Towards a proof system for admissibility

Rosalie Iemhoff**

Technical University Vienna, Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria
iemhoff@logic.at,
http://www.logic.at/people/iemhoff

Abstract. In [4] a basis for the admissible rules of intuitionistic propositional logic IPC was given. Here we strengthen this result by presenting a system ADM that has the following two properties. $A \vdash_{\mathsf{ADM}} B$ implies that A admissibly derives B. ADM is complete in the sense that for every formula A there exists a formula $A \vdash_{\mathsf{ADM}} \Lambda_A$ such that the admissibly derivable consequences of A are the (normal) consequences of Λ_A. This work is related to and partly relies upon research by Ghilardi on projective formulas [2] [3]....

1 Introduction

An interesting meta-mathematical property of intuitionistic propositional logic IPC is the existence of non-derivable admissible rules for this logic. That is, there exist rules that, when added to IPC, do not lead to new theorems, but that are not derivable in IPC either. We write $A \mid\!\sim B$ if A/B is an admissible rule. Some results during the last decade have shed some light on the structure of the admissible derivability relation $\mid\!\sim$. Rybakov [6] showed that $\mid\!\sim$ is decidable and Ghilardi [3] presented a transparent algorithm. In [4] a simple syntactical characterization for $\mid\!\sim$ was given. This result implied that Visser's rules $V = \{Vn \mid n = 1, 2, \ldots\}$, where

$$Vn \quad (\bigwedge_{i=1}^{n}(A_i \to B_i) \to A_{n+1} \vee A_{n+2}) \mid\!\sim \bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(A_i \to B_i) \to A_j),$$

form a basis for the admissible rules of IPC. Intuitively, this means that all admissible rules of IPC can be obtained from Visser's rules via derivability in IPC. Here we strengthen this result in the following way. Our aim is to present a decent proof system for $\mid\!\sim$, i.e. a system that given $A \mid\!\sim B$, tells what are the syntactical manipulations one has to apply to A to obtain B. With decent we mean something like "as cut-free as possible". The mentioned characterization does not fulfill this aim since it contains a full cut rule: if $A \mid\!\sim B$ and $B \mid\!\sim C$, then $A \mid\!\sim C$. The form of V does not seem to suggest the existence of a neat cut-free Gentzen calculus for $\mid\!\sim$, as the rules Vn violate the subformula property.

** Research supported by a Marie Curie fellowship of the European Union under grant HPMC-CT-2001-01383.

Although cut elimination might not be possible in full extent, it might however still be possible to reach the above mentioned goal in a satisfying way. Here we present a result in this direction. We present a proof system ADM such that $A \vdash_{\mathsf{ADM}} B$ implies $A \mathrel{|\!\sim} B$. The converse does not hold. However, the system is complete in the sense that for every formula A there exists a unique formula $A \vdash_{\mathsf{ADM}} \Lambda_A$ such that

$$\forall B: \ A \mathrel{|\!\sim} B \text{ iff } \Lambda_A \vdash B.$$

Thus, given A, once one has obtained Λ_A via ADM, $A \mathrel{|\!\sim} B$ is reduced to $\Lambda_A \vdash B$. The system ADM is "cut-free" in the sense that it consists solely of "rewriting" rules and applications of V. With "rewriting" rules we mean in this context simple rules that e.g. infer $(A \to B) \wedge (A \to C)$ from $(A \to B \wedge C)$. In contrast to V, all these rules of ADM are derivable admissible rules. Note that this result implies that V is a basis for the admissible rules of IPC. However, it is stronger in the sense that it provides more information about $\mathrel{|\!\sim}$.

These results are intimately linked with and inspired by results from Ghilardi's papers [2] and [3]. In fact, our main theorem heavily relies on results in [3]. His results stem from research on unification in intermediate and modal logics. To this end he defines in [2] the notion of a projective formula. A formula A is called *projective* if there exists a substitution σ such that

$$\vdash \sigma A \text{ and } \forall B \ (A \vdash \sigma B \leftrightarrow B).$$

Projective formulas are very useful in the context of admissible rules because of the following property: for every projective formula A, for all formulas B: $A \mathrel{|\!\sim} B$ iff $A \vdash B$. Ghilardi shows in [2] that every formula A has a *projective approximation* $\Pi(A)$. $\Pi(A)$ is a finite set of projective formulas that derive A and such that for every projective C that derives A there exists $D \in \Pi(A)$ such that $C \vdash D$. It follows from [2] that, like for Λ_A, it holds that $A \mathrel{|\!\sim} B$ iff $\bigvee \Pi(A) \vdash B$. However, for our aim of giving a proof system that derives Λ_A from A, it seems that the notion of admissible projective sets instead of projective approximations is more useful. An *admissible projective set* for a formula A is a set of formulas X such that all formulas in X are either inconsistent or projective and

$$A \mathrel{|\!\sim} \bigvee X \vdash A.$$

A formula A is inconsistent if $A \vdash \bot$. The special behavior of projective formulas w.r.t. admissibility clearly holds for inconsistent formulas too:

For every projective or inconsistent formula A, for all formulas B: $A \mathrel{|\!\sim} B$ iff $A \vdash B$. If X is an admissible projective set for a formula A, then for all formulas B: $A \mathrel{|\!\sim} B$ iff $\bigvee X \vdash B$.

Now the main theorems of the paper, Theorem 1 and 2, state the following:
Theorem 1: $A \vdash_{\mathsf{ADM}} B$ then $A \mathrel{|\!\sim} B$.
Theorem 2: For every formula A there exists a unique formula $\Lambda_A = \bigvee dj(\Lambda_A)$, such that $dj(\Lambda_A)$ is an admissible projective set for A and $A \vdash_{\mathsf{ADM}} \Lambda_A$.

These results have various consequences for other intermediate logics. The most interesting one being that V is a basis for the admissible rules for any intermediate logic for which it is admissible. Due to lack of space we are not able to include these results here. They are discussed in [5]. The system ADM is not yet as elegant and neat, and whence as useful, as one would hope and expect. In particular, instead of replacing previously derived expressions, it just adds expressions to the initial formula. We hope to be able to present a better system in the near future.

2 Preliminaries

In this paper we will only be concerned with intuitionistic propositional logic IPC. We write $\vdash$ for derivability in IPC. The letters A, B, C, D, E, F, H range of formulas, the letters p, q, r, s, t, range over propositional variables. We assume $\top$ and $\bot$ to be present in the language. $\neg A$ is defined as $(A \to \bot)$. We omit parentheses when possible; $\wedge$ binds stronger than $\vee$, which in turn binds stronger than $\to$. A *substitution* σ will in this paper always be a map from propositional formulas to propositional formulas that commutes with the connectives. A *(propositional) admissible rule* of IPC is a rule A/B such that adding the rule to the logic does not change the theorems of IPC, i.e. $\forall\sigma : \vdash \sigma A$ implies $\vdash \sigma B$. We write $A \mathrel{|\!\sim} B$ if A/B is admissible. The rule is called *derivable* if $A \vdash B$ and *proper* if $A \nvdash B$. We say that a collection $\mathcal{R}$ of rules, e.g. V, is admissible if all rules in $\mathcal{R}$ are admissible. We write $A \vdash^{\mathcal{R}} B$ if B is derivable from A in the logic consisting of IPC extended with the rules $\mathcal{R}$, i.e. there are $A = A_1, \ldots, A_n = B$ such that for all $i < n$, $A_i \vdash A_{i+1}$ or there exists a σ such that $\sigma B_i/\sigma B_{i+1} = A_i/A_{i+1}$ and $B_i/B_{i+1} \in \mathcal{R}$. A set $\mathcal{R}$ of admissible rules is a *basis for the admissible rules* if for every admissible rule $A \mathrel{|\!\sim} B$ we have $A \vdash^{\mathcal{R}} B$.

3 The system ADM

The aim of the system ADM is to derive a formula Λ_A, which is the disjunction of an admissible projective set for A, by applying some syntactical rules to A. Intuitively it works as follows. First A is rewritten into a disjunction of conjunctions. Under all other rewriting rules that follow it will remain in this form, i.e. in each step the algorithm will only change one of the conjuncts of some disjunct of the formula. We write these conjuncts as sequents, which are interpreted in the usual way as implications. For the outer conjunctions and disjunctions we use different symbols, to distinguish them from the symbols in the sequent. The symbol "$\cdot$" is interpreted as conjunction, the symbol "$|$" as disjunction, where $|$ binds stronger than $\cdot$. Thus e.g. $S_1 \mid S_2 \cdot S_3$ stands for $S_1 \vee (S_2 \wedge S_3)$. The interpretation of $|$ as disjunction is in accordance with its interpretation in hypersequent calculi, see e.g. [1]. Our sequents will be of the form $\Gamma, \Pi \Rightarrow \Delta$, where Γ, Π and Δ are sets of formulas. We denote these sets without the $\{$'s and the comma's. Thus e.g. $A\,(B \wedge C), D \Rightarrow E\ F$ is a sequent in the sense defined above, and means $\{A, B \wedge C\}, \{D\} \Rightarrow \{E, F\}$. Also, if we write $\Gamma\ \{A_i \mid i = 1, 2\}, \Pi \Rightarrow \Delta$ we mean $\Gamma\ A_1\ A_2, \Pi \Rightarrow \Delta$. We let S range over sequents, Θ over expressions of the form $S_1 \cdot S_2 \cdot \ \ldots\ \cdot S_n$, and Λ over expressions of the form $\Theta_1 \mid \Theta_2 \mid \ldots \mid \Theta_n$. The Θ are called *conjunction expressions* and the Λ are called *disjunction expressions*. Both Θ and Λ are considered as sets which components are respectively sequents and conjunction expressions. Thus e.g. $S_1 \cdot S_2 \cdot S_1 = S_2 \cdot S_1$. If $\Theta = S_1 \cdot S_2 \cdot \ \ldots \ \cdot S_n$, the S_i are called the *conjuncts* of Θ, and Θ is interpreted as the formula $S_1' \wedge \ldots \wedge S_n'$, where S_i' is the formula corresponding to the sequent S_i. We write $S \in \Theta$ if Θ is of the form $\Theta' \cdot S$, i.e. if S is a conjunct of Θ. For $\Lambda = \Theta_1 \mid \Theta_2 \mid \ldots \mid \Theta_n$, the Θ_i are called the *disjuncts* of Λ, and Λ is interpreted as the formula $(\bigwedge \Theta_1') \vee \ldots \vee (\bigwedge \Theta_n')$, where Θ_i' is the formula corresponding to Θ_i. Sometimes we use sequents and conjunction and disjunction expressions also for their corresponding propositional formula. If we write a sequent but mean a formula, we often leave out the comma between Γ and Π since it plays no role in the context of formulas: $\Gamma\Pi \Rightarrow \Delta$ has the same interpretation as $\Gamma, \Pi \Rightarrow \Delta$. We write $\Gamma\Pi$ for $\Gamma \cup \Pi$. The empty conjunction is taken to be $\top$ and the empty disjunction is $\bot$. For a formula A we define its corresponding disjunction expression (the s for succedent):

$$\Lambda_{sA} \equiv_{def} (\ , \ \Rightarrow A).$$

Note that $\vdash A \leftrightarrow \Lambda_{sA}$. In fact, following our convention, Λ_{sA} is interpreted as the formula $(\top \rightarrow A)$ in $\vdash A \leftrightarrow \Lambda_{sA}$.

Definition 1. *We associate the following sets with a given set Γ:*

$$\begin{array}{ll}
\Gamma^a & = \{A \mid \exists B\ (A \rightarrow B \in \Gamma)\} \\
\Gamma^p & = \{A \in \Gamma \mid \exists B\ (A = p \rightarrow B)\} \\
C^{A \rightarrow B} & = \begin{cases} B \rightarrow D & \textit{if } C = (A \rightarrow B) \rightarrow D \\ C & \textit{otherwise} \end{cases} \\
C^{A\wedge} & = \begin{cases} A \rightarrow (A' \rightarrow B) & \textit{if } C = (A \wedge A' \rightarrow B) \\ C & \textit{otherwise} \end{cases} \\
C^{\wedge A} & = \begin{cases} A \rightarrow (A' \rightarrow B) & \textit{if } C = (A' \wedge A \rightarrow B) \\ C & \textit{otherwise} \end{cases} \\
C^{A \vee A'} & = \begin{cases} (A \rightarrow B) \wedge (A' \rightarrow B) & \textit{if } C = (A \vee A' \rightarrow B) \\ C & \textit{otherwise} \end{cases}
\end{array}$$

For a set Γ we define $\Gamma^{A \rightarrow B} = \{C^{A \rightarrow B} \mid C \in \Gamma\}$. Similarly for the other superscripts.

ADM

$$L\bot:\ \frac{\Lambda \mid \Theta \ \cdot \ \Gamma \bot, \Pi \Rightarrow \Delta}{\Lambda \mid \Theta} \qquad R\bot:\ \frac{\Lambda \mid \Theta \quad (\Gamma, \Pi \Rightarrow \bot \Delta \in \Theta)}{\Lambda \mid \Theta \ \cdot \ \Gamma, \Pi \Rightarrow \Delta}$$

$$L\top:\ \frac{\Lambda \mid \Theta \quad (\Gamma \top, \Pi \Rightarrow \Delta \in \Theta)}{\Lambda \mid \Theta \ \cdot \ \Gamma, \Pi \Rightarrow \Delta} \qquad R\top:\ \frac{\Lambda \mid \Theta \ \cdot \ \Gamma, \Pi \Rightarrow \top \Delta}{\Lambda \mid \Theta}$$

$$L\wedge:\ \frac{\Lambda \mid \Theta \quad (\Gamma A \wedge B, \Pi \Rightarrow \Delta \in \Theta)}{\Lambda \mid \Theta \ \cdot \ \Gamma A B, \Pi \Rightarrow \Delta} \qquad L\vee:\ \frac{\Lambda \mid \Theta \quad (\Gamma A \vee B, \Pi \Rightarrow \Delta \in \Theta)}{\Lambda \mid \Theta \ \cdot \ \Gamma A, \Pi \Rightarrow \Delta \ \cdot \ \Gamma B, \Pi \Rightarrow \Delta}$$

$$L\Gamma\to:\ \frac{\Lambda \mid \Theta \quad (\Gamma\, A\to B, \Pi \Rightarrow \Delta \in \Theta)}{\Lambda \mid \Theta\ \cdot\ \Gamma\, A\to B, \Pi \Rightarrow A\,\Delta\ \cdot\ \Gamma\, B, \Pi \Rightarrow \Delta}$$

$$L\Pi\to:\ \frac{\Lambda \mid \Theta \quad (\Gamma, A\to B\, \Pi \Rightarrow \Delta \in \Theta)}{\Lambda \mid \Theta\ \cdot\ \Gamma, A\to B\, \Pi \Rightarrow A\,\Delta\ \cdot\ \Gamma\, B, \Pi \Rightarrow \Delta}$$

$$R\wedge:\ \frac{\Lambda \mid \Theta \quad (\Gamma, \Pi \Rightarrow A\wedge B\,\Delta \in \Theta)}{\Lambda \mid \Theta\ \cdot\ \Gamma^{A\wedge}, \Pi \Rightarrow A\,\Delta\ \cdot\ \Gamma^{\wedge B}, \Pi \Rightarrow B\,\Delta}$$

$$R\vee:\ \frac{\Lambda \mid \Theta \quad (\Gamma, \Pi \Rightarrow A\vee B\,\Delta \in \Theta)}{\Lambda \mid \Theta\ \cdot\ \Gamma^{A\vee B}, \Pi \Rightarrow A\, B\,\Delta}$$

$$R\to:\ \frac{\Lambda \mid \Theta \quad (\Gamma, \Pi \Rightarrow A\to B\,\Delta \in \Theta)}{\Lambda \mid \Theta\ \cdot\ A\,\Gamma^{A\to B}, \Pi \Rightarrow B\,\Delta}$$

$$Res:\ \frac{\Lambda \mid \Theta \quad (\Gamma, \Pi \Rightarrow p\,\Delta \in \Theta,\ p\,\Gamma', \Pi' \Rightarrow \Delta' \in \Theta)}{\Lambda \mid \Theta\ \cdot\ (\Gamma\backslash\Gamma^{p})\Gamma', \Gamma^{p}\Pi\Pi' \Rightarrow \Delta\Delta'}$$

$$Vg:\ \frac{\Lambda \mid \Theta \quad (\Gamma, \Pi \Rightarrow \Delta \in \Theta,\ n>0,\ \Gamma \text{ only implications, } \Gamma^{a} \subseteq \Delta = A_0..A_n)}{\Lambda \mid \Theta\ \cdot\ \Gamma\Pi, \Rightarrow A_0 \mid \ldots \mid \Theta\ \cdot\ \Gamma\Pi, \Rightarrow A_n}$$

Res stands for resolution. The rule is called after an analogous rule in Ghilardi's algorithm CHECK-PROJECTIVITY. The expressions Λ are called *side expressions*. The conjunction expressions in the inferences that do not occur in Λ are the *principal conjunction expressions* of the inference. The sequents that are explicitly indicated in the rules above, are called *principal sequents* of the inference. Note that rules have in general more than one principal formula, sequent and conjunction expression. If Λ/Λ' is an instance of a rule R, then we write $\Lambda \vdash^{R}_{\mathsf{ADM}} \Lambda'$ or $\Lambda \vdash^{1}_{\mathsf{ADM}} \Lambda'$. If there are $\Lambda_1 \ldots \Lambda_n$ such that $\Lambda = \Lambda_1 \vdash^{1}_{\mathsf{ADM}} \ldots \vdash^{1}_{\mathsf{ADM}} \Lambda_n = \Lambda'$, we write $\Lambda \vdash_{\mathsf{ADM}} \Lambda'$. The depth of a disjunction expression Λ in a derivation $\mathcal{D} = \Lambda_1, \ldots, \Lambda_n$ is the minimal i such that $\Lambda = \Lambda_i$. The depth of a conjunction expression Θ in $\mathcal{D}$ is the minimal i such that Θ is a conjunct of Λ_i.

We will be only interested in derivations $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda$, where the first expression is of the special form Λ_{sA} for some formula A. The Λ that occur in these derivations have some special properties, as we will see in Proposition 1. We leave it to the reader to check that the following holds. If $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda$, then for all sequents $\Gamma, \Pi \Rightarrow \Delta$ that occur in Λ, the set Π consists only of implications.

Definition 2. *A set Γ is called* simple *if all its elements are implications or propositional variables. A sequent $\Gamma, \Pi \Rightarrow \Delta$ is called simple if $\Gamma\Pi\Delta$ is simple. A derivation in* ADM *is in* normal form *if the principal sequents in any Res or Vg inference are simple. If $\mathcal{D}$ is the smallest normal form derivation $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda$ such that for all $\Lambda \vdash^{1}_{\mathsf{ADM}} \Lambda'$ we have $\Lambda' = \Lambda$, then we denote Λ by Λ_A. Observe that Λ_A is unique. We denote the set of all disjuncts of a disjunction expression*

Λ by $dj(\Lambda)$. Thus $\vdash \Lambda \leftrightarrow \bigvee dj(\Lambda)$. We define

$$p(\Lambda) = \{\Theta \mid \Theta \in dj(\Lambda),\ \Theta \textit{ has no conjunct } (\Gamma, \Pi \Rightarrow \bot) \textit{ for which } \Gamma^a = \{\bot\}\}.$$

3.1 Properties of ADM

All rules of ADM except $R\wedge$, $R\vee$ and $R \rightarrow$ have a subformula property: the formulas that occur in the sequents of the lower expression are subformulas of the formulas in the sequents in the upper expression. The reason that e.g. $R\vee$ is not formulated in the naive way $\Gamma, \Pi \Rightarrow A \vee B \Delta / \Gamma, \Pi \Rightarrow A\, B\, \Delta$ (we have left out the side expressions), in which case it would have the subformula property, is best explained by an example. Consider a formula $A = (B \rightarrow p \vee q)$ where $B = (p \vee q \rightarrow r)$. Formulating $R\vee$ in the naive way would give the following derivation.

$$Vg\,\dfrac{R\vee\,\dfrac{R\rightarrow\,\dfrac{,\Rightarrow A}{B, \Rightarrow p \vee q}}{B, \Rightarrow p\, q}}{B, \Rightarrow p \vee q \mid B, \Rightarrow p \mid B, \Rightarrow q}$$

It is not difficult to see that this implies that Λ_A will contain a disjunct that is equivalent to $A \wedge (r \rightarrow p \vee q)$. The semantical characterization of projective formulas given by Ghilardi [2], and recalled in the appendix, shows that neither is this formula projective, nor is it inconsistent. Therefore, the naive formulation of $R\vee$ does not suffice for our purposes. In our case the above derivation will become:

$$L\Gamma\rightarrow\,\dfrac{Vg\,\dfrac{R\vee\,\dfrac{R\rightarrow\,\dfrac{,\Rightarrow A}{B, \Rightarrow p \vee q}}{(p \rightarrow r)(q \rightarrow r), \Rightarrow p\, q}}{B, \Rightarrow p \mid B, \Rightarrow q}}{B, \Rightarrow p\ \cdot\ r \Rightarrow p \mid B, \Rightarrow q \cdot\ r \Rightarrow q}$$

This shows that the non-projective disjunct will not appear in this way. Similar reasons explain the form of $R\wedge$ and $R \rightarrow$.

Although there is no subformula property, the situation is not that bad. All formulas in the sequents that appear in a derivation from Λ_{sA} belong to the smallest set $\mathcal{A}$ that contains the subformulas of A and that is closed under the operations $()^{B \rightarrow C}, ()^{B \vee C}, ()^{B\wedge}, ()^{\wedge B}$. Clearly, this set is finite. Note that the finiteness of $\mathcal{A}$ implies that Λ_A exists.

The rule Vg is different from the other rules in that it adds a sequent S to a conjunct Θ that is in general not derivable from Θ. However, as we will see in Theorem 1, the upper expression of the Vg rule admissibly derives the lower expression, at least in derivations that start with a sequent of the form Λ_{sA}. If in the application of the Vg rule to a sequent $\Gamma, \Pi \Rightarrow \Delta$, the set Π is empty, this follows already from the admissibility of the rule V mentioned in the introduction. If Π is non-empty, Vg is a generalization of the rule V. In the remainder of this section we show that the admissibility of V implies the admissibility of Vg. This is the main ingredient of Theorem 1 in which we show that $A \mathrel{|\!\sim} \Lambda_A \vdash A$. We need this generalization of V to make the proof of Theorem 2 in which we show that $dj(\Lambda_A)$ is an admissible projective set, work.

Remark 1. We will need the following consequences of the admissibility of V, the verification of which we leave to the reader. If Γ and Σ are two sets of implications, it holds, if at least one of the conditions stated below is satisfied, that

$$\Gamma\Sigma \Rightarrow \Delta \mathrel{|\!\sim} \bigvee_{A \in \Delta \cup \Sigma^a} (\Gamma\Sigma \Rightarrow A).$$

The conditions are as follows. (1) If $\Gamma^a \subseteq \Delta$. (2) If for all $A \in \Gamma^a$ there exists a $B \in \Delta$ such that $A \vdash B$. (3) If $\Gamma^a = \bigvee \Delta$. For in this case $\bigwedge \Gamma$ is equivalent (over IPC, and whence over L) to $\bigwedge\{A \to B \mid \bigvee \Delta \to B \in \Gamma, A \in \Delta\}$.

Definition 3. *For a set of formulas I we denote $\{A \to B \in \Pi \mid A \in I\}$ by Π^I. For a sequent $(\Gamma, \Pi \Rightarrow C)$ we define*

$$(\Gamma, \Pi \Rightarrow \Delta)^I = \Gamma\{A \in \Pi^a \mid A \in I\}, \Pi\backslash\Pi^I \Rightarrow \Delta.$$

*For a sequent S we say that another sequent $(\Gamma, \Pi \Rightarrow \Delta)$ is S-*correct *if $\forall I \subseteq \Pi^a : \ S \vdash (\Gamma, \Pi \Rightarrow \Delta)^I$. For a set of implications Π we define*

$$\Pi_\Delta = \{\bigvee \Delta \to D \mid \exists C(C \to D \in \Pi)\},$$

i.e. Π_Δ is the result of replacing all antecedents of the implications in Π by $\bigvee \Delta$. $\Pi^I_\Delta = (\Pi^I)_\Delta$.

Remark 2. If $\Gamma, \Pi \Rightarrow \Delta$ is S-correct then $S \vdash \Gamma, \Pi \Rightarrow \Delta$. If Π is empty, then $\Gamma, \Pi \Rightarrow \Delta$ is S-correct iff S derives $\Gamma, \Pi \Rightarrow \Delta$.

The reader is advised to first read the proof of Proposition 1 before embarking on the following two lemma's, that may otherwise seem somewhat mysterious.

Lemma 1. *If the sequent $\Gamma, \Pi \Rightarrow \Delta$ is S-correct then for all $I \subseteq \Pi^a$ the sequent $\Gamma\Pi^I_\Delta, \Pi\backslash\Pi^I \Rightarrow \Delta$ is S-correct too.*

Proof. We use induction on $|I|$. The case $|I| = 0$ follows from the S-correctness of $\Gamma, \Pi \Rightarrow \Delta$. Suppose $|I| > 0$. Let $\Sigma = \Pi\backslash\Pi^I$. We have to show that for all $J \subseteq \Sigma^a$

$$S \vdash \Gamma\Pi^I_\Delta\{A \in \Sigma^a \mid A \in J\}, \Sigma\backslash\Sigma^J \Rightarrow \Delta. \tag{1}$$

Pick a $C \in I$ and let $I' = I\backslash\{C\}$ and $\Sigma' = \Pi\backslash\Pi^{I'}$. Note that $C \notin \Sigma^a$, whence $C \notin J$ and $\Sigma^a \cup \{C\} = (\Sigma')^a$. The induction hypothesis says that $\Gamma\Pi^{I'}_\Delta, \Pi\backslash\Pi^{I'} \Rightarrow \Delta$ is S-correct. This implies that we have for all $H \subseteq (\Sigma')^a$

$$S \vdash \Gamma\Pi^{I'}_\Delta\{A \in (\Sigma')^a \mid A \in H\}, \Sigma'\backslash(\Sigma')^H \Rightarrow \Delta. \tag{2}$$

Thus we have $S \vdash \Gamma\Pi^{I'}_\Delta\{A \in (\Sigma')^a \mid A \in J\}, \Sigma'\backslash(\Sigma')^J \Rightarrow \Delta$ by taking $H = J$. We show that

$$S \vdash \Gamma\Pi^I_\Delta\{A \in \Sigma^a \mid A \in J\}, \Sigma\backslash\Sigma^J \Rightarrow \bigwedge \Gamma\Pi^{I'}_\Delta\{A \in (\Sigma')^a \mid A \in J\}, \Sigma'\backslash(\Sigma')^J.$$

Then (1) will follow from (2). Clearly, $S \vdash \Gamma\Pi^I_\Delta \Rightarrow \bigwedge \Pi^{I'}_\Delta$. Since $\Sigma^a \cap J = (\Sigma')^a \cap J$ also $S \vdash \{A \in \Sigma^a \mid A \in J\} \Rightarrow \bigwedge\{A \in (\Sigma')^a \mid A \in J\}$. It remains to show that $S \vdash \Gamma\Pi^I_\Delta\{A \in \Sigma^a \mid A \in J\}, \Sigma\backslash\Sigma^J \Rightarrow \bigwedge \Sigma'\backslash(\Sigma')^J$. Observe that $\Sigma'\backslash(\Sigma')^J = \Sigma\backslash\Sigma^J \cup \{C \to D \mid C \to D \in \Pi\}$ since $C \notin J$. Therefore, it suffices to show that for all $(C \to D) \in \Pi$ we have

$$\Gamma\Pi^I_\Delta\{A \in \Sigma^a \mid A \in J\}, \Sigma\backslash\Sigma^J \Rightarrow C \to D.$$

The following derivation shows that this holds.

$$\begin{array}{l}
S \vdash \Gamma\Pi^{I'}_\Delta\, C\,\{A \in (\Sigma')^a \mid A \in J\}, \Sigma'\backslash(\Sigma')^{J\cup\{C\}} \Rightarrow \Delta \\
\quad \Gamma\Pi^{I'}_\Delta\{A \in (\Sigma')^a \mid A \in J\}, \Sigma'\backslash(\Sigma')^{J\cup\{C\}} \Rightarrow C \to \bigvee \Delta \\
\quad \Gamma\Pi^{I'}_\Delta(\bigvee \Delta \to D)\{A \in (\Sigma')^a \mid A \in J\}, \Sigma'\backslash(\Sigma')^{J\cup\{C\}} \Rightarrow C \to D \\
\quad \Gamma\Pi^{I}_\Delta\{A \in \Sigma^a \mid A \in J\}, \Sigma'\backslash(\Sigma')^{J\cup\{C\}} \Rightarrow C \to D \\
\quad \Gamma\Pi^{I}_\Delta\{A \in \Sigma^a \mid A \in J\}, \Sigma\backslash\Sigma^{J} \Rightarrow C \to D
\end{array}$$

The first step is (2) with $H = J \cup \{C\}$. The step from the 3th to 4th line holds, as $\Sigma^a \cap J = (\Sigma')^a \cap J$. The step from the 4th to 5th line holds, as $\Sigma\backslash\Sigma^J = \Sigma'\backslash(\Sigma')^{J\cup\{C\}}$. This proves the lemma.

Lemma 2. *If Π consist of implications only and the sequent $\Gamma, \Pi \Rightarrow \Delta$ is S-correct, then for all $I \subseteq \Pi^a$:*

$$S \wedge \bigwedge\{\Gamma\Pi \Rightarrow A \mid A \in I\} \mathrel{|\!\sim} \bigvee\{\Gamma\Pi \Rightarrow A \mid A \in \Delta \cup \Gamma^a \cup (\Pi^a\backslash I)\}.$$

Proof. Consider $I \subseteq \Pi^a$. Let S' denote $S \wedge \bigwedge\{\Gamma\Pi \Rightarrow A \mid A \in I\}$. Let Σ denote $\Gamma\Pi^I_\Delta, \Pi\backslash\Pi^I$ and consider the sequent $T = (\Sigma \Rightarrow \Delta)$. By Lemma 1, T is S-correct, whence it is S'-correct. In particular, $S' \vdash T$. By Remark 1 $S' \mathrel{|\!\sim} \bigvee\{\Sigma \Rightarrow A \mid A \in \Delta \cup \Gamma^a \cup \Pi^a\backslash I\}$. Whence we are done once we can show that $S' \vdash \Gamma\Pi \Rightarrow A$ for all $A \in \Sigma$. Clearly, $S' \vdash \Gamma\Pi \Rightarrow A$ for all $A \in I$. Therefore, for all $(A \to B) \in \Pi$ for which $A \in I$, $S' \vdash \Gamma\Pi \Rightarrow B$, and whence $S' \vdash \Gamma\Pi \Rightarrow \bigvee \Delta \to B$. Therefore, $S' \vdash \Gamma\Pi \Rightarrow A$ for all $A \in \Sigma$. This proves the lemma.

Proposition 1. *If Γ and Π are sets of implications such that the sequent $\Gamma, \Pi \Rightarrow \Delta$ is S-correct and $\Gamma^a \subseteq \Delta$, then $S \mathrel{|\!\sim} \bigvee\{\Gamma\Pi \Rightarrow D \mid D \in \Delta\}$.*

Proof. The lemma follows from the following derivation.

$$\begin{array}{ll}
S \mathrel{|\!\sim} \bigvee_{D\in\Delta}(\Gamma\Pi \Rightarrow D) \vee \bigvee_{A\in\Pi^a}(\Gamma\Pi \Rightarrow A) & (\textit{Remark } 1 \textit{ and } 2) \\
\quad \bigvee_{D\in\Delta}(\Gamma\Pi \Rightarrow D) \vee \bigvee_{A\in\Pi^a}(\Gamma\Pi \Rightarrow A) \wedge & \\
\quad\quad \wedge(\bigvee_{D\in\Delta}\Gamma\Pi \Rightarrow D \vee \bigvee_{B\in\Pi^a, B\neq A}(\Gamma\Pi \Rightarrow B)) & (\textit{Lemma } 2) \\
\quad \bigvee_{D\in\Delta}(\Gamma\Pi \Rightarrow D) \vee \bigvee_{A,B\in\Pi^a, A\neq B}(\Gamma\Pi \Rightarrow A) \wedge (\Gamma\Pi \Rightarrow B) & \\
\quad \vdots & \\
\quad \bigvee_{D\in\Delta}(\Gamma\Pi \Rightarrow D) \vee \bigwedge_{A\in\Pi^a}(\Gamma\Pi \Rightarrow A) & \\
\quad \bigvee_{D\in\Delta}(\Gamma\Pi \Rightarrow D) & (\textit{Lemma } 2)
\end{array}$$

Lemma 3. *If Π consist of implications only, $S' = (\Gamma, \Pi \Rightarrow \bot)$ is S-correct, and $\Gamma^a = \{\bot\}$, then $S \vdash \bot$.*

Proof. Let $n = |\Pi^a|$. With induction to $n - |I|$ we show that for all $I \subseteq \Pi^a$

$$S \vdash \bigwedge\{A \in \Pi^a \mid A \in I\} \to \bot. \tag{3}$$

Let us first see that this implies the lemma. For all $B \in \Pi^a$, (3) gives $S \vdash \neg B$, by taking $I = \{B\}$. This implies $S \vdash \bigwedge \Pi$. Also, $S \vdash \bigwedge \Gamma$ because $\Gamma^a = \{\bot\}$. The S-correctness of S' gives $S \vdash (\bigwedge \Gamma \Pi \Rightarrow \bot)$, by taking $I = \emptyset$. Whence $S \vdash \bot$.

It remains to prove (3). The case $|I| = n$ follows from the S-correctness of $(\Gamma, \Pi \Rightarrow \bot)$ and the fact that $S \vdash \bigwedge \Gamma$. The case that $|I| < n$. By the induction hypothesis we have for all $B \in \Pi^a \backslash I$ that $S \vdash B \wedge \bigwedge\{A \in \Pi^a \mid A \in I\} \to \bot$. Whence

$$\forall B \in \Pi^a \backslash I : S \vdash \bigwedge\{A \in \Pi^a \mid A \in I\} \to \neg B. \tag{4}$$

This implies that $S \vdash \bigwedge\{A \in \Pi^a \mid A \in I\} \to \bigwedge(\Pi \backslash \Pi^I)$. By S-correctness we have

$$S \vdash \bigwedge \Gamma \wedge \bigwedge\{A \in \Pi^a \mid A \in I\}, \bigwedge(\Pi \backslash \Pi^I) \to \bot.$$

Therefore, $S \vdash \bigwedge\{A \in \Pi^a \mid A \in I\} \to \bot$.

Lemma 4. *If $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda$ and $\Lambda = \Theta_1 \mid \ldots \mid \Theta_n$, then for all $i \leq n$, all $S \in \Theta_i$ are Θ_i-correct.*

Proof. We use induction to the depth d of Λ in the derivation $\mathcal{D}$. The case $d = 0$ follows from Remark 2. Assume $d > 0$. We distinguish by cases according to the last rule of $\mathcal{D}$. Observe that if $\Gamma, \Pi \Rightarrow \Delta$ is S-correct and if $S \vdash \Gamma' \Rightarrow \Gamma$ and $S \vdash \Gamma \Gamma' \Delta \Rightarrow \Delta'$, then $\Gamma', \Pi \Rightarrow \Delta'$ is S-correct. This observation suffices for all cases except $L\Pi \to$, Vg and Res. We leave $L\Pi \to$ to the reader. For the case Vg, apply Remark 2. We treat the case Res. In this case the one but last disjunction expression in $\mathcal{D}$ is $\Lambda' = \Lambda'' \mid \Theta'' \;\cdot\; \Gamma, \Pi \Rightarrow p\,\Delta \;\cdot\; p\Gamma', \Pi' \Rightarrow \Delta'$ and

$$\Lambda = \Lambda'' \mid \Theta'' \;\cdot\; \Gamma, \Pi \Rightarrow p\,\Delta \;\cdot\; p\Gamma', \Pi' \Rightarrow \Delta' \;\cdot\; (\Gamma \backslash \Gamma^p)\Gamma', \Gamma^p \Pi \Pi' \Rightarrow \Delta\,\Delta'.$$

Let $\Theta' = \Theta'' \;\cdot\; \Gamma, \Pi \Rightarrow p\,\Delta \;\cdot\; p\Gamma', \Pi' \Rightarrow \Delta'$ and

$$\Theta = \Theta' \;\cdot\; (\Gamma \backslash \Gamma^p)\Gamma', \Gamma^p \Pi \Pi' \Rightarrow \Delta\,\Delta'.$$

For the sequents in Λ'', Θ'' and the sequents $S_1 = (\Gamma, \Pi \Rightarrow p\,\Delta)$ and $S_2 = (p\Gamma', \Pi' \Rightarrow \Delta')$ the induction hypothesis applies. Whence they are Θ'-correct, and thus also Θ-correct. We consider the remaining sequent of Λ. Let $\Sigma = \Gamma^p \Pi \Pi'$. We have to show that for all $I \subseteq \Sigma^a$,

$$\Theta \vdash (\Gamma \backslash \Gamma^p)\Gamma'\{A \in \Sigma^a \mid A \in I\}, \Sigma \backslash \Sigma^I \Rightarrow \Delta\Delta'. \tag{5}$$

Let $I_1 = I \cap \Pi^a$ and $I_2 = I \cap (\Pi')^a$. Observe that $\Pi \backslash \Pi^{I_1}$ and $\Pi' \backslash (\Pi')^{I_2}$ are subsets of $\Sigma \backslash \Sigma^I$. The fact that S_2 is Θ-correct implies that we have $\Theta \vdash p\,\Gamma'\{A \in (\Pi')^a \mid A \in I_2\}, \Pi' \backslash (\Pi')^{I_2} \Rightarrow \Delta'$. Whence

$$\Theta \vdash p\,(\Gamma \backslash \Gamma^p)\Gamma'\{A \in \Sigma^a \mid A \in I\}, \Sigma \backslash \Sigma^I \Rightarrow \Delta\Delta'. \tag{6}$$

If $p \in I$, then $p \in \{A \in \Sigma^a \mid A \in I\}$. Hence (5) follows from (6). If $p \notin I$ we use the fact that S_1 is Θ-correct, i.e. $\Theta \vdash \Gamma\{A \in \Pi^a \mid A \in I_1\}, \Pi \backslash \Pi^{I_1} \Rightarrow p\,\Delta$. Since $p \notin I$, $\Gamma^p \subseteq \Sigma \backslash \Sigma^I$. Therefore it follows that

$$\Theta \vdash (\Gamma \backslash \Gamma^p)\Gamma'\{A \in \Sigma^a \mid A \in I\}, \Sigma \backslash \Sigma^I \Rightarrow p\,\Delta. \tag{7}$$

Combining (7) with (6) gives (5).

Note that the previous lemma does not hold without the assumption that $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda$. The special form of the first expression plays a role in the induction.

Theorem 1. *If $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda \vdash_{\mathsf{ADM}} \Lambda'$ then $\Lambda \mathrel{|\!\sim} \Lambda' \vdash \Lambda$. In particular, $A \mathrel{|\!\sim} \Lambda_A \vdash A$. In fact, for all rules R except Vg, even $\Lambda \vdash^R_{\mathsf{ADM}} \Lambda'$ implies $\Lambda \vdash \Lambda' \vdash \Lambda$.*

Proof. We use induction to the depth d of the derivation. The proof for $\Lambda' \vdash \Lambda$ is trivial. For $\Lambda \mathrel{|\!\sim} \Lambda'$, in the case $d = 0$ we have $\Lambda = \Lambda'$. For $d > 0$ we distinguish by cases according to the last rule that is applied. We only treat the case Vg. Suppose Λ derived from the application of the rule Vg to Λ. Suppose $\Lambda = \Lambda'' \mid \Theta \cdot S$, where $S = (\Gamma, \Pi \Rightarrow \Delta)$ with $\Delta = A_0 \dots A_n$ and Γ a set of implications such that $\Gamma^a \subseteq \Delta$. As observed before, the fact that $\Lambda_{sA} \vdash_{\mathsf{ADM}} \Lambda$ implies that Π consists of implications only. For $1 \leq i \leq n$, let $S_i = (\Gamma\Pi, \Rightarrow A_i)$. Whence

$$\Lambda' = \Lambda'' \mid \Theta \cdot S_0 \mid \dots \mid \Theta \cdot S_n.$$

Therefore, to prove the lemma it suffices to show that $\Theta \wedge S \mathrel{|\!\sim} S_0 \vee \ldots \vee S_n$. This follows from Lemma 4 in combination with Proposition 1.

Note that in the previous theorem, the fact that Λ and Λ' appear in derivations from Λ_{sA} is crucial: in general, $\Lambda \vdash^{Vg}_{\mathsf{ADM}} \Lambda'$ does not imply $\Lambda \mathrel{|\!\sim} \Lambda'$. The previous theorem together with the second main theorem, Theorem 2 proved in the next section, show that ADM has the properties we want it to have, as discussed in the introduction.

3.2 Properties of Λ_A

In this section we discuss only IPC. Whence $\vdash$ and $\mathrel{|\!\sim}$ stand again for $\vdash_{\mathsf{IPC}}$ and $\mathrel{|\!\sim}_{\mathsf{IPC}}$. Recall from Theorem 1 that $A \mathrel{|\!\sim} \Lambda_A \vdash A$. The last fact implies that for all disjuncts Θ of Λ_A we have $\Theta \vdash A$.

Lemma 5. *For every formula A, for every $\Theta \in dj(\Lambda_A) \backslash p(\Lambda_A)$, it holds that $\vdash \neg\Theta$.*

Proof. Let Θ be a disjunct of Λ_A that is not in $p(\Lambda_A)$. Since $\Theta \notin p(\Lambda_A)$ it follows that Θ contains a conjunct $(\Gamma, \Pi \Rightarrow \bot)$ for which $\Gamma^a = \{\bot\}$. By Lemma 4 $(\Gamma, \Pi \Rightarrow \bot)$ is Θ-correct. Thus by Lemma 3 we have $\Theta \vdash \bot$.

Lemma 6. *For any formula A, every formula in $p(\Lambda_A)$ is projective.*

Proof. Let Θ be one of the disjuncts of Λ_A. We show that if $\Theta \in p(\Lambda_A)$, then Θ is projective. We will use Ghilardi's algorithm, described in the appendix, to show this. As Ghilardi remarks in [3], the order in which the steps of the algorithm are applied is irrelevant. We will apply them in the following order.

$S1$ Apply $T\top$, $T\bot$, Tp, $T\wedge$, $T\vee$, $T\!\!\rightarrow$, $T\neg$ and the two Simplification Rules.
$S2$ Apply $F\top$, $F\bot$, Fp, $F\wedge$, $F\vee$, $F\!\!\rightarrow$ and $F\neg$.
$S3$ Apply the Resolution Rule.

We apply these steps $S1-S3$ as long as possible, and we will not apply $S3$ ($S2$) as long as we can apply $S2$ ($S1$). When computing the algorithm on a formula A, we will initialize with $\{FA\}$. The input formulas of the algorithm are conjunctions of implications. Thus in this case it would e.g. be $(\top \rightarrow A)$. According to the Initialization we should then start with $\{T\top, FA\}$. After one application of $T\top$ we have $\{FA\}$ instead. Thus we see that starting with $\{FA\}$ gives the same result as starting with the Initialization according to the algorithm. We do this only to obtain a smooth induction. Namely, we will show in Claim 1 that when applying the algorithm in this order to Θ, for every set O that is created there exists a conjunct $S_O = (\Gamma, \Pi \Rightarrow \Delta)$ of Θ with certain properties. Then we show, Claim 2 and Claim 3, that this shows that either all output sets of Θ contain atomic modalities or $\Theta \notin p(\Lambda_A)$. By Theorem 4.2 of [3] (cited here as Theorem 4) all formulas for which all output sets contain atomic modalities are projective. This then proves the theorem. We denote $\{A \in \Gamma \mid TA \notin O\}$ by $\Gamma \backslash TO$. For $X, Y, Z \in \{T, T^+, T_c\}$ or $X, Y, Z \in \{F, F^+, F_c\}$, XYO denotes $XO \cup YO$ and $XYZO$ denotes $XO \cup YO \cup ZO$.

Claim. For every O that is created and not removed there exists a conjunct $S_O = (\Gamma, \Pi \Rightarrow \Delta)$ of Θ that satisfies the following properties:

(a) $TT^+O \subseteq \Gamma\Pi$
(b) $\Delta = FF^-F_cO$
(c) $\forall B \in (\Gamma\Pi \backslash TO)$ (B is an implication and $\Theta \vdash \bigwedge T_cO \rightarrow B$)
(d) $\forall (B \rightarrow C) \in (\Gamma \backslash TO)(B \in \Delta \cup \{\bot\})$.

Proof. With induction to the number of sets that were created before O.

The Initialization is clear. We consider the induction step. We distinguish by cases according to the step Si in which O is created. Assume O is the result of an application of a rule to O' or, in the case of *Res* to (O', O''). The induction hypothesis for O' implies that there is a conjunct $S_{O'}$ of Θ that satisfies $(a)-(d)$ w.r.t. O'. Clearly, the Simplification Rules do not have to be considered. Observe that we can also omit the rules $T\neg$ and $F\neg$. As they can be seen as consecutive applications of $T \rightarrow$ and $T\bot$ or $F \rightarrow$ and $F\bot$ respectively.

$S1$. O is the result of an application of one of the rules $T\top$, $T\bot$, Tp, $T\wedge$, $T\vee$ and $T\!\!\rightarrow$ to some formula in O'. In the cases $T\top$ and Tp we take $S_O = S_{O'}$. In the case $T\bot$, O' is removed, so O does not exist. We treat the cases $T\wedge$ and $T\!\!\rightarrow$ and leave $T\vee$ to the reader. In the first case $T\wedge$ is applied to some formula $T(B \wedge C) \in O'$. By (a), $S_{O'} = (\Gamma\, B \wedge C, \Pi \Rightarrow \Delta)$ for some Γ, Π, Δ. We let $S_O = (\Gamma\, B C, \Pi \Rightarrow \Delta)$. This sequent is a conjunct of Θ by the rule $L\wedge$. We leave it to the reader to check that S_O satisfies $(a)-(d)$. For the case $T\!\!\to$, O is the result of an application of $T\!\!\to$ to some formula $T(B \to C)$ in O'. By (a), $S_{O'} = (\Gamma\, B \to C, \Pi \Rightarrow \Delta)$ or $S_{O'} = (\Gamma, B \to C\, \Pi \Rightarrow \Delta)$. Assume we are in the first situation. The second one is analogous. We have

$$O = O' \backslash \{T(B \to C)\} \cup \{T_c(B \to C), FB\} \text{ or } O = O' \backslash \{T(B \to C)\} \cup \{TC\}.$$

In the first case, take $S_O = (\Gamma\, B \to C, \Pi \Rightarrow B\, \Delta)$. In the second case take $S_O = (\Gamma\, C, \Pi \Rightarrow \Delta)$. The rule $L\Gamma \to$ in ADM guarantees that S_O is a conjunct of Θ. We leave it to the reader to check that in both cases S_O satisfies $(a)-(d)$.

$S2$. O is the result of an application of $F\top$, $F\bot$, Fp, $F\wedge$, $F\vee$ or $F \to$ to some formula in O'. We treat the cases $F\wedge$, $F \to$. The case $F\vee$ is similar to the case $F\wedge$. First, the case $F\wedge$. Then $F\wedge$ is applied to some $F(B \wedge C) \in O'$. By (b) $S_{O'} = (\Gamma, \Pi \Rightarrow B \wedge C\, \Delta)$. Thus

$$O = O' \backslash \{FB \wedge C\} \cup \{FB\} \text{ or } O = O' \backslash \{FB \wedge C\} \cup \{FC\}.$$

For reasons of symmetry we only have to consider one case, suppose the first one. The rule $R\wedge$ in ADM implies that Θ has a conjunct $\Gamma^{B\wedge}, \Pi \Rightarrow B\, \Delta$. Let S_O be this conjunct. To see that (a) holds, note that since $S1$ cannot be applied, there are no formulas TC in O' and whence in O. It is easy to see that (b) holds. For (c), observe that formulas in Γ, if replaced in going from $S_{O'}$ to S_O, are only replaced by equivalent formulas. Whence the induction hypothesis applies. For (d), note that formulas $(B \wedge C \to D) \in \Gamma$ are replaced by $(B \to (C \to D)) \in \Gamma^{B\wedge}$.

Second, the case $F \to$. There is some $F(B \to C) \in O$ such that

$$O = O' \backslash \{FB \to C\} \cup \{TB, FC\} \text{ or } O = O' \backslash \{FB \to C\} \cup \{F_c(B \to C)\}.$$

By (b), $S_{O'} = (\Gamma, \Pi \Rightarrow B \to C\, \Delta)$. In the first case take $S_O = (B\, \Gamma^{B\to C}, \Pi \Rightarrow C\, \Delta)$. This is a conjunct of Θ by $R \to$. To see that (a) holds: since $S1$ cannot be applied to O', there are no formulas of the form TD in O except TB. (b) is easy. The only problematic case for (c) might be formulas $(B \to C) \to D \in \Gamma$. However, note that the fact that $\Theta \vdash T_cO' \to ((B \to C) \to D)$ implies $\Theta \vdash T_cO \to (C \to D)$. For (d), observe that formulas $(B \to C) \to D \in \Gamma$ are replaced by $(C \to D)$ in $\Gamma^{B\to C}$. In the second case take $S_O = S_{O'}$. Here we leave the verification of $(a)-(d)$ to the reader.

$S3$. O is the result of applying the Resolution Rule to (O', O''). By (a) and (b) we may assume that $S_{O'}$ and $S_{O''}$ have the following form.

$$S_{O'} = (\Gamma, \Pi \Rightarrow p\, \Delta) \text{ and } S_{O''} = (p\, \Gamma', \Pi' \Rightarrow \Delta').$$

Note that since $S1$ and $S2$ cannot be applied, both sequents are simple. We take $S_O = (\Gamma \backslash \Gamma^p)\Gamma', \Gamma^p \Pi \Pi' \Rightarrow \Delta\Delta'$. Note that this sequent is a conjunct of Θ by the rule Res and the fact that $S_{O'}$ and $S_{O''}$ are simple (recall that we assume the proof to be in normal form). We leave the verification of (a), (b) and (c) to the reader. For (d), observe that the only possibly problematic case would be formulas $(p \to A) \in \Gamma$. However, for these formulas the property (d) does not have to hold anymore, because of their place in S_O. This finishes the proof of the claim.

Claim. If an algorithm set O contains only context modalities, and $F_c O$ is non-empty, then O is removed by the Simplification Rule.

Proof. Let $S_O = (\Gamma, \Pi \Rightarrow \Delta)$. By (c) this implies that $\Gamma\Pi$ consists of implications only. By (b) and the fact that O contains only context modalities, Δ consists of implications and propositional variables only. In particular, S_O is simple. Note that (b) implies that Δ is non-empty. By (d), $\Gamma^a \subseteq \Delta$. Whence the rule Vg implies that there exists an $A \in \Delta$ such that $(\Gamma\Pi, \Rightarrow A)$ is a conjunct of Θ. By (c), $\Theta \vdash T_c O \to \bigwedge \Gamma\Pi$. Whence $\Theta \vdash T_c O \to A$. Since $A \in F_c O$, this shows that O will be removed. This proves the claim.

Claim. If an algorithm set O contains only context modalities, and $F_c O$ is empty, then Θ contains a conjunct $(\Gamma, \Pi \Rightarrow\ )$ with $\Gamma^a = \{\bot\}$.

Proof. Let $S_O = (\Gamma, \Pi \Rightarrow \Delta)$. By (d), $\Gamma^a = \{\bot\}$. By (b), Δ is empty. This proves the claim.

Observe that if Θ contains a conjunct $(\Gamma, \Pi \Rightarrow\ )$, then it also contains a conjunct $(\Gamma, \Pi \Rightarrow \bot)$. Since $\Theta \in p(\Lambda_A)$, it does not contain such conjuncts. Therefore, Claim 3 implies that if an algorithm set of Θ contains only context modalities, then $F_c O$ is non-empty. Claim 2 implies that this set will be removed. This proves that all output sets of Θ contain atomic modalities.

Theorem 2. *For every formula A, the set $dj(\Lambda_A)$ is an admissible projective set for A.*

Proof. By the previous lemma and theorem, all formulas in $dj(\Lambda_A)$ are either projective or inconsistent. By Theorem 1 we have $A \mid\!\sim \bigvee dj(\Lambda_A) \vdash A$.

References

1. Avron, A.: The method of hypersequents in the proof theory of propositional non-classical logics. Logic: from Foundations to Applications, European Logic Colloquium (1996) 1–32
2. Ghilardi, S.: Unification in intuitionistic logic. J. of Symb. Logic **64** (1999) 859–880
3. Ghilardi, S.: A resolution/tableaux algorithm for projective approximations in IPC. Logic Journal of the IGPL **10** (2002) 229–243
4. Iemhoff, R.: On the admissible rules of Intuitionistic Propositional Logic. J. of Symb. Logic **66** (2001) 281–294
5. Iemhoff, R.: Intermediate logics and Visser's rules. Manuscript (2003)
6. Rybakov, V.: Admissibility of Logical Inference Rules. (1997)

4 Appendix

In [2] introduced the notion of a projective formula. The paper contains the following useful semantical characterization of projective formulas. For Kripke models $K_1, \ldots, K_n$ we denote by $(\sum_i K_i)'$ the Kripke model which is the result of attaching one new node below all nodes in $K_1, \ldots, K_n$, at which no propositional variable is valid. We say that two rooted Kripke models are *variants* of each other when they have the same domain and partial order, and their forcing relations only possibly differ at the roots. A class of rooted Kripke models has the *extension property* when for every finite set of Kripke models $K_1, \ldots, K_n$ in this class, there is a variant of $(\sum_i K_i)'$ which is in this class as well.

Theorem 3. *(Ghilardi [2]) A formula is projective if and only if its class of models has the extension property.*

In [3] Ghilardi presents an algorithm called CHECK-PROJECTIVITY that checks if a formula is projective. We will use this algorithm in the next section to show that the formulas in $p(\Lambda_A)$ are projective. In this section we will state the algorithm and cite the results in [3] that we will need in the next section. Thus this section is a recapturing of results in [3]. Only at one point, which we will indicate, our notation differs slightly from the one used by Ghilardi. In this section $\vdash$ stands again for $\vdash_{\mathsf{IPC}}$.

An input formula A of the CHECK-PROJECTIVITY algorithm is assumed to have the form $\bigwedge(A_i \to A_i')$ (w.l.o.g. one may assume that all formulas have this form). The algorithm manipulates sets O, O' of so-called *signed subformulas* of A. A signed formula is an expression of the following six kinds: TB, FB, T_cB, F_cB, p^+, p^-, where B is a subformula of A and p is an atomic formula. (Ghilardi uses x for atomic formulas, and Δ and Γ for the sets we denote by O). Signed formulas of the form TB, FB are called *truth modalities*, signed formulas of the form T_cB, F_cB are called *context modalities* and signed formulas of the form p^+ or p^- are called *atomic modalities*. Signed formulas express conditions on models K: TB (FB) means that B is true (false) at the root of K, T_cB (F_cB) means that B is true in the context of K (the *context* of K are all nodes different from the root), p^+ is a synonymous of Tp and p^- means that p is false at the root and true in the context of K. If O is a set of signed subformulas then $K \models O$ means that K matches all the requirements expressed by O. A set O is inconsistent iff it contains either $F_c\top$ or TB and FB, or TB and F_cB, or p^+ and p^-, or Tp and p^- or p^+ and Fp, for some B or some atomic formula p. For sets O, TO is the set of truth modalities of O, T^+O is the set of atomic modalities of O and T_cO is the set of context modalities of O. For $X, Y, Z \in \{T, T^+, T_c\}$, XYO denotes $XO \cup YO$ and $XYZO$ denotes $XO \cup YO \cup ZO$. The sets FO, F^-O and F_cO are defined in a similar way.

The algorithm analyzes reasons for which we may have that A is false in the root and true in the context of a given model. From the rules of the algorithm given below it is easy to deduce that the following Soundness lemma holds. We will not need this lemma, but thought it instructive to state since it captures the idea behind the algorithm.

Lemma 7. *Let A be the input formula and $O_1, \ldots, O_n$ a stage of the algorithm. If K is a model such that $k \models A$ for all $k \in K$ different from the root, then $K \not\models A$ iff ($K \models O_1$ or ... or $K \models O_n$).*

The CHECK-PROJECTIVITY algorithm (The names that are given to the rules do not come from Ghilardi, but for our purposes it is convenient to have them available.)

Initialization

$$\{TA_1, FA'_1\}, \ldots, \{TA_n, FA'_n\}$$

Tableaux Rules

$$T\wedge \frac{O_1, \ldots, O_j \cup \{TB_1 \wedge B_2\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{TB_1, TB_2\}, \ldots, O_n}$$

$$T\top \frac{O_1, \ldots, O_j \cup \{T\top\}, \ldots, O_n}{O_1, \ldots, O_j, \ldots, O_n}$$

$$T\vee \frac{O_1, \ldots, O_j \cup \{TB_1 \vee TB_2\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{TB_1\}, O_j \cup \{TB_2\}, \ldots, O_n}$$

$$T\bot \frac{O_1, \ldots, O_j \cup \{T\bot\}, \ldots, O_n}{O_1, \ldots, O_{j-1}, O_{j+1}, \ldots, O_n}$$

$$F\wedge \frac{O_1, \ldots, O_j \cup \{FB_1 \wedge FB_2\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{FB_1\}, O_j \cup \{FB_2\}, \ldots, O_n}$$

$$F\top \frac{O_1, \ldots, O_j \cup \{F\top\}, \ldots, O_n}{O_1, \ldots, O_{j-1}, O_{j+1}, \ldots, O_n}$$

$$F\vee \frac{O_1, \ldots, O_j \cup \{FB_1 \vee B_2\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{FB_1, FB_2\}, \ldots, O_n}$$

$$F\bot \frac{O_1, \ldots, O_j \cup \{F\bot\}, \ldots, O_n}{O_1, \ldots, O_j, \ldots, O_n}$$

$$T \to \frac{O_1, \ldots, O_j \cup \{TB_1 \to B_2\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{FB_1, T_cB_1 \to B_2\}, O_j \cup \{TB_2\} \ldots, O_n}$$

$$F \to \frac{O_1, \ldots, O_j \cup \{FB_1 \to B_2\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{F_cB_1 \to B_2\}, O_j \cup \{TB_1, FB_2\}, \ldots, O_n}$$

$$T\neg \frac{O_1, \ldots, O_j \cup \{T\neg B\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{FB, T_c\neg B\}, \ldots, O_n}$$

$$F\neg \frac{O_1, \ldots, O_j \cup \{F\neg B\}, \ldots, O_n}{O_1, \ldots, O_j \cup \{F_c\neg B\}, O_j \cup \{TB\}, \ldots, O_n}$$

$$Tp\,\frac{O_1,\ldots,O_j\cup\{Tp\},\ldots,O_n}{O_1,\ldots,O_j\cup\{p^+\},\ldots,O_n}$$

$$Fp\,\frac{O_1,\ldots,O_j\cup\{Fp\},\ldots,O_n}{O_1,\ldots,O_j\cup\{p^-\},O_j\{F_cp\},\ldots,O_n}$$

Resolution Rule

$$\frac{O_1,\ldots,O_j\cup\{p^+\},\ldots,O_i\cup\{p^-\},\ldots,O_n}{O_1,\ldots,O_j\cup\{p^+\},\ldots,O_i\cup\{p^-\},\ldots,O_n,O_j\cup O_i\cup\{T_cp\}}$$

Simplification Rule

$$\frac{O_1,\ldots,O_j,\ldots,O_n}{O_1,\ldots,O_{j-1},O_{j+1},\ldots,O_n}$$
provided $A\wedge\bigwedge T_cO_j\vdash C$, for some $C\in F_c(O_j)$.

Auxiliary Simplification Rule

$$\frac{O_1,\ldots,O_j,\ldots,O_n}{O_1,\ldots,O_{j-1},O_{j+1},\ldots,O_n}$$
provided $O_j\supseteq O_i$ for some $i\neq j$ or provided O_j is inconsistent.

Ghilardi shows that the algorithm terminates after a finite number of steps. The following theorem is the one we will need. It is Theorem 4.2 of [3].

Theorem 4. *(Ghilardi [3]) A is projective iff each of its output sets contains at least one atomic modality.*