Journal of Logic, Language and Information (2006) 15: 195–218
DOI: 10.1007/s10849-005-9001-y

Cut and Pay

MARCELO FINGER*
University of Sao Paulo, Brazil
E-mail: mfinger@ime.usp.br

DOV GABBAY
King's College, London
E-mail: dg@dcs.kcl.ac.uk

(Received 14 June 2005; in final form 27 September 2005)

Abstract. In this paper we study families of resource aware logics that explore resource restriction on rules; in particular, we study the use of controlled cut-rule and introduce three families of parameterised logics that arise from different ways of controlling the use of cut. We start with a formulation of classical logic in which cut is non-eliminable and then impose restrictions on the use of cut. Three Cut-and-Pay families of logics are presented, and it is shown that each family provides an approximation process for full propositional classical logic when the control over the use of cut is progressively weakened. A sound and complete semantics is given for each component of each of the three families of approximated logics. One of these families is shown to possess the uniform substitution property, a new result for approximated reasoning. A tableau based decision procedure is presented for each element of the approximation families and the complexity of each decision procedure is studied. We show that there are families in which every component logic can be decided polynomially.

1. Introduction

Resource aware logics allow for a form of reasoning in which one can keep track of and control the logical elements used in an inference. Many such logic systems have been analysed in the literature from the philosophical, mathematical and computational points of view. For instance, Intuitionistic Logic (Van Dalen, 1984) forbids the use of the *reductio ad absurdum* inference; Relevance Logic (Anderson and Belnap, 1975) further imposes that all premises in an inference must be used; and Linear Logic (Girard, 1987) constrains even more inferences, by imposing that every premise must be used exactly once. These logics are all members of the class of resource aware systems known as Substructural Logics (Restall, 2000). In another direction, da Costa's Paraconsistent Logics (da Costa, 1974) restrict the use of the *ex contradictio quodlibet* inference; in this case, a family of logics C_i, $1 \leq i \leq \omega$,

*Partly supported by CNPq grant PQ 300597/95-5 and FAPESP project 03/00312-0.

is generated by imposing a crescendo of restrictions on when the trivialisation of an inference follows from the presence of a contradiction.

Resource logics have restricted the number of uses of items of data. However, those logic systems do not impose resource restrictions on the number of uses of inference rules. However, it may well be possible to explore resource restriction on rules. So for example, the inference of

$$A, A \to (A \to B) \vdash B$$

requires two uses of the item A, one use of the item $A \to (A \to B)$ and two uses of the rule of Modus Ponens.

Linear logic allows the use of premise A only once. Suppose we allocate a fixed number of uses of Modus Ponens, involving certain formulas, but unlimit the number of uses of items of data, then what do we get?

In this paper, we explore a new way of controlling inferences, and focus on by restricting the use of the *cut rule*. The cut rule is a central most famous inference rule in a logical system. It is useful but is computationally inefficient when cut-formulas have to be non-deterministically guessed, and eliminating cut has become a central point in the study of logic inference.

Of course, in classical logic as well as in all of the above mentioned systems, the cut rule is eliminable. So, it does not make sense to control something that can always be avoided. However, it may be a good idea, from a computational and philosophical point of view, to have systems in which some form of cut is allowed or mandatory. From the computational point of view, we know that there are families of formulas whose proof size have an exponential lower bound for cut-free sequent systems, but that have very simple, linear or at most quadratic, sequent proofs with cuts (Boolos, 1984; Carbone and Semmes, 2000). Furthermore, the cut rule represents the use of auxiliary lemmas in complex theorem proofs, and are an inherent part of the mathematical activity.

Our starting point is thus a *cut-based* sequent system for classical logic, in which the cut rule is non-eliminable. Such a system is developed around a formulation by D'Agostino and Mondadori (1994). This system allows us to define proof methods that control the use of cuts in proofs. In these cut-limited systems, one has to be able to “pay” for each use of cut, thus generating cut-and-pay inferences. Different restrictions (i.e. forms of “payment”) generates different families of inferences.

We thus propose three families of cut-and-pay inference systems by means of distinct forms of restrictions on the cut rule. We show that each such family constitutes an *approximation process* towards classical logic. In this sense, the inferences defined here are in line with other families logics that perform approximation processes (see Schaerf and Cadoli, 1995; Dalal, 1996a; Massacci, 1998; Finger and Wassermann, 2004).

Sound and complete semantics are provided for each component of those three families of logics for approximate reasoning. We then provide decision procedures for each family of logics. In particular, a tableau based decision procedure is given for one family of logics, and the other inference systems use such tableaus as basic building blocs in their decision procedures. Detailed analysis of the complexity of each decision procedure is given. We show that one of the families has a linear time decision procedure for each component. We show that another, more expressive family has a polynomial time decision procedure for each component.

The rest of the paper develops as follows. Section 2 contains a Gentzen-system presentation of propositional classical logic in which cut is *not* eliminable, and which will serve as the basis for controlling the use of cut; it is shown that such systems also accepts some forms of proof normalisation which are very convenient for later complexity analysis of cut-controlled proofs. Section 3 presents three Cut-and-Pay families of logics, showing that each family provides an approximation process for full classical logic when the control over the use of cut is progressively weakened. The semantics of those families is presented in Section 4, with proofs of soundness and completeness correspondence for each component of each of the three families of approximated logics; one of the families of approximated logics is shown to possess uniform substitution property, an important meta-property of logics which is new for approximated systems. Finally, Section 5 presents tableau based decision procedures for each element of the approximation families and the complexity of each decision procedure is studied. We show that there are families in which each element can be decided polynomially, although the whole approximation process remains exponential.

2. A Cut-Based System for Classical Logic

Since our aim is to control – i.e. to *pay* for – the use of the cut rule, we cannot admit the usual rules for the Gentzen sequent calculus, in which case the cut rule is eliminable. With the usual connective and structural rules, any proof can be done without any use of cut, but the "price" one pays for the eliminating is that of having very large proofs when much smaller proofs with cut are possible.

Instead we start from a version of the sequent calculus in which cut is not eliminable. In this approach, the classical structural rules are kept, as shown in Figure 1. We assume a sequent to be represented by $\Gamma \vdash \Delta$, where Γ, Δ are sequences of formulas, with the intended meaning that the conjunction of the formulas in Γ prove the disjunction of the formulas in Δ. Note that, in Figure 1, the axiom rule contains implicitly the weakening of the premisses and the consequences, but this weakening is restricted to the leafs of the proof where axioms are found; in the axiom rule, we call the distinguished formula A the *main* formula and any formula in Γ and Δ a weak formula. As for the cut rule, the distinguished formula A is called the *cut formula*.

The connective rules of the sequent calculus are altered, so as to block cut elimination. Those connective rules are presented in Figure 2, which is greatly

$\dfrac{}{\Gamma, A \vdash A, \Delta}$ (Axiom)	$\dfrac{\Gamma_1 \vdash \Delta_1, A \quad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$ (Cut)
$\dfrac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta}$ (Commut $\vdash$)	$\dfrac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2}$ ($\vdash$ Commut)
$\dfrac{\Gamma_1, A, A, \Gamma_2 \vdash \Delta}{\Gamma_1, A, \Gamma_2 \vdash \Delta}$ (Contract $\vdash$)	$\dfrac{\Gamma \vdash \Delta_1, A, A, \Delta_2}{\Gamma \vdash \Delta_1, A, \Delta_2}$ ($\vdash$ Contract)

Figure 1. Classical structural rules for the sequent calculus.

$\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$ $(\wedge \vdash)$	$\dfrac{\Gamma \vdash \Delta, A}{\Gamma, B \vdash \Delta, A \wedge B}$ $(\vdash \wedge_1)$	$\dfrac{\Gamma \vdash \Delta, A}{B, \Gamma \vdash \Delta, B \wedge A}$ $(\vdash \wedge_2)$
$\dfrac{\Gamma, A \vdash \Delta}{\Gamma, A \vee B \vdash \Delta, B}$ $(\vee \vdash_1)$	$\dfrac{\Gamma, A \vdash \Delta}{\Gamma, B \vee A \vdash B, \Delta}$ $(\vee \vdash_2)$	$\dfrac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}$ $(\vdash \vee)$
$\dfrac{\Gamma, B \vdash \Delta}{\Gamma, A \to B, A \vdash \Delta}$ $(\to\vdash_1)$	$\dfrac{\Gamma \vdash A, \Delta}{\Gamma, A \to B \vdash \Delta, B}$ $(\to\vdash_2)$	$\dfrac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \to B}$ $(\vdash\to)$
$\dfrac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}$ $(\neg \vdash)$	$\dfrac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}$ $(\vdash \neg)$	

Figure 2. Connective rules for the cut-based sequent calculus.

inspired by the ideas of D'Agostino (1992) and D'Agostino and Mondadori (1994). Instead of being a cut-free calculus, this is a *cut-based* calculus.

If we compare the connective rules in Figure 2 with a standard, cut-free presentation, the cut-based one-premissed rules $(\vdash \wedge_1)$, $(\vdash \wedge_2)$, $(\vee \vdash_1)$, $(\vee \vdash_2)$, $(\to\vdash_1)$ e $(\to\vdash_2)$ in Figure 2 are usually replaced by the two-premissed sequent rules in Figure 3.

LEMMA 2.1. *In the presence of all structural rules, the cut-based one-premissed rules and their correspondent in Figure 3 rules are interdefinable such that, when*

$$\dfrac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} (\vdash \wedge)$$

$$\dfrac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} (\vee \vdash)$$

$$\dfrac{\Gamma_1, B \vdash \Delta_1 \quad \Gamma_2 \vdash \Delta_2, A}{\Gamma_1, A \to B, \Gamma_2 \vdash \Delta_1, \Delta_2} (\to\vdash)$$

Figure 3. Two-premissed sequent rules in usual presentations.

cut is applied, the cut formula is always a subformula of some formula occurring in the cut premisses.

Proof. We show the interdefinition between $(\rightarrow\vdash_1)$ and $(\rightarrow\vdash)$. Initially, assume $(\rightarrow\vdash_1)$, then

$$\dfrac{\Gamma_1 \vdash \Delta_1, A \qquad \dfrac{\Gamma_2, B \vdash \Delta_2}{\Gamma_2, A \rightarrow B, A \vdash \Delta_2}\ (\rightarrow\vdash_1)}{\Gamma_2, A \rightarrow B, \Gamma_1 \vdash \Delta_1, \Delta_2}\ (Cut).$$

Note that the cut formula A is a subformula $A \rightarrow B$ in the cut right premiss. Due to the commutativity structural rules, the order of the formulas is the sequent is not important. Now assume $(\rightarrow\vdash)$, then

$$\dfrac{\Gamma, B \vdash \Delta \qquad A \vdash A}{\Gamma, A \rightarrow B, A \vdash \Delta}\ (\rightarrow\vdash)$$

The other cases are analogous and thus omitted. □

The fact that the cut *can* be made over some subformula, as shown in Lemma 2.1, does not mean it *must* be made only on subformulas. This implies that the cut-based system possesses the *subformula property*, that is, every provable sequent has a proof in which, for every rule application, every formula in a premiss is a subformula of some formula in the conclusion.

A cut is called *analytic* if the cut formula is a subformula of some formula in the conclusion of the cut application. Clearly, analytic cuts preserve the subformula property. Furthermore, we have the following property.

LEMMA 2.2. *Every classically provable sequent has a cut-based proof using only analytic cuts.*

Proof. Consider a classically provable sequent and one cut-free proof of it. By Lemma 2.1, there is a cut-based proof that simulates such proof. □

We do not want to be restricted to cut-free or analytical inference systems because proofs with cut can sometimes be significantly smaller than cut-free proofs or proofs with restricted forms of cut. With respect to the size of proofs, the proof of Lemma 2.1 implies the following result.

LEMMA 2.3. *The cut-based inference system can linearly simulate cut-free proofs.*

Proof. Just note that the simulation of rules $(\vdash \wedge)$, $(\vee \vdash)$ and $(\rightarrow\vdash)$ in the cut-based system can be done with a linear (or sub-linear) addition of new lines to the proof. □

2.1. Normalising Cut-based Proofs

We say that a formula A *reappears* in a proof if there is a branch in which A occurs twice in the lhs (or in the rhs) of a sequent, but is absent in between these two occurrences. The following is a trivial consequence of cut elimination in cut-free systems, but it also applies to the cut-based system as a proof normalisation procedure.

LEMMA 2.4. *If there is a provable sequent* $\Gamma \vdash \Delta$ *whose proof contains a branch with a cut over a formula A that reappears further down in the proof. Then* $\Gamma \vdash \Delta$ *has a proof with that cut eliminated.*

Proof. Suppose $\Gamma \vdash \Delta$ has a proof with the following configuration

$$\cfrac{\Gamma_1, A \vdash \Delta_1 \qquad\qquad \Gamma_2 \vdash A, \Delta_2}{\begin{array}{c}\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2 \\ \vdots\ \Pi \\ \Gamma_3, A \vdash \Delta_3\end{array}}$$

where Π is the proof segment between the two sequents in that branch. Without loss of generality, we assume that A reappears on the rhs of a sequent. Then, by monotonicity, we can expand $\Gamma_1, A \vdash \Delta_1$ to $\Gamma_1, \Gamma_2, A \vdash \Delta_1, \Delta_2$, thus obtaining

$$\cfrac{\begin{array}{c}\Gamma_1, \Gamma_2, A \vdash \Delta_1, \Delta_2 \\ \vdots\ \Pi \\ \Gamma_3, A, A \vdash \Delta_3\end{array}}{\Gamma_3, A \vdash \Delta_3}$$

so that the upper cut is replaced by a lower application of contraction. □

This result allows us to assume that every provable sequent in the cut-based system has a proof where no branch contains two cuts over the same formula.

There are certain cases in cut-based proofs where some cuts can be eliminated, such as the case where a branch has cuts over A and $\neg A$, or cuts over A, B and $A \circ B$ for $\circ \in \{\wedge, \vee, \rightarrow\}$.

LEMMA 2.5. *Any sequent* $\Gamma \vdash \Delta$ *whose cut-based proof has a branch containing cuts on a compound formula and its immediate subformulas also has a cut-based proof with one of the cuts eliminated.*

Proof. We have to analyse two cases for a cut over $\neg A$ and six cases for each of the binary connectives. We present the proof only for the cuts over $\neg A$. Suppose we have a proof

$$\dfrac{\dfrac{\Gamma_1, \neg A \vdash \Delta_1 \qquad \Gamma_2 \vdash \neg A, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \quad \begin{array}{c} \vdots\ \Pi \\ \Gamma_3, A \vdash \Delta_3 \end{array} \qquad \Gamma_4 \vdash A, \Delta_4}{\Gamma_3, \Gamma_4 \vdash \Delta_3, \Delta_4}$$

Consider the sequent $\Gamma_1, \neg A \vdash \Delta_1$. One possibility is that $\neg A$ was obtained by rule $(\neg \vdash)$ from $\Gamma_1 \vdash A, \Delta_1$. Alternatively, $\neg A$ was introduced by an axiom, either as a main or a weak formula; in both cases, we can rearrange the proof so that $\Gamma_1 \vdash A, \Delta_1$ holds. Similarly, we can rearrange the proof so that $\Gamma_2, A \vdash \Delta_2$ is derivable so we have

$$\dfrac{\dfrac{\Gamma_1 \vdash A, \Delta_1 \qquad \Gamma_2, A \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \quad \begin{array}{c} \vdots\ \Pi \\ \Gamma_3, A \vdash \Delta_3 \end{array} \qquad \Gamma_4 \vdash A, \Delta_4}{\Gamma_3, \Gamma_4 \vdash \Delta_3, \Delta_4}$$

By Lemma 2.4 this proof can be rewritten with only one cut over A.

On the other hand, suppose we have a proof containing

$$\dfrac{\dfrac{\Gamma_1, A \vdash \Delta_1 \qquad \Gamma_2 \vdash A, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \quad \begin{array}{c} \vdots\ \Pi \\ \Gamma_3, \neg A \vdash \Delta_3 \end{array} \qquad \Gamma_4 \vdash \neg A, \Delta_4}{\Gamma_3, \Gamma_4 \vdash \Delta_3, \Delta_4}$$

This proof can be converted to

$$\dfrac{\dfrac{\dfrac{\Gamma_1, A \vdash \Delta_1}{\Gamma_1 \vdash \neg A, \Delta_1} \qquad \dfrac{\Gamma_2 \vdash A, \Delta_2}{\Gamma_2, \neg A \vdash \Delta_2}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \quad \begin{array}{c} \vdots\ \Pi \\ \Gamma_3, \neg A \vdash \Delta_3 \end{array} \qquad \Gamma_4 \vdash \neg A, \Delta_4}{\Gamma_3, \Gamma_4 \vdash \Delta_3, \Delta_4}$$

Again, by Lemma 2.4, this proof can be rewritten with only one cut over $\neg A$. The other cases are straightforward repetitions of the techniques displayed above and are thus omitted. □

Finally, the last property we show cut-based proofs has to do with the cut-order. The *cut-order* of a branch in a cut-based proof is the sequence of cut formulas in that branch. We then proof the *cut-order independence property*:

LEMMA 2.6. *If* $\Gamma \vdash \Delta$ *has a cut-based proof with a branch with a certain cut-order, then* $\Gamma \vdash \Delta$ *has a cut-based proof with a branch containing any permutation of that cut-order.*

Proof. All we have to do is show that the order of two cuts in a branch can be commuted. For that, suppose we have a proof chunk:

$$\dfrac{\dfrac{\dfrac{\Gamma_1, A \vdash \Delta_1 \qquad \Gamma_2 \vdash A, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}}{\vdots\ \Pi}}{}\qquad \dfrac{\Gamma_3, \vdash B, \Delta_3 \qquad \Gamma_4, B \vdash \Delta_4}{\Gamma_3, \Gamma_4 \vdash \Delta_3, \Delta_4}$$

Note that Π is a proof segment that applies proof rules to the elements of $\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2$. Let us split Π into Π_1 and Π_2, such that Π_i applies proof rules the the formulas originally in Γ_i, Δ_i. Without loss of generality, suppose that the formula B emerges from Π_1, such that we obtain the following proof chunk:

$$\dfrac{\dfrac{\dfrac{\Gamma_1, A \vdash \Delta_1}{\vdots\ \Pi_1 \atop \Gamma_1', A \vdash B, \Delta_1'} \qquad \Gamma_4, B \vdash \Delta_4}{\Gamma_1', \Gamma_4, A \vdash \Delta_1', \vdash \Delta_4} \qquad {\Gamma_2 \vdash A, \Delta_2 \atop {\vdots\ \Pi_2 \atop {\vdots \atop \Gamma_2' \vdash A, \Delta_2'}}}}{\Gamma_1', \Gamma_2', \Gamma_4 \vdash \Delta_1', \Delta_2', \Delta_4}$$

but $\Gamma_1', \Gamma_2' = \Gamma_3$ and $\Delta_1', \Delta_2' = \Delta_3$, which shows that the order of the cut formulas A and B can be exchanged, and the result is proved. □

A cut is called *analytic* if the cut formula is a subformula of some formula in the conclusion of the cut application. Analytic cuts preserve the *subformula property*.

3. Cut-and-Pay Inference Systems

There are many possible ways in which the use of the cut can be limited in a system so as to implement a Cut-and-Pay proof system. We are going to limit ourselves here to three forms of disciplining the use of cut over the cut-based deduction system.

Before we present the "cut-and-pay" limitations in the use of cut, we need to add to the inference systems a few extra rules, which are easily inferable in the cut-based inference system. However, since we are going to limit the use of cut, it

$$\frac{\Gamma \vdash \Delta, A}{\Gamma, B \vdash \Delta, A \wedge A}\ (\vdash \wedge_{id}) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma, A \vee A \vdash \Delta, B}\ (\vee \vdash_{id})$$

Figure 4. Idempotency inference rules for $\wedge$ and $\vee$.

is necessary to add those rules so as to provide a complete semantics for the limited systems. So we add the two *idempotency* inference rules in Figure 4.

These rules are *admissible* in the classical cut-based calculus, in the sense that they do no add any new derivable sequents to the logic, as can be easily seen. It is also important to note that the addition of the idempotency rules $(\vdash \wedge_{id})$ and $(\vee \vdash_{id})$ does not make cut eliminable. In fact, the cut rule is the only rule with two premisses and one conclusion in the cut-based calculus; the idempotency rules, as well as all the other "linear" rules in Figure 2, have only one premiss. The presence of two premised rules is essential to obtaining the full expressivity of classical logic, which means that cut is not eliminable with the addition of the idempotency rules.

Also note that idempotency rules preserve the subformula property. As will be clear in Section 4, the idempotency rules guarantee that the formulas A and $A \wedge A$ and $A \vee A$ are always assigned the same truth value.

We now proceed to present three families of cut-limited inference systems.

Formula Limitation
Cut can only be applied to formulas which belong to a set of formulas $\Sigma \subseteq \mathcal{L}$:

$$\frac{\Gamma_1 \vdash \Delta_1, A \quad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \text{ provided } A \in \Sigma$$

The application of cut in a proof is limited to cut formulas belonging to an externally provided set Σ. In this sense, Σ is a parameter of the limited system. We require the closure of Σ under *negation prefixing*, that is, if $\alpha \in \Sigma$ then $\neg\alpha \in \Sigma$.[1]

We write $\Gamma \vdash_{\Sigma} \Delta$ to represent the cut-based inference system in which cut is limited to the formulas in Σ.

Formula limitation is local in nature, for the decision on the application of the cut does not depend on the rest of the proof.

Atom Number Limitation
Cut can only be applied to a cut formula whose number of atoms is at most a fixed k:

$$\frac{\Gamma_1 \vdash \Delta_1, A \quad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \text{ provided } |\text{atoms}(A)| \leq k$$

where atoms(A) is the set of atoms occurring in the formula A.

[1] This differs from the presentation in Finger (2004a), for there Σ was required to be closed under *formula formation*, that is, if $A, B \in \Sigma$ the $\neg A$, $A \wedge B$, $A \vee B$, $A \to B \in \Sigma$.

In this case, k is a parameter of the limited system. This condition is clearly local.

We write $\Gamma \vdash_k^a \Delta$ to represent the cut-based inference system in which cut is limited to the formulas whose number of atoms is at most k.

It is also possible to view $\vdash_k^a$ as a restriction of $\vdash_\Sigma$. To see that, let $\Sigma_k = \{A \mid |\text{atoms}(A)| \leq k\}$, then $\vdash_k^a = \vdash_{\Sigma_k}$. Note that this makes $\vdash_k^a$ a direct generalisation of Dalal's clausal approximations of classical logic, which now accepts any propositional formula (Dalal, 1996a,b).

Limiting the Number of Distinct Cut Formulas

In this case, cut can only be applied to a certain number k of *distinct* formulas.

This does not imply that there are only k uses of cut in a proof, but among the uses of cut, at most k cut formulas can be found in a proof. In this case, k is a parameter of the limited system.

We write $\Gamma \vdash_k^d \Delta$ to represent the cut-based inference system in which proofs are limited to at most k distinct cut formulas. If we measure the height of a proof by the number of branching points, Lemma 2.4 implies that $\vdash_k^d$ also limits the height of a tree to at most k.

This limiting condition is global, for the decision on the application of the cut depends on the whole proof. As a consequence, $\vdash_k^d$ cannot be seen as a restriction of the inference $\vdash_\Sigma$.

To express $\vdash_k^d$ we have to make the following consideration. Because Σ is closed under negation prefixing, a non-empty Σ is certainly infinite. However, we can identify each sequence of formulas $\alpha, \neg\alpha, \neg\neg\alpha, \ldots$, with its first element. In this sense, we can say that Σ is *finitely generated* if, by identifying all elements of a chain with its first element, we obtain a finite set. In such case, let $|\Sigma|$ be the *cardinality* (of the basis) of a finitely generated Σ.

From Lemma 2.5, we know that every provable sequent has a proof in which all branches contains only one element of this chain, so in terms of total number of cuts in a proof we do not loose anything by identifying those chains.

Then $\vdash_k^d$ can be seen as a union of several $\vdash_\Sigma$, given by

$$\vdash_k^d = \bigcup_{|\Sigma| \leq k} \vdash_\Sigma,$$

which means that any $\vdash_k^d$-inference can be seen as a $\vdash_\Sigma$-inference for some Σ whose cardinality is at most k.

3.1. Approximation Processes

An important property of the logics above is that they constitute *families* of logics that perform an *approximation process* towards classical logic.

In the case of the family of formulas generated by formula limitation, the parameter that defines an element in the family is the set Σ of formulas over which branching is allowed. The approximation process is obtained when the parameter set Σ varies over an ascending chain

$$\varnothing \subseteq \Sigma' \subseteq \Sigma'' \ldots \subseteq \Sigma^{n'} \subseteq \mathcal{L}$$

so that we obtain a family of logics with increasing expressivity, whose inferences are all subclassical:

$$\vdash_{\varnothing} \subseteq \vdash_{\Sigma'} \subseteq \vdash_{\Sigma''} \ldots \subseteq \vdash_{\Sigma^{n'}} \subseteq \vdash_{\mathcal{L}} = \vdash_{\mathrm{CL}},$$

where $\vdash_{\mathrm{CL}}$ is classical inference. This family of logics was first studied in Finger (2004a,b).

In the family of logics generated by atom number limitation, the parameter defining the logic is simply the number k of atoms, and clearly we have

$$\vdash_0^a \subseteq \vdash_1^a \subseteq \vdash_2^a \ldots \subseteq \vdash_k^a \subseteq \cdots \subseteq \vdash_{\mathrm{CL}} .$$

The family of logics generated by limiting distinct cut formulas is parameterised by the number k of distinct cut formulas, such that

$$\vdash_0^d \subseteq \vdash_1^d \subseteq \vdash_2^d \ldots \subseteq \vdash_k^d \subseteq \cdots \subseteq \vdash_{\mathrm{CL}} .$$

For every classically provable sequent $\Gamma \vdash_{\mathrm{CL}} \Delta$, it is easy to see that there exists a (smallest) k such that $\Gamma \vdash_k^a \Delta$ and a (smallest) k' such that $\Gamma \vdash_{k'}^d \Delta$. In fact, suppose we have a cut-based proof for $\Gamma \vdash_{\mathrm{CL}} \Delta$, then $k = \max |\mathrm{atom}(A)|$ where A is a cut formula, and k' is the number of distinct cuts in the proof.

As a consequence, the process of approximating a classical sequent $\Gamma \vdash_{\mathrm{CL}} \Delta$ is always finite in all three approximation processes above.

4. Semantics for Cut-and-Pay

The semantics presented here is a variation of that presented in Finger (2004a), although not exactly equivalent. We start by defining the semantics for the system $\vdash_\Sigma$, as the semantics for $\vdash_k^a$ and $\vdash_k^d$ can be derived in terms of that. In the following we define $\models_\Sigma$ as the semantics of *limited bivalence*, with limitation Σ.

The semantics of Σ-limited bivalence is based on a three-level lattice, $\mathsf{L} = (L, \sqcap, \sqcup, \sim, 0, 1)$, where L is a countable set of elements $L = \{0, 1, \epsilon_0, \epsilon_1, \epsilon_2, \ldots\}$ such that $0 \sqsubseteq \epsilon_i \sqsubseteq 1$ for every $i < \omega$ and $\epsilon_i \not\sqsubseteq \epsilon_j$ for $i \neq j$. The ϵ_i's are called *neutral* truth values; as usual, $\sqcup$ represents the least upper bound operation and $\sqcap$ is the greatest lower bound operation. This three-level lattice is illustrated in Figure 5(a). The lattice contains a *converse operation*, $\sim$, defined as: $\sim 0 = 1$, $\sim 1 = 0$ and $\sim \epsilon_i = \epsilon_i$ for all $i < \omega$. The converse operation is illustrated in Figure 5(b).

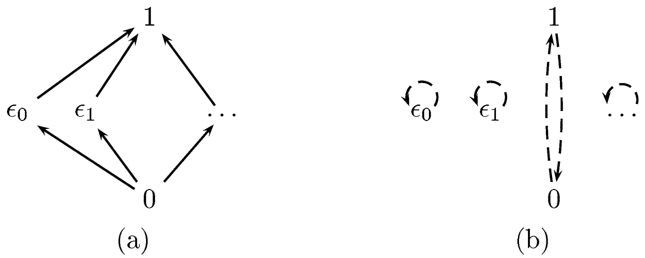

Figure 5. The 3-level lattice (a) and its converse operation (b).

An *unlimited (propositional) valuation* is a function $v_\Sigma\colon \mathcal{P} \to L$ that maps atoms to elements of the lattice. We extend v_Σ to all propositional formulas, $v_\Sigma\colon \mathcal{L} \to L$, in the following way:

$$v_\Sigma(\neg\alpha) = \sim v_\Sigma(\alpha)$$
$$v_\Sigma(\alpha \wedge \beta) = v_\Sigma(\alpha) \sqcap v_\Sigma(\beta)$$
$$v_\Sigma(\alpha \vee \beta) = v_\Sigma(\alpha) \sqcup v_\Sigma(\beta)$$
$$v_\Sigma(\alpha \to \beta) = \begin{cases} 1 & \text{if } v(\alpha) \sqsubseteq v(\beta) \\ \sim v_\Sigma(\alpha) \sqcup v_\Sigma(\beta) & \text{otherwise} \end{cases}$$

For example, consider $\Sigma = \varnothing$. Then

- if $v_\Sigma(p) = v_\Sigma(q)$, then $v_\Sigma(p \to q) = 1$;
- if $v_\Sigma(p) = \epsilon_p$ and $v_\Sigma(q) = \epsilon_q$, $\epsilon_p \neq \epsilon_q$, then $v_\Sigma(p \to q) = v_\Sigma(p \vee q) = 1$ and $v_\Sigma(p \wedge q) = 0$.
- if $v_\Sigma(p) = v_\Sigma(q) = \epsilon_i$ then $v_\Sigma(p \to q) = v_\Sigma(p \vee q) = v_\Sigma(p \wedge q) = v_\Sigma(\neg p) = \epsilon_i$.
- If $v_\Sigma(p) \in \{0, 1\}$ or $v_\Sigma(q) \in \{0, 1\}$, the behaviour of $v_\Sigma(p \to q)$, $v_\Sigma(p \vee q)$ and $v_\Sigma(p \wedge q)$ is classical, namely:
  - if $v_\Sigma(p) = 1$, then $v_\Sigma(p \to q) = v_\Sigma(p \wedge q) = v_\Sigma(q)$, $v_\Sigma(p \vee q) = 1$;
  - if $v_\Sigma(p) = 0$, then $v_\Sigma(p \to q) = 1$, $v_\Sigma(p \wedge q) = 0$, $v_\Sigma(p \vee q) = v_\Sigma(q)$;
  - if $v_\Sigma(q) = 0$, then $v_\Sigma(p \to q) = \sim v_\Sigma(p)$;
  - if $v_\Sigma(q) = 1$, then $v_\Sigma(p \to q) = 1$;

A Σ-*limited valuation*, or simply a *valuation*, is an unlimited valuation which, with regards to whether a formula is or is not in the parameter set Σ, satisfies the Limited Bivalence Restrictions:

(a) if $\alpha \in \Sigma$ then $v_\Sigma(\alpha)$ must be a *bivalent* unlimited valuation, that is, $v_\Sigma(\alpha) = 0$ or $v_\Sigma(\alpha) = 1$;
(b) if $\alpha \notin \Sigma$ then either $v_\Sigma(\alpha)$ obeys the rules of unlimited valuations *or* $v_\Sigma(\alpha) = \epsilon_i$, for some ϵ_i.

The first conditions forces the elements of Σ to be bivalent. The second condition tells us that the truth value assigned to a formula $\alpha \notin \Sigma$ is not always *compositional*, for a neutral value may be assigned to α independently of the truth value of its components. This is the case so that the bivalence of $\alpha \in \Sigma$ can always be satisfied without forcing all of α's subformulas to be bivalent.

If $\alpha \in \Sigma$ it is always possible to have $v_\Sigma(\alpha) \in \{0, 1\}$ by making for every atom p in α, $v_\Sigma(p) \in \{0, 1\}$. However, this is not the only possibility. For example, if $\beta, \gamma \notin \Sigma$ then we can make $v_\Sigma(\beta) = \epsilon_i \neq \epsilon_j = v_\Sigma(\gamma)$, so that $v_\Sigma(\beta \wedge \gamma) = 0$; similarly, we obtain $v_\Sigma(\beta \vee \gamma) = 1$ and $v_\Sigma(\beta \to \gamma) = 1$.

A valuation v_Σ *satisfies* α if $v_\Sigma(\alpha) = 1$, and α is called *satisfiable*; a set of formulas Γ is satisfied by v_Σ if all its formulas are satisfied by v_Σ. A valuation v_Σ *contradicts* α if $v_\Sigma(\alpha) = 0$; if α is neither satisfied nor contradicted by v_Σ, we say that v_Σ is *neutral* with respect to α. A valuation is *classical* if it assigns only 0 or 1 to all proposition symbols, and hence to all formulas.

4.1. Approximated Entailments

We first define here three notions of entailment, namely $\models_\Sigma$, $\models_k^a$ and $\models_k^d$, corresponding respectively to $\vdash_\Sigma$, $\vdash_k^a$ and $\vdash_k^d$.

- $\Gamma \models_\Sigma \Delta$ if it is not possible to satisfy Γ and contradict every $\alpha \in \Delta$ at the same time. More specifically, $\Gamma \models_\Sigma \Delta$ if no valuation v_Σ such that $v_\Sigma(\Gamma) = 1$ also makes $v_\Sigma(\alpha) = 0$ for every $\alpha \in \Delta$. Note that since this logic is not classical, if $\Gamma \models_\Sigma \Delta$ and $v_\Sigma(\Gamma) = 1$ it is possible that there is an $\alpha \in \Delta$ that is either neutral or satisfied by v_Σ.
- We define $\models_k^a$ in analogy to what was done in Section 3. Let $\Sigma_k = \{A \mid |\text{atoms}(A)| \leq k\}$ be the set of formulas that contain at most k atoms. Then $\models_k^a = \models_{\Sigma_k}$.
- Similarly, define $\models_k^d$ as the union of all $\models_\Sigma$ for which the cardinality of Σ is at most k:

$$\models_k^d = \bigcup_{|\Sigma| \leq k} \models_\Sigma .$$

We first note that $\models_\varnothing = \models_0^a = \models_0^d$. Let us see a more complex example. Consider $\Gamma_0 = \{p \vee q, q \to p, p \to (s \vee t), (p \wedge t) \to s\}$ and make $\Sigma = \varnothing$. Then we have that

(1) $\Gamma_0 \models_\Sigma p$ and
(ii) $\Gamma_0, p \models_\Sigma s$ but
(iii) $\Gamma_0 \not\models_\Sigma s$.

To check for (i), suppose there is a v_Σ such that $v_\Sigma(p) = 0$. Then we have $v_\Sigma(p \vee q) = v_\Sigma(q)$ and $v_\Sigma(q \to p) = {\sim} v_\Sigma(q)$. Since it is not possible to satisfy both, we cannot have $v_\Sigma(\Gamma_0) = 1$, so $\Gamma_0 \models_\Sigma p$.

To obtain (ii), suppose there is a v_Σ such that $v_\Sigma(s) = 0$ and $v_\Sigma(p) = 1$. Then $v_\Sigma(p \to (s \vee t)) = v_\Sigma(t)$ and $v_\Sigma((p \wedge t) \to s) = {\sim} v_\Sigma(t)$. Again, it is not possible to satisfy both, so $\Gamma_0, p \models_\Sigma s$.

For the verification of (iii), take a valuation v_Σ such that $v_\Sigma(s) = 0$, $v_\Sigma(p) = \epsilon_p$, $v_\Sigma(q) = \epsilon_q$, $v_\Sigma(t) = \epsilon_t$. Then $v_\Sigma(\Gamma_0) = 1$.

However, if we enlarge Σ and make $p \in \Sigma$, then we have only two possibilities for $v_\Sigma(p)$. If $v_\Sigma(p) = 1$, we have already seen that no valuation that contradicts s will satisfy Γ_0. If $v_\Sigma(p) = 0$, we have also seen that no valuation that contradicts s will satisfy Γ_0. So for $p \in \Sigma$, we obtain $\Gamma_0 \models_\Sigma s$. As a consequence, we also obtain $\Gamma_0 \models_1^a s$ and $\Gamma_0 \models_1^d s$.

It follows from the above that, with respect to the $\vdash_k^a$-inference:

- $\Gamma_0 \not\models_0^a s$, for no inference is possible without the use of a cut.
- $\Gamma_0 \models_1^a s$, for a deduction is possible with a cut on a single atom, namely p.

Similarly, with respect to the $\vdash_k^d$-inference, we observe that:

- $\Gamma_0 \not\models_0^d s$, for no inference is possible without the use of a cut (this is the same as $\models_0^a$ and $\models_\varnothing$).
- $\Gamma_0 \models_1^d s$, for a deduction is possible with a single cut formula, namely p.

Idempotency

From the semantics above, it is easy to see that $p \models_\Sigma p \wedge p$ even when $p \notin \Sigma$. Let us consider now whether $p \models_\Sigma (p \vee q) \wedge (p \vee q)$ when none of p, $p \vee q$ and $(p \vee q) \wedge (p \vee q)$ are in Σ. To satisfy the antecedent we make $v_\Sigma(p) = 1$, so even if we assign $v_\Sigma(p \vee q) = \epsilon_1$ we obtain $v_\Sigma((p \vee q) \wedge (p \vee q)) = \epsilon_1 \neq 0$, so it is not possible to contradict the consequent, and thus $p \models_\Sigma (p \vee q) \wedge (p \vee q)$. This fact is responsible for the addition of the idempotency rules in Figure 4 to the cut-and-pay inference system, otherwise it would not be possible to infer $p \vdash_\Sigma (p \vee q) \wedge (p \vee q)$ without cuts.

So the idempotency rules are used to cover a potential completeness gap.

4.2. Soundness and Completeness

As usual, an inference system $\vdash$ is considered *sound* with respect to a semantics $\models$ iff whenever $\Gamma \vdash \Delta$ then $\Gamma \models \Delta$. Conversely, the inference system is considered complete iff whenever $\Gamma \models \Delta$ then $\Gamma \vdash \Delta$.

THEOREM 4.1. *Let Γ and Δ be sequences of formulas. Then*

(i) The inference system $\vdash_\Sigma$ is sound and complete with respect to $\models_\Sigma$.

(ii) The inference system $\vdash^a_k$ *is sound and complete with respect to* $\models^a_k$*.*
(iii) The inference system $\vdash^d_k$ *is sound and complete with respect to* $\models^d_k$ *a.*

Proof. We need only to show (i), as (ii) and (iii) follow directly from it. Soundness is a straightforward verification of the validity of the rules in Figures 1 and 2 and the limited cut rule; details omitted.

For completeness, let $\Gamma = \gamma_1, \ldots, \gamma_n$ and $\Delta = \delta_1, \ldots, \delta_m$, and assume $\Gamma \not\vdash_\Sigma \Delta$. We want to show that $\Gamma \not\models_\Sigma \Delta$.

Let $\mathrm{Lit}(\Gamma, \Delta) = \mathrm{Subf}(\Gamma) \cup \mathrm{Subf}(\Delta) \cup \{\neg\alpha | \alpha \in \mathrm{Subf}(\Gamma) \cup \mathrm{Subf}(\Delta)\}$, where $\mathrm{Subf}(\Psi)$ is the set of all subformulas of formulas in the sequence Ψ. Take an enumeration of the formulas in $\mathrm{Lit}(\Gamma, \Delta) = \lambda_1, \ldots, \lambda_\ell$ and construct a maximal consistent set Φ restricted to $\mathrm{Lit}(\Gamma, \Delta)$ as follows:

$$\Phi^0 = \{\gamma_1, \ldots, \gamma_n, \neg\delta_1, \ldots \neg\delta_m\}$$
$$\Phi^{i+1} = \begin{cases} \Phi^i, & \Phi^i \vdash_\Sigma \neg\lambda_{i+1} \\ \Phi^i \cup \{\lambda_{i+1}\}, & \text{otherwise} \end{cases}$$

and let $\Phi = \Phi^\ell$. We construct a valuation v_Σ in the following way. Let p be an atom. If $p \in \Phi$ then $v_\Sigma(p) = 1$; if $\neg p \in \Phi$ then $v_\Sigma(p) = 0$. If neither $p \in \Phi$ nor $\neg p \in \Phi$, then if $p \in \Sigma$, $v_\Sigma(p) = 0$; otherwise, $v_\Sigma(p) = \epsilon_j$, such that no two atoms are assigned to the same neutral value.

We now show that v_Σ can be extended such that $v_\Sigma(\varphi) = 1$ for every $\varphi \in \Phi$. We proceed by structural induction on φ.

If $\varphi = \neg\neg\alpha$, it is easy to see that $\alpha \in \Phi$, so by induction hypothesis, $v_\Sigma(\alpha) = 1$, and we take $v_\Sigma(\neg\neg\alpha) = 1$.

If $\varphi = \alpha\wedge\beta$, then $\alpha, \beta \in \Phi$; by induction hypothesis, $v_\Sigma(\alpha) = v_\Sigma(\beta) = 1$, so we choose $v_\Sigma(\alpha \wedge \beta) = 1$, even if $\varphi \notin \Sigma$. If $\varphi = \neg(\alpha\wedge\beta)$, if neither $\alpha \in \Sigma$ nor $\beta \in \Sigma$, then if $\alpha \neq \beta$ we can make $v_\Sigma(\alpha) = \epsilon_i \neq \epsilon_j = v_\Sigma(\beta)$ such that $v_\Sigma(\neg(\alpha \wedge \beta)) = 1$, and if $\alpha = \beta$, using the idempotency rule we obtain that $\neg\alpha \in \Phi$, so that the induction hypothesis gives $v_\Sigma(\alpha) = 0$ and we can make $v_\Sigma(\neg(\alpha \wedge \beta)) = 1$. Otherwise, suppose without loss of generality that $\alpha \in \Sigma$, then $\neg\alpha \in \Phi$ or $\neg\beta \in \Phi$ (using classical reasoning), so that the induction hypothesis gives us that either $v_\Sigma(\alpha) = 0$ or $v_\Sigma(\beta) = 0$; in both cases, we can make $v_\Sigma(\neg(\alpha \wedge \beta)) = 1$.

The other four cases of $\alpha \vee \beta$, $\neg(\alpha \vee \beta)$, $\alpha \to \beta$ and $\neg(\alpha \to \beta)$ are totally analogous, and are thus omitted.

We have thus creates a valuation v_Σ that satisfies all elements of Γ and falsifies all elements of Δ, so $\Gamma \not\models_\Sigma \Delta$ and we are done. □

4.3. Uniform substitution

It is interesting to compare the inference systems $\vdash_\Sigma$, $\vdash^a_k$ and $\vdash^d_k$ with respect to the property of uniform substitution.

An inference system $\vdash$ has the property of *uniform substitution* if whenever $\Gamma \vdash \Delta$ and we uniformly substitute some atoms in Γ and Δ by any formulas, thus generating Γ' and Δ', we have that $\Gamma' \vdash \Delta'$. That is, inferences are preserved by uniform substitution.

With respect to the inference $\vdash_\Sigma$, it only has a limited form of uniform substitution. If p is an atom to be substituted, it can be substituted by a formula $\alpha \in \Sigma$ iff $p \in \Sigma$. For example, we have seen that, if $\Sigma = \{p\}$, then $p \vee q, q \rightarrow p, p \rightarrow (s \vee t), (p \wedge t) \rightarrow s \vdash_\Sigma s$, but for an $\alpha \notin \Sigma$, we have also seen that $\alpha \vee q, q \rightarrow \alpha, \alpha \rightarrow (s \vee t), (\alpha \wedge t) \rightarrow s \nvdash_\Sigma s$. Conversely, if we make $\Sigma = \{\alpha\}$ the situation reverts, as the first sequent is inferable but not the second.

With respect to $\vdash^a_k$, it is easy to see that it does not possess the uniform substitution property. In fact, it is clear that $p \vee q, q \rightarrow p, p \rightarrow (s \vee t), (p \wedge t) \rightarrow s \vdash^a_1 s$; however, if we substitute p by α, such that atoms has more than one atom, it follows that $\alpha \vee q, q \rightarrow \alpha, \alpha \rightarrow (s \vee t), (\alpha \wedge t) \rightarrow s \nvdash^a_1 s$.

With respect to $\vdash^d_k$, we have the following.

LEMMA 4.1. *The $\vdash^d_k$-inference possesses the uniform substitution property.*

Proof. Consider an inferable sequent $\Gamma \vdash^d_k \Delta$ and let Γ' and Δ' be the result of uniformly substituting p by α. Now consider a derivation of $\Gamma \vdash^d_k \Delta$ in which all occurrences of p are replaced by α. Clearly, this is a classical inference of $\Gamma' \vdash \Delta'$, and since it uses exactly the same number or cut-formulas as in the original proof, it is also an inference of $\Gamma' \vdash^d_k \Delta'$. □

5. Decision Procedures for Cut-and-Pay

We analyse here decision procedures for the three inference systems, $\vdash_\Sigma$, $\vdash^a_k$ and $\vdash^d_k$.

5.1. Tableaux for $\vdash_\Sigma$

A tableau system for $\vdash_\Sigma$ was proposed in Finger (2004a). It is based on Mondadori's and D'Agostino's KE-tableaux (D'Agostino and Mondadori, 1994; D'Agostino, 1999), for this particular form of analytic tableau permits direct control of the cut rule, unlike Smullyan's Semantic Tableaux (Smullyan, 1968), that is based on a cut-free calculus. Furthermore, KE-tableaux have better computational properties than semantic tableaux (D'Agostino, 1992). Since this version of KE-tableaux is based on the Limited Bivalence semantics, it is called a KELB(Σ)-tableaux, parameterised by a set Σ of formulas.

KE-tableaux deal with T- and F-signed formulas. So if α is a formula, $T\ \alpha$ and $F\ \alpha$ are signed formulas. $T\ \alpha$ is the *conjugate formula* of $F\ \alpha$, and vice versa. Each connective is associated with a set of *linear expansion rules*. Linear expansion rules always have a *main premiss*; two-premissed rules also have an *auxiliary premiss*.

$$\frac{\begin{array}{c}T\ \alpha \to \beta\\ T\ \alpha\end{array}}{T\ \beta}\,(T\to_1) \qquad \frac{\begin{array}{c}T\ \alpha \to \beta\\ F\ \beta\end{array}}{F\ \alpha}\,(T\to_2) \qquad \frac{F\ \alpha \to \beta}{\begin{array}{c}T\ \alpha\\ F\ \beta\end{array}}\,(F\to)$$

$$\frac{\begin{array}{c}F\ \alpha \wedge \beta\\ T\ \alpha\end{array}}{F\ \beta}\,(F\wedge_1) \qquad \frac{\begin{array}{c}F\ \alpha \wedge \beta\\ T\ \beta\end{array}}{F\ \alpha}\,(F\wedge_2) \qquad \frac{T\ \alpha \wedge \beta}{\begin{array}{c}T\ \alpha\\ T\ \beta\end{array}}\,(T\wedge)$$

$$\frac{\begin{array}{c}T\ \alpha \vee \beta\\ F\ \alpha\end{array}}{T\ \beta}\,(T\vee_1) \qquad \frac{\begin{array}{c}T\ \alpha \vee \beta\\ F\ \beta\end{array}}{T\ \alpha}\,(T\vee_2) \qquad \frac{F\ \alpha \vee \beta}{\begin{array}{c}F\ \alpha\\ F\ \beta\end{array}}\,(F\vee)$$

$$\frac{T\ \neg\alpha}{F\ \alpha}\,(T\neg) \qquad \frac{F\ \neg\alpha}{T\ \alpha}\,(F\neg)$$

Figure 6. KE expansion rules.

$$\begin{array}{ccc} & \alpha \in \Sigma & \\ & / \quad \backslash & \\ T\ \alpha & & F\ \alpha \end{array}$$

Figure 7. Limited principle of bivalence LPB(Σ).

Figure 6 shows KE-tableau linear connective expansion rules for classical logic, which are the same for KELB-tableaux.

The only branching rule in KE is the *Principle of Bivalence*, stating that a formula α must be either true or false. In KELB(Σ)-tableaux, this rule is *limited* by a proviso stating that it can only occur over a formula $\alpha \in \Sigma$. This *limited principle of bivalence*, LPB(Σ) is illustrated in Figure 7.

The idempotency mentioned above is also present in KELB-tableaux, imposing some further linear rules which are redundant in classical KE:

$$\frac{F\ \alpha \wedge \alpha}{F\ \alpha}\,(F\wedge_{\alpha\alpha}) \qquad \frac{T\ \alpha \vee \alpha}{T\ \alpha}\,(T\vee_{\alpha\alpha})$$

An expansion of a tableau branch is allowed when the premisses of an expansion rule are present in the branch; the expansion consists of adding the conclusions of the rule to the end of all branches passing through the set of all premisses of that rule. The LPB(Σ) branching rule splits a branch into two.

A branch in a KELB-tableau is *closed* if it contains $F\ \alpha$ and $T\ \alpha$. The tableau is closed if all its branches are closed. We define the inference $\vdash_{\Sigma}^{\mathsf{KELB}}$ such that $\alpha_1, \ldots, \alpha_n \vdash_{\Sigma}^{\mathsf{KELB}} \beta$ iff there is a closed KELB(Σ)-tableau for $T\ \alpha_1, \ldots, T\ \alpha_n, F\ \beta$.

It is worth noting that KELB-tableaux are incremental, in the following sense. To prove (classically) that $\Gamma \vdash \alpha$, we start a KELB(Σ)-tableau with $\Sigma = \varnothing$. We then proceed with the linear expansion rules until the tableau either closes or is blocked due to the impossibility of applying the PB rule. If the former is the case, the tableau is closed and we are done. Otherwise, we have to chose a formula β over which to branch the tableau, and we make $\Sigma' = \{\beta\}$ and proceed with the linear expansion of both branches; note that by expanding Σ we are in fact changing the logic, and advancing in the approximation process. If we get another blocked branch, the process of Σ-expansion is repeated, until eventually we get a saturated open branch (in which case $\Gamma \not\vdash \alpha$), or all branches are closed (in which case $\Gamma \vdash \alpha$). That is, the KELB-tableau proof provides a heuristic for accomplishing an approximation process.

THEOREM 5.1. $\Gamma \vdash_\Sigma \Delta$ *iff* $\Gamma \vdash_\Sigma^{\mathsf{KELB}} \Delta$.

Proof. There are two ways to show the result. The first is to show directly how the sequent rules of $\vdash_\Sigma$ simulate the rules in $\vdash_\Sigma^{\mathsf{KELB}}$ and vice versa.

However, a shorter way is possible, via semantics, for it has been shown in Finger (2004a) that $\Gamma \vdash_\Sigma^{\mathsf{KELB}} \Delta$ iff $\Gamma \models_\Sigma \Delta$.[2] Then by Theorem 4.1(i) we obtain the result. □

It is worth noting that there is a close correlation between sequent and tableau proofs. A branch in a sequent proof corresponds to a branch in a tableau proof. An application of a connective sequent formula in Figure 2 corresponds to an application of a tableau rule in Figure 6. An application of a structural rule in Figure 1 does not correspond to any rule in the tableau since tableau branches are treated as sets of formulas. A cut formula in a sequent proof corresponds in a tableau proof to a formula over which the principle of bivalence is applied.

The Complexity of $\vdash_\Sigma$

We now study the complexity of $\vdash_\Sigma$ via KELB(Σ)-tableaux *for a fixed set of formulas* Σ. By a *symbol* in a formula α we mean an atom or a connective, and we define $|\alpha|$ as the number of symbol occurrences in α. We define $|\Gamma| = \sum_{\alpha_i \in \Gamma} |\alpha_i|$.

Given a sequent $\Gamma \vdash \Delta$, let N be the size of the set of atoms occurring in Γ and Δ and let $L = |\Gamma| + |\Delta|$. Recall that $|\Sigma|$ is the cardinality of the basis of a finitely generated Σ.

THEOREM 5.2. *There exists a decision procedure for* $\Gamma \vdash_\Sigma \Delta$ *that runs in time* $O(L \times 2^{|\Sigma|})$.

Proof. Consider a tableau for $\Gamma \vdash_\Sigma \Delta$ that respects the normalisation of Lemmas 2.4 and 2.5. Then a branch does not contain a cut formula (or a PB formula) and its negation, so a branch contains at most $|\Sigma|$ cut formulas, which implies that there are at most $2^{|\Sigma|}$ possible branches. As the number of formulas in a tableau branch is at most L, we have that the proof size is at most $L \times 2^{|\Sigma|}$ as desired. □

[2] It is worth noting that in the setting of Finger (2004a), the set Σ was closed under formula formation; however, that restriction was used for other purposes an does not play a part in the soundness/completeness of the system.

Note that, in a given logic, the set Σ is fixed, so that the the term $2^{|\Sigma|}$ can be considered a constant, so that the decision of $\vdash_{\Sigma}$, *for a fixed parameter set* Σ, is linear with the size of the sequent. This is similar to the complexity result for clausal approximations obtained by Schaerf and Cadoli (1995).

Also note that since every branch can be explored independently of the others, and a branch contains most L formulas, each of which with size at most L, then the proof can be done in $O(L^2)$-space.

We also note that the approximation process can be done *incrementally*, in a process that has been described in detail in Finger and Wassermann (2004) and Finger (2004a). Suppose we are build an open tableau for $\Gamma \vdash_{\Sigma} \Delta$, and it remains open due to the impossibility to apply the Principle of Bivalence (which corresponds to a cut in a sequent proof) due to the fact that a given formula $\alpha \notin \Sigma$. Then we can expand Σ into $\Sigma' = \Sigma \cup \{\alpha\}$, which means that we are moving to another logic in an approximation process. So the open branches provide a heuristics to which formula to be added to Σ. Furthermore, the proof proceeds incrementally, for the proof with parameter set Σ' can continue from the point that the proof with parameter set Σ stopped. Classical logic is reached when all formulas is Γ and Δ have been inserted in Σ, and in classical logic we can always prove or refute a sequent by branching only over the subformulas of the original sequent (i.e. all cuts are analytic) so the whole approximation process can be done in time $O(L \times 2^L)$.

5.2. A Decision Procedure for $\vdash_k^a$

We analyse the complexity of $\vdash_k^a$-deductions *for a fixed value of* k. Recall that $\vdash_k^a$ can be reduced to $\vdash_{\Sigma}$ in the following way. Let $\Sigma_k = \{A|\ |\text{atoms}(A)| \leq k\}$, then $\vdash_k^a = \vdash_{\Sigma_k}$.

However, to decide whether $\Gamma \vdash_k^a \Delta$ we can avoid the direct proof of $\Gamma \vdash_{\Sigma_k} \Delta$. First, we can limit the proof to contain analytic cuts only, which is equivalent to limiting the application of tableau branches over subformulas of Γ and Δ; this limitation implies that all atoms considered will be included in the original sequent, so $k \leq N$.

As an auxiliary step in the study of $\vdash_k^a$, define the set Θ_k of $\vdash_{\varnothing}$-provable formulas in the context of the original sequent $\Gamma \vdash \Delta$ using as auxiliary provable formulas of at most k-atoms. This is done by a double induction, as follows:

$$\Theta_1 = \{A \in \Sigma_1 \mid \Gamma \vdash_{\varnothing} \Delta, A\}$$

$$\Theta_{k+1} = \bigcup_j \Theta_{k+1}^j \text{ where } \Theta_{k+1}^j \text{ is inductively defined by:}$$

$$\Theta_{k+1}^0 = \Theta_k^j$$

$$\Theta_{k+1}^{j+1} = \left\{A \in \Sigma_{k+1} \,\middle|\, \Gamma, \Theta_{k+1}^j \vdash_{\varnothing} \Delta, A\right\}$$

Since Σ_k is the set of all formulas with at most k atoms, Θ_{k+1}^j is the subset of Σ_k formulas that are provable from Θ_k in the context of $\Gamma \vdash \Delta$ with the use of j-cuts.

Furthermore, Θ_{k+1} is the fixed point of the equation $X = \{A \in \Sigma_k | \Gamma, \Theta_k, X \vdash_\varnothing \Delta, A\}$, which means that it can be reached in a finite number of steps; in fact, as there are at most L elements in Σ_k, this fixed point is reached in at most L steps. Clearly, $\Theta_k \subseteq \Theta_{k+1}$.

The relationship between $\vdash^a_k$ and Θ_{k+1} is given by the following.

LEMMA 5.1. *Let $A \in \Sigma_{k+1}$. If $\Gamma \vdash^a_k \Delta, A$ then $A \in \Theta_{k+1}$.*

Proof. By induction on the number of cuts used in the proof of $\Gamma \vdash^a_k \Delta, A$. If no cuts are used, then $\Gamma \vdash_\varnothing \Delta, A$ and clearly $A \in \Theta_{k+1}$. Otherwise, suppose that the last cut formula on the proof of $\Gamma \vdash^a_k \Delta, A$ is $B \in \Sigma_k$, such that we have the following:

$$\cfrac{\Gamma_1 \vdash^a_k \Delta_1, B \qquad \Gamma_2, B \vdash^a_k \Delta_2}{\Gamma_1, \Gamma_2 \vdash^a_k \Delta_1, \Delta_2}\ (\text{Cut})$$
$$\vdots\}\Pi$$
$$\Gamma \vdash^a_k \Delta, A$$

From $\Gamma_1 \vdash^a_k \Delta_1, B$ and the induction hypothesis, we obtain that $B \in \Theta'_{k+1}$, where Θ'_{k+1} is Θ_{k+1} restricted to the subformulas of Γ_1 and Δ_1. Thus, $B \in \Theta_{k+1}$, that is, there is a positive integer $j \leq L$ such that $B \in \Theta^j_{k+1}$. It follow that $A \in \Theta^{j+1}_{k+1}$ and so $A \in \Theta_{k+1}$. □

A second result of the construction, now on the semantic side is the following.

LEMMA 5.2. *Suppose $k \geq 1$, let $A \in \Theta_k$ and let v_Σ be a valuation such that $v_\Sigma(\Gamma) = 1$ and $v_\Sigma(\Delta) = 0$. Then $v_\Sigma(A) \neq 0$.*

Proof. We prove by induction on k. For $k = 1$, we have that $\Gamma \vdash_\varnothing \Delta, A$; by completeness, if $v_\Sigma(\Gamma) = 1$ and $v_\Sigma(\Delta) = 0$ then necessarily $v_\Sigma(A) \neq 0$.

For $k > 1$, we have that $\Gamma, \Theta^j_k \vdash_\varnothing \Delta, A$ for some j. By completeness, if $v_\Sigma(\Gamma) = 1$, $v_\Sigma(\Theta^j_k)$ and $v_\Sigma(\Delta) = 0$ then necessarily $v_\Sigma(A) \neq 0$. □

We can now prove the translation of $\vdash^a_k$ in terms of $\vdash_\varnothing$.

THEOREM 5.3. *$\Gamma \vdash^a_k \Delta$ iff $\Gamma, \Theta_{k+1} \vdash_\varnothing \Delta$, for $k \geq 0$.*

Proof. ($\Rightarrow$) By induction on the number c of cuts in the proof of $\Gamma \vdash^a_k \Delta$. If $c = 0$, $\vdash^a_k = \vdash_\varnothing$, so the result is trivial.

If $c \geq 1$, let A be the last cut formula in the proof of $\Gamma \vdash^a_k \Delta$:

$$\cfrac{\Gamma_1 \vdash^a_k \Delta_1, A \qquad \Gamma_2, A \vdash^a_k \Delta_2}{\Gamma_1, \Gamma_2 \vdash^a_k \Delta_1, \Delta_2}\ (\text{Cut})$$
$$\vdots\}\Pi$$
$$\Gamma \vdash^a_k \Delta$$

From $\Gamma_1 \vdash_k^a \Delta_1, A$ and Lemma 5.1 we obtain that $A \in \Theta_{k+1}$. From $\Gamma_2, A \vdash_k^a \Delta_2$ and the induction hypothesis it follows that $\Gamma_2, A, \Theta_{k+1} \vdash_\varnothing \Delta_2$ and since $A \in \Theta_{k+1}$, the contraction rule gives us $\Gamma_2, \Theta_{k+1} \vdash_\varnothing \Delta_2$. By monotonicity, $\Gamma_1, \Gamma_2, \Theta_{k+1} \vdash_\varnothing \Delta_1, \Delta_2$ and by applying the proof steps in Π we finally obtain $\Gamma, \Theta_{k+1} \vdash_\varnothing \Delta$, as desired.

($\Leftarrow$) Assume that $\Gamma \not\vdash_k^a \Delta$. Recall that $\vdash_k^a = \vdash_{\Sigma_k}$, so $\Gamma \not\vdash_{\Sigma_k} \Delta$. By soundness, the latter implies that there exists a valuation v_{Σ_k} such that $v_{\Sigma_k}(\Gamma) = 1$ and $v_{\Sigma_k}(\Delta) = 0$. As $\Theta_k \subseteq \Sigma_k$, Lemma 5.2 yields $v_{\Sigma_k}(\Theta_k) = 1$. This all implies that $\Gamma, \Theta_k \not\models_\varnothing \Delta$, which by completeness implies that $\Gamma, \Theta_k \not\vdash_\varnothing \Delta$, thus finishing the proof. □

The result above suggests an algorithm for deciding $\vdash_k^a$, which is presented in Algorithm 5.1.

Computing the sizes of the sets Θ_k and Σ_k is a hard task. If we consider, for example, the sets Θ_1 and Σ_1, these sets can vary from containing just a few atoms to containing all the formulas and subformulas of Γ and Δ, that is, it is possible that a large sequent may contain in theory just a single atom. Owing to that, an exact description of the time complexity $\vdash_k^a$ will not be presented, as the only limit imposed is $O(L \times 2^L)$, independently of the value of k and N. We point, however, to two very similar systems. Dalal's $\vdash_k^{\mathrm{BCP}}$-inference (Dalal, 1996a), which can be

```
Input: Sequences Γ and Δ, and k ≥ 0.
Output: true if Γ ⊢ₖᵃ Δ; false otherwise.
  Let Λ := Subf(Γ, Δ) ∪ {¬A|A ∈ Subf(Γ, Δ)}.
    {Compute Θₖ}
  Let Θ₀ := ∅.
  for i = 1 to k do
    Θᵢ := Θᵢ₋₁
    changed := true
    while changed do
      changed := false
      for all A ∈ (Λ ∩ Σᵢ) − Θᵢ do
        if Γ, Θᵢ ⊢∅ Δ, A then
          Θᵢ := Θᵢ ∪ {A}
          changed := true
        end if
      end for
    end while
  end for
    {Final decision based on Theorem 5.3}
  if Γ, Θₖ ⊢∅ Δ then
    return true
  else
    return false
  end if
```

Algorithm 5.1. Decision procedure for $\vdash_k^a$.

Input: Sequences Γ and Δ, and $k \geq 0$.
Output: **true** if $\Gamma \vdash_k^d \Delta$; **false** otherwise.
1: Let $\Lambda := \text{Subf}(\Gamma, \Delta)$.
2: **for all** $\Sigma \subseteq \Lambda$ such that $|\Sigma| = k$ **do**
3:    **if** $\Gamma \vdash_\Sigma \Delta$ **then**
4:       **return true**;
5:    **end if**
6: **end for**
7: **return false**;

Algorithm 5.2. Decision procedure for $\vdash_k^d$.

seen as a restriction of $\vdash_k^a$ to Horn clauses, has polynomial time complexity in L, for a fixed k. Similarly, the inference $\vdash_k^{\mathsf{KELB}}$ is a less expressive version of $\vdash_k^a$, where each proof can have cuts over formulas that are all based on a single set of k-atoms, whose complexity is also polynomial time in L for a fixed k.

5.3. A Decision Procedure for $\vdash_k^d$

We explore here an algorithm that decide $\vdash_k^d$-inferences and computes its complexity. This procedure is defined in terms of the $\vdash_\Sigma$-inference.

Recall that $\vdash_k^d$ can be expressed in terms of $\vdash_\Sigma$ as:

$$\vdash_k^d = \bigcup_{|\Sigma| \leq k} \vdash_\Sigma .$$

This idea is translated into Algorithm 5.2 for deciding $\vdash_k^d$.

THEOREM 5.4. *Algorithm 5.2 decides* $\Gamma \vdash_k^d \Gamma$ *in time* $O(2^k \times L \times \binom{L}{k})$.

Proof. The correctness of Algorithm 5.2 comes directly form the definition of $\vdash_k^d$ in terms of a union of $\vdash_\Sigma$, as above.

We see that the loop between lines 2 and 6 can be executed once for every subset of Λ of size k. As $|\Lambda| = L$, this execution is done in the worse case $\binom{L}{k}$ times. For each such iteration, the test in line 3 is executed, which according to Theorem 5.2, has complexity $O(L \times 2^{|\Sigma|})$.

Therefore, Algorithm 5.2 has worst-case complexity of $O(2^k \times L \times \binom{L}{k})$. □

Note that $O(\binom{L}{k}) = O(L^k)$, so for a fixed k, Theorem 5.4 implies a polynomial time complexity of $O(2^k \times L^{k+1})$ for each k. This does not mean, of course, that the classical inference $\Gamma \vdash \Delta$ can be decided in polynomial time. In fact, if we decide $\Gamma \vdash \Delta$ by performing an *iterated depth search* using Algorithm 5.2, in which k varies from 0 to a maximum L. In this way, we end up with an algorithm of time complexity:

$$L \times \sum_{k=0}^{L} \binom{L}{k} \times 2^k = L \times (1+2)^L = L \times 3^L.$$

So each approximation step is done in polynomial time with respect to L, but the whole approximation process is exponential, as one would expect.

6. Conclusions

We have described in this paper families of resource aware systems that control the use of the cut inference rule. We have proposed three families of parameterised logics – $\vdash_\Sigma$, $\vdash_k^a$ $\vdash_k^d$ – and that limit the use of the cut rule, all of which perform an approximation process when the parameter is increased. The original presentation was proof theoretical in terms a Gentzen system presentation. A sound and complete semantics was presented for each element of those families of logics. A KE-tableau decision procedure was then presented for each family of logics.

The main results of these paper are that:

- The elements of the parameterised families $\vdash_\Sigma$ and $\vdash_k^d$ are decidable in polynomial time, but the whole approximation process remains exponential.
- The family $\vdash_k^d$ has the substitution property, and this is the first time there is an approximation family for full propositional logic which enjoys this basic matalevel property of logical systems.

Many other families of cut-controlling logics are possible. In particular, on can try to "fix" the fact that the component logics of family $\vdash_k^a$ cannot be decided in polynomial time. The family $\vdash_k^a$ is a direct generalisation of Dalal's clausal approximations of classical logic, which also controls the number of atoms over cut formulas. However, Dalal (1996a,b) considers a clause to be a *set* of literals, in which case there are no repetition of literals in a clause, and also ruling out clauses containing p and $\neg p$. So the control of the *number* of atoms may not be a direct generalisation of Dalal's approach. A finer control may be achieved if the *number of atom occurrences* in a cut formula, in which case a cut over p would be distinguished from a cut over $p \wedge p$.

Another interesting problem would be to investigate if this approach of polynomial approximations can be applied to logics whose decision procedure are above NP-complete. For instance, most modal and temporal logics (such as K, T, S4, LTL, CTL, etc) and Quantified Boolean Logics, are PSPACE-complete; somewhere in between, within the polynomial hierarchy, there are the non-monotonic semantics of classical logics, such as the Gelfond-Lifschitz stable models and well-founded semantics.

Finally, the practical applications of approximated logics must be investigated. Can practical problems be reasonably modelled using a single polynomial sub-classical logic in the family proposed here? These are all interesting research paths for the future.

References

Anderson, A.R. and Belnap, N.D. Jr., 1975, *Entailment: The Logic of Relevance and Necessity*, Vol. 1. Princeton University Press.

Boolos, G., 1984, "Don't eliminate cut," *Journal of Philosophical Logic* **13**, 373–378.

Carbone, A. and Semmes, S., 2000, *A Graphic Apology for Symmetry and Implicitness*, Oxford Mathematical Monographs: Oxford University Press.

D'Agostino, M., 1992, "Are tableaux an improvement on truth-tables? – Cut-free proofs and bivalence," *Journal of Logic Language and Information* **1**, 235–252.

D'Agostino, M., 1999, "Tableau methods for classical propositional logic," pp. 45–124 in *Handbook of Tableau Methods*, Marcello D'Agostino, Dov Gabbay, Rainer Haehnle, and Joachim Posegga, eds., Kluwer.

Dalal, M., 1996a, "Anytime families of tractable propositional reasoners," pp. 42–45 in *International Symposium of Artificial Intelligence and Mathematics AI/MATH-96*.

Dalal, M., 1996b, "Semantics of an anytime family of reasoners," pp. 360–364 in *12th European Conference on Artificial Intelligence*.

da Costa, N.C.A., 1974, "On the theory of inconsistent formal systems," *Notre Dame Journal of Formal Logic* **15**(4), 497–510.

D'Agostino, M. and Mondadori, M., 1994, "The taming of the cut. Classical refutations with analytic cut," *Journal of Logic and Computation* **4**, 285–319.

Finger, M., 2004a, "Polynomial approximations of full propositional logic via limited bivalence," pp. 526–538 in *9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, Vol. 3229 of *Lecture Notes in Artificial Intellingence (LNAI)*, Lisbon, Portugal, Springer.

Finger, M., 2004b, "Towards polynomial approximations of full propositional logic," pp. 11–20 in *XVII Brazilian Symposium on Artificial Intelligence (SBIA 2004)*, Ana L. C. Bazzan and Sofiane Labidi, eds., Vol. 3171 of *Lecture Notes in Artificial Intellingence (LNAI)*, Springer.

Finger, M. and Wassermann, R., 2004, "Approximate and limited reasoning: Semantics, proof theory, expressivity and control," *Journal of Logic And Computation* **14**(2), 179–204.

Girard, J.-Y., 1987, "Linear logic," *Theoretical Computer Science* **50**, 1–102.

Massacci, F., 1998, "Anytime approximate modal reasoning," pp. 274–279 in *AAAI-98*, Jack Mostow and Charles Rich, eds., AAAIP.

Restall, G., 2000, *An Introduction to Substructural Logics*. Routledge.

Schaerf, M. and Cadoli, M., 1995, "Tractable reasoning via approximation," *Artificial Intelligence* **74**(2), 249–310.

Smullyan, R.M., 1968, *First-Order Logic*, Springer-Verlag.

Van Dalen, D., 1984, "Intuitionistic logic," in *Handbook of Philosophical Logic*, D. Gabbay and F. Guenthner, eds., Vol. III.