



Decidability, Introduction Rules and Automata

Gilles Dowek, Ying Jiang

► **To cite this version:**

Gilles Dowek, Ying Jiang. Decidability, Introduction Rules and Automata. International Conferences on Logic for Programming, Artificial Intelligence and Reasoning, Nov 2015, Bula, Fiji. 10.1007/978-3-662-48899-7_8 . hal-01252135

HAL Id: hal-01252135

<https://hal.inria.fr/hal-01252135>

Submitted on 7 Jan 2016

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

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

Decidability, Introduction Rules, and Automata

Gilles Dowek¹ and Ying Jiang²

¹ Inria, 23 avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France,
gilles.dowek@inria.fr.

² State Key Laboratory of Computer Science, Institute of Software, Chinese
Academy of Sciences, 100190 Beijing, China, jy@ios.ac.cn.

Abstract. We present a method to prove the decidability of provability in several well-known inference systems. This method generalizes both cut-elimination and the construction of an automaton recognizing the provable propositions.

1 Introduction

The goal of this paper is to connect two areas of logic: proof theory and automata theory, that deal with similar problems, using a different terminology.

To do so, we first propose to unify the terminology, by extending the notions of *introduction rule*, *automaton*, *cut*, and *cut-elimination* to arbitrary inference systems. An *introduction rule* is defined as any rule whose premises are smaller than its conclusion and an *automaton* as any inference system containing introduction rules only. Provability in an automaton is obviously decidable. A *cut* is defined as any proof ending with a non-introduction rule, whose major premises are proved with a proof ending with introduction rules. We show that a cut-free proof contains introduction rules only. A system is said to have the *cut-elimination property* if every proof can be transformed into a cut-free proof. Such a system is equivalent to an automaton.

Using this unified terminology, we then propose a general *saturation* method to prove the decidability of an inference system, by transforming it into a system that has the cut-elimination property, possibly adding extra rules. The outline of this method is the following. Consider a proof containing a non-introduction rule and focus on the sub-proof ending with this rule

$$\frac{\frac{\pi^1}{s^1} \dots \frac{\pi^n}{s^n}}{s} \text{ non-intro}$$

Assume it is possible to recursively eliminate the cuts in the proofs π^1, \dots, π^n , that is to transform them into proofs containing introduction rules only, hence ending with an introduction rule. We obtain a proof of the form

$$\frac{\frac{\frac{\rho_1^1}{s_1^1} \dots \frac{\rho_{m_1}^1}{s_{m_1}^1} \text{ intro} \quad \dots \quad \frac{\rho_1^n}{s_1^n} \dots \frac{\rho_{m_n}^n}{s_{m_n}^n} \text{ intro}}{s^1} \text{ non-intro}}{s} \text{ non-intro}$$

We may moreover tag each premise s^1, \dots, s^n of the non-introduction rule as *major* or *minor*. For instance, each elimination rule of Natural Deduction [15] has one major premise and the cut rule of Sequent Calculus [13] has two. If the major premises are s^1, \dots, s^k and *minor* ones s^{k+1}, \dots, s^n , the proof above can be decomposed as

$$\frac{\frac{\frac{\rho_1^1}{s_1^1} \dots \frac{\rho_{m_1}^1}{s_{m_1}^1} \text{ intro}}{s^1} \quad \dots \quad \frac{\frac{\frac{\rho_1^k}{s_1^k} \dots \frac{\rho_{m_k}^k}{s_{m_k}^k} \text{ intro}}{s^k} \quad \frac{\pi^{k+1}}{s^{k+1}} \quad \dots \quad \frac{\pi^n}{s^n} \text{ non-intro}}{s} \text{ non-intro}}$$

A proof of this form is called a *cut* and it must be reduced to another proof. The definition of the reduction is specific to each system under consideration. In several cases, however, such a cut is reduced to a proof built with the proofs $\rho_1^1, \dots, \rho_{m_1}^1, \dots, \rho_1^k, \dots, \rho_{m_k}^k, \pi^{k+1}, \dots, \pi^n$ and a derivable rule allowing to deduce the conclusion s from the premises $s_1^1, \dots, s_{m_1}^1, \dots, s_1^k, \dots, s_{m_k}^k, s^{k+1}, \dots, s^n$. Adding such derivable rules in order to eliminate cuts is called a *saturation* procedure.

Many cut-elimination proofs, typically the cut-elimination proofs for Sequent Calculus [9], do not proceed by eliminating cuts step by step, but by proving that a non-introduction rule is admissible in the system obtained by dropping this rule, that is, proving that if the premises s^1, \dots, s^n of this rule are provable in the restricted system, then so is its conclusion s . Proceeding by induction on the structure of proofs of s^1, \dots, s^n leads to consider cases where each major premise s^i has a proof ending with an introduction rule, that is also proofs of the form

$$\frac{\frac{\frac{\rho_1^1}{s_1^1} \dots \frac{\rho_{m_1}^1}{s_{m_1}^1} \text{ intro}}{s^1} \quad \dots \quad \frac{\frac{\frac{\rho_1^k}{s_1^k} \dots \frac{\rho_{m_k}^k}{s_{m_k}^k} \text{ intro}}{s^k} \quad \frac{\pi^{k+1}}{s^{k+1}} \quad \dots \quad \frac{\pi^n}{s^n} \text{ non-intro}}{s} \text{ non-intro}}$$

In some cases, the saturation method succeeds showing that every proof can be transformed into a proof formed with introduction rules only. Then, the inference system under consideration is equivalent, with respect to provability, to the automaton obtained by dropping all its non-introduction rules. This equivalence obviously ensures the decidability of provability in the inference system. In other cases, in particular when the inference system under consideration is undecidable, the saturation method succeeds only partially: typically some non-introduction rules can be eliminated but not all, or only a subsystem is proved to be equivalent to an automaton.

This saturation method is illustrated with examples coming from both proof theory and automata theory: Finite Domain Logic, Alternating Pushdown Systems, and three fragments of Constructive Predicate Logic, for which several formalizations are related: Natural Deduction, Gentzen style Sequent Calculus, Kleene style Sequent Calculus, and Vorob'ev-Hudelmaier-Dyckhoff-Negri style Sequent Calculus. The complexity of these provability problems, when they are decidable, is not discussed in this paper and is left for future work, for instance in the line of [1, 14].

In the remainder of this paper, the notions of introduction rule, automaton, and cut are defined in Section 2. Section 3 discusses the case of Finite State Automata. In Sections 4 and 5, examples of cut-elimination results are presented. In the examples of Section 4, the non-introduction rules can be completely eliminated transforming the inference systems under considerations into automata, while this elimination is only partially successful in the undecidable examples of Section 5. The proofs, and some developments, are omitted from this extended abstract. They can be found in the long version of the paper <https://who.rocq.inria.fr/Gilles.Dowek/Publi/introlong.pdf>.

2 Introduction rules, Automata, and Cuts

2.1 Introduction rules and Automata

Consider a set S , whose elements typically are propositions, sequents, etc. Let S^* be the set of finite lists of elements of S .

Definition 1 (Inference rule, Inference system, Proof). *An inference rule is a partial function from S^* to S . If R is an inference rule and $s = R(s_1, \dots, s_n)$, we say that the conclusion s is proved from the premises s_1, \dots, s_n with the rule R and we write*

$$\frac{s_1 \dots s_n}{s} R$$

Some rules are equipped with an extra piece of information, tagging each premise s_1, \dots, s_n as major or minor. An inference system is a set of inference rules. A proof in an inference system is a finite tree labeled by elements of S such that for each node labeled with s and whose children are labeled with s_1, \dots, s_n , there exists an inference rule R of the system such that

$$\frac{s_1 \dots s_n}{s} R$$

A proof is a proof of s if its root is labeled by s . An element of S is said to be provable, if it has a proof.

Definition 2 (Introduction rule, Pseudo-automaton). *Consider a set S and a well-founded order \prec on S . A rule R is said to be an introduction rule with respect to this order, if whenever*

$$\frac{s_1 \dots s_n}{s} R$$

we have $s_1 \prec s, \dots, s_n \prec s$. A pseudo-automaton is an inference system containing introduction rules only.

Except in the system \mathcal{D} (Section 5.4), this order \prec is always that induced by the size of the propositions and sequents. It is left implicit.

Definition 3 (Finitely branching system, Automaton). *An inference system is said to be finitely branching, if for each conclusion s , there is only a finite number of lists of premises $\bar{s}_1, \dots, \bar{s}_p$ such that s can be proved from \bar{s}_i with a rule of the system. An automaton is a finitely branching pseudo-automaton.*

2.2 Cuts

We define a general notion of cut, that applies to all inference systems considered in this paper. More specific notions of cut will be introduced later for some systems, and the general notion of cut defined here will be emphasized as *general cut* to avoid ambiguity.

Definition 4 (Cut). A (general) cut is a proof of the form

$$\frac{\frac{\frac{\rho_1^1}{s_1^1} \dots \frac{\rho_{m_1}^1}{s_{m_1}^1} \text{ intro} \quad \dots \quad \frac{\frac{\rho_1^k}{s_1^k} \dots \frac{\rho_{m_k}^k}{s_{m_k}^k} \text{ intro} \quad \frac{\pi_{k+1}}{s^{k+1}} \dots \frac{\pi_n}{s^n} \text{ non-intro}}{s^k} \text{ intro}}{s} \text{ non-intro}}$$

where s^1, \dots, s^k are the major premises of the non-introduction rule. A proof contains a cut if one of its sub-proofs is a cut. A proof is cut-free if it contains no cut. An inference system has the cut-elimination property if every element that has a proof also has a cut-free proof.

Lemma 1 (Key lemma). A proof is cut-free if and only if it contains introduction rules only.

Proof. If a proof contains introduction rules only, it is obviously cut-free. We prove the converse by induction over proof structure. Consider a cut-free proof. Let R be the last rule of this proof and π_1, \dots, π_n be the proofs of the premises of this rule. The proof has the form

$$\frac{\frac{\pi_1}{s_1} \dots \frac{\pi_n}{s_n}}{s} R$$

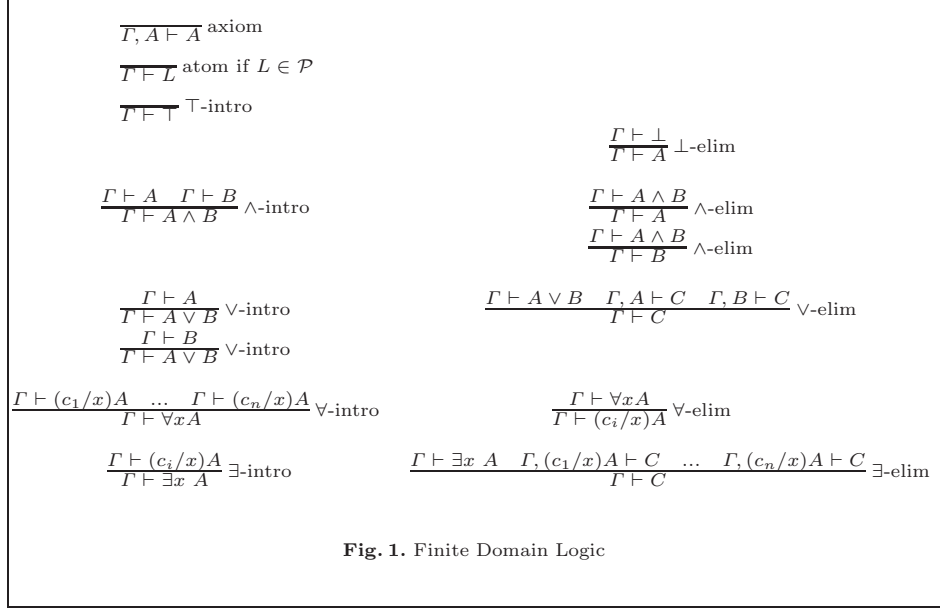
By induction hypothesis, the proofs π_1, \dots, π_n contain introduction rules only. As the proof is cut-free, the rule R must be an introduction rule.

Consider a finitely-branching inference system \mathcal{I} and the automaton \mathcal{A} formed with the introduction rules of \mathcal{I} . If \mathcal{I} has the cut-elimination property, then every element that has a proof in \mathcal{I} has a cut-free proof, that is a proof formed with introduction rules of \mathcal{I} only, that is a proof in \mathcal{A} . Thus, \mathcal{I} and \mathcal{A} are equivalent with respect to provability. Since \mathcal{A} is decidable, so is \mathcal{I} .

3 Finite State Automata

In this section, we show that the usual notion of finite state automaton is a particular case of the notion of automaton introduced in Definition 3.

Consider a finite state automaton \mathcal{A} . We define a language \mathcal{L} in predicate logic containing a constant ε ; for each symbol γ of the alphabet of \mathcal{A} , a unary function symbol, also written γ ; and for each state P of \mathcal{A} a unary predicate symbol, also written P . A closed term in \mathcal{L} has the form $\gamma_1(\gamma_2(\dots(\gamma_n(\varepsilon))))$, where $\gamma_1, \dots, \gamma_n$ are



function symbols. Such a term is called a *word*, written $w = \gamma_1\gamma_2\dots\gamma_n$. A closed atomic proposition has the form $P(w)$, where P is a state and w a word. We build an inference system that consists of, for each transition rule $P \xrightarrow{\gamma} Q$ of \mathcal{A} , the introduction rule

$$\frac{Q(x)}{P(\gamma(x))}$$

and, for each final state F of \mathcal{A} , the introduction rule

$$\overline{F(\varepsilon)}$$

It is routine to check that a word w is recognized by the automaton \mathcal{A} in a state I if and only if the proposition $I(w)$ has a proof in the corresponding system.

4 From cut-elimination to automata

In this section, we present two cut-elimination theorems, that permit to completely eliminate the non-introduction rules and prove, this way, the decidability of Finite Domain Logic and of Alternating Pushdown Systems, respectively.

4.1 Finite Domain Logic

We begin with a toy example, *Finite Domain Logic*, motivated by its simplicity: we can prove a cut-elimination theorem, showing the system is equivalent to the automaton obtained by dropping its non-introduction rules.

$\frac{P_1(x) \dots P_n(x)}{Q(\gamma(x))} \text{ intro } n \geq 0$	$\frac{P_1(\gamma(x)) P_2(x) \dots P_n(x)}{Q(x)} \text{ elim } n \geq 1$
$\overline{Q(\varepsilon)} \text{ intro}$	$\frac{P_1(x) \dots P_n(x)}{Q(x)} \text{ neutral } n \geq 0$

Fig. 2. Alternating Pushdown Systems

Finite Domain Logic is a version of Natural Deduction tailored to prove the propositions that are valid in a given finite model \mathcal{M} . The differences with the usual Natural Deduction are the following: a proposition of the form $A \Rightarrow B$ is just an abbreviation for $\neg A \vee B$ and negation has been pushed to atomic propositions using de Morgan's laws; the \forall -intro and the \exists -elim rules are replaced by enumeration rules, and an *atom* rule is added to prove closed atomic propositions and their negations valid in the underlying model.

If the model \mathcal{M} is formed with a domain $\{a_1, \dots, a_n\}$ and relations R_1, \dots, R_m over this domain, we consider the language containing constants c_1, \dots, c_n for the elements a_1, \dots, a_n and predicate symbols P_1, \dots, P_m for the relations R_1, \dots, R_m . The *Finite Domain Logic* of the model \mathcal{M} is defined by the inference system of Figure 1, where the set \mathcal{P} contains, for each atomic proposition $P_i(c_{j_1}, \dots, c_{j_k})$, either the proposition $P_i(c_{j_1}, \dots, c_{j_k})$ if $\langle a_{j_1}, \dots, a_{j_k} \rangle$ is in R_i , or the proposition $\neg P_i(c_{j_1}, \dots, c_{j_k})$, otherwise.

In this system, the introduction rules are those presented in the first column: the axiom rule, the atom rule, and the rules \top -intro, \wedge -intro, \vee -intro, \forall -intro, and \exists -intro. The non-introduction rules are those presented in the second column. Each rule has one major premise: the leftmost one. A cut is as in Definition 4.

Theorem 1 (Soundness, Completeness, and Cut-elimination). *Let B be a closed proposition, the following are equivalent: (1.) the proposition B has a proof, (2.) the proposition B is valid in \mathcal{M} , (3.) the proposition B has a cut-free proof, that is a proof formed with introduction rules only.*

Therefore, provability in Finite Domain Logic is decidable, as the provable propositions are recognized by the automaton obtained by dropping the non-introduction rules. Since the introduction rules preserve context emptiness, the contexts can be ignored and the axiom rule can be dropped. This automaton could also be expressed in a more familiar way with the transition rules

$$\begin{array}{ll}
L \leftrightarrow \emptyset \text{ if } L \in \mathcal{P} & A \vee B \leftrightarrow \{A\} \\
\top \leftrightarrow \emptyset & A \vee B \leftrightarrow \{B\} \\
A \wedge B \leftrightarrow \{A, B\} & \forall x A \leftrightarrow \{(c_1/x)A, \dots, (c_n/x)A\} \\
& \exists x A \leftrightarrow \{(c_i/x)A\} \text{ for each } c_i
\end{array}$$

4.2 Alternating Pushdown Systems

The second example, *Alternating Pushdown Systems*, is still decidable [2], but a little bit more complex. Indeed these systems, in general, need to be saturated—that is extended with derivable rules—in order to enjoy cut-elimination.

Consider a language \mathcal{L} containing a finite number of unary predicate symbols, a finite number of unary function symbols, and a constant ε . An *Alternating Pushdown System* is an inference system whose rules are like those presented in Figure 2. The rules in the first column are introduction rules and those in the second column, the elimination and neutral rules, are not. Elimination rules have one major premise, the leftmost one, and all the premises of a neutral rule are major. A cut is as in Definition 4.

Not all Alternating Pushdown Systems enjoy the cut-elimination property. However, every Alternating Pushdown System has an extension with derivable rules that enjoys this property: each time we have a cut of the form

$$\frac{\frac{\frac{\rho_1^1}{s_1^1} \dots \frac{\rho_{m_1}^1}{s_{m_1}^1} \text{ intro}}{s^1} \dots \frac{\frac{\rho_1^k}{s_1^k} \dots \frac{\rho_{m_k}^k}{s_{m_k}^k} \text{ intro}}{s^k} \dots \frac{\frac{\pi_{k+1}}{s^{k+1}} \dots \frac{\pi_n}{s^n} \text{ non-intro}}{s}}{s} \text{ cut}$$

we add a derivable rule allowing to deduce directly s from $s_1^1, \dots, s_{m_1}^1, \dots, s_1^k, \dots, s_{m_k}^k, s^{k+1}, \dots, s^n$. This leads to the following saturation algorithm [4, 10, 11].

Definition 5 (Saturation). *Given an Alternating Pushdown System,*

- if it contains an introduction rule

$$\frac{P_1(x) \dots P_m(x)}{Q_1(\gamma(x))} \text{ intro}$$

and an elimination rule

$$\frac{Q_1(\gamma(x)) \ Q_2(x) \ \dots \ Q_n(x)}{R(x)} \text{ elim}$$

then we add the neutral rule

$$\frac{P_1(x) \dots P_m(x) \ Q_2(x) \ \dots \ Q_n(x)}{R(x)} \text{ neutral}$$

- if it contains introduction rules

$$\frac{P_1^1(x) \dots P_{m_1}^1(x)}{Q_1(\gamma(x))} \text{ intro} \quad \dots \quad \frac{P_1^n(x) \dots P_{m_n}^n(x)}{Q_n(\gamma(x))} \text{ intro}$$

and a neutral rule

$$\frac{Q_1(x) \dots Q_n(x)}{R(x)} \text{ neutral}$$

then we add the introduction rule

$$\frac{P_1^1(x) \dots P_{m_1}^1(x) \ \dots \ P_1^n(x) \ \dots \ P_{m_n}^n(x)}{R(\gamma(x))} \text{ intro}$$

- if it contains introduction rules

$$\overline{Q_1(\varepsilon)} \text{ intro} \quad \dots \quad \overline{Q_n(\varepsilon)} \text{ intro}$$

and a neutral rule

$$\frac{Q_1(x) \dots Q_n(x)}{R(x)} \text{neutral}$$

then we add the introduction rule

$$\overline{R(\varepsilon)} \text{intro}$$

As there is only a finite number of possible rules, this procedure terminates.

It is then routine to check that if a closed proposition has a proof in a saturated system, it has a cut-free proof [4], leading to the following result.

Theorem 2 (Decidability). *Provability of a closed proposition in an Alternating Pushdown System is decidable.*

Example 1. Consider the Alternating Pushdown System S

$$\begin{array}{cccc} \frac{Q(x)}{P(ax)} \mathbf{i1} & \frac{T(x)}{P(bx)} \mathbf{i2} & \frac{T(x)}{R(ax)} \mathbf{i3} & \overline{R(bx)} \mathbf{i4} \\ \frac{P(x) R(x)}{Q(x)} \mathbf{n1} & \overline{T(x)} \mathbf{n2} & \frac{P(ax)}{S(x)} \mathbf{e1} & \end{array}$$

The system S' obtained by saturating the system S contains the rules of the system S and the following rules

$$\begin{array}{cccc} \frac{Q(x)}{S(x)} \mathbf{n3} & \overline{T(\varepsilon)} \mathbf{i5} & \overline{T(ax)} \mathbf{i6} & \frac{Q(x) T(x)}{Q(ax)} \mathbf{i7} \\ \frac{Q(x) T(x)}{S(ax)} \mathbf{i8} & \overline{T(bx)} \mathbf{i9} & \frac{T(x)}{Q(bx)} \mathbf{i10} & \frac{T(x)}{S(bx)} \mathbf{i11} \end{array}$$

The automaton S'' contain the rules **i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11.**

The proof in the system S

$$\frac{\frac{\overline{T(\varepsilon)} \mathbf{n2}}{P(b)} \mathbf{i2} \quad \overline{R(b)} \mathbf{i4}}{\frac{Q(b)}{P(ab)} \mathbf{i1}} \mathbf{n1} \quad \frac{\overline{T(b)} \mathbf{n2}}{R(ab)} \mathbf{i3}}{\frac{Q(ab)}{P(aab)} \mathbf{i1}} \mathbf{n1} \quad \frac{Q(ab)}{S(ab)} \mathbf{e1}}$$

reduces to the cut-free proof in the system S''

$$\frac{\overline{T(\varepsilon)} \mathbf{i5}}{\frac{Q(b)}{S(ab)} \mathbf{i8}} \mathbf{i10} \quad \overline{T(b)} \mathbf{i9}$$

$\overline{\Gamma, A \vdash A}$ axiom	
$\frac{}{\Gamma \vdash \top}$ \top -intro	$\frac{\Gamma \vdash \perp}{\Gamma \vdash A}$ \perp -elim
$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$ \wedge -intro	$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}$ \wedge -elim
$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}$ \vee -intro	$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$ \wedge -elim
$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$ \vee -intro	$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$ \vee -elim
$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$ \Rightarrow -intro	$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$ \Rightarrow -elim
$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x A}$ \forall -intro if x not free in Γ	$\frac{\Gamma \vdash \forall x A}{\Gamma \vdash (t/x)A}$ \forall -elim
$\frac{\Gamma \vdash (t/x)A}{\Gamma \vdash \exists x A}$ \exists -intro	$\frac{\Gamma \vdash \exists x A \quad \Gamma, A \vdash B}{\Gamma \vdash B}$ \exists -elim if x not free in Γ, B

Fig. 3. Constructive Natural Deduction

5 Partial results for undecidable systems

In this section, we focus on Constructive Predicate Logic, leaving the case of Classical Predicate Logic for future work. We start with Natural Deduction [15]. As provability in Predicate Logic is undecidable, we cannot expect to transform Natural Deduction into an automaton. But, as we shall see, saturation permits to transform first Natural Deduction into a Gentzen style Sequent Calculus [13], then the latter into a Kleene style Sequent Calculus [13], and then the latter into a Vorob'ev-Hudelmaier-Dyckhoff-Negri style Sequent Calculus [16, 12, 5, 7]. Each time, a larger fragment of Constructive Predicate Logic is proved decidable.

Note that each transformation proceeds in the same way: first, we identify some general cuts. Then, like in the saturation procedure of Section 4.2, we add some admissible rules to eliminate these cuts. Finally, we prove a cut-elimination theorem showing that some non-introduction rules can be dropped.

5.1 Natural Deduction

In Natural Deduction (Figure 3), the introduction rules are those presented in the first column, they are the axiom rule and the rules \top -intro, \wedge -intro, \vee -intro, \Rightarrow -intro, \forall -intro, and \exists -intro. The non-introduction rules are those presented in the second column, each of them has one major premise: the leftmost one.

Natural Deduction has a specific notion of cut: a proof ending with a \wedge -elim, \vee -elim, \Rightarrow -elim, \forall -elim, \exists -elim rule, whose major premise is proved with a proof

ending with a \wedge -intro, \vee -intro, \Rightarrow -intro, \forall -intro, \exists -intro rule, respectively. The only difference between this specific notion of cut and the general one (Definition 4) is that the general notion has one more form of cut: a proof built with an elimination rule whose major premise is proved with the axiom rule. For instance

$$\frac{\overline{P \wedge Q \vdash P \wedge Q} \text{ axiom}}{P \wedge Q \vdash P} \wedge\text{-elim}$$

So proofs free of specific cuts can still contain general cuts of this form.

Saturating the system, like in Section 4.2, to eliminate the specific cuts, would add derivable rules such as

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A} R_{\wedge}$$

But they are not needed, as they are admissible in cut-free Natural Deduction.

The admissibility of some rules however are based on a substitution of proofs, that may create new cuts on smaller propositions, that need in turn to be eliminated. In other words, the termination of the specific cut-elimination algorithm needs to be proved [15].

As general cuts with an axiom rule are not eliminated, this partial cut-elimination theorem is not sufficient to eliminate all elimination rules and to prove the decidability of Constructive Natural Deduction, but it yields a weaker result: a (specific-)cut-free proof ends with introduction rules, as long as the context of the proved sequent contains atomic propositions only. To formalize this result, we introduce a modality $[]$ and define a translation that freezes the non atomic left-hand parts of implications, $f(A \Rightarrow B) = [A] \Rightarrow f(B)$, if A is not atomic, and $f(A \Rightarrow B) = A \Rightarrow f(B)$, if A is atomic, $f(A \wedge B) = f(A) \wedge f(B)$, etc., and the converse function u is defined in a trivial way.

Definition 6. *Let \mathcal{A} be the pseudo-automaton formed with the introduction rules of Constructive Natural Deduction, including the axiom rule, plus the introduction rule*

$$\overline{\Gamma, [A] \vdash B} \text{ delay}$$

Theorem 3. *Let $\Gamma \vdash A$ be a sequent such that Γ contains atomic propositions only. If $\Gamma \vdash A$ has a (specific-)cut-free proof in Constructive Natural Deduction, then $\Gamma \vdash f(A)$ has a proof in the pseudo-automaton \mathcal{A} and for each leaf $\Delta \vdash B$ proved with the delay rule, the sequent $u(\Delta \vdash B)$ has a proof in Constructive Natural Deduction.*

A first corollary of Theorem 3 is the decidability of the small fragment

$$A = P \mid \top \mid \perp \mid A \wedge A \mid A \vee A \mid P \Rightarrow A \mid \forall x A \mid \exists x A$$

where the left-hand side of an implication is always atomic, that is no connective or quantifier has a negative occurrence. As the pseudo-automaton obtained this way is not finitely branching, we need, as well-known, to introduce meta-variables to prove this decidability result.

A second corollary is that if A is a proposition starting with n connectors or quantifiers different from \Rightarrow , then a (specific-)cut-free proof of the sequent $\vdash A$ ends with $n+1$ successive introduction rules. For $n = 0$, we obtain the well-known last rule property of constructive (specific-)cut-free proofs. For a proposition A of the form $\forall x (B_1 \vee B_2)$, for instance, we obtain that a (specific-)cut-free proof of $\vdash \forall x (B_1 \vee B_2)$ ends with three introduction rules. Thus, it has the form

$$\frac{\frac{\frac{\pi'}{\vdash B_i}}{\vdash B_1 \vee B_2} \vee\text{-intro}}{\vdash \forall x (B_1 \vee B_2)} \forall\text{-intro}$$

and π' itself ends with an introduction rule. As a consequence, if the proposition $\forall x (B_1 \vee B_2)$ has a proof, then either the proposition B_1 or the proposition B_2 has a proof, thus the proposition $(\forall x B_1) \vee (\forall x B_2)$ has a proof. This commutation of the universal quantifier with the disjunction is called a *shocking equality* [8].

5.2 Eliminating elimination rules: Gentzen style Sequent Calculus

To eliminate the general cuts of the form

$$\frac{\frac{A \wedge B \vdash A \wedge B}{A \wedge B \vdash A} \wedge\text{-elim}}{A \wedge B \vdash A} \text{axiom}$$

we could add an introduction rule of the form

$$\frac{}{A \wedge B \vdash A} I$$

But, this saturation procedure would not terminate.

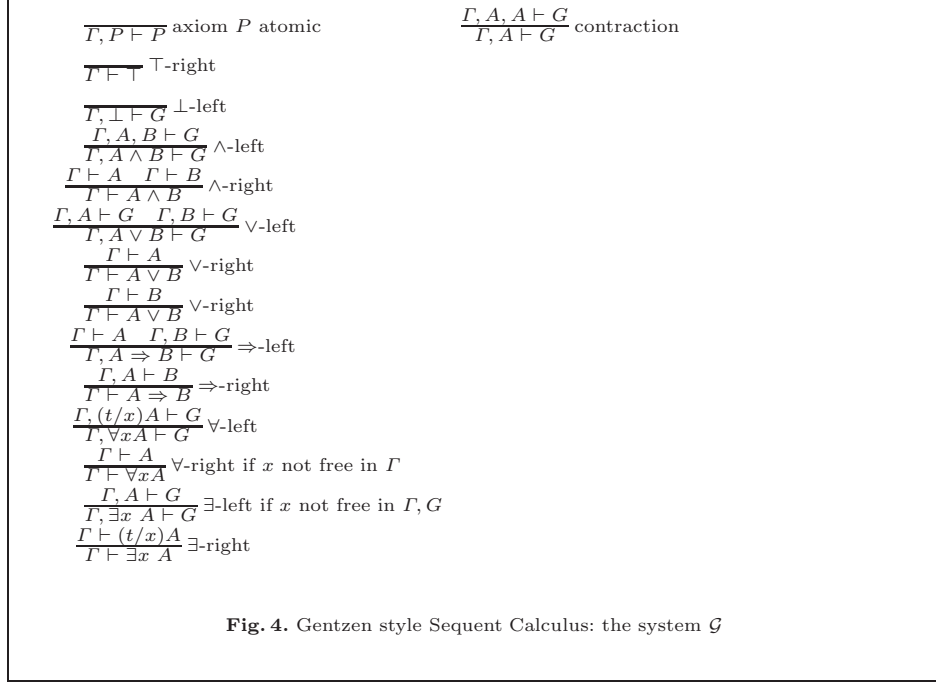
A way to keep the number of rules finite is to add left introduction rules to decompose the complex hypotheses, before they are used by the axiom rule: the left rules of Sequent Calculus. However, this is still not sufficient to eliminate the elimination rules of Constructive Natural Deduction. For instance, the sequent $\forall x (P(x) \wedge (P(f(x)) \Rightarrow Q)) \vdash Q$ has a proof using elimination rules

$$\frac{\frac{\frac{\frac{\Gamma \vdash \forall x (P(x) \wedge (P(f(x)) \Rightarrow Q))}{\Gamma \vdash P(c) \wedge (P(f(c)) \Rightarrow Q)} \forall\text{-elim}}{\Gamma \vdash P(f(c)) \Rightarrow Q} \wedge\text{-elim}}{\Gamma \vdash Q} \Rightarrow\text{-elim}}{\frac{\frac{\frac{\Gamma \vdash \forall x (P(x) \wedge (P(f(x)) \Rightarrow Q))}{\Gamma \vdash P(f(c)) \wedge (P(f(f(c))) \Rightarrow Q)} \forall\text{-elim}}{\Gamma \vdash P(f(c))} \wedge\text{-elim}}{\Gamma \vdash Q} \Rightarrow\text{-elim}}$$

where $\Gamma = \forall x (P(x) \wedge (P(f(x)) \Rightarrow Q))$, but none using introduction rules only.

So, we need to add a contraction rule, to use an hypothesis several times

$$\frac{\Gamma, A, A \vdash G}{\Gamma, A \vdash G} \text{contraction}$$



To prove that the elimination rules of Natural Deduction can now be eliminated, we prove, using Gentzen's theorem [9], that they are admissible in the system \mathcal{G} (Figure 4), the Gentzen style Sequent Calculus, obtained by dropping the elimination rules of Constructive Natural Deduction. In this system, all the rules are introduction rules, except the contraction rule. The system \mathcal{G} does not allow to prove the decidability of any larger fragment of Constructive Predicate Logic, but it is the basis of the two systems presented in the Sections 5.3 and 5.4.

5.3 Eliminating the contraction rule: Kleene style Sequent Calculus

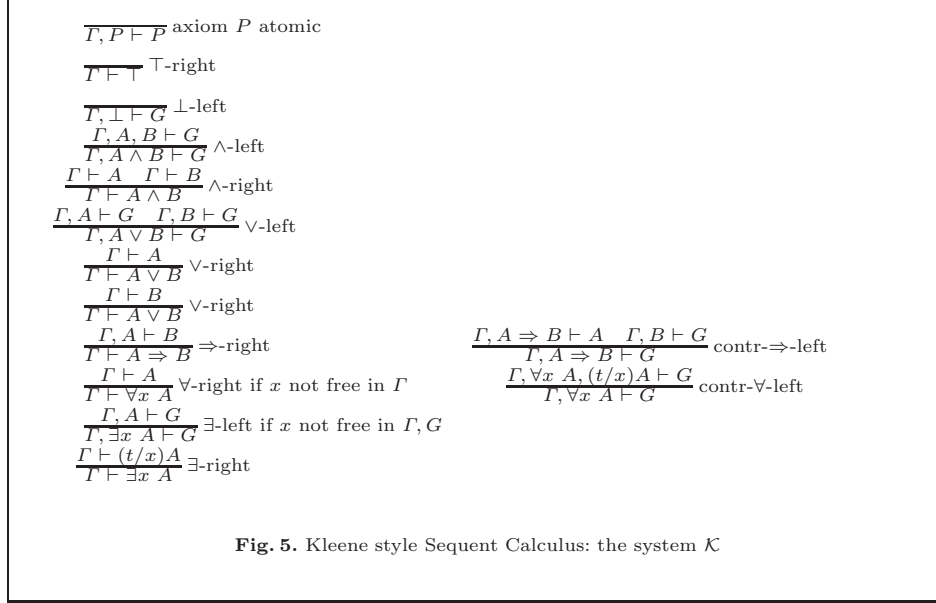
In the system \mathcal{G} , the proof

$$\frac{\frac{\rho}{\Gamma, \forall x A, (t/x)A \vdash B}}{\Gamma, \forall x A, \forall x A \vdash B} \forall\text{-left}}{\Gamma, \forall x A \vdash B} \text{contraction}$$

is a general cut and we may replace it by the application of the derivable rule

$$\frac{\rho}{\Gamma, \forall x A, (t/x)A \vdash B} \text{contr-}\forall\text{-left}}{\Gamma, \forall x A \vdash B}$$

which is a rule *à la* Kleene. The other general cuts yields similar derivable rules. But, as noticed by Kleene, the derivable rules for the contradiction, the conjunction, the disjunction and the existential quantifier can be dropped, while that



for the implication can be simplified to

$$\frac{\Gamma, A \Rightarrow B \vdash A \quad \Gamma, B \vdash G}{\Gamma, A \Rightarrow B \vdash G} \text{contr-}\Rightarrow\text{-left}$$

The rules \Rightarrow -left and \forall -left of the system \mathcal{G} , that are subsumed by the rules $\text{contr-}\Rightarrow$ -left and $\text{contr-}\forall$ -left, can be also dropped. There are also other general cuts, where the last rule is a contraction and the rule above is an introduction applied to another proposition, but these cuts can be eliminated without introducing any extra rule. In other words, after applying the contraction rule, we can focus on the contracted proposition [6].

We get this way the system \mathcal{K} (Figure 5). In this system, all rules are introduction rules, except the rules $\text{contr-}\Rightarrow$ -left and $\text{contr-}\forall$ -left. The system \mathcal{K} plus the contraction rule is obviously sound and complete with respect to the system \mathcal{G} . To prove that the contraction rule can be eliminated from it, and hence the system \mathcal{K} also is sound and complete with respect to the system \mathcal{G} , we prove the admissibility of the contraction rule in the system \mathcal{K} —see the long version of the paper for the full proof. The system \mathcal{K} gives the decidability of a larger fragment of Constructive Predicate Logic, where the implication and the universal quantifier have no negative occurrences.

5.4 Eliminating the $\text{contr-}\Rightarrow$ -left rule: Vorob'ev-Hudelmaier-Dyckhoff-Negri style Sequent Calculus

In order to eliminate the $\text{contr-}\Rightarrow$ -left rule, we consider the general cuts where a sequent $\Gamma, A \Rightarrow B \vdash G$ is proved with a $\text{contr-}\Rightarrow$ -left rule whose major premise

$$\begin{array}{c}
\frac{}{\Gamma, P \vdash P} \text{ axiom } P \text{ atomic} \\
\frac{}{\Gamma \vdash \top} \top\text{-right} \\
\frac{}{\Gamma, \perp \vdash G} \perp\text{-left} \\
\frac{\Gamma, A, B \vdash G}{\Gamma, A \wedge B \vdash G} \wedge\text{-left} \\
\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge\text{-right} \\
\frac{\Gamma, A \vdash G \quad \Gamma, B \vdash G}{\Gamma, A \vee B \vdash G} \vee\text{-left} \\
\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee\text{-right} \\
\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \vee\text{-right} \\
\frac{\Gamma, P, B \vdash G}{\Gamma, P, P \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\text{axiom}} \\
\frac{\Gamma, B \vdash G}{\Gamma, \top \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\top} \\
\frac{\Gamma, C \Rightarrow B \vdash C \quad \Gamma, D \Rightarrow B \vdash D \quad \Gamma, B \vdash G}{\Gamma, (C \wedge D) \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\wedge} \\
\frac{\Gamma, C \Rightarrow B, D \Rightarrow B \vdash C \quad \Gamma, B \vdash G}{\Gamma, (C \vee D) \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\vee} \\
\frac{\Gamma, C \Rightarrow B, D \Rightarrow B \vdash D \quad \Gamma, B \vdash G}{\Gamma, (C \vee D) \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\vee} \\
\frac{\Gamma, D \Rightarrow B, C \vdash D \quad \Gamma, B \vdash G}{\Gamma, (C \Rightarrow D) \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\Rightarrow} \\
\frac{\Gamma, (\forall x C) \Rightarrow B \vdash C \quad \Gamma, B \vdash G}{\Gamma, (\forall x C) \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\forall} \quad x \text{ not free in } \Gamma, B \\
\frac{\Gamma, (\exists x C) \Rightarrow B \vdash (t/x)C \quad \Gamma, B \vdash G}{\Gamma, (\exists x C) \Rightarrow B \vdash G} \Rightarrow\text{-left}_{\exists} \\
\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \Rightarrow\text{-right} \\
\frac{\Gamma \vdash A \quad \forall\text{-right}}{\Gamma \vdash \forall x A \quad x \text{ not free in } \Gamma} \forall\text{-right} \\
\frac{\Gamma, A \vdash G}{\Gamma, \exists x A \vdash G \quad x \text{ not free in } \Gamma, G} \exists\text{-left} \\
\frac{\Gamma \vdash (t/x)A}{\Gamma \vdash \exists x A} \exists\text{-right} \\
\frac{\Gamma, \forall x A, (t/x)A \vdash G}{\Gamma, \forall x A \vdash G} \text{contr-}\forall\text{-left}
\end{array}$$

Fig. 6. The system \mathcal{D}

$\Gamma, A \Rightarrow B \vdash A$ is proved with an introduction rule, applied to the proposition A . This leads to consider the various cases for A , that is hypotheses of the form $P \Rightarrow B$, $\top \Rightarrow B$, $(C \wedge D) \Rightarrow B$, $(C \vee D) \Rightarrow B$, $(C \Rightarrow D) \Rightarrow B$, $(\forall x C) \Rightarrow B$, and $(\exists x C) \Rightarrow B$. The case $A = P$, atomic, needs to be considered because the premise $\Gamma, A \Rightarrow B \vdash A$ may be proved with the axiom rule, but the case $\perp \Rightarrow B$ does not, because there is no right rule for the symbol \perp . This enumeration of the various shapes of A is the base of the sequent calculi in the style of Vorob'ev, Hudelmaier, Dyckhoff, and Negri [16, 12, 5, 7].

We obtain this way several types of general cuts that can be eliminated by introducing derivable rules. These rules can be simplified leading to the system \mathcal{D} (Figure 6). The system \mathcal{D} plus the $\text{contr-}\Rightarrow\text{-left}$ rule is obviously sound and complete with respect to the system \mathcal{K} . To prove that the $\text{contr-}\Rightarrow\text{-left}$ rule can be eliminated, and hence the system \mathcal{D} also is sound and complete with respect to

the system \mathcal{K} , we use a method similar to that of [7], and prove the admissibility of the $\text{contr}\text{-}\Rightarrow\text{-left}$ rule—see the long version of the paper for the full proof.

This system \mathcal{D} gives the decidability of a larger fragment of Constructive Predicate Logic containing all connectives, shallow universal and existential quantifiers—that is quantifiers that occur under no implication at all—and negative existential quantifiers. This fragment contains the prenex fragment of Constructive Predicate Logic, that itself contains Constructive Propositional Logic.

Acknowledgements

This work is supported by the ANR-NSFC project LOCALI (NSFC 61161130530 and ANR 11 IS02 002 01) and the Chinese National Basic Research Program (973) Grant No. 2014CB340302.

References

1. D. Basin and H. Ganzinger, Automated Complexity Analysis Based on Ordered Resolution, *Journal of the ACM*, 48, 1, 2001, 70-109.
2. A. Bouajjani, J. Esparza, and O. Maler, Reachability analysis of pushdown automata: Application to model-checking A. W. Mazurkiewicz, J. Winkowski (Eds.) *Concurrency theory*, Lecture Notes in Computer Science, 1243, 1997, 135-150.
3. N. Dershowitz and Z. Manna, Proving termination with multiset orderings, *Communications of the ACM* 22(8), 1979, 465-476.
4. G. Dowek and Y. Jiang, Cut-elimination and the decidability of reachability in alternating pushdown systems, arXiv:1410.8470 [cs.LO], 2014.
5. R. Dyckhoff, Contraction-Free Sequent Calculi for Intuitionistic Logic, *The Journal of Symbolic Logic*, 57, 3, 1992, 795-807.
6. R. Dyckhoff and S. Lengrand, LJQ: A Strongly Focused Calculus for Intuitionistic Logic, (A. Beckmann et al. eds.) *Computability in Europe*, Lecture Notes in Computer Science 3988, Springer, 2006, 173-185.
7. R. Dyckhoff and S. Negri, Admissibility of structural rules for contraction-free systems of intuitionistic logic *The Journal of Symbolic Logic*, 65, 2000, 1499-1518.
8. J.-Y. Girard, Locus Solum, *Mathematical Structures in Computer Science*, 11, 2001, 301-506.
9. J.-Y. Girard, Y. Lafont, and P. Taylor, *Proofs and Types*, Cambridge University Press, 1989.
10. T. Frühwirth, E. Shapiro, M. Vardi E. Yardeni, Logic Programs as Types for Logic Programs, *Logic in Computer Science*, 1991, 300-309.
11. J. Goubault-Larrecq, Deciding H_1 by Resolution, *Information Processing Letters*, 95(3), 2005, 401-408.
12. J. Hudelmaier, An $O(n \log n)$ -space decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3, 1993, 63-76.
13. S.C. Kleene, *Introduction to Metamathematics*, North-Holland, 1952.
14. D.A. McAllester, Automatic Recognition of Tractability in Inference Relations, *Journal of the ACM*, 40, 2, 1993, 284-303.
15. D. Prawitz, *Natural Deduction*, Almqvist & Wiksell, 1965.
16. N.N. Vorob'ev, A new algorithm for derivability in the constructive propositional calculus, *American Mathematical Society Translations*, 2, 94, 1970, 37-71.