



HAL
open science

A Sequent Calculus and Theorem Prover for Standard Conditional Logics

Nicolas Olivetti, Gian Luca Pozzato, Camilla Schwind

► **To cite this version:**

Nicolas Olivetti, Gian Luca Pozzato, Camilla Schwind. A Sequent Calculus and Theorem Prover for Standard Conditional Logics. The Computing Research Repository, 2004. halshs-00261169

HAL Id: halshs-00261169

<https://shs.hal.science/halshs-00261169>

Submitted on 26 Mar 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Sequent Calculus and a Theorem Prover for Standard Conditional Logics

NICOLA OLIVETTI and GIAN LUCA POZZATO

Università di Torino

and

CAMILLA B. SCHWIND

École d'Architecture de Marseille

In this paper we present a cut-free sequent calculus, called SeqS, for some standard conditional logics, namely CK, CK+ID, CK+MP and CK+MP+ID. The calculus uses labels and transition formulas and can be used to prove decidability and space complexity bounds for the respective logics. We also present CondLean, a theorem prover for these logics implementing SeqS calculi written in SICStus Prolog.

Categories and Subject Descriptors: D.1.6 [Programming Techniques]: Logic Programming; F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—*Computational Logic; Logic and Constraint Programming; Proof Theory*; I.2.3 [Artificial Intelligence]: Deduction and Theorem Proving—*Deduction; Logic Programming*

General Terms: Languages, Logic, Theory

Additional Key Words and Phrases: Analytic Sequent Calculi, Automated Deduction, Conditional Logics, Labelled Deductive Systems, Logic Programming, Proof Theory

1. INTRODUCTION

Conditional logics have a long history. They have been studied first by Lewis ([Lewis 1973; Nute 1980; Chellas 1975; Stalnaker 1968]) in order to formalize a kind of hypothetical reasoning (if A were the case then B), that cannot be captured by classical logic with material implication.

In the last years, interesting applications of conditional logic to several domains of artificial intelligence such as knowledge representation, non-monotonic reasoning, belief revision, representation of counterfactual sentences, deductive databases have

Authors' addresses: N. Olivetti, Dipartimento di Informatica - Università degli Studi di Torino, corso Svizzera 185 - 10149 Turin - Italy, e-mail: olivetti@di.unito.it.

G.L. Pozzato, Dipartimento di Informatica - Università degli Studi di Torino, corso Svizzera 185 - 10149 Turin - Italy, e-mail: pozzato@di.unito.it.

C.B. Schwind, École d'Architecture de Marseille - Luminy, 184 avenue de Luminy - 13288 Marseille cedex 9 - France,

e-mail: Camilla.Schwind@map.archi.fr

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 1529-3785/20YY/0700-0001 \$5.00

been proposed ([Crocco et al. 1995]). For instance, in [Grahne 1998] knowledge and database update is formalized by some conditional logic. Conditional logics have also been used to modelize belief revision ([Gardenfors and Rott 1995; Lindstrom and Rabinowicz 1992; Giordano et al. 1998; 2002]). Conditional logics can provide an axiomatic foundation of non-monotonic reasoning ([Kraus et al. 1990]), as it turns out that all forms of inference studied in the framework of non-monotonic (preferential) logics are particular cases of conditional axioms ([Crocco and Lamarre 1992]). Causal inference, which is very important for applications in action planning ([Schwind 1999]), has been modelled by conditional logics ([Giordano and Schwind 2004]). Conditional Logics have been used to model hypothetical queries in deductive databases and logic programming; the conditional logic CK+ID is the basis of the logic programming language defined in [Gabbay et al. 2000]. In system diagnosis, conditional logics can be used to reason hypothetically about the expected functioning of system components with respect to the observed faults. [Obeid 2001] introduces a conditional logic, DL, suitable for diagnostic reasoning and which allows to represent and reason with assumptions in model-based diagnosis. Another interesting application of conditional logics is the formalization of *prototypical reasoning*, that is to say reasoning about typical properties and exceptions. Delgrande in [Delgrande 1987] proposes a conditional logic for prototypical reasoning.

Finally, an obvious application concerns natural language semantics where conditional logics are used in order to give a formal treatment of hypothetical and counterfactual sentences as presented in [Nute 1980]. A broader discussion about counterfactuals can be found in [Costello and McCarthy 1999].

In spite of their significance, very few proof systems have been proposed for conditional logics: we just mention [Lamarre 1993; Delgrande and Groeneboer 1990; Crocco and del Cerro 1995; Artosi et al. 2002; Gent 1992; de Swart 1983; Giordano et al. 2003]. One possible reason of the underdevelopment of proof-methods for conditional logics is the lack of a universally accepted semantics for them. This is in sharp contrast to modal and temporal logics which have a consolidated semantics based on a standard kind of Kripke structures.

Similarly to modal logics, the semantics of conditional logics can be defined in terms of possible world structures. In this respect, conditional logics can be seen as a generalization of modal logics (or a type of multi-modal logic) where the conditional operator is a sort of modality indexed by a formula of the same language.

The two most popular semantics for conditional logics are the so-called *sphere semantics* ([Lewis 1973]) and the *selection function semantics* ([Nute 1980]). Both are possible-world semantics, but are based on different (though related) algebraic notions. Here we adopt the selection function semantics, which is more general than the sphere semantics.

Since we adopt the selection function semantics, **CK** is the fundamental system; it has the same role as the system K (from which it derives its name) in modal logic: CK-valid formulas are formulas that are valid in every selection function model.

In this work we present a sequent calculus for CK and for three standard extensions of it, namely CK+ID, CK+MP¹ and CK+MP+ID. This calculus makes use of labels, following the line of [Viganò 2000] and [Gabbay 1996]. To the best of our

¹This conditional system is related to modal logic T.

knowledge, this is the first calculus for these systems. Some tableaux calculi were developed in [Giordano et al. 2003] and in [Olivetti and Schwind 2000] for other more specific conditional systems.

Our goal is to obtain a decision procedure for the logics under consideration. For this reason, we undertake a proof theoretical analysis of our calculi. In order to get a terminating calculus, it is crucial to control the application of the *contraction rule*. This rule allows for duplicating a formula in a backward proof search and thus is a potential source of an infinite expansion of a branch. Generally speaking, the status of the contraction rules varies for different logical systems: in some cases, contraction rules can be just removed without losing completeness, in some others they cannot be eliminated, but their application can be controlled in such a way that the branch expansion terminates. This is what happens also for our conditional logics. In other cases, contraction rules can be eliminated, but at the price of changing the logical rules, as it happens in [Hudelmaier 1993].

In this work, we show that the contraction rules can be eliminated in the calculi for CK and CK+ID. In this way, the calculus not only provides a decision procedure, but it can also be used to establish a complexity bound for these logics (the decidability for these logics has been shown in [Nute 1980]). Roughly speaking, if the rules are analytic, the length of each branch is bounded essentially by the length of the initial sequent; therefore, we can easily obtain that the calculus give a polynomial space complexity.

For CK+MP and CK+MP+ID the situation is different: contraction rules cannot be eliminated without losing completeness. However, we show that they can be used in a controlled way, namely it is necessary to apply the contraction at most one time on each conditional formula of the form $x : A \Rightarrow B$ in every branch of a proof tree. This is sufficient to obtain a decision procedure for these logics.

It is worth noting that the elimination of contractions is connected with a remarkable property, the so-called *disjunction property* for conditional formulas: if $(A_1 \Rightarrow B_1) \vee (A_2 \Rightarrow B_2)$ is valid, then either $(A_1 \Rightarrow B_1)$ or $(A_2 \Rightarrow B_2)$ is valid too.

As a difference with modal logics, for which there are lots of efficient implementations ([Beckert and Posegga 1995], [Fitting 1998], [Beckert and Gorè 1997]), to the best of our knowledge very few theorem provers have been implemented for conditional logics ([Lamarre 1993] and [Artosi et al. 2002]). We present here a simple implementation of our sequent calculi, called **CondLean**; it is a Prolog program which follows the *lean* methodology ([Beckert and Posegga 1995], [Fitting 1998]), in which every clause of a predicate `prove` implements an axiom or rule of the calculus and the proof search is provided for free by the mere depth-first search mechanism of Prolog, without any ad hoc mechanism. We also present an alternative version of our theorem prover inspired by the tableau calculi for modal logics introduced in [Beckert and Gorè 1997].

The plan of the paper is as follows: in section 2 we introduce the conditional systems we consider, in section 3 we present the sequent calculi for conditional systems above. In section 4 we analyze the calculi in order to obtain a decision procedure for the basic conditional system, CK, and for the three mentioned extensions of it. In section 5 we present the theorem prover CondLean. In section 6 we discuss some related work.

2. CONDITIONAL LOGICS

Conditional logics are extensions of classical logic obtained by adding the conditional operator \Rightarrow . In this paper, we only consider propositional conditional logics.

A propositional conditional language \mathcal{L} contains the following items:

- a set of propositional variables ATM ;
- the symbol of *false* \perp ;
- a set of connectives² \rightarrow, \Rightarrow .

We define formulas of \mathcal{L} as follows:

- \perp and the propositional variables of ATM are *atomic formulas*;
- if A and B are formulas, $A \rightarrow B$ and $A \Rightarrow B$ are *complex formulas*.

We adopt the *selection function semantics*. We consider a non-empty set of possible worlds \mathcal{W} . Intuitively, the selection function f selects, for a world w and a formula A , the set of worlds of \mathcal{W} which are *closer* to w given the information A . A conditional formula $A \Rightarrow B$ holds in a world w if the formula B holds in *all the worlds selected by f for w and A* .

A model is a triple:

$$\mathcal{M} = \langle \mathcal{W}, f, [\] \rangle$$

where:

- \mathcal{W} is a non empty set of items called *worlds*;
- f is the so-called *selection function* and has the following type:

$$f: \mathcal{W} \times 2^{\mathcal{W}} \longrightarrow 2^{\mathcal{W}}$$

- $[\]$ is the *evaluation function*, which assigns to an atom $P \in ATM$ the set of worlds where P is true, and is extended to the other formulas as follows:

- * $[\perp] = \emptyset$
- * $[A \rightarrow B] = (\mathcal{W} - [A]) \cup [B]$
- * $[A \Rightarrow B] = \{w \in \mathcal{W} \mid f(w, [A]) \subseteq [B]\}$

Observe that we have defined f taking $[A]$ rather than A (i.e. $f(w, [A])$ rather than $f(w, A)$) as argument; this is equivalent to define f on formulas, i.e. $f(w, A)$ but imposing that if $[A] = [A']$ in the model, then $f(w, A) = f(w, A')$. This condition is called *normality*.

The semantics above characterizes the *basic conditional system*, called **CK**. An axiomatization of the CK system is given by:

- all tautologies of classical propositional logic.

- (Modus Ponens)
$$\frac{A \quad A \rightarrow B}{B}$$

- (RCEA)
$$\frac{A \leftrightarrow B}{(A \Rightarrow C) \leftrightarrow (B \Rightarrow C)}$$

²The usual connectives \top, \wedge, \vee and \neg can be defined in terms of \perp and \rightarrow .

$$- \text{ (RCK) } \frac{(A_1 \wedge \dots \wedge A_n) \rightarrow B}{(C \Rightarrow A_1 \wedge \dots \wedge C \Rightarrow A_n) \rightarrow (C \Rightarrow B)}$$

Other conditional systems are obtained by assuming further properties on the selection function; we consider the following three standard extensions of the basic system CK:

| System | Axioms | Model condition |
|---------------------|---|---|
| CK + ID | $A \Rightarrow A$ | $f(x, [A]) \subseteq [A]$ |
| CK + MP | $(A \Rightarrow B) \rightarrow (A \rightarrow B)$ | $w \in [A] \rightarrow w \in f(w, [A])$ |
| CK + MP + ID | $(A \Rightarrow B) \rightarrow (A \rightarrow B),$ $A \Rightarrow A$ | $w \in [A] \rightarrow w \in f(w, [A]),$ $f(x, [A]) \subseteq [A]$ |

3. A SEQUENT CALCULUS FOR CONDITIONAL LOGICS

In this section we present **SeqS**, a sequent calculus for the conditional systems introduced above. S stands for {CK, ID, MP, ID+MP}; the calculi make use of labels to represent possible worlds.

We consider a conditional language \mathcal{L} and a denumerable alphabet of labels \mathcal{A} , whose elements are denoted by x, y, z, \dots

There are two kinds of formulas:

- (1) *labelled formulas*, denoted by $x: A$, where $x \in \mathcal{A}$ and $A \in \mathcal{L}$, used to represent that A holds in a world x ;
- (2) *transition formulas*, denoted by $x \xrightarrow{A} y$, where $x, y \in \mathcal{A}$ and $A \in \mathcal{L}$. A transition formula $x \xrightarrow{A} y$ represents that $y \in f(x, [A])$.

A **sequent** is a pair $\langle \Gamma, \Delta \rangle$, usually denoted with $\Gamma \vdash \Delta$, where Γ and Δ are multisets of formulas. The intuitive meaning of $\Gamma \vdash \Delta$ is: every model that satisfies all labelled formulas of Γ in the respective worlds (specified by the labels) satisfies at least one of the labelled formulas of Δ (in those worlds). This is made precise by the notion of *validity* of a sequent given in the next definition:

DEFINITION 3.1 SEQUENT VALIDITY. *Given a model*

$$\mathcal{M} = \langle \mathcal{W}, f, [] \rangle$$

for \mathcal{L} , and a label alphabet \mathcal{A} , we consider any mapping

$$I : \mathcal{A} \rightarrow \mathcal{W}$$

Let F be a labelled formula, we define $\mathcal{M} \models_I F$ as follows:

- $\mathcal{M} \models_I x: A$ iff $I(x) \in [A]$
- $\mathcal{M} \models_I x \xrightarrow{A} y$ iff $I(y) \in f(I(x), [A])$

We say that $\Gamma \vdash \Delta$ is valid in \mathcal{M} if for every mapping $I : \mathcal{A} \rightarrow \mathcal{W}$, if $\mathcal{M} \models_I F$ for every $F \in \Gamma$, then $\mathcal{M} \models_I G$ for some $G \in \Delta$. We say that $\Gamma \vdash \Delta$ is valid in a system (CK or one of its extensions) if it is valid in every \mathcal{M} satisfying the specific conditions for that system (if any).

In Figure 1 we present the calculi for CK and its mentioned extensions.

| | |
|--|---|
| $\text{(AX)} \quad \Gamma, F \vdash \Delta, F$ | $\text{(A}\perp\text{)} \quad \Gamma, x : \perp \vdash \Delta$ |
| $\text{(WeakL)} \quad \frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}$ | $\text{(WeakR)} \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, F}$ |
| $\text{(ContrL)} \quad \frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}$ | $\text{(ContrR)} \quad \frac{\Gamma \vdash \Delta, F, F}{\Gamma \vdash \Delta, F}$ |
| $\text{(}\rightarrow\text{R)} \quad \frac{\Gamma, x : A \vdash x : B, \Delta}{\Gamma \vdash x : A \rightarrow B, \Delta}$ | $\text{(}\rightarrow\text{L)} \quad \frac{\Gamma \vdash x : A, \Delta \quad \Gamma, x : B \vdash \Delta}{\Gamma, x : A \rightarrow B \vdash \Delta}$ |
| $\text{(EQ)} \quad \frac{u : A \vdash u : B \quad u : B \vdash u : A}{\Gamma, x \xrightarrow{A} y \vdash x \xrightarrow{B} y, \Delta}$ | |
| $\text{(}\Rightarrow\text{L)} \quad \frac{\Gamma \vdash x \xrightarrow{A} y, \Delta \quad \Gamma, y : B \vdash \Delta}{\Gamma, x : A \Rightarrow B \vdash \Delta}$ | $\text{(}\Rightarrow\text{R)} \quad \frac{\Gamma, x \xrightarrow{A} y \vdash y : B, \Delta}{\Gamma \vdash x : A \Rightarrow B, \Delta} \quad (y \notin \Gamma, \Delta)$ |
| $\text{(ID)} \quad \frac{\Gamma, y : A \vdash \Delta}{\Gamma, x \xrightarrow{A} y \vdash \Delta}$ | $\text{(MP)} \quad \frac{\Gamma \vdash x : A, \Delta}{\Gamma \vdash x \xrightarrow{A} x, \Delta}$ |

Fig. 1. Sequent calculi SeqS; the (ID) rule is for SeqID and SeqID+MP only; the (MP) rule is for SeqMP and SeqID+MP only.

| | |
|--|--|
| $\text{(}\wedge\text{L)} \quad \frac{\Gamma, x : A, x : B \vdash \Delta}{\Gamma, x : A \wedge B \vdash \Delta}$ | $\text{(}\wedge\text{R)} \quad \frac{\Gamma \vdash \Delta, x : A \quad \Gamma \vdash \Delta, x : B}{\Gamma \vdash \Delta, x : A \wedge B}$ |
| $\text{(}\vee\text{L)} \quad \frac{\Gamma, x : A \vdash \Delta \quad \Gamma, x : B \vdash \Delta}{\Gamma, x : A \vee B \vdash \Delta}$ | $\text{(}\vee\text{R)} \quad \frac{\Gamma \vdash \Delta, x : A, x : B}{\Gamma \vdash \Delta, x : A \vee B}$ |
| $\text{(}\neg\text{L)} \quad \frac{\Gamma \vdash \Delta, x : A}{\Gamma, x : \neg A \vdash \Delta}$ | $\text{(}\neg\text{R)} \quad \frac{\Gamma, x : A \vdash \Delta}{\Gamma \vdash \Delta, x : \neg A}$ |
| $\text{(AT)} \quad \Gamma \vdash \Delta, x : \top$ | |

Fig. 2. Additional axioms and rules in SeqS for the other boolean operators, derived from the rules in Figure 1 by the usual equivalences.

EXAMPLE 3.2. *We show a derivation of the (ID) axiom.*

$$\frac{\frac{y : A \vdash y : A}{\Gamma \vdash y : A} \text{(ID)}}{\frac{x \xrightarrow{A} y \vdash y : A}{\vdash x : A \Rightarrow A} \text{(}\Rightarrow\text{R)}}$$

EXAMPLE 3.3. We show a derivation of the (MP) axiom.

$$\begin{array}{c}
\frac{x : A \vdash x : A, x : B}{x : A \vdash x : A, x : B} \text{ (MP)} \\
\frac{x : A \vdash x \xrightarrow{A} x, x : B \quad x : A, x : B \vdash x : B}{x : A \Rightarrow B, x : A \vdash x : B} (\Rightarrow L) \\
\frac{x : A \Rightarrow B, x : A \vdash x : B}{x : A \Rightarrow B \vdash x : A \rightarrow B} (\rightarrow R) \\
\frac{x : A \Rightarrow B \vdash x : A \rightarrow B}{\vdash x : (A \Rightarrow B) \rightarrow (A \rightarrow B)} (\rightarrow R)
\end{array}$$

In the following, we will need to consider the *permutability* of a rule over another one³. It is easy to observe the following Lemma:

LEMMA 3.4 PERMUTABILITY OF THE RULES. *All the SeqS's rules permute over the other rules, with the exception of ($\Rightarrow L$), which does not always permute over ($\Rightarrow R$).*

In particular, it does not permute over ($\Rightarrow R$) which introduces the label y used to apply ($\Rightarrow L$), as shown in the following counterexample in SeqCK:

$$\begin{array}{c}
\frac{y : B, y : C, x \xrightarrow{A} y \vdash y : B}{y : B \wedge C, x \xrightarrow{A} y \vdash y : B} (\wedge L) \\
\frac{x \xrightarrow{A} y \vdash x \xrightarrow{A} y, y : B \quad y : B \wedge C, x \xrightarrow{A} y \vdash y : B}{x : A \Rightarrow (B \wedge C), x \xrightarrow{A} y \vdash y : B} (\Rightarrow L) \\
\frac{x : A \Rightarrow (B \wedge C), x \xrightarrow{A} y \vdash y : B}{x : A \Rightarrow (B \wedge C) \vdash x : A \Rightarrow B} (\Rightarrow R)
\end{array}$$

The application of ($\Rightarrow L$) with x , the only available label, leads to a proof which fails:

$$\begin{array}{c}
\frac{x \xrightarrow{A} y \vdash y : B, x \xrightarrow{A} x}{\vdash x : A \Rightarrow B, x \xrightarrow{A} x} (\Rightarrow R) \quad \frac{x : B, x : C, x \xrightarrow{A} y \vdash y : B}{x : B, x : C \vdash x : A \Rightarrow B} (\Rightarrow R) \\
\frac{\vdash x : A \Rightarrow B, x \xrightarrow{A} x \quad x : B \wedge C \vdash x : A \Rightarrow B}{x : A \Rightarrow (B \wedge C) \vdash x : A \Rightarrow B} (\wedge L) \\
\frac{\vdash x : A \Rightarrow B, x \xrightarrow{A} x \quad x : B \wedge C \vdash x : A \Rightarrow B}{x : A \Rightarrow (B \wedge C) \vdash x : A \Rightarrow B} (\Rightarrow L)
\end{array}$$

If A , B and C are atoms, neither of the two branches can be closed.

The sequent calculus SeqS is sound and complete with respect to the semantics.

THEOREM 3.5 SOUNDNESS. *If $\Gamma \vdash \Delta$ is derivable in SeqS then it is valid in the corresponding system.*

Proof. By induction on the height of a derivation of $\Gamma \vdash \Delta$. As an example, we examine the cases of ($\Rightarrow R$) and (MP). The other cases are left to the reader.

- ($\Rightarrow R$) Let $\Gamma \vdash \Delta, x : A \Rightarrow B$ be derived from (1) $\Gamma, x \xrightarrow{A} y \vdash \Delta, y : B$, where y does not occur in Γ, Δ and it is different from x . By induction hypothesis

³In general, we say that a rule r_1 permutes over a rule r_2 if the following condition holds: if they are both applicable to a sequent $\Gamma \vdash \Delta$ and there is a proof tree where r_2 is applied to $\Gamma \vdash \Delta$ and r_1 is applied to one of the premises of r_2 , then there exists a proof tree of $\Gamma \vdash \Delta$ where r_1 is applied to the sequent $\Gamma \vdash \Delta$ and r_2 is applied to one of the premises of r_1 .

we know that the latter sequent is valid. Suppose the former is not, and that it is not valid in a model $\mathcal{M} = \langle \mathcal{W}, f, [\] \rangle$, via a mapping I , so that we have:

$$\mathcal{M} \models_I F \text{ for every } F \in \Gamma, \mathcal{M} \not\models_I F \text{ for any } F \in \Delta \text{ and } M \not\models_I x : A \Rightarrow B.$$

As $M \not\models_I x : A \Rightarrow B$ there exists $w \in f(I(x), [A]) - [B]$. We can define an interpretation $I'(z) = I(z)$ for $z \neq y$ and $I'(y) = w$. Since y does not occur in Γ, Δ and is different from x , we have that $\mathcal{M} \models_{I'} F$ for every $F \in \Gamma$, $\mathcal{M} \not\models_{I'} F$ for any $F \in \Delta$, $\mathcal{M} \not\models_{I'} y : B$ and $\mathcal{M} \models_{I'} x \xrightarrow{A} y$, against the validity of (1).

- (MP) Let $\Gamma \vdash \Delta, x \xrightarrow{A} x$ be derived from (2) $\Gamma \vdash \Delta, x : A$. Let (2) be valid and let $\mathcal{M} = \langle \mathcal{W}, f, [\] \rangle$ be a model satisfying the MP condition. Suppose that for one mapping I , $\mathcal{M} \models_I F$ for every $F \in \Gamma$, then by the validity of (2) either $\mathcal{M} \models_I G$ for some $G \in \Delta$, or $\mathcal{M} \models_I x : A$. In the latter case, we have $I(x) \in [A]$, thus $I(x) \in f(I(x), [A])$, by MP, this means that $\mathcal{M} \models_I x \xrightarrow{A} x$.

□

Completeness is an easy consequence of the admissibility of cut. By cut we mean the following rule:

$$\frac{\Gamma \vdash \Delta, F \quad F, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \text{ (cut)}$$

where F is any labelled formula. To prove cut admissibility, we need the following lemma about label substitution.

LEMMA 3.6. *If a sequent $\Gamma \vdash \Delta$ has a derivation of height h , then $\Gamma[x/y] \vdash \Delta[x/y]$ has a derivation of height h , where $\Gamma[x/y] \vdash \Delta[x/y]$ is the sequent obtained from $\Gamma \vdash \Delta$ by replacing a label x by a label y wherever it occurs.*

Proof. By a straightforward induction on the height of a derivation.

□

THEOREM 3.7 ADMISSIBILITY OF CUT. *If $\Gamma \vdash \Delta, F$ and $F, \Gamma \vdash \Delta$ are derivable, so $\Gamma \vdash \Delta$.*

Proof. As usual, the proof proceeds by a double induction over the complexity of the cut formula and the sum of the heights of the derivations of the two premises of the cut inference, in the sense that we replace one cut by one or several cuts on formulas of smaller complexity, or on sequents derived by shorter derivations. We have several cases: (i) one of the two premises is an axiom, (ii) the last step of one of the two premises is obtained by a rule in which F is *not* the principal formula⁴, (iii) F is the principal formula in the last step of *both* derivations.

(i). If one of the two premises is an axiom then either $\Gamma \vdash \Delta$ is an axiom, or the premise which is not an axiom contains two copies of F and $\Gamma \vdash \Delta$ can be obtained by contraction.

⁴The principal formula of an inference step is the formula introduced by the rule applied in that step.

(ii). We distinguish two cases: the sequent where F is not principal is derived by any rule (R), except the (EQ) rule. This case is standard, we can permute (R) over the cut: i.e. we cut the premise(s) of (R) and then we apply (R) to the result of cut. If one of the sequents, say $\Gamma \vdash \Delta, F$ is obtained by the (EQ) rule, where F is not principal, then also $\Gamma \vdash \Delta$ is derivable by the (EQ) rule and we are done.

(iii). F is the principal formula in both the inferences steps leading to the two cut premises. There are six subcases: F is introduced by (a) a classical rule, (b) by $(\Rightarrow L), (\Rightarrow R)$, (c) by (EQ), (d) F by (ID) on the left and by (EQ) on the right, (e) by (MP) on the left and by (EQ) on the right, (f) by (ID) on the left and by (MP) on the right. The list is exhaustive.

(a). This case is standard and left to the reader.

(b). $F = x : A \Rightarrow B$ is introduced by $(\Rightarrow R)$ and $(\Rightarrow L)$. Then we have

$$\frac{\frac{(*) \Gamma, x \xrightarrow{A} z \vdash z : B, \Delta}{\Gamma \vdash x : A \Rightarrow B, \Delta} (\Rightarrow R) \quad \frac{\Gamma \vdash x \xrightarrow{A} y, \Delta \quad \Gamma, y : B \vdash \Delta}{\Gamma, x : A \Rightarrow B \vdash \Delta} (\Rightarrow L)}{\Gamma \vdash \Delta} (cut)$$

where z does not occur in Γ, Δ and $z \neq x$; By Lemma 3.6, we obtain that $\Gamma, x \xrightarrow{A} y \vdash y : B, \Delta$ is derivable by a derivation of no greater height than (*); thus we can replace the cut as follows

$$\frac{\frac{\Gamma \vdash x \xrightarrow{A} y, \Delta}{\Gamma \vdash x \xrightarrow{A} y, \Delta, y : B} (WeakR) \quad \frac{\Gamma, x \xrightarrow{A} y \vdash y : B, \Delta}{\Gamma \vdash \Delta, y : B} (cut)}{\Gamma \vdash \Delta, y : B} (cut) \quad \frac{\Gamma, y : B \vdash \Delta}{\Gamma \vdash \Delta} (cut)$$

The upper cut uses the induction hypothesis on the height, the lower the induction hypothesis on the complexity of the formula.

(c). $F = x \xrightarrow{B} y$ is introduced by (EQ) in both premises, we have

$$\frac{\frac{(5) u : A \vdash u : B \quad (6) u : B \vdash u : A}{\Gamma', x \xrightarrow{A} y \vdash x \xrightarrow{B} y, \Delta} (EQ) \quad \frac{(7) u : B \vdash u : C \quad (8) u : C \vdash u : B}{\Gamma, x \xrightarrow{B} y \vdash x \xrightarrow{C} y, \Delta'} (EQ)}{\Gamma', x \xrightarrow{A} y \vdash x \xrightarrow{C} y, \Delta'} (cut)$$

where $\Gamma = \Gamma', x \xrightarrow{A} y, \Delta = x \xrightarrow{C} y, \Delta'$. (5)-(8) have been derived by a shorter derivation; thus we can replace the cut by cutting (5) and (7) on the one hand, and (8) and (6) on the other, which give respectively

$$(9) u : A \vdash u : C \text{ and } (10) u : C \vdash u : A.$$

Using (EQ) we obtain $\Gamma', x \xrightarrow{A} y \vdash \Delta', x \xrightarrow{C} y$

(d). $F = x \xrightarrow{B} y$ is introduced on the left by (ID) rule, and it is introduced on

the right by (EQ). Thus we have

$$\frac{\frac{u : A \vdash u : B \quad u : B \vdash u : A}{\Gamma', x \xrightarrow{A} y \vdash \Delta, x \xrightarrow{B} y} (EQ) \quad \frac{\Gamma', x \xrightarrow{A} y, y : B \vdash \Delta}{x \xrightarrow{B} y, \Gamma', x \xrightarrow{A} y \vdash \Delta} (ID)}{\Gamma', x \xrightarrow{A} y \vdash \Delta} (cut)$$

where $\Gamma = \Gamma', x \xrightarrow{A} y$. By Lemma 3.6 and weakening, the sequent $\Gamma', x \xrightarrow{A} y, y : A \vdash y : B, \Delta$ can be derived by a derivation of the same height as $u : A \vdash u : B$. Thus, the cut is replaced as follows

$$\frac{\frac{\Gamma', x \xrightarrow{A} y, y : A \vdash y : B, \Delta \quad \Gamma', x \xrightarrow{A} y, y : B \vdash \Delta}{\Gamma', x \xrightarrow{A} y, y : A \vdash \Delta} (cut)}{\frac{\Gamma', x \xrightarrow{A} y, y : A \vdash \Delta}{\Gamma', x \xrightarrow{A} y, x \xrightarrow{A} y \vdash \Delta} (ID)} (ContrL)$$

$$\frac{\Gamma', x \xrightarrow{A} y, x \xrightarrow{A} y \vdash \Delta}{\Gamma', x \xrightarrow{A} y, \vdash \Delta}$$

(e). $F = x \xrightarrow{A} x$ is introduced on the left by (MP) rule, and it is introduced on the right by (EQ). Thus we have

$$\frac{\frac{\Gamma \vdash x : A, \Delta'}{\Gamma \vdash \Delta', x \xrightarrow{A} x} (MP) \quad \frac{u : A \vdash u : B \quad u : B \vdash u : A}{\Gamma, x \xrightarrow{A} x \vdash \Delta', x \xrightarrow{B} x} (EQ)}{\Gamma \vdash \Delta', x \xrightarrow{B} x} (cut)$$

where $\Delta = \Delta', x \xrightarrow{B} x$. By Lemma 3.6 and weakening, the sequent $\Gamma, x : A \vdash x : B, \Delta'$ can be derived by a derivation of the same height as $u : A \vdash u : B$. Thus the cut is replaced as follows:

$$\frac{\frac{\Gamma \vdash x : A, \Delta' \quad \Gamma, x : A \vdash \Delta', x : B}{\Gamma \vdash \Delta', x : B} (cut)}{\Gamma \vdash \Delta', x \xrightarrow{B} x} (MP)$$

(f). $F = x \xrightarrow{A} x$ is introduced on the right by (MP) rule and on the left by (ID). Thus we have

$$\frac{\frac{\Gamma \vdash x : A, \Delta}{\Gamma \vdash \Delta, x \xrightarrow{A} x} (MP) \quad \frac{\Gamma, x : A \vdash \Delta}{\Gamma, x \xrightarrow{A} x \vdash \Delta} (ID)}{\Gamma \vdash \Delta} (cut)$$

We replace this cut by the following:

$$\frac{\Gamma \vdash x : A, \Delta \quad \Gamma, x : A \vdash \Delta}{\Gamma \vdash \Delta} (cut)$$

THEOREM 3.8 COMPLETENESS. *If A is valid in $CK \{+MP\}\{+ID\}$, then $\vdash x : A$ is derivable in the respective SeqS system.*

Proof. We must show that the axioms are derivable and that the set of derivable formulas is closed under (Modus Ponens), (RCEA), and (RCK). A derivation of axioms (ID) and (MP) is shown in examples 3.2 and 3.3 respectively.

Let us examine the other axioms.

For (Modus Ponens), suppose that $\vdash x : A \rightarrow B$ and $\vdash x : A$ are derivable. We easily have that $x : A \rightarrow B, x : A \vdash x : B$ is derivable too. Since cut is admissible, by two cuts we obtain $\vdash x : B$.

For (RCEA), we have to show that if $A \leftrightarrow B$ is derivable, then also $(A \Rightarrow C) \leftrightarrow (B \Rightarrow C)$ is so. The formula $A \leftrightarrow B$ is an abbreviation for $(A \rightarrow B) \wedge (B \rightarrow A)$. Suppose that $\vdash x : A \rightarrow B$ and $\vdash x : B \rightarrow A$ are derivable, we can derive $x : B \Rightarrow C \vdash x : A \Rightarrow C$ as follows: (the other half is symmetrical).

$$\frac{\frac{x : A \vdash x : B \quad x : B \vdash x : A}{x \xrightarrow{B} y \vdash x \xrightarrow{A} y, y : C} (EQ) \quad x \xrightarrow{B} y, y : C \vdash y : C}{x \xrightarrow{B} y, x : A \Rightarrow C \vdash y : C} (\Rightarrow L) \quad \frac{x \xrightarrow{B} y, x : A \Rightarrow C \vdash y : C}{x : A \Rightarrow C \vdash x : B \Rightarrow C} (\Rightarrow R)$$

For (RCK), suppose that $(1) \vdash x : B_1 \wedge B_2 \dots \wedge B_n \rightarrow C$, it must be derivable also $x : B_1, \dots, x : B_n \vdash x : C$. We set $\Gamma_i = x : A \Rightarrow B_i, x : A \Rightarrow B_{i+1}, \dots, x : A \Rightarrow B_n$, for $1 \leq i \leq n$. Then we have (we omit side formulas in $x \xrightarrow{A} y \vdash x \xrightarrow{A} y$):

$$\frac{x \xrightarrow{A} y \vdash x \xrightarrow{A} y \quad x : B_1, \dots, x : B_n \vdash x : C}{x \xrightarrow{A} y, x : A \Rightarrow B_n, x : B_1, \dots, x : B_{n-1} \vdash y : C} (\Rightarrow L) \quad \vdots \quad \frac{x \xrightarrow{A} y \vdash x \xrightarrow{A} y \quad x \xrightarrow{A} y, \Gamma_2, y : B_1 \vdash y : C}{x \xrightarrow{A} y, x : A \Rightarrow B_1, x : A \Rightarrow B_2, \dots, x : A \Rightarrow B_n \vdash y : C} (\Rightarrow L) \quad \frac{x \xrightarrow{A} y, x : A \Rightarrow B_1, x : A \Rightarrow B_2, \dots, x : A \Rightarrow B_n \vdash y : C}{x : A \Rightarrow B_1, x : A \Rightarrow B_2, \dots, x : A \Rightarrow B_n \vdash x : A \Rightarrow C} (\Rightarrow R)$$

□

4. PROOF-THEORETICAL ANALYSIS OF SEQ S

In this section we analyze the sequent calculus SeqS in order to obtain a decision procedure for our conditional systems CK, CK+ID, CK+MP and CK+MP+ID. In particular, we show that the contraction rules (Contr L) and (Contr R) can be eliminated in SeqCK and SeqID, but they cannot in SeqMP and SeqID+MP; however, in the last two systems one can control the application of these rules, in order to get a terminating calculus. Using the contraction rule in a controlled way is essential to prove the existence of a decision procedure for the associated logics; in fact, without this control, one can apply contraction duplicating arbitrarily any formula in the sequent.

First of all, we prove that we can eliminate the weakening rules.

THEOREM 4.1 ELIMINATION OF WEAKENING RULES. *Let $\Gamma \vdash \Delta$ be a sequent derivable in SeqS. Then $\Gamma \vdash \Delta$ has a derivation in SeqS with no application of (Weak L) and (Weak R).*

Proof. By induction on the height of the proof tree. □

Now we introduce the notion of *regular sequent*. Intuitively, regular sequents are those sequents whose set of transitions in the antecedent forms a *forest*. As we show in Theorem 4.4 below, any sequent in a proof beginning with a sequent of the form $\vdash x_0 : D$, for an arbitrary formula D , is regular. For this reason, we will restrict our concern to regular sequents.

We define the multigraph \mathcal{G} of the transition formulas in the antecedent of a sequent:

DEFINITION 4.2 MULTIGRAPH OF TRANSITIONS \mathcal{G} . *Given a sequent $\Gamma \vdash \Delta$, where $\Gamma = \Gamma', T$ and T is the multiset of transition formulas and Γ' does not contain transition formulas, we define the multigraph $\mathcal{G} = \langle V, E \rangle$ associated to $\Gamma \vdash \Delta$ with vertexes V and edges E . V is the set of labels occurring in $\Gamma \vdash \Delta$ and $\langle x, y \rangle \in E$ whenever $x \xrightarrow{F} y \in T$.*

DEFINITION 4.3 REGULAR SEQUENT. *A sequent $\Gamma \vdash \Delta$ is called regular if its associated multigraph of transitions \mathcal{G} is a forest. In particular, there is at most one link between two vertexes and there are no loops.*

We can observe that we can always restrict our concern to regular sequents, since we have the following theorem:

THEOREM 4.4 PROOFS WITH REGULAR SEQUENTS. *Every proof tree beginning with a sequent $\vdash x_0 : D$ and obtained by applying SeqS's rules, contains only regular sequents.*

Proof. First, we show that \mathcal{G} is a *graph*, i.e. there is at most one link between two vertexes. This can be seen by an easy inductive argument: $\vdash x_0 : D$ obviously respects this condition. Consider an arbitrary $\Gamma \vdash \Delta$ which respects this condition, $(\Rightarrow R)$ is the only rule of the calculus which introduces, looking backward, a transition formula in the antecedent of the sequent to which it is applied. In particular, $(\Rightarrow R)$ with principal formula $x : A \Rightarrow B$ introduces a transition $x \xrightarrow{A} y$ where y is a "new label", then there cannot be another transition $x \xrightarrow{F} y$ in the antecedent. To see that \mathcal{G} is a forest, again we do a simple inductive argument: the graph associated to $\vdash x_0 : D$ is certainly a forest ($\mathcal{G} = \langle \{x_0\}, \emptyset \rangle$, which is a tree); consider a rule application which has $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$ as premises and $\Gamma \vdash \Delta$ as a conclusion, and assume by induction hypothesis that the graph \mathcal{G} associated to $\Gamma \vdash \Delta$ is a forest. It is easy to observe that applying any rule of SeqS, the graphs \mathcal{G}_1 and \mathcal{G}_2 , associated to $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$ respectively, are forests. (Contr L), (Contr R), (Weak L), (Weak R), $(\Rightarrow L)$, $(\rightarrow L)$, $(\rightarrow R)$ and (MP) do not modify the graph \mathcal{G} ; $(\Rightarrow R)$ adds a transition $x \xrightarrow{F} y$ in the initial forest, but y is a "new" label as discussed above, thus \mathcal{G}_1 is still a forest obtained by adding a new vertex and a new edge; consider the (ID) rule:

$$\frac{\Gamma, y : A \vdash \Delta}{\Gamma, x \xrightarrow{A} y \vdash \Delta} (ID)$$

The application of (ID) deletes the edge $\langle x, y \rangle$ from \mathcal{G} , and \mathcal{G}_1 is still a forest. When (EQ) is applied, the calculus tries to find two derivations starting with only one label u and no transitions; therefore, \mathcal{G}_1 and \mathcal{G}_2 are trees $\langle \{u\}, \emptyset \rangle$. \square

As mentioned above, we restrict from now on our attention to regular sequents. In the following, we prove some elementary properties of regular sequents.

THEOREM 4.5 PROPERTY OF $(\Rightarrow L)$. *Let the sequent*

$$\Gamma \vdash \Delta, x \xrightarrow{A} y$$

with $x \neq y$, be derivable in SeqS, then one of the following sequents:

(1) $\Gamma \vdash \Delta$

(2) $x \xrightarrow{F} y \vdash x \xrightarrow{A} y$, where $x \xrightarrow{F} y \in \Gamma$

is also derivable in SeqS.

Proof. (EQ) is the only SeqS's rule which operates on a transition formula on the right hand side (consequent) of a sequent. Thus, we have to consider only two cases, analyzing the proof tree of $\Gamma \vdash \Delta, x \xrightarrow{A} y$:

(1) $x \xrightarrow{A} y$ is introduced by weakening: in this case, $\Gamma \vdash \Delta$ is derivable;

(2) $x \xrightarrow{A} y$ is introduced by the (EQ) rule: in this case, another transition $x \xrightarrow{F} y$ *must* be in Γ , in order to apply (EQ). To see this, observe that the only rule that could introduce a transition formula (looking backward) in the antecedent of a sequent is $(\Rightarrow R)$, but it can only introduce a transition of the form $x \xrightarrow{F} z$, where z *does not occur in that sequent* (it is a *new* label), thus it cannot introduce the transition $x \xrightarrow{F} y$.

The (EQ) rule is only applied to transition formulas:

$$\frac{u : F \vdash u : A \quad u : A \vdash u : F}{x \xrightarrow{F} y \vdash x \xrightarrow{A} y} (EQ)$$

therefore we can say that $x \xrightarrow{F} y \vdash x \xrightarrow{A} y$ is derivable in SeqS. \square

Notice that this theorem holds for all the systems SeqS, but only if $x \neq y$. In SeqMP and SeqID+MP the (MP) rule operates on transitions in the consequent, although on transitions like $x \xrightarrow{A} x$. In this case the theorem does not hold, as shown by the following counterexample:

$$\frac{x : A \vdash x : A, x : B}{x : A \vdash x \xrightarrow{A} x, x : B} (MP)$$

for A and B arbitrary. The sequent $x : A \vdash x \xrightarrow{A} x, x : B$ is derivable in SeqMP, but $x : A \vdash x : B$ is not derivable in this system and the second condition is not applicable (no transition formula occurs in the antecedent).

The first hypothesis of the theorem ($x \neq y$) excludes this situation.

THEOREM 4.6 ELIMINATION OF THE CONTRACTION RULES ON TRANSITION FORMULAS.
Given a sequent $\Gamma \vdash \Delta$, derivable in SeqS, there is a proof tree with no applications of (Contr L) and (Contr R) on transition formulas.

Proof Consider one maximal⁵ contraction of the proof tree: it can be eliminated as follows.

In SeqCK a transition formula $x \xrightarrow{A} y$ can be only introduced by the (EQ) rule (looking forward):

$$\frac{\frac{u : B \vdash u : A \quad u : A \vdash u : B}{\Gamma'', x \xrightarrow{B} y \vdash \Delta'', x \xrightarrow{A} y, x \xrightarrow{A} y} (EQ)}{\Gamma' \vdash \Delta', x \xrightarrow{A} y, x \xrightarrow{A} y} (\text{ContrR})$$

We can obtain the following proof erasing the contraction step:

$$\frac{\frac{u : B \vdash u : A \quad u : A \vdash u : B}{\Gamma'', x \xrightarrow{B} y \vdash \Delta'', x \xrightarrow{A} y} (EQ)}{\Gamma' \vdash \Delta', x \xrightarrow{A} y}$$

where Π'_1 is obtained by removing an occurrence of $x \xrightarrow{A} y$ on the right side of every sequent of Π_1 .

In SeqID and SeqID+MP we can have proofs like the following one:

$$\frac{\frac{\frac{\Pi_3}{\Gamma''', y : A \vdash \Delta'''} (ID)}{\Gamma''', x \xrightarrow{A} y \vdash \Delta'''} (\text{ContrL})}{\Gamma'', x \xrightarrow{A} y, y : A \vdash \Delta''} (ID)}{\Gamma'', x \xrightarrow{A} y, x \xrightarrow{A} y \vdash \Delta''} (\text{ContrL})$$

⁵Of maximal distance from the root of the proof tree.

By the permutability of (ID) over the other rules of SeqS, we can have:

$$\begin{array}{c}
 \Pi_3 \\
 \Gamma''', y: A \vdash \Delta''' \\
 \Pi'_1 \\
 \frac{\Gamma', y: A, y: A \vdash \Delta'}{\Gamma', y: A, x \xrightarrow{A} y \vdash \Delta'} (ID) \\
 \frac{\Gamma', y: A, x \xrightarrow{A} y \vdash \Delta'}{\Gamma', x \xrightarrow{A} y, x \xrightarrow{A} y \vdash \Delta'} (ID) \\
 \frac{\Gamma', x \xrightarrow{A} y, x \xrightarrow{A} y \vdash \Delta'}{\Gamma', x \xrightarrow{A} y \vdash \Delta'} (ContrL)
 \end{array}$$

where Π'_1 is obtained from Π_1 and Π_2 by permuting the two applications of (ID). We can then eliminate the contraction on the transition formula, introducing an application of (Contr L) on the subformula $y: A$:

$$\begin{array}{c}
 \Pi_3 \\
 \Gamma''', y: A \vdash \Delta''' \\
 \Pi'_1 \\
 \frac{\Gamma', y: A, y: A \vdash \Delta'}{\Gamma', y: A \vdash \Delta'} (ContrL) \\
 \frac{\Gamma', y: A \vdash \Delta'}{\Gamma', x \xrightarrow{A} y \vdash \Delta'} (ID)
 \end{array}$$

In SeqMP and SeqID+MP we can eliminate a contraction step on a transition formula $x \xrightarrow{A} x$ in the consequent of a sequent by replacing it with an application of (Contr R) on the subformula $x: A$ in a similar way to the case of (ID).

In all the SeqS calculi, if a transition formula is introduced by (implicit) weakening, the contraction is eliminated by eliminating that weakening.

□

4.1 Elimination of (Contr R) on Conditional Formulas

In this subsection we show that SeqS calculi are still complete without the (Contr R) rule applied to conditional formulas $x: A \Rightarrow B$. The elimination of the right contraction on conditionals is a direct consequence of the so-called *disjunction property* for conditional formulas: if $(A_1 \Rightarrow B_1) \vee (A_2 \Rightarrow B_2)$ is valid, then either $(A_1 \Rightarrow B_1)$ or $(A_2 \Rightarrow B_2)$ is valid too. This property follows an important proposition, which does not hold for all sequents, but only for non- x -branching sequents, i.e. those sequents which do not create a branching in x or in a predecessor of x . Let us introduce some essential definitions.

DEFINITION 4.7 PREDECESSOR AND SUCCESSOR, FATHER AND SON. *Given a sequent $\Gamma', T \vdash \Delta$, where all the transitions in the antecedent are in T , we say that a world w is a predecessor of a world x if there is a path from w to x in the graph of transitions $\mathcal{G} = \langle V, E \rangle$ of the sequent. In this case, we also say that x is a successor of w . If $\langle w, x \rangle \in E$, we say that w is the father of x and that x is a son of w .*

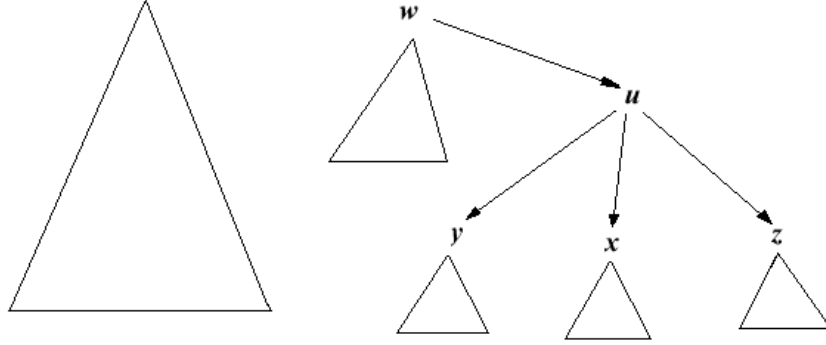


Fig. 3. The forest \mathcal{G} of a sequent; w is predecessor of x , x is successor of w , u is father of x and x is son of u .

As we mentioned above, the graph of transitions forms a forest, as shown in Figure 3.

DEFINITION 4.8 POSITIVE AND NEGATIVE OCCURRENCES OF A FORMULA. *Given a formula A , we say that:*

- A occurs positively in A ;
- if a formula $B \rightarrow C$ occurs positively (negatively) in A , then C occurs positively (negatively) in A and B occurs negatively (positively) in A ;
- if a formula $B \Rightarrow C$ occurs positively (negatively) in A , then C occurs positively (negatively) in A .

A formula F occurs positively (negatively) in a multiset Γ if F occurs positively (negatively) in some formula $G \in \Gamma$.

Now we introduce the definition of x -branching formula. Intuitively, $\mathcal{B}(x, T)$ contains formulas that create a branching in x or in a predecessor of x according to T . $\mathcal{B}(x, T)$ also contains the conditionals $u : A \Rightarrow B$ such that $T \vdash u \xrightarrow{A} v$ and B creates a branching in x (i.e. $v = x$) or in a predecessor v of x .

DEFINITION 4.9 x -BRANCHING FORMULAS. *Given a multiset of transition formulas T , we define the set of x -branching formulas, denoted with $\mathcal{B}(x, T)$, as follows:*

- $x : A \rightarrow B \in \mathcal{B}(x, T)$;
- $u : A \rightarrow B \in \mathcal{B}(x, T)$ if $T \vdash u \xrightarrow{F} x$ for some formula F ;
- $u : A \Rightarrow B \in \mathcal{B}(x, T)$ if $T \vdash u \xrightarrow{A} v$ and $v : B \in \mathcal{B}(x, T)$.

We also introduce the notion of x -branching sequent. Intuitively, we say that $\Gamma \vdash \Delta$ is x -branching if it contains an x -branching formula occurring positively in Γ or if it contains an x -branching formula occurring negatively in Δ . Since in systems containing (ID) a transition $u \xrightarrow{F} v$ in the antecedent can be derived from

$v : F$ and $v : F$ can be x -branching, we impose that a sequent $\Gamma', u \xrightarrow{F} v \vdash \Delta$ is x -branching if $\Gamma', v : F \vdash \Delta$ is x -branching; for the same reason, in systems containing (MP) we impose that a sequent $\Gamma \vdash \Delta', u \xrightarrow{F} u$ is x -branching if $\Gamma \vdash \Delta', u : F$ is x -branching.

In systems containing (MP) we also impose that a sequent $\Gamma', w : A \Rightarrow B \vdash \Delta$ is x -branching if the sequent $\Gamma' \vdash \Delta, w \xrightarrow{A} w$ is derivable and w is a predecessor of x (or $w = x$), since $w : A$ can introduce x -branching formula(s) in the sequent.

DEFINITION 4.10 *x*-BRANCHING SEQUENTS. *Given a sequent $\Gamma \vdash \Delta$, we denote by Γ' the labelled formulas in Γ and by T the transition formulas in Γ , so that $\Gamma = \Gamma', T$. To define when a sequent $\Gamma \vdash \Delta$ is x -branching according to each system, we consider the following conditions:*

- (1) a formula $u : F \in \mathcal{B}(x, T)$ occurs positively in Γ ;
- (2) a formula $u : F \in \mathcal{B}(x, T)$ occurs negatively in Δ .
- (3) $T = T', u \xrightarrow{F} v$ and the sequent $\Gamma', T', v : F \vdash \Delta$ is x -branching;
- (4) $u \xrightarrow{F} u \in \Delta$ and the sequent $\Gamma \vdash \Delta', u : F$ is x -branching ($\Delta = \Delta', u \xrightarrow{F} u$);
- (5) a formula $w : A \Rightarrow B \in \Gamma$, w is a predecessor of x in the graph \mathcal{G} of transitions or $w = x$ and $\Gamma'' \vdash \Delta, w \xrightarrow{A} w$ is derivable, where $\Gamma = \Gamma'', w : A \Rightarrow B$.

We say that $\Gamma \vdash \Delta$ is x -branching for each system if the following combinations of the previous conditions hold:

- CK: 1, 2
- CK+ID: 1, 2, 3
- CK+MP: 1, 2, 4, 5
- CK+MP+ID: 1, 2, 3, 4, 5

The disjunction property characterizes *only* the sequents that are not x -branching. To prove the disjunction property, we need to consider a more general setting; namely, we shall consider a sequent of the form $\Gamma \vdash \Delta, y : A, z : B$, whose forest of transitions has the form represented in Figure 4, i.e. it has one subtree with root u and another subtree with root v , with $u \neq v$; y is a member of the tree with root u and z is a member of the tree with root v ; x is the father of u and v and the tree containing x has root r .

We need some more definitions. In particular, given a sequent $\Gamma \vdash \Delta$ with its associated forest $\mathcal{G} = \langle V, E \rangle$, consider a label k contained in the tree with root r . We define the set T_k° of the labels contained in the tree of \mathcal{G} with root k and the sets Γ_k° and Δ_k° , containing all the formulas of Γ and Δ whose labels are in the tree of \mathcal{G} with root k . We also define the set T_k^* of the labels contained in the tree of \mathcal{G} with root k or in a path from r to k , and the sets Γ_k^* and Δ_k^* , containing all the formulas of Γ and Δ whose labels are in the tree of \mathcal{G} with root k or on the path from r to k .

DEFINITION 4.11 T_k° . T_k° is the set of labels in the tree of \mathcal{G} with root k ; more precisely:

- $k \in T_k^\circ$

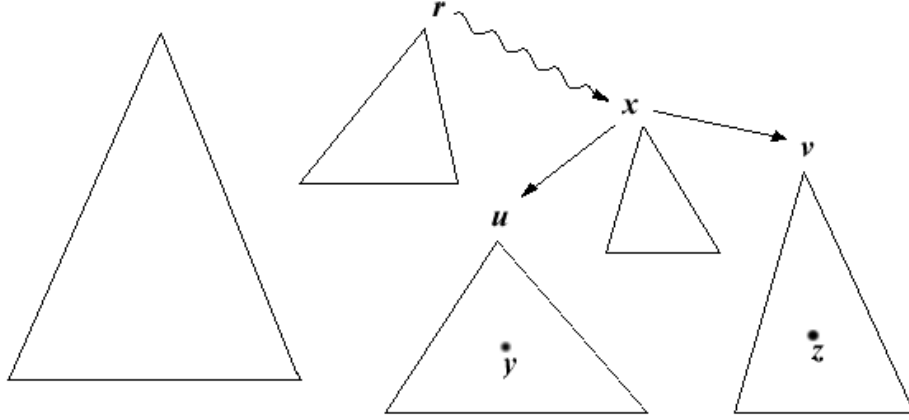


Fig. 4. The forest \mathcal{G} of transitions used to prove the disjunction property.

- if $\langle u, w \rangle \in E$ and $u \in T_k^\circ$, then $w \in T_k^\circ$.

DEFINITION 4.12 T_k^* . T_k^* is the set of labels in the tree of \mathcal{G} with root k or on a path from r^6 to k ; more precisely:

$$T_k^* = T_k^\circ \cup P_k$$

where P_k is the set of labels on a path from r to k :

- $k \in P_k$
- if $\langle u, w \rangle \in E$ and $w \in P_k$, then $u \in P_k$.

DEFINITION 4.13 Γ_k° . Γ_k° is the multiset of formulas of Γ contained in the tree of \mathcal{G} with root k ; if $w \xrightarrow{F} k \in \Gamma$, then $w \xrightarrow{F} k \in \Gamma_k^\circ$; more precisely:

$$\Gamma_k^\circ = \{u : F \in \Gamma \mid u \in T_k^\circ\} \cup \{w \xrightarrow{F} u \in \Gamma \mid w \in T_k^\circ \text{ or } u \in T_k^\circ\}$$

DEFINITION 4.14 Δ_k° . Δ_k° is the multiset of formulas of Δ contained in the tree of \mathcal{G} with root k ; if $w \xrightarrow{F} k \in \Delta$, then $w \xrightarrow{F} k \in \Delta_k^\circ$; more precisely:

$$\Delta_k^\circ = \{u : F \in \Delta \mid u \in T_k^\circ\} \cup \{w \xrightarrow{F} u \in \Delta \mid w \in T_k^\circ \text{ or } u \in T_k^\circ\}$$

DEFINITION 4.15 Γ_k^* . Γ_k^* is the multiset of formulas of Γ contained in the tree of \mathcal{G} with root k or on a path from r to k ; more precisely:

$$\Gamma_k^* = \{w : F \in \Gamma \mid w \in T_k^*\} \cup \{w \xrightarrow{F} w' \in \Gamma \mid w' \in T_k^*\}$$

DEFINITION 4.16 Δ_k^* . Δ_k^* is the multiset of formulas of Δ contained in the tree of \mathcal{G} with root k or on a path from r to k ; more precisely:

$$\Delta_k^* = \{w : F \in \Delta \mid w \in T_k^*\} \cup \{w \xrightarrow{F} w' \in \Delta \mid w' \in T_k^*\}$$

Now we have all the elements to prove the following:

⁶The label r is the root of the tree containing k . It could be $r=k$.

PROPOSITION 4.17. *Given a sequent $\Gamma \vdash \Delta, y : A, z : B$ and its forest of transitions \mathcal{G} , if it is derivable in SeqS and has the following features:*

- (1) \mathcal{G} is a forest of the form as shown in Figure 4 (thus y is a member of the tree with root u and z is a member of the tree with root v , with $u \neq v$; u and v are sons of x);
- (2) $\Gamma \vdash \Delta, y : A, z : B$ is not x -branching

then one of the following sequents is derivable in SeqS:

- (1) $\Gamma_u^* \vdash \Delta_u^*, y : A$
- (2) $\Gamma_v^* \vdash \Delta_v^*, z : B$
- (3) $\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash \Delta - (\Delta_u^\circ \cup \Delta_v^\circ)$

Moreover, the proofs of 1, 2 and 3 do not add any contraction to the proof of $\Gamma \vdash \Delta, y : A, z : B$.

Proof. By induction on the height of the proof tree of the sequent $\Gamma \vdash \Delta, y : A, z : B$. We present two examples, the other cases are left to the reader.

- (1) Consider the case where $y : A$ is a conditional formula $y : C \Rightarrow D$ and is the principal formula of an application of the (\Rightarrow R) rule. The proof tree of the sequent is ended by:

$$\frac{\Gamma, y \xrightarrow{C} k \vdash \Delta, k : D, z : B}{\Gamma \vdash \Delta, y : C \Rightarrow D, z : B} (\Rightarrow R)$$

We can apply the inductive hypothesis on the only premise of the (\Rightarrow R) rule; in fact, $\Gamma, y \xrightarrow{C} k \vdash \Delta, k : D, z : B$ is not x -branching. It could become x -branching as an effect of the introduction of $k : D$ and $y \xrightarrow{C} k$, but this is impossible since k is a "new" label, then it is in the same tree of y and not on a path to x . Applying the inductive hypothesis, we must consider the three possible situations:

- (a) $(\Gamma, y \xrightarrow{C} k)_u^* \vdash \Delta_u^*, k : D$ is derivable: it is easy to see that $y \xrightarrow{C} k \in (\Gamma, y \xrightarrow{C} k)_u^*$, since u is a predecessor of y and thus of k ; then we obtain the following derivation:

$$\frac{\Gamma_u^*, y \xrightarrow{C} k \vdash \Delta_u^*, k : D}{\Gamma_u^* \vdash \Delta_u^*, y : C \Rightarrow D} (\Rightarrow R)$$

- (b) $(\Gamma, y \xrightarrow{C} k)_v^* \vdash \Delta_v^*, z : B$ is derivable: k is in the tree with root u , thus $y \xrightarrow{C} k \notin (\Gamma, y \xrightarrow{C} k)_v^*$: we obtain that

$$\Gamma_v^* \vdash \Delta_v^*, z : B$$

is derivable;

- (c) $(\Gamma, y \xrightarrow{C} k) - ((\Gamma, y \xrightarrow{C} k)_u^\circ \cup (\Gamma, y \xrightarrow{C} k)_v^\circ) \vdash \Delta - (\Delta_u^\circ \cup \Delta_v^\circ)$ is derivable: k is in the subtree with root u , thus $y \xrightarrow{C} k \in (\Gamma, y \xrightarrow{C} k)_u^\circ$ and then $y \xrightarrow{C} k \notin (\Gamma, y \xrightarrow{C} k) - ((\Gamma, y \xrightarrow{C} k)_u^\circ \cup (\Gamma, y \xrightarrow{C} k)_v^\circ)$, from which we obtain that

$$\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash \Delta - (\Delta_u^\circ \cup \Delta_v^\circ)$$

is derivable.

- (2) Let us now analyze the case where the principal formula of the sequent is a formula $w : F \in \Delta$; the (Contr R) rule is applied to that formula, as shown below:

$$\frac{\Gamma \vdash \Delta', w : F, w : F, y : A, z : B}{\Gamma \vdash \Delta', w : F, y : A, z : B} \text{ (ContrR)}$$

We can obviously apply the inductive hypothesis on the premise, obtaining the three following alternatives:

- (a) $\Gamma_u^* \vdash (\Delta', w : F, w : F)_u^*, y : A$ is derivable: if $w \in T_u^*$, then the sequent $\Gamma_u^* \vdash \Delta'_u^*, w : F, w : F, y : A$ is derivable, as $w : F$ occurs in $(\Delta', w : F, w : F)_u^*$. Therefore we can obtain the following proof:

$$\frac{\Gamma_u^* \vdash \Delta'_u^*, w : F, w : F, y : A}{\Gamma_u^* \vdash \Delta'_u^*, w : F, y : A} \text{ (ContrR)}$$

and that's it, since $\Delta = \Delta', w : F$ and then $\Delta_u^* = \Delta'_u^*, w : F$.

If $w \notin T_u^*$, $w : F$ is not member of the multiset $(\Delta', w : F, w : F)_u^*$, then the sequent $\Gamma_u^* \vdash \Delta'_u^*, y : A$ is derivable by the inductive hypothesis, from what we can conclude since $\Delta = \Delta', w : F$ and then $\Delta_u^* = \Delta'_u^*$.

- (b) $\Gamma_v^* \vdash (\Delta', w : F, w : F)_v^*, z : B$ is derivable: the proof is similar to the previous one and $\Gamma_v^* \vdash (\Delta', w : F)_v^*, z : B$ is derivable .
- (c) $\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash (\Delta', w : F, w : F) - ((\Delta', w : F, w : F)_u^\circ \cup (\Delta', w : F, w : F)_v^\circ)$ is derivable: if $w \in T_u^\circ \cup T_v^\circ$, then $\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash \Delta' - (\Delta'_u^\circ \cup \Delta'_v^\circ)$ is derivable, and we can conclude the proof as $\Delta = \Delta', w : F$, but $w : F \in \Delta_u^\circ \cup \Delta_v^\circ$, then it is not member of the difference, thus $\Delta - (\Delta_u^\circ \cup \Delta_v^\circ) = \Delta' - (\Delta'_u^\circ \cup \Delta'_v^\circ)$.

If $w : F \notin T_u^\circ \cup T_v^\circ$, then $\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash (\Delta', w : F, w : F) - (\Delta'_u^\circ \cup \Delta'_v^\circ)$ is derivable by the inductive hypothesis, from what we can have:

$$\frac{\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash \Delta' - (\Delta'_u^\circ \cup \Delta'_v^\circ), w : F, w : F}{\Gamma - (\Gamma_u^\circ \cup \Gamma_v^\circ) \vdash \Delta' - (\Delta'_u^\circ \cup \Delta'_v^\circ), w : F} \text{ (ContrR)}$$

and we can conclude the proof, since $\Delta = \Delta', w : F$ and $w \notin T_u^\circ \cup T_v^\circ$; then, we observe that $\Delta - (\Delta_u^\circ \cup \Delta_v^\circ) = \Delta' - (\Delta'_u^\circ \cup \Delta'_v^\circ), w : F$.

In this case, we introduce a (Contr R) to prove the Proposition; however, this contraction is already in the proof tree of the initial sequent, thus we do not *add* any contraction on it (we use the same contraction on $w : F$).

□

THEOREM 4.18 DISJUNCTION PROPERTY. *Given a non x -branching sequent*

$$\Gamma \vdash \Delta, x : A_1 \Rightarrow B_1, x : A_2 \Rightarrow B_2$$

derivable in SeqS with a derivation Π , one of the following sequents:

- (1) $\Gamma \vdash \Delta, x : A_1 \Rightarrow B_1$
- (2) $\Gamma \vdash \Delta, x : A_2 \Rightarrow B_2$

is derivable in SeqS by a proof tree which does not add any application of the contraction rules (Contr L) and (Contr R) to Π .

Proof. The sequent $\Gamma \vdash \Delta, x : A_1 \Rightarrow B_1, x : A_2 \Rightarrow B_2$ is derivable in SeqS, then we can find a derivation Π of it; the two conditional formulas can be introduced (looking forward) in two ways:

- (1) by weakening;
- (2) by the application of the (\Rightarrow R) rule.

In case 1 suppose that $x : A_1 \Rightarrow B_1$ is introduced by weakening (by Theorem 4.1 we can only consider implicit weakenings): the proof is ended by erasing all the instances of $x : A_1 \Rightarrow B_1$ introduced by weakening in Π , obtaining a proof of $\Gamma \vdash \Delta, x : A_2 \Rightarrow B_2$.

In case 2 both the conditional formulas are introduced by an application of the (\Rightarrow R) rule; by the permutability of this rule over all the others, we can consider a proof tree ending as follows:

$$\frac{\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z \vdash \Delta, y : B_1, z : B_2}{\Gamma, x \xrightarrow{A_1} y \vdash \Delta, y : B_1, x : A_2 \Rightarrow B_2} (\Rightarrow R)$$

$$\frac{\Gamma, x \xrightarrow{A_1} y \vdash \Delta, y : B_1, x : A_2 \Rightarrow B_2}{\Gamma \vdash \Delta, x : A_1 \Rightarrow B_1, x : A_2 \Rightarrow B_2} (\Rightarrow R)$$

The sequent $\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z \vdash \Delta, y : B_1, z : B_2$ respects all the conditions to apply the Proposition 4.17 (y and z are "new" labels), then we have that one of the following sequents:

- (1) $(\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_y^* \vdash \Delta_y^*, y : B_1$
- (2) $(\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_z^* \vdash \Delta_z^*, z : B_2$
- (3) $(\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z) - ((\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_y^\circ \cup (\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_z^\circ) \vdash \Delta - (\Delta_y^\circ \cup \Delta_z^\circ)$

is derivable in SeqS without adding any application of the contraction rules. In all these cases we can proof the disjunction property:

- (1) z is not in the tree with root y , and is not on a path towards y , then the transition formula $x \xrightarrow{A_2} z$ is not member of the multiset $(\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_y^*$; the

sequent $\Gamma_y^*, x \xrightarrow{A_1} y \vdash \Delta_y^*, y : B_1$ is then derivable, from what we have a proof:

$$\frac{\frac{\Gamma_y^*, x \xrightarrow{A_1} y \vdash \Delta_y^*, y : B_1}{\Gamma_y^* \vdash \Delta_y^*, x : A_1 \Rightarrow B_1} (\Rightarrow R)}{\Gamma \vdash \Delta, x : A_1 \Rightarrow B_1} (Weak)$$

since $\Gamma_y^* \subseteq \Gamma$ and $\Delta_y^* \subseteq \Delta$;

(2) symmetric to the previous case;

(3) we can observe that $x \xrightarrow{A_1} y \in (\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_y^\circ$, and that $x \xrightarrow{A_2} z \in (\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_z^\circ$; both the transition formulas are members of $(\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_y^\circ \cup (\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_z^\circ$ and then they are *not* members of $(\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z) - ((\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_y^\circ \cup (\Gamma, x \xrightarrow{A_1} y, x \xrightarrow{A_2} z)_z^\circ)$. Therefore, the sequent $\Gamma - (\Gamma_y^\circ \cup \Gamma_z^\circ) \vdash \Delta - (\Delta_y^\circ \cup \Delta_z^\circ)$ is derivable and, observing that $\Gamma - (\Gamma_y^\circ \cup \Gamma_z^\circ) \subseteq \Gamma$ and that $\Delta - (\Delta_y^\circ \cup \Delta_z^\circ) \subseteq \Delta$, we have the proof:

$$\frac{\frac{\Gamma - (\Gamma_y^\circ \cup \Gamma_z^\circ) \vdash \Delta - (\Delta_y^\circ \cup \Delta_z^\circ)}{\Gamma \vdash \Delta} (Weak)}{\Gamma \vdash \Delta, x : A_1 \Rightarrow B_1} (WeakR)$$

or

$$\frac{\frac{\Gamma - (\Gamma_y^\circ \cup \Gamma_z^\circ) \vdash \Delta - (\Delta_y^\circ \cup \Delta_z^\circ)}{\Gamma \vdash \Delta} (Weak)}{\Gamma \vdash \Delta, x : A_2 \Rightarrow B_2} (WeakR)$$

□

By the correctness and completeness of SeqS, it is easy to prove the following corollary of the disjunction property:

COROLLARY 4.19. *If $(A \Rightarrow B) \vee (C \Rightarrow D)$ is valid in $CK\{+MP\}\{+ID\}$, then either $A \Rightarrow B$ or $C \Rightarrow D$ is valid in $CK\{+MP\}\{+ID\}$.*

Finally, we can prove the eliminability of the (Contr R) in SeqS systems on conditional formulas as another corollary of the disjunction property.

COROLLARY 4.20 ELIMINATION OF THE (CONTR R) RULE ON CONDITIONAL FORMULAS. *If $\vdash x_0 : D$ is derivable in SeqS, then it has a proof where there are no right contractions on conditional formulas.*

Proof. By permutation properties, a proof Π ending with

$$\Gamma \vdash \Delta, x : A \Rightarrow B, x : A \Rightarrow B$$

can be transformed into a proof Π' , where all the rules introducing x -branching formulas are permuted over the other rules (i.e. they are applied at the bottom of the tree). As an example, let the end sequent of Π have the form

$$\Gamma, x : C \rightarrow D \vdash \Delta, x : A \Rightarrow B, x : A \Rightarrow B$$

We have that the lower sequent is x -branching, (at least) since $x : C \rightarrow D$. We can permute Π so that the last step is the introduction of the x -branching formula $x : C \rightarrow D$ from the two sequents:

$$\Gamma \vdash \Delta, x : C, x : A \Rightarrow B, x : A \Rightarrow B \text{ and } \Gamma, x : D \vdash \Delta, x : A \Rightarrow B, x : A \Rightarrow B.$$

We have decomposed the x -branching formula, if the two sequents are still x -branching we perform a similar permutation upwards, so that at the end every branch of Π' will contain a sequent $\Gamma_i \vdash \Delta_i, x : A \Rightarrow B, x : A \Rightarrow B$, such that Γ_i, Δ_i are no longer x -branching. Notice that if a sequent $\Gamma, w : C \Rightarrow D \vdash \Delta, x : A \Rightarrow B, x : A \Rightarrow B$ is x -branching because of $w : C \Rightarrow D$, we can permute (\Rightarrow L) over the other rules, since w is a predecessor of x in the tree of transitions (see the definition 4.10 above): the label used to decompose the conditional formula is already in the sequent, then the permutation is possible⁷. Then we can apply the disjunction property and obtain that for each i ,

$$\Gamma_i \vdash \Delta_i, x : A \Rightarrow B \text{ is derivable.}$$

Thus, deleting one occurrence of $x : A \Rightarrow B$ in the consequent of any sequent in Π' below $\Gamma_i \vdash \Delta_i, x : A \Rightarrow B, x : A \Rightarrow B$ we get a derivation of $\Gamma \vdash \Delta, x : A \Rightarrow B$.

□

4.2 Elimination of Contractions in SeqCK and SeqID

In this subsection we show that we can eliminate the application of contraction rules in SeqCK and SeqID systems.

THEOREM 4.21 ELIMINATION OF CONTRACTIONS IN SEQCK AND SEQID. *Given a sequent $\vdash x_0 : D$, derivable in SeqCK or in SeqID, it has a derivation with no applications of (Contr L) and (Contr R).*

Proof (Sketch). As we proved above, we do not have to consider the case of (Contr R) applied to conditional formulas and contractions on transitions. The proof proceeds similarly to the one of Theorem 9.1.1 in [Viganò 2000] by triple induction respectively: (i) on the number of contractions in a proof of the sequent, (ii) on the complexity of the formula involved in a contraction step, and (iii) on the rank of the contraction; we need the following two definitions:

DEFINITION 4.22 COMPLEXITY OF A FORMULA $CP(F)$. *We define the complexity of a formula F as follows:*

- (1) $cp(x : A) = 2^* | A |$
- (2) $cp(x \xrightarrow{A} y) = 2^* | A | + 1$

where $| A |$ is the number of symbols occurring in the string representing the formula A .

⁷As explained, (\Rightarrow L) does not permute over the application of (\Rightarrow R) which introduces the label used by (\Rightarrow L).

DEFINITION 4.23 RANK OF A CONTRACTION. *We define the rank of a contraction as the largest number of steps between the conclusions of a contraction and an upward sequent containing at least one of the two copies of the formula that is contracted.*

For example, given the following proof tree:

$$\frac{\frac{\frac{\frac{\Pi}{\Gamma, x: A, x: A \vdash \Delta, x: B, x: B}}{\Gamma, x: A \vdash \Delta, x: B, x: A \rightarrow B} (\rightarrow R)}{\Gamma \vdash \Delta, x: A \rightarrow B, x: A \rightarrow B} (\rightarrow R)}{\Gamma \vdash \Delta, x: A \rightarrow B} (\text{Contr}R)$$

the rank of the contraction applied to $x : A \rightarrow B$ is 2 (the minimum rank available), since there are two sequents between the conclusion of the (Contr R) rule and the first sequent in which there are no instances of the constituent of the contraction.

The third induction, on the rank, is needed since the rule (\Rightarrow L), which does not permute over the (\Rightarrow R) rule: we cannot assume that there is a proof which introduces the two copies of the conditional formulas by (\Rightarrow L) rule one after the other (this happens when the introduction of the first copy is separated by the introduction of the second copy by (\Rightarrow R) rule occurring in the middle) and we need to consider separately the two cases.

To carry on the proof, suppose that a derivation Π of $\vdash x_0 : D$ contains $i + 1$ contractions. Concentrate on a maximal instance of contraction, say on a formula F (so that the portion of the derivation above this step is contraction-free). In order to eliminate this contraction step, thereby obtaining a proof Π' that contain i contractions, we proceed by induction on the complexity of F , and then by induction on the rank of the contraction step. We only sketch the proof of the most difficult case, the one of a left contraction on a conditional formula $x : A \Rightarrow B$; the other cases are easy and left to the reader. One can find the entire proof in [Pozzato 2003].

We consider proof trees where (\Rightarrow L) is applied as follows:

$$\frac{x \xrightarrow{A'} y \vdash x \xrightarrow{A} y \quad \Gamma, y : B \vdash \Delta}{\Gamma, x : A \Rightarrow B \vdash \Delta} (\Rightarrow L)$$

In fact, no rules in SeqCK and SeqID introduces (looking forward) a transition formula $x \xrightarrow{A} x$ in the consequent of a sequent, then we can also apply the Theorem 4.5 to all the applications of (\Rightarrow L).

Given the following proof:

$$\frac{\frac{\Pi_1}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta}}{\Gamma, x: A \Rightarrow B \vdash \Delta} (\text{Contr}L)}{\vdash x_0: D} (\Pi_0)$$

we can obtain a proof Π^* of the sequent $\Gamma, x : A \Rightarrow B \vdash \Delta$, removing that contraction on $x : A \Rightarrow B$.

By induction on the rank of the contraction, we have the following cases:

(1) *Base: rank=2*: we have the following proof:

$$\begin{array}{c}
 \Pi_A \quad \frac{\Pi_B \quad \Pi_C^*}{x \xrightarrow{A''} w \vdash x \xrightarrow{A} w \quad x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1, y: B, w: B \vdash \Delta_1, y: B', w: B''} (\Rightarrow L) \\
 \frac{x \xrightarrow{A'} y \vdash x \xrightarrow{A} y \quad \Gamma_1, x: A \Rightarrow B, y: B, x \xrightarrow{A'} y, x \xrightarrow{A''} w \vdash \Delta_1, y: B', w: B''} (\Rightarrow L)}{\Gamma_1, x: A \Rightarrow B, x: A \Rightarrow B, x \xrightarrow{A'} y, x \xrightarrow{A''} w \vdash \Delta_1, y: B', w: B''} (\text{ContrL}) \\
 \frac{\Gamma_1, x: A \Rightarrow B, x \xrightarrow{A'} y, x \xrightarrow{A''} w \vdash \Delta_1, y: B', w: B''} (\Rightarrow R)}{\Gamma_1, x: A \Rightarrow B, x \xrightarrow{A'} y \vdash \Delta_1, y: B', x: A'' \Rightarrow B''} (\Rightarrow R) \\
 \frac{\Gamma_1, x: A \Rightarrow B, x \xrightarrow{A'} y \vdash \Delta_1, y: B', x: A'' \Rightarrow B''} (\Rightarrow R)}{\Gamma_1, x: A \Rightarrow B \vdash \Delta_1, x: A' \Rightarrow B', x: A'' \Rightarrow B''} (\Rightarrow R) \\
 \frac{\Gamma_1, x: A \Rightarrow B \vdash \Delta_1, x: A' \Rightarrow B', x: A'' \Rightarrow B''}{\vdash x_0: D} \Pi_0^*
 \end{array}$$

Notice that in Π_A , Π_B and Π_C^* there are no applications of (Contr L) on $x: A \Rightarrow B$. If $x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1, y: B, w: B \vdash \Delta_1, y: B', w: B''$ is not x -branching, then we can apply the Proposition 4.17 to $x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1, y: B, w: B \vdash \Delta_1, y: B', w: B''$, obtaining that one of the following sequents is derivable:

- (a) $(\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B)_y^* \vdash \Delta_{1y}^*, y: B'$
- (b) $(\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B)_w^* \vdash \Delta_{1w}^*, w: B''$
- (c) $(\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B) - ((\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B)_y^\circ \cup (\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B)_w^\circ) \vdash \Delta_1 - (\Delta_{1y}^\circ \cup \Delta_{1w}^\circ)$

We observe that $x \xrightarrow{A'} y$ and $y: B$ are both members of $(\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B)_y^*$, whereas they are *not* members of the multiset $(\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B)_w^*$; vice versa for the formulas $x \xrightarrow{A''} w$ and $w: B$. Then we have that one of the following sequents is derivable:

- (a) $\Gamma_{1y}^*, x \xrightarrow{A'} y, y: B \vdash \Delta_{1y}^*, y: B'$
- (b) $\Gamma_{1w}^*, x \xrightarrow{A''} w, w: B \vdash \Delta_{1w}^*, w: B''$
- (c) $\Gamma_1 - (\Gamma_{1y}^\circ \cup \Gamma_{1w}^\circ) \vdash \Delta_1 - (\Delta_{1y}^\circ \cup \Delta_{1w}^\circ)$

In each of these cases, we can obtain a proof without adding any contraction:

(a) we have the following proof:

$$\begin{array}{c}
\Pi^\circ \\
\frac{\Pi_A \quad \frac{\Gamma_{1y}^*, x \xrightarrow{A'} y, y: B \vdash \Delta_{1y}^*, y: B'}{\Gamma_1, x \xrightarrow{A'} y, y: B \vdash \Delta_1, y: B'} (Weak)}{x \xrightarrow{A'} y \vdash x \xrightarrow{A} y} (\Rightarrow L)}{\Gamma_1, x \xrightarrow{A'} y, x: A \Rightarrow B \vdash \Delta_1, y: B'} (Weak)} \\
\frac{\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, x: A \Rightarrow B \vdash \Delta_1, y: B', w: B''}{\Gamma_1, x: A \Rightarrow B, x \xrightarrow{A'} y \vdash \Delta_1, y: B', x: A'' \Rightarrow B''} (\Rightarrow R)}{\Gamma_1, x: A \Rightarrow B \vdash \Delta_1, x: A' \Rightarrow B', x: A'' \Rightarrow B''} (\Rightarrow R)}{\Pi_0^*} \\
\vdash x_0: D
\end{array}$$

- (b) symmetric to the previous case;
(c) in this case, we obtain:

$$\begin{array}{c}
\Pi^\circ \\
\frac{\Gamma_1 - (\Gamma_{1y}^\circ \cup \Gamma_{1w}^\circ) \vdash \Delta_1 - (\Delta_{1y}^\circ \cup \Delta_{1w}^\circ)}{\Gamma_1 \vdash \Delta_1} (Weak)}{\Gamma_1, x: A \Rightarrow B \vdash \Delta_1, x: A' \Rightarrow B', x: A'' \Rightarrow B''} (Weak)}{\Pi_0^*} \\
\vdash x_0: D
\end{array}$$

If $x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1, y: B, w: B \vdash \Delta_1, y: B', w: B''$ is x -branching, we can permute all the rules introducing x -branching formulas over the others of the subtree Π_C^* , in a similar way to the proof of the disjunction property. We then apply the Proposition 4.17 to the non x -branching sequents $\Gamma_{1i}, x \xrightarrow{A'} y, x \xrightarrow{A''} w, y: B, w: B \vdash \Delta_{1i}, y: B', w: B''$, obtaining a derivation for one of the following:

- (a) $\Gamma_{1iy}^*, x \xrightarrow{A'} y, y: B \vdash \Delta_{1iy}^*, y: B'$
(b) $\Gamma_{1iw}^*, x \xrightarrow{A''} w, w: B \vdash \Delta_{1iw}^*, w: B''$
(c) $\Gamma_{1i} - (\Gamma_{1iy}^\circ \cup \Gamma_{1iw}^\circ) \vdash \Delta_{1i} - (\Delta_{1iy}^\circ \cup \Delta_{1iw}^\circ)$

In each case (proceeding as in the previous case) we can obtain a proof of the sequents $\Gamma_{1i}, x \xrightarrow{A'} y, x \xrightarrow{A''} w, x: A \Rightarrow B \vdash \Delta_{1i}, y: B', w: B''$. Be Π_C^{**} the proof obtained by permuting the rules introducing x -branching formulas in Π_C^* ; reapplying all the rules of Π_C^{**} to the sequents $\Gamma_{1i}, x \xrightarrow{A'} y, x \xrightarrow{A''} w, x: A \Rightarrow B \vdash \Delta_{1i}, y: B', w: B''$, removing from the antecedent of each sequent an instance of $y: B$ and $w: B$ and adding an occurrence of $x: A \Rightarrow B$, we have a proof $\Pi^{\circ\circ}$ of the sequent:

$$\Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, x: A \Rightarrow B \vdash \Delta_1, y: B', w: B''$$

from which we can obtain:

$$\begin{array}{c}
 \Pi^{\circ\circ} \\
 \hline
 \Gamma_1, x \xrightarrow{A'} y, x \xrightarrow{A''} w, x: A \Rightarrow B \vdash \Delta_1, y: B', w: B'' \quad (\Rightarrow R) \\
 \hline
 \Gamma_1, x \xrightarrow{A'} y, x: A \Rightarrow B \vdash \Delta_1, y: B', x: A'' \Rightarrow B'' \quad (\Rightarrow R) \\
 \hline
 \Gamma_1, x: A \Rightarrow B \vdash \Delta_1, x: A' \Rightarrow B', x: A'' \Rightarrow B'' \\
 \Pi_0 \\
 \vdash x_0: D
 \end{array}$$

(2) *Inductive step: rank>2*: the situation is as follows:

$$\begin{array}{c}
 \Pi_B \qquad \qquad \qquad \Pi_E \\
 \hline
 x \xrightarrow{A''} w \vdash x \xrightarrow{A} w \quad x \xrightarrow{A''} w, \Gamma_2, w: B \vdash \Delta_2 \quad (\Rightarrow L) \\
 \hline
 x \xrightarrow{A''} w, \Gamma_2, x: A \Rightarrow B \vdash \Delta_2 \\
 \Pi_A \qquad \qquad \qquad \Pi_D \\
 \hline
 x \xrightarrow{A'} y \vdash x \xrightarrow{A} y \quad x \xrightarrow{A'} y, \Gamma_1, y: B, x: A \Rightarrow B \vdash \Delta_1 \quad (\Rightarrow L) \\
 \hline
 x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta_1 \quad (ContrL) \\
 \hline
 x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B \vdash \Delta_1 \\
 \Pi_0 \\
 \vdash x_0: D
 \end{array}$$

We proceed in order to reduce the rank of the contraction and then apply the inductive hypothesis; we have the following two subcases:

(1) $x \xrightarrow{A''} w \in \Gamma_1$: in this case Π_D is not empty, (otherwise rank=2). we can permute the upper $(\Rightarrow L)$ over the rules in Π_D , reducing the rank;

(2) the transition formula $x \xrightarrow{A''} w$ is introduced by $(\Rightarrow R)$ in Π_D , then we have a proof like:

$$\begin{array}{c}
\frac{\frac{\frac{\Pi_B}{x \xrightarrow{A''} w \vdash x \xrightarrow{A} w} \quad \frac{\Pi_E}{x \xrightarrow{A''} w, \Gamma_2, w: B \vdash \Delta_2}}{x \xrightarrow{A''} w, \Gamma_2, x: A \Rightarrow B \vdash \Delta_2} (\Rightarrow L)}{\Pi_D''} \\
\frac{\frac{\frac{\Pi_A}{x \xrightarrow{A'} y \vdash x \xrightarrow{A} y} \quad \frac{\Pi_D'}{x \xrightarrow{A'} y, \Gamma_1, y: B, x: A \Rightarrow B \vdash \Delta_1}}{x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta_1} (\Rightarrow R)}{\frac{\frac{\Pi_D''}{x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta_1} (\text{ContrL})}{x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B \vdash \Delta_1}}{\Pi_0} \\
\vdash x_0: D
\end{array}$$

Π_D'' could be empty: if it is not, we can easily diminish the rank of the contraction by permuting the upper $(\Rightarrow L)$ over the rules of Π_D'' ; then, we consider the most difficult case that Π_D'' is empty. We observe that:

- (a) x , y and w are all distinct;
- (b) $x: A'' \Rightarrow B''$ is not a subformula of $y: B$, as x is a predecessor of y ; it is necessarily a subformula of a formula in Γ_1 or in Δ_1 .

We can divide the subtree Π_D' in two subproofs, $\Pi_D'^a$ and $\Pi_D'^b$, such that $\Pi_D'^b$ introduces $y: B$ and $x \xrightarrow{A'} y$ ⁸, whereas the formula $x: A'' \Rightarrow B''$ is used as a premise of a rule in $\Pi_D'^a$. Due this separation, we can permute $(\Rightarrow R)$ over the

⁸If $\Pi_D'^b$ is empty, we have that $y: B$ and $x \xrightarrow{A'} y$ are both in Γ_1 .

lowest ($\Rightarrow L$), obtaining the following proof:

$$\begin{array}{c}
 \frac{\frac{\frac{\frac{\Pi_B}{x \xrightarrow{A''} w \vdash x \xrightarrow{A} w} \quad \frac{\Pi_E}{x \xrightarrow{A''} w, \Gamma_2, y: B \vdash \Delta_2}}{\frac{x \xrightarrow{A''} w, \Gamma_2, x: A \Rightarrow B \vdash \Delta_2}}{(\Rightarrow L)} \quad \Pi_D''}{x \xrightarrow{A''} w, \Gamma_2', x: A \Rightarrow B \vdash \Delta_2', w: B''} \quad \Pi_D'^b}{\frac{\frac{\frac{\Pi_A}{x \xrightarrow{A'} y \vdash x \xrightarrow{A} y} \quad \frac{x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1^*, y: B, x: A \Rightarrow B \vdash \Delta_2^*, w: B''}}{(\Rightarrow L)} \quad \Pi_D'^a}{\frac{x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1^*, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta_1^*, w: B''}}{(\Rightarrow R)} \quad \Pi_D'^a}{x \xrightarrow{A'} y, \Gamma_1^*, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta_1^*, x: A'' \Rightarrow B''} \quad \Pi_D'^a}{\frac{x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta_1}}{(\text{ContrL})} \quad \Pi_0}{x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B \vdash \Delta_1} \quad \Pi_0}{\vdash x_0: D}
 \end{array}$$

Now we can permute ($\Rightarrow R$) over the contraction rule, obtaining the tree:

$$\begin{array}{c}
 \frac{\frac{\frac{\frac{\Pi_B}{x \xrightarrow{A''} w \vdash x \xrightarrow{A} w} \quad \frac{\Pi_E}{x \xrightarrow{A''} w, \Gamma_2, y: B \vdash \Delta_2}}{\frac{x \xrightarrow{A''} w, \Gamma_2, x: A \Rightarrow B \vdash \Delta_2}}{(\Rightarrow L)} \quad \Pi_D''}{x \xrightarrow{A''} w, \Gamma_2', x: A \Rightarrow B \vdash \Delta_2', w: B''} \quad \Pi_D'^b}{\frac{\frac{\frac{\Pi_A}{x \xrightarrow{A'} y \vdash x \xrightarrow{A} y} \quad \frac{x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1^*, y: B, x: A \Rightarrow B \vdash \Delta_2^*, w: B''}}{(\Rightarrow L)} \quad \Pi_D'^a}{\frac{x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1^*, x: A \Rightarrow B, x: A \Rightarrow B \vdash w: B'', \Delta_1^*}}{(\text{ContrL})} \quad \Pi_D'^a}{\frac{x \xrightarrow{A'} y, x \xrightarrow{A''} w, \Gamma_1^*, x: A \Rightarrow B \vdash w: B'', \Delta_1^*}}{(\Rightarrow R)} \quad \Pi_D'^a}{x \xrightarrow{A'} y, \Gamma_1^*, x: A \Rightarrow B \vdash \Delta_1^*, x: A'' \Rightarrow B''} \quad \Pi_D'^{a*}}{x \xrightarrow{A'} y, \Gamma_1, x: A \Rightarrow B \vdash \Delta} \quad \Pi_0}{\vdash x_0: D}
 \end{array}$$

formula F n times, $n > 1$.

DEFINITION 4.25 REDUNDANT CONTRACTIONS ON A FORMULA F . Given a proof tree Π , we say that it has redundant contractions on a formula F if it has a branch \mathcal{B} with at least one of the following conditions:

- (1) a contraction rule is applied to a non conditional formula F ;
- (2) the right contraction rule (*Contr R*) is applied to a conditional formula F ;
- (3) there are multiple contractions of (*Contr L*) on a conditional formula F .

DEFINITION 4.26 INTEGER MULTISSET ORDERING $<_m$. Given Γ and Δ , multisets of integers, we say that:

- (1) $\Gamma <_m \Delta$ if $\Gamma \subset \Delta$;
- (2) $\Gamma <_m \Delta$ if $\Gamma <_m \Delta'$, where $\Delta' = \Delta - \{j\} \cup \{i, i, \dots, i\}$ and $i < j$.

As it is well known, $<_m$ is a well-order on multisets.

In the proof for bounding contractions in SeqMP and SeqID+MP we need the following:

LEMMA 4.27. If $\Gamma, x : A \Rightarrow B, x : A \Rightarrow B \vdash \Delta, x : A$ has a derivation Π with no contractions on $x : A \Rightarrow B$ in SeqMP (SeqID+MP), then $\Gamma, x : A \Rightarrow B \vdash \Delta, x : A$ is derivable in SeqMP (SeqID+MP) and it has a derivation which at most adds to the contractions in Π contractions on formulas with lower complexities than the complexity of $x : A \Rightarrow B$.

Proof. If an instance of $x : A \Rightarrow B$ is introduced in Π by implicit weakening, then we can conclude by deleting this weakening. If the two instances of $x : A \Rightarrow B$ are both derived from (\Rightarrow L) using transitions of the form $x \xrightarrow{A} y$, with $x \neq y$, then the contraction can be eliminated as in CK{+ID}. If an instance of $x : A \Rightarrow B$ is introduced in Π by an application of (\Rightarrow L) using a transition $x \xrightarrow{A} x$ derived from (MP), then it permutes over the other rules in Π ; therefore, we have the following proof:

$$\frac{\frac{\frac{\Pi_A}{\Gamma, x : A \Rightarrow B \vdash \Delta, x : A, x : A} \quad (MP)}{\Gamma, x : A \Rightarrow B \vdash \Delta, x : A, x \xrightarrow{A} x} \quad \frac{\Pi_B}{\Gamma, x : A \Rightarrow B, x : B \vdash \Delta, x : A}}{\Gamma, x : A \Rightarrow B, x : A \Rightarrow B \vdash \Delta, x : A} (\Rightarrow L)$$

We can conclude the proof by adding a contraction on the sub-formula $x : A$ as follows:

$$\frac{\frac{\Pi_A}{\Gamma, x : A \Rightarrow B \vdash \Delta, x : A, x : A}}{\Gamma, x : A \Rightarrow B \vdash \Delta, x : A} (ContrR)$$

□

DEFINITION 4.28 $c(\Pi)$. Given a proof tree Π we define $c(\Pi)$ as the multiset of integers of the complexities of the formulas to which are applied redundant contractions in Π .

For example, consider a proof tree Π having three branches with the following features: a contraction on a formula $x : A \rightarrow B$ with complexity 7 and two applications of (Contr L) on $x : C \Rightarrow D$ with complexity 5 in the left branch; an application of (Contr L) on a conditional formula $x : E \Rightarrow F$ with complexity 9 in the central branch; two contractions on an atomic formula (complexity 2) and three contractions (Contr L) on $x : G \Rightarrow H$ with complexity 5 in the right branch. We have that $c(\Pi) = \{7, 5, 2, 5\}$ (the contraction in the central branch is not redundant).

We can control the application of the contractions in SeqMP and SeqID+MP as explained by the following:

THEOREM 4.29 BOUND FOR THE CONTRACTIONS IN SEQMP AND SEQID+MP.
Given a sequent $\vdash x_0 : D$, derivable in SeqMP or in SeqID+MP, then it has a derivation where there is at most one application of (Contr L) on each conditional formula $x : A \Rightarrow B$ in every branch of the proof tree. In other words, we can find a derivation of it with no redundant contractions.

Proof (Sketch). Using the results of the previous subsections, we can say, without loss of generality, that the sequent $\vdash x_0 : D$ has a derivation Π with no applications of (Contr R) on conditional formulas and no contractions on transitions. Then, we proceed by induction on the multiset ordering defined by $c(\Pi)$. The idea is the following: at each step of the proof, we build a proof tree Π' such that $c(\Pi') <_m c(\Pi)$, to which we apply the inductive hypothesis; Π' is obtained by removing the redundant contraction on a formula with complexity M such that $M = \max(c(\Pi))$. We can have two different situations:

- (1) the redundant contractions on formulas with complexity M are deleted *without adding any other contraction in the proof tree*; therefore, we have that $c(\Pi') = c(\Pi) - \{M\}$;
- (2) the redundant contractions on formulas with complexity M are replaced by redundant contractions on formulas with complexities M_1, M_2, \dots, M_k , where $M_i < M$. $c(\Pi')$ is then obtained by replacing an occurrence of M in $c(\Pi)$ with the values M_1, M_2, \dots, M_k .

We only present the most interesting case; the entire proof is contained in [Pozzato 2003], pages 163-198.

Consider a branch \mathcal{B} with n applications of (Contr L), $n > 1$, on $x : A \Rightarrow B$, where $\max(c(\Pi)) = \text{cp}(x : A \Rightarrow B)$; we can consider that all the n contractions are applied in sequence, then we consider the upper two instances of (Contr L) and proceed to eliminate the upper one:

$$\frac{\frac{\frac{\Pi}{\Gamma, x : A \Rightarrow B, x : A \Rightarrow B, x : A \Rightarrow B \vdash \Delta} \text{(ContrL)}}{\Gamma, x : A \Rightarrow B, x : A \Rightarrow B \vdash \Delta} \text{(ContrL)}}{\Gamma, x : A \Rightarrow B \vdash \Delta} \text{(ContrL)}$$

$$\frac{\Pi_0}{\vdash x_0 : D}$$

Π does not contain any application of (Contr L) with constituent $x : A \Rightarrow B$. We can also observe that an instance of $x : A \Rightarrow B$ is the principal formula of the

sequent, by the permutability of the rules⁹. We present the following situation: the $(\Rightarrow L)$ rule is applied to $\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta$ using the label x :

$$\begin{array}{c}
\frac{\frac{\frac{\frac{\Pi_1}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta, x \xrightarrow{A} x} \quad \frac{\Pi_2}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: B \vdash \Delta}}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta} (\Rightarrow L)}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta} (\text{ContrL})}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta} (\text{ContrL})}{\Gamma, x: A \Rightarrow B \vdash \Delta} (\text{ContrL})}{\Pi_0} \vdash x_0: D
\end{array}$$

If the transition formula $x \xrightarrow{A} x$ is introduced by implicit weakening, then we have that $\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta$ is derivable and we can immediately conclude (the upper (Contr L) has been eliminated); if $x \xrightarrow{A} x$ is not introduced by weakening, it can only be treated (looking backward) by the (MP) rule, then we consider the following proof:

$$\begin{array}{c}
\frac{\frac{\frac{\frac{\Pi'_1}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta, x: A} \quad \frac{\Pi_2}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: B \vdash \Delta}}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash x \xrightarrow{A} x, \Delta} (\text{MP})}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta} (\Rightarrow L)}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta} (\text{ContrL})}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta} (\text{ContrL})}{\Gamma, x: A \Rightarrow B \vdash \Delta} (\text{ContrL})}{\Pi_0} \vdash x_0: D
\end{array}$$

By Lemma 4.27, $\Gamma, x: A \Rightarrow B \vdash \Delta, x: A$ is derivable with a proof Π° that does not add any contraction on $x: A \Rightarrow B$; we can obtain:

$$\frac{\Pi^\circ}{\Gamma, x: A \Rightarrow B \vdash \Delta, x: A} (\text{MP})$$

$$\frac{\Gamma, x: A \Rightarrow B \vdash \Delta, x: A}{\Gamma, x: A \Rightarrow B \vdash \Delta, x \xrightarrow{A} x}$$

Our target is now to find a proof tree Π_2^* , with no contractions on $x: A \Rightarrow B$, of the sequent $\Gamma, x: A \Rightarrow B, x: B \vdash \Delta$. Consider the case when both the occurrences of $x: A \Rightarrow B$ are introduced in Π_2 by the $(\Rightarrow L)$ rule. We analyze the situation in which an occurrence of $x: A \Rightarrow B$ is introduced by $(\Rightarrow L)$ by using the label x ; we

⁹As we explained above, $(\Rightarrow L)$ does not always permute over the $(\Rightarrow R)$ rule. We can assume, without loss of generality, that $x: A \Rightarrow B$ is the principal formula since, if all its three instances are derived by $(\Rightarrow L)$ and the labels used are introduced by $(\Rightarrow R)$ in Π , then we can permute $(\Rightarrow R)$ over the contractions; therefore, $(\Rightarrow L)$ can be permuted over the rules of Π .

can then consider the following proof tree:

$$\frac{\frac{\frac{\Pi_A}{\Gamma, x: A \Rightarrow B, x: B \vdash \Delta, x: A} (MP)}{\Gamma, x: A \Rightarrow B, x: B \vdash \Delta, x \xrightarrow{A} x} \quad \frac{\Pi_B}{\Gamma, x: A \Rightarrow B, x: B, x: B \vdash \Delta}}{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B, x: B \vdash \Delta} (\Rightarrow L)$$

Π_A and Π_B do not contain any application of (Contr L) on $x: A \Rightarrow B$, then we have the following proof:

$$\frac{\Pi_B}{\Gamma, x: A \Rightarrow B, x: B, x: B \vdash \Delta} (ContrL)$$

The other cases, when both the conditionals $x: A \Rightarrow B$ are introduced without using the label x in the applications of ($\Rightarrow L$), are left to the reader.

We have found two proofs, with no contractions on $x: A \Rightarrow B$, at most introducing contractions on formulas with lower complexity than the complexity of $x: A \Rightarrow B$, of the sequents $\Gamma, x: A \Rightarrow B \vdash x \xrightarrow{A} x, \Delta$ and $\Gamma, x: A \Rightarrow B, x: B \vdash \Delta$. We can then obtain the following proof, erasing the upper application of (Contr L) in the initial proof tree :

$$\frac{\frac{\frac{\Pi^\circ}{\Gamma, x: A \Rightarrow B \vdash \Delta, x: A} (MP)}{\Gamma, x: A \Rightarrow B \vdash \Delta, x \xrightarrow{A} x} \quad \frac{\Pi_2^*}{\Gamma, x: A \Rightarrow B, x: B \vdash \Delta}}{\frac{\Gamma, x: A \Rightarrow B, x: A \Rightarrow B \vdash \Delta}{\Gamma, x: A \Rightarrow B \vdash \Delta} (ContrL)} (\Rightarrow L)$$

$$\frac{\Pi_0}{\vdash x_0: D}$$

□

4.4 Reformulation of SeqMP and SeqID+MP

In the previous subsection we have found that, in SeqMP and SeqID+MP, the left contraction rule (Contr L) is only needed if applied to conditional formulas $x: A \Rightarrow B$ and, in particular, *at most once on each formula occurring in every derivation branch of a proof tree*. By this fact, we can reformulate the calculi for these two systems, obtaining **BSeqMP** and **BSeqID+MP** (the prefix B stands for "bounded contractions") with the following features:

- (1) proof trees do not contain redundant contractions;
- (2) contractions on $x: A \Rightarrow B$ are *absorbed* into the ($\Rightarrow L$) rule.

First of all, we represent a single node of a proof tree as

$$K \mid \Psi \mid \Gamma \vdash \Delta$$

K is the set containing all the conditional formulas that have already been contracted in that branch of the proof tree. The antecedent of a sequent is then split into two parts:

$$\begin{array}{c}
 (\Rightarrow \mathbf{L})_1 \frac{K \cup \{x : A \Rightarrow B\} \mid \Psi \cup \{x : A \Rightarrow B\} \mid \Gamma' \vdash x \xrightarrow{A} y, \Delta}{K \cup \{x : A \Rightarrow B\} \mid \Psi \cup \{x : A \Rightarrow B\} \mid y : B, \Gamma' \vdash \Delta} \quad , \text{if } x : A \Rightarrow B \notin K \\
 \\
 (\Rightarrow \mathbf{L})_2 \frac{K \mid \Psi \mid \Gamma' \vdash x \xrightarrow{A} y, \Delta \quad K \mid \Psi \mid y : B, \Gamma' \vdash \Delta}{K \mid \Psi \mid x : A \Rightarrow B, \Gamma' \vdash \Delta} \quad , \text{if } x : A \Rightarrow B \in K \\
 \\
 (\Rightarrow \mathbf{L})_3 \frac{K' \cup \{x : A \Rightarrow B\} \mid \Psi' \mid \Gamma \vdash x \xrightarrow{A} y, \Delta}{K' \cup \{x : A \Rightarrow B\} \mid \Psi' \mid y : B, \Gamma \vdash \Delta} \\
 \frac{K' \cup \{x : A \Rightarrow B\} \mid \Psi' \cup \{x : A \Rightarrow B\} \mid \Gamma \vdash \Delta}{K' \cup \{x : A \Rightarrow B\} \mid \Psi' \cup \{x : A \Rightarrow B\} \mid \Gamma \vdash \Delta} \\
 (\mathbf{EQ}) \frac{\emptyset \mid \emptyset \mid u : A \vdash u : B \quad \emptyset \mid \emptyset \mid u : B \vdash u : A}{K \mid \Psi \mid x \xrightarrow{A} y, \Gamma' \vdash x \xrightarrow{B} y, \Delta'}
 \end{array}$$

Fig. 5. Sequent calculi BSeqMP and BSeqID+MP.

- (1) the set Ψ of the conditional formulas duplicated by contraction;
- (2) the multiset Γ with the other formulas.

$(\Rightarrow \mathbf{L})$ is split in three rules:

- (1) $(\Rightarrow \mathbf{L})_1$ is applied to $x : A \Rightarrow B \in \Gamma$ if $x : A \Rightarrow B$ does not belong to K , i.e. if this conditional formula has not yet been contracted in that branch. The principal formula $x : A \Rightarrow B$ is decomposed and a copy of it is added to Ψ and to K ;
- (2) $(\Rightarrow \mathbf{L})_2$ is applied to $x : A \Rightarrow B \in \Gamma$ if $x : A \Rightarrow B$ belongs to K , i.e. it has already been contracted in that branch. The conditional formula $x : A \Rightarrow B$ is decomposed without adding any copy of it in the auxiliary sets Ψ and K ;
- (3) $(\Rightarrow \mathbf{L})_3$ is applied to $x : A \Rightarrow B \in \Psi$, i.e. $x : A \Rightarrow B$ has been previously duplicated by an application of $(\Rightarrow \mathbf{L})_1$. The principal formula $x : A \Rightarrow B$ is decomposed without adding any copy of it in the auxiliary sets Ψ and K .

In other words, if a conditional formula $x : A \Rightarrow B$ in Γ has not been duplicated in a branch, then it is decomposed by $(\Rightarrow \mathbf{L})_1$, which adds a copy of it in K and in Ψ . K keeps trace of duplicated formulas in that branch. Duplicated conditionals in Ψ will be only decomposed, but no duplicated. If $x : A \Rightarrow B$ in Γ has already been duplicated in a branch, i.e. $x : A \Rightarrow B$ belongs to K , it is only decomposed and no further duplicated.

Sequent calculi BSeqMP and BSeqID+MP are shown in Figure 5; we omit the reformulation for axioms and some rules, since they are identical to SeqS's rules (see Figures 1 and 2) with the exception of the form of the sequents.

Next theorem follows immediately from the above reformulation:

THEOREM 4.30. *A sequent $\Gamma \vdash \Delta$ is derivable in SeqMP (SeqID+MP) if and only if $\emptyset \mid \emptyset \mid \Gamma \vdash \Delta$ is derivable in BSeqMP (BSeqID+MP).*

We give a derivation in BSeqMP of the sequent $x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B)) \vdash$, introduced at the beginning of the previous subsection as an example of sequent derivable in SeqMP with a necessary application of the left contraction on conditionals:

$$\frac{\frac{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \vdash x \xrightarrow{\top} x}{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid x : B \wedge \neg(\top \Rightarrow B) \vdash}}{\emptyset \mid \emptyset \mid x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B)) \vdash} (\Rightarrow L)_1$$

The upper premise is derived as follows:

$$\frac{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \vdash x : \top}{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \vdash x \xrightarrow{\top} x} (MP)$$

The other one has the following derivation:

$$\frac{\frac{\frac{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \emptyset \mid x \xrightarrow{\top} y, x : B, y : B, y : \neg(\top \Rightarrow B) \vdash y : B}{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \emptyset \mid x \xrightarrow{\top} y, x : B, y : B \wedge \neg(\top \Rightarrow B) \vdash y : B} (\wedge L)}{\frac{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid x \xrightarrow{\top} y, x : B \vdash y : B}{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid x : B \vdash x : \top \Rightarrow B} (\Rightarrow R)}{\frac{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid x : B \vdash x : \top \Rightarrow B}{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid x : B, x : \neg(\top \Rightarrow B) \vdash} (\neg L)}{\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid x : B \wedge \neg(\top \Rightarrow B) \vdash} (\wedge L)} \Pi (\Rightarrow L)_3$$

where Π is the axiom $\{x : \top \Rightarrow (B \wedge \neg(\top \Rightarrow B))\} \mid \emptyset \mid x \xrightarrow{\top} y, x : B \vdash x \xrightarrow{\top} y, y : B$.

4.5 Complexity of CK{+ID}

Since we can eliminate contraction, it is relatively easy to prove both decidability and a space complexity bound.

In a proof without contractions, in all rules the premises have a smaller complexity than the conclusion. By this fact we get that the length of each branch in a proof of a sequent $\vdash x_0 : D$ is bounded by $O(|D|)$.

Moreover, observe that the rules are analytic, so that the premises contains only (labelled) subformulas of the formulas in the conclusion. In the search of a proof of $\vdash x_0 : D$, with $|D| = n$, new labels are introduced only by (positive) conditional subformulas of D . Thus, the number of different labels occurring in a proof is $O(n)$; it follows that the total number of distinct labelled formulas is $O(n^2)$, and only $O(n)$ of them can actually occur in each sequent.

This itself gives decidability:

THEOREM 4.31 CK{+ID} DECIDABILITY. *Logic CK{+ID} is decidable.*

Proof. We just observe that there is only a finite number of derivations to check of a given sequent $\vdash x_0 : D$, as both the length of a proof and the number of labelled formulas which may occur in it is finite. □

Notice that, as usual, a proof may have an exponential size since the branching introduced by the rules. However we can obtain a much sharper space complexity

bound using a standard technique [Hudelmaier 1993; Viganò 2000], namely we do not need to store the whole proof, but only a sequent at a time plus additional information to carry on the proof search. In searching a proof there are two kinds of branching to consider: AND-branching caused by the rules with multiple premises and OR-branching (backtracking points in a depth first search) caused by the choice of the rule to apply, and how to apply it in the case of (\Rightarrow L).

We store only one sequent at a time and maintain a stack containing information sufficient to reconstruct the branching points of both types. Each stack entry contains the principal formula (either a labelled sentence $x; B$, or a transition formula $x \xrightarrow{B} y$), the name of the rule applied and an index which allows to reconstruct the other branches on return to the branching points. The stack entries represent thus backtracking points and the index within the entry allows one to reconstruct both the AND branching and to check whether there are alternatives to explore (OR branching). The working sequent on a return point is recreated by replaying the stack entries from the bottom of the stack using the information in the index (for instance in the case of (\Rightarrow L) applied to the principal formula $x : A \Rightarrow B$, the index will indicate which premise-first or second-we have to expand and the label y involved in the transition formula $x \xrightarrow{A} y$).

A proof begins with the end sequent $\vdash x_0 : D$ and the empty stack. Each rule application generates a new sequent and extends the stack. If the current sequent is an axiom we pop the stack until we find an AND branching point to be expanded. If there are not, the end sequent $\vdash x_0 : D$ is provable and we have finished. If the current sequent is not an axiom and no rule can be applied to it, we pop the stack entries and we continue at the first available entry with some alternative left (a backtracking point). If there are no such entries, the end sequent is not provable.

The entire process must terminate since: (i) the depth of the stack is bounded by the length of a branch proof, thus it is $O(n)$, where $|D| = n$, (ii) the branching is bounded by the number of rules, the number of premises of any rule and the number of formulas occurring in one sequent, the last being $O(n)$.

To evaluate the space requirement, we have that each subformula of the initial formula can be represented by a positional index into the initial formula, which requires $O(\log n)$ bits. Moreover, also each label can be represented by $O(\log n)$ bits. Thus, to store the working sequent we need $O(n \log n)$ space, since there may occur $O(n)$ labelled subformulas. Similarly, each stack entry requires $O(\log n)$ bits, as the name of the rule requires constant space and the index $O(\log n)$ bits. Having depth $O(n)$, to store the whole stack requires $O(n \log n)$ space. Thus we obtain:

THEOREM 4.32 SPACE COMPLEXITY OF $CK\{+ID\}$. *Provability in $CK\{+ID\}$ is decidable in $O(n \log n)$ space.*

4.6 Decidability and Complexity of $CK+MP\{+ID\}$

Let us conclude this section with a quick analysis of the systems $CK+MP\{+ID\}$. We explained above that we cannot eliminate the (Contr L) rule on conditional formulas in SeqMP and SeqID+MP systems; however, we can control its application, in order to obtain a decision procedure for these logics, too.

We can prove the following:

THEOREM 4.33 $CK+MP\{+ID\}$ DECIDABILITY. *Logic $CK+MP\{+ID\}$ is decidable.*

Proof. It is easy to prove that BSeqMP and BSeqID+MP terminate, from which we obtain a decision procedure for logics CK+MP and CK+MP+ID. As we explained above, the (Contr L) rule is absorbed by the (\Rightarrow L) rule; a conditional formula $x : A \Rightarrow B$ is contracted at most one time on each branch with an application of (\Rightarrow L)₁ and (eventually) the following application of (\Rightarrow L)₃, therefore the length of each branch in a proof is limited, as only a finite number of conditional formulas can be introduced in that branch. □

Let us say something about the space complexity of BSeqMP and BSeqID+MP. These systems are decidable in exponential space in the number of nested conditional formulas. Consider, for example, an application of the (\Rightarrow L)₁ rule with principal formula $x : A_1 \Rightarrow (A_2 \Rightarrow B)$, which introduces the subformula $y : A_2 \Rightarrow B$ in the antecedent of one premise and is duplicated in K ; if the (\Rightarrow L)₁ rule is also applied to $y : A_2 \Rightarrow B$, then it is duplicated in K , too, obtaining two copies of $x : A_1 \Rightarrow (A_2 \Rightarrow B)$ and two of $y : A_2 \Rightarrow B$. And so on, for every nested conditional formula (if B is a conditional formula $C \Rightarrow D$, eight formulas are generated).

A more detailed analysis on the structure of conditionals in the initial sequent is needed to refine these complexity results. By the relation between CK+MP and modal logic T, we strongly conjecture that provability for both BSeqMP and BSeqID+MP is decidable in $O(n^2 \log n)$ -space. Viganò presents a substructural analysis for modal system T (chapter 10 of [Viganò 2000]) that could inspire a refinement of the exponential complexity bound we obtained by the previous informal discussion.

5. CONDLEAN: A THEOREM PROVER FOR CONDITIONAL LOGICS

In this section we present CondLean, a theorem prover implementing the sequent calculi SeqS; it is a SICStus Prolog program inspired by leanTAP [Beckert and Posegga 1995]. The program comprises a set of clauses, each one of them represents a sequent rule or axiom. The proof search is provided for free by the mere depth-first search mechanism of Prolog, without any additional ad hoc mechanism. CondLean is available for free download at <http://www.di.unito.it/~olivetti/CONDLEAN>.

We represent each component of a sequent (antecedent and consequent) by a *list* of formulas, partitioned into three sub-lists: atomic formulas, transitions and complex formulas. Atomic and complex formulas are represented by a list like $[x, a]$, where x is a Prolog constant and a is a formula. A transition $x \xrightarrow{A} y$ is represented by $[x, a, y]$. For example, the sequent $x : A \Rightarrow B, x : A \Rightarrow C, x \xrightarrow{A} y \vdash y : B, x : C, x : A \rightarrow B$ is represented by the following lists (the upper one represents the antecedent, the lower one represents the consequent):

$$\begin{aligned} & [[], [[x, a, y]], [[x, a \Rightarrow b], [x, a \Rightarrow c]]] \\ & [[y, b], [x, c]], [], [[x, a \rightarrow b]] \end{aligned}$$

We present three different implementations:

- (1) a *constant labels* version;

- (2) a *free-variables* version;
- (3) an heuristic version.

The *constant labels* version makes use of Prolog constants to represent SeqS's labels. The sequent calculi are implemented by the predicate

prove(Sigma, Delta, Labels).

This predicate succeeds if and only if $\Sigma \vdash \Delta$ is derivable in SeqS, where **Sigma** and **Delta** are the lists representing the multisets Σ and Δ , respectively and **Labels** is the list of labels introduced in that branch. For example, to prove

$$x: A \Rightarrow (B \wedge C)^{10} \vdash x: A \Rightarrow B, x: C$$

in CK, one queries `CondLean` with the goal:

```
prove([[ ], [], [[x, a=>(b and c)]]], [[x,c], [], [[x, a=>b]]], [x]).
```

Each clause of `prove` implements one axiom or rule of SeqS; for example, the clause implementing $(\Rightarrow L)$ is:

```
prove([LitSigma,TransSigma,ComplexSigma],[LitDelta,TransDelta,
ComplexDelta], Labels):-
select([X,A=>B],ComplexSigma,ResComplexSigma), member(Y,Labels),
put([Y,B],LitSigma,ResComplexSigma,NewLitSigma,NewComplexSigma),
prove([LitSigma,TransSigma,ResComplexSigma],
[LitDelta,[[X,A,Y]|TransDelta],ComplexDelta],Labels),
prove([NewLitSigma,TransSigma,NewComplexSigma],
[LitDelta,TransDelta,ComplexDelta],Labels).
```

The predicate `select` removes $[X,A \Rightarrow B]$ from `ComplexSigma` returning `ResComplexSigma` as result. The predicate `put` is used to put $[Y,B]$ in the proper sub-list of the antecedent.

To search a derivation of a sequent $\Sigma \vdash \Delta$, `CondLean` proceeds as follows. First of all, if $\Sigma \vdash \Delta$ is an axiom, the goal will succeed immediately by using the clauses for the axioms. If it is not, then the first applicable rule will be chosen, e.g. if `ComplexSigma` contains a formula $[X, A \text{ and } B]$, then the clause for $(\wedge L)$ rule will be used, invoking `prove` on the unique premise of $(\wedge L)$. `CondLean` proceeds in a similar way for the other rules. The ordering of the clauses is such that the application of the branching rules is postponed as much as possible. When the $(\Rightarrow L)$ clause is used to prove $\Sigma \vdash \Delta$, a backtracking point is introduced by the choice of a label Y occurring in the two premises of the rule; in case of failure, Prolog's backtracking tries every instance of the rule with every available label (if more than one). Choosing, sooner or later, the right label to apply $(\Rightarrow L)$ may strongly affect the theorem prover's efficiency: if there are n labels to choose for an application of $(\Rightarrow L)$ the computation might succeed only after $n-1$ backtracking steps, with a significant loss of efficiency.

Our second implementation, called *free-variables*, makes use of *Prolog variables* to represent all the labels that can be used in a single application of the $(\Rightarrow L)$ rule.

¹⁰`CondLean` extends the sequent calculi to formulas containing also \neg , \wedge , \vee and \top .

This version represents labels by integers starting from 1; by using integers we can easily express constraints on the range of the variable-labels. To this regard the library `clpfd` is used to manage free-variable domains (see [Marriott and Stuckey 1998] and [Jaffar and Michaylov 1987] for details about the constraints satisfaction problems and the constraint logic programming). As an example, in order to prove $\Sigma', 1: A \Rightarrow B \vdash \Delta$ the theorem prover will call `prove` on the following premises: $\Sigma' \vdash \Delta, 1 \xrightarrow{A} V$ and $V: B, \Sigma' \vdash \Delta$, where V is a Prolog variable. This variable will be then instantiated by Prolog's pattern matching to apply either the (EQ) rule, or to *close a branch with an axiom*. Here below is the clause implementing the $(\Rightarrow L)$ rule:

```

prove([LitSigma,TransSigma,ComplexSigma],[LitDelta,
  TransDelta,ComplexDelta],Max):-
  select([X,A => B],ComplexSigma,ResComplexSigma),
  domain([Y],1,Max), Y#>X,
  put([Y,B],LitSigma,ResComplexSigma,NewLitSigma,NewComplexSigma),
  prove([NewLitSigma,TransSigma,NewComplexSigma],
    [LitDelta,TransDelta,ComplexDelta],Max),
  prove([LitSigma,TransSigma,ResComplexSigma],
    [LitDelta,[X,A,Y]|TransDelta],ComplexDelta],Max).

```

The atom `Y#>X` adds the constraint `Y>X` to the constraint store: the constraints solver will verify the consistency of it during the computation. In `SeqCK` and `SeqID` we can only use labels introduced *after* the label `X`, thus we introduce the previous constraint. In `SeqMP` and `SeqID+MP` we can also use `X` itself, thus we shall add the constraint `Y#>=X`.

The third argument of predicate `prove` is `Max` and is used to define variables domains.

On a sequent with 65 labels on the antecedent this version succeeds in 460 mseconds, whereas the constant labels version takes 4326 mseconds.

We have also developed a third version, called *heuristic version*, that performs a "two-phase" computation: in "Phase 1" an *incomplete* theorem prover searches a derivation exploring a *reduced search space*; in case of failure, the free-variables version is called ("Phase 2"). Intuitively, the reduction of the search space in Phase 1 is obtained by committing the choice of the label to instantiate a free variable, whereby blocking the backtracking.

For `SeqMP` and `SeqID+MP`, we have developed a theorem prover which simplifies the reformulations given by `BSeqMP` and `BSeqID+MP`. In particular, the reformulation given in the previous section uses two different auxiliaries sets, namely K and Ψ , whereas the theorem prover implements only Ψ , by introducing another argument `CondContr` to the predicate `prove`; therefore, only $(\Rightarrow L)_1$ and $(\Rightarrow L)_3$ are implemented. The `prove` predicate now becomes:

```

prove(Sigma, Delta, Labels, CondContr).

```

The list `CondContr` stores the conditional formulas of the antecedent that have been duplicated so far. When $(\Rightarrow L)$ is applied to a formula $x: A \Rightarrow B$ in the antecedent, the formula is duplicated at the same time into the `CondContr` list; when $(\Rightarrow L)$ is applied to a formula in `CondContr`, in contrast, the formula is *no longer* duplicated.

Thus the (\Rightarrow L) rule is split in two rules, one taking care of "unused" conditionals of the antecedent, the other taking care of "used" (or duplicated) conditionals. Observe that this ensures termination. To understand the difference between the calculus and CondLean implementation, we can observe that the calculus BSeqMP (BSeqID+MP) ensures that every conditional formula $x : A \Rightarrow B$ is duplicated only once, no matter how many times it occurs in a branch. As a difference, CondLean ensures that every *occurrence* of $x : A \Rightarrow B$ is duplicated at most once. However, we have chosen of implementing this simplified version since it is easier and, at the current state, it is not clear if the exact implementation of BSeqMP (BSeqID+MP) would perform significantly better.

5.1 Performances of the Theorem Prover

The performances of the three versions of the theorem prover are promising even on a small machine. To test our program we used samples generated by modifying the samples from [Beckert and Gorè 1997] and from [Viganò 2000].

We have tested CondLean, SeqCK system, obtaining the following results¹¹:

- (1) the constant labels version succeeds in 79 tests over 90 in less than 2 seconds (78 in less than one second);
- (2) the free-variables version succeeds in 73 tests over 90 in less than 2 seconds (but 67 in less than 10 mseconds);
- (3) the heuristic version succeeds in 78 tests over 90 in less than 2 seconds (70 in less than 500 mseconds).

Considering the sequent-degree (defined as the maximum level of nesting of the \Rightarrow operator) as a parameter, we have the following results, obtained by testing the SeqCK free-variables version:

| | | | | | |
|----------------------------|---|-----|-----|------|------|
| Sequent degree | 2 | 6 | 9 | 11 | 15 |
| Time to succeed (mseconds) | 5 | 500 | 650 | 1000 | 2000 |

We have also tested CondLean, SeqMP system, on some sequents that require duplications of conditional formulas; in particular, we have obtained the following results running the heuristic version on an AMD Athlon XP 2400+ (2.0 GHz), 512 MB RAM machine, using SICStus Prolog 3.11.1:

| | | | | | | | | | | |
|---|---|------|---|---|---|---|---|---|---|----|
| Sequent | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Number of applications of (\Rightarrow L) ₃ | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 3 | 4 | 5 |
| Time to succeed (mseconds) | 1 | 2500 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

As expected, in (MP) systems the free-variables version offers better performances than the constant label version; in the following table we show how many sequents have been derived by each implementation in less than 1 ms, 1 second and 2 seconds over 97 valid sequents:

| | | | |
|-----------------|-----|----|----|
| Time to succeed | 1ms | 1s | 2s |
| Constant labels | 61 | 66 | 67 |
| Free-variables | 75 | 82 | 82 |

¹¹These results are obtained running SICStus Prolog 3.10.0 on an Intel Pentium 166 MMX, 96 MB RAM machine.

6. CONCLUSIONS, COMPARISON WITH OTHER WORKS AND FUTURE WORK

In this work we have provided a labelled calculus for minimal conditional logic CK, and its standard extensions with conditions ID and MP. The calculus is cut-free and analytic. By a proof-theoretical analysis, we have shown that CK and CK+ID are decidable in $O(n \log n)$ space. We have also introduced a decision procedure for CK+MP and CK+MP+ID. To the best of our knowledge, sequent calculi for these logics have not been previously studied and the complexity bound for these conditional systems is new.

We have also developed CondLean, a theorem prover implementing the calculus written in SICStus Prolog.

We briefly remark on some related works. Most of the works have concentrated on extensions of CK.

De Swart [de Swart 1983] and Gent [Gent 1992] give sequent/tableaux calculi for the strong conditional logics VC and VCS. Their proof systems are based on the entrenchment connective \leq , from which the conditional operator can be defined. Their systems are analytic and comprise an infinite set of rules $\leq F(n, m)$, with a uniform pattern, to decompose each sequent with m negative and n positive entrenchment formulas.

Crocco and Fariñas [Crocco and del Cerro 1995] present sequent calculi for some conditional logics including CK, CEM, CO and others. Their calculi comprise two levels of sequents: principal sequents with \vdash_P correspond to the basic deduction relation, whereas auxiliary sequents with \vdash_a correspond to the conditional operator: thus the constituents of $\Gamma \vdash_P \Delta$ are sequents of the form $X \vdash_a Y$, where X, Y are sets of formulas.

Artosi, Governatori, and Rotolo [Artosi et al. 2002] develop labelled tableau for the *first-degree* fragment (i.e. without nested conditionals) of the conditional logic CU that corresponds to cumulative non-monotonic logics. In their work they use labels similarly to ours. Formulas are labelled by path of worlds containing also variable worlds (see also our free-variable implementation). Differently from us, they do not use a specific rule to deal with equivalent antecedents of conditionals. They use instead a unification procedure to propagate positive conditionals. The unification process itself provides to check the equivalence of antecedents. Their tableau system is not analytic, since it contains a cut-rule, called PB, which is not eliminable. Moreover it is not clear how to extend it to nested conditionals.

Lamarre [Lamarre 1993] presents tableaux systems for the conditional logics V, VN, VC, and VW. Lamarre's method is a consistency-checking procedure which tries to build a system of sphere falsifying the input formulas. The method makes use of a subroutine to compute the *core*, that is defined as the set of formulas characterizing the minimal sphere. The computation of the core needs in turn the consistency checking procedure. Thus there is a mutual recursive definition between the procedure for checking consistency and the procedure to compute the core.

Groeneboer and Delgrande [Delgrande and Groeneboer 1990] have developed a tableau method for the conditional logic VN which is based on the translation of this logic into the modal logic S4.3.

[Giordano et al. 2003] have defined a labelled tableaux calculus for the logic CE and some of its extensions. The flat fragment of CE corresponds to the nonmono-

tonic preferential logic P and admits a semantics in terms of preferential structures (possible worlds together with a family of preference relations). The tableau calculus makes use of pseudo-formulas, that are modalities in a hybrid language indexed on worlds. In that paper it is shown how to obtain a decision procedure for that logic by performing a kind of loop checking.

Finally, complexity results for some conditional logics have been obtained by Friedman and Halpern [Friedman and Halpern 1994]. Their results are based on a semantic analysis, by an argument about the size of possible countermodels.

In the future, we intend to continue our work in two directions:

- (1) We want to investigate if it is possible to develop sequent calculi based on the selection function semantics for stronger conditional logics (than CK+MP+ID). If this is possible, we would like to extend CondLean to support these stronger systems.
- (2) We hope to increase the efficiency of our theorem prover CondLean by experimenting standard refinements and heuristics.

REFERENCES

- ARTOSI, A., GOVERNATORI, G., AND ROTOLO, A. 2002. Labelled tableaux for non-monotonic reasoning: Cumulative consequence relations. *Journal of Logic and Computation*, 12(6): 1027–1060.
- BECKERT, B. AND GORÈ, R. 1997. Free variable tableaux for propositional modal logics. *Tableaux-97, LNCS 1227, Springer*, 91–106.
- BECKERT, B. AND POSEGGA, J. 1995. leantap: Lean tableau-based deduction. *Journal of Automated Reasoning*, 15(3), 339–358.
- BELL, J. 1990. The logic of nonmonotonicity. *Artificial Intelligence*, 41(3), 365–374.
- BOUTILIER, C. 1994. Conditional logics of normality: a modal approach. *Artificial Intelligence*, 68(1), 87–154.
- CHELLAS, B. F. 1975. Basic conditional logics. *Journal of Philosophical Logic*, 4, 133–153.
- COSTELLO, T. AND MCCARTHY, J. 1999. Useful counterfactuals. *ETAI (Electronic Transactions on Artificial Intelligence)*, Vol. 3, Section A.
- CROCCO, G. AND DEL CERRO, L. F. 1995. Structure, consequence relation and logic, volume 4. In D. Gabbay (ed.), *What is a Logical System*, Oxford University Press, 239–259.
- CROCCO, G., DEL CERRO, L. F., AND HERZIG, A. 1995. Conditionals: From philosophy to computer science. *Oxford University Press, Studies in Logic and Computation*.
- CROCCO, G. AND LAMARRE, P. 1992. On the connection between non-monotonic inference systems and conditional logics. In B. Nebel and E. Sandewall editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference*, 565–571.
- DE SWART, H. C. M. 1983. A Gentzen-or Beth-type system, a practical decision procedure and a constructive completeness proof for the counterfactual logics VC and VCS. *Journal of Symbolic Logic*, 48(1), 1–20.
- DELGRANDE, J. P. 1987. A first-order conditional logic for prototypical properties. *Artificial Intelligence*, 33(1), 105–130.
- DELGRANDE, J. P. 1988. An approach to default reasoning based on a first-order conditional logic: Revised report. *Artificial Intelligence*, 36(1), 63–90.
- DELGRANDE, J. P. AND GROENEBOER, C. 1990. A general approach for determining the validity of commonsense assertions using conditional logics. *International Journal of Intelligent Systems*, 5(5), 505–520.
- FITTING, M. 1983. Proof methods for modal and intuitionistic logic. *Vol 169 of Synthese library, D. Reidel, Dordrecht*.
- FITTING, M. 1998. leantap revisited. *Journal of Logic and Computation*, 8(1), 33–47.

- FRIEDMAN, N. AND HALPERN, J. 1994. On the complexity of conditional logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference, KR 1994*, 202–213.
- FRIEDMAN, N. AND HALPERN, J. 1999. Belief revision: a critique. *Journal of Logic, Language and Information*, 8(4), 401–420.
- GABBAY, D. M. 1996. Labelled deductive systems (vol i). *Oxford Logic Guides, Oxford University Press*.
- GABBAY, D. M., GIORDANO, L., MARTELLI, A., OLIVETTI, N., AND SAPINO, M. L. 2000. Conditional reasoning in logic programming. *Journal of Logic Programming* 44 (1-3), 37–74.
- GARDENFORS, P. AND ROTT, H. 1995. Belief revision. *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4*, ed. by D. M. Gabbay, C. J. Hogger, and J. A. Robinson, Oxford University Press.
- GENT, I. P. 1992. A sequent or tableaux-style system for Lewis’s counterfactual logic VC. *Notre Dame Journal of Formal Logic*, 33(3), 369–382.
- GINSBERG, M. L. 1986. Counterfactuals. *Artificial Intelligence*, 30(1), 35–79.
- GIORDANO, L., GLIOZZI, V., AND OLIVETTI, N. 1998. A conditional logic for belief revision. In *Proc. European Workshop on Logics in Artificial Intelligence JELIA 98, Springer LNAI 1489*, 294–308.
- GIORDANO, L., GLIOZZI, V., AND OLIVETTI, N. 2002. Iterated belief revision and conditional logic. *Studia Logica* 70(1), 23–47.
- GIORDANO, L., GLIOZZI, V., OLIVETTI, N., AND SCHWIND, C. 2003. Tableau calculi for preference-based conditional logics. In *Proc. of TABLEAUX 2003 (Automated Reasoning with Analytic Tableaux and Related Methods)*, volume 2796 of LNAI, Springer, 81–101.
- GIORDANO, L. AND SCHWIND, C. 2004. Conditional logic of actions and causation. *Artificial Intelligence* 157, 1-2, 239–279.
- GRAHNE, G. 1998. Updates and counterfactuals. In *Journal of Logic and Computation*, 8(1):, 87–117.
- HUDELMAIER, J. 1993. An $\mathcal{O}(n \log n)$ -space decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1), 63–75.
- JAFFAR, J. AND MICHAYLOV, S. 1987. Methodology and implementation of a clp system. *J.L. Lassez (ed.), Logic Programming - Proceedings of the 4th International Conference - Volume 1*, MIT Press, Cambridge, MA, 196–218.
- KRAUS, S., LEHMANN, D., AND MAGIDOR, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2), 167–207.
- LAMARRE, P. 1992. Etude des raisonnements non-monotones: Apports des logiques des conditionnels et des logiques modales. *PhD thesis, Université Paul Sabatier, Toulouse*.
- LAMARRE, P. 1993. A tableaux prover for conditional logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference*, 572–580.
- LEWIS, D. 1973. Counterfactuals. *Basil Blackwell Ltd*.
- LINDSTROM, S. AND RABINOWICZ, W. 1992. Belief revision, epistemic conditionals and the Ramsey test. *Synthese* 91, 195–237.
- MARRIOTT, K. AND STUCKEY, P. J. 1998. Programming with constraints: An introduction. *MIT Press*.
- NUTE, D. 1980. Topics in conditional logic. *Reidel, Dordrecht*.
- OBEID, N. 2001. Model-based diagnosis and conditional logic. *Applied Intelligence*, 14, 213–230.
- OLIVETTI, N. AND POZZATO, G. L. 2003. Condean: A theorem prover for conditional logics. In *Proc. of TABLEAUX 2003 (Automated Reasoning with Analytic Tableaux and Related Methods)*, volume 2796 of LNAI, Springer, 264–270.
- OLIVETTI, N. AND SCHWIND, C. B. 2000. Analytic tableaux for conditional logics. *Technical Report, University of Torino*.
- OLIVETTI, N. AND SCHWIND, C. B. 2001. A calculus and complexity bound for minimal conditional logic. *Proc. ICTCS01 - Italian Conference on Theoretical Computer Science, vol LNCS 2202*, 384–404.

- POZZATO, G. L. 2003. Deduzione automatica per logiche condizionali: Analisi e sviluppo di un theorem prover. *Tesi di laurea, Informatica, Università di Torino. In Italian*, download at <http://www.di.unito.it/~pozzato/tesiPozzato.html>.
- SCHWIND, C. B. 1999. Causality in action theories. *Electronic Transactions on Artificial Intelligence (ETAI)*, 3(A), 27–50.
- STALNAKER, R. 1968. A theory of conditionals. In *N. Rescher (ed.), Studies in Logical Theory, American Philosophical Quarterly, Monograph Series no.2, Blackwell, Oxford*, 98–112.
- VIGANÒ, L. 2000. Labelled non-classical logics. *Kluwer Academic Publishers, Dordrecht*.