Propositional logics complexity and the sub-formula property

Edward Hermann Haeusler
PUC-Rio
Rio de Janeiro, Brasil
hermann@inf.puc-rio.br

In 1979 Richard Statman proved, using proof-theory, that the purely implicational fragment of Intuitionistic Logic ($\mathbf{M}_{\rightarrow}$) is PSPACE-complete. He showed a polynomially bounded translation from full Intuitionistic Propositional Logic into its implicational fragment. By the PSPACE-completeness of S4, proved by Ladner, and the Gœdel translation from S4 into Intuitionistic Logic, the PSPACE-completeness of $\mathbf{M}_{\rightarrow}$ is drawn. The sub-formula principle for a deductive system for a logic $\mathscr{L}$ states that whenever $\{\gamma_1,\ldots,\gamma_k\} \vdash_{\mathscr{L}} \alpha$ there is a proof in which each formula occurrence is either a sub-formula of α or of some of γ_i. In this work we extend Statman's result and show that any propositional (possibly modal) structural logic satisfying a particular formulation of the sub-formula principle is in PSPACE. If the logic includes the minimal purely implicational logic then it is PSPACE-complete. As a consequence, EXPTIME-complete propositional logics, such as PDL and the common-knowledge epistemic logic with at least 2 agents satisfy this particular sub-formula principle, if and only if, PSPACE=EXPTIME. We also show how our technique can be used to prove that any finitely many-valued logic has the set of its tautologies in *PSPACE*.

1 Introduction

In [17], R. Statman showed a polynomial-time reduction from Intuitionistic Propositional Logic into its implicational fragment. This reduction proves that Purely Implicational Minimal Logic is PSPACE-complete. The methods that Statman uses in [17] are based on proof-theory and Natural Deduction in Prawitz Style. The sub-formula principle for a Natural Deduction system $\mathscr{N}_{\mathscr{L}}$ for a logic $\mathscr{L}$ states that whenever α is provable from Γ, in $\mathscr{L}$, there is a derivation of α from a set of assumptions $\{\delta_1,\ldots,\delta_k\} \subseteq \Gamma$ built up only with sub-formulas of α and/or $\{\delta_1,\ldots,\delta_k\}$. In this article we show that the Validity problem for any propositional logic $\mathscr{L}$, with a Natural Deduction system that satisfies the sub-formula principle, is in PSPACE. Besides that, if $\mathscr{L}$ includes the usual $\rightarrow$-rules then PSPACE-complete.

In [13], we can found a general approach to Natural Deduction that allows the definition of general introduction and elimination rules in a way that, any intuitionistic logical constant (sometimes called operator or connective) that is expressible in terms of these general rules is also expressible by means of the intuitionistic logical constants ($\bot, \rightarrow, \wedge, \neg$ and $\vee$). Precisely, in [13] it is shown that any constant $\mathbf{c}$ determined by a set of introduction and elimination rules, in Natural Deduction style, is such that, there is a formula $F(A_1,\ldots,A_k)$ built up with constants from $\{\bot, \rightarrow, \wedge, \neg, \vee\}$, and, $\mathbf{c}(A_1,\ldots,A_k)$ is provable, iff, $F(A_1,\ldots,A_k)$ is provable in intuitionistic propositional logic. In order to prove this statement, namely, the functional completeness of intuitionistic logical constants, some proof-theoretical assumptions on the relationship between elimination and introduction rules are considered. Precisely, these assumptions have to do with the inversion principle that roughly states that an elimination rule for a logical constant $\mathbf{c}$ has to be a function of respective introduction rules of this $\mathbf{c}$. This means that the elimination rule is only determined by the introduction rules and this inversion principle[1].

[1] A good discussion on the form of the inversion principle can be found in [14]

U. Dal Lago and R. Harmer (Eds.): Developments
in Computational Models 2014 (DCM 2014).
EPTCS 179, 2015, pp. 1–16, doi:10.4204/EPTCS.179.1

Although strongly based on [13], the results shown here, do not explicitly need inversion principle. We do use a general form of introduction and elimination rules proposed in [13] and after extended by Roy Dickhoff and Nissim Francez to the term *general-elimination harmony*, such that, there is only one **c**-elimination rule associated to the many **c**-introduction rules. As we see in Section 2, this *general-elimination harmony* assumption is not needed, indeed. In [14] there is a very good discussion on the Higher-Level rules, proposed in [15] as an alternative and extension of Prawitz [13], and the *general-elimination harmony* as well. To sum it up , our technique relies in the sub-formula property only, as it is stated in the Definition 4 in the following section. We have to mention that [1] also provides an approach that does not require a full harmony between introduction and elimination rules, but [1] does not have the same goal as ours, in providing a computational complexity analysis based on proof-theoretical arguments as ours.

2 Translating a propositional logic $\mathscr{L}$ into $\mathbf{M}_{\rightarrow}$

Consider a propositional logic $\mathscr{L}$ with Natural Deduction system having introduction and elimination rules according to the following general schema. A **c**-introduction rule, as shown in Figure 1, derives $\mathbf{c}(\beta_1,\ldots,\beta_n,\phi_1^1,\ldots,\phi_{j_1}^1,\ldots,\phi_1^n,\ldots,\phi_{j_1}^n,\gamma_1,\ldots,\gamma_m)$, discharging occurrences of the bracket formulas $[\phi_1^i],\ldots,[\phi_{j_i}^i]$ in the respective derivation of β_i. For reasons of readability, we sometimes write $\mathbf{c}(\beta_i,\phi^j,\gamma_k)$ as a shortened (abbreviation) of $\mathbf{c}(\beta_1,\ldots,\beta_n,\phi_1^1,\ldots,\phi_{j_1}^1,\ldots,\phi_1^n,\ldots,\phi_{j_1}^n,\gamma_1,\ldots,\gamma_m)$[2]. An elimination rule, corresponding to the same operator **c** is shown in Figure 2 below. It is important to say that the discharging discipline is liberal, that is, when applying a rule, it is possible to discharge multiple assumptions including vacuous discharges. In this way, *this general schema cannot be used with sub-structural logics like Relevance and Linear Logics.* It is also important to note that each ϕ_j^i determine the type of the formula that can be discharged, not an instance of formula. For example, when representing the $\rightarrow$-introduction rule, there is only one type $\phi_1^1$and $c(\beta_1,\phi_1^1)$ is $\phi_1^1 \rightarrow \beta_1$. The $\rightarrow$-intro rule is shown in Figure 3 below as an instance of the general schema.

$$\dfrac{\begin{array}{ccccc}[\phi_1^1],\ldots,[\phi_{j_1}^1] & & & & [\phi_1^n],\ldots,[\phi_{j_n}^n] \\ | & & & & | \\ \beta_1 & & \ldots & & \beta_n\end{array}}{\mathbf{c}(\beta_1,\ldots,\beta_n,\phi_1^1,\ldots,\phi_{j_1}^1,\ldots,\phi_1^n,\ldots,\phi_{j_1}^n,\gamma_1,\ldots,\gamma_m)}$$

Figure 1: **c**-introduction rule schema

$$\dfrac{\mathbf{c}(\beta_i,\phi^j) \quad (\phi_i^j)_{i,j} \qquad \begin{array}{cc}[\beta_1] & [\beta_n] \\ |\ldots & \ldots| \\ \chi & \chi\end{array}}{\chi}$$

Figure 2: **c**-elimination rules schemata

[2]Consider usual $\vee$-intro rules. Both have $\beta_1 \vee \beta_2$ as conclusion, one has β_1 as premise and the other has β_2 as the only premise. According to our schema, β_2 plays the role of γ_1 in the first rule, while β_1 plays this role in the second $\vee$-intro rule

$$\frac{\begin{array}{c}[\phi_1^1]\\ |\\ \beta_1\end{array}}{\phi_1^1 \to \beta_1}$$

Figure 3: $\to$-introduction as an instance of the general schema

In elimination rules (see Figure 2), ϕ_i^j, $i = 1, j_i$, are called simple-minor premise, for each j. The premises χ are called discharging minor premises (d-minor premise). The premise $\mathbf{c}(\beta_i, \phi^j)$ is the major premise of the elimination rule. There may be more than one introduction rule, but at most one elimination rule. This elimination rule has one discharging minor premise for each introduction rule. As a matter of illustration we show below how the usual $\to$-elim rule is seen as instance of a general elimination schema. In order to see that, remember that the usual (minimal) implication has only one introduction rule that can be easily seen as an instance of an introduction schema. The elimination schema consider this instance of introduction schema, resulting in the shown rule. The usual form of the $\to$-elim is obtained by proving B instead of discharging it. For a more detailed explanation on these general rules, we recommend [13].

$$\frac{A \to B \qquad A \qquad \begin{array}{c}[B]\\ |\\ C\end{array}}{C}$$

Besides intuitionistic logic, some modal classical logics have Natural Deduction systems that conforms with our schema. For example, the Natural Deduction system presented in [12], page 75, for the logics S4 and S5 can be viewed as instances of our schemata, considering additional provisos for ensuring the soundness of the respective system. The rule for the $\diamond$-introduction has no proviso and is as follows.

$$\frac{A}{\diamond A}$$

The $\diamond$-elimination rule has a proviso, stated below, and it is as follows:

$$\frac{\diamond A \qquad \begin{array}{c}A\\ |\\ B\end{array}}{B}$$

where: (1) For S4, in the proof of the minor premise B, B only depends on formulas $\Box\alpha$ or $\neg\diamond\alpha$, but the formula occurrences A discharged by the rule (2) For S5, in the proof of the minor premise B, B only depends on formulas $\Box\alpha$, $\diamond\alpha$,$\neg\diamond\alpha$, $\neg\Box\alpha$, but the formula occurrences A discharged by the rule.

In abstract proof-theory, we would require that the principle of inversion holds concerning the elimination and introduction rules. This usually is among the requirements to have a normalization theorem holding for a Natural Deduction system. However, for our analysis, nothing is required but the sub-formula principle, defined in Definition 4.

A logical constant **c**, also called *operator*, can be a propositional connective or a modality. A Natural Deduction system for a logic $\mathscr{L}$ is a set of introduction and/or elimination rules for the operators of $\mathscr{L}$. Given a Natural Deduction system $\mathscr{N}_{\mathscr{L}}$ for a logic , the usual notions of derivation and proof of a formula from a set of assumptions are considered in this article. It is worth observing the difference between formula and formula occurrence.

Definition 1 (Linked formulas and sequences) *Let $\mathscr{N}_{\mathscr{L}}$ be a Natural Deduction for $\mathscr{L}$. Let Π be a derivation of a formula α from a set of assumptions Γ in $\mathscr{N}_{\mathscr{L}}$. Let β_1 and β_2 be two formula occurrences in Π. We say that β_1 is linked to β_2, if and only if, there is a sequence $\delta_0,\ldots,\delta_k$ of formula occurrences in Π, such that: (1) δ_0 is β_1 and δ_k is β_2, and, for each $i=0,\ldots,k-1$; (2) δ_i is major premise of an application of a* **c***-elim rule, in Π, and δ_{i+1} is a formula occurrence discharged by this application of elimination rule, or; (3) δ_i is a discharging minor premise of an application of a* **c***-elim rule, in Π, and δ_{i+1} is its conclusion, or; (4) δ_i is a premise of an application of a* **c***-intro rule, in Π, and δ_{i+1} is its conclusion.*

We call $\delta_0,\ldots,\delta_k$ a linking-sequence in Π. Observing the definition above, we note that a formula occurrence that is either a simple-minor premise of an application of a **c**-elim rule in Π or the conclusion of Π cannot be linked to any other formula occurrence in Π any more. Thus, the longest possible linking-sequences in any derivation Π have δ_k as a simple-minor premise or the conclusion of Π. Our definition of linking-sequence is similar, and generalizes, the definition of branches as stated in [12].

Definition 2 *Let Π be a derivation in $\mathscr{N}_{\mathscr{L}}$. Consider a linking-sequence $\delta_0,\ldots,\delta_k$ in Π. We say that $\delta_0,\ldots,\delta_k$ is a sub-formula linking-sequence, if and only if, for every $i=0,\ldots,k$, either δ_i is sub-formula of δ_{i+1}, or, δ_{i+1} is sub-formula of δ_i.*

Definition 3 (Derivations satisfying the Sub-Formula Property) *Let $\mathscr{L}$ be a logic and $\mathscr{N}_{\mathscr{L}}$ a Natural Deduction system for $\mathscr{L}$. Let Π be a derivation of α from $\{\delta_1,\ldots,\delta_k\}$ in $\mathscr{N}_{\mathscr{L}}$. We say that Π satisfy the sub-formula principle, if and only if, every linking-sequence in Π is a sub-formula linking-sequence and every formula occurring in this linking-sequence is either a sub-formula of α or sub-formula of some of the formulas in Γ.*

Definition 4 (Systems satisfying the Sub-Formula Property) *We say that a system $\mathscr{N}_{\mathscr{L}}$ satisfy the sub-formula property, if and only if, for every derivation Π of α from Γ in it, there is a derivation Π' of α from $\Gamma'\subseteq\Gamma$ satisfying the sub-formula property.*

When a logic $\mathscr{L}$ has a system $\mathscr{N}_{\mathscr{L}}$ satisfying the sub-formula property, we say that the logic $\mathscr{L}$ itself satisfies the sub-formula property. Thus, it is important to emphasize that to satisfy the sub-formula property means that the logic has a Natural Deduction system according the general rules shown here. This also means that sub-structural logics do not satisfy our notion of sub-formula property.

Proposition 1 *Let Π be a derivation satisfying the sub-formula principle. If Π' is a sub-derivation of Π then Π' satisfies the sub-formula principle too.*

Proof of proposition. We observe that a sub-sequence of a linking-sequence that is a sub-formula linking-sequence is a sub-formula linking-sequence too. Since every linking-sequence of Π is a sub-formula linking sequence, then every linking sequence of Π' is a linking sequence too.

Q.E.D.

In what follows we define a translation $\star$ from $\mathscr{L}$ into $\mathbf{M}_{\rightarrow}$, such that, α is provable in $\mathscr{L}$, if and only if, $\alpha^{\star}$ is provable in $\mathbf{M}_{\rightarrow}$. In order to define $\star$, we need auxiliary functions $\mathscr{M}$ and $\mathscr{A}$, defined in the sequel. In the following definition, p_{ω} is a notation used to uniquely identify the symbol p indexed by the string (word) ω. It this way we ensure that p is the unique propositional symbol that is named by means of the word ω, such that, p_{ω_1} and p_{ω_2} are equal, if and only if, ω_1 and ω_2 are the same word (string). We remember that we call operator any propositional connective or modality. We consider that the logic $\mathscr{L}$ has a finite set $\{\mathbf{c_1},\ldots,\mathbf{c_k}\}$ of operators. For reasons of readability, we do not consider the minimal implication $\rightarrow$ in this set, even in the case that the logic has it as one of its propositional connectives. As we will see, the implication is the only logical constant (operator) that is not translated. So, in what follows we consider only logics including the purely minimal implicational logic.

The propositional logics considered in this article have in their Natural Deduction systems the usual $\rightarrow$-intro and $\rightarrow$-elim rules.

The following definitions provide us axioms schemata concerning each **c**-introduction and/or **c**-elimination rule.

Definition 5 (ι-axiom) *Consider an introduction rule r for an operator* **c** *as shown in Figure 1. The ι-axiom concerning this rule schema r instantiated to formulas β_i and $\phi^i_1,\ldots,\phi^i_{j_i}$, denoted by $\iota(r,\beta_i,\phi^j,\gamma_k)$, is the following implicational formula:*

$$(\phi^1_1 \rightarrow (\ldots \rightarrow (\phi^1_{j_1} \rightarrow \beta_1))) \rightarrow \ldots (\phi^n_1 \rightarrow (\ldots \rightarrow (\phi^n_{j_n} \rightarrow \beta_n))) \rightarrow p_{\mathbf{c}(\beta_i,\phi^j,\gamma_k)})$$

Definition 6 (ε-axiom) *Consider a formula χ of a logic $\mathscr{L}$ and an elimination rule r for* **c***, as shown right in Figure 2 instantiated to ϕ^j, β_i and χ. The ε-axiom concerning this rule schema and χ, denoted by $\varepsilon(r,\chi,\beta_i,\phi^j)$, is the following implicational formula:*

$$\phi^1_1 \rightarrow (\ldots (\phi^n_{j_n} \rightarrow (..(\beta_1 \rightarrow \chi) \rightarrow \ldots (\beta_n \rightarrow \chi) \rightarrow (p_{\mathbf{c}(\beta_i,\phi^j)} \rightarrow \chi))))$$

For example, considering the usual $\vee$-introduction ($\vee - intro_1$ and $\vee - intro_2$) and $\vee$-elimination ($\vee - elim$) natural deduction schemata, we have that: (1) $\iota(\vee - intro_1,\beta_1,\beta_2)$ is $\beta_1 \rightarrow p_{\vee(\beta_1,\beta_2)}$; (2) $\iota(\vee - intro_2,\beta_1,\beta_2)$ is $\beta_2 \rightarrow p_{\vee(\beta_1,\beta_2)}$, and; (3) $\varepsilon(\vee - elim,\beta_1,\beta_2,\chi)$ is $(\beta_1 \rightarrow \chi) \rightarrow ((\beta_2 \rightarrow \chi) \rightarrow (p_{\vee(\beta_1,\beta_2)} \rightarrow \chi))$.

Definition 7 (Atomizing Operators) *The mapping $\mathscr{M}$ from the language of $\mathscr{L}$ into the one of $\mathbf{M}_{\rightarrow}$ is defined inductively, as follows:* **Atoms** $\mathscr{M}(p) = p$*, if p is a propositional letter;* **Implication** $\mathscr{M}(\alpha_1 \rightarrow \alpha_2) = \mathscr{M}(\alpha_1) \rightarrow \mathscr{M}(\alpha_2)$*;* **Operators** $\mathscr{M}(\mathbf{c_m}(\beta_i,\phi^j,\gamma_k)) = p_{\mathbf{c_m}(\beta_i,\phi^j,\gamma_k)}$*, if* $\mathbf{c_m}$ *is an operator of* $\mathscr{L}$.

The second clause in the definition above is only used when the language of $\mathscr{L}$ includes the minimal implication $\rightarrow$. Otherwise it is not used and the translation is also well-defined. In the following we define an auxiliary function that for each formula α yields the set of implicational formulas that express the "deductive meaning"[3] of the elimination and introduction rules for each operator in $\mathscr{L}$.

[3]This informal expression is made precise in the statement of Theorem 1

Definition 8 (Axiomatizing Operators) *Given a formula α in $\mathscr{L}$, the mapping $\mathscr{A}^{\alpha}$ from the language of $\mathscr{L}$ into (finite) sets of formulas in the language of $\mathbf{M}_{\rightarrow}$ is defined inductively, as follows.*

- **Atoms** $\mathscr{A}^{\alpha}(p) = \emptyset$;
- **Implication** $\mathscr{A}^{\alpha}(\alpha_1 \rightarrow \alpha_2) = \mathscr{A}^{\alpha}(\alpha_1) \cup \mathscr{A}^{\alpha}(\alpha_2)$;
- **Operators** $\mathscr{A}^{\alpha}(\mathbf{c_m}(\beta_i, \phi^j, \gamma_k)) =$

$$\begin{array}{c} \{\iota(r, \mathscr{M}(\beta_i), \mathscr{M}(\phi^j), \mathscr{M}(\gamma_k)) / r \text{ is a } \mathbf{c_m}\text{-intro rule}\} \\ \cup \\ \{\varepsilon(r, \mathscr{M}(\chi), \mathscr{M}(\beta_i), \mathscr{M}(\phi^j)) / r \text{ is a } \mathbf{c_m}\text{-elim rule and } \chi \in sub(\alpha)\} \end{array}$$

Lemma 1 *Let $\mathscr{L}$ be a logic having a Natural Deduction system satisfying sub-formula property. Let Π be a proof of α from Γ in $\mathscr{L}$. There is a derivation Π' of $\mathscr{M}(\alpha)$ from $\Gamma' \subseteq \mathscr{M}(\Gamma) \cup \bigcup_{\gamma \in \Gamma} \mathscr{A}^{\gamma}(\Gamma, \alpha) \cup \mathscr{A}^{\alpha}(\Gamma, \alpha)$ in $\mathbf{M}_{\rightarrow}$. We use the notation $\mathscr{A}^{\alpha}(\Gamma)$ to denote $\bigcup_{\gamma \in \Gamma} \mathscr{A}^{\alpha}(\gamma)$ and "Γ, α" to denote $\Gamma \cup \{\alpha\}$.*

Proof of Lemma. Since $\mathscr{L}$ satisfies the sub-formula principle, we can consider that Π is a derivation satisfying the sub-formula principle. The proof proceeds by induction on the size of this derivation. The basis is the derivation of α from α itself, and hence, Π' is $\mathscr{M}(\alpha)$ only. The inductive step is according the last rule applied in the derivation satisfying the sub-formula principle. There are only two (general) cases:

- **Last rule is a $\mathbf{c_m}$-intro rule.** Then Π is as following:

$$\dfrac{\begin{array}{ccc} [\phi_1^1], \ldots, [\phi_{j_1}^1] & & [\phi_1^n], \ldots, [\phi_{j_n}^n] \\ \Pi_1^{\star} & & \Pi_n^{\star} \\ \beta_1 & \ldots & \beta_n \end{array}}{\mathbf{c}(\beta_1, \ldots, \beta_n, \phi_1^1, \ldots, \phi_{j_1}^1, \ldots, \phi_1^n, \ldots, \phi_{j_1}^n, \gamma_1, \ldots, \gamma_m)}$$

Where the last rule is r, a $\mathbf{c}$-introduction rule. Consider $\iota(r, \beta_i, \phi^j, \gamma_k)$ the implicational formula related to $\mathbf{c}$. By Proposition 1 $\Pi_i^{\star}$ satisfies the sub-formula principle, $i = 1, n$. By inductive hypothesis, for each $i = 1, n$, there is a derivation Π_i' satisfying the statement of the lemma. Each Π_i' is of the following form.

$$\begin{array}{c} [\mathscr{M}(\phi_1^i)], \ldots, [\mathscr{M}(\phi_{j_i}^i)] \\ \Pi_i' \\ \mathscr{M}(\beta_i) \end{array}$$

Thus, for each i we can build, from Π_i', a derivation as the following, which we will denote by Σ_i.

$$\dfrac{\begin{array}{c}[\mathscr{M}(\phi_1^i)],\ldots,[\mathscr{M}(\phi_{j_i}^i)]\\ \Pi_i'\\ \mathscr{M}(\beta_i)\end{array}}{\begin{array}{c}\mathscr{M}(\phi_{j_i-1}^i)\to(\mathscr{M}(\phi_{j_i}^i)\to\mathscr{M}(\beta_i))\\ \vdots\\ (\mathscr{M}(\phi_1^i)\to(\ldots\to(\mathscr{M}(\phi_{j_i}^i)\to\mathscr{M}(\beta_i))..))\end{array}}\to\text{-i}$$

Finally, the derivation below is a derivation in $\mathbf{M}_{\to}$ of the form stated by the lemma, that is, of $\mathscr{M}(\mathbf{c}(\beta_1,\ldots,\beta_n,\phi_1^1,\ldots,\phi_{j_1}^1,\ldots,\phi_1^n,\ldots,\phi_{j_1}^n,\gamma_1,\ldots,\gamma_m)) = p_{\mathbf{c}(\beta_i,\phi^j,\gamma_k)}$. For reasons of space we abbreviate $(\mathscr{M}(\phi_1^i)\to(\ldots\to(\mathscr{M}(\phi_{j_i}^i)\to\mathscr{M}(\beta_i))..))$ by B_i.

$$\dfrac{\begin{array}{ll}[\mathscr{M}(\phi_1^2)],\ldots,[\mathscr{M}(\phi_{j_2}^2)] & \dfrac{\begin{array}{ll}[\mathscr{M}(\phi_1^1)],\ldots,[\mathscr{M}(\phi_{j_1}^1)] & \\ \Sigma_1 & \\ B_1 & \iota(r,\mathscr{M}(\beta_i),\mathscr{M}(\phi^j),\mathscr{M}(\gamma_k))\end{array}}{B_2\to(B_3\to\ldots(B_n\to p_{\mathbf{c}(\beta_i,\phi^j,\gamma_k)}))}\\ \Sigma_2 & \\ B_2 & \end{array}}{\begin{array}{c}B_3\to\ldots(B_n\to p_{\mathbf{c}(\beta_i,\phi^j,\gamma_k)})\\ \vdots\\ \vdots\\ \vdots\\ \vdots\end{array}}$$

$$\dfrac{\begin{array}{ll}[\mathscr{M}(\phi_1^n)],\ldots,[\mathscr{M}(\phi_{j_n}^n)] & \\ \Sigma_n & \\ B_n & B_n\to p_{\mathbf{c}(\beta_i,\phi^j,\gamma_k)}\end{array}}{p_{\mathbf{c}(\beta_i,\phi^j,\gamma_k)}}$$

The derivation above shows the existence of a derivation of the translated conclusion from the translated premises.

• **Last rule is a $\mathbf{c_m}$-elim rule.** Then $\Pi^\star$ is as following:

$$\dfrac{\begin{array}{llcl}\Pi_{pm}^\star & \Pi_{i,j}^\star & [\beta_1] & [\beta_n]\\ & & \Pi_1^\star & \Pi_n^\star\\ \mathbf{c}(\beta_i,\phi^j) & (\phi_i^j)_{i,j} & \chi\quad\ldots & \chi\end{array}}{\chi}$$

By Proposition 1 $\Pi_{pm}^\star$, $\Pi_{i,j}^\star$ and $\Pi_i^\star$, $i=1,n$, $j=1,k$ satisfies the sub-formula principle. By inductive hypothesis, for each i (and j), there are derivations Π_i', $\Pi_{i,j}'$ and Π_{pm}' satisfying the

statement of the lemma. Using, for each i and j the derivations $\Pi'_{i,j}$ we obtain the following derivation from the ε-axiom $\varepsilon(r, \mathscr{M}(\chi), \mathscr{M}(\beta_i), \varphi_{i,j})$.

$$\dfrac{\begin{array}{c}\Pi'_{1,2}\\ \mathscr{M}(\phi^1_2)\end{array} \qquad \dfrac{\begin{array}{c}\Pi'_{1,1}\\ \mathscr{M}(\phi^1_1)\end{array} \qquad \mathscr{M}(\phi^1_1) \to (\ldots(\mathscr{M}(\phi^n_{j_n}) \to (..(\mathscr{M}(\beta_1) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi)))))}{\mathscr{M}(\phi^1_2) \to (\ldots(\mathscr{M}(\phi^n_{j_n}) \to (..(\mathscr{M}(\beta_1) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi)))))}}{\mathscr{M}(\phi^1_3) \to (\ldots(\mathscr{M}(\phi^n_{j_n}) \to (..(\mathscr{M}(\beta_1) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi)))}$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\dfrac{\begin{array}{c}\Pi'_{n,j_n}\\ \mathscr{M}(\phi^n_{j_n})\end{array} \quad \begin{array}{c}\vdots\\ \mathscr{M}(\phi^n_{j_n}) \to (..\mathscr{M}(\beta_1) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi)))\end{array}}{(\mathscr{M}(\beta_1) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi)))}$$

We denote the derivation above by Σ. Using Π'_i, for each i, we have the following derivation.

$$\dfrac{\begin{array}{c}[\mathscr{M}(\beta_i)]\\ \Pi'_i\\ \mathscr{M}(\chi)\end{array}}{\mathscr{M}(\beta_i) \to \mathscr{M}(\chi)}$$

Then the derivation below, built using the above derivations, shows how to conclude the translation of the conclusion of $\Pi^\star$ from the translation of its premises and the implicational introduction and elimination schemata. $\varepsilon(r, \mathscr{M}(\chi), \mathscr{M}(\beta_i), \varphi_{i,j})$ is the ε-axiom that implements the implicational elimination on the implicational fragment of minimal logic. We remind the reader that $\mathscr{M}(\mathbf{c}(\beta_i, \phi^j))$ is $p_{\mathbf{c}(\beta_i,\phi^j)}$. We can check and find out that the derivation satisfies the lemma.

$$\dfrac{\begin{array}{c}[\mathscr{M}(\beta_1)]\\ \Pi'_1\\ \mathscr{M}(\beta_1) \to \mathscr{M}(\chi)\end{array} \qquad \begin{array}{c}\Sigma\\ (\mathscr{M}(\beta_1) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi))\end{array}}{(\mathscr{M}(\beta_2) \to \mathscr{M}(\chi)) \to \ldots(\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi))}$$

$$\dfrac{\begin{array}{c}\Pi'_{pm}\\ p_{\mathbf{c}(\beta_i,\phi^j)}\end{array} \qquad \dfrac{\begin{array}{c}[\mathscr{M}(\beta_n)]\\ \Pi'_n\\ \mathscr{M}(\beta_n) \to \mathscr{M}(\chi)\end{array} \qquad \begin{array}{c}\vdots\\ \vdots\\ (\mathscr{M}(\beta_n) \to \mathscr{M}(\chi)) \to (p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi))\end{array}}{p_{\mathbf{c}(\beta_i,\phi^j)} \to \mathscr{M}(\chi)}}{\mathscr{M}(\chi)}$$

Q.E.D.

For logics that have rules with a proviso, the respective implicational axioms ι and ε must reflect the corresponding conditions. Whenever we have these axioms we have Lemma 1 holding for the respective logics. For example, this can be done with the Natural Deduction rules provided in [12] for S4 and S5 and already discussed before in this article.

Proposition 2 *Let $\mathscr{L}$ be a propositional logic satisfying the sub-formula principle. Consider the following translation $\star$ from formulas of $\mathscr{L}$ into formulas of $\mathbf{M}_{\rightarrow}$: Let $\mathscr{A}^{\alpha}(\alpha)$ be $\{\varphi_1,\ldots,\varphi_k\}$ $\alpha^{\star}$ is defined as $\varphi_1 \rightarrow (\varphi_2 \rightarrow \ldots(\varphi_k \rightarrow \mathscr{M}(\alpha)))$. Thus, $\vdash_{\mathscr{L}} \alpha$ if and only if $\vdash_{\mathbf{M}_{\rightarrow}} \alpha^{\star}$.*

Proof of proposition. The proposition follows immediately from Lemma 1.

Q.E.D.

Proposition 3 *If α is any propositional formula, then the number of sub-formulas of α is polynomially bounded on the length of α.*

Proof. Each sub-formula of α is determined by the main connective of it. In α there is at most one connective or logical constant by symbol position in α. Thus, the number of sub-formulas of α is bounded by the length of α.

We can see that the size of $\alpha^{\star}$ is $O(m^3)$, if m is the size of α. The formula $\alpha^{\star}$ depends on three choices of sub-formulas of α. This polynomial bound on the size of $\alpha^{\star}$ entails the following main conclusions of our article.

Theorem 1 *If $\mathscr{L}$ satisfies the sub-formula principle then the problem of knowing whether α, a formula of $\mathscr{L}$, is provable or not, is in PSPACE. If $\mathscr{L}$ includes $\mathbf{M}_{\rightarrow}$ then this problem, also known as* **Validity***, is PSPACE-complete.*

Proof of theorem. From the PSPACE-completeness of provability in $\mathbf{M}_{\rightarrow}$ and the polynomial reduction of L $\mathscr{L}$ to $\mathbf{M}_{\rightarrow}$, we have that **Validity** in $\mathscr{L}$, namely, knowing whether α, a formula of $\mathscr{L}$, is provable or not, is in PSPACE. If $\mathscr{L}$ includes $\mathbf{M}_{\rightarrow}$, and since $\mathbf{M}_{\rightarrow}$ is PSPACE-complete, then **Validity** is PSPACE-complete.

Q.E.D.

Corollary 1 *If $\mathscr{L}$ satisfies the sub-formula principle and includes $\mathbf{M}_{\rightarrow}$ then the problem of knowing whether α, a formula of $\mathscr{L}$, is invalid or not, is PSPACE-complete.*

Proof of corollary. Since *PSPACE* is a deterministic class, we have that CoPSPACE=PSPACE. The problem stated in the corollary is in CoPSPACE, by definition.

Q.E.D.

3 A brief discussion on extending the results to finitely many-valued propositional logics

An stronger version of the statement of Theorem 1 can be used to prove that some finitely many-valued logics have **Validity** (cf. Theorem 1) in PSPACE. In [5] it is provided a general schema to define normalizable Natural Deduction systems for some finitely many-valued logics. The kind of Natural Deduction rules used in [5] are similar, in fact almost the same, to those shown here in this article. However, when

formalizing logics with more than 2 truth-values, auxiliary formulas are considered in order to perform the *binary print*, explained in sequel, of a truth-value. We briefly discuss here the main idea on an extension of Theorem 1 and how it could be used to prove that Validity/Provability in a finitely many-valued logic is in PSPACE, by adjusting the sub-formula principle defined in Definition 4. More details on the Natural Deduction systems for finitely many-valued logics as used here can be found in [5].

Instead of defining a rule for each of the $n > 2$ truth-values, it is used a method to reduce many-valued semantics to bivalent one. We use the three-valued logic $Ł_3$, due to Łukasiewicz to illustrate our discussion. See [4] for the original article (1920) on $Ł_3$ and [9] for a very good and concise presentation of many-valued logics.

The truth-values in $Ł_3$ are $\{0, i, 1\}$, where 0 and i are undesignated values and 1 is the designated value. With the sake of providing some context on the meaning of these three values, we can say that the truth-value i means indetermination or in numerical terms that it corresponds[4] to 1/2. Thus, a formula having i as value should not be taken as a true formula. The truth-values i and 0 in this case are considered as undesignated. For some many-valued logics, a formula is valid, iff, it yields designated value for each possible truth-value assignment. This is just the case with $Ł_3$. $Ł_3$ has the truth-table shown in Table 4 for the implication $\rightarrow$ and in Table 5 for the negation $\neg$. We explain what is a *bitprint* of a truth-value in the sequel. Consider a function $t(P)$ that yields 1 if P is designated and 0 if it is undesignated. Using the formula $\phi(P)$, defined as $\neg P \rightarrow P$, it is possible to identify i with the pair $\langle 0,1\rangle$, 1 with the pair $\langle 1,1\rangle$ and 0 with $\langle 0,0\rangle$, that is, each truth-value x is identified by the pair $\langle t(x), \phi(x)\rangle$. In [10], the sequence $\langle P, \phi(P)\rangle$ that denotes each one of the values in $\{0, i, 1\}$ is called the bitprint of the respective truth-value. It is interesting to note that the pair $\langle 1,0\rangle$ has no truth-value associated to it. Besides that, it is worth observing that for a many-valued logic with k truth-values, we have to find $maxint(log_2(k)) - 1$ separating formulas for performing the role of $\phi(P)$. Using bitprints, the truth-table of the $\supset$ is as shown in Table 6. In this way the line of the truth-table saying that $P \supset Q$ yields 1, when P is i and Q is 1, is coded as: P is 0, $\phi(P)$ is 1, Q is 1 and $\phi(Q)$ is 1. This line, the fourth line of the truth-table, is related to the Natural Deduction rule shown in Figure 7. The rule in Figure 8, very similar to an elimination rule in Natural Deduction, corresponds to the third line in Table 6. Because some pairs $\langle t(P), \phi(P)\rangle$ do not correspond to any truth-value, there is need of a Natural Deduction rule to take care of this. The rule that is related to the pair $\langle 1,0\rangle$ in $Ł_3$ is shown in Figure 9. Besides that, rules as shown in Figure 10, are need in order to have a complete system, for, in the general case, formulas of the form $\phi(X)$, may have X as composed formulas. Figure 10 shows the case when X is $P \rightarrow Q$. This is the case related to the first line of the truth-table in Figure 6.

Consider a general propositional connective c, that forms formulas of the form $c(A_1, \ldots, A_n)$. The rules that discharge $c(A_1, \ldots, A_n)$ are considered c-introduction rules. The rules that have $c(A_1, \ldots, A_n)$ as premise (also called major premise) are the c-elimination rules. The same is said about $\phi(c(A_1, \ldots, A_n))$, where ϕ is a separating formula. In [5] it is shown how to eliminate sequences that starts with c-intro rules and ends with c-elimination rules. Derivations having this kind of sequences are said to be non-normal. Normal derivations do not have these sequences, and, do not have c-rules, like the one shown in Figure 9, proving premise of c-elimination rules either. It is the case that any non-normal derivation of a formula α from a set of formulas Γ can be effectively transformed in a normal derivation of α from $\Gamma' \subseteq \Gamma$. This is the statement of the normalization theorem for finitely many-valued logics proved in [5].

A branch in a derivation in a Natural Deduction system for a finitely many-valued logic, as those discussed in this section, is any sequence of formulas starting in an undischarged hypothesis and ending in the minor premise of an elimination rule. This is essentially what is stated in Definition 1. One of the

[4] This depends whether we are in an epistemic position, the former, or an ontic position, the later case.

P	Q	$P \supset Q$
1	1	1
1	i	i
1	0	0
i	1	1
i	i	1
i	0	i
0	1	1
0	i	1
0	0	1

Figure 4: Truth-table for $\supset$ in Ł_3

P	$\neg P$
1	0
i	i
0	1

Figure 5: Truth-table for $\neg$ in Ł_3

P	Q	$P \supset Q$	$t(P)$	$\phi(P)$	$t(Q)$	$\phi(Q)$	$t(P \supset Q)$	$\phi(P \supset Q)$
1	1	1	1	1	1	1	1	1
1	i	i	1	1	0	1	0	1
1	0	0	1	1	0	0	0	0
i	1	1	0	1	1	1	1	1
i	i	1	0	1	0	1	1	1
i	0	i	0	1	0	0	0	1
0	1	1	0	0	1	1	1	1
0	i	1	0	0	0	1	1	1
0	0	1	0	0	0	0	1	1

Figure 6: Truth-table for $\supset$ in Ł_3 using bitprints in the 6 last columns

$$\dfrac{\begin{array}{cc}[P \supset Q] & [P] \\ \vdots & \vdots \\ C & C\end{array} \qquad \phi(P) \qquad Q \qquad \phi(Q)}{C}$$

Figure 7: $\supset$-introduction rule for the logic Ł_3 related to the fourth line in Figure 6

properties of normal derivations is that for any branch starting with a formula A and ending with B in this derivation, the formulas that occur in it are either sub-formulas of A, or B, or both. This is one of the main features of normal proofs, in usual systems as Classical, Intuitionistic and Minimal logic. As was already mentioned in the introduction, normalization implies in the sub-formula property in many logics. A quick inspection on the form of the general Natural Deduction introduction and elimination rules as defined in Section 2 shows that for these rules the premises are either sub-formula of the conclusion or vice-versa. The discharged formulas of both, introduction and elimination are also sub-formula either of the major premise or of the conclusion. In fact, the sub-formula principle holds because of this feature

$$\dfrac{\begin{array}{cccccc}[P] & [\phi(P)] & & & & \\ \vdots & \vdots & & & & \\ C & C & Q & \phi(Q) & P \supset Q & \end{array}}{C}$$

Figure 8: $\supset$-elimination-like rule for the logic Ł$_3$ related to the third line in Figure 6

$$\dfrac{\begin{array}{cc} & [\phi(P)] \\ & \vdots \\ P & C \end{array}}{C}$$

Figure 9: $\supset$-rule for the logic Ł$_3$ related to the value $\langle 1,0\rangle$ that is not a bitprint

$$\dfrac{\begin{array}{ccccc}[P] & [\phi(P)] & [Q] & [\phi(Q)] & [\phi(P \supset Q)] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C & C & C & C & C \end{array}}{C}$$

Figure 10: $\phi(\supset)$-intro rule in Ł$_3$ regard to $\phi(P \supset Q)$ and first line in Figure 6

just discussed. However, the format of the Natural Deduction rules for the finitely many-valued logics does not show this perfect relationship. In their case, the use of the separating formulas $\phi(X)$ disturbs this relationship based only on sub-formula occurrence in rules. In order to apply our technique to finitely many-valued logic, we will consider the sub-formula relationship through application of any formula in the set of separating formulas $\{\phi_1,\ldots,\phi_n\}$, so that the relationship between conclusions , premises and discharged formulas, in this case of Natural Deduction rule, includes the separating formulas. For example, P, the discharged formula, and Q, a minor premise, are sub-formulas of the major premise $P \supset Q$, of the rule in Figure 8. However, $\phi(P)$, a discharged formula, and $\phi(Q)$, other minor premise, are not sub-formulas. They are, in fact, images of sub-formulas of the major premise. So, we can say that the formulas occurrences in a Natural Deduction rule, as in Figure 8, are ϕ-related. This is defined below.

Definition 9 *Consider a Natural Deduction rule in a system for finitely many-valued logic using a set $\Phi = \{\phi_1,\ldots,\phi_n\}$ of separating formulas. The relationship between premises, conclusions and discharged formulas under sub-formula of ϕ_i, $i = 1,n$, image of a sub-formula is denoted by Φ-sub-formula relationship.*

It is a trivial fact that if $\Phi \subseteq \Phi'$ and α is Φ-sub-formula of β, or vice-versa, then they are Φ'-sub-

formula too. From the form of a branch beginning with β_1 and ending in β_k in a normal derivation we can see that any formula in this branch is either a sub-formula of β_1 or of β_n, and for any $i = 1, n-1$, β_i is Φ-sub-formula of β_{i+1}, or vice-versa. Using this, we can use the same polynomial simulation used for proving Theorem 1 for any finitely many-valued logic in $\mathbf{M}_{\rightarrow}$. It is worth noting that the conclusion and hypothesis of any normal derivation impose that the set of formulas that occur in it are either sub-formulas of the conclusion or of the hypothesis. Does not matter the fact that some of then are Φ-images of some other sub-formula.

As a matter of illustration of the discussion in the last paragraph, we show the implicational schemata for the elimination, introduction and ϕ-rules, including the ϕ-rules related to no bitprint representation (as the one shown in Figure 9). For example, in the case of Ł$_3$, the implicational schema for the rule shown in Figure 7 is below, where C is a sub-formula of either the conclusion α of the derivation, or of some hypothesis of the derivation. There must be one such formula for each C, this already includes the images under the separating formula ϕ. We remember that $p_{(P \supset Q)}$ is the new (fresh) propositional variable associated to $P \supset Q$. The same holds for the propositional variables $p_{\phi(P)}$ and $p_{\phi(Q)}$

$$(p_{(P \supset Q)} \rightarrow C) \rightarrow ((P \rightarrow C) \rightarrow (p_{\phi(P)} \rightarrow (Q \rightarrow (p_{\phi(Q)} \rightarrow C))))$$

The formula below is the implicational schema for the corresponding ϕ-introduction rule in Figure 10.

$$(P \rightarrow C) \rightarrow ((p_{\phi(P)} \rightarrow C) \rightarrow ((Q \rightarrow C) \rightarrow ((p_{\phi(Q)} \rightarrow C) \rightarrow ((p_{(P \supset Q)} \rightarrow C) \rightarrow C))))$$

The implicational schema for the $\supset$-elimination rule (Figure 8) and the rule shown in Figure 10 are show in the sequel.

$$(P \rightarrow C) \rightarrow ((p_{\phi(P)} \rightarrow C) \rightarrow (Q \rightarrow (p_{\phi(Q)} \rightarrow (p_{(P \supset Q)} \rightarrow C))))$$

$$P \rightarrow ((p_{\phi(P)} \rightarrow C) \rightarrow C)$$

Important Observation: The reader must have noted that the forms of the implicational schemata are not totally in formal agreement with the ι-axioms presented in the previous sections and used to prove that **Validity** of the respective logic is in PSPACE. This is only a formal disagreement, indeed. This can be seen, if we consider the usual conjunction $\wedge$. The $\wedge$ very well-known introduction rules is the left rule below, but it can be also defined as drawn by the right rule.

$$\frac{A \qquad B}{A \wedge B} \qquad \frac{\begin{array}{c}[A \wedge B] \\ \vdots \\ A \qquad B \qquad C\end{array}}{C}$$

Finally, from the proof-theoretical results obtained in [5] we can state the following theorem. A detailed proof of this theorem and a more systematic treatment of the subject discussed in this section is omitted due to the lack of space.

Theorem 2 *Let $\mathscr{L}$ be a propositional k-valued logic with finitely-many connectives with semantics determined by functional truth-tables. The* **Validity** *decision problem for $\mathscr{L}$ is in PSPACE.*

Proof. The fact that the connectives are defined by functional truth-tables, i.e. deterministic and defined for every k-value truth-tables, entails that the Natural Deduction system is sound and complete regarding the semantics provided by the truth-tables. Consider the set $\{\phi_1, \ldots, \phi_{log_2(k)}\}$ of separating formulas that provide bitprints for each of the k truth-values. There is at most one rule for each line in each truth-value. Since the number of lines in each truth-table is fixed (depends only on k), there is at most one implicational schema for each line. Let $F(k)$ be this number. $F(k)$ depends only on k. In fact $F(k)$ is induced by the arity of each connective of the language of $\mathscr{L}$. Taking into account a formula α in the language of $\mathscr{L}$. Each implicational schema depends also on the number of sub-formulas of α, including those sub-formulas that are image under each ϕ_i. Let us say that n is the number of sub-formulas of the formula α, that is linear on the size of α. Thus, the total number of implicational schemata is at most $n^{a \star log_2(k)}$, where $a = F(k)$. This upper-bound is polynomial in n. Thus, the translation used in Theorem 1 can the used here to prove that **Validity** is in PSPACE. It is worth noting that $F(k)$ depends only on the logic.

Q.E.D.

If the Validity of a logic $\mathscr{L}$ is in PSPACE and it includes $\mathbf{M}_{\rightarrow}$ then its (Validity decision problem) is PSPACE-complete. A logic $\mathscr{L}$ includes $\mathbf{M}_{\rightarrow}$, iff, the $\rightarrow$-intro and $\rightarrow$-elim are derived rules in $\mathscr{L}$.

4 Conclusion

We have shown that structural propositional logics satisfying the sub-formula principle have their provability problem in PSPACE and that if they include $\mathbf{M}_{\rightarrow}$ then they are PSPACE-complete problems. We used proof-theory to show this. An important consequence of our result is to show that some logics hardly satisfy the sub-formula principle in the terms here presented. Any propositional logic that we believe that it is beyond PSPACE cannot satisfy the sub-formula principle.

An immediate application of Theorem 1 shows that hardly some well-known logics, as Propositional Dynamic Logic (*PDL*, see [6]) and the common-knowledge epistemic logic with at least 2 agents [7], have Natural Deduction systems satisfying the sub-formula principle. In [8], for example, it is shown a Natural Deduction system for PDL, with normalization theorem. However, one of its rules, the iteration rule, does not satisfy the sub-formula principle in the terms stated in this article, since it is an infinitary ND-system. Anyway it does not satisfy sub-formula relationship between premises and conclusion either. Any logic that is EXPTIME-complete, as the ones just discussed, cannot satisfy the sub-formula principle too.

Any PSPACE-complete problem is NP-Hard. Since Classical propositional Logic can be specified using the general elimination and introduction rules presented here, then it is PSPACE-HARD and so its also NP-Hard. We know, by the truth-table method that Classical propositional logic is in NP, so it is NP-complete.

We discussed the extension of this proof-theoretical way of providing upper-bounds for provability to finitely many-valued logics, concluding that their respective provability (validity) problems are in PSPACE. This is in our opinion the far we can go. The fact that the set of valid formulas of a logic is in PSPACE may be of no help in some particular analysis. There are many-valued logics, as for example Kleene's logic, that have no tautologies and this is of course trivially in PSPACE. On the other hand, knowing that many-valued usual logics are in PSPACE seems to be an interesting fact in general. There are finitely many-valued logics that have non-trivial set of tautologies. As far as we know there should be no novelty in proving that Validity in finitely many-valued logics is in PSPACE. A polynomially space-bounded algorithm for testing Validity can be designed by observing that evaluating the truth-value of

a formula is polynomial on the size of the formula and that the evaluation of other rows can reuse the space. Here we presented an alternative proof of the fact that Validity is in PSPACE. The interesting part is to conclude that if the logic includes $\mathbf{M}_{\rightarrow}$, then its Validity is PSPACE-complete.

4.1 Further investigation

We have to consider the known PSPACE-complete propositional logics. The best well-known are the Modal Logics K, $S4$, and KD.We know that (unlabeled) Natural Deduction systems for these logics are problematic. They fail to satisfy normalization and have rules with complex provisos. A consequence of this is that the natural reductions expected to work according the inversion principle produce derivations out of the system. Thus, the application of our technique to these logics depends on a concrete Natural Deduction system with rules that are instances of the general rules we deal with. Anyway, we already know, by other methods that these logics are PSPACE-complete, so this application would not be worth of presenting. However, there are labeled Natural Deduction systems with normalization theorem and a related sub-formula lemma.

For logics that we do not know the complexity class of the provability problem, the extension of the technique presented in this article to the a kind of general labeled introduction and elimination system of rules would be worth of developing. Our general form for introduction and elimination rules does not consider the many approaches of labeled Natural Deduction. It is our intention to discuss how Theorem 1 can be also obtained for the usual labeled systems used to specify modal logics, namely, [11], [16] and finally [2]. This would be quite useful in providing a proof-theoretical complexity analysis for the intuitionistic versions of Modal Logics.

Extending this technique to first-order logics would be interesting too. There are decidable fragments of pure predicate (without functions and the equality "=") that are beyond PSPACE, and surely the statement of the sub-formula principle for these cases are worth of studying. Finally, investigations concerning the feasibility of the interpolant (see [3]) and the existence of ND provers with sub-formula property are worth of studying too.

Acknowledgments: We thank many colleagues for suggesting improvements in the work reported in this article. Particularly, Luiz Carlos Pereira, Lew Gordeev, Mario Benevides, Wagner Sanz and Peter Schroeder-Heister, Ceclia Englander, Bruno Lopes and Jefferson de Barros Santos. We would like to mention Carlos Caleiro and João Marcos for pointing out that finitely many-valued logics could be a better example of positive application of our proof-theoretical technique than Modal Logics. We are very glad and thankful to have followed their suggestions.

References

[1] Arnon Avron & Ori Lahav (2010): *Strict Canonical Constructive Systems*. In Andreas Blass, Nachum Dershowitz & Wolfgang Reisig, editors: *Fields of Logic and Computation*, *Lecture Notes in Computer Science* 6300, Springer Berlin Heidelberg, pp. 75–94, doi:10.1007/978-3-642-15025-8_4.

[2] D. Basin, S. Matthews & L. Vigano (1997): *Labelled Propositional Modal Logics*. *Journal of Logic and Computation* 7(6), pp. 685–717, doi:10.1093/logcom/7.6.685.

[3] M.L. Bonet, T Pitassi & R. Raz (2000): *On Interpolation and Automatization for Frege Systems*. *SIAM Journal of Computing* 29(6), pp. 1939–1967, doi:10.1137/S0097539798353230.

[4] L. Borkowski, editor (1970): *Selected Works of Jan Łukasiewicz*. Studies in Logic, North-Holland, Amsterdam.

[5] Cecilia Englander, Edward Hermann Haeusler & Luiz Carlos Pereira (2014): *Finitely many-valued logics and natural deduction*. Logic Journal of the IGPL 22(2), pp. 333–354, doi:10.1093/jigpal/jzt032.

[6] M. J. Fischer & R. E. Ladner (1979): *Propositional dynamic logic of regular programs*. Journal of Computer and Systems Science 18(2), pp. 194–211, doi:10.1016/0022-0000(79)90046-1.

[7] J.Y. Halpern & Y. Moses (1990): *Knowledge and Common Knowledge in a Distributed Environment*. Journal of the ACM 37(3), pp. 549–587, doi:10.1145/79147.79161.

[8] F. Honsell & M. Miculan (1996): *A natural deduction approach to dynamic logic*. In S. Berardi & M. Coppo, editors: Types for Proofs and Programs, Lecture Notes in Computer Science 1158, Springer Berlin Heidelberg, pp. 1–16, doi:10.1007/3-540-61780-9_69.

[9] Grzegorz Malinowski (1993): *Many-Valued Logics*. Oxford University Press.

[10] J. Marcos & C. Caleiro (2009): *Classic-like analytic tableaux for finite-valued logics*. In H. Ono, R de Queiroz & M. Kanazawa, editors: Proceedings of WOLLIC 2009, Lecture Notes in Artificial Intelligence 5514, Springer Berlin Heidelberg, pp. 268–280, doi:10.1007/978-3-642-02261-6_22.

[11] S. Martini & A. Masini (1994): *A Computational Interpretation of Modal Proofs*. In: Proof Theory of Modal Logics, Kluwer, pp. 213–241, doi:10.1007/978-94-017-2798-3_12.

[12] D. Prawitz (1965): *Natural deduction: a proof-theoretical study*. Ph.D. thesis, Philosophy department, University of Stockholm.

[13] D. Prawitz (1979): *Proofs and the Meaning and Completeness of the Logical Constants*. In J. Hintikka, I. Niiniluotu & E. Saarinen, editors: Proceedings 4th Scandinavian Logic Symp, reidel, Dordrecht, pp. 25–40, doi:10.1007/978-94-009-9825-4_2.

[14] S. Read (2010): *General-Elimination Harmony and the Meaning of the Logical Constants*. Journal of Philosophical Logic 39(5), pp. 557–576, doi:10.1007/s10992-010-9133-7.

[15] Peter Schroeder-Heister (1984): *A Natural Extension of Natural Deduction*. Journal of Symbolic Logic 49(4), pp. 1284–1300, doi:10.2307/2274279.

[16] Alex K. Simpson (1994): *The Proof Theory and Semantics of Intuitionistic Modal Logic*. Ph.D. thesis, Computer Science.

[17] R. Statman (1979): *Intuitionistic propositional logic is polynomial-space complete*. Theoretical Computer Science 9(1), pp. 67 – 72, doi:10.1016/0304-3975(79)90006-9.