”}) \supset T(\text{“}B\text{”}))}\ \supset\mathrm{I}_{i+1} \tag{7}$$

For a detailed discussion on the relation among MK and modal logics see (Giunchiglia and Serafini, 1994). Here we report the main correspondence theorem, as proved in (Giunchiglia and Serafini, 1994).

THEOREM 20. *$A_1, \ldots A_n \vdash_K A$ if and only if there is an $i \geq 0$ such that $i : A_1^+, \ldots, i : A_n^+ \vdash_{MK} i : A^+$, where $(.)^+$ is a mapping from modal formulas to formulas of MK languages, that maps $\Box A$ into $T(\text{“}A\text{”})$ and distributes over connectives.*

One of the most important aspects of ML systems concerns how theories affect one another, or, in other words, how the truth/derivability of a set of formulas in a language affect the truth/derivability of formulas in another language. This relation tightly depends from the form of the bridge rules. In the case of MK, for instance, despite the fact that there are bridge rules from $i$ to $i+1$, theory $i$ does not affect theory $i+1$. Indeed, in MK upper theories affect lower theories but not vice-versa. This is formalized by the following theorem.

THEOREM 21. *$\Gamma \vdash_{MK} i : A$ if and only if $\Gamma_{\geq i} \vdash_{MK} i : A$, where $\Gamma_{\geq i} = \{j : B \in \Gamma \mid j \geq i\}$.*

*Proof.* The “if” direction is trivial as $\Gamma_{\geq i} \subseteq \Gamma$. The “only if” direction can be proved by induction on the complexity of the deductions in MK. The base case and the induction step for $i$-rules and $R_{\mathrm{dn}.i}$ are trivial. If a deduction of $i+1 : T(\text{“}A\text{”})$ from $\Gamma$ ends with an application of $R_{\mathrm{up}.i}$ to $i : A$, then the restriction on the applicability of $R_{\mathrm{up}.i}$ implies that the conclusion $i+1 : T(\text{“}A\text{”})$, depends only from assumptions greater than or equal to $i+1$, and therefore contained in $\Gamma_{\geq i+1}$. $\Box$

The intuitive interpretation of Theorem 21 is that an assumption at one level doesn't have any effects on the upper levels. We call $\Gamma_{\geq i}$ the subset of *effective assumptions* at level $i$ of $\Gamma$.

A second point that has to be clarified concerns lemma composition, namely the possibility of combining subdeductions in order to obtain other deductions. We need the following notation. If $\Pi$ is a deduction of $i : A$ and $\Pi'$ is another deduction then,

$$\begin{array}{c} \Pi \\ (i : A) \\ \Pi' \end{array} \tag{8}$$

is $\Pi'$ itself if $\Pi'$ does not contain any undischarged assumptions of the form $i : A$, or it is the formula tree obtained by writing $\Pi$, without its bottom formula $i : A$, above *all* the undischarged assumptions of $\Pi'$ of the form $i : A$.

LEMMA 22 (Composing deductions). *If $\Pi$ is a deduction of $i : A$ depending on $\Gamma$ and $\Pi'$ is a deduction of $j : B$ depending on $\Sigma$, then (8) is a deduction of $j : B$ from $\Gamma \cup (\Sigma \setminus \{i : A\})$.*

*Proof.* Notice that the operation described in (8) replaces all the undischarged assumptions of the form $i : A$ with a new set of assumptions $\Gamma$, from which $i : A$ is derivable. To prove Lemma 22 it is enough to show that (8) is a deduction in MK, namely that any application of a restricted rule (i.e., $R_{\mathrm{up.}}$) in $\Pi'$ is still possible in (8). Therefore, we show that, if an occurrence $k : C$ in $\Pi'$ depends on a set of assumptions with index greater than or equal to $h$, then the same occurrence in (8) depends on a set of assumptions with index greater than or equal to $h$. This holds because, if $k : C$ does not depend from $i : A$ in $\Pi'$, then $k : C$ in (8) depends on the same set of assumptions as in $\Pi'$; otherwise, if $k : C$ depends from $i : A$ in $\Pi'$, then, by Theorem 21, $i \geq h$, and, since $i : A$ depends on $\Gamma$ in $\Pi$, by the same theorem, the indexes of the formulas in $\Gamma$ are greater than or equal to $i \geq h$. This implies that $k : C$ depends on a set of assumptions with indexes greater than or equal to $h$. □

THEOREM 23. *If $\Gamma \vdash_{MK} i : A$, and $i : A, \Sigma \vdash_{MK} j : B$, then $\Gamma, \Sigma \vdash_{MK} j : B$.*

*Proof.* Let $\Pi$ be a deduction of $i : A$ depending on $\Gamma' \subseteq \Gamma$, and $\Pi'$ a deduction of $j : B$ depending on $\Sigma' \subseteq \Sigma \cup \{i : A\}$. By Lemma 22, (8) is a deduction of $j : B$ from $\Gamma' \cup \Sigma' \setminus \{i : A\}$ which is contained in $\Gamma \cup \Sigma$. □

MR–CUT is stronger than ML–CUT(i) (Definition 1) as in ML–CUT(i) the index of the conclusion $(j : B)$ must be the same as that of the "cut" formula $(i : A)$. In MR–CUT, instead, this is not necessary. A consequence of MR–CUT is the *deduction theorem.*

THEOREM 24. *$\Gamma, i : A \vdash_{MK} i : B$ if and only if $\Gamma \vdash_{MK} i : A \supset B$.*

*Proof.* If $\Pi$ is a deduction of $i : B$ from $\Gamma, i : A$, then

$$\dfrac{\begin{array}{c}\Gamma, i : A \\ \Pi \\ i : B\end{array}}{i : A \supset B} \supset\mathrm{I}_i$$

is a deduction of $i : A \supset B$ from $\Gamma$. Vice-versa, since $i : A, i : A \supset B \vdash_{\mathrm{MK}} i : B$, if $\Gamma \vdash_{\mathrm{MK}} i : A \supset B$, by (MR–CUT) we have $\Gamma, i : A \vdash_{\mathrm{MK}} i : B$. □

Notice that Theorem 24 holds only when the assumption and the conclusion of $\Pi$ belong to the same language, (in this case $L_i$). Deduction theorem cannot be applied when $\Gamma, i : A \vdash_{\mathrm{MK}} j : B$, if $i \neq j$, as there is no language where to express the implication of $A$ and $B$. A consequence of this fact is that derivability in MK cannot be reduced to provability.

## 4. Normal Forms for Deductions in MK

The goal of this section is to study the structure of deductions in MK. First, we characterize the subset of indexes which must be used in order to prove a theorem. This leads to the definition of the *weak normal form*. Deductions in weak normal form enjoy the *sublevel property*, which states that, to prove a theorem with index $i$, it is enough to reason in the space of the formulas with index $\leq i$. A consequence of the sublevel property is the consistency of MK. Subsequently we define the *normal form* for MK deductions, which in turn, allows us to provide the *subformula property*. Roughly speaking, the subformula property guarantees that, to prove $i : A$, it is enough to reason in the finite space of subformulas of $i : A$ (where the notion of subformulas for labelled formulas has to be defined properly).

### 4.1. Weak Normal Deductions and Sublevel Property

In MK there are two main reasoning patterns that can be combined to prove a theorem in $i$. The former starts in $i$, switches down, by $R_{\text{dn}.i-1}$ to $i-1$, reasons in the space of formulas with index lower than $i$, and finally switches back, by $R_{\text{up}.i-1}$, to $i$. An example of this reasoning pattern is proof of $i+1 : T(\text{“}A \supset B\text{”}) \supset T(\text{“}A\text{”}) \supset T(\text{“}B\text{”})$ (deduction (7)). The latter runs in the opposite direction: it starts in $i$, it switches up, by $R_{\text{up}.i}$, to $i+1$, it reasons in the space of the formulas with index greater than $i$, and finally it switches back, by $R_{\text{dn}.i}$, to $i$. However notice that, the application of $R_{\text{up}.i}$ transforms a complex formula $A$ into an atomic formula $T(\text{“}A\text{”})$ in $i+i$. From the fact that the local inference rules in $i$ and $i+1$ are equal, and from the fact that in the $i+1$-th theory there are no specific axioms for $T(\text{“}A\text{”})$, the same reasoning performed on $T(\text{“}A\text{”})$ in $i+1$, can be performed on $A$ in $i$. So the reasoning pattern that first switches up and then down does not add any new theorems to the $i$-th theory. This intuition is proved by showing that each deduction can be reduced to a *weak normal form*, which do not contain any instances of such a redundant reasoning pattern.

DEFINITION 25. If $\Pi$ is a deduction of $i : A$ from $\Gamma_{\leq j}$ then an occurrence $k : B$ in $\Pi$ is an *overflowing formula* of $\Pi$, if and only if

$i, j < k$. $\Pi$ is a *weak normal deduction* (is in *weak normal form*) if and only if it does not contain any overflowing formula.

EXAMPLE 26. Consider the following deduction of $0 : q$ from $1 : T(\text{“}p \supset p\text{”}) \supset T(\text{“}q\text{”})$,

$$\dfrac{\dfrac{\dfrac{{}^{(\bullet)}\dfrac{\dfrac{\dfrac{0 : p}{0 : p \supset p}\supset\mathrm{I}_0}{1 : T(\text{“}p \supset p\text{”})} R_{\mathrm{up}.0}}{2 : T(\text{“}T(\text{“}p \supset p\text{”})\text{”})} R_{\mathrm{up}.1}}{1 : T(\text{“}p \supset p\text{”})} R_{\mathrm{dn}.1} \quad 1 : T(\text{“}p \supset p\text{”}) \supset T(\text{“}q\text{”})}{1 : T(\text{“}q\text{”})} \supset\mathrm{E}_1}{0 : q} R_{\mathrm{dn}.0} \tag{9}$$

The occurrence labelled with $^{(\bullet)}$ is an overflowing formula of deduction (9).

Overflowing formulas correspond to redundant reasoning steps. To remove overflowing formulas from a deduction $\Pi$, we define the operator $\overline{\Pi}^i$, which pushes down to $i$ all the occurrences of $\Pi$ with index greater than $i$. This operator is based on a simpler operator $(.)^{-1}$ on wffs. Intuitively $(.)^{-1}$ transforms an $L_{i+1}$-wff into an $L_i$-wff by removing, if any, the “most external” occurrences of the predicate $T$.

DEFINITION 27. The operator $(.)^{-1}$ is defined as follows:

1. $\bot^{-1} = \bot$;
2. $p^{-1} = p$, if $p$ is a propositional constant;
3. $T(\text{“}A\text{”})^{-1} = A$;
4. $(.)^{-1}$ distributes over connectives.

Furthermore, for each $n \geq 1$, $A^{-n}$ is the result of $n$ applications of $(.)^{-1}$ to $A$.

EXAMPLE 28. Some examples of applications of $(.)^{-n}$:

$$\begin{aligned}
(p)^{-1} &= p \\
(T(\text{“}p \wedge q\text{”}) \vee r)^{-1} &= (p \wedge q) \vee r \\
(p \wedge T(\text{“}q\text{”}) \wedge T(\text{“}T(\text{“}r\text{”})\text{”}))^{-1} &= p \wedge q \wedge T(\text{“}r\text{”}) \\
(p \wedge T(\text{“}q\text{”}) \wedge T(\text{“}T(\text{“}r\text{”})\text{”}))^{-2} &= p \wedge q \wedge r
\end{aligned}$$

Note that, if $A$ is an $L_i$-wff, then $A^{-n}$ is an $L_{i-n}$-wff if and $n < i$, and an $L_0$-wff otherwise.

DEFINITION 29. For any natural number $i_0$, $\overline{i : A}^{i_0}$ is $i_0 : A^{(i_0-i)}$ if $i > i_0$ and $i : A$ otherwise. If $\Gamma$ is a set of wffs, $\overline{\Gamma}^{i_0} = \{\overline{i : A}^{i_0} : i : A \in \Gamma\}$. If $\Pi$ is a deduction, then $\overline{\Pi}^{i_0}$ is the formula tree obtained by substituting every occurrence $i : A$ in $\Pi$ with $\overline{i : A}^{i_0}$ and by removing all the consequences of $R_{\mathrm{up}.i}$ and $R_{\mathrm{dn}.i+1}$, with $i \geq i_0$.

EXAMPLE 30. Let $\Pi$ be deduction (9). $\overline{\Pi}^1$ is the deduction

$$\dfrac{\dfrac{\dfrac{0 : p}{0 : p \supset p}\supset\mathrm{I}_0}{1 : T(\text{“}p \supset p\text{”})}R_{\mathrm{up}.0} \qquad 1 : T(\text{“}p \supset p\text{”}) \supset T(\text{“}q\text{”})}{\dfrac{1 : T(\text{“}q\text{”})}{0 : q}R_{\mathrm{dn}.0}}\supset\mathrm{E}_1 \tag{10}$$

of $0 : q$ from $1 : T(\text{“}p \supset p\text{”}) \supset T(\text{“}q\text{”})$. Notice that in (10) the overflowing formula of (9) has been removed without changing the assumptions and the conclusions of (9).

LEMMA 31. *If $\Pi$ is a deduction of $i : A$ from $\Gamma$, then $\overline{\Pi}^{i_0}$ is a deduction of $\overline{i : A}^{i_0}$ from $\overline{\Gamma}^{i_0}$.*

*Proof.* We prove by induction that, if $\Pi$ is a deduction of $i : A$ depending on $\Gamma$, then $\overline{\Pi}^{i_0}$ is a deduction of $\overline{i : A}^{i_0}$ from $\overline{\Gamma}^{i_0}$.

(Base Case) If $\Pi$ is $i : A$ then $\overline{i : A}^{i_0}$ is a deduction of $\overline{i : A}^{i_0}$ from $\overline{i : A}^{i_0}$.

(Step Case) If $\Pi$ ends with an application of an $i$-rule, then the theorem trivially follows from the distributivity of $\overline{(.)}^{i_0}$ and from the fact that the $i$-rules are the same in all levels.

If it ends with an application of a $R_{\mathrm{up}.i}$, then $\Pi$ is of the following form:

$$\dfrac{\begin{array}{c}\Gamma\\ \Pi'\\ i : A\end{array}}{i+1 : T(\text{“}A\text{”})}R_{\mathrm{up}.i}$$

By induction, $\overline{\Pi'}^{i_0}$ is a deduction of $\overline{i : A}^{i_0}$ from $\overline{\Gamma}^{i_0}$. We have two cases: $i < i_0$ and $i \geq i_0$.

If $i < i_0$, then $\overline{i : A}^{i_0}$ is $i : A$ and, by induction, $\overline{\Pi}^{i_0}$ is

$$\dfrac{\begin{array}{c}\overline{\Gamma}^{i_0}\\ \overline{\Pi'}^{i_0}\\ i : A\end{array}}{i+1 : T(\text{“}A\text{”})}R_{\mathrm{up}.i}$$

Then, since $R_{\mathrm{up}.i}$ is applicable in $\Pi$, the indexes of the elements of $\overline{\Gamma}^{i_0}$ are greater than $i$ and $i < i_0$. Hence $R_{\mathrm{up}.i}$ is also applicable in $\overline{\Pi}^{i_0}$. This implies that $\overline{\Pi}^{i_0}$ is a deduction of $\overline{i+1 : T(\text{“}A\text{”})}^{i_0} = i+1 : T(\text{“}A\text{”})$ from $\overline{\Gamma}^{i_0}$.

If $i \geq i_0$, then $\overline{\Pi}^{i_0}$ is $\overline{\Pi'}^{i_0}$, that is a deduction of $\overline{i+1 : T(\text{“}A\text{”})}^{i_0} = \overline{i : A}^{i_0}$ from $\overline{\Gamma}^{i_0}$.

If it ends with an application of a $R_{\text{dn}.i}$, then $\Pi$ is of the form:

$$\dfrac{\begin{array}{c}\Gamma\\ \Pi'\\ i+1 : T(\text{“}A\text{”})\end{array}}{i : A}\ R_{\text{dn}.i}$$

By induction $\overline{\Pi'}^{i_0}$ is a deduction of $\overline{i+1 : T(\text{“}A\text{”})}^{i_0}$ from $\overline{\Gamma}^{i_0}$. As for $R_{\text{up}.i}$, we have two cases: $i < i_0$ and $i \geq i_0$.

If $i < i_0$ then $\overline{\Pi}^{i_0}$ is:

$$\dfrac{\begin{array}{c}\overline{\Gamma}^{i_0}\\ \overline{\Pi'}^{i_0}\\ i+1 : T(\text{“}A\text{”})\end{array}}{i : A}\ R_{\text{dn}.i}$$

which is a deduction of $\overline{i : A}^{i_0} = i : A$ from $\overline{\Gamma}^{i_0}$.

If $i \geq i_0$, then $\overline{\Pi}^{i_0}$ is $\overline{\Pi'}^{i_0}$, which is a deduction of $\overline{i : A}^{i_0} = \overline{i+1 : T(\text{“}A\text{”})}^{i_0}$ from $\overline{\Gamma}^{i_0}$. □

THEOREM 32. *If* $\Gamma \vdash_{MK} i : A$, *then there is a weak normal deduction of* $i : A$ *from* $\Gamma$.

*Proof.* Let $\Pi$ be a deduction of $i : A$ depending on $\Gamma' \subseteq \Gamma$ and $i_0$ the maximum index of $\Gamma' \cup \{i : A\}$. By Lemma 31, $\overline{\Pi}^{i_0}$ is a deduction of $\overline{i : A}^{i_0}$ from $\overline{\Gamma'}^{i_0}$. Since $i_0$ is the greatest index of the elements of $\Gamma' \cup \{i : A\}$, we have $\overline{\Gamma'}^{i_0} = \Gamma'$ and $\overline{i : A}^{i_0} = i : A$. Furthermore, $\overline{\Pi}^{i_0}$ does not contain any overflowing formula, and it is a weak normal deduction of $i : A$ from $\Gamma$. □

COROLLARY 33 (Consistency). *For any* $i \geq 0$, $\nvdash_{MK} i : \bot$.

*Proof.* By contradiction. Let $\Pi$ be a proof of $i : \bot$. By Lemma 31, $\overline{\Pi}^0$ is a proof of $0 : \bot$. Since it does not contain any occurrence with index greater than 0, $\overline{\Pi}^0$ does not contain any application of reflection rules. Thus $\overline{\Pi}^0$ is a proof of $\bot$ in Classical Propositional Logic, which does not exists. □

Not only is any theory of MK consistent, but inconsistency does not propagate upwards. This corresponds to the fact that a metatheory can consistently speak about an inconsistent object theory. Notice that this is an important feature which is not supported by logical systems where object theory and metatheory are amalgamated in a unique theory (see for instance (Bowen and Kowalski, 1982)).

COROLLARY 34. *For any* $i \geq 0$, $i : \bot \nvdash_{MK} i+1 : \bot$.

### 4.2. Normal Deductions

The normal form for deductions in MK is a natural generalization of the normal form for propositional classical ND as defined in (Prawitz, 1965). Before defining normal form for MK, we first extend the terminology to deal with the reflection rules.

By I-rule we mean one of $\supset$I, $\wedge$I, $\vee$I and $R_{\text{up.}}$; by E-rule we mean one of $\supset$E, $\wedge$E, $\vee$E and $R_{\text{dn.}}$. The reflection rules $R_{\text{up.}}$ and $R_{\text{dn.}}$ are seen as the introduction of $T$ and the elimination of $T$, respectively. Furthermore, we say that an occurrence $A$ is a premise or the consequence of a rule when $A$ is a premise or the consequence of an application of that rule.

DEFINITION 35. In an application of an $\supset\text{E}_i$, $i : A \supset B$ and $i : A$ are called *major premise* and *minor premise*, respectively. In an application of an $\vee\text{E}_i$, $i : A \vee B$ and the $i : C$'s are called *major premise* and *minor premises*, respectively. The premises of the application of any other rule are major premises.

DEFINITION 36. An occurrence $i : A$ in a deduction is a *maximum formula* if and only if it satisfies one of the two conditions:

1. $i : A$ is the consequence of an I-rule and the major premise of an E-rule;
2. $i : A$ is the consequence of a $\bot$-rule and the major premise of an E-rule other than $R_{\text{dn}.i-1}$.

A deduction is in *normal form* (or is *normal*) if and only if it does not contain any maximum formula.

EXAMPLE 37. The following deductions are not in normal form.

$$\underset{(a)}{\dfrac{(\bullet)\dfrac{i : A \quad i : B}{i : A \wedge B}\,\wedge\text{I}_i}{i : A}\,\wedge\text{E}_i} \qquad \underset{(b)}{\dfrac{(\bullet)\dfrac{\dfrac{i : A}{i : A \supset A}\,\supset\text{I}_i}{i+1 : T(\text{“}A \supset A\text{”})}\,R_{\text{up}.i}}{i : A \supset A}\,R_{\text{dn}.i}} \qquad \underset{(c)}{\dfrac{(\bullet)\dfrac{i : \bot}{i : A \wedge B}\,\bot_i}{i : A}\,\wedge\text{E}_i} \qquad (11)$$

The occurrences labelled with $^{(\bullet)}$ are maximum formulas. As overflowing formulas, maximum formulas correspond to redundant reasoning steps. These steps can be removed by suitable transformation. As an example, the non normal deductions (11.a–c) can be reduced to the following equivalent deductions:

$$\underset{(a)}{i : A} \qquad\qquad \underset{(b)}{\dfrac{i : A}{i : A \supset A}\ \supset\mathrm{I}_i} \qquad\qquad \underset{(c)}{\dfrac{i : \bot}{i : A}\ \bot_i} \tag{12}$$

The proof of the existence of a normal deduction is based on a procedure that removes maximum formulas from deductions. Lemma 38 and Lemma 39 describe how to remove maximum formulas satisfying point 1 and 2 in Definition 36, respectively.

LEMMA 38. *If* $\Gamma \vdash_{MK} i : A$*, then there exists a deduction of* $i : A$ *from* $\Gamma$ *with no occurrence which is both the consequence of an I-rule and the major premise of an E-rule.*

*Proof.* Let $\Pi$ be a deduction of $i : A$ from $\Gamma$. The occurrences which are consequences of an I-rule and major premises of an E-rule different from reflection rules, can be removed from $\Pi$ by applying the reduction steps described in (Prawitz, 1965), and reported in Appendix C. The reduction step for $R_{\mathrm{up}.j}$ and $R_{\mathrm{dn}.j}$, called *T-reduction*, is the following:

$$\begin{array}{c} \Pi_1 \\ \dfrac{\dfrac{j : C}{j+1 : T(\text{“}C\text{”})}\ R_{\mathrm{up}.j}}{j : C}\ R_{\mathrm{dn}.j} \\ \Pi_2 \end{array} \qquad\qquad \begin{array}{c} \Pi_1 \\ j : C \\ \Pi_2 \end{array} \tag{13}$$

The reduction steps defined in Appendix C and (13) are applied to those maximum formulas which have the greatest complexity, where the complexity of a formula is the number of nested connectives and $T$ predicates. It is easy to show that repeated applications of such reduction steps converge to a deduction without occurrences which are consequences of an I-rule and maximum premises of an E-rule. The convergence is guaranteed by the fact that the application of a reduction step to a maximum formula which has the greatest complexity removes it, without introducing other maximum formulas of the same or greater complexity. □

By Lemma 38, any deduction in MK can be reduced to a deduction where the consequence of any I-rule is not the maximum premise of an E-rule. To obtain a normal deduction we have to remove also the consequences of the $\bot$-rule which are major premises of an E-rule. Following what originally done in (Prawitz, 1965), to simplify the treatment, we consider the ML system MK$'$ obtained from MK by excluding the rules for $\vee$. MK$'$ is equivalent to MK once we uniformly rewrite $A \vee B$ as $\neg A \supset B$. Deductions in MK$'$ can be reduced in a form where the consequences of the $\bot$-rule are atomic. This implies that they cannot be the premise of E-rules with the exception of $R_{\mathrm{dn}.}$.

LEMMA 39. *If $\Gamma \vdash_{MK'} i : A$, then there is a deduction where the consequences of all the applications of a $\bot$-rule are atomic.*

Lemma 39 can be proved by applying the reduction step "$\frac{\bot}{\wedge}$-reduction" described in (Prawitz, 1965) and reported in Appendix C.

THEOREM 40. *If $\Gamma \vdash_{MK'} i : A$, then there is a normal deduction of $i : A$ from $\Gamma$ in MK$'$.*

*Proof.* By Lemma 39, there is a deduction $\Pi$ of $i : A$ from $\Gamma$ where all the consequences of a $\bot$-rule are atomic. By Lemma 38, $\Pi$ can be reduced to a deduction $\Pi'$, where any consequence of an I-rule is not the major premise of an E-rule. Since the reduction steps defined in the proof of Lemma 38 only introduce atomic consequences of $\bot$-rules, we can conclude that $\Pi'$ is normal. □

## 4.3. The Form of Normal Deductions

In this section we study the form of normal deductions in MK$'$. This study is necessary in order to prove the subformula property.

DEFINITION 41. Let $\beta = i_1 : A_1, \ldots, i_n : A_n$ be the initial part of a thread $\tau$ in a deduction $\Pi$. We say that $\beta$ is a *branch* of $\Pi$ if and only if it satisfies one of the following two conditions:

1. $i_n : A_n$ is the first formula occurrence of $\tau$ that is the minor premise of an application of $\supset\mathrm{E}_{i_n}$;
2. $i_n : A_n$ is the last formula occurrence of $\tau$ and no minor premise of $\supset\mathrm{E}$ occurs in $\beta$.

A branch which satisfies condition 2 is called *main branch*.

EXAMPLE 42. The branches of the deduction:

$$
\begin{prooftree}
\AxiomC{$i+1 : T(``A \supset C")$}
\RightLabel{$R_{\mathrm{dn}.i}$}
\UnaryInfC{$i : A \supset C$}
\AxiomC{$i : A \wedge B$}
\RightLabel{$\wedge\mathrm{E}_i$}
\UnaryInfC{$i : A$}
\RightLabel{$\supset\mathrm{E}_i$}
\BinaryInfC{$i : C$}
\AxiomC{$i : \neg C$}
\RightLabel{$\supset\mathrm{E}_i$}
\BinaryInfC{$i : \bot$}
\RightLabel{$\supset\mathrm{I}_i$}
\UnaryInfC{$i : \neg(A \wedge B)$}
\AxiomC{$i : \neg C$}
\RightLabel{$\wedge\mathrm{I}_i$}
\BinaryInfC{$i : \neg(A \wedge B) \wedge \neg C$}
\RightLabel{$\supset\mathrm{I}_i$}
\UnaryInfC{$i : \neg C \supset \neg(A \wedge B) \wedge \neg C$}
\RightLabel{$R_{\mathrm{up}.i}$}
\UnaryInfC{$i+1 : T(``\neg C \supset \neg(A \wedge B) \wedge \neg C")$}
\end{prooftree}
\tag{14}
$$

are:

$$\beta_1 = \begin{array}{l} i+1 : T(\text{“}A \supset C\text{”}) \\ i : A \supset C \\ i : C \end{array} \qquad \beta_2 = \begin{array}{l} i : A \wedge B \\ i : A \end{array}$$

$$\beta_3 = \begin{array}{l} i : \neg C \\ i : \bot \\ i : \neg(A \wedge B) \\ i : \neg(A \wedge B) \wedge \neg C \\ i : \neg C \supset \neg(A \wedge B) \wedge \neg C \\ i+1 : T(\text{“}\neg C \supset \neg(A \wedge B) \wedge \neg C\text{”}) \end{array} \qquad \beta_4 = \begin{array}{l} i : \neg C \\ i : \neg(A \wedge B) \wedge \neg C \\ i : \neg C \supset \neg(A \wedge B) \wedge \neg C \\ i+1 : T(\text{“}\neg C \supset \neg(A \wedge B) \wedge \neg C\text{”}) \end{array}$$

$\beta_3$ and $\beta_4$ are the main branches of 14.

DEFINITION 43. A *part* (or *subpart*) $\tau'$ of a thread $\tau$ is a sequence of adjacent elements of $\tau$.

THEOREM 44. *Let* $\Pi$ *be a normal deduction and* $\beta = i_1 : A_1, i_2 : A_2, \ldots, i_n : A_n$ *a branch of* $\Pi$. *There is an occurrence* $i_h : A_h$ *in* $\beta$, *called* minimum formula of $\beta$, *which separates* $\beta$ *in two possibly empty parts, called the* E-part *(*$i_1 : A_1, \ldots, i_{h-1} : A_{h-1}$*) and the* I-part *(*$i_{h+1} : A_{h+1}, \ldots, i_n : A_n$*) of* $\beta$, *with the following properties:*

1. *The E-part is either empty or composed of* $m+1$ $(m \geq 0)$ *not empty parts of* $\beta$, *called* subE-parts. *The k-th subE-part from the top of* $\beta$ *is called subE*$_k$*-part;*

2. *each subE-part, except the first, begins with a premise of a* $R_{\mathrm{dn.}}$*;*

3. *each occurrence of the subE-parts, except the last (occurrence), is a major premise of an E-rule;*

4. *each subE-part, except the last, ends with a premise of a* $\bot$*-rule;*

5. *the minimum formula* $i_h : A_h$ *of* $\beta$, *provided that* $k \neq n$, *is the premise of a* $\bot$*-rule or of an I-rule;*

6. *each occurrence in the I-part, except the last, is the premise of an I-rule.*

Figures 5 and 6 give two graphical views of the form of branches in a normal deduction. An example is given below.

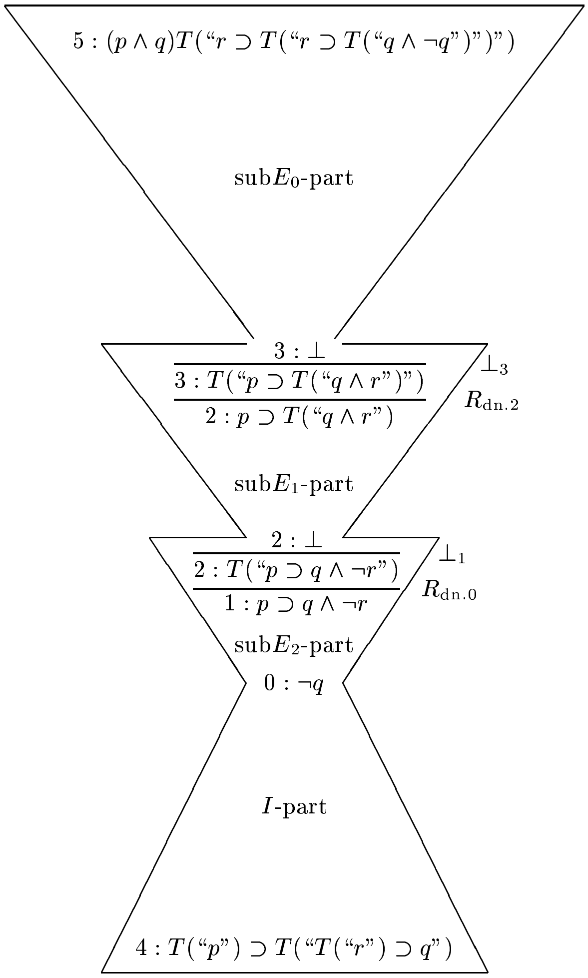


*Figure 5.* The increasing/decreasing complexity of formulas in a branch of a normal deduction.

EXAMPLE 45. Consider the deduction:

$$\dfrac{\dfrac{i : C \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+2 : A \quad i+2 : \neg A}{i+2 : \bot}\,\supset\mathrm{E}_{i+2}}{i+2 : T(\text{“}B \wedge T(\text{“}C \supset D\text{”})\text{”})}\,\bot_{i+2}}{i+1 : B \wedge T(\text{“}C \supset D\text{”})}\,R_{\mathrm{dn}.i+1}}{i+1 : T(\text{“}C \supset D\text{”})}\,\wedge\mathrm{E}_{i+1}}{i : C \supset D}\,R_{\mathrm{dn}.i}}{i : D}\,\supset\mathrm{E}_i}{i+1 : T(\text{“}D\text{”})}\,R_{\mathrm{up}.i} \tag{15}$$

$$
\begin{array}{lll}
 & & i_1 : A_1 \\
\text{E-part} \left\{ \begin{array}{l}
\text{assumption} \\[6pt]
\text{subE}_0\text{-part} \\[6pt]
\dfrac{\dfrac{i_{k_1} : \bot}{i_{k_1} : T(\text{“}C_1\text{”})}\ \bot_{i_{k_1}}}{i_{k_1} - 1 : C_1}\ R_{\mathrm{dn}.i_{k_1}-1} \\[6pt]
\text{subE}_1\text{-part} \\[6pt]
\dfrac{\dfrac{i_{k_2} : \bot}{i_{k_2} : T(\text{“}C_2\text{”})}\ \bot_{i_{k_2}}}{i_{k_2} - 1 : C_2}\ R_{\mathrm{dn}.i_{k_2}-1} \\[6pt]
\text{subE}_2\text{-part} \\[6pt]
\vdots \\[6pt]
\text{subE}_{m-1}\text{-part} \\[6pt]
\dfrac{\dfrac{i_{k_m} : \bot}{i_{k_m} : T(\text{“}C_m\text{”})}\ \bot_{i_{k_m}}}{i_{k_m} - 1 : C_m}\ R_{\mathrm{dn}.i_{k_m}} \\[6pt]
\text{subE}_n\text{-part}
\end{array} \right. & &
\begin{array}{c}
\vdots \\[6pt]
i_{k_1} : A_{k_1} \\[6pt]
\vdots \\[6pt]
i_{k_2} : A_{k_2} \\[6pt]
\vdots \\[6pt]
i_{k_m} : A_{k_m} \\[6pt]
\vdots
\end{array} \\[6pt]
 & \text{minimum formula} & i_h : A_h \\[6pt]
\text{I-part} \left\{ \begin{array}{c}
\text{introductions} \\[6pt]
\text{bottom formula} \\
\text{or minor premise} \\
\text{of an } \supset\text{E}
\end{array} \right. & &
\begin{array}{c}
\vdots \\[6pt]
i_n : A_n
\end{array} \\[6pt]
 & i_{k_1} > i_{k_2} > \ldots > i_{k_m} > i_h \leq i_n &
\end{array}
$$

*Figure 6.* Shape of a branch $\beta = i_1 : A_1, \ldots, i_n : A_n$ in a normal deduction.

The parts of the main branch of deduction (15) are shown in the following picture:

$$
\begin{array}{rcl}
\text{subE}_0\text{-part} & = & \left\{ \begin{array}{l} i+2 : \neg A \\ i+2 : \bot \end{array} \right. \\
\text{subE}_1\text{-part} & = & \left\{ \begin{array}{l} i+2 : T(\text{“}B \wedge T(\text{“}C \supset D\text{”})\text{”}) \\ i+1 : B \wedge T(\text{“}C \supset D\text{”}) \\ i+1 : T(\text{“}C \supset D\text{”}) \\ i : C \supset D \end{array} \right. \\
\text{minimum formula} & = & \quad i : D \\
\text{I-part} & = & \left\{ \ i+1 : T(\text{“}D\text{”}) \right.
\end{array}
$$

Notice that a subE-part can contain formulas of different layers.

*Proof (of Theorem 44).* Let $\beta$ be the branch of a normal deduction, as shown in Figure 6. In $\beta$ all the applications of E-rules and $\perp$-rules precede all the applications of I-rules. Indeed, if this were not the case, since the consequence of an I-rule cannot be the premise of an $\perp$-rule, then there should be an occurrence that is the consequence of an I-rule and major premise of an E-rule. But this contradicts the fact that $\Pi$ is normal.

Let $i_h : A_h$ be the first occurrence in $\beta$ that is the premise of an I-rule, if it exists; otherwise let $i_h : A_h$ be $i_n : A_n$. Let $i_h : A_h$ be the *minimum formula*. Let the E-part of $\beta$ be the subpart of $\beta$ which starts with $i_1 : A_1$ and ends with $i_{h-1} : A_{h-1}$. Let the I-part of $\beta$ be the subpart of $\beta$ which starts with $i_{h+1} : A_{h+1}$ and ends with $i_n : A_n$.

Let $i_{k_1} : A_{k_1}, \ldots, i_{k_m} : A_{k_m}$ be all the occurrences of the E-part of $\beta$ which are consequences of a $\perp$-rule and premises of $R_{\mathrm{dn.}}$ (see Figure 6). Let the subE-parts be the subparts of the E-part separated by these occurrences. Each subE-part, but the first one, begins with $i_{k_j} : A_{k_j}$.

Items 1, 2 and 5 are verified by the choice of the subE-parts.

Let us consider item 3. Let $i : A$ be any occurrence of a subE-part. It is not the premise of an I-rule. If it is the premise of a $\perp$-rule, then the consequence is either the premise of an I-rule and $i : A$ is the minimum formula, or the premise of $R_{\mathrm{dn.}}$ and $i : A$ is the last occurrence of the subE-part.

Let us consider item 6. Let $i : A$ be any occurrence in the I-part. It is not the premise of an E-rule. If it is the premise of a $\perp$-rule, then it cannot be the consequence of an I-rule and it cannot be the consequence of a $\perp$-rule (see restriction on the $\perp$-rule in Definition 15). Therefore it is the consequence of an E-rule, which means that it is the minimum formula $i_h : A_h$. This contradicts the fact that it occurs in the I-part. We can conclude that $i : A$ is either the last occurrence of the I-part or the premise of an I-rule.

### 4.4. The Subformula Property

The subformula property says that, if $A$ is derivable from a set of assumptions $\Gamma$, then there is a deduction containing only subformulas of $\Gamma$ and $A$. One of the main consequences of the subformula property is that, in order to prove that a certain formula is a theorem, it is sufficient to reason in the space of its subformulas, which is finite. If a deduction system enjoys the subformula property, it can be used to support the development of complete automatic methods of proof search and for the definition an automatic decision procedure.

Let us start by defining when a labelled formula $i : A$ is a subformula of another labelled formula $j : B$. Recall that in a language $L$, a formula $A$ is a subformula of a formula $B$, if $A$ appears as a constitutive element of $B$, according to the inductive definition of the formulas of $L$.

DEFINITION 46. $i : A$ is a *subformula* of $j : B$ according to the following rules:

1. if $A$ is a subformula of $B$ in $L_i$, then $i : A$ is a *subformula* of $i : B$;
2. $i : A$ is *subformula* of $i+1 : T(\text{“}A\text{”})$;
3. if $i : A$ is a *subformula* of $j : B$, and $j : B$ is a *subformula* of $k : C$, then $i : A$ is a *subformula* of $k : C$.

Our notion of subformula is the natural extension of the corresponding notion for propositional languages. In particular we keep the property that any formula has a finite number of subformulas.

In the classical natural deduction system C′, as defined in (Prawitz, 1965), the *subformula property* holds under very general hypotheses. That is: all the occurrences of a normal deduction are subformulas of the assumptions, or of the conclusion, except for the assumptions discharged by the application of a $\bot_c$. One of the critical points in the proof of subformula property for C′ concerns the effects of the $\bot_c$ rule. Prawitz in (Prawitz, 1965) solves this problem by restricting the applications of the $\bot_c$-rule to those with atomic consequences. He provides a way to replace the applications of the $\bot_c$-rules in normal deductions with applications of the $\bot_c$-rule with atomic consequences. This has the nice effect that no E-rule can be applied after a $\bot_c$-rule. The presence of multiple layers makes this trick not sufficient. Indeed, in MK, atomic formulas can be premises of E-rules (namely the atomic formula $T(\text{“}A\text{”})$ can be a premise of a $R_{\mathrm{dn.}}$). To understand the effects of this fact on the subformula property, consider the following simple deduction:

$$\dfrac{\dfrac{i+1 : \bot}{i+1 : T(\text{“}A\text{”})}}{i : A}\, R_{\mathrm{dn}.i} \tag{16}$$

(16) is in normal form, but it does not have the subformula property. Indeed $i+1 : T(\text{“}A\text{”})$ is not a subformula of $i+1 : \bot$ nor of $i : A$. Due to counterexamples such as (16), we need to adapt Prawitz's idea as follows. In a first step (Lemma 47) we prove a weak version of subformula property, which holds for all normal deductions, and, in particular, for deductions such as (16). In a second step (Lemma 50) we exploit the weak version of subformula property to show how deductions of the form (16) can be transformed into normal deductions which enjoy the subformula property.

LEMMA 47. *Every formula occurrence in a normal deduction of* $i : A$ *from* $\Gamma$, *with the exception of the assumptions discharged by an application of the* $\perp$*-rule, and the occurrences of the form* $j : \perp$ *occurring immediately below an assumption, has one of the following properties:*

1. *it is a subformula of an element of* $\Gamma$, *or*

2. *it is a subformula of* $i : A$, *or*

3. *it is a subformula of the premise of an* $R_{\mathrm{dn.}}$ *which is a consequence of a* $\perp$*-rule,*

*Proof.* We define the *order of a branch* in a normal deduction as follows. The main branches have order 0. A branch that ends with a minimum premise of an $\supset$E-rule, whose maximum premise belongs to a branch with order $n$, has order $n+1$.

Let $\beta$ be a branch of order $n+1$. Let $j : B$ be an occurrence of the I-part of $\beta$. Then $j : B$ is a subformula of an element of a branch of order $n$.

Let $j : B$ be an occurrence of the subE$_0$-part of $\beta$. Then $j : B$ is a subformula of an assumption that, either belongs to $\Gamma$, or it is discharged in the I-part of $\beta$, or in the I-part of a branch of order less then or equal to $n$.

Finally, let $j : B$ be an occurrence the subE$_k$-part (with $k > 0$) of $\beta$. Then $j : B$ is a subformula of the first occurrence of the subE$_k$-part, which is a consequence of a $\perp$-rule and a premise of a $R_{\mathrm{dn.}}$. □

In order to prove the subformula property, we have to provide a way to remove those formulas which are both consequences of $\perp$-rules and premises of $R_{\mathrm{dn.}}$. If the index of these formulas is less than, or equal to that of the conclusion, they can be removed. This operation is described in Lemma 48. There are cases, however, where this is not possible (see deduction (16)). In these cases, by Lemma 50, we provide a way to reduce them to a small, and finite, set of special cases.

LEMMA 48. *If* $\Gamma \vdash_{MK} i : A$, *there is a normal deduction of* $i : A$ *from* $\Gamma$, *where all the occurrences which are both consequences of a* $\perp$*-rule and premises of a* $R_{\mathrm{dn.}}$ *have index greater than* $i$.

*Proof.* Let $\Pi$ be a normal deduction for $i : A$. Let $j+1 : T(\text{“}B\text{”})$ be a consequence of a $\perp$-rule and a premise of $R_{\mathrm{dn}.j}$, with $j < i$. As $\Pi$ ends in $i$ and $j < i$, there must be an occurrence below $j+1 : T(\text{“}B\text{”})$ which is a consequence of $R_{\mathrm{up}.j}$. Let $j+1 : T(\text{“}C\text{”})$ be the first of such occurrences. To remove $j+1 : T(\text{“}B\text{”})$, we apply the following transformation:

$$
\begin{array}{c}
j+1 : \neg T(``B") \\
\Pi_1 \\
\dfrac{\dfrac{j+1 : \bot}{j+1 : T(``B")}}{j : B} \\
\Pi_2 \\
\dfrac{j : C}{j+1 : T(``C")} \\
\Pi_3
\end{array}
\qquad\qquad
\begin{array}{c}
\dfrac{j+1 : T(``B")}{j : B} \\
\Pi_2 \\
\dfrac{\dfrac{\dfrac{j : C}{j+1 : T(``C")} \quad j+1 : \neg T(``C")}{j+1 : \bot}}{(j+1 : \neg T(``B"))} \\
\Pi_1 \\
\dfrac{j+1 : \bot}{j+1 : T(``C")} \\
\Pi_3
\end{array}
\tag{17}
$$

□

We are now left with the consequences of the $\bot$-rule which are premises of $R_{\text{dn.}}$ with index greater than the index of the conclusion. As shown by the counterexample (16), there are cases where a normal deduction cannot be reduced anymore and "not-subformula" occurrences cannot be removed. We overcome this problem by restricting such occurrences to the single formula $T(``\bot")$.

DEFINITION 49. Given a normal deduction $\Pi$ of $i : A$ from $\Gamma$, we call *redundant formulas* of $\Pi$ those occurrences that:

1. have index $j > i$;
2. are not of the form $j : T(``\bot")$;
3. are consequences of a $\bot$-rule and premises of $R_{\text{dn}.j-1}$,
4. are not subformulas of any element in $\Gamma \cup \{i : A\}$.

The next lemma shows how to remove all redundant formulas, possibly by replacing them with occurrences of the form $j : T(``\bot")$. The underlying idea can be seen by noticing that deduction (16) can be rewritten as:

$$
\dfrac{\dfrac{i+1 : \bot}{i+1 : T(``\bot")}\,\bot}{\dfrac{i : \bot}{i : A}\,\bot_i}\,R_{\text{dn}.i}
$$

LEMMA 50. *If $\Gamma \vdash_{MK} i : A$, then there is a normal deduction of $i : A$ from $\Gamma$, without redundant formulas.*

*Proof.* Let $\Pi$ be a normal deduction of $i : A$ from $\Gamma$. Lemma 50 is proved by defining an operator that either removes redundant formulas, or replaces them with occurrences of the form $j+1 : T(\text{“}\bot\text{”})$.

Let $j+1 : T(\text{“}B\text{”})$ be a redundant formula of $\Pi$, such that all the other redundant formulas have index lower than or equal to $i+1$, and such that all the redundant formulas with index equal to $i+1$ occur below $j+1 : T(\text{“}B\text{”})$. Then, $\Pi$ is of the form:

$$
\dfrac{\dfrac{\begin{array}{c}\Gamma_{>j} \quad j+1 : \neg T(\text{“}B\text{”}) \\ \Pi_1 \\ j+1 : \bot\end{array}}{j+1 : T(\text{“}B\text{”})}\,\bot_{j+1}}{\begin{array}{c} j : B \\ \Pi_2 \\ i : A\end{array}}\,R_{\mathrm{dn}.j} \tag{18}
$$

The operation that, either removes $j+1 : T(\text{“}B\text{”})$, or replaces it by $j+1 : T(\text{“}\bot\text{”})$, is defined in two steps:

(Step 1) We remove all the occurrences of the form $j+1 : T(\text{“}B\text{”})$ which are consequences of $R_{\mathrm{up}.j}$ and which occur above the redundant formula $j+1 : T(\text{“}B\text{”})$. We do this by transforming the subdeduction $\Pi_1$ of $\Pi$, shown in (19), either into the deduction (20), or into the deduction (21).

$$
\dfrac{\dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\Gamma_{>j} \quad j+1 : \neg T(\text{“}B\text{”}) \\ \Pi_1^1 \\ j+1 : T(\text{“}B_1\text{”})\end{array}}{j : B_1}\,R_{\mathrm{dn}.j} \quad \ldots \quad \dfrac{\begin{array}{c}\Gamma_{>j} \quad j+1 : \neg T(\text{“}B\text{”}) \\ \Pi_1^n \\ j+1 : T(\text{“}B_n\text{”})\end{array}}{j : B_n}\,R_{\mathrm{dn}.j} \\ \Pi_1' \\ j : B\end{array}}{\begin{array}{c} j+1 : T(\text{“}B\text{”}) \\ \Pi_1'' \\ j+1 : \bot \end{array}}\,R_{\mathrm{up}.j}}{\dfrac{j+1 : T(\text{“}B\text{”})}{j : B}\,R_{\mathrm{dn}.j}}\,\bot_{j+1} \tag{19}
$$

In (19), $\Pi_1'$ is a deduction of $j : B$ from $j : B_1, \ldots, j : B_n$, with indexes lower than, or equal to $j$. We consider two cases, either there is a $B_k$ $(1 \leq k \leq n)$ equal to $B$, or, for all $1 \leq k \leq n$, $B_k$ is different from $B$.

In the first case deduction (19) is reduced as follows:

$$\begin{array}{c} \Gamma_{>j} \quad j+1:\neg T(\text{“}B\text{”}) \\ \Pi_1^k \\ j+1:T(\text{“}B_k\text{”}) \\ \Pi_1'' \\ \dfrac{\dfrac{j+1:\bot}{j+1:T(\text{“}B\text{”})}\,\bot_{j+1}}{j:B}\,R_{\mathrm{dn}.j} \end{array} \tag{20}$$

Otherwise (19) is rewritten as follows:

$$\begin{array}{c} \dfrac{\dfrac{\dfrac{j+1:\neg T(\text{“}B_1\text{”}) \qquad \begin{array}{c}\Gamma_{>j} \quad j+1:\neg T(\text{“}B\text{”}) \\ \Pi_1^1 \\ j+1:T(\text{“}B_1\text{”})\end{array}}{j+1:\bot}\supset\!\mathrm{E}_{j+1}}{j+1:T(\text{“}B_1\text{”})}\,\bot_{j+1}}{j:B_1}\,R_{\mathrm{dn}.j} \quad \ldots \quad \dfrac{\dfrac{\dfrac{j+1:\neg T(\text{“}B_n\text{”}) \qquad \begin{array}{c}\Gamma_{>j} \quad i+1:\neg T(\text{“}B\text{”}) \\ \Pi_1^n \\ j+1:T(\text{“}B_n\text{”})\end{array}}{j+1:\bot}\supset\!\mathrm{E}_{j+1}}{j+1:T(\text{“}B_n\text{”})}\,\bot_{j+1}}{j:B_n}\,R_{\mathrm{dn}.j} \\ \Pi_1' \\ j:B \end{array} \tag{21}$$

Notice that, in the transformation from (19) to (21), we do not introduce new redundant formulas. Indeed, from Lemma 47, $j:T(\text{“}B_k\text{”})$ is a subformula of an assumption in $\Gamma$ or of $i:A$, and therefore it cannot be a redundant formula.

(Step 2) From the previous step we obtain a deduction of $i:A$ from $\Gamma$ and $j+1:\neg T(\text{“}B\text{”})$, such that $j+1:T(\text{“}B\text{”})$ does not appear as a consequence of $R_{\mathrm{up}.j}$. The next step is to substitute $T(\text{“}B\text{”})$ with $\bot$ in all the occurrences $k:C$, such that $j+1:T(\text{“}B\text{”})$ is a subformula of $k:C$. We also replace each application of $R_{\mathrm{dn}.j}$ to $j+1:T(\text{“}B\text{”})$ with the following deduction:

$$\dfrac{\dfrac{\dfrac{j+1:\bot}{j+1:T(\text{“}\bot\text{”})}\,\bot}{j:\bot}\,R_{\mathrm{dn}.j}}{j:B}\,\bot_j$$

As a result we obtain a deduction of $i:A$ from $\Gamma$ and $\neg\bot$. This deduction can be easily rewritten into a deduction of $i:A$ from $\Gamma$. □

We can finally prove our main theorem.

THEOREM 51. *If* $\Gamma_{\leq k} \vdash_{MK} i:A$, *there is a normal deduction* $\Pi$ *of* $i:A$ *from* $\Gamma_{\leq k}$ *such that every formula occurrence of* $\Pi$ *is a subformula either of an element of* $\Gamma_{\leq k}$, *or of* $i:A$, *or of* $j:T(\text{“}\bot\text{”})$ *with* $k \geq j > i$, *with the exception of the assumptions discharged by an application of* *the $\bot$-rule, and the occurrences of the form $j : \bot$ occurring immediately below an assumption.*

*Proof.* The proof follows from Lemma 48 and Lemma 50. □

## 5. Related Work

In the area of logical frameworks for the formalization of contexts, there are various approaches which are somehow similar to ML systems. In particular we consider Gabbay's Labelled Deductive Systems (LDS) (Gabbay, 1994) and McCarthy's propositional logic of contexts (Buvac and Mason, 1993). We discuss below Masini's 2-sequent calculi (Masini, 1992), because of its similarity with MK and their relation to modal logics.

### 5.1. Gabbay's Labelled Deductive Systems

LDSs are very general logical systems. They are based on two logical languages: a data language and a labelling language. The data language, usually a propositional, modal or first order language, allows to express facts about a specific domain. As it happens with ML systems, in LDSs facts are stated in a context, which is a point, in an algebraic structure. The labelling language, which in most cases is a first order language, allows us to predicate about such an algebraic structure. A theory in an LDS is composed of two classes of statements: *declarative units*, namely pairs *label : formula* (e.g., $\lambda : A$) representing the fact that a formula holds at a point in the algebraic structure, and formulas in the labelling language (e.g., $\lambda_1 \prec \lambda_2$), which express properties of the algebraic structure. Analogously, the deductive machinery of an LDS allows one to reason about both declarative units and formulas in the labelling language.

The most natural analogy between LDSs and ML systems can be obtained by seeing LDS's labels as indexes in ML systems. This analogy is reasonable as indexes in ML systems allow to structure a flat set of statements into a set of interacting theories, and labels in LDS allow to structure a flat set of statements into a set of points of an algebraic structure.

A first main difference between the two approaches concerns the data languages. An LDS contains a single data language, while in ML system each index has associated its own data language. In LDSs, any formula of the data language can be stated in any point of the algebraic structure, namely, the set of declarative units is the cartesian product of the set of labels (= terms of the labelling language) and the formulas of the data language. Differently in ML systems, formulas can be stated only in those contexts where they belongs to the associated language.

The second main difference concerns labels. In LDSs, labels are part of the logic while in ML systems labels (indexes) are not. This means that an LDS contains terms for labels (constant variables, and functions) denoting points in the algebraic structure and predicates on labels representing relations among points of an algebraic structure. Moreover in LDSs, it is possible to reason about labels. In ML systems indexes (labels) are just constants, they are not part of a logical language, and there is no specific set of inference rules to reason about them.

A more effective and detailed comparison among ML systems and LDSs can be done by focusing on a specific sub-class of LDSs. A special class of LDSs, called *Modal LDSs* (MLDSs), have been studied in (Russo, 1996) and used to provide a *uniform* natural deduction style proof system for all normal modal logics. The class of MLDSs is the analogous of the class of MR systems. To make this analogy clearer let us recall some intuitions and terminology of MLDSs.

The data language of MLDSs is any propositional modal language, and its labelling language is a first order language containing the binary relation $R$. The data language represents the syntax of modal logics and the labelling language formalizes the set of Kripke frames on a set of possible worlds, where labels denote possible worlds, and the relational symbol $R$ denotes the accessibility relation. An MLDS is associated with a *labelling algebra*, which is a set of axioms about $R$. An MLDS *configuration* is composed of a set of $R$-literals (expressions of the form $R(\lambda_1, \lambda_2)$ $[\neg R(\lambda_1, \lambda_2)]$ with the intuitive meaning that $\lambda_2$ is [not] accessible from $\lambda_1$) and a set of declarative units. Inference rules for MLDS are given in a Natural Deduction style. They are defined among configurations. The application of an inference rule to a configuration can add new declarative units and/or $R$-literals.

In MLDSs there are three classes of rules, namely, *Classical Rules*, *Modal Rules*, and *Structural Rules*. Classical Rules infer a declarative unit from a set of declarative units with the same label. An example of Classical Rule is the following:

$$\frac{\mathcal{C}\langle\lambda : A \wedge B\rangle}{\mathcal{C}'\langle\lambda : A\rangle}\ \mathcal{I}_{\wedge E}$$

with the following meaning: from the configuration $\mathcal{C}$ containing $\lambda : A \wedge B$, it is possible to infer a configuration $C'$ containing $C$ and $\lambda : A$. Classical Rules correspond to $i$-rules in MR systems.

Modal Rules allow for the introduction and elimination of modal operators. This is done by inferring declarative units in a label $\lambda$ from a declarative unit in a label $\lambda'$ that is related with $\lambda$ via $R$. Two examples of Modal Rules are the following:

$$\frac{\mathcal{C}\langle \lambda : \Box A, R(\lambda, \lambda') \rangle}{\mathcal{C}'\langle \lambda' : A \rangle}\ \mathcal{I}_{\Box E} \qquad \frac{\begin{array}{c}\mathcal{C}\langle [R(\lambda, box_A(\lambda))] \rangle \\ \vdots \\ \mathcal{C}'\langle box_A(\lambda) : A \rangle\end{array}}{\mathcal{C}''\langle \lambda : \Box A \rangle}\ \mathcal{I}_{\Box I}$$

$[\mathcal{I}_{\Box E}]$ can be read as follows: if $\Box A$ holds in possible world $\lambda$ ($\lambda : \Box A$) and $\lambda'$ is accessible from $\lambda$ ($R(\lambda, \lambda')$), then $A$ holds in $\lambda'$. $[\mathcal{I}_{\Box I}]$ can be intuitively read as, to prove that $\Box A$ holds in $\lambda$, consider any world accessible from $\lambda$, (notice that this world depends from $\lambda$ and from $A$ then it is denoted by $box_A(\lambda)$), if $A$ is true in such a world ($box_A(\lambda) : A$), then it is possible to conclude that $\Box A$ is true in $\lambda$. Modal Rules correspond to reflection rules in MR systems. The above rules, for instance, correspond to $R_{\mathrm{dn.}}$ and restricted $R_{\mathrm{up.}}$ respectively. Notice that the technical trick of introducing the label $box_A(\lambda)$ is the counterpart of the restriction in the application of the $R_{\mathrm{up.}}$ rule.

Finally, Structural Rules allow us to infer $R$-literals from the axioms of the labelling algebra of an MLDS. Structural Rules are composed of a rule for classical first order reasoning in the labelling language, and rules for propagation of $\perp$ through accessible worlds. These rules don't have any counterpart in MR systems, as in MR systems there is no reasoning about the structure of contexts.

Let's now discuss analogies and differences between MR systems and MLDSs. The first point concerns the different intuitive interpretation of labels. In MLDS a label, in most cases, denotes a possible world in a Kripke Structure (in some cases, a label denotes a point with no associated possible world, when a label $\lambda$ is such that $\lambda : \perp$ holds), while in an MR system an index (label) $i$ denotes a theory in the language $L_i$, i.e., set of interpretations of $L_i$ (truth assignments to the atoms of $L_i$). As a consequence, the assumption $\lambda : A$ in a deduction of an MLDS formalizes the hypothesis "let's suppose that $A$ holds at the world $\lambda$", while the assumption $i : A$ in a deduction of an MR system formalizes the hypothesis "let's suppose the set of interpretations of $L_i$ that satisfy $A$". The accessibility relation $R$ of an MLDS corresponds to the partial order $\prec$ in an MR system. The correspondence, however, is not straightforward, as $R$ and $\prec$ are defined on different objects (worlds versus theories). The $R$-literal $R(\lambda, \lambda')$ in an MLDS configuration corresponds, in MR systems, to the fact that, if $\lambda$ denotes a world with a the truth assignment which is a model of the $i$-th theory, and $j \prec i$, then $\lambda'$ denotes a world with a truth assignment which is a model of the $j$-th theory. The analogy is deeper when we consider the special label $box_A(\lambda)$. Such a label, indeed, is supposed to span over a set of possible worlds accessible from $\lambda$ and associated with $A$. Therefore if $\lambda$ denotes a world with an assignment which is a model of the $i$-th theory, and $j \prec i$, then $box_A(\lambda)$ denotes any world with an assignment which is a model of the $j$-th theory.

The second difference concerns the structure of the labels/indexes. An MR system contains a single relation $\prec$ on the set of indexes. There is no trivial way to consider multiple or partially specified accessibility relations. In an MLDS, instead, it is possible to reason about partially specified accessibility relations. An MR system corresponds to an MLDS in which the labelling algebra has a single model. This difference makes MR systems not applicable when label structure is partially known and it is necessary to reason about it. On the other hand, MR systems are more suitable when the structure of the labels is completely known. Such a structure can be left implicit both in the syntax (there is no logical language and inference rules for labels) and in the semantics (there is no specific model for the domain of the labels). This is the case, for instance, of well known modal logics. Indeed the MR systems for normal modal logics MBK, MB4, MB5, and MB45 defined in Subsection 3.2, result simpler than the analogous MLDSs, as they don't contain any structural rule (i.e., rules for labels).

### 5.2. McCarthy's Propositional Logics of Contexts

In (Buvac and Mason, 1993) Buvač and Mason proposed a propositional modal logic (called, from now on, BM-logic) which formalizes McCarthy's intuitions about contexts, as discussed in (McCarthy, 1993). Given a set $\mathcal{K}$ of labels, intuitively denoting contexts, BM-Logic is a multi modal logic based on a set of atomic propositions $P$ and the modality $ist(\kappa, \phi)$ for each context (label) $\kappa \in \mathcal{K}$. Differently from MR systems, where formulas are stated in contexts, in BM-logic formulas are stated in (possible empty) sequences of contexts $\overline{\kappa} = \kappa_1 \ldots \kappa_n$. Intuitively the sequence of contexts $\kappa_1\kappa_2$ represents how context $\kappa_2$ is viewed from context $\kappa_1$. As a consequence, the intuitive meaning of the formula $ist(\kappa, \phi)$ in the empty sequence of contexts, is that $\phi$ holds in the context $\kappa$; while the intuitive meaning of the same formula in the sequence composed of the single context $\kappa'$, is that $\phi$ holds in context $\kappa$ as viewed from context $\kappa'$. In general a formula $ist(\kappa, \phi)$ in a sequence of contexts $\kappa_1 \ldots \kappa_n$ is interpreted as $\phi$ is true in the context $\kappa$ according to how context $\kappa_1$ sees context $\kappa_2$ ... sees context $\kappa_n$.

The semantics of BM-logic is a function which assigns to each sequence of contexts $\overline{\kappa}$ a set of truth assignments for a subset $\mathsf{Vocab}(\overline{\kappa})$ of the set of propositions $P$. Satisfiability is defined as usual; in particular $ist(\kappa, \phi)$ is satisfied at $\overline{\kappa}$ by an assignment, if $\phi$ is satisfied at $\overline{\kappa}\kappa$ by all the assignments associated to $\overline{\kappa}\kappa$. ($\overline{\kappa}\kappa$ is the sequence obtained by the concatenation of the sequence $\overline{\kappa}$ with the context label $\kappa \in \mathcal{K}$). Fixed a vocabulary $\mathsf{Vocab}$, the axiomatization of the validity in the class of models based on $\mathsf{Vocab}$ (i.e., the models that associate to each sequence of contexts $\overline{\kappa}$ a set of interpretations for $\mathsf{Vocab}(\overline{\kappa})$) contains the axioms in each $\overline{\kappa}$ for modal K restricted to $\mathsf{Vocab}(\overline{\kappa})$, Modus Ponens, a form of Necessitation rule (similar to our reflection up), and the following axiom schema:

$$\vdash_{\overline{\kappa}} ist(\kappa_1, ist(\kappa_2, A) \vee B) \supset ist(\kappa_1, ist(\kappa_2, A)) \vee ist(\kappa_1, B) \qquad (\Delta)$$

A detailed comparison between ML systems and MB logic is provided in (Bouquet and Serafini, 2000). In the following we provide some hints, without entering into the technical details. BM-logic can be reformulated in a specific MR system. The ML system corresponding to the BM-logic for a finite number $n$ of contexts,[2] called MMCC (ML system for McCarthy's Contexts) is based on MBK(n), with the following simple changes on the languages and on the applicability restriction of $R_{\mathrm{up.}}$:

- The language of $L_{\overline{\kappa}}$ is built on the set of propositions of $\mathsf{Vocab}(\overline{\kappa})$;
- $R_{\mathrm{up.}\overline{\kappa}\kappa}$ is applicable with *no restrictions* if its premise of the form $\overline{\kappa}\kappa : ist(\kappa', \phi)$. Otherwise it is applicable with the usual restriction.

As proved in (Bouquet et al., 1999), MB and MMCC are equivalent. This equivalence is based on the analogy between the axiom ($\Delta$) and the relaxation on the restriction of $R_{\mathrm{up.}\overline{\kappa}}$. Indeed both formalize the semantic property that the assignments to the formula $ist(\kappa', \phi)$ in the context $\overline{\kappa}\kappa$ coincide in all $\overline{\kappa}\kappa$ models, as they depend only on the assignments to $\phi$ in the contexts $\overline{\kappa}\kappa\kappa'$. In Appendix D we provide a deduction of the formula in MMCC corresponding to the axiom ($\Delta$) above. In spite of their equivalence, there are two main differences between MMCC and BM-logic.

The first difference is that MB logic, being a modal logic, does not support the notion of logical consequence between formulas in different context sequences. Satisfiability of a formula is defined with respect to a context sequence; notationally this corresponds to the fact that the satisfiability symbol is indexed by context sequence ($\models_{\overline{\kappa}}$). The logical consequence definable on the basis of $\models_{\overline{\kappa}}$ is "local" to the context sequence $\overline{\kappa}$. This approach does not allow for a definition of

[2] The same process can be applied for a countable set.

a logical consequence relation between formulas in different context sequences. This limitation is reflected by the calculus of MB logic. This calculus, indeed, allows one to prove only that a formulas is valid in a context $\overline{\kappa}$ (denote by $\vdash_{\overline{\kappa}}$), but it does not allow one to derive the consequences in $\overline{\kappa}'$ of a set of assumptions in $\overline{\kappa}$ different from $\overline{\kappa}'$. Differently, MMCC allows us to derive a formula in a context under a certain set of assumptions in different contexts.

A second difference concerns how BM-logic extends the multi modal logic K(n). BM-logic extend K(n) with the axiom ($\Delta$), while MMCC simply modifies MBK(n) by weakening the applicability restriction of $R_{\mathrm{up.}}$. From the proof theoretical point of view, ($\Delta$) is a complex axiom schema and it might constitute an obstacle to the development of an elegant proof theory and automatic decision procedures by generalizing those for K(n). On the other side, MMCC is a simple extension of MBK(n) (just a weakening of an applicability condition, no new rules are introduced). This allows for a relatively easy adaptation of the proof theoretical results of MBK (analogous to that of MK) to MMCC. Finally the equivalence between MBK(n) and multi modal K, allow for the definition of decision procedures for MMCC, which are simple variants of the decision procedure for modal K(n).

### 5.3. Masini's 2-Sequent Calculi

2-sequent calculi were introduced by Masini in (Masini, 1992) with the goal of providing a cut free sequence calculus for the classical and intuitionistic modal logics KD = K+$\neg\Box\bot$. Conceptually, 2-sequent calculi are very similar to ML systems and LDSs. Indeed, as for ML systems and LDSs, the intuition behind 2-sequents is to introduce more structure in a set of formulas, and in particular in Gentzen sequents. Masini, however, does not add structure using labels; he rather introduces a *vertical dimension* in sequents. The resulting sequences are called 2-sequents. Roughly speaking, 2-sequents are layered sequents. The formulas on the two sides of the $\vdash$ symbol are arranged in a (possible infinite) number of layers. 2-sequents are graphically represented as follows:

$$\begin{array}{c} A \\ B, C \\ Df \\ \phantom{H} \end{array} \vdash \begin{array}{c} \phantom{A} \\ E, F \\ G \\ H \end{array} \tag{22}$$

(22) represents a 2-sequent composed of 4 layers. From top to bottom, the first layer contains the formula $A$ on the left hand side, and no formula on the right hand side; the second layer contains the formulas

$B$ and $C$ on the left side, and the formulas $E$ and $F$ on the right hand side, and so on. 2-sequents are interpreted in Kripke Structures; a 2-sequent $\Gamma \vdash \Delta$ is true at a world $w_1$ of a Kripke Structure, if, for any sequence of worlds $w_1, w_2, \ldots$ such that $w_{i+1}$ is accessible from $w_i$, either there is a layer $i$ and a formula $A \in \Gamma$ at layer $i$ such that $w_i \not\models A$, or for all layers $j$ and for all formulas $B \in \Delta$ at layer $j$, $w_j \models B$. For instance, 2-sequent (22) is satisfied in $w_1$ if and only if, for any 4-tuple of worlds $w_1, w_2, w_3$ and $w_4$ such that, each $w_{i+1}$ is accessible from $w_i$, we have that

$$\left.\begin{array}{l} w_1 \models A \text{ and} \\ w_2 \models B \text{ and } w_2 \models C \text{ and} \\ w_3 \models D \end{array}\right\} \text{implies that} \left\{\begin{array}{l} w_2 \models E \text{ or } w_2 \models F \text{ or} \\ w_3 \models G \text{ or} \\ w_4 \models H \end{array}\right.$$

The definition of satisfiability for 2-sequents suggests that layers in 2-sequents are similar to indexes of the MR system MBK (see also Definition 16). A 2-sequent $\Gamma \vdash \Delta$, with $\Gamma$ finite and $\Delta$ a singleton or empty, can be translated into a statement on the consequence relation of MBK. Namely the 2-sequents:

$$\begin{array}{lll} \Gamma_0 & & \\ \vdots & & \\ \Gamma_i & \vdash & A \\ \vdots & & \\ \Gamma_n & & \end{array} \qquad\qquad \begin{array}{ll} \Gamma_0 & \\ \vdots & \\ \Gamma_i & \vdash \\ \vdots & \\ \Gamma_n & \end{array}$$

are translated in the following statements about MBK consequence relation:

$$\begin{array}{l} 0 : \Gamma_0, \ldots, i : \Gamma_i, \ldots, n : \Gamma_n \;\vdash\; i : A \\ 0 : \Gamma_0, \ldots, i : \Gamma_i, \ldots, n : \Gamma_n \;\vdash\; 0 : \bot \end{array}$$

2-sequents with multiple consequences cannot be directly translated into statements about an ML.C.R.. However, we have that, because of the definition of satisfiability, any 2-sequent with multiple consequences can be transformed into an equivalent 2-sequent with a single consequence. This can be done by negating and moving to the left of the sequent sign all the consequences but one of the lower level. With this simple transformation any finite 2-sequent can be translated in a statement of a ML.C.R.. For instance the 2-sequent (22) corresponds to

$$0 : A,\;\; 1 : B,\;\; 1 : C,\;\; 2 : D,\;\; 1 : \neg E,\;\; 1 : \neg F,\;\; 2 : \neg G \;\vdash\; 3 : H \qquad (23)$$

Masini defines a sequent calculus (called C-2SC) for 2-sequents, which is sound and complete w.r.t. the modal logic KD. C-2SC is composed of structural rules, rules for connectives, and rules for modal operators. Structural rules and rules for connectives are a trivial generalization of the corresponding rules for plain sequents; they involve only formulas in a single layer. Rules for left and right introduction of the modal operator $\Box$, instead, involve multiple layers. These rules are reported in the following:

$$
\dfrac{\begin{array}{c}\Gamma\\ \alpha\\ \beta, A\\ \Gamma'\end{array} \vdash \Delta}{\begin{array}{c}\Gamma\\ \alpha, \Box A\\ \beta\\ \Gamma'\end{array} \vdash \Delta}\ \Box\vdash
\qquad\qquad
\dfrac{\Gamma \vdash \begin{array}{c}\Delta\\ \mu\\ A\end{array}}{\Gamma \vdash \begin{array}{c}\Delta\\ \mu, \Box A\end{array}}\ \vdash\Box
$$

RESTRICTION: $\vdash \Box$ is applicable only if $\Gamma$ does not contain any formula at a layer below $\mu$.

Under the standard translation of the modal formulas $\Box A$ into $B(\text{“}A\text{”})$, and of 2-sequents into statements about an ML consequence relation (see above), C-2SC is logically equivalent and structurally very similar to MBK extended with a special bridge rule, called $\bot$-Prop, which propagates $\bot$ to upper layers.

$$
\frac{i+1 : \bot}{i : \bot}\ \bot\text{-Prop}
$$

The extension of MBK with the $\bot$-Prop rule is justified as MBK is equivalent to modal K, while C-2SC is equivalent to KD. The difference between K and KD concerns the propagation of $\bot$ through the $\Box$ operator. In K, indeed, $\bot$ does not propagate through $\Box$, which means that $\Box\bot \supset \bot$ is not K-valid; while in KD, $\bot$ propagates, and $\Box\bot \supset \bot$ is KD-valid.

Not only is MBK+$\bot$-Prop equivalent to C-2SC, but it is also structurally similar. Indeed, the rules for left and right introduction of the connectives at level $i$ correspond to the $i$-rules of MBK; the inference rule $\Box \vdash$ applied to a formula at level $i$ corresponds to $R_{\text{dn}.i}$ and the rule $\vdash \Box$ applied to a formula at level $i$ corresponds to restricted $R_{\text{up}.i}$. Notice also the analogy between the restriction of the application of $\vdash \Box$ and that on the applicability $R_{\text{up}.i}$ in MBK (see Definition 13).

An important difference between 2-sequents and MBK (and in general ML systems) concerns the propagation of inconsistency through layers. In 2-sequents there is a notion of *global inconsistency*. Global inconsistency is represented by the empty set of layers. For instance the 2-sequent

$$\begin{array}{c} A \\ B \end{array} \vdash$$

states that $B$ cannot hold in a world $w$ which is accessible from a world $w'$ where $A$ holds. On the other hand 2-sequents do not allow the representation of the fact that inconsistency is localized inside a single layer. For instance, the following 2-sequent

$$\begin{array}{c} \\ A \wedge \neg A \end{array} \vdash \begin{array}{c} A \wedge \neg A \\ \\ \end{array} \tag{24}$$

is valid in C-2SC. (24) represents the fact that if layer 1 is inconsistent, then the upper layer is inconsistent as well. Differently, in MBK, it is possible to have inconsistency localized in a context (see Corollary 34).

While the weakest modal system formalized by C-2SC is KD, MR systems allow for the formalization of normal modal logic K and also for weaker systems (see (Giunchiglia and Giunchiglia, 2000)). As a last remark, notice that the generalization of 2-sequents to multi modal logics is not trivial, in the sense that the vertical structure of the sequents should be translated in a more complex tree structure.

## 6. Conclusion

The work presented in this paper is part of a much larger project whose goal is to provide logical and philosophical foundations for distributed knowledge representation and reasoning systems. As argued in the seminal paper (Giunchiglia, 1993) and in other successive papers, MR systems closely resemble the current practice in distributed knowledge representation and reasoning systems. In this paper we have provided a proof theory for a large subclass of ML systems the class of MR systems. The result we have proved have the following main consequences. The first concerns the representational properties of ML systems. Indeed, this work offers an in depth understanding of the behaviour of a specific ML system, which can be reused for similar ML systems. The second concerns the proof theoretical methodology for working with ML systems. Indeed, the proof theory of MK (and its generalization to MR systems) can be used as a guideline for the developement of the proof theory of other ML systems. The last consequence concerns the proof of properties of logical systems which can be embedded in ML systems. Indeed, by proving the subformula property for MK we have indirectly proved the subformula property for a calculus for modal K. Furthermore the generalization of subformula property to MR systems, such as MB4, MB5 and MB45, (equivalent to modal K4, K5, and K45) will constitute a positive answer to the problem of the esistence of a calculus with normal form for these logics.

## Acknowledgements

The research described in this paper owes a lot to the openness and sharing of ideas which exists in the Automated Reasoning Area at IRST and in the Mechanized Reasoning Group of the University of Trento and Genoa. Alex Simpson and Vanni Criscuolo have provided useful feedback on many of the topics described in this paper. The discussions with Massimo Benerecetti, Paolo Bouquet, Alessandro Cimatti, Chiara Ghidini, Enrico Giunchiglia, Paolo Traverso, have provided useful insights. We also thank Alessandro Agostini for his reading of earlier versions of this paper.

## References

Avron, A.: 1987, 'Simple consequence relations'. LFCS Report Series, Laboratory for the Foundations of Computer Science, Computer Science Department, University of Edinburgh.

Benerecetti, M., P. Bouquet, and C. Ghidini: 1998a, 'Formalizing Belief Reports – The Approach and a Case Study'. In: F. Giunchiglia (ed.): *Proceedings AIMSA'98, 8th International Conference on Artificial Intelligence, Methodology, Systems, and Applications*, Vol. 1480 of *LNAI*. Sozopol, Bulgaria, pp. 62–75, Springer Verlag.

Benerecetti, M., P. Bouquet, and C. Ghidini: 2000, 'Contextual Reasoning Distilled'. *Journal of Theoretical and Experimental Artificial Intelligence*. To appear.

Benerecetti, M., E. Giunchiglia, L. Serafini, and A. Villafiorita: 1999, 'Formal specification of beliefs in Multi-Agent Systems'. *International Journal of Intelligent Systems* **14**(10), 1021–1040.

Benerecetti, M., F. Giunchiglia, and L. Serafini: 1998b, 'Model Checking Multiagent Systems'. *Journal of Logic and Computation, Special Issue on Computational & Logical Aspects of Multi-Agent Systems* **8**(3), 401–423. Also IRST-Technical Report 9708-07, IRST, Trento, Italy.

Bouquet, P. and F. Giunchiglia: 1995, 'Reasoning about Theory Adequacy. A New Solution to the Qualification Problem'. *Fundamenta Informaticae* **23**(2–4), 247–262. Also IRST-Technical Report 9406-13, IRST, Trento, Italy.

Bouquet, P. and L. Serafini: 2000, 'Two formalizations of Context: a comparison'. Technical Report 0006-01, ITC-IRST, Trento, Italy.

Bouquet, P., L. Serafini, P. Brezillon, M. Benerecetti, and F. Castellani (eds.): 1999, 'Modelling and Using Context – Proceedings of the 2nd International and Interdisciplinary Conference (9–11 September 1999, Trento, Italy)', Vol. 1688 of *Lecture Notes in Artificial Intelligence*. Springer Verlag - Heidelberg.

Bowen, K. and R. Kowalski: 1982, 'Amalgamating language and meta-language in logic programming'. In: S. Tarlund (ed.): *Logic Programming*. New York, pp. 153–173, Academic Press.

Buvac, S. and I. A. Mason: 1993, 'Propositional logic of context'. *Proc. of the 11th National Conference on Artificial Intelligence*.

Cimatti, A. and L. Serafini: 1996, 'Multi-Agent Reasoning with Belief Contexts II: Elaboration Tolerance'. In: *Proc. 1st Int. Conference on Multi-Agent Systems (ICMAS-95)*. pp. 57–64. Also IRST-Technical Report 9412-09, IRST, Trento, Italy. *Commonsense-96*, Third Symposium on Logical Formalizations of Commonsense Reasoning, Stanford University, 1996.

Clarke, E., O. Grumberg, and D. Long: 1994, 'Model Checking'. In: *Proceedings of the International Summer School on Deductive Program Design*. Marktoberdorf, Germany.

Dichev, C.: 1993, 'Theory Reations and Context Dependencies'. In: *Proceedings of the 6th Australian Joint Conference on Artificial Intelligence (AI-93)*. Melburne, pp. 266–272.

Dinsmore, J.: 1991, *Partitioned Representations*. Kluwer Academic Publishers.

Fagin, R. and J. Halpern: 1988, 'Belief, awareness, and limited reasoning'. *Artificial Intelligence* **34**, 39–76.

Fisher, M. and C. Ghidini: 1999, 'Programming Resource-Bounded Deliberative Agents'. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*. pp. 200–206, Morgan Kaufmann Publ., Inc.

Gabbay, D.: 1994, 'Labeled Deductive Systems Volume 1 – Foundations'. Technical Report MPI-I-94-223, Max-Planck-institut für Informatik.

Giunchiglia, E. and F. Giunchiglia: 2000, 'Ideal and Real Belief about Belief'. *Journal of Logic and Computation*. To appear, Also IRST-Technical Report 9605-04, IRST, Trento, Italy.

Giunchiglia, F.: 1993, 'Contextual reasoning'. *Epistemologia, special issue on I Linguaggi e le Macchine* **XVI**, 345–364. Short version in Proceedings IJCAI'93 Workshop on Using Knowledge in its Context, Chambery, France, 1993, pp. 39–49. Also IRST-Technical Report 9211-20, IRST, Trento, Italy.

Giunchiglia, F. and P. Bouquet: 1998, 'A Context-Based Framework for Mental Representation'. In: *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Society - CogSci'98*. Madison, Wisconsin.

Giunchiglia, F. and C. Ghidini: 1998, 'Local Models Semantics, or Contextual Reasoning = Locality + Compatibility'. In: *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*. pp. 282–289, Morgan Kaufmann. Also IRST-Technical Report 9701-07, IRST, Trento, Italy.

Giunchiglia, F. and L. Serafini: 1994, 'Multilanguage Hierarchical Logics (or: how we can do without modal logics)'. *Artificial Intelligence* **65**, 29–70. Also IRST-Technical Report 9110-07, IRST, Trento, Italy.

Giunchiglia, F., L. Serafini, E. Giunchiglia, and M. Frixione: 1993, 'Non-Omniscient Belief as Context-Based Reasoning'. In: *Proc. of the 13th International Joint Conference on Artificial Intelligence*. Chambery, France, pp. 548–554. Also IRST-Technical Report 9206-03, IRST, Trento, Italy.

Giunchiglia, F., L. Serafini, and A. Simpson: 1992, 'Hierarchical meta-logics: intuitions, proof theory and semantics'. In: *Proc. of META-92, Workshop on Metaprogramming in Logic*. Uppsala, Sweden, pp. 235–249, Springer Verlag. Also IRST-Technical Report 9101-05, IRST, Trento, Italy.

Giunchiglia, F. and P. Traverso: 1996, 'A Metatheory of a Mechanized Object Theory'. *Artificial Intelligence* **80**(2), 197–241. Also IRST-Technical Report 9211-24, IRST, Trento, Italy, 1992.

Masini, A.: 1992, '2-Sequent calculus: a proof theory of modalities'. *Annals of Pure and Applied Logic* **58**, 229–246.

McCarthy, J.: 1979, 'Ascribing Mental Qualities to Machines'. In: M. Ringle (ed.): *Philosophical Perspectives in Artificial Intelligence*. pp. 161–195, Humanities Press. Also in V. Lifschitz (ed.), *Formalizing common sense: papers by John McCarthy*, Ablex Publ., 1990, pp. 93–118.

McCarthy, J.: 1993, 'Notes on Formalizing Context'. In: *Proc. of the 13th International Joint Conference on Artificial Intelligence*. Chambery, France, pp. 555–560.

McCarthy, J. and S. Buvač: 1998, 'Formalizing Context (Expanded Notes)'. In: A. Aliseda, R. van Glabbeek, and D. Westerståhl (eds.): *Computing Natural Language*, Vol. 81 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, pp. 13–50.

Mylopoulos, J. and R. Motschnig-Pitrik: 1995, 'Partitioning Information Bases with Contexts'. In: *Third International Conference on Cooperative Information Systems*. Vienna.

Noriega, P. and C. Sierra: 1996, 'Towards Leayered Dialogical Agents'. In: J. P. Müller, M. J. Wooldridge, and N. R. Jennings (eds.): *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), ECAI-96*, Vol. 1193 of *LNAI*. Budapest, Hungary.

Parsons, S., C. Sierra, and N. Jennings: 1998, 'Agents that reason and negotiate by arguing'. *Journal of Logic and Computation* **8**(3), 261–292.

Penco, C.: 1999, 'Objective and Cognitive Contexts'. In: P. Bouquet, L. Serafini, P. Brezillon, M. Benerecetti, and F. Castellani (eds.): *Modelling and Using Context – Proceedings of the 2nd International and Interdisciplinary Conference, Context'99*, Vol. 1688 of *Lecture Notes in Artificial Intelligence*. pp. 270–283, Springer Verlag - Heidelberg. `http://www.dif.unige.it/epi/hp/penco/pub/OC.ps`.

Prawitz, D.: 1965, *Natural Deduction - A proof theoretical study*. Almquist and Wiksell, Stockholm.

Russo, A.: 1996, 'Modal Logics as Labelled Deductive Systems'. Ph.D. thesis, Imperial College of Science, Technology and Medicine, University of London, London, UK.

Serafini, L. and C. Ghidini: 2000, 'Using wrapper agents to answer queries in distributed information systems'. In: *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS'2000)*. IEEE Computer Society Press. Long version in IRST-Technical Report 9911-01, IRST, Trento, Italy.

Sharma, N.: 1995, 'On Formalising and Reasoning with Contexts'. Technical Report Technical Report 352, Department of Computer Science, The University of Queensland, Australia.

Subrahmanian, V.: 1994, 'Amalgamating Knowledge Bases'. *ACM Trans. Database Syst.* **19**(2), 291–331.

Weyhrauch, R.: 1980, 'Prolegomena to a Theory of Mechanized Formal Reasoning'. *Artificial Intelligence* **13**(1), 133–176.

## Appendix

### A. ML inference rules

An $n$-any inference rule is a tuple of three elements: an $n+1$-ary *relation* between formulas, which represents all the possible applications of the rule; $n$ *discharging functions*, one for each premiss, each computing the assumptions discharged by any application of the rule; and a *restriction relation* which represents the cases where the rule is not applicable.

DEFINITION 52. An $n$-ary inference rule $\text{I}$ from $i_1, \dots i_n$ to $i$, is composed of:

1. an $n+1$-ary *relation*:
   $\rho(\text{I})$ is a recursive subset of $(i_1 : L_{i_1} \times \dots \times i_n : L_{i_n} \times i : L_i)$;
2. $n$ *discharging functions*:
   $d_1(\text{I}), \dots, d_n(\text{I})$ are $n$ recursive functions where $d_k(\text{I}) : \rho(\text{I}) \longrightarrow 2^{L_I}$, for each $1 \leq k \leq n$.
3. an *applicability restriction*:
   $rest(\text{I})$ is a recursive subset of $(2^{L_I} \times i_1 : L_{i_1}) \times \dots \times (2^{L_I} \times i_n : L_{i_n}) \times i : L_i$

We say that $\langle i_1 : A_1, \dots, i_n : A_n, i : A\rangle$ is an *application* of $\text{I}$ if and only if it is in $\rho(\text{I})$.

An $i$-rule is defined as a *multi-language version of some ND inference rule at $i$.*

DEFINITION 53. If $\text{I}$ is an $n$-ary ND inference rule, described as a tuple $\langle \rho(\text{I}), d_1(\text{I}), \dots, d_n(\text{I}), rest(\text{I})\rangle$, then its *multi-language version* with index $i$ is $\text{I}_i = \langle \rho(\text{I}_i), d_1(\text{I}_i), \dots, d_n(\text{I}_i), rest(\text{I}_i)\rangle$ where:

1. $\langle i : A_1, \dots, i : A_n, i : A\rangle \in \rho(\text{I}_i)$ if and only if $\langle A_1, \dots, A_n, A\rangle \in \rho(\text{I})$
2. $i : B \in d_k(\text{I}_i)(i : A_1, \dots, i : A_n, i : A)$ if and only if $B \in d_k(\text{I})(A_1, \dots, A_n, A)$.
3. $\langle\langle \Gamma_1, i : A_1\rangle, \dots, \langle \Gamma_n, i : A_n\rangle, i : A_n\rangle \in rest(\text{I}_i)$ if and only if $\langle\langle G_1, A_1\rangle, \dots, \langle G_n, A\rangle_n, A\rangle \in rest(\text{I})$. where $i : G_k \subseteq \Gamma_k$ $(1 \leq k \leq n)$ is the set of all the wffs with index $i$ occurring in $\Gamma_k$.

A *bridge rule* is any ML inference rule which is not an $i$-rule.

### B. MK basic properties

PROPOSITION 54. *Let $A, B$ and $C$ be any $L_i$-wffs. Then the following are theorems of MK:*

*1.* $i+1 : T(\text{“}A \wedge B\text{”}) \equiv T(\text{“}A\text{”}) \wedge T(\text{“}B\text{”})$

*2.* $i+1 : T(\text{“}A\text{”}) \vee T(\text{“}B\text{”}) \supset T(\text{“}A \vee B\text{”})$

*3.* $i+1 : T(\text{“}A \vee B\text{”}) \supset (T(\text{“}A \supset C\text{”}) \wedge T(\text{“}B \supset C\text{”}) \supset T(\text{“}C\text{”}))$

*4.* $i+1 : T(\text{“}\bot\text{”}) \supset T(\text{“}A\text{”})$

*5.* $i+1 : \neg T(\text{“}\bot\text{”}) \supset (T(\text{“}A\text{”}) \supset \neg T(\text{“}\neg A\text{”}))$

*6.* $i+1 : T(\text{“}A\text{”}) \supset T(\text{“}B \supset A\text{”})$

*7.* $i+1 : T(\text{“}\neg A\text{”}) \supset T(\text{“}A \supset B\text{”})$

*8.* $i+1 : T(\text{“}A \supset B\text{”}) \supset (\neg T(\text{“}\neg A\text{”}) \supset \neg T(\text{“}\neg B\text{”}))$

*9.* $i+1 : (T(\text{“}A\text{”}) \wedge \neg T(\text{“}B\text{”})) \supset \neg T(\text{“}\neg(A \wedge \neg B)\text{”})$

*10.* $i+1 : T(\text{“}A \vee B\text{”}) \supset (\neg T(\text{“}\neg A\text{”}) \vee T(\text{“}B\text{”}))$

*11.* $i+1 : \neg T(\text{“}\neg(A \supset B)\text{”}) \vee T(\text{“}B \supset A\text{”})$

*12.* $i+1 : \neg T(\text{“}\neg(A \supset B)\text{”}) \equiv (T(\text{“}A\text{”}) \supset \neg T(\text{“}\neg B\text{”}))$

*13.* $i+1 : (\neg T(\text{“}\neg A\text{”}) \supset T(\text{“}B\text{”})) \supset T(\text{“}A \supset B\text{”})$

*14.* $i+1 : (\neg T(\text{“}\neg A\text{”}) \supset T(\text{“}B\text{”})) \supset (\neg T(\text{“}\neg A\text{”}) \supset \neg T(\text{“}\neg B\text{”}))$

*15.* $i+1 : T(\text{“}A \supset B\text{”}) \supset ((\neg T(\text{“}B\text{”}) \supset T(\text{“}A\text{”})) \supset T(\text{“}B\text{”}))$

*Proof.* We provide the proofs of the first six theorems, we let the others to the reader.

1.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A \supset B\text{”})}{i : A \supset B}\, R_{\mathrm{dn}.i} \qquad \dfrac{i+1 : T(\text{“}A\text{”})}{i : A}\, R_{\mathrm{dn}.i}}{i : B}\, \supset\!\mathrm{E}_i}{i+1 : T(\text{“}B\text{”})}\, R_{\mathrm{up}.i}}{i+1 : T(\text{“}A\text{”}) \supset T(\text{“}B\text{”})}\, \supset\!\mathrm{I}_{i+1}}{i+1 : T(\text{“}A \supset B\text{”}) \supset (T(\text{“}A\text{”}) \supset T(\text{“}B\text{”}))}\, \supset\!\mathrm{I}_{i+1}$$

2.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A \wedge B\text{”})}{i : A \wedge B}\, R_{\mathrm{dn}.i}}{i : A}\, \wedge\mathrm{E}_i}{i+1 : T(\text{“}A\text{”})}\, R_{\mathrm{up}.i} \qquad \dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A \wedge B\text{”})}{i : A \wedge B}\, R_{\mathrm{dn}.i}}{i : B}\, \wedge\mathrm{E}_i}{i+1 : T(\text{“}B\text{”})}\, R_{\mathrm{up}.i}}{i+1 : T(\text{“}A\text{”}) \wedge T(\text{“}B\text{”})}\, \wedge\mathrm{I}_{i+1}}{i+1 : T(\text{“}A \wedge B\text{”}) \supset T(\text{“}A\text{”}) \wedge T(\text{“}B\text{”})}\, \supset\!\mathrm{I}_{i+1}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A\text{”}) \wedge T(\text{“}B\text{”})}{i+1 : T(\text{“}A\text{”})}\;\wedge\mathrm{E}_{i+1}}{i : A}\;R_{\mathrm{dn}.i} \qquad \dfrac{\dfrac{i+1 : T(\text{“}A\text{”}) \wedge T(\text{“}B\text{”})}{i+1 : T(\text{“}B\text{”})}\;\wedge\mathrm{E}_{i+1}}{i : B}\;R_{\mathrm{dn}.i}}{i : A \wedge B}\;\wedge\mathrm{I}_i}{i+1 : T(\text{“}A \wedge B\text{”})}\;R_{\mathrm{up}.i}}{i+1 : T(\text{“}A\text{”}) \wedge T(\text{“}B\text{”}) \supset T(\text{“}A \wedge B\text{”})}\;\supset\mathrm{I}_{i+1}$$

3.

$$\dfrac{\dfrac{i+1 : T(\text{“}A\text{”}) \vee T(\text{“}B\text{”}) \qquad \dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A\text{”})}{i : A}\;R_{\mathrm{dn}.i}}{i : A \vee B}\;\vee\mathrm{I}_i}{i+1 : T(\text{“}A \vee B\text{”})}\;R_{\mathrm{up}.i} \qquad \dfrac{\dfrac{\dfrac{i+1 : T(\text{“}B\text{”})}{i : B}\;R_{\mathrm{dn}.i}}{i : A \vee B}\;\vee\mathrm{I}_i}{i+1 : T(\text{“}A \vee B\text{”})}\;R_{\mathrm{up}.i}}{i+1 : T(\text{“}A \vee B\text{”})}\;\vee\mathrm{E}_{i+1}}{i+1 : T(\text{“}A\text{”}) \vee T(\text{“}B\text{”}) \supset T(\text{“}A \vee B\text{”})}\;\supset\mathrm{I}_{i+1}$$

4.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A \vee B\text{”})}{i : A \vee B}\;R_{\mathrm{dn}.i} \qquad \dfrac{i : A \qquad \dfrac{\dfrac{i+1 : T(\text{“}A \supset C\text{”}) \wedge T(\text{“}B \supset C\text{”})}{i+1 : T(\text{“}A \supset C\text{”})}\;\wedge\mathrm{E}_{i+1}}{i : A \supset C}\;R_{\mathrm{dn}.i}}{i : C}\;\supset\mathrm{E}_i \qquad \dfrac{i : B \qquad \dfrac{\dfrac{i+1 : T(\text{“}A \supset C\text{”}) \wedge T(\text{“}B \supset C\text{”})}{i+1 : T(\text{“}A \supset C\text{”})}\;\wedge\mathrm{E}_{i+1}}{i : B \supset C}\;R_{\mathrm{dn}.i}}{i : C}\;\supset\mathrm{E}_i}{i : C}\;\vee\mathrm{E}_i}{i+1 : T(\text{“}C\text{”})}\;R_{\mathrm{up}.i}}{i+1 : T(\text{“}A \supset C\text{”}) \wedge T(\text{“}B \supset C\text{”}) \supset T(\text{“}C\text{”})}\;\supset\mathrm{I}_{i+1}}{i+1 : T(\text{“}A \vee B\text{”}) \supset (T(\text{“}A \supset C\text{”}) \wedge T(\text{“}B \supset C\text{”}) \supset T(\text{“}C\text{”}))}\;\supset\mathrm{I}_{i+1}$$

5.

$$\dfrac{\dfrac{\dfrac{\dfrac{i+1 : T(\text{“}\bot\text{”})}{i : \bot}\;R_{\mathrm{dn}.i}}{i : A}\;\bot_i}{i+1 : T(\text{“}A\text{”})}\;R_{\mathrm{up}.i}}{i+1 : T(\text{“}\bot\text{”}) \supset T(\text{“}A\text{”})}\;\supset\mathrm{I}_{i+1}$$

6.

$$\dfrac{\dfrac{\dfrac{\dfrac{i+1 : \neg T(\text{“}\bot\text{”}) \qquad \dfrac{\dfrac{\dfrac{i+1 : T(\text{“}A\text{”})}{i : A}\;R_{\mathrm{dn}.i} \qquad \dfrac{i+1 : T(\text{“}\neg A\text{”})}{i : \neg A}\;R_{\mathrm{dn}.i}}{i : \bot}\;\supset\mathrm{E}_i}{i+1 : T(\text{“}\bot\text{”})}\;R_{\mathrm{up}.i}}{i+1 : \bot}\;\supset\mathrm{E}_{i+1}}{i+1 : \neg T(\text{“}\neg A\text{”})}\;\bot_{i+1}}{i+1 : T(\text{“}A\text{”}) \supset \neg T(\text{“}\neg A\text{”})}\;\supset\mathrm{I}_{i+1}}{i+1 : \neg T(\text{“}\bot\text{”}) \supset (T(\text{“}A\text{”}) \supset \neg T(\text{“}\neg A\text{”}))}\;\supset\mathrm{I}_{i+1}$$

□

PROPOSITION 55. *Let $A$, $B$ and $C$ be $L_i$-wffs. Then:*

1. $i+1 : T(\text{“}A\text{”}) \vee T(\text{“}B\text{”}), i+1 : T(\text{“}A \supset C\text{”}), i+1 : T(\text{“}B \supset C\text{”}) \vdash_{MK} i : C$

2. $i+1 : T(``A") \lor T(``B"), i : A \supset C, i : B \supset C \vdash_{MK} i : C$

3. $i+1 : \neg T(``A") \supset T(``B"), i : \neg A \vdash_{MK} i : B$

*Proof.* 1

$$\dfrac{\dfrac{i+1 : T(``A") \lor T(``B") \quad \dfrac{\dfrac{\dfrac{i+1:T(``A")}{i:A}\,R_{\mathrm{dn}.i} \quad \dfrac{i+1:T(``A \supset C")}{i:A \supset C}\,R_{\mathrm{dn}.i}}{i:C}\,\supset\mathrm{E}_i}{i+1:T(``C")}\,R_{\mathrm{up}.i} \quad \dfrac{\dfrac{\dfrac{i+1:T(``B")}{i:B}\,R_{\mathrm{dn}.i} \quad \dfrac{i+1:T(``B \supset C")}{i:B \supset C}\,R_{\mathrm{dn}.i}}{i:C}\,\supset\mathrm{E}_i}{i+1:T(``C")}\,R_{\mathrm{up}.i}}{i+1:T(``C")}\,\lor\mathrm{E}_{i+1}}{i:C}\,R_{\mathrm{dn}.i}$$

2

$$\dfrac{\dfrac{\dfrac{i+1 : T(``A") \lor T(``B") \quad \dfrac{\dfrac{\dfrac{i+1:T(``A")}{i:A}\,R_{\mathrm{dn}.i}}{i:A \lor B}\,\lor\mathrm{I}_i}{i+1:T(``A \lor B")}\,R_{\mathrm{up}.i} \quad \dfrac{\dfrac{\dfrac{i+1:T(``B")}{i:B}\,R_{\mathrm{dn}.i}}{i:A \lor B}\,\lor\mathrm{I}_i}{i+1:T(``A \lor B")}\,R_{\mathrm{up}.i}}{i+1:T(``A \lor B")}\,\lor\mathrm{E}_{i+1}}{i:A \lor B}\,R_{\mathrm{dn}.i} \quad \dfrac{i:A \quad i:A \supset C}{i:C}\,\supset\mathrm{E}_i \quad \dfrac{i:B \quad i:B \supset C}{i:C}\,\supset\mathrm{E}_i}{i:C}\,\lor\mathrm{E}_i$$

3

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{i+1:\neg T(``A") \quad i+1:\neg T(``A") \supset T(``B")}{i+1:T(``B")}\,\supset\mathrm{E}_{i+1}}{i:B}\,R_{\mathrm{dn}.i}}{i:\neg A \supset B}\,\supset\mathrm{I}_i}{i+1:T(``\neg A \supset B")}\,R_{\mathrm{up}.i} \quad i+1:\neg T(``\neg A \supset B")}{i+1:\bot}\,\supset\mathrm{E}_{i+1}}{i+1:T(``A")}\,\bot_{i+1}}{i:A}\,R_{\mathrm{dn}.I} \quad i:\neg A}{i:\bot}\,\supset\mathrm{E}_1}{i:B}\,\bot_i}{i:\neg A \supset B}\,\supset\mathrm{I}_i}{i+1:T(``\neg A \supset B")}\,R_{\mathrm{up}.i} \quad i+1:\neg T(``\neg A \supset B")}{i+1:\bot}\,\supset\mathrm{E}_{i+1}}{i+1:T(``\neg A \supset B")}\,\bot_{i+1}}{i:\neg A \supset B}\,R_{\mathrm{dn}.i} \quad i:\neg A}{i:B}\,\supset\mathrm{E}_i}{}$$

□

## C. Normal Form for MK

The reduction steps for the connectives described in (Prawitz, 1965) are the following:

$\wedge$*-reduction:*

$$\begin{array}{c} \Pi_1 \qquad \Pi_2 \\ \dfrac{\dfrac{j:C \quad j:D}{j:C\wedge D}\wedge\mathrm{I}_j}{j:C^\bullet}\wedge\mathrm{E}_j \\ \Pi_3 \end{array} \qquad\qquad \begin{array}{c} \Pi_1 \\ \{j:C\} \\ \Pi_3 \end{array} \tag{25}$$

$\Pi'$ is a deduction of $i : A$ from $\Gamma$. Indeed $j : C$ in $\Pi'$ depends on a subset of the assumptions of $j : C^\bullet$ in $\Pi$.

$\vee$ *-reduction:*

$$\begin{array}{c} \dfrac{\begin{array}{c}\Pi_1 \\ \dfrac{j:C}{j:C\vee D}\vee\mathrm{I}_j\end{array} \quad \begin{array}{c}[j:C] \\ \Pi_2 \\ j:E\end{array} \quad \begin{array}{c}[j:D] \\ \Pi_3 \\ j:E\end{array}}{j:E^\bullet}\vee\mathrm{E}_j \\ \Pi_4 \end{array} \qquad\qquad \begin{array}{c} \Pi_1 \\ (j:C) \\ \Pi_2 \\ \{j:E\} \\ \Pi_4 \end{array} \tag{26}$$

$\Pi'$ is a deduction of $i : A$ from $\Gamma$. Indeed by Lemma 22, $j : E$ in $\Pi'$ depends on a subset of the assumptions of $j : E^\bullet$ in $\Pi$. The symmetrical versions of the $\wedge$-reduction and of the $\vee$-reduction, (when the consequence of $\wedge\mathrm{E}_i$ is $i : D$ and the premiss of the $\vee\mathrm{I}_i$ is $i : D$ respectively) are defined analogously.

$\supset$ *-reduction:*

$$\begin{array}{c} \dfrac{\begin{array}{c}[j:C] \\ \Pi_1 \\ \dfrac{j:D}{j:C\supset D}\supset\mathrm{I}_j\end{array} \quad \begin{array}{c}\Pi_2 \\ j:C\end{array}}{j:D^\bullet}\supset\mathrm{E}_j \\ \Pi_3 \end{array} \qquad\qquad \begin{array}{c} \Pi_2 \\ (j:C) \\ \Pi_1 \\ \{j:D\} \\ \Pi_3 \end{array} \tag{27}$$

$\Pi'$ is a deduction of $i : A$ from $\Gamma$. Indeed by Lemma 22 the occurrence $j : D$ in $\Pi'$, depends on a subset of the assumptions of $j : D^\bullet$ in $\Pi$.

$\frac{\perp}{\wedge}$-reduction

$$\begin{array}{c} j:\neg(C\wedge D) \\ \Pi_1 \\ \dfrac{j:\perp}{j:C\wedge D} \\ \Pi_2 \end{array} \qquad \begin{array}{c} \dfrac{\begin{array}{c} \dfrac{\dfrac{j:\neg C \quad \dfrac{j:C\wedge D}{j:C}\wedge\mathrm{E}_j}{j:\perp}\supset\mathrm{E}_j}{(j:\neg(C\wedge D))}\supset\mathrm{I}_j \\ \Pi_1 \\ \dfrac{j:\perp}{i:C}\perp_i \end{array} \quad \begin{array}{c} \dfrac{\dfrac{j:\neg D \quad \dfrac{j:C\wedge D}{j:D}\wedge\mathrm{E}_j}{j:\perp}\supset\mathrm{E}_j}{(j:\neg(C\wedge D))}\supset\mathrm{I}_j \\ \Pi_1 \\ \dfrac{j:\perp}{i:D}\perp_i \end{array}}{\{j:C\wedge D\}}\wedge\mathrm{I}_j \\ \Pi_2 \end{array} \tag{28}$$

$\frac{\perp}{\supset}$-reduction

$$
\begin{array}{c}
j:\neg(C\supset D)\\
\Pi_1\\
\dfrac{j:\perp}{j:C\supset D}\\
\Pi_2
\end{array}
\qquad\qquad
\begin{array}{c}
\dfrac{\dfrac{\dfrac{j:C \quad j:C\supset D}{j:D} \qquad j:\neg D}{j:\perp}}{(j:\neg(C\supset D))}\\
\Pi_1\\
\dfrac{\dfrac{j:\perp}{j:D}}{j:C\supset D}\\
\Pi_2
\end{array}
\tag{29}
$$

## D. From BM-logic to MMCC

Proof of the $\Delta$-axiom schema of Buvač and Masons's propositional logics of contexts:

$(\Delta)\quad ist(\kappa_1, \text{“}ist(\kappa_2, \text{“}A\text{”}) \vee B\text{”}) \supset ist(\kappa_1, \text{“}ist(\kappa_2, \text{“}A\text{”})\text{”}) \vee ist(\kappa_1, \text{“}B\text{”})$

In the following proof $Prem(\Delta)$ and $Cons(\Delta)$ denote the premise and the consequence of $\Delta$.

$$
\begin{prooftree}
\AxiomC{$\overline{\kappa}:Prem(\Delta)$}
\RightLabel{$R_{\mathrm{dn}.\overline{\kappa}\kappa_1}$}
\UnaryInfC{$\overline{\kappa}\kappa_1: ist(\kappa_2,A)\vee B$}
\AxiomC{$\overline{\kappa}\kappa_1:\neg B$}
\doubleLine
\BinaryInfC{$\overline{\kappa}\kappa_1: ist(\kappa_2,A)$}
\RightLabel{$R_{\mathrm{up}.\overline{\kappa}\kappa_1}$}
\UnaryInfC{$\overline{\kappa}: ist(\kappa_1, ist(\kappa_2,A))$}
\RightLabel{$\vee\mathrm{I}_{\overline{\kappa}}$}
\UnaryInfC{$\overline{\kappa}: Cons(\Delta)$}
\AxiomC{$\overline{\kappa}:\neg Cons(\Delta)$}
\RightLabel{$\supset\mathrm{E}_{\overline{\kappa}}$}
\BinaryInfC{$\overline{\kappa}:\perp$}
\RightLabel{$\perp$}
\UnaryInfC{$\overline{\kappa}: ist(\kappa_1,\perp)$}
\RightLabel{$R_{\mathrm{dn}.\overline{\kappa}\kappa_1}$}
\UnaryInfC{$\overline{\kappa}\kappa_1:\perp$}
\RightLabel{$\perp$}
\UnaryInfC{$\overline{\kappa}\kappa_1: B$}
\RightLabel{$R_{\mathrm{up}.\overline{\kappa}\kappa_1}$}
\UnaryInfC{$\overline{\kappa}: ist(\kappa_1,B)$}
\RightLabel{$\vee\mathrm{I}_{\overline{\kappa}}$}
\UnaryInfC{$\overline{\kappa}: Cons(\Delta)$}
\AxiomC{$\overline{\kappa}:\neg Cons(\Delta)$}
\RightLabel{$\supset\mathrm{E}_{\overline{\kappa}}$}
\BinaryInfC{$\overline{\kappa}:\perp$}
\RightLabel{$\perp$}
\UnaryInfC{$\overline{\kappa}: Cons(\Delta)$}
\end{prooftree}
$$