*Under consideration for publication in Math. Struct. in Comp. Science*

# Complexity of Translations from Resolution to Sequent Calculus (*Presentation-Only - Draft*)

Giselle Reis[1] and Bruno Woltzenlogel Paleo[2]

[1] *Carnegie Mellon University, Doha, Qatar* (giselle@cmu.edu)
[2] *Australian National University, Canberra, Australia* (bruno.wp@gmail.com)

*Received 26 June 2017*

Resolution and sequent calculus are two well-known formal proof systems. Their differences make them more suitable for mutually distinct tasks. Resolution and its variants are very efficient for automated reasoning and are in fact the theoretical basis of many theorem provers. However, being intentionally machine-oriented, the resolution calculus is not as natural for human beings and the input problem needs to be pre-processed to clause normal form. Sequent calculus, on the other hand, is a modular formalism that is useful for analyzing meta-properties of various logics and is, therefore, popular among proof-theorists. The input problem does not need to be pre-processed, and proofs are more detailed. However, proofs also tend to be larger and more verbose. When the worlds of proof theory and automated theorem proving meet, translations between resolution and sequent calculus are often necessary. In this paper we compare three translation methods and analyze their complexity.

### Note to PxTP Reviewers

We intend to submit this paper to the special issue of Dale Miller's 60th Birthday. As the topic is within the scope of PxTP, we are submitting it for presentation only at PxTP. A shorter version of this work, containing only the first two translations was presented (without publication) in the first Women in Logic Workshop, in Reykjavik in June 2017. Our work on the third translation is ongoing, but we expect that it will be finished by the end of June.

## 1. Introduction

The representation of proofs as structured mathematical objects is in the core of proof theory. Nevertheless, there is no single best representation. Depending on what one needs proofs for, it makes sense to prefer one proof system over another. Two widely used formalisms are the resolution calculus and the sequent calculus.

Variants of resolution are used in many contemporary theorem provers due to their simplicity and efficiency in proof search. Simplicity is achieved by requiring the input

problem to be transformed to clause normal form (i.e. conjunction of disjunctions of literals that are either atomic formulas or negated atomic formulas), which allows the calculus to have only two inference rules (resolution and factoring). Efficiency in proof search is achieved by restricting instantiation through unification and by using various refinements that restrict the application of the inference rules while retaining completeness. As a result, proofs are relatively compact, but do not hold much information. To begin with, a resolution refutation is a proof of the *unsatisfiability* of the *negation of the theorem*. This means that the theorem is valid, but the refutation is not a direct validity proof. Then, since the need for a clause normal form requires modification of the conjecture in a number of ways (negations are pushed deeper, quantifiers are prenexified and skolemized, and disjunctions are distributed over conjunctions (or new symbols are introduced to avoid the exponential blow-up of the distribution)), it might be hard to map each resolution step into some insight about the original problem statement.

Sequent calculus was introduced by Gentzen as a meta-calculus to reason about natural deduction derivations and it continues to be used by most proof theorists for proof analysis and for meta-analysis of a logic's properties. In principle, the calculus is composed of two or more rules for each connective, which represent the semantics of the connective when it appears on a goal or on a hypothesis (e.g. $\wedge$ on a goal means one needs to prove two subgoal, whereas $\wedge$ on a hypothesis means one has two subhypotheses available to use). The proximity to a semantic interpretation makes it suitable to show the calculus' soundness and completeness. Proving the logic's consistency is also straightforward (usually a corollary of cut elimination). Additionally, sequent calculi have been used as proof systems for many different logics, as the formalism is modular and easily adaptable. The existence of many rules for connectives in different contexts makes it possible to work on a theorem without having to transform it. This characteristic also enables a better mapping of human reasoning steps to the formal proof steps. Consequently, much more information can be extracted from a sequent calculus proof. It is not a coincidence that many proof assistants (where proofs are constructed through scripts written by humans) are based on (higher-order) natural deduction and their basic tactic commands resemble sequent calculus rules.

At times, translations between the two systems are necessary. Below we discuss three situations with which we are familiarized.

Two methods for compressing sequent calculus proofs via the introduction of cuts were proposed in [Paleo, 2010] (atomic cuts) and [Hetzl et al., 2014] (first order cuts). In order to validate and evaluate the second method, the authors provided an implementation and performed experiments. For this task a large database of sequent calculus proofs was obtained by translating resolution proofs, although the paper provided no detailed description of the translation used.

As automated theorem provers are complex pieces of software and therefore vulnerable to bugs, the output of proofs is a common approach for providing a certificate that the solution given by the prover is correct, even if the prover itself might not be completely correct. This allows the user to trust the solution even without fully trusting the prover, if he or she successfully checks the proof. The *foundational proof certificate* initiative [Miller, 2013] adheres to this approach and proposes a conceptual framework to uniformly check

proofs in a variety of calculi and formats. Due to its versatility, a sequent calculus was chosen as the meta-calculus for this general proof-checking task and, in [Chihani et al., 2016] in particular, an embedding of resolution into LKF (focused sequent calculus for classical logic) was defined. Focusing is used to obtain a fine grained correspondence between the sequent calculus proof and the resolution proof, but it is not absolutely necessary. Leaving focusing aside, their embedding can be seen as a translation from resolution to regular sequent calculus.

A third possible situation in which a translation from resolution to sequent calculus proofs may arise is when proving soundness and completeness of one system with respect to the other. This is the goal in [Hermant, 2010].

All translations proposed in the situations mentioned previously appeared as a side-product of the main work, and thus not much attention has been paid to them. This paper fills this gap by defining all these transformations precisely (and with a common notation style) and analyzing their complexity.

## 2. Preliminaries

### 2.1. *Resolution*

Resolution is a calculus used for proving unsatisfiability of a formula in propositional or first-order (classical) logic. It works on skolemized formulas in conjunctive normal form (CNF) and it is used in most first-order automated theorem provers in some modified and extended form. A formula $F$ is unsatisfiable iff there exists a resolution refutation of $F$ (i.e. a derivation of the empty clause $\Box$ from the CNF of $F$) [Robinson, 1965]. Due to the duality between unsatisfiability and validity in classical logic, one can show the validity of a formula $F$ by presenting a resolution refutation of $\neg F$.

**Definition 1 (Resolution calculus).** Let $C_i$ be a disjunction of literals, $\overline{A}$ denote the dual of a literal $A$ and $\sigma$ be the most general unifier (m.g.u.) of $A$ and $A'$. The *resolution* and *factoring* rules of the resolution calculus are:

$$\frac{C_1 \vee A \vee C_2 \quad D_1 \vee \overline{A'} \vee D_2}{(C_1 \vee C_2 \vee D_1 \vee D_2)\sigma}\ R$$

$$\frac{C_1 \vee A \vee C_2 \vee A' \vee C_3}{(C_1 \vee A \vee C_2 \vee C_3)\sigma}\ F$$

**Example 1.** The following is the specification of a family of unsatisfiable clause sets, where $a, b$ are constants and $x_i$ are variables:

$$
\begin{aligned}
&p_1(x_1) \vee ... \vee p_n(x_n)\\
&q_1 \vee \neg p_1(a)\\
&\neg q_1 \vee \neg p_1(b)\\
&q_2 \vee p_1(x_1) \vee \neg p_2(a)\\
&\neg q_2 \vee p_1(x_1) \vee \neg p_2(b)\\
&\vdots\\
&q_n \vee p_1(x_1) \vee ... \vee p_{n-1}(x_{n-1}) \vee \neg p_n(a)\\
&\neg q_n \vee p_1(x_1) \vee ... \vee p_{n-1}(x_{n-1}) \vee \neg p_n(b)
\end{aligned}
$$

We take the clause set when $n = 2$ for our example:

$\{p_1(x_1) \vee p_2(x_2),\ q_1 \vee \neg p_1(a),\ \neg q_1 \vee \neg p_1(b),\ q_2 \vee p_1(x_1) \vee \neg p_2(a),\ \neg q_2 \vee p_1(x_1) \vee \neg p_2(b)\}$

The resolution refutation (with omitted parentheses) is:

$$\dfrac{\begin{matrix}\rho_1 & \rho_2\\ q_1 & \neg q_1\end{matrix}}{\square}\ R$$

Where $\rho_1^+$ is:

$$\dfrac{\begin{matrix}q_1 \vee \neg p_1 a & \overset{\eta_1}{p_1 x_1}\end{matrix}}{q_1}\ R$$

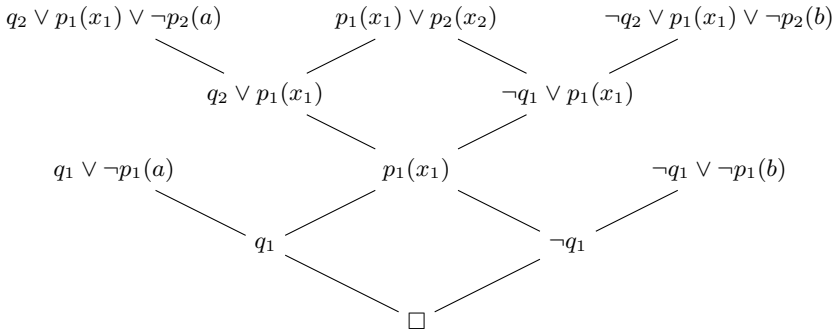
And $\rho_1-$ is:

$$\dfrac{\begin{matrix}\neg q_1 \vee \neg p_1 b & \overset{\eta_1}{p_1 x_1}\end{matrix}}{\neg q_1}\ R$$

And $\eta_1$ is:

$$\dfrac{\dfrac{\begin{matrix}q_2 \vee p_1 x_1 \vee \neg p_2 a & p_1 x_1 \vee p_2 x_2\end{matrix}}{q_2 \vee p_1 x_1}\ R, F \qquad \dfrac{\begin{matrix}\neg q_2 \vee p_1 x_1 \vee \neg p_2 b & p_1 x_1 \vee p_2 x_2\end{matrix}}{\neg q_2 \vee p_1 x_1}\ R, F}{p_1 x_1}\ R, F$$

Notice how the sub-derivation $\eta_1$ of $p_1(x_1)$ is used by both $\rho_1^+$ and $\rho_1^-$. By representing this directed acyclic graph (DAG) graphically, this can be seen more clearly:



It is straightforward to generalize the example DAG proof shown above for $n = 2$ to other values of $n$. For any $n$, the DAG proof is a tower of fixed width, height $2n$, length (i.e. number of nodes) $O(n)$ and size (i.e. number of symbols) $O(n^2)$. If, however, the DAG were expanded to a tree, its length and its size would be $\Omega(2^n)$ (i.e. an exponential blow up would occur), because the sub-derivations $\eta_k$ $(1 \leq k \leq n)$ would have to be duplicated.

Because of the worst-case exponential blow-up that can happen if DAG proofs are expanded to proof trees, as illustrated in Example 1, automated theorem provers invariably represent resolution proofs as DAGs during and after proof search. Since the transformations analyzed here use resolution proofs produced by automated theorem provers, their complexities will be parametrized by the length of resolution proofs as DAGs.

Moreover, the result of all transformations are sequent calculus proofs of the refuted clause set, i.e. skolemized and in CNF. Therefore, we shall not account for any blow-up on the input size due to normalization.

### 2.2. *Sequent Calculus*

Sequent calculus proof systems were proposed by Gentzen [Gentzen, 1969] in order to show strong normalization of propositional logic. Their adaptability to many logics and the uniformity with which one could prove the system's consistency made the formalism very popular among logicians.

A sequent is a structure $\Gamma \vdash \Delta$, where $\Gamma$ and $\Delta$ are multi-sets of formulas and $\vdash$ denotes the entailment relation. Its meaning is that the conjunction of the formulas in $\Gamma$ imply the disjunction of the formulas in $\Delta$. A sequent calculus is a collection of inference rules on sequents. In this paper we will use the sequent calculus **LK** for classical logic in Figure 1.

Note that we are using the additive version of the binary rules. For certain transformations, a multiplicative version of the cut-rule is needed (i.e. one that splits the conclusion contexts between the premises). In this case, we can safely assume the use of weakening, since this rule is structure-preserving admissible. We formally show this property after defining proof length[†].

**Example 2.** We show a sequent calculus proof for the same example as before. Since all transformations obtain proofs of the refuted clause set, we do the same, and show one of the possible proofs. We abbreviate the set $\{\forall x_1.\forall x_2.(p_1(x_1) \vee p_2(x_2)),\ q_1 \vee \neg p_1(a),\ \neg q_1 \vee \neg p_1(b),\ \forall x_1.(q_2 \vee p_1(x_1) \vee \neg p_2(a)),\ \forall x_1.(\neg q_2 \vee p_1(x_1) \vee \neg p_2(b))\}$ as $\Gamma$, and omit the parentheses.

[†] The dual argument also works: with multiplicative rules, contraction is structure-preserving admissible. Choosing additive rules maximizes the number of invertible rules and eases the definition of the translation described in Section 4.

$$\frac{}{\Gamma, A \vdash \Delta, A}\ \mathsf{init} \qquad \frac{\Gamma \vdash \Delta, P \quad \Gamma, P \vdash \Delta}{\Gamma \vdash \Delta}\ \mathsf{cut}$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot_l \qquad \frac{}{\Gamma \vdash \Delta, \top}\ \top_r$$

$$\frac{\Gamma, \neg P \vdash \Delta, P}{\Gamma, \neg P \vdash \Delta}\ \neg_l \qquad \frac{\Gamma, P \vdash \Delta, \neg P}{\Gamma \vdash \Delta, \neg P}\ \neg_r$$

$$\frac{P, Q, P \wedge Q, \Gamma \vdash \Delta}{P \wedge Q, \Gamma \vdash \Delta}\ \wedge_l \qquad \frac{\Gamma \vdash \Delta, P \wedge Q, P \quad \Gamma \vdash \Delta, P \wedge Q, Q}{\Gamma \vdash \Delta, P \wedge Q}\ \wedge_r$$

$$\frac{P, P \vee Q, \Gamma \vdash \Delta \quad Q, P \vee Q, \Gamma \vdash \Delta}{P \vee Q, \Gamma \vdash \Delta}\ \vee_l \qquad \frac{\Gamma \vdash \Delta, P \vee Q, P, Q}{\Gamma \vdash \Delta, P \vee Q}\ \vee_r$$

$$\frac{P \to Q, \Gamma \vdash \Delta, P \quad Q, P \to Q, \Gamma \vdash \Delta}{P \to Q, \Gamma \vdash \Delta}\ \to_l \qquad \frac{\Gamma, P \vdash \Delta, P \to Q, Q}{\Gamma \vdash \Delta, P \to Q}\ \to_r$$

$$\frac{P\{x \leftarrow \alpha\}, \exists x.P, \Gamma \vdash \Delta}{\exists x.P, \Gamma \vdash \Delta}\ \exists_l \qquad \frac{\Gamma \vdash \Delta, \exists x.P, P\{x \leftarrow t\}}{\Gamma \vdash \Delta, \exists x.P}\ \exists_r$$

$$\frac{P\{x \leftarrow t\}, \forall x.P, \Gamma \vdash \Delta}{\forall x.P, \Gamma \vdash \Delta}\ \forall_l \qquad \frac{\Gamma \vdash \Delta, \forall x.P, P\{x \leftarrow \alpha\}}{\Gamma \vdash \Delta, \forall x.P}\ \forall_r$$

$$\frac{P, P, \Gamma \vdash \Delta}{P, \Gamma \vdash \Delta}\ c_l \qquad \frac{\Gamma \vdash \Delta, P, P}{\Gamma \vdash \Delta, P}\ c_r$$

Fig. 1. **LK**: Sequent calculus for classical logic ($A$ is an atom, $\alpha$ is a variable not contained in $P$, $\Gamma$ or $\Delta$, and $t$ does not contain variables bound in $P$).

$$\dfrac{\dfrac{\dfrac{}{\Gamma, q_1 \vdash q_1}\ \mathsf{init}}{\Gamma, q_1, \neg q_1 \vdash}\ \neg_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, q_1, p_1 b \vdash p_1 b}\ \mathsf{init}}{\Gamma, q_1, \neg p_1 b, p_1 b \vdash}\ \neg_l \qquad \dfrac{}{\Gamma, q_1, \neg p_1 b, p_2 x_2 \vdash}}{\Gamma, q_1, \neg p_1 b, p_1 b \vee p_2 x_2 \vdash}\ \vee_l}{\Gamma, q_1, \neg p_1 b \vdash}\ \forall_l}{\Gamma, q_1 \vdash}\ \vee_l \qquad \dfrac{}{\Gamma, \neg p_1(a) \vdash}}{\Gamma \vdash}\ \vee_l$$

2.3. *Length measures*

Proofs are measured by number of nodes. Although the size of first order terms are not always negligible, all translations use the same term instantiation as the resolution proof and, therefore, term size has no impact in comparing them.

**Definition 2.** The *length* $|\psi|$ of a proof $\psi$ is the number of nodes in the proof. In the case of resolution, each node is a clause occurring in the DAG. In the case of sequent calculus, each node is a sequent occurring in the proof tree.

**Definition 3.** The *length* $|F|$ of a formula $F$ is the number of logical connectives occurring in $F$.

**Admissibility of weakening** We can now prove that weakening is structure-preserving admissible. The weakening rules are:

$$\frac{\Gamma \vdash \Delta}{P, \Gamma \vdash \Delta}\ w_l \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, P}\ w_r$$

We show that, given a proof of the premise, we can obtain a proof of the conclusion with exactly the same structure. This means that any use of the weakening rule during a proof is harmless for proof complexity, i.e. nodes corresponding to weakening can be ignored.

**Lemma 1.** Let $\varphi$ be a proof of $\Gamma \vdash \Delta$, then there exist proofs $\varphi_l$ and $\varphi_r$ of $\Gamma, P \vdash \Delta$ and $\Gamma \vdash \Delta, P$, respectively, such that $|\varphi| = |\varphi_r| = |\varphi_l|$.

*Proof sketch* The proof proceeds by structural induction on the proof tree. For the base case, we simply observe that $\Gamma, A \vdash \Delta, A$ is still provable if $\Gamma$ or $\Delta$ has an extra formula. For the inductive cases, on each rule we use the induction hypothesis to get an extra formula on its premise and apply the rule normally. □

## 3. First Translation: Resolutions as Cuts on the Resolved Literal

The translation described in this section is basically the one defined by the algorithm from [Hetzl et al., 2013]. One notable difference is that we consider the problem to be in CNF. Note that there is no complexity analysis in [Hetzl et al., 2013], and the transformation is presented as a pseudo-code without a formal definition.

Let $\mathcal{R}$ be a resolution refutation of a set of clauses $C_1^*, ..., C_n^*$ with free variables. The translation defined will transform $\mathcal{R}$ into an **LK** proof of the sequent $C_1, ..., C_n \vdash$, where $C_k$ is the universal closure of $C_k^*$. This is achieved by using the resolution refutation as a skeleton for the sequent calculus proof, where each clause is represented as a sequent and each resolution step is interpreted as an atomic cut. In order to do this, we must ground the resolution DAG (because the cut rule does not allow unification as the resolution rule does). In turn, grounding requires expansion of the DAG into a tree, because a DAG node that is used as a premise more than once may have to be distinctly instantiated more than once.

**Definition 4.** Let $\mathcal{R}$ be a resolution refutation DAG, we denote by $\widehat{\mathcal{R}}$ the representation of $\mathcal{R}$ as a ground derivation tree in the resolution calculus (Definition 1). This is achieved in two steps. First the DAG is transformed into a tree by duplicating sub-graphs with more than one parent. Grounding is achieved by propagating the m.g.u. computed at each rule application upwards, starting from the upper-most inferences.

**Theorem 1.** $|\widehat{\mathcal{R}}| = \Omega(2^{|\mathcal{R}|})$.

**Definition 5.** Let $C$ be a clause $\neg A_1 \vee \ldots \vee \neg A_n \vee B_1 \vee \ldots \vee B_m$ where $A_i$ $(1 \leq i \leq n)$ and $B_k$ $(1 \leq k \leq m)$ are atoms. Then $seq(C)$ is the sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$.

**Definition 6.** Let $\mathcal{R}$ be a resolution refutation DAG. We define $seq(\mathcal{R})$ as the **LK** proof obtained by taking $\widehat{\mathcal{R}}$ and transforming each clause $C$ into $seq(C)$, and interpreting resolution and factoring inferences as cut and contraction, respectively.

Observe that the cuts obtained from resolution inferences will be multiplicative. Nevertheless, we can obtain additive cuts using the weakening lemma (Lemma 1).

Note $seq(\mathcal{R})$ is an **LK** proof of the empty sequent $\cdot \vdash \cdot$ from non-tautological axioms. This structure is transformed into the desired proof of $C_1, ..., C_n \vdash$ via the operation of *context product*, defined as follows.

In the definition below, the notation $\circ$ denotes merging of sequents, i.e. $(\Gamma \vdash \Delta) \circ (\Lambda \vdash \Pi)$ is $\Gamma, \Lambda \vdash \Delta, \Pi$.

**Definition 7 (Context product).** Let $T$ be a sequent and $\varphi$ be an **LK** derivation with end-sequent $S$ such that no free variable in $T$ occurs as eigen-variable in $\varphi$. We define the *context product* $T \star \varphi$ (which yields a derivation of $T \circ S$) inductively:

— If $\varphi$ consists only of an axiom, then $T \star \varphi$ is composed of one sequent: $T \circ S$.
— If $\varphi$ ends with a unary rule $\xi$:

$$\dfrac{\begin{array}{c}\varphi' \\ S'\end{array}}{S}\ \xi$$

then $T \star \varphi$ is defined as:

$$\dfrac{\begin{array}{c}T \star \varphi' \\ T \circ S'\end{array}}{T \circ S}\ \xi$$

Since $T$ does not contain free variables which are eigen-variables of $\varphi$, the context product is well defined also if $\xi$ is $\forall_r$ or $\exists_l$, although this case does not occur in our application.

— If $\varphi$ ends with a binary rule $\xi$:

$$\dfrac{\begin{array}{cc}\varphi_1 & \varphi_2 \\ S_1 & S_2\end{array}}{S}\ \xi$$

then $T \star \varphi$ is defined as:

$$\dfrac{\begin{array}{cc}T \star \varphi_1 & T \star \varphi_2 \\ T \circ S_1 & T \circ S_2\end{array}}{T \circ S}\ \xi$$

Since $T$ is part of the context, and all binary rules are additive in the calculus considered, their formulas are copied to both premises. $S_1$ and $S_2$ differ from each other at most on auxiliary formulas.

The result of $(C_1, ..., C_n \vdash) \star seq(\mathcal{R})$ is a derivation of $C_1, ..., C_n \vdash$ from axioms $(C_1, ..., C_n \vdash) \circ seq(C'_i)$, where $C'_i$ is a ground instance of $C_i$ (see first case of the inductive definition). These axioms are tautological and can be proved easily.

**Theorem 2.** Let $C_1, ..., C_n$ be the universally closed formulas of a refutable clause set. Let $C'_i$ be a ground instance of one of those formulas. Then the sequent $(C_1, ..., C_n \vdash) \circ seq(C'_i)$ has an **LK** proof $\varphi$ such that $|\varphi| = O(|C_i|)$.

*Proof.* To see that the sequent is provable, it suffices to observe that all atoms occurring in $seq(C_i')$ occur in the defined sequent in a dual position because of $C_i$ on the left (remember that in $seq(C_i')$ positive atoms are on the right and negative atoms are on the left).

A proof of the sequent can be obtained by instantiating the variables properly and decomposing $C_i$ exhaustively, until its atomic parts. Since this formula has $|C_i|$ connectives, this will be the length of the proof. □

All those steps are summarized in the definition of the transformation below.

**Definition 8.** Let $\mathcal{R}$ be a resolution refutation DAG of a clause set $C_1^*, ..., C_n^*$. We define $T_L(\mathcal{R})$ as the **LK** proof obtained from $(C_1, ..., C_n \vdash) \star seq(\mathcal{R})$, where each $C_i$ is the universal closure of $C_i^*$ and all axioms are proved (according to the proof of Theorem 2.

### 3.1. *Complexity*

**Theorem 3.** If $\mathcal{R}$ is a resolution refutation, then $|T_L(\mathcal{R})| = \Omega(2^{|\mathcal{R}|})$ in the worst case.

*Proof.* The first step of $T_L(\mathcal{R})$ consists of obtaining $seq(\mathcal{R})$, which requires expanding the DAG. According to Theorem 1, this operation may be cause an exponential blow-up in the length of the proof. □

## 4. Second Translation: Resolutions as Cuts on the Resolvent

The second translation of resolution to sequent calculus analyzed here is essentially the one used in the *foundational proof certificates* (FPC) framework for checking resolution proofs [Chihani et al., 2016], with only one minor difference. Whereas the FPC framework uses a single-sided focused sequent calculus, here the two-sided calculus for classical logic without focusing is used.

This transformation also translates resolution steps into cuts, but this time the cut formula is the resolvent, instead of the resolved atom. The key idea is that each resolution step deriving a clause $C_k^*$ from clauses $C_i^*$ and $C_j^*$ can be represented in sequent calculus by a derivation of the following form

$$\dfrac{\overset{\varphi_k}{\Delta \vdash C_k} \quad \overset{\vdots}{\Delta, C_k \vdash}}{\Delta \vdash} \; cut$$

where $C_h$ denotes the universal closure of clause $C_h^*$ for any $h$, $\Delta$ is a set of formulas containing $C_i$ and $C_j$. $\Delta \vdash C_k$ is provable (not surprisingly, because $C_k^*$ is derived from $C_i^*$ and $C_j^*$, which are in $\Delta$ and resolution is sound and sequent calculus is complete), and the construction of its proof $\varphi_k$ is explained in the demonstration of Theorem 4. On the right branch, the same construction is repeated for the next resolvent, and this continues until the empty clause is reached, in which case the right branch can be closed by the rule $\bot_l$. The translation procedure based on this idea is formally defined below.

**Definition 9.** Let $\mathcal{R}$ be a resolution refutation of the clause set $C_1^*, ..., C_n^*$, with resolvents $C_{n+1}^*, ..., C_{n+m}^*$ (possibly partially instantiated) and $C_{n+m}^* = \bot$. Letting $\Gamma$ be the

set of universally closed clauses $C_1, \ldots, C_n$, the sequence of proofs $\psi_j$ $(0 \leq j < m)$ is defined as :

$$\frac{\overset{\varphi_{j+1}}{\Gamma, \ldots, C_{n+j} \vdash C_{n+j+1}} \quad \overset{\psi_{j+1}}{\Gamma, \ldots, C_{n+j+1} \vdash}}{\Gamma, \ldots, C_{n+j} \vdash} \; cut$$

with $\psi_m$ being defined as:

$$\frac{}{\Gamma, C_{n+1}, ..., C_{n+m} \vdash} \; \bot_l$$

Finally, $T_R(\mathcal{R})$ is defined as the sequent calculus proof $\psi_0$ of $\Gamma \vdash$.

**Theorem 4.** Let $C_1^*$ and $C_2^*$ be two clauses that resolve to $C_3^*$, and let $C_i$ denote the universal closure of a clause $C_i^*$. Then the sequent $C_1, C_2 \vdash C_3$ has an **LK** proof $\varphi$ such that $|\varphi| = O(|C_1| + |C_2| + |C_3|)$.

*Proof.* The proof $\varphi$ can be constructed in a bottom-up manner as follows. Begin by instantiating the quantified variables of $C_3$ and decomposing $C_3$ until only atoms are left. Then instantiate the variables of $C_1$ and $C_2$ using either the eigen-variable used for $C_3$ or terms used in the unifier of the resolution step that derives $C_3^*$ from $C_1^*$ and $C_2^*$. Finally, apply $\vee_l$ to $C_1$ and then to $C_2$ exhaustively. Note that after $C_1$ is completely decomposed, all branches will be closed (the dual atom is available from $C_3$), except the one that contains the resolved literal. This is continued by the decomposition of $C_2$ and eventually the dual of the resolved atom will be in the sequent.

The total number of nodes in this proof is equal to the number of connectives and universal quantifiers occurring in $C_1$, $C_2$ and $C_3$. Therefore, $|\varphi| = O(|C_1| + |C_2| + |C_3|)$. □

Due to the weakening lemma (Lemma 1), the sequent $\Gamma, C_1, C_2 \vdash \Delta, C_3$ is provable.

### 4.1. *Complexity*

In this translation, the resolution refutation DAG does not need to be expanded. Each resolution step in the DAG is translated to a single cut in the sequent calculus proof. Resolvents become universally closed clauses in the antecedent of the sequent in the right branch of the proof being constructed, which remain in the context and can be reused as many times as needed. Consequently, the length of the sequent calculus proof is linear on the length of the DAG resolution refutation.

**Theorem 5.** Let $\mathcal{R}$ be a DAG resolution refutation of a set of clauses such that $k$ is the size of the biggest universally closed clause. Then $|T_R(\mathcal{R})| = O(|\mathcal{R}| * k)$ in the worst case.

*Proof.* The DAG resolution refutation $\mathcal{R}$ contains at most $O(|\mathcal{R}|)$ resolution steps (each node is either an input clause or the result of a resolution step). In the result of $T_R(\mathcal{R})$, each of these steps will be a cut, whose left branch has a proof of size at most $3*k$ (considering that $k$ is the size of the biggest universally closed clause). Consequently, $|T_R(\mathcal{R})| = O(|\mathcal{R}| * k)$. □

## 5. Third Translation: Resolutions as Axioms

The third translation defined and analyzed here is inspired by (and essentially the same as) the translation used in the proof of relative soundness of Resolution Modulo (i.e. Extended Narrowing and Resolution, more precisely) with respect to (cut-free) Sequent Calculus Modulo in [Hermant, 2010] (which was itself inspired by works on the inverse method) For the sake of simplicity, deduction modulo is left aside, as it is (like focusing, in the case of the second translation) outside the scope and unnecessary to the goals of this paper.

The key idea is to translate a resolution step between $p \vee q$ and $\neg q \vee r$ in the DAG refutation into an axiom/init inference with conclusion $\Gamma, q \vdash q$ in the corresponding sequent calculus proof. This is done recursively, starting with the bottom-most node in the DAG refutation and traversing the DAG upward, following any topological order. For the bottom-most node (containing the empty clause), the resulting sequent calculus proof is the trivial derivation of $\bot \vdash$ using the $\bot_l$ rule. For the $n$-th node $\eta_n$ deriving $C_n^*$ from $D_n^*$ and $E_n^*$ with resolved atom $q$, the resulting sequent calculus proof $S_n$ is a derivation of $\Gamma_n \vdash$, where $\Gamma_n = (\Gamma_{n-1} \backslash \{C_n\}) \cup \{D_n \text{ and } E_n\}$, and is obtained from $S_{n_1}$ by replacing occurrences of $C_n$ by $D_n$ and $E_n$ and adding a new branch starting with an axiom having main formula $q$ (cf. [Hermant, 2010], Lemma 14).

**Note to PXTP reviewers:** Work on this section is ongoing. A formal definition of the translation will be included here. In the meanwhile, please refer to section 8 of Hermant's paper (`http://www.cri.ensmp.fr/people/hermant/docs/Resolution_CutFree.pdf`).

### 5.1. *Complexity*

**Note to PXTP reviewers:** We conjecture that there is a sequence of resolution proofs $R_n$ with length $O(n^k)$ for which the third translation results in cut-free sequent calculus of length $\Omega(2^n)$ whereas the first translation produces sequent calculus proofs with atomic cuts of length $O(n^k)$. We plan to use the sequence of proofs shown in [Paleo, 2010] to prove this conjecture.

## 6. Discussion

In the previous sections, we have surveyed and compared (from a complexity perspective) three different translations from resolution to sequent calculus. The first one translates resolution steps to cuts having (the grounding of) the resolved atom as cut-formula. The second one translates resolution steps to cuts having (a universal closure of) the whole resolvent as the cut-formula. And, finally, the third one translates resolution steps to axiom/init inferences having the resolved atom as main formula.

To ease comparison, all three translations were defined here using the same resolution calculus and the same simple sequent calculus, with neither focusing nor deduction modulo. This is the first time that the first translation is formally defined. And we hope that the re-definition of the second and third translations using simpler calculi will make

them more accessible to people working on applications where focusing and deduction modulo are not essential.

Complexity-wise, the second translation is clearly superior to the other two. It avoids the expansion and the worst-case exponential blow-up by using universally quantified cuts. However, it is important to note that, if these cuts were eliminated (using Gentzen's cut-elimination procedure), duplications (leading to a worst-case exponential blow-up) would happen whenever a cut inference has to be moved above a series of contractions and universal quantification inferences that occur in the translated sequent calculus proofs when a resolvent is used more than once in the DAG resolution refutation.

The other two translations have their advantages too, which become clear when we look at the contexts in which they were developed. The first translation was developed in the context of proof analysis, where there is an interest in extracting Herbrand sequents or expansion trees and in generating potentially interesting new lemmas.For these goals, a sequent calculus proof without quantified cuts is essential. For the third translation, the goal was to prove relative soundness of a resolution calculus with deduction modulo. A soundness proof under the assumption that cut is admissible was already known, but cut-admissibility in sequent calculi with deduction modulo is tricky and depends on the rewrite system. By providing a direct translation to a cut-free sequent calculus, the third translation strengthened the soundness for the resolution calculus with deduction modulo, making it independent of assumptions about the rewrite system.

Finally, we hope that the comparison pursued here will shed some additional light on the debate of whether resolution steps are better seen as cuts or axioms. Two of the three known translations see resolution steps as cuts, albeit as cuts of crucially different kinds; the third translation sees resolutions as axioms and, in [Hermant, 2010], the view of resolution steps as cuts is claimed to be confusing and misleading.

In our (subjective) view, the first translation is the most straightforward: the resulting cut preserves as much as possible the local structure of the resolution step (i.e. exploiting a natural analogy, a resolution with resolved atom $p$ becomes a cut with cut-formula $p\sigma$ for some $\sigma$ and with a context that is (an instantiation of) a super-set of the context in the resolution step); however, the need for the substitution $\sigma$ requires expansion of the DAG and hence breaks the global structure of the proof. The preservation of the local structure is a strong support for the *resolution-as-cut* view. On the other hand, the breaking of the global structure is a clear (although unsurprising) indication that the analogy between resolution steps and cuts is not perfect. This imperfection is closely related to Hermant's observation that resolution is forward-chaining, whereas sequent calculus is backward-chaining. It is resolution's forward-chaining nature that naturally gives rise to non-tree DAG proofs.

The second translation also uses cuts for resolution steps, but in a way that does not exploit the natural analogy between resolution steps and cuts. The whole sequent calculus is used more as a *meta* calculus in this translation, with one premise of the cut storing information that a whole resolvent is derivable from previously derived clauses and the other premise continuing the procedure, now with that resolvent added to the derived clauses. The global structure of the proof is preserved and a shorter polynomially bound sequent calculus proof is obtained, at the cost of having more complex cut-formulas.

In the third translation, a resolution step resolving a literal in a clause and its dual in another clause becomes an axiom/init inference connecting these two literals. However, it is important to note that, in fact, each resolution step may become *several* axiom/init inferences. This one-to-many correspondence speaks against the *resolution-as-axiom* view (although note that the first translation also suffers this problem to a lesser extent). Moreover, from a complexity perspective, the third translation is the longest proofs, possibly exponentially longer than those obtained with the first translation.

We conjecture that all three translations are essentially the same (for a suitable notion of "essentially"), but in different stages of cut-elimination. If we start with the second translation and partially eliminate the quantified cuts until only atomic cuts are left, the resulting sequent calculus proof might be essentially the same as the proof obtained through the first translation. Furthermore, if we then eliminate the atomic cuts completely, the result might be essentially the same as the proof obtained through the third translation. In each of the cut-elimination steps (i.e. from quantified cuts to atomic cuts, and then to no cuts), an exponential blow-up in proof length may occur.

**Acknowledgements:** Bruno would like to thank Gilles Dowek for pointing out, back in 2010, that resolution steps could also be translated as axioms, right after a talk about cut-introduction by resolution [Paleo, 2010] given by Bruno at INRIA-Paris to Gilles Dowek's research team.

## References

Zakaria Chihani, Dale Miller, and Fabien Renaud. A semantic framework for proof evidence. *Journal of Automated Reasoning*, pages 1–44, 2016. ISSN 1573-0670. .

Gerhard Gentzen. Investigations into logical deductions. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.

Olivier Hermant. Resolution is cut-free. *Journal of Automated Reasoning*, 44(3):245–276, 2010. ISSN 1573-0670. .

Stefan Hetzl, Tomer Libal, Martin Riener, and Mikheil Rukhaia. *Understanding Resolution Proofs through Herbrand's Theorem*, pages 157–171. Springer, 2013. ISBN 978-3-642-40537-2. .

Stefan Hetzl, Alexander Leitsch, Giselle Reis, and Daniel Weller. Algorithmic introduction of quantified cuts. *Theoretical Computer Science*, 549:1 – 16, 2014. ISSN 0304-3975. .

Dale Miller. Foundational proof certificates: making proof universal and permanent. In *Proceedings of the Eighth ACM SIGPLAN International Workshop on Logical Frameworks & Meta-languages: Theory & Practice, LFMTP*, pages 1–2, 2013. .

Bruno Woltzenlogel Paleo. Atomic cut introduction by resolution: Proof structuring and compression. In *Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Revised Selected Papers*, pages 463–480, 2010. .

J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965. ISSN 0004-5411. .