

THE COMPUTATIONAL COMPLEXITY OF PROPOSITIONAL CIRQUENT CALCULUS

MATTHEW S. BAUER

University of Illinois at Urbana-Champaign, U.S.A.
e-mail address: msbauer2@illinois.edu

ABSTRACT. Introduced in 2006 by Japaridze, cirquent calculus is a refinement of sequent calculus. The advent of cirquent calculus arose from the need for a deductive system with a more explicit ability to reason about resources. Unlike the more traditional proof-theoretic approaches that manipulate tree-like objects (formulas, sequents, etc.), cirquent calculus is based on circuit-style structures called *cirquents*, in which different peer (sibling, cousin, etc.) substructures may share components. It is this resource sharing mechanism to which cirquent calculus owes its novelty (and its virtues). From its inception, cirquent calculus has been paired with an abstract resource semantics. This semantics allows for reasoning about the interaction between a resource provider and a resource user, where resources are understood in their most general and intuitive sense. Interpreting resources in a more restricted computational sense has made cirquent calculus instrumental in axiomatizing various fundamental fragments of *Computability Logic*, a formal theory of (interactive) computability. The so-called “classical” rules of cirquent calculus, in the absence of the particularly troublesome *contraction* rule, produce a sound and complete system CL5 for Computability Logic. In this paper, we investigate the computational complexity of CL5, showing it is Σ_2^p -complete. We also show that CL5 without the *duplication* rule has polynomial size proofs and is NP-complete.

1. INTRODUCTION

Introduced in 2006 by Japaridze [7], cirquent calculus is a refinement of classical sequent calculus. The advent of cirquent calculus arose from the need for a deductive system with a more explicit ability to reason about resources. Unlike the more traditional proof-theoretic approaches that manipulate tree-like objects (formulas, sequents, etc.), cirquent calculus is based on circuit-style structures called *cirquents*, in which different peer (sibling, cousin, etc.) substructures may share components. It is this resource sharing mechanism to which cirquent calculus owes its novelty (and its virtues). Cirquents come in a variety of forms. One way to characterize the sort of cirquents studied in this paper in familiar terms is to say that a cirquent is a multiset of sequents (called groups) where each formula — more precisely, each occurrence of a formula — may simultaneously belong to more than one

2012 ACM CCS: [Theory of computation]: Logic - Proof theory; Computational complexity and cryptography - Problems, reductions and completeness.

Key words and phrases: cirquent calculus, computability logic, resource semantics, proof theory, substructural logics.

sequent. This explains the origin of the word *cirquent*, which is a hybrid of *circuit* and *sequent*. From its inception, *cirquent calculus* has been paired with an abstract resource semantics. This semantics allows for reasoning about the interaction between a resource provider and a resource user, where resources are understood in their most general and intuitive sense.

Interpreting resources in a more restricted computational sense has made *cirquent calculus* instrumental in axiomatizing various fundamental fragments of *Computability Logic*, a formal theory of (interactive) computability. The so-called “classical” rules of *cirquent calculus*, in the absence of the particularly troublesome *contraction* rule, produce a sound and complete system CL5 for *Computability Logic*. Born in [2], *Computability Logic (CoL)* is an ambitious research program aimed at developing a formal theory of interactive computability. To this end, formulas of CoL represent computational problems modeled at games. The notion of “truth” for such formulas becomes synonymous with the notion of “computability” for the computational problems they represent.

While CL5 has shown to be sound and complete with respect to Japaridze’s abstract resource semantics, it has also shown to validate a strictly larger class of formulas than affine logic (sequent calculus without the contraction rule), the latter being well studied and sound as a logic of resources. However, due to no shortcomings in effort, a complete resource aware semantics has not been found for affine logic. This has led Japaridze to conclude that “CL5 rather than affine logic adequately materializes the resource philosophy traditionally associated with the latter”. Indeed, the “semantics before syntax” philosophy upon which CL5 was conceived, together with its completeness result provide compelling evidence in this direction.

In parallel with the development of *cirquent calculus*, recent work [3–6, 10] has introduced sound and complete axiomatizations for several fragments of CoL, the most fundamental of these being CL4, which contains the propositional connectives \neg (negation), \vee (parallel disjunction), \wedge (parallel conjunction), \sqcup (choice disjunction) and \sqcap (choice conjunction) as well as the “choice” quantifiers \sqcup and \sqcap and the “blind” quantifiers \forall and \exists . For a full discussion of these operators and the semantics of CoL, see [9]. While the semantics of CoL differs drastically from that of classical logic, the set of valid formulas of CL4 is identical to that of classical logic when restricted to its operators and the so-called “elementary” sort of atoms. Thus, the presence of the quantifiers \forall and \exists within CL4’s language guarantees its undecidability. In [6], however, the \forall, \exists -free fragment of CL4 was shown to be decidable in polynomial space. A PSPACE-completeness proof was later given in [1].

The deductive apparatus of CL4 is far removed from traditional Gentzen or Hilbert style axiomatizations. Further, the system does not extend very naturally to other fragments of *Computability Logic*. A rectification to this shortcoming came with the advent of *cirquent calculus*. As mentioned previously, the fragment of *cirquent calculus* not containing the *contraction* rule, produces a sound and complete system for the fragment of *Computability Logic* known as CL5. This system contains the operators \vee , \wedge and \neg and in this paper we show it to be Σ_2^p -complete.

We further investigate the complexity of the logical system described by the rules of CL5, but with the absence of the *duplication* rule. We call this logic CL5[−] and show that every theorem in CL5[−] has a polynomial size proof. This effectively places the problem of deciding CL5[−] provability in NP. A reduction from the k vertex cover problem to CL5[−] provability is then given, solidifying the logic as NP-complete. In some sense, CL5 is the CoL “counterpart” to multiplicative linear logic, the former being a proper extension of

the latter. Here, $CL5^-$ lies strictly between multiplicative affine logic (linear logic with weakening) and $CL5$. Among the virtues of multiplicative affine logic is that, unlike the coNP-complete classical logic, it is in NP. $CL5^-$ thus presents a nice and natural extension of multiplicative affine logic that gets “closer” to classical logic while still remaining in NP.

With the tight relationship between cirquent calculus and Computability Logic, one should not overlook the potential impact of cirquent calculus on other areas of logic and proof theory. In [19], Xu constructs a cirquent calculus based system for the propositional fragment of independence friendly (IF) logic, allowing one to account for independence from propositional connectives in the same vein that traditional IF logic accounts for independence from quantifiers. In [8] a cirquent calculus based proof system was developed which, among other benefits, yields polynomial size proofs for all instances of the pigeonhole principle. To date, this is the first proof system to achieve such a result without leveraging cut, extension or substitution.

This paper is organized as follows. In section 2, we provide an introduction to the rules of cirquent calculus and describe the systems $CL5$ and $CL5^-$. Section 3 studies the system $CL5^-$ and proves its NP-completeness. In the final section, we show the full $CL5$ to be Σ_2^p -complete by a reduction from the TQBF- Σ_2 problem.

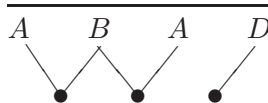
2. CORE CIRQUENT CALCULUS

The present section provides the technical definitions and information necessary for understanding the results of this paper. For a full discussion of the concepts of cirquent calculus, see [7, 9]. As noted previously, a **formula** of the language of cirquent calculus is syntactically identical to that of classical propositional logic, where formulas are built from the connectives \neg , \wedge and \vee as well as non-logical **atoms** (also known as propositional letters). The only difference between the two being that the language of cirquent calculus does not contain logical atoms such as \top or \perp ; the latter can simply be understood as abbreviations of $P \wedge \neg P$ and $P \vee \neg P$, respectively, for some (whatever) atom P . Our language further mandates that \neg can only be applied to atoms. As always, a **literal** is an atom with or without the prefix \neg .

A k -ary **pool**, for $k \in \mathbb{N}$, is a sequence $Pl = \langle F_1, F_2, \dots, F_k \rangle$ of formulas. The formulas in a pool are not required to be unique. We refer to a particular occurrence of a formula as an **oformula**. A k -ary **structure**, for $k \in \mathbb{N}$, is a finite sequence $St = \langle \Gamma_1, \Gamma_2, \dots, \Gamma_m \rangle$ for $m \geq 0$, where each Γ_i , called a **group** of St , is a subset of $\{1, 2, \dots, k\}$. It is permitted that $\Gamma_i = \Gamma_j$ for $i \neq j$ and hence we use the term **ogroup** to refer to a particular occurrence of a group. We are now ready to give our definition for a cirquent.

Definition 2.1. [7] A k -ary ($k \geq 0$) **cirquent** is a pair $C = (St^C, Pl^C)$, where St^C , called the structure of C , is a k -ary **structure**, and Pl^C , called the **pool** of C , is a k -ary pool.

For example, let $Pl^C = \langle A, B, A, D \rangle$ and $St^C = \langle \{1, 2\}, \{2, 3\}, \{4\} \rangle$. Here, C has 4 oformulas and 3 ogroups and is typically represented by the diagram below.

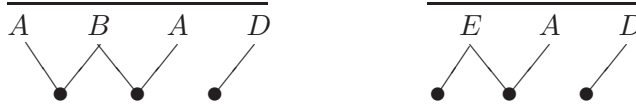


Following [7], we will adopt the notion that a diagram simply “is” (rather than just “represents”) a cirquent. When an ogroup Γ is connected with an arc to an oformula G , we say that Γ **contains** G .

Before defining the rules for our inference system, let us first give some additional terminology necessary for the definition of our rules. For a given cirquent, two oformulas F and G are called **adjacent** if G is positioned immediately to the right of F . In such a case, it is said that F **immediately precedes** G and G **immediately follows** F . **Merging two adjacent ogroups** Γ and Δ in a given cirquent C means replacing in C the two ogroups Γ and Δ by a single ogroup $\Gamma \cup \Delta$. The rightmost cirquent below represents the cirquent that results from merging the first and second ogroups in the leftmost cirquent.



Merging two adjacent oformulas F and G into a new oformula H is the result of replacing F and G by H and redirecting to it all of the arcs that were pointing to F or G . The rightmost cirquent below represents the cirquent obtained from merging, in the leftmost cirquent, (the first) A and B into a single oformula E .



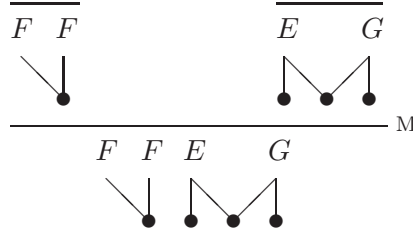
With these definitions in mind, we are now equipped to present the so-called “core” cirquent calculus rules. The rules we give here are only those relevant to our language and are by no means exhaustive. In its full generality, CoL encompasses numerous other logical operators all accompanied by a deep and meaningful semantics. As predicted earlier in [7], recent works have produced cirquent calculus axiomatizations for logics containing some of the more powerful operators of Computability Logic. See, for example, [10, 11].

2.1. Axioms (A). Axioms are rules with an empty set of premises. In cirquent calculus, they come in two forms: the *empty cirquent axiom* and the *identity axiom*. Both varieties of the axiom rule are illustrated below. It is important to note that the *identity axiom* is actually a scheme of infinitely many axioms, as F stands for an arbitrary formula.



As will be our convention throughout, the letter placed to the right of the horizontal line represents the rule by which the conclusion was obtained.

2.2. **Mix (M)**. The *mix* rule takes two premises. Its conclusion is obtained by placing each of the two premise cirquents side by side in a single cirquent, as illustrated below.

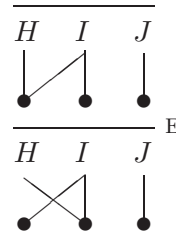
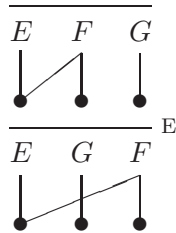


The remaining rules described in subsections 2.3 to 2.8 all take a single premise cirquent.

2.3. **Exchange (E)**. The *exchange* rule comes in two varieties, *oformula exchange* and *ogroup exchange*. The conclusion of the *oformula exchange* rule is obtained by swapping the positions of two adjacent oformulas in the premise cirquent. The *ogroup exchange* rule is similar in that it allows two adjacent ogroups in a premise cirquent to exchange positions in the conclusion. In both varieties of the rule, the arcs from each ogroup to its oformulas should be preserved. Below is an example of *oformula* (resp. *ogroup*) *exchange* in which the oformulas F and G (resp. ogroups #1 and #2) are swapped.

oformula exchange

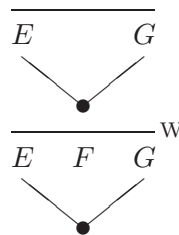
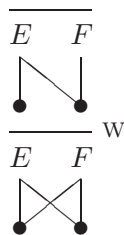
ogroup exchange



2.4. **Weakening (W)**. The *weakening* rule has two forms; *ogroup weakening* and *pool weakening*. In the case of *ogroup weakening*, the conclusion is obtained from the premise by adding a new arc between a pre-existing ogroup and oformula pair. In an application of *pool weakening*, the conclusion is obtained by inserting a new oformula at any position in the pool of the premise.

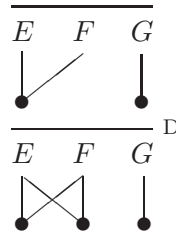
ogroup weakening

pool weakening

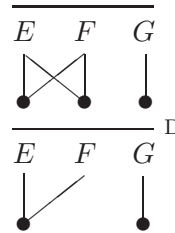


2.5. Duplication (D). The *duplication* rule can be applied in one of two ways, called *downward duplication* and *upward duplication*. The conclusion of the *downward duplication* rule is obtained from its premise by replacing an ogroup with two adjacent ogroups that each have arcs to exactly the same oformulas as the original ogroup. An application of *upward duplication* works in the opposite direction in that the premise cirquent is obtained by replacing an ogroup in the conclusion by two adjacent ogroups that are both identical to the original ogroup.

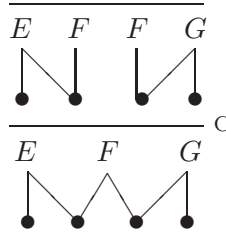
downward duplication



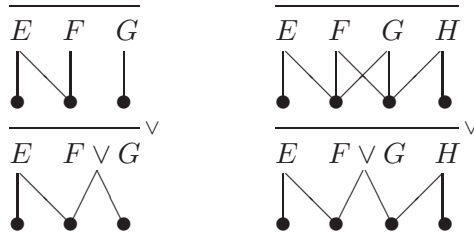
upward duplication



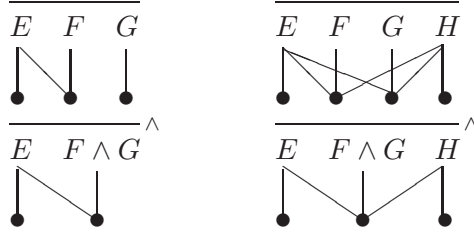
2.6. Contraction (C). The *contraction* rule takes a premise cirquent which contains two adjacent and identical oformulas. The conclusion of an application of this rule is obtained from the premise by merging two adjacent and identical oformulas F and F into a single oformula F .



2.7. \vee -introduction (\vee). The conclusion of this rule is obtained from the premise by merging two adjacent oformulas F and G into a single oformula $F \vee G$ such that all arcs pointing to either F or G now point to $F \vee G$. Two examples of an application of this rule are given below.



2.8. **\wedge -introduction (\wedge).** This rule takes a premise cirquent that contains two adjacent oformulas F and G such that no ogroup contains both F and G and every ogroup that contains F (resp. G) is immediately followed (resp. preceded) by an ogroup containing G (resp. F). The conclusion in an application of this rule is obtained from its premise by merging each ogroup that contains F with the ogroup containing G that immediately follows it. The oformulas F and G should then merge into $F \wedge G$. We again give two examples below.



2.9. **The systems CCC, CL5 and CL5⁻.** The cirquent calculus system built from all eight of the above rules has been aptly named “Classical Cirquent Calculus”, or **CCC**. By removing the *contraction* rule from CCC, we get the system **CL5**. We further let **CL5⁻** denote the system that results from additionally removing the *duplication* rule from CL5.

Let S be one of the cirquent calculus systems **CCC**, **CL5** or **CL5⁻**. A **proof** of a cirquent C in S is a tree of cirquents whose root is C where each node follows from its children by one of the rules of S . When we say a formula F is provable in S , we mean the cirquent containing a single oformula F with one ogroup and arc is provable in S . In this paper, for simplicity, we are only interested in proving formulas, even though all of our results almost straightforwardly extend from formulas (as special cases of cirquents) to all cirquents. Correspondingly, we agree that, unless suggested otherwise by the context, “provability” means “provability of formulas”. It was shown in [7] that the provable formulas of **CCC** coincide exactly with those of classical propositional logic. Interestingly enough, the provable formulas of **CL5** can also be described in a very natural way. Because we will rely on this result and its associated concepts in several of our later proofs, we shall give the relevant details here.

A **substitution** for a formula C is a function σ that maps every atom P in C to some formula $\sigma(P)$. If $\sigma(P)$ is an atom for every P in C , then we say that σ is an **atomic-level substitution**. This notion can be extended to all formulas by requiring that $\sigma(\neg P) = \neg\sigma(P)$, $\sigma(F \vee G) = \sigma(F) \vee \sigma(G)$ and $\sigma(F \wedge G) = \sigma(F) \wedge \sigma(G)$. Let A and B be formulas. B is said to be an **instance** of A iff there exists a substitution σ such that $\sigma(B) = A$. If σ is an atomic level substitution, then B is an **atomic-level instance** of A . A formula is called **binary** iff no atom has more than two occurrences in it. A binary formula is said to be **normal** iff, whenever an atom occurs twice in the formula, one occurrence is positive and the other is negative. The following theorem is a combination of Theorem 12 and Lemma 9 of [7].

Theorem 2.2. (*Japaridze*) *A formula is provable in CL5 iff it is an instance of a binary tautology iff it is an atomic-level instance of a normal binary tautology.*

At this point, a semantical characterization of the provable formulas of $\mathbf{CL5}^-$ has yet to be given. In light of the results of this paper, such a finding would perhaps be very interesting.

3. CL5 WITHOUT DUPLICATION IS NP-COMplete

In this section, we show that deciding provability for a formula of the logic $\mathbf{CL5}^-$ is NP-complete. This result is achieved by a combination of two theorems, the first of which states that every formula F provable in $\mathbf{CL5}^-$ has a proof whose size is polynomial in the size of F . Towards this goal, we begin by presenting a set of technical lemmas.

Lemma 3.1. *Every formula provable in $CL5^-$ is an atomic-level instance of a normal binary tautology.*

Proof. This follows immediately from the “ \Rightarrow ” direction of Theorem 2.2, given that the theorems of $CL5^-$ form a subset of the theorems of $CL5$. \square

Lemma 3.2. *Let F be a formula provable in $CL5^-$ with n positive occurrences of atoms. No $CL5^-$ proof of F can contain a cirquent with more than n ogroups.*

Proof. Let F and n be as in the condition of the lemma and let Δ be a proof tree for F . Assume for a contradiction that Δ has a cirquent D that contains more than n ogroups. Now let Δ' be the subtree of Δ rooted at D . A careful examination of the rules of $CL5^-$ will show that, no matter which series of rules are applied in Δ' , the total number of ogroups in all the leaves of Δ' must be greater than or equal to the number of ogroups in D . This is because the premise cirquent(s) in the application of a $CL5^-$ rule must, together, contain at least as many ogroups as the conclusion. Every leaf in Δ' must be derived by either the *identity axiom* or the *empty cirquent axiom* and thus cannot contain more than 1 ogroup. This means Δ' has at least $n + 1$ leaves with a single ogroup, each of which must follow by the *identity axiom*. To be a consequence of the *identity axiom*, a cirquent must contain exactly two oformulas K and $\neg K$. But the total number of positive occurrences of atoms in oformulas in the leaves of Δ cannot exceed n . This is because, by our assumption, the conclusion of Δ contains a single oformula with n positive literals. Further, the oformulas of the premise cirquent(s) in the application of a $CL5^-$ rule cannot contain more positive occurrences of atoms than the oformulas of the conclusion. This means at most n applications of the *identity axiom* are possible, contradiction. \square

The **length of a formula** F is defined as the total number of occurrences of literals and connectives in F . The **size of a group** is the total number of oformulas it contains. The **size of a cirquent** is then the sum of the lengths of the oformulas in its pool plus the sum of the sizes of each ogroup in its structure. Naturally, the **size of a proof** is the sum of the sizes of the cirquents it contains.

Lemma 3.3. *Let F be a formula provable in $CL5^-$ and let k be the length of F . There exists a $CL5^-$ proof of F using $O(k^6)$ applications of rules.*

Proof. Let F be a formula and let Δ be a $CL5^-$ proof tree for F . Given Δ , we show that a new proof Δ' can be obtained from Δ such that Δ' proves F using $O(k^6)$ applications of rules.

We begin by noting that any proof containing applications of the *empty cirquent axiom* can be transformed into one in which no applications of the rule are made¹. Indeed, observe that the empty cirquent can only be a (“dummy”) premise of *mix*. In such a case, the conclusion of *mix* is simply the same as its other premise. Therefore, the empty cirquent can be deleted and the application of *mix* can be skipped. This can be done for all empty cirquents contained in the proof until none remain. Let Δ_1 be the proof that results from removing all applications of the *empty cirquent axiom* in the preceding manner.

Observe now that Δ_1 must contain exactly one application of *conjunction introduction* or *disjunction introduction* for each occurrence of \wedge or \vee in F , respectively. This is because every application of either rule introduces, in its conclusion, a single \wedge or \vee connective that cannot be later removed by the application of any CL5^- rule. The number of positive occurrences of atoms in F also bounds the number of applications of the *identity axiom*. This is because, in a bottom up view of a proof, no CL5^- rule allows formulas to be removed or merged. With the number of applications of *conjunction introduction*, *disjunction introduction* and the *identity axiom* all bounded by $O(k)$ in Δ_1 , we can also bound the number of applications of the *mix* rule. To see this, first note that because Δ_1 contains no applications of the *empty cirquent axiom*, every leaf node must be derived by the *identity axiom*. That is, there are $O(k)$ paths from root to leaf in Δ_1 . Viewing the proof tree in a top down fashion, *mix* is the only CL5^- rule which allows the proof tree to “branch”, given it is the only rule with more than one premise. Thus, each application of *mix*, which must take two premises, increases the number of paths from root to leaf by 1. Because the number of such paths is bounded by $O(k)$, so too must be the number of applications of the *mix* rule.

Each application of *pool weakening* introduces, in its conclusion, an oformula that must be present (perhaps only as a proper subformula of some oformula) in the conclusion of Δ_1 . This is because no CL5^- rule can remove, in its conclusion, an oformula contained in its premise cirquent. Thus we have an immediate bound of $O(k)$ for the number of applications of *pool weakening* in Δ_1 . Observe that this also implies a bound of $O(k)$ on the number of oformulas contained in any cirquent. To bound the number of possible applications of *ogroup weakening* in Δ_1 , notice that no cirquent in Δ_1 can contain more than k^2 arcs. This is because, by Lemma 3.2, k bounds the maximum number of ogroups in any cirquent and, as noted previously, $O(k)$ bounds the number of oformulas in any cirquent. Further, *conjunction introduction* and *disjunction introduction* are the only CL5^- rules whose conclusion can contain fewer arcs than its premise. The number of applications of these rules in Δ_1 does not exceed $O(k)$ and each application can remove no more than the maximum k^2 arcs contained in any premise cirquent. Thus, no more than $O(k^3)$ total arcs can be removed from premise to conclusion for all applications of *conjunction* and *disjunction introduction* in Δ_1 . Each application of *ogroup weakening* creates a single arc in its conclusion that was not present in its premise. As the conclusion of Δ_1 contains a single arc, no more than $O(k^3)$ arcs can be created from premise to conclusion in Δ_1 by applications of *ogroup weakening*.

Every leaf of Δ_1 must be derived by the *identity axiom*, which is applied at most $O(k)$ times in Δ_1 . That is, there are at most $O(k)$ paths from root to leaf in Δ_1 . Each such path Δ_1^i for $1 \leq i \leq O(k)$ contains at most $O(k^3)$ applications of CL5^- rules other than *oformula* or *ogroup exchange*. Thus, any Δ_1^i can contain at most $O(k^3)$ sequences

¹The only purpose of this axiom in [7] is to ensure the provability of the empty cirquent itself; everything else is provable without using the empty cirquent axiom.

Φ_w for $0 \leq w \leq O(k^3)$ where each $\Phi_w := C_1^w, C_2^w, \dots, C_n^w$ (for $n \in \mathbb{N}$) is a sequence of cirquents such that every cirquent C_i^w follows from C_{i+1}^w by *exchange* (of either sort) for all $0 \leq i < n$. In Δ_1 , each Φ_w can be of arbitrary length. We show, however, that a new proof Δ' can be obtained from Δ_1 such that each Φ_w uses no more than $O(k^2)$ applications of *oformula* and *ogroup exchange*. This is because, the last cirquent C_n^w in any Φ_w can always be obtained from C_1^w using no more than $O(k^2)$ applications of *oformula* and *ogroup exchange*. By lemma 3.2, the first cirquent C_1^w can contain at most k ogroups. Moving all of these ogroups to an arbitrary position in C_n^w can be done with $O(k^2)$ applications of *ogroup exchange*. Additionally, moving each of $O(k)$ possible oformulas in C_1^w to an arbitrary position in C_n^w also requires no more than $O(k^2)$ applications of *oformula exchange*. This means that every sequence Φ_w in Δ_1 can be replaced with a new sequence Φ'_w such that Φ'_w uses no more than $O(k^2)$ applications of *oformula* or *ogroup exchange*. Let Δ' be the $CL5^-$ proof tree that results from replacing each Φ_w in Δ_1 by Φ'_w . Δ' must again contain $O(k)$ paths from root to leaf. Each such path now has a bound of $O(k^3) \times O(k^2)$ applications of *oformula* or *ogroup exchange*, resulting in a total bound of $O(k^6)$ applications of *exchange* in all branches of Δ' .

It should be clear that if Δ proves F , then Δ' will also prove F . The number of applications of non-*exchange* rules does not change from Δ_1 to Δ' and remains $O(k^3)$. Further, Δ' uses no more than $O(k^6)$ applications of *exchange*. Totaling the number of applications of every rule in Δ' , we obtain a bound of $O(k^3) + O(k^6)$. \square

Theorem 3.4. *Let F be a formula provable in $CL5^-$ and let k be the size of F . There exists a $CL5^-$ proof of F whose size is polynomial in k .*

Proof. By lemma 3.3, there exists a $CL5^-$ proof Δ of F that uses a polynomial number of rule applications. Further, by lemma 3.2, the size of any cirquent in Δ must be polynomial in k . Thus, the maximum size of each cirquent multiplied by the maximum number of cirquents in Δ yields a polynomial bound on the size of Δ . \square

The result of Theorem 3.4 effectively places the provability problem for $CL5^-$ formulas in NP. We further this result by additionally showing that the problem is NP-complete. Before giving the proof, we solidify some standard concepts that will be used within it. A **graph** $G = (V, E)$ is an ordered pair made up of a set of vertices (V) and edges (E), with each edge being an unordered pair of vertices. The **degree** of a vertex v in G , denoted $\deg(v)$, is defined as the number of edges incident to v . Given a graph G , a **vertex cover** is $U \subseteq V$ such that every edge of E is incident to at least one vertex in U . In complexity theory, the **vertex cover problem** can be stated as a decision problems as follows. Given a graph G and number k , does G have a k vertex cover, i.e. a vertex cover using at most k vertices? It is well known that this problem is NP-complete.

Theorem 3.5. *Deciding provability for the logic $CL5^-$ is NP-complete.*

Proof. It follows from Theorem 3.4 that $CL5^-$ provability is in NP. To see it is NP-hard, we give a polynomial time reduction f to it from the vertex cover problem. Fix some arbitrary graph $G = (V, E)$ and some $k \in \mathbb{N}$. The reduction, borrowed from [12], follows.

$$f(V, E, k) := (\Psi(k)) \vee (\Theta(V, E)) \vee (\Omega(E))$$

$$\Psi(k) := \underbrace{q \vee q \vee \dots \vee q}_{\text{total of } k \text{ literals}}$$

$$\Theta(V, E) := \underbrace{(\neg q \wedge (\underbrace{\neg v_1 \vee \neg v_1 \vee \dots \vee \neg v_1}_{\text{deg}(v_1)}) \vee \dots \vee (\neg q \wedge (\underbrace{\neg v_n \vee \neg v_n \vee \dots \vee \neg v_n}_{\text{deg}(v_n)})))}_{\text{for each vertex } v_1, v_2, \dots, v_n \in V}$$

$$\Omega(E) := \underbrace{(e_1^1 \vee e_1^2) \wedge (e_2^1 \vee e_2^2) \wedge \dots \wedge (e_m^1 \vee e_m^2)}_{\text{for each edge } e_1, e_2, \dots, e_m \in E}$$

Above, q is a new atom that differs from all other atoms in the formula. In $\Omega(E)$, e_i^1 and e_i^2 are the vertices on the endpoints of edge e_i for $1 \leq i \leq m$. It is also important to note that each v_j for $1 \leq j \leq n$ represents both an atom in $f(V, E, k)$ as well as the label of a vertex in G .

Obviously the mapping f is computable in polynomial time. It now remains to show that a graph (V, E) has a k vertex cover if and only if the formula $f(V, E, k)$ is provable in CL5^- .

“ \Rightarrow ” Our reduction is identical to that of [12] (Section 5.2) for multiplicative affine (direct) logic. As a direct consequence of Theorem 3 from [7], CL5^- proves every formula provable in multiplicative affine logic², so the result follows immediately.

“ \Leftarrow ” Assume $f(V, E, k)$ is provable in CL5^- . We need to show that there exists a k vertex cover of $G = (V, E)$. By Theorem 2.2, there exists a normal binary tautology β and an atomic-level substitution σ such that $\sigma(\beta) = f(V, E, k)$. Let Ψ' be the subformula of β such that $\sigma(\Psi') = \Psi(k)$. Let Π be the set of the k atoms of Ψ' , and let $\Sigma \subseteq \Pi$ be the set of those members of Π that have (not only positive but also) negative occurrences of β . For each $l \in \Sigma$, let Θ_l be the conjunct in the $\Theta(V, E)$ component of $f(V, E, k)$ that contains $\neg\sigma(l)$. By our construction, each Θ_l takes the form

$$(\neg q \wedge (\neg v_j \vee \neg v_j \vee \dots \vee \neg v_j))$$

for $1 \leq j \leq n$. Define a vertex cover $V' \subset V$ for G as the set of all vertices in G labeled v_j where v_j is contained in Θ_l for some $l \in \Sigma$.

To see that V' is indeed a k vertex cover for G , note first that σ is an atomic-level substitution, and hence each positive literal $l \in \Sigma$ must be mapped to a unique positive occurrence of q in $f(V, E, k)$. By the definition of $\Psi(k)$, however, there are exactly k positive occurrences of q in $f(V, E, k)$. This means Σ contains no more than k atoms and subsequently the vertex cover derived from Σ contains no more than k vertices.

Because $f(V, E, k)$ is an atomic-level instance of β , both formulas take exactly the same form, the only difference being in the atoms they are built from. Let us now define a model $*$ for β . For each $l \in \Pi$ let $l^* = \perp$. For any atom $l \notin \Pi$ with the property that $\sigma(l) = q$, let $l^* = \top$. Further, for each atom l of β where $\sigma(l) = a$ for some $a \in V'$, let $l^* = \top$. All remaining atoms l of β should be interpreted as $l^* = \perp$.

Note that every atom l of the earlier defined Φ' is interpreted as $l^* = \perp$, meaning that $(\Phi')^* = \perp$. Next let Θ' be the subformula of β such that $\sigma(\Theta') = \Theta(V, E)$. Each disjunct of Θ' takes the form $\neg l_1 \wedge (\neg l_2 \vee \neg l_3 \vee \dots \vee \neg l_n)$ for some positive integer n . If $l_1 \notin \Sigma$ then

²From Theorem 3 of [7], we have affine logic = CL5^* , where CL5^* is the system CL5 with the limitation that cirquents contained in proofs do not have groups that share formulas. Because *upward* (resp. *downward*) *duplication* is only applicable when the premise (resp. conclusion) of the rule is a cirquent in which groups share formulas, all cirquents provable in CL5^* are also provable in CL5^- .

$l_1^* = \top$ and $\neg(l_1^*) = \perp$, making the entire subformula evaluate to \perp under $*$. If $l_1 \in \Sigma$ then $l_1^* = \perp$ and $\neg(l_1^*) = \top$. Note, however, that when $l_1 \in \Sigma$ then $\sigma(l_m) = a$ where $a \in V'$ and $m \in \{2, 3, \dots\}$. This means $l_m^* = \top$ and $\neg(l_m^*) = \perp$, again making the subformula evaluate to \perp under $*$. Because each disjunct of Θ' evaluates to \perp under $*$, we have $(\Theta')^* = \perp$.

Finally, let Ω' be the subformula of β such that $\sigma(\Omega') = \Omega(E)$. Because $\beta = (\Psi') \vee (\Theta') \vee (\Omega')$ is a tautology where $(\Psi')^* = \perp$ and $(\Theta')^* = \perp$, we have $(\Omega')^* = \top$. Each conjunct of Ω' takes the form $l_1 \vee l_2$ such that $\sigma(l_1)$ and $\sigma(l_2)$ are the endpoints of an edge in E . Further, by our construction, every edge in E is represented by such a conjunct in Ω' . Observe that, by our definition of $*$, we have $l^* = \top$ for an atom l of Ω' only when $\sigma(l) \in V'$. Thus, Ω' is true only when every edge in E has an endpoint in V' . That is, V' is a vertex cover of G . \square

4. CL5 IS Σ_2^p -COMPLETE

This section contains our main result, namely, the Σ_2^p -completeness of the decision problem for provability of formulas in **CL5**.

Lemma 4.1. *Deciding provability for CL5 is in Σ_2^p .*

Proof. The following is a Σ_2^p algorithm that, in view of Theorem 2.2, decides provability of a formula G in CL5. On input G , existentially guess a binary formula F such that G is an instance of F . Then, universally guess a truth assignment $*$ for F . If F is true under $*$, accept. Otherwise, reject. \square

Let TQBF- Σ_2 be the problem of deciding truth for a quantified Boolean formula of the form $\exists X \forall Y \Theta$, where X and Y are sequences of variables and Θ is a quantifier-free Boolean formula all of whose variables are among X or Y . As shown in Theorem 4.1 of [15], this problem is Σ_2^p -complete. We say an atom z is **isolated** in a formula ϕ if there is only a single (positive or negative) occurrence of z in ϕ . Otherwise it is **non-isolated**.

Theorem 4.2. *Deciding provability for the logic CL5 is Σ_2^p -complete.*

Proof. By lemma 4.1, the problem of deciding provability for CL5 is in Σ_2^p . To show it is Σ_2^p -hard, we construct a polynomial time mapping reduction f from TQBF- Σ_2 to CL5-provability. We can safely restrict our attention to instances of TQBF- Σ_2 where the Boolean portion of the formula is in disjunctive normal form, as the complexity of the problem is not reduced with these limitations. Our reduction follows.

Let ϕ be a formula that takes the form $\exists X \forall Y \Theta$ with X and Y being sets of variables and Θ being a boolean formula in disjunctive normal form all of whose variables are among $X \cup Y$. The following steps are used to construct the corresponding CL5 formula $f(\phi)$.

- (1) For each $z \in X$, let k^z and t^z be the number of positive occurrences of the literals z and $\neg z$ in Θ , respectively. Define

$$g(z) = Z^z \wedge \underbrace{(Z^z \rightarrow u_1^z \wedge u_2^z \wedge \dots \wedge u_{k^z}^z)}_{k^z \text{ literals}} \wedge \underbrace{(Z^z \rightarrow \neg v_1^z \wedge \neg v_2^z \wedge \dots \wedge \neg v_{t^z}^z)}_{t^z \text{ literals}}$$

where Z^z , $u_1^z, u_2^z, \dots, u_{k^z}^z$ and $v_1^z, v_2^z, \dots, v_{t^z}^z$ are all fresh³ pairwise distinct variables, uniquely chosen for z . Here, if k^z (resp. t^z) is 0, the second (resp. third) conjunct should be

³Here, fresh variables are those not occurring elsewhere in ϕ .

omitted. Now let $z_1, z_2, \dots, z_l = X$ and let ϕ_0 be the formula $(g(z_1) \wedge g(z_2) \wedge \dots \wedge g(z_l)) \rightarrow \Theta_1$. Here Θ_1 is a formula derived from Θ such that, for each $z \in X$, every positive occurrence of z in Θ is replaced by a unique literal from $u_1^z, u_2^z, \dots, u_k^z$ and every occurrence of $\neg z$ in Θ is replaced by a unique literal from $\neg v_1^z, \neg v_2^z, \dots, \neg v_t^z$.

- (2) Consider any $y \in Y$. Let r_y be the number of positive occurrences of y in ϕ_0 , and s_y be the number of negative occurrences. For each pair i, j with $1 \leq i \leq r_y$ and $1 \leq j \leq s_y$, we choose a fresh and unique variable $P_{i,j}^y$. Now, define $f(\phi)$ to be the result of replacing in ϕ_0 , for each $y \in Y$, every positive occurrence of the literal y by $(P_{i,1}^y \vee P_{i,2}^y \vee \dots \vee P_{i,s_y}^y)$ and every (positive) occurrence of the literal $\neg y$ by $(\neg P_{1,j}^y \vee \neg P_{2,j}^y \vee \dots \vee \neg P_{r_y,j}^y)$ where $1 \leq i \leq r_y$ (resp. $1 \leq j \leq s_y$) is unique for each replacement of an occurrence of y (resp. $\neg y$).

“ \Rightarrow ” Assume ϕ is true. Then there exists some truth assignment $^\circ : X \rightarrow \{\top, \perp\}$ such that Θ^* is true under any truth assignment * that extends $^\circ$ to $X \cup Y$. By our assumption, Θ is in disjunctive normal form and hence takes the form

$$(\psi_1 \wedge \psi_2 \wedge \dots \wedge \psi_n) \vee \dots \vee (\psi_{m+1} \wedge \psi_{m+2} \wedge \dots \wedge \psi_{m+l})$$

where each ψ_w is either a positive or negative literal. Now let Ω and Σ represent the antecedent and consequent of the outermost implication in $f(\phi)$ such that $f(\phi) = \Omega \rightarrow \Sigma$. Our construction guarantees that Σ takes the form

$$(\Psi_1 \wedge \Psi_2 \wedge \dots \wedge \Psi_n) \vee \dots \vee (\Psi_{m+1} \wedge \Psi_{m+2} \wedge \dots \wedge \Psi_{m+l})$$

where each Ψ_w is a disjunction of literals (such a “disjunction” may have only a single “disjunct”). That is, Σ is obtained from Θ by replacing each literal ψ_w by Ψ_w , where Ψ_w is a disjunction of literals. We will henceforth use ψ_w to represent a unique position in Θ and Ψ_w to represent the corresponding position in Σ .

We want to show that $f(\phi)$ is provable in CL5. By Theorem 2.2, it suffices to show that $f(\phi)$ is an instance of a binary tautology. We construct a binary tautology Φ of which $f(\phi)$ is an instance. Namely, we let Φ be the formula obtained from $f(\phi)$ as follows. For each $z \in X$, if $z^\circ = \top$ (resp. \perp) replace the third (resp. second) occurrence of Z^z by an atom Q^z such that Q^z does not occur elsewhere in Φ and is unique for each $z \in X$. It should be easy to see that Φ is a quantifier free binary formula and $f(\phi)$ is an instance of Φ . We need only show that Φ is a tautology. Again notice that Φ takes form $\Phi = \Pi \rightarrow \Sigma$ where Σ is the same as in $f(\phi)$.

Given $X = z_1, z_2, \dots, z_l$, we have $\Pi = g(z_1)' \wedge g(z_2)' \wedge \dots \wedge g(z_l)'$ where each $g(z)'$ matches one of the following forms⁴.

- (1) $Z^z \wedge (Z^z \rightarrow u_1^z \wedge u_2^z \wedge \dots \wedge u_k^z) \wedge (Q^z \rightarrow \neg v_1^z \wedge \neg v_2^z \wedge \dots \wedge \neg v_t^z)$
- (2) $Z^z \wedge (Q^z \rightarrow u_1^z \wedge u_2^z \wedge \dots \wedge u_k^z) \wedge (Z^z \rightarrow \neg v_1^z \wedge \neg v_2^z \wedge \dots \wedge \neg v_t^z)$

For any $g(z)'$ in Π where $z \in X$, if $(Z^z)^* = \perp$ then $\Pi^* = \perp$, and hence $\Phi^* = \top$. If $(Z^z)^* = \top$, then all of the literals in the consequent of the implication in $g(z)'$ with antecedent literal Z^z must be true under * , otherwise we will have $\Pi^* = \perp$ and again $\Phi^* = \top$. Thus, we need only guarantee $\Sigma^* = \top$ under truth assignments such that, for each conjunct $g(z)'$ in Π , Z^z and every literal in the consequent of the implication containing antecedent literal Z^z are true under * .

Let Θ' be the formula that results from replacing in Θ every positive occurrence of z or $\neg z$, where $z \in X$, by its truth value under $^\circ$. For example, if $z^\circ = \perp$, replace $\neg z$ by

⁴It is possible that the third (resp. second) conjunct in formula 1 (resp. 2) is absent.

\top and z by \perp . It should be easy to see that Θ' is a tautology. For any position ψ_w that contains z (resp. $\neg z$) in Θ and \top in Θ' , the position Ψ_w in Σ must contain a single literal u_i^z (resp. $\neg v_i^z$) such that $(u_i^z)^* = \top$ (resp. $(\neg v_i^z)^* = \top$) if $\Pi^* = \top$. This is because, by our construction, if $z^\circ = \top$ (resp. $(\neg z)^\circ = \top$), every u_i^z for $1 \leq i \leq k^z$ (resp. $\neg v_i^z$ for $1 \leq i \leq t^z$) must occur in the consequent of the implication in Π with non-isolated antecedent Z^z . For reasons already discussed, such a literal must be true under any truth assignment $*$ where $\Pi^* = \top$. Define Σ' as the formula such that, for each occurrence of \top or \perp in position ψ_w of Θ' , the literal in position Ψ_w of Σ is replaced by the same value (\top or \perp) as ψ_w . Note that Σ' only substitutes the logical atom in position Ψ_w of Σ when Ψ_w contains a positive or negative occurrence of an atom in X . Since our goal is to show that Σ is true under truth assignments that make Π true, we need only show that Σ' is a tautology.

For a contradiction, assume Σ' is not a tautology. Then there exists some truth assignment \dagger defined on the variables of Σ' such that $(\Sigma')^\dagger = \perp$. We define a truth assignment \ddagger for Θ' as follows. If a variable $y \in Y$ is such that, for some i with $1 \leq i \leq r_y$, $(P_{i,1}^y \vee \dots \vee P_{i,s_y}^y)^\dagger = \perp$, we let $y^\ddagger = \perp$; otherwise we let $y^\ddagger = \top$. It is not hard to see that, if a subformula of Σ' in a position Ψ_w is false under \dagger , then the subformula of Θ' in the corresponding position ψ_w is false under \ddagger (but not necessarily vice versa). This, in view of the monotonicity of \vee and \wedge , obviously implies that $(\Theta')^\ddagger = \perp$, because Σ' and Θ' have the same (\vee, \wedge) -structures. Now we are dealing with a contradiction, because the tautological Θ' cannot be false.

“ \Leftarrow ” Assume $f(\phi)$ is provable in CL5. By Theorem 2.2, for some normal binary tautology Φ there exists an atomic level substitution σ such that $\sigma(\Phi) = f(\phi)$. Let $\Phi = \Pi \rightarrow \Sigma$, where Π and Σ represent the antecedent and consequent of the outermost implication in Φ . We want to define a truth assignment $^\circ : X \rightarrow \{\top, \perp\}$ such that Θ^* is true for any truth assignment $*$ that extends $^\circ$ to $X \cup Y$. In what follows, we define such a partial truth assignment $^\circ$ for Θ while concurrently defining a partial truth assignment $^\bullet$ for Φ .

Procedure 1 - Consider a $z \in X$ and let A be the atom of Φ such that the first occurrence of Z^z in $f(\phi)$ **originates** from A (i.e., A gets replaced by $\sigma(A) = Z^z$) when transitioning from Φ to $\sigma(\Phi) = f(\phi)$.

Case 1: The second and third occurrences of Z^z in $f(\phi)$ originate from A and B , respectively (for some $B \neq A$ in Φ). Define $z^\circ = \top$, $A^\bullet = \top$ and $B^\bullet = \perp$. The consequent⁵ of the second occurrence of A in Φ should take the form $a_1 \wedge a_2 \wedge \dots \wedge a_{kz}$ for some positive literals a_1, a_2, \dots, a_{kz} in Φ and the consequent of B should take the form $\neg b_1 \wedge \neg b_2 \wedge \dots \wedge \neg b_{tz}$ for some negative literals $\neg b_1, \neg b_2, \dots, \neg b_{tz}$. Let all of a_1, a_2, \dots, a_{kz} and b_1, b_2, \dots, b_{tz} be true under $^\bullet$.

Case 2: The second and third occurrences of Z^z in $f(\phi)$ originate from B and A , respectively (for some $B \neq A$ in Φ). Define $z^\circ = \perp$, $A^\bullet = \top$ and $B^\bullet = \perp$. The consequent of the second occurrence of A in Φ should take the form $\neg b_1 \wedge \neg b_2 \wedge \dots \wedge \neg b_{tz}$ for some negative literals $\neg b_1, \neg b_2, \dots, \neg b_{tz}$ and the consequent of B should take the form $a_1 \wedge a_2 \wedge \dots \wedge a_{kz}$ for some positive literals a_1, a_2, \dots, a_{kz} . Let all of a_1, a_2, \dots, a_{kz} and b_1, b_2, \dots, b_{tz} be false under $^\bullet$.

Case 3: There are only two occurrences of Z^z in $f(\phi)$, both of which originate from A . If the consequent of the second occurrence of A in Φ takes the form $a_1 \wedge a_2 \wedge \dots \wedge a_{kz}$ for some positive literals a_1, a_2, \dots, a_{kz} then define $z^\circ = \top$ and $A^\bullet = \top$. Further, let all of

⁵Here and later in similar contexts, the “consequent of W ” should be understood as the consequent of the implication whose antecedent is W . Similarly for “antecedent of W ”.

a_1, a_2, \dots, a_{kz} be true under \bullet . If the consequent of the second occurrence of A in Φ takes the form $\neg b_1 \wedge \neg b_2 \wedge \dots \wedge \neg b_{tz}$ for some negative literals $\neg b_1, \neg b_2, \dots, \neg b_{tz}$ then define $z^\circ = \perp$ and $A^\bullet = \top$. Further, let all of b_1, b_2, \dots, b_{tz} be false under \bullet .

Case 4: If none of the cases 1 – 3 are satisfied, then the second and third occurrences of Z^z in $f(\phi)$ must originate from B and C , respectively (where $A \neq B$ and $A \neq C$). Define $z^\circ = \perp$, $A^\bullet = \top$ and $B^\bullet = C^\bullet = \perp$. Further, let every atom in the consequent of B evaluate to \perp under \bullet and every atom in the consequent of C evaluate to \top under \bullet .

As we remember, Θ in disjunctive normal form, taking the form

$$(\psi_1 \wedge \psi_2 \wedge \dots \wedge \psi_n) \vee \dots \vee (\psi_{m+1} \wedge \psi_{m+2} \wedge \dots \wedge \psi_{m+l})$$

where each ψ_w is a literal. It is also the case that Σ takes the form

$$(\Psi_1 \wedge \Psi_2 \wedge \dots \wedge \Psi_n) \vee \dots \vee (\Psi_{m+1} \wedge \Psi_{m+2} \wedge \dots \wedge \Psi_{m+l})$$

where each Ψ_w is a disjunction of literals (possibly with just a single “disjunct”). Thus, we will use our previous convention in which ψ_w represents a unique position in Θ and Ψ_w represents the corresponding position in Σ .

Our construction guarantees that for each position ψ_w in Θ containing some positive (resp. negative) literal z (resp. $\neg z$) such that $z \in X$, Ψ_w contains a positive (resp. negative) literal a (resp. $\neg a$). By *procedure 1* we have $z^\circ = a^\bullet$. As in the previous direction, let Θ' be the formula that results from replacing, in Θ , for every $z \in X$, all positive occurrences of the literals z and $\neg z$ by their truth values under \circ . Let Σ' be the formula such that, for each position ψ_w in Θ' containing a logical atom $d \in \{\top, \perp\}$, the literal in the corresponding position Ψ_w of Σ is replaced by d . The interpretation \bullet defined as part of *procedure 1* is such that $\Pi^\bullet = \top$. Because Φ is a tautology, any extension of \bullet defined on all atoms of Σ must make Σ true. We also know that for each position Ψ_w in Σ' that contains a logical atom \top (resp. \perp), the literal in position Ψ_w of Σ evaluates to \top (resp. \perp) under \bullet . This means that Σ' is true regardless of how its non-logical atoms are interpreted and is a tautology. Our goal is to show that Θ' is a tautology as well.

Pick some arbitrary truth assignment $*$ for Θ' . We show $(\Theta')^* = \top$. For each position ψ_w in Θ' containing a positive literal y , the corresponding position Ψ_w in Σ' contains a disjunction of positive literals $(q_{i,1}^y \vee q_{i,2}^y \vee \dots \vee q_{i,s_y}^y)$ such that $\sigma(q_{i,1}^y) = P_{i,1}^y$, $\sigma(q_{i,2}^y) = P_{i,2}^y$, ..., $\sigma(q_{i,s_y}^y) = P_{i,s_y}^y$ for some $1 \leq i \leq r_y$. Similarly, for each position ψ_w in Θ' containing a negative literal $\neg y$, the corresponding position Ψ_w in Σ' contains a disjunction of negative literals $(\neg t_{1,j}^y \vee \neg t_{2,j}^y \vee \dots \vee \neg t_{r_y,j}^y)$ such that $\sigma(t_{1,j}^y) = P_{1,j}^y$, $\sigma(t_{2,j}^y) = P_{2,j}^y$, ..., $\sigma(t_{r_y,j}^y) = P_{r_y,j}^y$ for some $1 \leq j \leq s_y$. We now define a truth assignment \dagger for Σ' . If $y^* = \top$, let \dagger be such that all of the corresponding literals $\neg t_{1,j}^y, \neg t_{2,j}^y, \dots, \neg t_{r_y,j}^y$, for every $1 \leq j \leq s_y$ evaluate to false under \dagger . If $y^* = \perp$, let \dagger be such that all corresponding literals $q_{i,1}^y, q_{i,2}^y, \dots, q_{i,s_y}^y$, for every $1 \leq i \leq r_y$ evaluate to false under \dagger . With some thought, one can see that whenever Θ' has a false (under $*$) literal in a position ψ_w , Σ' has a false (under \dagger) disjunction of literals in the corresponding position Ψ_w . So, if $(\Theta')^* = \perp$, then $(\Sigma')^\dagger = \perp$. That is, $(\Sigma')^\dagger = \top$ (which is the case due to the tautologicity of Σ') implies $(\Theta')^* = \top$. \square

REFERENCES

- [1] M. Bauer. *A PSPACE-complete first order fragment of computability logic*. **ACM Transactions on Computational Logic** 15 (2014), Article 1.
- [2] G. Japaridze. *Introduction to computability logic*. **Annals of Pure and Applied Logic** 123 (2003), pp. 1-99.
- [3] G. Japaridze. *Propositional computability logic I*. **ACM Transactions on Computational Logic** 7 (2006), pp. 302-330.
- [4] G. Japaridze. *Propositional computability logic II*. **ACM Transactions on Computational Logic** 7 (2006), pp. 331-362.
- [5] G. Japaridze. *From truth to computability I*. **Theoretical Computer Science** 357 (2006), pp. 100-135.
- [6] G. Japaridze. *From truth to computability II*. **Theoretical Computer Science** 379 (2007), pp. 20-52.
- [7] G. Japaridze. *Introduction to cirquent calculus and abstract resource semantics*. **Journal of Logic and Computation** 16 (2006), pp. 489-532.
- [8] G. Japaridze. *Cirquent calculus deepened*. **Journal of Logic and Computation** 16 (2008), pp. 983-1028.
- [9] G. Japaridze. *In the beginning was game semantics*. In: **Games: Unifying Logic, Language and Philosophy**. O. Majer, A.-V. Pietarinen and T. Tulenheimo, eds. Springer 2009, pp. 249-350.
- [10] G. Japaridze. *The taming of recurrences in computability logic through cirquent calculus, Part I*. **Archive for Mathematical Logic** 52 (2013), pp. 173-212.
- [11] G. Japaridze. *The taming of recurrences in computability logic through cirquent calculus, Part II*. **Archive for Mathematical Logic** 52 (2013), pp. 213-259.
- [12] P. Lincoln, J. Mitchell, A. Scedrov and N. Shankar. *Decision problems for propositional linear logic*. **Annals of Pure and Applied Logic** 56 (1992), pp. 239-311.
- [13] I. Mezhirova and N. Vereshchagin. *On abstract resource semantics and computability logic*. **Journal of Computer and System Sciences** 76 (2010), pp. 356-372.
- [14] M. Qu, J. Luan, D. Zhu and M. Du. *On the toggling-branching recurrence of computability logic*. **Journal of Computer Science and Technology** 28 (2013), pp. 278-284.
- [15] L. Stockmeyer. *The polynomial-time hierarchy*. **Theoretical Computer Science** 3 (1977), pp. 1-22.
- [16] W. Xu and S. Liu. *Deduction theorem for symmetric cirquent calculus*. **Advances in Intelligent and Soft Computing** 82 (2010), pp. 121-126.
- [17] W. Xu and S. Liu. *Soundness and completeness of the cirquent calculus system CL6 for computability logic*. **Logic Journal of the IGPL** 20 (2012), pp. 317-330.
- [18] W. Xu and S. Liu. *The parallel versus branching recurrences in computability logic*. **Notre Dame Journal of Formal Logic** 54 (2013), pp. 61-78.
- [19] W. Xu. *A propositional system induced by Japaridze's approach to IF logic*. **Logic Journal of the IGPL** first published online June 30, 2014 doi:10.1093/jigpal/jzu020.