The complexity of disjunction in intuitionistic logic

R. Ramanujam
The Institute of Mathematical Sciences, Chennai, India

Vaishnavi Sundararajan[1]
Chennai Mathematical Institute, Chennai, India

S. P. Suresh[2]
Chennai Mathematical Institute, Chennai, India

Abstract

We study procedures for the derivability problem of fragments of intuitionistic logic. Intuitionistic logic is known to be PSPACE-complete, with implication being one of the main contributors to this complexity. In fact, with just implication alone, we still have a PSPACE-complete logic. We study fragments of intuitionistic logic with restricted implication, and develop algorithms for these fragments which are based on the proof rules. We identify a core fragment whose derivability is solvable in linear time. Adding disjunction elimination to this core gives a logic which is solvable in co-NP. These sub-procedures are applicable to a wide variety of logics with rules of a similar flavour. We also show that we cannot do better than co-NP whenever disjunction elimination interacts with other rules.

2012 ACM Subject Classification Theory of computation → Proof theory

Keywords and phrases Intuitionistic logic, fragments, proof theory, disjunction, complexity, linear time algorithm, co-NP procedure

1 Introduction

Intuitionistic logic is well-known to be a PSPACE-complete logic [17]. One of the main contributors to this complexity is implication, particularly the implication introduction rule. The very form of the rule, displayed below, provides a hint as to the difficulties involved.

$$\frac{X, \alpha \vdash \beta}{X \vdash \alpha \to \beta} \to i$$

Suppose we try to determine if a set of formulas X derives α. If α were derivable only using rules for conjunction, say, then we could compute the "closure" of X by repeatedly adding to it all formulas derivable in one step, and see if we ever reach α this way. The point to note here is that the "context", X, is fixed. But the moment the proof involves the $\to i$ rule, we cannot work with a fixed context anymore. To verify whether X derives $\alpha \to \beta$, we have to change context, and verify if $X \cup \{\alpha\}$ derives β. This "context switch" is at the heart of the complexity of intuitionistic logic.

Another rule which forces this kind of context switch in intuitionistic logic is disjunction elimination, displayed below.

$$\frac{X \vdash \alpha \vee \beta \quad X, \alpha \vdash \gamma \quad X, \beta \vdash \gamma}{X \vdash \gamma} \vee e$$

[1] Supported by a TCS Research Fellowship, and partially by a grant from the Infosys Foundation.
[2] Partially supported by a grant from the Infosys Foundation.

To verify if X derives γ when we already know that it derives $\alpha \vee \beta$, we might have to check if γ is derived both from $X \cup \{\alpha\}$ and $X \cup \{\beta\}$.

An interesting question to consider is whether these two rules have similar contributions to the complexity of derivability. Since implication by itself is PSPACE-complete [17], we need to study disjunction in the presence of restricted forms of implication to address this question. We are able to show that disjunction by itself is solvable in PTIME, and in the presence of conjunction and restricted forms of negation and implication, it is solvable in co-NP. We also show that we can do no better than co-NP, even for minimal fragments involving disjunction.

The decision procedure is interesting in its own right. We identify a core fragment that does not involve context switches of the kind alluded to above. We solve this fragment in PTIME (linear time, in fact) in Section 3, and show how to extend this to a co-NP procedure when we add disjunction elimination (in Section 4). This offers a source of parametrization, and we obtain a PTIME decision procedure when we bound the set of formulas on which we apply disjunction elimination (in Section 4.2).

We have already mentioned that we study disjunction in the presence of restricted forms of implication and negation. What sort of restrictions are natural to impose? We have seen that the problematic thing about implication is that the $\rightarrow i$ rule forces a context switch. So we could restrict intuitionistic logic by only considering the elimination rule for implication[3], while leaving out the introduction rule. We can do a little better, by considering a variant known as *primal implication* [11] (also referred to as *semi-implication* [2]). This is defined by the following introduction rule.

$$\frac{X \vdash \beta}{X \vdash \alpha \rightarrow \beta} \rightarrow_p$$

Primal implication defines a strict sublogic of intuitionistic logic. For instance, intuitionistic validities like $p \rightarrow p$ are not provable using the above rule. But more importantly, the rule is so designed that no context switch is forced while trying to check if X derives α. In fact, this feature is crucial to the PTIME solvability of *primal infon logic* [11], which is a logic with primal implication, conjunction, and some modalities.

It is worth noting that we consider the complexity of the *derivability problem*, and not the *validity problem* as is usually done. While these two problems are equivalent in the presence of the $\rightarrow i$ rule, for the restricted fragments that we consider, derivability is more general.

For precisely the same reason (that we consider fragments without $\rightarrow i$), we cannot define $\neg\alpha$ as $\alpha \rightarrow \bot$. We need to have explicit rules for negation. Of these, the elimination rule for negation does not force a context switch, but the introduction rule does, just as in the case of implication. The precise effect of negation introduction in the absence of $\rightarrow i$ on complexity is unclear, so we only consider restrictions where we drop negation introduction altogether.

[3] It is clear that modus ponens does not force context switches of the kind we are discussing.

Applications

Intuitionistic logic is a subject with a rich history, with connections to fundamental aspects of mathematics, philosophy and computer science, but it also finds application in such concrete areas of computer science as system security and communication security in distributed protocols. Consider the derivability question: given a finite set of formulas X, a formula α, does X derive α? This question is of practical importance when X is a security policy that specifies permissions and α is the assertion of someone being permitted some action [1, 10]. Or it might be the case that X is a set of terms picked by an eavesdropper watching a channel and α is a term to be kept secret [8]. Inference in such situations is typically intuitionistic. This is essentially due to the fact that terms are constructed in cryptographic protocols using encryption keys etc, and when a term t is not constructible by an agent A, then A cannot even assert "(m occurs in t) $\vee \neg$(m occurs in t)." For examples of such reasoning, see [15].

In the applications mentioned above, the complexity of derivability is of prime importance, since a derivability check is often a vital component of more detailed security structures [6]. These systems are usually disjunction-free, with a PTIME derivability procedure [3, 7, 11]. But reasoning about disjunction is also important for security applications, even though it typically increases the complexity of the derivability problem (see [15], for example).

Related work

The results reported in this paper are very close to work done in the realm of authorization logics, specifically primal infon logic and its extensions. It was shown that primal infon logic is in PTIME [3, 11] but adding disjunction makes the problem co-NP-complete [4]. Specifically, it was shown that a system with primal implication, conjunction, disjunction and $\bot$ is co-NP-hard, using a translation from classical logic. Our lower bound results can be seen as a refinement of the result in [4], as we show that disjunction with *any one* of these other connectives is already co-NP-hard. The upper bound results are also very similar to those in [4], but we provide explicit algorithms while the results there are obtained via a translation to classical logic. Our procedures can be seen as a way of lifting PTIME decision procedures for *local theories* [7, 14] to co-NP procedures for the same logics with disjunction. More recently, the complexity of primal logic with disjunction was studied in further detail in [13], but the proofs are via semantic methods.

Another important area of study is the *disjunction property* and its effect on complexity. A system is said to have the disjunction property if it satisfies the following condition: whenever $X \vdash \alpha \vee \beta$ and X satisfies some extra conditions (for example, $\vee$ does not occur in any formula of X), then $X \vdash \alpha$ or $X \vdash \beta$. The disjunction property and its effect on decidability and complexity have been the subject of study for many years. For example, it has been proved that as long as any (propositional) extension of intuitionistic logic satisfies the disjunction property, derivability is PSPACE-hard, while otherwise it is in co-NP (see Chapter 18 of [5]). Various other papers also investigate extensions of intuitionistic logic with the disjunction property [9, 12, 16]. In contrast to these results, our paper considers *subsystems* of intuitionistic logic obtained by restricting implication. Further, in our paper, the focus is more on the *left disjunction property*: namely that $X, \alpha \vee \beta \vdash \gamma$ iff $X, \alpha \vdash \gamma$ and $X, \beta \vdash \gamma$.

2 Preliminaries

Assume a countably infinite set of atomic propositions $\mathscr{P}$. The set of formulas Φ is given by

$$\alpha, \beta ::= p \mid \neg\alpha \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \alpha \to \beta$$

For a set of operators $\mathscr{O}$, we denote by $\Phi^{\mathscr{O}}$ the set of all formulas consisting only of the operators in $\mathscr{O}$. For example, $\Phi^{\{\vee\}}$ is the set of all formulas built only using the $\vee$ operator, $\Phi^{\{\vee,\wedge\}}$ is the set of all formulas built only using the $\vee$ and $\wedge$ operators, &c. For ease of notation, we ignore the braces and instead use $\Phi^{\vee}$, $\Phi^{\vee,\wedge}$, &c.

The set of *subformulas* of a formula α, denoted $\mathsf{sf}(\alpha)$, is defined to be the smallest set S such that: $\alpha \in S$; if $\neg\beta \in S$, $\beta \in S$; and if $\beta \wedge \gamma \in S$ or $\beta \vee \gamma \in S$ or $\beta \to \gamma \in S$, $\{\beta, \gamma\} \subseteq S$. For a set X of formulas, $\mathsf{sf}(X)$ is defined to be $\bigcup\limits_{\alpha \in X} \mathsf{sf}(\alpha)$.

$\dfrac{}{X, \alpha \vdash \alpha}\,ax$	
$\dfrac{X, \alpha \vdash \beta \quad X, \alpha \vdash \neg\beta}{X \vdash \neg\alpha}\,\neg i$	$\dfrac{X \vdash \beta \quad X \vdash \neg\beta}{X \vdash \alpha}\,\neg e$
$\dfrac{X \vdash \alpha \quad X \vdash \beta}{X \vdash \alpha \wedge \beta}\,\wedge i$	$\dfrac{X \vdash \alpha_0 \wedge \alpha_1}{X \vdash \alpha_j}\,\wedge e$
$\dfrac{X \vdash \alpha_j}{X \vdash \alpha_0 \vee \alpha_1}\,\vee i$	$\dfrac{X \vdash \alpha \vee \beta \quad X, \alpha \vdash \gamma \quad X, \beta \vdash \gamma}{X \vdash \gamma}\,\vee e$
$\dfrac{X, \alpha \vdash \beta}{X \vdash \alpha \to \beta}\,\to i$	$\dfrac{X \vdash \alpha \to \beta \quad X \vdash \alpha}{X \vdash \beta}\,\to e$
$\dfrac{X \vdash \beta}{X \vdash \alpha \to \beta}\,\to_p$	

Figure 1 The system IL. Note that $\to i$ subsumes $\to_p$, but we need to define both these rules since we consider fragments without $\to i$ but with $\to_p$.

The logic IL is defined by the derivation system in Figure 1. Two important fragments of IL are DL (*disjunction logic*), which has all the rules except for $\neg i$ and $\to i$, and CL (*core logic*), which has all the rules except for $\neg i$, $\to i$, and $\vee e$. By $X \vdash_{\mathsf{L}} \alpha$, we mean that there is a derivation of $X \vdash \alpha$ in the subsystem $\mathsf{L} \subseteq \mathsf{IL}$. (For ease of notation, we drop the suffix and use $X \vdash \alpha$ when it is clear from the context which subsystem is being referred to.)

▶ **Definition 1** (Derivability problem). *Given X, α, and a subsystem $L \subseteq IL$, is it the case that $X \vdash_L \alpha$?*

Among the rules, *ax*, $\wedge e$ and $\to e$ are the *pure elimination rules*, $\neg e$, $\neg i$ and $\vee e$ are the *hybrid rules* and the rest are the *pure introduction rules*. A *normal derivation* is one where the major premise of every pure elimination rule and hybrid rule is the conclusion of a pure elimination rule. The following fundamental properties hold, and the proofs are standard in the proof theory literature. Detailed proofs are presented in Appendix A, for ease of reference.

▶ **Proposition 2.**

1. *(Monotonicity) If $X \vdash \alpha$ and $X \subseteq X'$, then $X' \vdash \alpha$.*
2. *(Admissibility of Cut) If $X \vdash \alpha$ and $X, \alpha \vdash \beta$, then $X \vdash \beta$.*

▶ **Theorem 3** (Weak normalization). *If there is a derivation π of $X \vdash \alpha$ then there is a normal derivation ϖ of $X \vdash \alpha$. Further, if a formula $\alpha \vee \beta$ occurs as the major premise of an instance of $\vee e$ in ϖ, it also occurs as the major premise of an instance of $\vee e$ in π.*

▶ **Theorem 4** (Subformula property). *Let π be a normal derivation with conclusion $X \vdash \alpha$ and last rule r. Let $X' \vdash \beta$ occur in π. Then $X' \subseteq sf(X \cup \{\alpha\})$ and $\beta \in sf(X \cup \{\alpha\})$. Furthermore, if r is a pure elimination rule, then $X' \subseteq sf(X)$ and $\beta \in sf(X)$.*

▶ **Proposition 5** (Left Disjunction Property). *$X, \alpha \vee \beta \vdash \gamma$ iff $X, \alpha \vdash \gamma$ and $X, \beta \vdash \gamma$.*

Proof. Suppose $X, \alpha\vee\beta \vdash \gamma$. Note that $X, \alpha \vdash \alpha\vee\beta$ and $X, \beta \vdash \alpha\vee\beta$ (by the $\vee i$ rule). By Admissibility of Cut, we have $X, \alpha \vdash \gamma$ and $X, \beta \vdash \gamma$.

On the other hand, suppose $X, \alpha \vdash \gamma$ and $X, \beta \vdash \gamma$. By Monotonicity, it follows that $X, \alpha\vee\beta, \alpha \vdash \gamma$ and $X, \alpha\vee\beta, \beta \vdash \gamma$. We also have $X, \alpha \vee \beta \vdash \alpha \vee \beta$ (by the *ax* rule). From these, we can build a proof of $X \vdash \gamma$ using the $\vee e$ rule.

◀

▶ **Proposition 6** (Left Conjunction Property). *$X, \alpha \wedge \beta \vdash \gamma$ iff $X, \alpha, \beta \vdash \gamma$.*

Proof. Suppose $X, \alpha\wedge\beta \vdash \gamma$. Note that $X, \alpha, \beta \vdash \alpha\wedge\beta$ (by the $\wedge i$ rule). By Admissibility of Cut, we have $X, \alpha, \beta \vdash \gamma$.

On the other hand, suppose $X, \alpha, \beta \vdash \gamma$. Note that $X, \alpha\wedge\beta \vdash \alpha$ and $X, \alpha\wedge\beta \vdash \beta$ (by the $\wedge e$ rule). By Admissibility of Cut, we have that $X, \alpha\wedge\beta \vdash \gamma$.

◀

3 Linear time algorithm for CL

In this section, we consider CL and solve its derivability problem in linear time. The algorithm is based on the linear time procedure presented in [11], and can be thought of as a core subroutine in the solution of more complex fragments.

▶ **Theorem 7.** *The derivability problem for CL is solvable in linear time.*

Fix a set of formulas $X_0 \cup \{\alpha_0\}$ for the rest of the section. Let $\mathsf{sf} = \mathsf{sf}(X_0 \cup \{\alpha_0\})$. Let $\mathsf{N} = |\mathsf{sf}|$.

▶ **Definition 8.** *For any $X \subseteq sf$:*

- $closure(X) = \{\alpha \in sf \mid X \vdash_{IL} \alpha\}$.
- $derive(X) = \{\alpha \in sf \mid X \vdash_{DL} \alpha\}$.
- $core(X) = \{\alpha \in sf \mid X \vdash_{CL} \alpha\}$.

Checking if $X_0 \vdash_{CL} \alpha_0$ amounts to checking if $\alpha_0 \in core(X_0)$. In the rest of this section, we describe how to compute $core(X)$ for any $X \subseteq \mathsf{sf}$. We compute $core(X)$ by a marking procedure which initially marks all elements of X and propagates the marking in a clever manner. To understand its working, we first consider a naïve strategy for propagating the marking. It would, for example, detect all pairs α, β of marked formulas and mark $\alpha \wedge \beta$, $\alpha \vee \gamma$, $\beta \vee \delta$, &c. (if those formulas are in sf). This propagation step itself is repeated many times till no new formula can be marked. In the course of this, the same formula α may be "touched" many times – in deriving $\alpha \wedge \beta$, $\alpha \vee \gamma$, &c. Our marking proceeds differently. When we "process" a marked α, we mark all of its consequences that we can determine at that stage, and do not process it again. For this to work, we need information about α being already marked when we process some other marked β (to mark $\alpha \wedge \beta$, for instance). Towards this, we maintain some auxiliary lists. For instance, for each formula α there is a list of conjunctions whose left conjunct is α. While processing α, we mark each $\beta = \alpha \wedge \gamma$ in this list such that γ is also marked. We maintain similar lists for other operators and the position of α, as is made clear below.

For a formula α, we define $left(\alpha)$, $right(\alpha)$, and $op(\alpha)$ as follows:

- If $\alpha \in \mathscr{P}$, $left(\alpha)$, $right(\alpha)$ and $op(\alpha)$ are all undefined.
- If $\alpha = \beta \wedge \gamma$, $left(\alpha) = \beta$, $right(\alpha) = \gamma$, and $op(\alpha) = \wedge$.
- If $\alpha = \beta \vee \gamma$, $left(\alpha) = \beta$, $right(\alpha) = \gamma$, and $op(\alpha) = \vee$.
- If $\alpha = \beta \to \gamma$, $left(\alpha) = \beta$, $right(\alpha) = \gamma$, and $op(\alpha) = \to$.
- If $\alpha = \neg\beta$, $left(\alpha) = \beta$, $right(\alpha)$ is undefined, and $op(\alpha) = \neg$.

For every $\alpha \in \mathsf{sf}$, we define the following sets.

- $\mathsf{A}_l(\alpha) = \{\beta \in \mathsf{sf} \mid op(\beta) = \wedge \text{ and } left(\beta) = \alpha\}$.
- $\mathsf{A}_r(\alpha) = \{\beta \in \mathsf{sf} \mid op(\beta) = \wedge \text{ and } right(\beta) = \alpha\}$.
- $\mathsf{O}_l(\alpha) = \{\beta \in \mathsf{sf} \mid op(\beta) = \vee \text{ and } left(\beta) = \alpha\}$.
- $\mathsf{O}_r(\alpha) = \{\beta \in \mathsf{sf} \mid op(\beta) = \vee \text{ and } right(\beta) = \alpha\}$.
- $\mathsf{I}_r(\alpha) = \{\beta \in \mathsf{sf} \mid op(\beta) = \to \text{ and } right(\beta) = \alpha\}$.
- $\mathsf{N}_l(\alpha) = \{\beta \in \mathsf{sf} \mid op(\beta) = \neg \text{ and } left(\beta) = \alpha\}$.

The procedure to compute $core(X)$ is described in Algorithm 1. For each $\alpha \in \mathsf{sf}$ it maintains a variable $status(\alpha) \in \{raw, pending, processed\}$. It also uses a queue Q of formulas, with the corresponding *enqueue* and *dequeue* functions. The correctness of the algorithm is presented below.

▶ **Lemma 9** (Soundness). *If $status(\beta) = processed$, then $\beta \in core(X)$.*

Proof. Initially, the status of every $\beta \in \mathsf{sf}$ is either *raw* or *pending*. It is clear from the code that $status(\beta)$ becomes *processed* only after becoming *pending*. It is also clear that any formula is enqueued only after it becomes *pending*. We prove by induction that if $status(\beta)$ becomes *pending* at any stage of the *while* loop, $\beta \in core(X)$.

The base case is when $status(\beta)$ becomes *pending* before the start of the loop. This means that $\beta \in X$ and hence $\beta \in core(X)$.

Algorithm 1 Linear time algorithm for $core(X)$

```
$Q \leftarrow \emptyset$;
$for\ all\ \alpha \in X : status(\alpha) \leftarrow pending;\ enqueue(Q, \alpha)$;
$for\ all\ \alpha \in sf \setminus X : status(\alpha) \leftarrow raw$;
while $Q \neq \emptyset$ do
    $\alpha \leftarrow dequeue(Q)$;

    if $(op(\alpha) = \wedge)$ & $(status(left(\alpha)) = raw)$ then        ▷ $\alpha$ is premise of $\wedge e$.
        $status(left(\alpha)) \leftarrow pending$;    $enqueue(Q, left(\alpha))$;
    end if
    if $(op(\alpha) = \wedge)$ & $(status(right(\alpha)) = raw)$ then       ▷ $\alpha$ is premise of $\wedge e$.
        $status(right(\alpha)) \leftarrow pending$;    $enqueue(Q, right(\alpha))$;
    end if
    if $\big(op(\alpha) = \rightarrow)$ & $(status(left(\alpha)) \neq raw)$ & $(status(right(\alpha)) = raw\big)$ then
        $status(right(\alpha)) \leftarrow pending$;                  ▷ $\alpha$ is major premise of $\rightarrow e$.
    end if

    if $(\alpha = \neg\gamma)$ & $(status(\gamma) \neq raw)$ then
        $status(\beta) \leftarrow processed$ for all $\beta$; $return\ sf$;      ▷ Everything derivable.
    end if
    if $(\neg\alpha \in sf)$ & $(status(\neg\alpha) \neq raw)$ then
        $status(\beta) \leftarrow processed$ for all $\beta$; $return\ sf$;      ▷ Everything derivable.
    end if

    for all $\beta \in A_l(\alpha)$ s.t $\big(status(right(\beta)) \neq raw\big)$ & $\big(status(\beta) = raw\big)$ do
        $status(\beta) \leftarrow pending$;                  ▷ $\alpha$ is left premise of $\wedge i$.
        $enqueue(Q, \beta)$;
    end for
    for all $\beta \in A_r(\alpha)$ s.t. $\big(status(left(\beta)) \neq raw\big)$ & $\big(status(\beta) = raw\big)$ do
        $status(\beta) \leftarrow pending$;                  ▷ $\alpha$ is right premise of $\wedge i$.
        $enqueue(Q, \beta)$;
    end for
    for all $\beta \in O_l(\alpha)$ s.t. $status(\beta) = raw$ do              ▷ $\alpha$ is premise of $\vee i$.
        $status(\beta) \leftarrow pending$;    $enqueue(Q, \beta)$;
    end for
    for all $\beta \in O_r(\alpha)$ s.t. $status(\beta) = raw$ do              ▷ $\alpha$ is premise of $\vee i$.
        $status(\beta) \leftarrow pending$;    $enqueue(Q, \beta)$;
    end for
    for all $\beta \in I_r(\alpha)$ s.t. $status(\beta) = raw$ do
        $status(\beta) \leftarrow pending$;                  ▷ $\alpha$ is premise of $\rightarrow_p$.
    end for

    $status(\alpha) \leftarrow processed$;
end while
$return\ \{\alpha \in sf \mid status(\alpha) = processed\}$;
```

Suppose *status*(β) becomes *pending* in some iteration of the loop. We consider a few sample cases that might occur.

***status*(β) changes at line 8:** Here $\beta = \mathit{left}(\alpha)$. Since α has been dequeued, *status*(α) became *pending* in an earlier iteration. So by induction hypothesis, $\alpha \in \mathit{core}(X)$. Now, since $\beta = \mathit{left}(\alpha)$, $\alpha = \beta \wedge \gamma$ for some γ. It is clear that β can be derived from α using the $\wedge e$ rule. Thus $\beta \in \mathit{core}(X)$.

***status*(β) changes at line 18:** Here β changes status because $\alpha = \neg\gamma$ and $\mathit{status}(\gamma) \neq \mathit{raw}$. Since α has been dequeued, *status*(α) became *pending* in an earlier iteration. Since $\mathit{status}(\gamma) \neq \mathit{raw}$, *status*($\gamma$) became *pending* at an earlier stage of the *while* loop (in an earlier iteration or earlier in the same iteration). So $\alpha, \gamma \in \mathit{core}(X)$, by induction hypothesis. It is clear that any β can be derived from α and γ using the $\neg e$ rule. Thus $\beta \in \mathit{core}(X)$.

***status*(β) changes at line 25:** Here $\alpha = \mathit{left}(\beta)$. Let $\mathit{right}(\beta) = \gamma$. Since α was dequeued at the start of the iteration, *status*(α) became *pending* in an earlier iteration. We see that $\mathit{status}(\gamma) \neq \mathit{raw}$, and therefore *status*(γ) became *pending* at an earlier stage of the *while* loop (in an earlier iteration or earlier in the same iteration). So by induction hypothesis, $\alpha, \gamma \in \mathit{core}(X)$. It is clear that β can be derived from α and γ using the $\wedge i$ rule. Thus $\beta \in \mathit{core}(X)$.

The other cases proceed similarly, and we have the required claim.

◀

▶ **Lemma 10** (Completeness)**.** *If $\beta \in \mathit{core}(X)$, eventually $\mathit{status}(\beta) = \mathit{processed}$.*

Proof. We first prove by induction on size of proofs that if $\beta \in \mathit{core}(X)$ then eventually *status*(β) becomes *pending* (it is enough to show this – if β ever becomes *pending*, it is enqueued, and upon dequeue it gets assigned the status *processed*).

Suppose π is a proof of $X \vdash \beta$ that does not use the $\vee e$ rule. Let r be the last rule of π. We consider a few sample cases.

r is *ax*: In this case, $\beta \in X$ and *status*(β) becomes *pending* in line 2.

r is $\wedge e$**:** Suppose the premise of r is $X \vdash \beta \wedge \gamma$. Then by induction hypothesis, *status*($\beta \wedge \gamma$) becomes *pending* eventually and enters the queue Q. Since each iteration of the *while* loop terminates, eventually $\beta \wedge \gamma$ is dequeued. In that iteration of the loop, *status*(β) becomes *pending* in line 8.

r is $\neg e$**:** Suppose the premises of r are $X \vdash \alpha$ and $X \vdash \gamma$, where $\alpha = \neg\gamma$. By induction hypothesis, *status*(α) and *status*(γ) become *pending* eventually. Without loss of generality, let α be the second element to be dequeued from Q. When α is dequeued, it is certainly the case that $\mathit{status}(\gamma) \neq \mathit{raw}$. Now one can see that *status*(β) directly becomes *processed* in line 18. (In fact, the status of all formulas becomes *processed* and the algorithm terminates.)

r is $\wedge i$**:** Suppose $\beta = \alpha \wedge \gamma$. Then the premises of r are $X \vdash \alpha$ and $X \vdash \gamma$. By induction hypothesis, *status*(α) and *status*(γ) become *pending* eventually. Without loss of generality, let α be the second element to be dequeued from Q. When α is dequeued, it is certainly the case that $\mathit{status}(\gamma) \neq \mathit{raw}$. Now one can see that *status*(β) becomes *pending* in line 25, when α is being processed.

A similar analysis for the other cases completes the proof.

◀

▶ **Lemma 11** (Running time)**.** *The algorithm terminates in* O(N) *time.*

Proof. Each element enters Q at most once (when its status changes from *raw* to *pending*). We process each element of Q exactly once, after dequeuing it and before marking it *processed*. Processing an element involves setting the status of $left(\alpha)$ and $right(\alpha)$ and perhaps enqueueing them (all these operations take constant time). It also involves going through each element of the five sets $A_l(\alpha)$, $A_r(\alpha)$, $O_l(\alpha)$, $O_r(\alpha)$, and $I_r(\alpha)$. Each of these *for all* loops takes at most $O(N)$ time. So it would appear that the algorithm takes $O(N^2)$ overall. But we need to use the following crucial property. For distinct α and β and $f \in \{A_l, A_r, O_l, O_r, I_r\}$: $f(\alpha) \cap f(\beta) = \emptyset$.

Thus the total time spent processing the sets $A_l(\alpha)$ is $O(N)$ across all $\alpha \in sf$. And similarly for A_r, O_l, &c. From this it follows that the algorithm terminates in $O(N)$ time.

◀

4 Decision procedures for DL

We now solve the derivability problem for the fragment DL, the fragment of IL without $\rightarrow i$ or $\neg i$.[4]

Fix a set of formulas X_0 and a formula α_0 for the rest of the section. Let $sf = sf(X_0 \cup \{\alpha_0\})$ and $N = |sf|$, as in Section 3.

Recall the definitions of $closure(X)$, $derive(X)$, and $core(X)$ from Section 3. In this section, we show how to compute $derive(X)$ for any $X \subseteq sf$. We can then check whether $X_0 \vdash_{DL} \alpha_0$ by checking if $\alpha_0 \in derive(X_0)$. The following properties of *derive* and *core* are useful in this regard.

- $X \subseteq core(X) \subseteq derive(X)$.
- $derive(X) = core(derive(X)) = derive(derive(X))$ (by Admissibility of Cut).
- $core(X) = core(core(X))$ (by Admissibility of Cut).
- If $X = core(Y)$, then $core(X) = X$. If $X = derive(Y)$, then $derive(X) = X$.

4.1 A co-NP procedure for *derive*

Algorithm 2 checks if $X_0 \nvdash \alpha_0$. It uses the notion of a *down-closed set*. A set X of formulas is *down-closed* if it satisfies the following two conditions:

- $core(X) \subseteq X$.
- whenever $\alpha \vee \beta \in X$, then either $\alpha \in X$ or $\beta \in X$.

Y is said to be a *down-closure* of X if Y is down-closed and $X \subseteq Y$.

Algorithm 2 Algorithm to check if $X_0 \nvdash \alpha_0$

```
1: $Y \leftarrow core(X_0)$;
2: **while** ($Y$ is not down-closed) **do**
3:     guess a formula $\beta_0 \vee \beta_1 \in Y$ such that $\beta_0 \notin Y$ and $\beta_1 \notin Y$;
4:     guess $i \in \{0, 1\}$;
5:     $Y \leftarrow core(Y \cup \{\beta_i\})$;
6: **end while**
7: Return "Yes" if $\alpha_0 \notin Y$, and "No" otherwise.
```

[4] It is important to note that we consider only the negation elimination rule. The algorithms in this section do not work in the presence of the $\neg i$ rule. Nor do we know of a straightforward modification to handle the $\neg i$ rule.

In Algorithm 2, it is an invariant that $Y = core(Z)$ for some Z and hence $core(Y) \subseteq Y$. Thus when Y is not down-closed, there exists $\beta_0 \vee \beta_1 \in Y$ such that neither β_0 nor β_1 is in Y.

The algorithm guesses a down-closure Y of X_0 such that $\alpha_0 \notin Y$. The next theorem guarantees that one can successfully guess such a Y iff $X_0 \nvdash \alpha_0$. This ensures the correctness of the algorithm.

▶ **Theorem 12.** *For any X and α (with $X \cup \{\alpha\} \subseteq sf$), $X \vdash \alpha$ iff $\alpha \in Y$ for every down-closure Y of X.*

This theorem is a consequence of the following three lemmas. But first we need a general claim related to the Left Disjunction Property.

▶ **Proposition 13.** *Suppose $\varphi_0 \vee \varphi_1 \in Z$ and $i \in \{0,1\}$. Then $Z \setminus \{\varphi_0 \vee \varphi_1\}, \varphi_i \vdash \theta$ iff $Z, \varphi_i \vdash \theta$.*

▶ **Lemma 14.** *For any X and α (with $X \cup \{\alpha\} \subseteq sf$), $X \vdash \alpha$ iff $Y \vdash \alpha$ for every down-closure Y of X.*

Proof. Suppose $X \vdash \alpha$ and Y is a down-closure of X. Then $X \subseteq Y$ and hence it is immediate that $Y \vdash \alpha$.

Suppose on the other hand that $X \nvdash \alpha$. We show that there is a sequence

$$Y_0 \subsetneq Y_1 \subsetneq \cdots \subsetneq Y_n \subseteq \mathsf{sf}$$

of sets such that

- $X \subseteq Y_0$,
- Y_n is down-closed,
- for all $i \leqslant n$, $core(Y_i) \subseteq Y_i$, and
- for all $i \leqslant n$, $Y_i \nvdash \alpha$.

The sequence is constructed by induction. Y_0 is defined to be $core(X)$. Since $X \nvdash \alpha$, it follows that $Y_0 \nvdash \alpha$. Suppose Y_k has been defined for some $k \geq 0$ such that $Y_k \nvdash \alpha$. If Y_k is down-closed, we are done. Otherwise, since $core(Y_k) \subseteq Y_k$, there is a $\beta_0 \vee \beta_1 \in Y_k$ such that $\beta_0 \notin Y_k$ and $\beta_1 \notin Y_k$. Since $Y_k \nvdash \alpha$, it follows that $Y_k \setminus \{\beta_0 \vee, \beta_1\}, \beta_i \nvdash \alpha$ for some $i \in \{0,1\}$, by the Left Disjunction property. By Claim 13 it follows that $Y_k, \beta_i \nvdash \alpha$ for some $i \in \{0,1\}$.

$$Y_{k+1} = \begin{cases} core(Y_k \cup \{\beta_0\}) & \text{if } Y_k, \beta_0 \nvdash \alpha \\ core(Y_k \cup \{\beta_1\}) & \text{otherwise} \end{cases}$$

Clearly $Y_k \subsetneq Y_{k+1}$ and $core(Y_{k+1}) = Y_{k+1}$. Assume w.l.o.g. that $Y_{k+1} = core(Y_k \cup \{\beta_0\})$. By construction, $Y_k \cup \{\beta_0\} \nvdash \alpha$. Now suppose $Y_{k+1} \vdash \alpha$. Since $Y_k \cup \{\beta_0\} \vdash \varphi$ for every $\varphi \in Y_{k+1}$, it would follow by Admissibilty of Cut that $Y_k \cup \{\beta_0\} \vdash \alpha$, which is a contradiction. Thus $Y_{k+1} \nvdash \alpha$ and we can always extend the sequence as desired.

Further, the Y_i's are strictly increasing, and are all subsets of sf. Thus $n \leqslant |\mathsf{sf}|$ and the above construction terminates. Y_n is a down-closure of X that does not derive α.

◀

▶ **Lemma 15.** *Let π be a proof of $X \vdash \alpha$ with at least one occurrence of the $\vee e$ rule. Then there is an occurrence of $\vee e$ in π with major premise $X \vdash \varphi \vee \psi$ such that $\varphi \vee \psi \in core(X)$.*

Proof. In any proof of the form

$$\dfrac{\begin{matrix}\pi_1\\ \vdots\\ X_1 \vdash \alpha_1\end{matrix} \quad \begin{matrix}\pi_2\\ \vdots\\ X_2 \vdash \alpha_2\end{matrix} \quad \begin{matrix}\pi_3\\ \vdots\\ X_3 \vdash \alpha_3\end{matrix}}{Y \vdash \gamma}\,\mathsf{r}$$

we say that any rule in π_1 is to the left of r, r is to the left of any rule in π_2, and any rule in π_2 is to the left of any rule in π_3.

Now consider the leftmost occurrence of $\vee e$ in π. It is the last rule of a subproof π' of π which looks as follows.

$$\dfrac{\begin{matrix}\pi_1'\\ \vdots\\ X' \vdash \varphi \vee \psi\end{matrix} \quad \begin{matrix}\pi_2'\\ \vdots\\ X', \varphi \vdash \theta\end{matrix} \quad \begin{matrix}\pi_3'\\ \vdots\\ X', \psi \vdash \theta\end{matrix}}{X' \vdash \theta}\,\vee e$$

Since this is the leftmost occurrence of $\vee e$, there is no occurrence of $\vee e$ in π_1'. Further, if $X' \neq X$, it means that π' is part of the proof of a minor premise of some other $\vee e$ rule in π. But that contradicts the fact that π' ends in the leftmost $\vee e$ in π. Thus $X' = X$, and π_1' witnesses the fact that $\varphi \vee \psi \in core(X)$.

◀

▶ **Lemma 16.** *For a down-closed Y, $Y \vdash \alpha$ iff $\alpha \in Y$.*

Proof. If $\alpha \in Y$, then it is obvious that $Y \vdash \alpha$.

In the other direction, suppose $Y \vdash \alpha$ via a proof π with k instances of $\vee e$. We prove the required claim by induction on k.

In the base case, $k = 0$, and $\alpha \in core(Y)$. Since Y is down-closed, $core(Y) \subseteq Y$, and hence $\alpha \in Y$.

In the induction step, suppose there is an instance of $\vee e$ in the proof of $Y \vdash \alpha$. By Lemma 15, we know that there is at least one occurrence of $\vee e$ (say $Y \vdash \gamma$) with major premise $Y \vdash \varphi \vee \psi$ such that $\varphi \vee \psi \in core(Y) \subseteq Y$, which looks as follows.

$$\dfrac{\begin{matrix}\pi_1\\ \vdots\\ Y \vdash \varphi \vee \psi\end{matrix} \quad \begin{matrix}\pi_2\\ \vdots\\ Y, \varphi \vdash \gamma\end{matrix} \quad \begin{matrix}\pi_3\\ \vdots\\ Y, \psi \vdash \gamma\end{matrix}}{Y \vdash \gamma}\,\vee e$$

Thus we have $\varphi \vee \psi \in Y$. Since Y is down-closed either $\varphi \in Y$ or $\psi \in Y$. Suppose, without loss of generality, that $\varphi \in Y$. Now consider π_2. Since $\varphi \in Y$, we know that $Y \cup \{\varphi\} = Y$, and we can replace the big proof of $Y \vdash \gamma$ by π_2, thereby reducing the number of instances of $\vee e$ in the proof of $Y \vdash \alpha$. By induction hypothesis, $\alpha \in Y$, and the lemma follows.

◀

Running time

We now analyze the running time of Algorithm 2. Since Y strictly increases with each iteration of the loop, there are at most $N = |sf|$ iterations of the loop. In each iteration, we test whether Y is down-closed, which amounts to checking whether there is some $\beta_0 \vee \beta_1 \in Y$ such that neither β_0 nor β_1 is in Y. This check takes $O(N)$ time. We also compute $core(Y)$ in each iteration, which takes time $O(N)$. Thus the overall running time is $O(N^2)$.

4.2 Bounding disjunction elimination

It is evident from the algorithm for *derive* that the use of the $\vee e$ rule is an important resource. It makes sense to bound its use and explore its effect on the efficiency of the algorithm. In this section, we show that if we bound the set of formulas on which disjunction elimination is performed, we get a procedure whose running time is polynomial in the input size, though exponential in the number of disjunction eliminations allowed. The following definition makes this notion precise.

▶ **Definition 17.** *Let A be a set of disjunctive formulas. We define a proof of α from X using A (denoted $X \vdash_A \alpha$) as a proof where any $\vee e$ rules are applied only to formulas which appear in A.*

Recall that we have fixed a set sf of size N, and consider the derivability of $X \vdash \alpha$ where $\mathsf{sf}(X \cup \{\alpha\}) \subseteq \mathsf{sf}$. We define $derive_A(X)$ to be $\{\beta \in \mathsf{sf} \mid X \vdash_A \beta\}$. Note that $derive_\varnothing(X)$ is $core(X)$. The check for $X \vdash_A \alpha$ is done by using Algorithm 3 to compute $derive_A(X)$ and then testing whether $\alpha \in derive_A(X)$. (For the purposes of the algorithm, we assume that the set A is equipped with a linear order, so we can refer to the least formula in any subset of A.)

Algorithm 3 Algorithm to compute $derive_A(X)$

```
1: function f(A, X)
2:     Y ← core(X);
3:     if A ∩ Y = ∅ then
4:         return Y;
5:     else
6:         A′ ← A \ {α ∨ β}, where α ∨ β is the least formula in A ∩ Y;
7:         return f(A′, Y ∪ {α}) ∩ f(A′, Y ∪ {β});
8:     end if
9: end function
```

In order to prove the correctness of the above algorithm, we require the following claim.

▶ **Proposition 18.** *Let A be a set of disjunctions and $\alpha \vee \beta \in A$. Let $A' = A \setminus \{\alpha \vee \beta\}$. Then the following hold:*

- *If $X \vdash_A \gamma$ then $X, \alpha \vdash_{A'} \gamma$ and $X, \beta \vdash_{A'} \gamma$.*
- *If $X \vdash_A \alpha \vee \beta$, $X, \alpha \vdash_{A'} \gamma$ and $X, \beta \vdash_{A'} \gamma$, then $X \vdash_A \gamma$.*

Proof.

- Suppose $X \vdash_A \gamma$. Then by monotonicity, we obtain a proof π of $X, \alpha \vdash \gamma$, such that the major premise of every instance of the $\vee e$ rule in π is in A. Note that for every sequent $X' \vdash \delta$ in π, it is the case that $\alpha \in X'$. Consider any subproof π' of π whose conclusion is $X' \vdash \delta$ and last rule is $\vee e$ with major premise $\alpha \vee \beta$ (if there is no such subproof, then π witnesses the fact that $X, \alpha \vdash_{A'} \gamma$). π' has the following form.

$$\dfrac{\begin{array}{c}\pi_1' \\ \vdots \\ X' \vdash \alpha \vee \beta\end{array} \quad \begin{array}{c}\pi_2' \\ \vdots \\ X', \alpha \vdash \delta\end{array} \quad \begin{array}{c}\pi_3' \\ \vdots \\ X', \beta \vdash \delta\end{array}}{X' \vdash \delta} \vee e$$

But observe that since $\alpha \in X'$, $X' \cup \{\alpha\} = X'$. Thus π_2' is itself a proof of $X' \vdash \delta$. We can replace π' by π_2', thereby removing at least one instance of the $\vee e$ rule involving $\alpha \vee \beta$ in π. Repeating this, we obtain that $X, \alpha \vdash_{A'} \gamma$. A similar reasoning gives us the result for $X, \beta \vdash_{A'} \gamma$.

- Aplying $\vee e$ on $\alpha \vee \beta$ using the given proofs of $X, \alpha \vdash_{A'} \gamma$ and $X, \beta \vdash_{A'} \gamma$ and $X \vdash_A \alpha \vee \beta$ for premises gives us the required result of $X \vdash_A \gamma$.

◀

▶ **Lemma 19** (Correctness of Algorithm 3)**.** *For all X and A,*

$$\mathit{derive}_A(X) = f(A, X).$$

Proof. The proof is by induction on the size of A. The base case is when $A = \emptyset$, when clearly the procedure f returns $\mathit{core}(X)$.

For the induction case, suppose $X \vdash_A \gamma$, and let $Y = \mathit{core}(X)$. Consider a normal proof π witnessing $X \vdash_A \gamma$ and assume without loss of generality that there is at least one instance of $\vee e$ in π. From Lemma 15, we see that there is an instance of $\vee e$ in π with major premise $X \vdash \varphi \vee \psi$, where $\varphi \vee \psi \in \mathit{core}(X)$. Thus $A \cap Y \neq \emptyset$. Let $\alpha \vee \beta$ be the least formula in $A \cap Y$. Now since $X \subseteq Y$, $Y \vdash_A \gamma$. Furthermore, $\alpha \vee \beta \in Y$. Hence, by Claim 18, $Y, \alpha \vdash_{A'} \gamma$ and $Y, \beta \vdash_{A'} \gamma$, where $A' = A \setminus \{\alpha \vee \beta\}$. Since A' is of smaller size than A, by the induction hypothesis, $\mathit{derive}_{A'}(Z) = f(A', Z)$ for any Z. Thus $\gamma \in f(A', Y \cup \{\alpha\}) \cap f(A', Y \cup \{\beta\})$. It follows from the definition of f that $\gamma \in f(A, X)$. Thus $\mathit{derive}_A(X) \subseteq f(A, X)$.

On the other hand, suppose that $\gamma \in f(A, X)$, and assume w.l.o.g. that $A \cap Y \neq \emptyset$, where $Y = \mathit{core}(X)$. Letting $\alpha \vee \beta$ be the least formula in $A \cap Y$ and $A' = A \setminus \{\alpha \vee \beta\}$, it is clear that $\gamma \in f(A', Y \cup \{\alpha\}) \cap f(A', Y \cup \{\beta\})$ from the definition of f. Since A' is of smaller size than A, it follows from the induction hypothesis that $Y, \alpha \vdash_{A'} \gamma$ and $Y, \beta \vdash_{A'} \gamma$. Since $Y = \mathit{core}(X)$, it is the case that $X \vdash_{A'} \gamma$ for every $\gamma \in Y$. Thus we can appeal to the admissibility of cut to conclude that $X, \alpha \vdash_{A'} \gamma$ and $X, \beta \vdash_{A'} \gamma$. It follows from Claim 18 that $X \vdash_A \gamma$. Thus $f(A, X) \subseteq \mathit{derive}_A(X)$.

◀

▶ **Theorem 20.** *If $|A| = k$, then $\mathit{derive}_A(X)$ is computable in time $O(2^k \cdot N)$.*

Proof. There are at most 2^k recursive calls to f, and in each invocation we make one call to *core*, which takes $O(N)$ time. Thus the overall running time is $O(2^k \cdot N)$.

◀

5 The complexity of disjunction

We have seen that CL is solvable in linear time, and that DL is solvable in co-NP. Is this the best we can do? Might there not be some clever way to handle disjunction elimination that yields a PTIME algorithm? In this section, we answer these questions by proving co-NP-hardness for three (reasonably minimal) fragments involving disjunction. But first, as a study in contrast, we consider disjunction by itself, and show that that fragment is solvable in PTIME. This indicates that the lower bound results that appear later in this section are a result of *interaction* between the various logical rules, rather than due to disjunction alone.

5.1 The disjunction-only fragment

Let IL$[\vee]$ denote the fragment of IL consisting of the *ax*, $\vee i$ and $\vee e$ rules, and involving formulas of $\Phi^{\vee}$.

▶ **Theorem 21.** *The derivability problem for IL$[\vee]$ is in PTIME.*

Suppose $X = \{\alpha_i^1 \vee \alpha_i^2 \vee \cdots \vee \alpha_i^k \mid 1 \leqslant i \leqslant n\}$ is a set of formulas from $\Phi^{\vee}$, with each $\alpha_i^j \in \mathscr{P}$. Let $\beta = \beta^1 \vee \beta^2 \vee \cdots \vee \beta^k \in \Phi^{\vee}$, with each $\beta^j \in \mathscr{P}$. (Note that any input to the derivability problem of IL$^{\vee}$ can be converted to the above form by choosing appropriate k, flattening the disjunctions, and repeating disjuncts). We now have the following claim.

▶ **Proposition 22.** *$X \vdash \beta$ iff there exists $i \leqslant n$ such that $\alpha_i^1 \vee \alpha_i^2 \vee \cdots \vee \alpha_i^k \vdash \beta$.*

Proof. It is obvious that if $\alpha_i^1 \vee \alpha_i^2 \vee \cdots \vee \alpha_i^k \vdash \beta$ then $X \vdash \beta$ (by Monotonicity).

For proving the other direction, suppose (towards a contradiction) $X \vdash \beta$, but there is no i such that $\alpha_i^1 \vee \alpha_i^2 \vee \cdots \vee \alpha_i^k \vdash \beta$. In particular, from the Left Disjunction Property, for every i, some $\alpha_i^{j_i} \nvdash \beta$. W.l.o.g., assume that $j_i = 1$ for every i. Therefore we have

$$\alpha_1^1 \nvdash \beta,\ \alpha_2^1 \nvdash \beta,\ \ldots,\ \alpha_n^1 \nvdash \beta.$$

Now, since $X \vdash \beta$ and $\alpha_i^1 \vdash \alpha_i^1 \vee \cdots \vee \alpha_i^k$ for each $i \leqslant n$, it follows by Admissibility of Cut that $\alpha_1^1, \ldots, \alpha_n^1 \vdash \beta$ (and there is a normal proof π with that conclusion). Since all the α_i^1s are atomic propositions, the only rules that can appear in π are *ax* and $\vee i$. Therefore, at some point, one of the α_i^1s must have contributed to a β^j via an *ax* rule. However, this gives us $\alpha_i^1 \vdash \beta$ (by deriving β^j and then applying $\vee i$), which is a contradiction. Thus we have the required claim.

◀

Given this claim, we know that it is enough to see if a particular formula on the left (say α_i) derives β. In particular, from the Left Disjunction Property, we get that every disjunct in α_i needs to derive β. Therefore, the derivability problem is equivalent to checking if there is a formula in X all of whose disjuncts occur in β, and thus we obtain the required PTIME procedure.

5.2 Disjunction and conjunction

We have now confirmed that the $\vee$-only fragment is in PTIME. It is also known that some other fragments without disjunction elimination (CL, for example) give rise to PTIME logics. However, we obtain the following result for the logic with conjunction and disjunction.[5]

[5] This fragment is interesting: one can show that $X \vdash_{\mathsf{IL}[\vee,\wedge]} \alpha$ iff α is a consequence of X in classical logic. Thus an NP procedure for non-derivability would just guess a valuation that satisfied X but not α. While this translation to classical logic yields a simple algorithm, it is not clear to us how to use this translation to prove lower bounds.

Let $\mathsf{IL}[\vee,\wedge]$ denote the fragment of IL consisting of the *ax*, $\vee i$, $\vee e$, $\wedge i$ and $\wedge e$ rules, and involving formulas of $\Phi^{\vee,\wedge}$.

▶ **Theorem 23.** *The derivability problem for $\mathsf{IL}[\vee,\wedge]$ is co-NP-hard.*

The hardness result is obtained by reducing the validity problem for boolean formulas to the derivability problem for $\mathsf{IL}[\vee,\wedge]$. In fact, it suffices to consider the validity problem for boolean formulas in disjunctive normal form. We show how to define for each DNF formula φ a set of $\mathsf{IL}[\vee,\wedge]$-formulas S_φ and an $\mathsf{IL}[\vee,\wedge]$-formula $\overline{\varphi}$ such that $S_\varphi \vdash \overline{\varphi}$ iff φ is a tautology.

Let $\{x_1, x_2, \ldots\}$ be the set of all boolean variables. For each boolean variable x_i, fix two distinct atomic propositions $p_i, q_i \in \mathscr{P}$. We define $\overline{\varphi}$ as follows, by induction.

- $\overline{x_i} = p_i$
- $\overline{\neg x_i} = q_i$
- $\overline{\varphi \vee \psi} = \overline{\varphi} \vee \overline{\psi}$
- $\overline{\varphi \wedge \psi} = \overline{\varphi} \wedge \overline{\psi}$

Let $\mathrm{Voc}(\varphi)$, the set of all boolean variables occurring in φ, be $\{x_1, \ldots, x_n\}$. Then

$$S_\varphi = \{p_1 \vee q_1, \ldots, p_n \vee q_n\}.$$

▶ **Lemma 24.** *$S_\varphi \vdash \overline{\varphi}$ iff φ is a tautology.*

Proof. Recall that a propositional valuation v over a set of variables $\mathscr{V}$ is just a subset of $\mathscr{V}$, namely those variables that are set to *true* by v.

For a valuation $v \subseteq \{x_1, \ldots, x_n\}$, define $S_v = \{p_i \mid x_i \in v\} \cup \{q_i \mid x_i \notin v\}$.

By repeated appeal to the Left Disjunction Property, it is easy to see that $S_\varphi \vdash \overline{\varphi}$ iff for all valuations v over $\{x_1, \ldots, x_n\}$, $S_v \vdash \overline{\varphi}$. We now show that $S_v \vdash \overline{\varphi}$ iff $v \models \varphi$. The statement of the lemma follows immediately from this.

- We first show by induction on $\psi \in \mathsf{sf}(\varphi)$ that whenever $v \models \psi$, it is the case that $S_v \vdash \overline{\psi}$.
  - If $\psi = x_i$ or $\psi = \neg x_i$, then $S_v \vdash \overline{\psi}$ follows from the *ax* rule.
  - If $\psi = \psi_1 \wedge \psi_2$, then it is the case that $v \models \psi_1$ and $v \models \psi_2$. By induction hypothesis, $S_v \vdash \overline{\psi_1}$ and $S_v \vdash \overline{\psi_2}$. Hence, by using $\wedge i$, it follows that $S_v \vdash \overline{\psi_1 \wedge \psi_2}$.
  - If $\psi = \psi_1 \vee \psi_2$, then it is the case that either $v \models \psi_1$ or $v \models \psi_2$. By induction hypothesis, $S_v \vdash \overline{\psi_1}$ or $S_v \vdash \overline{\psi_2}$. In either case it follows that $S_v \vdash \overline{\psi_1 \vee \psi_2}$, by using $\vee i$.
- We now show that if $S_v \vdash \overline{\varphi}$, then $v \models \varphi$. Suppose π is a normal proof of $S_v \vdash \overline{\varphi}$, and that there is an occurrence of the $\wedge e$ rule or $\vee e$ rule in π with major premise $S' \vdash \gamma$. We denote by ϖ this subproof with conclusion $S' \vdash \gamma$. Note that ϖ ends in a pure elimination rule, since π is normal and every pure elimination rule and hybrid rule has as its major premise the conclusion of a pure elimination rule. By Theorem 4, we see that $S' \subseteq \mathsf{sf}(S_v) = S_v$, and $\gamma \in \mathsf{sf}(S')$. But γ is of the form $\alpha \vee \beta$ or $\alpha \wedge \beta$, and this contradicts the fact that $S_v \subseteq \mathscr{P}$. Thus π consists of only the *ax*, $\wedge i$ and $\vee i$ rules. We now show by induction that for all subproofs π' of π with conclusion $S_v \vdash \overline{\psi}$, it is the case that $v \models \psi$.
  - Suppose the last rule of π' is *ax*. Then $\overline{\psi} \in S_v$, and for some $i \leqslant n$, $\psi = x_i$ or $\psi = \neg x_i$. It can be easily seen that $v \models \psi$ (by the definition of S_v).

- Suppose the last rule of π' is $\wedge i$. Then $\overline{\psi} = \overline{\psi_1} \wedge \overline{\psi_2}$, and $S_v \vdash \overline{\psi_1}$ and $S_v \vdash \overline{\psi_2}$. Thus, by induction hypothesis, $v \models \psi_1$ and $v \models \psi_2$. Therefore $v \models \psi$.
- Suppose the last rule of π' is $\vee i$. Then $\overline{\psi} = \overline{\psi_1} \vee \overline{\psi_2}$, and either $S_v \vdash \overline{\psi_1}$ or $S_v \vdash \overline{\psi_2}$. Thus, by induction hypothesis, either $v \models \psi_1$ or $v \models \psi_2$. Therefore $v \models \psi$.

◀

5.3 Disjunction and implication elimination

We now consider another minimal system, IL$[\vee, \rightarrow e]$, consisting of the rules *ax*, $\vee i$, $\vee e$ and $\rightarrow e$ and involving formulas from $\Phi^{\vee,\rightarrow}$, and prove the following result.

► **Theorem 25.** *The derivability problem for IL$[\vee, \rightarrow e]$ is co-NP-hard.*

The proof is by reduction from the validity problem for 3-DNF, as detailed below.

Let φ be a 3-DNF formula with each clause having exactly 3 literals. Let Voc(φ) be $\{x_1, \ldots, x_n\}$. We define $indx(\varphi) = \{1, \ldots, n\} \cup \{1', \ldots, n'\}$, where $(i')' = i$ for any $i \in indx(\varphi)$. For $i \leqslant n$, we define $l(i) = x_i$ and $l(i') = \neg x_i$.

We define the following sets.

$S_\varphi := \left\{ p_a \vee p_{a'} \mid a \in indx(\varphi) \right\}$.

$T_\varphi := \left\{ p_a \rightarrow p_b \rightarrow p_c \rightarrow p_{abc} \mid a, b, c \in indx(\varphi) \right\}$.

We define $\overline{\varphi}$ as follows:

$\overline{\varphi} := \bigvee \left\{ p_{abc} \mid l(a) \wedge l(b) \wedge l(c) \text{ is a disjunct of } \varphi \right\}$.

For each valuation $v \subseteq \{x_1, \ldots, x_n\}$, define S_v to be

$\{p_i \mid x_i \in v\} \cup \{p_{i'} \mid x_i \notin v\}$.

► **Lemma 26.** *$S_\varphi, T_\varphi \vdash \overline{\varphi}$ iff φ is a tautology.*

Proof. By repeated appeal to the Left Disjunction Property, it is easy to see that $S_\varphi, T_\varphi \vdash \overline{\varphi}$ iff $S_v, T_\varphi \vdash \overline{\varphi}$ for all valuations v over $\{x_1, \ldots, x_n\}$. We now show that for all such valuations, $v \models \varphi$ iff $S_v, T_\varphi \vdash \overline{\varphi}$.

Let π be a normal proof of $S_v, T_\varphi \vdash \overline{\varphi}$. The last rule of π has to be $\vee i$, since if π ends in an elimination rule, from the Subformula Property it follows that a disjunction is a subformula of $S_v \cup T_\varphi$, which is not the case. Repeating this argument, we see that there is a subproof of π with conclusion $S_v, T_\varphi \vdash p_{abc}$ for some disjunct $l(a) \wedge l(b) \wedge l(c)$ of φ. We now show that for any valuation v, $S_v, T_\varphi \vdash p_{abc}$ iff $v \models l(a) \wedge l(b) \wedge l(c)$.

If $v \models l(a) \wedge l(b) \wedge l(c)$, then we have $p_a, p_b, p_c \in S_v$ (from the definition of S_v), and therefore by applying the $\rightarrow e$ rule to $p_a \rightarrow p_b \rightarrow p_c \rightarrow p_{abc}$ in T_φ, we have $S_v, T_\varphi \vdash p_{abc}$. In the other direction, suppose we have a normal proof π of $S_v, T_\varphi \vdash p_{abc}$. By examining S_v and T_φ, we see that only $p_a \rightarrow p_b \rightarrow p_c \rightarrow p_{abc}$ mentions p_{abc}. So it is clear that p_c must be derivable from S_v, T_φ, and the last rule of π must be $\rightarrow e$, applied to $p_c \rightarrow p_{abc}$. Now in order for this formula to be derivable, p_b must be derivable, and similarly p_a must be derivable. Since p_a, p_b and p_c can only be obtained by *ax*, it must be that $p_a, p_b, p_c \in S_v$ and therefore $v \models l(a) \wedge l(b) \wedge l(c)$.

Thus we have that $S_v, T_\varphi \vdash p_{abc}$ iff $v \models l(a) \wedge l(b) \wedge l(c)$, and the required claim follows.

◀

5.4 Disjunction and negation elimination

We consider yet another minimal system, $\mathsf{IL}[\vee, \neg e]$, consisting of the rules ax, $\vee i$, $\vee e$ and $\rightarrow e$ and involving formulas from $\Phi^{\vee,\neg}$, and prove the following result.

▶ **Theorem 27.** *The derivability problem for $\mathsf{IL}[\vee, \neg e]$ is co-NP-hard.*

The proof is again by reduction from the validity problem for 3-DNF, as detailed below.

Let φ be a 3-DNF formula with each clause having exactly 3 literals. Let $\mathrm{Voc}(\varphi)$ be $\{x_1, \dots, x_n\}$. We define $indx(\varphi) = \{1, \dots, n\} \cup \{1', \dots, n'\}$, where $(i')' = i$ for any $i \in indx(\varphi)$. For $i \leqslant n$, we define $l(i) = x_i$ and $l(i') = \neg x_i$. For $i \leqslant n$, we define $\widehat{p_i} = \neg p_i$ and $\widehat{p_{i'}} = p_i$.

We define the following sets.

$$S_\varphi := \{p_i \vee \neg p_i \mid i \leqslant n\}.$$

$$T_\varphi := \{\widehat{p_a} \vee \widehat{p_b} \vee \widehat{p_c} \vee p_{abc} \mid a, b, c \in indx(\varphi)\}.$$

We define $\overline{\varphi}$ as follows:

$$\overline{\varphi} := \bigvee \{p_{abc} \mid (l(a) \wedge l(b) \wedge l(c)) \in \varphi\}.$$

For each valuation $v \subseteq \{x_1, \dots, x_n\}$, define S_v to be

$$\{p_i \mid x_i \in v\} \cup \{p_{i'} \mid x_i \notin v\}.$$

▶ **Lemma 28.** *$S_\varphi, T_\varphi \vdash \overline{\varphi}$ iff φ is a tautology.*

Proof. Let us assume that $\varphi = \bigvee \{l(a_i) \wedge l(b_i) \wedge l(c_i) \mid i \leqslant m\}$, for ease of notation. By repeated appeal to the Left Disjunction Property, it is easy to see that $S_\varphi, T_\varphi \vdash \overline{\varphi}$ iff $S_v, T_\varphi \vdash \overline{\varphi}$ for all valuations v over $\{x_1, \dots, x_n\}$. We now show that for all valuations v,

$$v \models \varphi \text{ iff } S_v, T_\varphi \vdash \overline{\varphi}.$$

- Suppose $v \models \varphi$. This means that $v \models l(a_i) \wedge l(b_i) \wedge l(c_i)$ for some i. For the sake of readability, let us refer to this disjunct as $l(a) \wedge l(b) \wedge l(c)$. It is clear in this case that $\{p_a, p_b, p_c\} \subseteq S_v$. Let $T' = T_\varphi \setminus \{\widehat{p_a} \vee \widehat{p_b} \vee \widehat{p_c} \vee p_{abc}\}$. Now

$$\begin{aligned} & S_v, T_\varphi \vdash \overline{\varphi} && (1)\\ & \qquad \text{iff} && \\ & S_v, T', \widehat{p_a} \vdash \overline{\varphi} \quad \text{and} && (2)\\ & S_v, T', \widehat{p_b} \vdash \overline{\varphi} \quad \text{and} && (3)\\ & S_v, T', \widehat{p_c} \vdash \overline{\varphi} \quad \text{and} && (4)\\ & S_v, T', p_{abc} \vdash \overline{\varphi} && (5) \end{aligned}$$

Now it is easily seen that (2), (3) and (4) are true, since $\{p_a, p_b, p_c\} \subseteq S_v$ and we can use the $\neg e$ rule. Now $S_v, T', p_{abc} \vdash p_{abc}$, and hence (5) follows (by a series of applications of $\vee i$ rules). Thus $S_v, T_\varphi \vdash \overline{\varphi}$.

- Suppose $v \not\models \varphi$. This means that for every $i \leqslant m$, either $v \not\models l(a_i)$ or $v \not\models l(b_i)$ or $v \not\models l(c_i)$. Without loss of generality, let us assume that $v \not\models l(a_i)$ for all $i \leqslant m$. Now it is clear that $p_{a_i} \notin S_v$, and thus $\widehat{p_{a_i}} \in S_v$, for each $i \leqslant m$.
  Now suppose $S_v, T_\varphi \vdash \overline{\varphi}$. This implies that $S_v, \{\widehat{p_{a_i}} \mid i \leqslant m\} \vdash \overline{\varphi}$. However, $\widehat{p_{a_i}} \in S_v$ for all $i \leqslant m$, and therefore $S_v \vdash \overline{\varphi}$. Suppose there is a normal proof π for the same. Since $S_v \subseteq \mathscr{P}$, the only rules that can occur in π are *ax* and $\vee i$. This in turn implies that $S_v \vdash p_{a_i b_i c_i}$ for some $i \leqslant m$. But a proof of $S_v \vdash p_{a_i b_i c_i}$ can only use the *ax* rule, but $p_{a_i b_i c_i} \notin S_v$, so we have a contradiction. Thus we have proved that if $v \not\models \varphi$ then $S_v, T_\varphi \not\vdash \overline{\varphi}$.

◀

6 Discussion

To summarize our results, we have presented a core fragment of IL, which we call CL, whose derivability problem is solvable in linear time, and used this algorithm as a core subroutine in a co-NP decision procedure for the larger fragment DL (which includes the $\vee e$ rule). We cannot do better than co-NP when we consider disjunction interacting with other operators, as demonstrated by the lower bound proofs we have provided for IL$[\vee, \wedge]$, IL$[\vee, \rightarrow e]$, and IL$[\vee, \neg e]$. The fragment IL$[\vee]$ (which includes only the *ax*, $\vee i$, and $\vee e$ rules) though, is solvable in PTIME, in contrast to the implication-only fragment of IL. Of the two rules for negation, $\neg e$ does not modify the assumptions in the sequents, whereas $\neg i$ discharges the assumption α while concluding $\neg\alpha$. There does not appear to be a straightforward adaptation of our algorithms to handle $\neg i$.[6]

We can also consider adding $\Box$-like modalities to the $[\wedge, \vee]$ fragment of our logic. This system is in co-NP, and the algorithm proceeds along similar lines to the one in [15]. On the other hand, if we add modalities to a logic with implication (even primal implication), the system is PSPACE-complete [4].

Perhaps the most important way to take this work further is to identify restricted forms of disjunction that are efficiently solvable, as this would be of use in many practical applications. If we can identify scenarios in which a bounded number of applications of the disjunction elimination rule would suffice, our PTIME algorithm in Section 4.2 would come very handy, opening the door for parametrized algorithms for derivability.

References

1 M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, 1993.

2 Arnon Avron. Tonk – a full mathematical solution. In *Hues of Philosophy: Essays in Memory of Ruth Manor*, pages 17–42. College Publications, 2010.

3 A. Baskar, P. Naldurg, K. R. Raghavendra, and S. P. Suresh. Primal Infon Logic: Derivability in Polynomial Time. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013)*, volume 24 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 163–174, 2013.

[6] Note that the fragment studied in [4], consisting of rules for primal implication, disjunction, and a $\perp$ operator, does not subsume $\neg i$. While full implication and $\perp$ can be used to code the negation rules, primal implication and $\perp$ can only capture the effect of the $\neg e$ rule.

4 L.D. Beklemishev and Y. Gurevich. Propositional Primal Logic with Disjunction. *Journal of Logic and Computation*, 24(1):257–282, 2014.

5 A. Chagrov and M. Zakharyaschev. *Modal Logic*. Clarendon Press, Oxford, 1997.

6 H. Comon-Lundh and V. Shmatikov. Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive or. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 271–282, 2003.

7 H. Comon-Lundh and R. Treinen. Easy Intruder Deductions. In N. Dershowitz, editor, *Verification: Theory and Practice*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242, 2003.

8 D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

9 D.M. Gabbay and D.H.J. de Jongh. A Sequence of Decidable Finitely Axiomatizable Intermediate Logics with the Disjunction Property. *Journal of Symbol Logic*, 39(1):67–78, 1974.

10 Y. Gurevich and I. Neeman. DKAL: Distributed-Knowledge Authorization Language. In *21st IEEE CSF Symposium*, pages 149–162, 2008.

11 Y. Gurevich and I. Neeman. Logic of Infons: The Propositional Case. *ACM Trans. Comput. Logic*, 12(2):9:1–9:28, January 2011.

12 H. Kurokawa. Hypersequent Calculi for Intuitionistic Logic with Classical Atoms. *Annals of Pure and Applied Logic*, 161(3):427–446, 2009.

13 M. Magirius, Mundhenk, and R. M., Palenta. The Complexity of Primal Logic with Disjunction. *Information Processing Letters*, 115(5):536–542, 2015.

14 D.A. McAllister. Automatic Recognition of Tractability in Inference Relations. *J. ACM*, 40(2):284–303, 1993.

15 R. Ramanujam, V. Sundararajan, and S. P. Suresh. Extending Dolev-Yao with Assertions. In *Information Systems Security - 10th International Conference, ICISS 2014*, pages 50–68, 2014.

16 A. Sakharov. Median Logic. Technical report, 2004. URL: `http://www.mathsoc.spb.ru/preprint/2004/index.html`.

17 R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9(1):67–72, 1979.

A Weak normalization and subformula property

Among the rules, *ax*, $\wedge e$ and $\rightarrow e$ are the *pure elimination rules*, $\neg e$, $\neg i$ and $\vee e$ are the *hybrid rules* and the rest are *pure introduction rules*. A *normal derivation* is one where the major premise of every pure elimination rule and hybrid rule is the conclusion of a pure elimination rule. A derivation is normal iff its *cut rank* is 0, as given by the following definition.

▶ **Definition 29** (Cut rank of a derivation). *Let π be a derivation with conclusion $X \vdash \alpha$ and last rule r. Let $\pi_1, \ldots, \pi_n$ be the immediate subproofs of π. Let each π_i end with rule r_i and have conclusion $X_i \vdash \alpha_i$. Also, let $X_1 \vdash \alpha_1$ be the major premise of r. By induction on π, we define cutrank(π) as follows:*

- *If r is a pure elimination rule or a hybrid rule and r_1 is not a pure elimination rule, then*

$$cutrank(\pi) = max(|\alpha_1|, cutrank(\pi_1), \cdots, cutrank(\pi_n)).$$

- *Otherwise*

$$cutrank(\pi) = max(cutrank(\pi_1), \cdots, cutrank(\pi_n)).$$

(Note that if r *is ax, $cutrank(\pi) = 0$, by the second clause above.)*

▶ **Proposition 30** (Monotonicity). *If there is a proof of $X \vdash \alpha$ with cut rank m and $X \subseteq X'$, then there is a proof of $X' \vdash \alpha$ with cut rank m.*

Proof. Let π be a proof of $X \vdash \alpha$, and let $Y = X' \setminus X$. It is easy to check that replacing every sequent $Z \vdash \beta$ occurring in π by $Z \cup Y \vdash \beta$, we still have a valid proof π', with conclusion $X' \vdash \alpha$. (The point is that in rules involving a discharge of the premises, the discharge is optional, so if some rule in π discharges a formula in Y, we can apply the same rule in π' without discharging that formula.) Since the structure of the proof does not change, the cut rank remains the same. ◀

▶ **Proposition 31** (Admissibility of Cut). *If π_1 is a derivation of $X \vdash \alpha$ (with last rule* r$_1$*) and π a derivation of $Y \vdash \beta$ (with last rule* r*), then there is a derivation ϖ of $X, Y - \alpha \vdash \beta$ such that*

$$cutrank(\varpi) \leqslant max(cutrank(\pi_1), cutrank(\pi), |\alpha|).$$

Further, either the last rule of ϖ is r *or $\beta = \alpha$ and the last rule of ϖ is* r$_1$.

Proof. The proof is by induction on the size of π, and a case analysis on r. For notational ease, we let $cutrank(\pi_1) = m_1$, $cutrank(\pi) = m$, and $n = max(m_1, m, |\alpha|)$. We present a few sample cases below.

r is *ax*: If $\beta \neq \alpha$, then $\beta \in Y - \alpha$ and we can take ϖ to be the following proof:

$$\frac{}{X, Y - \alpha \vdash \beta}\ ax$$

Clearly $cutrank(\varpi) = 0 \leqslant n$ and the last rule of ϖ is r.

If $\beta = \alpha$, then we take ϖ to be the proof of $X, Y - \alpha \vdash \beta$ guaranteed by Monotonicity (applied to π_1). Clearly $cutrank(\varpi) = m_1 \leqslant n$, and the last rule of ϖ is r$_1$ as required.

r is $\wedge i$: Then π has the following structure:

$$\frac{\begin{matrix}\tau_1 \\ \vdots \\ Y \vdash \beta_1\end{matrix} \quad \begin{matrix}\tau_2 \\ \vdots \\ Y \vdash \beta_2\end{matrix}}{Y \vdash \beta}\ \wedge i$$

By induction hypothesis, there exist proofs ϖ_1 and ϖ_2 with conclusions $X, Y - \alpha \vdash \beta_1$ and $X, Y - \alpha \vdash \beta_2$ respectively, both of which have cut ranks at most n. We define ϖ to be the following proof:

$$\frac{\begin{matrix}\varpi_1 \\ \vdots \\ X, Y - \alpha \vdash \beta_1\end{matrix} \quad \begin{matrix}\varpi_2 \\ \vdots \\ X, Y - \alpha \vdash \beta_2\end{matrix}}{X, Y - \alpha \vdash \beta}\ \wedge i$$

Clearly $cutrank(\varpi) = max(cutrank(\varpi_1), cutrank(\varpi_2)) \leqslant n$. Further, the last rule of ϖ is r.

r is $\rightarrow i$: Then $\beta = \varphi \rightarrow \psi$ and π has the following structure:

$$\frac{\begin{matrix}\tau_1 \\ \vdots \\ Y, \varphi \vdash \psi\end{matrix}}{Y \vdash \varphi \rightarrow \psi}\ \rightarrow i$$

By induction hypothesis, there exist a proof ϖ_1 with conclusion $X, (Y, \varphi) - \alpha \vdash \psi$ whose cut rank is at most n. By appealing to Monotonicity if necessary (in the case when $\varphi = \alpha$), we can take the conclusion of ϖ_1 to be $X, \varphi, Y - \alpha \vdash \psi$. ϖ is the following proof:

$$\dfrac{\begin{array}{c}\varpi_1\\ \vdots\\ X,\varphi,Y-\alpha\vdash\psi\end{array}}{X,Y-\alpha\vdash\varphi\to\psi}\to i$$

Clearly $\mathit{cutrank}(\varpi) = \mathit{cutrank}(\varpi_1) \leqslant n$. Further, the last rule of ϖ is r.

r is $\vee e$**:** Then π has the following structure:

$$\dfrac{\dfrac{\begin{array}{c}\tau_1\\ \vdots\end{array}}{Y\vdash\varphi\vee\psi}\mathrm{r}_1' \qquad \begin{array}{c}\tau_2\\ \vdots\\ Y,\varphi\vdash\beta\end{array} \qquad \begin{array}{c}\tau_3\\ \vdots\\ Y,\psi\vdash\beta\end{array}}{Y\vdash\beta}\vee e$$

By induction hypothesis, there exist proofs ϖ_1, ϖ_2 and ϖ_3 with conclusions respectively $X, Y - \alpha \vdash \varphi \vee \psi$, $X, (Y, \varphi) - \alpha \vdash \beta$, and $X, (Y, \psi) - \alpha \vdash \beta$, all of whose cut ranks are $\leqslant n$. By appealing to Monotonicity if necessary (in the cases when α is φ or ψ), we can take the conclusion of ϖ_2 and ϖ_3 to be $X, \varphi, Y - \alpha \vdash \beta$ and $X, \psi, Y - \alpha \vdash \beta$. ϖ is the following proof:

$$\dfrac{\dfrac{\begin{array}{c}\varpi_1\\ \vdots\end{array}}{X,Y-\alpha\vdash\varphi\vee\psi}\mathrm{r}_1'' \qquad \begin{array}{c}\varpi_2\\ \vdots\\ X,\varphi,Y-\alpha\vdash\beta\end{array} \qquad \begin{array}{c}\varpi_3\\ \vdots\\ X,\psi,Y-\alpha\vdash\beta\end{array}}{X,Y-\alpha\vdash\beta}\vee e$$

Now if r_1'' is a pure elimination, $\mathit{cutrank}(\varpi) \leqslant n$. Otherwise, $\mathit{cutrank}(\varpi) \leqslant \mathit{max}(|\varphi \vee \psi|, n)$. But then either $\mathrm{r}_1'' = \mathrm{r}_1'$ (in which case $|\varphi \vee \psi| \leqslant m \leqslant n$), or $\alpha = \varphi \vee \psi$ and $\mathrm{r}_1'' = \mathrm{r}_1$ (in which case $|\varphi \vee \psi| = |\alpha| \leqslant n$). Thus $\mathit{cutrank}(\varpi) \leqslant n$. Again the last rule of ϖ is r.

◀

▶ **Lemma 32.** *Let π be a derivation with conclusion $X \vdash \alpha$ and last rule r with $\mathit{cutrank}(\pi) = m > 0$, such that all proper subderivations of π are of rank $< m$. Then the following hold.*

1. *If r is a pure elimination rule, $|\alpha| < m$.*
2. *There is a derivation π' of $X \vdash \alpha$ such that $\mathit{cutrank}(\pi') < m$.*

Proof. Let $\pi_1, \ldots, \pi_n$ be the immediate subproofs of π. Let each π_i end with rule r_i and have conclusion $X_i \vdash \alpha_i$, and let $X_1 \vdash \alpha_1$ be the major premise of r. Given the conditions of the lemma, it is clear that $\mathit{cutrank}(\pi) = |\alpha_1| = m$, r_1 is not a pure elimination rule, r is a pure elimination rule or a hybrid rule, and $X_1 = X$.

1. If r is a pure elimination rule, then we have the following cases:
   - $\alpha_1 = \alpha \wedge \beta$ or $\alpha_1 = \beta \wedge \alpha$, for some β.
   - $\alpha_1 = \beta \to \alpha$ for some β.

   In both these cases, it is clear that $|\alpha| < |\alpha_1| = m$.
2. To show the existence of π', we perform an induction on $\|\pi\|$ and a case analysis on r_1.

- Suppose r_1 is $\wedge i$. Then r has to be $\wedge e$. In this case we can take π' to be one of the immediate subproofs of π_1, and clearly $cutrank(\pi') < m$.
- Suppose r_1 is $\vee i$. Then r has to be $\vee e$. Say $\alpha_1 = \beta \vee \gamma$ and the major premise of r_1 is β. Note that $|\beta| < |\beta \vee \gamma| = m$. Let π_2 be the immediate subproof of π with conclusion $X, \beta \vdash \alpha$, and let π_{11} be the subproof of π_1 with conclusion $X \vdash \beta$. Thus we can apply cut on π_{11} and π_2 to get a derivation π' of $X \vdash \alpha$ such that

$$cutrank(\pi') \leqslant max(|\beta|, cutrank(\pi_{11}), cutrank(\pi_2)) < m.$$

- Suppose r_1 is $\rightarrow i$. Then r is $\rightarrow e$ and $\alpha_1 = \beta \rightarrow \alpha$, and π has the following form:

$$\dfrac{\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X,\beta\vdash\alpha\end{matrix}}{X\vdash\beta\rightarrow\alpha}\rightarrow i \qquad \begin{matrix}\pi_2\\ \vdots\\ X\vdash\beta\end{matrix}}{X\vdash\alpha}\rightarrow e$$

Now by applying cut on π_{11} and π_2, we see that there is a proof π' of $X \vdash \alpha$ of cut rank $\leqslant max(m - 1, |\beta|) < m$ (since $|\beta| < |\beta \rightarrow \alpha| = m$).
- Suppose r_1 is $\neg i$. Then r is either $\neg i$ or $\neg e$. We consider the case when r is $\neg i$ – the other case is handled similarly. In this case $\alpha = \neg\alpha'$ and π has the following form:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\pi_{11}\\ \vdots\end{matrix}}{X,\alpha',\beta\vdash\neg\gamma}r_{11} \qquad \dfrac{\begin{matrix}\pi_{12}\\ \vdots\end{matrix}}{X,\alpha',\beta\vdash\gamma}r_{12}}{X,\alpha'\vdash\neg\beta}\neg i \qquad \dfrac{\begin{matrix}\pi_2\\ \vdots\end{matrix}}{X,\alpha'\vdash\beta}r_2}{X\vdash\neg\alpha'}\neg i$$

Now by applying cut on π_2 and π_{11}, as well as on π_2 and π_{12}, we see that there are proofs π'_1 and π'_2 of $X, \alpha' \vdash \neg\gamma$ and $X, \alpha' \vdash \gamma$ respectively, with last rules r'_1 and r'_2, and both of cut rank $\leqslant max(m - 1, |\beta|) < m$ (since $|\beta| < |\neg\beta| = m$). π' can be taken to be the following:

$$\dfrac{\begin{matrix}\pi'_1\\ \vdots\\ X,\alpha'\vdash\neg\gamma\end{matrix} \qquad \begin{matrix}\pi'_2\\ \vdots\\ X,\alpha'\vdash\gamma\end{matrix}}{X\vdash\neg\alpha'}\neg i$$

Now if r'_1 is a pure elimination,

$$cutrank(\pi') = max(cutrank(\pi'_1), cutrank(\pi'_2)) < m.$$

Otherwise, $cutrank(\pi') \leqslant max(m - 1, |\neg\gamma|)$. But by Proposition 31, either $r'_1 = r_{11}$, or $r'_1 = r_2$ (and $\neg\gamma = \beta$). In the former case, $|\neg\gamma| \leqslant cutrank(\pi_1) < m$. Otherwise $|\neg\gamma| < |\neg\beta| = m$. Therefore $cutrank(\pi') < m$.
- Suppose r_1 is $\neg e$. r could be any pure elimination or hybrid rule. We shall consider the cases when it is $\neg i$ and $\vee e$. The other cases are similar or simpler.

- Suppose r is $\neg i$. Then $\alpha = \neg\alpha'$ and π has the following form:

$$\dfrac{\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X,\alpha'\vdash\neg\gamma\end{matrix}\quad\begin{matrix}\pi_{12}\\ \vdots\\ X,\alpha'\vdash\gamma\end{matrix}}{X,\alpha'\vdash\neg\beta}\,\neg e\qquad\begin{matrix}\pi_{2}\\ \vdots\\ X,\alpha'\vdash\beta\end{matrix}}{X\vdash\neg\alpha'}\,\neg i$$

  π' is taken to be the following:

$$\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X,\alpha'\vdash\neg\gamma\end{matrix}\quad\begin{matrix}\pi_{12}\\ \vdots\\ X,\alpha'\vdash\gamma\end{matrix}}{X\vdash\neg\alpha'}\,\neg i$$

  Clearly $\mathit{cutrank}(\pi') = \mathit{cutrank}(\pi_1) < m$.

- Suppose r is $\vee e$. Then π has the following form:

$$\dfrac{\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X\vdash\neg\gamma\end{matrix}\quad\begin{matrix}\pi_{12}\\ \vdots\\ X\vdash\gamma\end{matrix}}{X\vdash\varphi\vee\psi}\,\neg e\qquad\begin{matrix}\pi_{2}\\ \vdots\\ X,\varphi\vdash\alpha\end{matrix}\quad\begin{matrix}\pi_{3}\\ \vdots\\ X,\psi\vdash\alpha\end{matrix}}{X\vdash\alpha}\,\vee e$$

  π' is taken to be the following:

$$\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X\vdash\neg\gamma\end{matrix}\quad\begin{matrix}\pi_{12}\\ \vdots\\ X\vdash\gamma\end{matrix}}{X\vdash\alpha}\,\neg e$$

  Clearly $\mathit{cutrank}(\pi') = \mathit{cutrank}(\pi_1) < m$.

- Suppose r_1 is $\vee e$. Now r can be any pure elimination or hybrid rule. We consider the case when it is $\vee e$. The rest of the cases are similar. Now $\alpha_1 = \beta\vee\beta'$ and π has the following form:

$$\dfrac{\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X\vdash\gamma\vee\gamma'\end{matrix}\quad\begin{matrix}\pi_{12}\\ \vdots\\ X,\gamma\vdash\beta\vee\beta'\end{matrix}\quad\begin{matrix}\pi_{13}\\ \vdots\\ X,\gamma'\vdash\beta\vee\beta'\end{matrix}}{X\vdash\beta\vee\beta'}\,\vee e\qquad\begin{matrix}\pi_{2}\\ \vdots\\ X,\beta\vdash\alpha\end{matrix}\quad\begin{matrix}\pi_{3}\\ \vdots\\ X,\beta'\vdash\alpha\end{matrix}}{X\vdash\alpha}\,\vee e$$

  Let τ_2 be the following proof

$$\dfrac{\begin{matrix}\pi_{12}\\ \vdots\\ X,\gamma\vdash\beta\vee\beta'\end{matrix}\quad\begin{matrix}\pi_{2}\\ \vdots\\ X,\gamma,\beta\vdash\alpha\end{matrix}\quad\begin{matrix}\pi_{3}\\ \vdots\\ X,\gamma,\beta'\vdash\alpha\end{matrix}}{X,\gamma\vdash\alpha}\,\vee e$$

  and let τ_3 be the following proof.

$$\dfrac{\begin{matrix}\pi_{13}\\ \vdots\\ X,\gamma'\vdash\beta\vee\beta'\end{matrix}\quad\begin{matrix}\pi_{2}\\ \vdots\\ X,\gamma',\beta\vdash\alpha\end{matrix}\quad\begin{matrix}\pi_{3}\\ \vdots\\ X,\gamma',\beta'\vdash\alpha\end{matrix}}{X,\gamma'\vdash\alpha}\,\vee e$$

Now it is possible that $cutrank(\tau_2) = cutrank(\tau_3) = m$, but $\|\tau_2\| < \|\pi\|$ and $\|\tau_3\| < \|\pi\|$. Hence by induction hypothesis, there are proofs π_2' and π_3', both of cut rank $< m$, with conclusions $X, \gamma \vdash \alpha$ and $X, \gamma' \vdash \alpha$ respectively. We take π' to be the following proof:

$$\dfrac{\begin{matrix}\pi_{11}\\ \vdots\\ X\vdash\gamma\vee\gamma'\end{matrix}\quad \begin{matrix}\pi_2'\\ \vdots\\ X,\gamma\vdash\alpha\end{matrix}\quad \begin{matrix}\pi_3'\\ \vdots\\ X,\gamma'\vdash\alpha\end{matrix}}{X\vdash\alpha}\vee e$$

Now if π_{11} ends in a pure elimination,

$$cutrank(\pi') = max(cutrank(\pi_{11}), cutrank(\pi_2'), cutrank(\pi_3')) < m.$$

Otherwise $cutrank(\pi') \leqslant max(m-1, |\gamma \vee \gamma'|)$. But if π_{11} does not end in a pure elimination, $|\gamma \vee \gamma'| \leqslant cutrank(\pi_1) < m$, and it follows that $cutrank(\pi') < m$.

◀

▶ **Theorem 33** (Weak normalization). *If there is a derivation π of $X \vdash \alpha$ then there is a normal derivation ϖ of $X \vdash \alpha$. Further, if a formula $\alpha \vee \beta$ occurs as the major premise of an instance of $\vee e$ in ϖ, it also occurs as the major premise of an instance of $\vee e$ in π.*

Proof. For every derivation π, define $\mu(\pi)$ to be the pair (m, n) where $m = cutrank(\pi)$, and n is the number of subderivations of π of rank m. If $cutrank(\pi)$ is 0, π is already normal. If not, let $cutrank(\pi) = m > 0$ and let ϖ be a subderivation of π with conclusion $X' \vdash \beta$ such that $cutrank(\varpi) = m$ and no proper subderivation of ϖ is of rank $\geq m$. By Lemma 32, there is another derivation ϖ' with the same conclusion such that $cutrank(\varpi') < m$. Replace ϖ by ϖ' in π to get the proof π'. Now one subderivation of rank m has been eliminated in the process of going from π to π'. But we need to check that no new derivations of rank $\geq m$ have been introduced in π'. The only way this can happen is if ϖ' is not a pure elimination rule and is the major premise of an elimination rule or hybrid rule in π'. But then either $|\beta| < m$ or ϖ itself ends in a hybrid rule. In either case, no new subderivation of rank $\geq m$ gets introduced. Thus $\mu(\pi') < \mu(\pi)$. Since lexicographic ordering on pairs of natural numbers is a well order, by repeating the above process we eventually reach a proof of rank 0 – a normal proof, in other words.

Also note that the transformations in Lemma 32 do not introduce new formulas as major premises of $\vee e$, even though it might increase the number of instances of $\vee e$. This proves the second part of the theorem.

◀

▶ **Theorem 34** (Subformula property). *Let π be a normal derivation with conclusion $X \vdash \alpha$ and last rule* r. *Let $X' \vdash \beta$ occur in π. Then $X' \subseteq sf(X \cup \{\alpha\})$ and $\beta \in sf(X \cup \{\alpha\})$. Furthermore, if* r *is a pure elimination rule, then $X' \subseteq sf(X)$ and $\beta \in sf(X)$.*

Proof. The proof is by induction on the structure of π, and based on a case analysis on r. We present a few representative cases here.

- Suppose r is $\wedge i$. Then $\alpha = \alpha' \wedge \alpha''$ and π is of the following form:

$$\dfrac{\begin{matrix}\pi' & \pi'' \\ \vdots & \vdots \\ X \vdash \alpha' & X \vdash \alpha''\end{matrix}}{X \vdash \alpha}\wedge i$$

  Clearly $\alpha' \in \mathsf{sf}(\alpha)$ and $\alpha'' \in \mathsf{sf}(\alpha)$. Now either $X' = X$ and $\beta = \alpha$ or $X' \vdash \beta$ occurs in π' or π''. In the second and third cases, $X' \subseteq \mathsf{sf}(X \cup \{\alpha', \alpha''\})$ and $\beta \in \mathsf{sf}(X \cup \{\alpha', \alpha''\})$, by induction hypothesis. But $\mathsf{sf}(X \cup \{\alpha', \alpha''\}) \subseteq \mathsf{sf}(X \cup \{\alpha\})$, and hence we are done.

- Suppose r is $\rightarrow i$. Then $\alpha = \alpha' \rightarrow \alpha''$ and π is of the following form:

$$\dfrac{\begin{matrix}\pi_1 \\ \vdots \\ X, \alpha' \vdash \alpha''\end{matrix}}{X \vdash \alpha}\rightarrow i$$

  Clearly $\alpha' \in \mathsf{sf}(\alpha)$ and $\alpha'' \in \mathsf{sf}(\alpha)$. Now either $X' = X$ and $\beta = \alpha$ or $X' \vdash \beta$ occurs in π_1. In the latter case, by induction hypothesis $X' \subseteq \mathsf{sf}(X \cup \{\alpha', \alpha''\})$ and $\beta \in \mathsf{sf}(X \cup \{\alpha', \alpha''\})$. But $\mathsf{sf}(X \cup \{\alpha', \alpha''\}) \subseteq \mathsf{sf}(X \cup \{\alpha\})$, and hence we are done.

- Suppose r is $\vee e$. Then π is of the following form:

$$\dfrac{\begin{matrix}\pi_1 & \pi_2 & \pi_3 \\ \vdots & \vdots & \vdots \\ X \vdash \varphi \vee \psi & X, \varphi \vdash \alpha & X, \psi \vdash \alpha\end{matrix}}{X \vdash \alpha}\vee e$$

  Again, either $X' = X$ and $\beta = \alpha$ or $X' \vdash \beta$ occurs in one of the π_i's. Suppose it occurs in π_1. Notice that since π is normal, the last rule of π_1 is a pure elimination, and hence by induction hypothesis, $X' \subseteq \mathsf{sf}(X)$ and $\beta \in \mathsf{sf}(X)$. In particular, $\varphi \vee \psi \in \mathsf{sf}(X)$ and thus $\{\varphi, \psi\} \subseteq \mathsf{sf}(X)$. Now suppose that $X' \vdash \beta$ occurs in π_2 or π_3. Then we have $X' \subseteq \mathsf{sf}(X \cup \{\varphi, \psi, \alpha\}) \subseteq \mathsf{sf}(X \cup \{\alpha\})$ and $\beta \in \mathsf{sf}(X \cup \{\varphi, \psi, \alpha\}) \subseteq \mathsf{sf}(X \cup \{\alpha\})$, by induction hypothesis.

- Suppose r is $\rightarrow e$. Then π is of the following form (w.l.o.g.):

$$\dfrac{\begin{matrix}\pi' & \pi'' \\ \vdots & \vdots \\ X \vdash \varphi \rightarrow \alpha & X \vdash \varphi\end{matrix}}{X \vdash \alpha}\rightarrow e$$

  Again, either $X' = X$ and $\beta = \alpha$ or $X' \vdash \beta$ occurs in π' or π''. Since π is normal, π' ends in a pure elimination rule. Therefore for any $X' \vdash \beta$ occurring in π', $X' \subseteq \mathsf{sf}(X)$ and $\beta \in \mathsf{sf}(X)$. In particular, $\varphi \rightarrow \alpha \in \mathsf{sf}(X)$ and so $\{\varphi, \alpha\} \subseteq \mathsf{sf}(X)$. If $X' \vdash \beta$ occurs in π'', by induction hypothesis $X' \subseteq \mathsf{sf}(X \cup \{\varphi\}) \subseteq \mathsf{sf}(X)$ and $\beta \in \mathsf{sf}(X \cup \{\varphi\}) \subseteq \mathsf{sf}(X)$. Finally, as already shown, $\alpha \in \mathsf{sf}(X)$, as required for a proof ending in a pure elimination rule.

◀