The Computational Complexity of Propositional Cirquent Calculus

Matthew S. Bauer
Department of Computer Science
University of Illinois at Urbana-Champaign

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. In particular, cirquent calculus has been 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 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 \( \Sigma_2^p \)-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. In particular, cirquent calculus has been instrumental in axiomatizing various fundamental fragments of 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.

Recent works [3–6, 9] have introduced sound and complete axiomizations 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 \( \exists \) and \( \exists \) and the “blind” quantifiers \( \forall \) and \( \forall \). For a full discussion of these operators and the semantics CoL, see [8]. 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 ∀ and ∃ within CL4’s language guarantees it undecidability. In [6], however, the ∀, ∃-free fragment of CL4 was shown to be decidable in polynomial space. A PSPACE-completeness proof was later given in [11].

The deductive apparatus of CL4 is far removed from traditional Gentzen or Hilbert style axiomizations. 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. The so-called “classical” rules of cirquent calculus, in the absence the particularly troublesome contraction rule, produce a sound and complete system for the fragment of CoL containing the operators ∨, ∧ and ¬. In this paper, we show that the logic axiomitized by this system, known as CL5, is Σ₂⁰-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.

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 and Computability Logic, see [7,8]. As noted previously, a formula of the language of cirquent calculus is akin to that of classical propositional logic, where formulas are built from the connectives ¬, ∧ and ∨ as well as non-logical atoms (also known as propositional letters). The language does not contain logical atoms such as ⊤ or ⊥; the latter can simply be understood as abbreviations of P ∧ ¬P and P ∨ ¬P, respectively, for some (whatever) atom P. Also, ¬ can only be applied to atoms.

A k-ary pool, for k ∈ N, is a sequence Pl = (F₁, F₂, ..., Fₖ) of formulas. There is no requirement of uniqueness for a formula in a pool. We refer to a particular occurrence of a formula as an oformula. A k-ary structure, for k ∈ N, is a finite sequence St = (Γ₁, Γ₂, ..., Γₘ) for m ≥ 0, where each Γᵢ, called a group of St, is a subset of {1, 2, ..., k}. It is permitted that Γᵢ = Γⱼ for i ≠ 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 ≥ 0) cirquent is a pair C = (StC, PlC), 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.

\[ \begin{array}{cccc}
A & B & A & D \\
\end{array} \]

Following the style of [7], we will adopt the notion that a diagram “is” a cirquent (and vise versa). For the remainder of the paper, we will use diagrams as cirquents, rather than “representations” of such entities. When an ogroup $\Gamma$ is connected with an arc to an oformula $G$, we say that $\Gamma$ 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 $\Gamma$ and $\Delta$ in a given cirquent $C$ means replacing in $C$ the two ogroups $\Gamma$ and $\Delta$ 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.

\[ \begin{array}{cccc}
A & B & A & D \\
\end{array} \quad \begin{array}{cccc}
\ast & A & D \\
\end{array} \]

Merging two adjacent oformulas $F$ and $G$ into a new oformula $H$ is the result of replacing $F$ an $G$ 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 $E$.

\[ \begin{array}{cccc}
A & B & A & D \\
\end{array} \quad \begin{array}{cccc}
E & A & D \\
\end{array} \]

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 axiomazations for logics containing some of the more powerful operators of Computability Logic. See, for example, [9][10].
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 actually represents an infinite axiom scheme, as \( F \) stands for an arbitrary formula.

\[
\text{empty cirquent axiom} \quad \text{identity axiom} \\
\begin{array}{c}
\text{A} \\
\hline
\end{array} \\
\begin{array}{c}

\begin{array}{c}
\neg F \\
\hline
F
\end{array}
\end{array}
\]

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. It’s conclusion is obtained by placing each of the two premise cirquents side by side in a single cirquent, as illustrated below.

\[
\begin{array}{c}
F \\
\hline
\\
E \\
\hline
G
\end{array} \\
\begin{array}{c}
\text{M} \\
F \\
\hline
\end{array}
\]

2.3 Exchange (E)

The exchange rule comes in two varieties, oformula exchange and ogroup exchange, each of which takes a single premise. 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 oformula 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.
2.4 Weakening (W)

The weakening rule has two forms; *ogroup weakening* and *pool weakening*, each of which takes a single premise. 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 a application of *pool weakening*, the conclusion is obtained by adding a new oformula to any position in the pool of the premise.

2.5 Duplication (D)

The duplication rule takes a single premise and 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 opposition 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.
2.6 Contraction (C)

The *contraction* rule takes a single 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 $\lor$-introduction ($\lor$)

This rule takes a single premise. Its conclusion is obtained from the premise by merging two adjacent oformulas $F$ and $G$ into a single oformula $F \lor G$ such that all arcs pointing to either $F$ or $G$ now point to $F \lor G$. Two examples of an application of this rule are given below.
2.8 $\wedge$-introduction ($\wedge$)

This rule takes a single 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 be combined into a single oformula $F \wedge G$. 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 $C$ 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 $\sigma$ that maps every atom $P$ in $C$ to some propositional formula $\sigma(P)$. If $\sigma(P)$ is an atom for every $P$ in $C$, then we say $\sigma$ is atomic-level substitution. This notion can be extended to all formulas by requiring that $\sigma(\neg P) = \neg \sigma(P)$, $\sigma(F \lor G) = \sigma(F) \lor \sigma(G)$ and $\sigma(F \land G) = \sigma(F) \land \sigma(G)$. Let $A$ and $B$ be formulas. $B$ is said to be an instance of $A$ iff there exists a substitution $\sigma$ such that $\sigma(B) = A$. If $\sigma$ 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 \cite{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 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 CL5 is NP-complete. This result is achieved by a combination of two theorems, the first of which states that every formula \( \mathcal{F} \) provable in CL5 has a proof that is polynomial in the size of \( \mathcal{F} \). 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. \( \Box \)

**Lemma 3.2** Let \( \mathcal{F} \) be a formula provable in CL5 with \( n \) occurrences of positive literals. No CL5 proof of \( \mathcal{F} \) can contain a cirquent with more than \( n \) ogroups.

**Proof.** Let \( \mathcal{F} \) and \( n \) be as in the statement of the lemma and let \( \Delta \) be a proof tree for \( \mathcal{F} \). Assume for a contradiction that \( \Delta \) has a cirquent \( \mathcal{D} \) that contains more than \( n \) ogroups. Now let \( \Delta' \) be the subtree of \( \Delta \) rooted at \( \mathcal{D} \). A careful examination of the rules of CL5 will show that, no matter which series of rules are applied in \( \Delta' \), the total number of ogroups in all the leaves of \( \Delta' \) must be greater than or equal to the number of ogroups in \( \mathcal{D} \). 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 \( \Delta' \) must be derived by either the identity axiom or the empty cirquent axiom and thus cannot contain more than 1 ogroup. This means \( \Delta' \) 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 ofomulas \( K \) and \( \neg K \). But the total number of occurrences of positive literals in ofomulas (and hence positive ofomulas) in the leaves of \( \Delta \) cannot exceed \( n \). This is because, by our assumption, the conclusion of \( \Delta \) contains a single oformula with \( n \) positive literals. Further, the ofomulas of the premise cirquent(s) in the application of a CL5 rule cannot contain more literals than
the oformulas of the conclusion. This means at most $n$ applications of the identity axiom are possible, contradiction.

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 formulas in its pool plus the sum of the sizes of each group 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 CL5$^-$ formula and let $\Delta$ be a CL5$^-$ proof tree for $F$. Given $\Delta$, we show that a new proof $\Delta'$ can be obtained from $\Delta$ such that $\Delta'$ 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 made. 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 $\Delta_1$ be the proof that results from removing all applications of the empty cirquent axiom in the preceding manner.

Observe now that $\Delta_1$ must contain exactly one application of conjunction introduction or disjunction introduction for each occurrence of $\land$ or $\lor$ in $F$, respectively. This is because every application of either rule introduces, in its conclusion, a single $\land$ or $\lor$ connective that cannot be later removed by the application of any CL5$^-$ rule. The number of occurrences of positive literals 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 $\Delta_1$, we can also bound the number of applications of the mix rule. To see this, first note that because $\Delta_1$ 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 $\Delta_1$. 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 in the conclusion of $\Delta_1$. This is because no CL5$^-$ rule can remove, in its conclusion, an oformula contained in its premise cirquent. Thus

\footnote{The only purpose of this axiom in [?] is to ensure the provability of the empty cirquent itself; everything else is provable without using the empty cirquent axiom.}
we have have an immediate bound of $O(k)$ for the number of applications of pool weakening in $\Delta_1$. To bound the number of possible applications of ogroup weakening in $\Delta_1$, notice that no cirquent in $\Delta_1$ can contain more than $k^2$ arcs. This is because, by lemma \[k\] bounds the maximum number of ogroups and formulas 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 $\Delta_1$ 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 $\Delta_1$. Each application of ogroup weakening creates a single arc in its conclusion that was not present in its premise. As the conclusion of $\Delta_1$ contains a single arc, no more than $O(k^3)$ arcs can be created from premise to conclusion in $\Delta_1$ by applications of ogroup weakening.

Every leaf of $\Delta_1$ must be derived from the identity axiom, which is applied at most $O(k)$ times in $\Delta_1$. That is, there are at most $O(k)$ paths from root to leaf in $\Delta_1$. Each such path $\Delta_1^i$ 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 $\Delta_1^i$ can contain at most $O(k^3)$ sequences $\Phi_w$ for $0 \leq w \leq O(k^3)$ where each $\Phi_w := C_1^w, C_2^w, ..., C_n^w$ (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 $\Delta_1$, each $\Phi_w$ can be of arbitrary length. We show, however, a new proof $\Delta'$ can be obtained from $\Delta_1$ such that each $\Phi_w$ uses no more than $O(k^2)$ applications of oformula and ogroup exchange. This is because, the last cirquent $C_n^w$ in any $\Phi_w$ can always be obtained from $C_1^w$ using no more than $O(k^3)$ applications of oformula and ogroup exchange. By lemma \[\text{3.2}\] the first cirquent $C_1^w$ can contain at most $k$ ogroups. Moving all of these ogroups to an arbitrary position in $C_1^w$ 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 $C_1^w$ also requires no more than $O(k^2)$ applications of oformula exchange. This means that every sequence $\Phi_w$ in $\Delta_1$ can be replaced with a new sequence $\Phi'_w$ such that $\Phi'_w$ uses no more than $O(k^2)$ applications of oformula or ogroup exchange. Let $\Delta'$ be the CL5− proof tree that results from replacing each $\Phi_w$ in $\Delta_1$ by $\Phi'_w$. $\Delta'$ 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 $\Delta'$.

It should be clear that if $\Delta$ proves $F$, then $\Delta'$ will also prove $F$. The number of applications of non-exchange rules does not change from $\Delta_1$ to $\Delta'$ and remains $O(k^3)$. Further, $\Delta'$ uses no more than $O(k^6)$ applications of exchange. Totaling the number of applications of every rule in $\Delta'$, 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 $k$. 

10
Proof. By lemma \textbf{3.3}, there exists a CL5\(^-\) proof \(\Delta\) of \(F\) that uses a polynomial number of rule applications. Further, by lemma \textbf{3.2}, the size of any cirquent in \(\Delta\) must be polynomial in \(k\). Thus, the maximum size of each cirquent multiplied by the maximum number of cirquents in \(\Delta\) yields a polynomial bound on the size of \(\Delta\).

The result of Theorem \textbf{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 unsorted pair of vertices. The degree of a vertex \(v\) in \(G\), denoted \(\text{deg}(v)\), is defined as the number of edges incident to \(v\). Furthermore, we say two vertices are adjacent if they are connected by an edge. 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 \(V\). In complexity theory, finding a vertex cover can be stated as a decision problem as follows. Given a graph \(G\) and parameter \(k\), 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 \textbf{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 \(k\) vertex cover problem. Fix some arbitrary graph \(G = (V, E)\) and some \(k \in \mathbb{N}\). The reduction, borrowed from [11], follows.

\[
f(V, E, k) := (\Psi(k)) \lor (\Theta(V, E)) \lor (\Omega(E))
\]

\[
\Psi(k) := q \lor q \lor \ldots \lor q
\]

\[\text{total of } k \text{ literals}
\]

\[
\Theta(V, E) := (\neg q \land (\neg v_1 \lor \neg v_2 \lor \ldots \lor \neg v_1)) \lor \ldots \lor (\neg q \land (\neg v_n \lor \neg v_n \lor \ldots \lor \neg v_n))
\]

\[\text{deg}(v_1) \quad \text{deg}(v_n)
\]

\[\text{for each vertex } v_1, v_2, \ldots, v_n \in V
\]

\[
\Omega(E) := (e^1_1 \lor e^2_1 \lor e^3_1 \lor \ldots \land (e^1_m \lor e^2_m \lor e^3_m)
\]

\[\text{for each edge } e_1, e_2, \ldots, e_m \in E,
\]

Above, \(q\) is a new atom that differs from all other atoms in the formula. In \(\Omega(E)\), \(e^1_i\) and \(e^2_i\) 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\(^-\).
“⇒” Our reduction is identical to that of [11] (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 logic so the result follows immediately.

“⇐” Assume \( f(V, E, k) \) is provable in CL5−. We need to show that there exists a \( k \) vertex cover of \( G \). By Theorem 2, there exists a normal binary tautology \( \beta \) and an atomic-level substitution \( \sigma \) such that \( \sigma(\beta) = f(V, E, k) \). Let \( \Sigma \) be the set of atoms in \( \beta \) such that for any \( l \in \Sigma \), \( \beta \) contains both \( l \) and \( \neg l \) and \( \sigma(l) = q \). For each \( l \in \Sigma \), let \( \Theta_l \) be the conjunct in the \( \Theta(V, E) \) component of \( f(V, E, k) \) that contains \( \neg \sigma(l) \). By our construction, each \( \Theta_l \) takes the form

\[
(\neg q \land (\neg v_j \lor \neg v_j \lor \ldots \lor \neg v_j))
\]

for \( 1 \leq j \leq n \). Define a vertex cover \( V' \subset V \) for \( G \) as the set all vertices in \( G \) labeled \( v_j \) where \( v_j \neq q \) and \( v_j \) is contained in \( \Theta_l \) for some \( l \in \Sigma \).

To see that \( V' \) is indeed a \( k \) vertex cover for \( G \), note first that \( \sigma \) 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 \( q \) in \( f(V, E, k) \). This means \( \Sigma \) contains no more than \( k \) atoms and subsequently the vertex cover derived from \( \Sigma \) contains no more than \( k \) vertices.

We now show that every vertex in \( V \) is adjacent to some vertex in \( V' \). Note that because \( f(V, E, k) \) is an atomic-level instance of \( \beta \), 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 \( \beta \). For each \( l \in \Sigma \) let \( l^* = \bot \). For any atom \( l \in \Sigma \) with the property that \( \sigma(l) = q \), let \( l^* = \top \). Further, for each atom \( l \) of \( \beta \) where \( \sigma(l) = a \) for some \( a \in V' \), let \( l^* = \top \). All remaining atoms \( l \in \beta \) should be interpreted as \( l^* = \bot \).

Let \( \Psi' \) be the subformula of \( \beta \) such that \( \sigma(\Psi') = \Psi(k) \). Note that every atom \( l \) of \( \Psi' \) is also in \( \Sigma \) and hence is interpreted as \( l^* = \bot \). Because \( \Psi' \) is a disjunction \( l_1 \lor l_2 \lor \ldots \lor l_n \) of \( n \) positive literals, we have \( (\Psi')^* = \bot \). Next let \( \Theta' \) be the subformula of \( \beta \) such that \( \sigma(\Theta') = \Theta(V, E) \). Each disjunct of \( \Theta' \) takes the form \( \neg l_1 \land (\neg l_2 \lor \neg l_3 \lor \ldots \lor \neg l_n) \) for some positive integer \( n \). If \( l_1 \notin \Sigma \) then \( l_1^* = \top \) and \( \neg l_1^* = \bot \), making the entire subformula evaluate to \( \bot \) under *.

If \( l_1 \in \Sigma \) then \( l_1^* = \bot \) and \( \neg l_1^* = \top \). Note, however, that when \( l_1 \in \Sigma \) then \( \sigma(l_m) = a \) where \( a \in V' \) and \( m \in \{2, 3, \ldots \} \). This means \( l_m^* = \top \) and \( \neg l_m^* = \bot \), again making the subformula evaluate to \( \bot \) under *.

Because each disjunct of \( \Theta' \) evaluates to \( \bot \) under *, we have \( (\Theta')^* = \bot \).

Finally, let \( \Omega' \) be the subformula of \( \beta \) such that \( \sigma(\Omega') = \Omega(E) \). Because \( \beta = (\Psi') \lor (\Theta') \lor (\Omega') \) is a tautology where \( (\Psi')^* = \bot \) and \( (\Theta')^* = \bot \), we have \( (\Omega')^* = \top \). Each conjunct of \( \Omega' \) takes the form \( l_1 \lor l_2 \) such that \( \sigma(l_1) \) and \( \sigma(l_2) \) are the endpoints of an edge in \( E \). Further, by our construction, every edge in

\(^2\)From Theorem 3, 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−.
E is represented by such a conjunct in Ω'. By our definition of *, we have l* = T for a positive literal l of Ω' only when σ(l) ∈ 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. □

4 CL5 is $\Sigma_2^P$-complete

This section contains our main result, namely, the $\Sigma_2^P$-completeness of the decision problem for provability of formulas in CL5.

Lemma 4.1 Deciding provability for CL5 is in $\Sigma_2^P$.

Proof. The following is a $\Sigma_2^P$ algorithm that 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. □

Let TQBF-$\Sigma_2$ be the problem of deciding truth for a totally 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 [14], this problem is $\Sigma_2^P$-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. An atom z is non-isolated in formula φ if there are at least two (positive or negative) occurrences of z in φ.

Theorem 4.2 Deciding provability for the logic CL5 is $\Sigma_2^P$-complete.

Proof. By lemma 4.1, the problem of deciding provability for CL5 is in $\Sigma_2^P$. To show it is $\Sigma_2^P$-hard, we construct a polynomial time mapping reduction f from TQBF-$\Sigma_2$ to CL5-provability. We can safely restrict our attention to instances of TQBF-$\Sigma_2$ where the Boolean portion of the formula is in disjunctive normal form and only contains bound variables, 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. The following steps are used to construct the CL5 formula f(φ).

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

$$g(z) = Z^z \land (Z^z \rightarrow u^z_1 \land u^z_2 \land \ldots \land u^z_{k^z}) \land (Z^z \rightarrow \neg v^z_1 \land \neg v^z_2 \land \ldots \land \neg v^z_{t^z})$$

where $Z^z$, $u^z_1$, $u^z_2$, ..., $u^z_{k^z}$ and $v^z_1$, $v^z_2$, ..., $v^z_{t^z}$ are all fresh variables, uniquely chosen for z. Here, if $k^z$ (resp. $t^z$) is 0, the second (resp. third) conjunct

3Here, fresh variables are those not occurring elsewhere in φ.
should be omitted. Now let \( z_1, z_2, ..., z_l = X \) and let \( \phi_0 \) be the formula 
\[ (g(z_1) \land g(z_2) \land ... \land g(z_l)) \rightarrow \Theta_1. \]
Here \( \Theta_1 \) is a formula derived from \( \Theta \) such that, for each \( z \in X \), every positive occurrence of \( z \) in \( \Theta \) is replaced by a unique literal from \( u_1^z, u_2^z, ..., u_k^z \) and every occurrence of \( \neg z \) in \( \Theta \) is replaced by a unique literal from \( -v_1^z, -v_2^z, ..., -v_l^z \).

2. Consider any \( y \in Y \). Let \( r_y \) be the number of positive occurrences of \( y \) in \( \phi_0 \), 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 \( P_{i,j} \). Now, define \( f(\phi) \) to be the result of replacing in \( \phi_0 \), for each \( y \in Y \), every positive occurrence of the literal \( y \) by \( (P_{i,1} \lor P_{i,2} \lor ... \lor P_{i,s_y}) \) and every (positive) occurrence of the literal \( \neg y \) by \( (\neg P_{i,1} \lor \neg P_{i,2} \lor ... \lor \neg P_{i,s_y}) \) 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 \( \phi \) is true. Then there exists some truth assignment \( \circ : X \rightarrow \{ \top, \bot \} \) such that \( \Theta^* \) is true under any truth assignment \( \ast \) that extends \( \circ \) to \( X \cup Y \). By our assumption, \( \Theta \) is in disjunctive normal form and hence takes the form
\[ (\psi_1 \land \psi_2 \land ... \land \psi_m) \lor ... \lor (\psi_{m+1} \land \psi_{m+2} \land ... \land \psi_{m+t}) \]
where each \( \psi_w \) is either a positive or negative literal. Now let \( \Omega \) and \( \Sigma \) represent the antecedent and consequent of the outermost implication in \( f(\phi) \) such that \( f(\phi) = \Omega \rightarrow \Sigma \). Our construction guarantees that \( \Sigma \) takes the form
\[ (\Psi_1 \land \Psi_2 \land ... \land \Psi_n) \lor ... \lor (\Psi_{m+1} \land \Psi_{m+2} \land ... \land \Psi_{m+t}) \]
where each \( \Psi_w \) is a disjunction of literals (such a “disjunction” may have only a single “disjunct”). That is, \( \Sigma \) is obtained from \( \Theta \) by replacing each literal \( \psi_w \) by \( \Psi_w \), where \( \Psi_w \) is a disjunction of literals. We will henceforth use \( \psi_w \) to represent a unique position in \( \Theta \) and \( \Psi_w \) to represent the corresponding position in \( \Sigma \).

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 \( \Phi \) of which \( f(\phi) \) is an instance. Namely, we let \( \Phi \) be the formula obtained from \( f(\phi) \) as follows. For each \( z \in X \), if \( z^0 = \top \) (resp. \( \bot \) ) replace the third (resp. second) occurrence of \( Z^z \) by an atom \( Q^z \) such that \( Q^z \) does not occur elsewhere in \( \Phi \) and is unique for each \( z \in X \). It should be easy to see that \( \Phi \) is a quantifier free binary formula and \( f(\phi) \) is an instance of \( \Phi \). We need only show that \( \Phi \) is a tautology, i.e., \( \Phi \) is true under any truth assignment \( \ast \) defined on the variables of \( \Phi \). Again notice that \( \Phi \) takes form \( \Phi = \Pi \rightarrow \Sigma \) where \( \Sigma \) is the same as in \( f(\phi) \).

Given \( X = z_1, z_2, ..., z_l \), we have \( \Pi = g(z_1)' \land g(z_2)' \land ... \land g(z_l)' \) where each \( g(z)' \) matches one of the following forms:

1. \( Z^z \land (Z^z \rightarrow u_1^z \land u_2^z \land ... \land u_k^z) \land (Q^z \rightarrow -v_1^z \land -v_2^z \land ... \land -v_l^z) \)

\[ \text{It is possible that the third (resp. second) conjunct in formula 1 (resp. 2) is absent.} \]
2. \( Z^z \land (Q^z \rightarrow u_1^z \land u_2^z \land \ldots \land u_{k_z}^z) \land (Z^z \rightarrow \neg v_1^z \land \neg v_2^z \land \ldots \land \neg v_{k_z}^z) \)

For any \( g(z)' \) in \( \Pi \) where \( z \in X \), if \( (Z^z)'^* = \bot \) then \( \Pi^* = \bot \), 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 \( \Pi^* = \bot \) and again \( \Phi^* = \top \). Thus, we need only guarantee \( \Sigma^* = \top \) under truth assignments such that, for each conjunct \( g(z)' \) in \( \Pi \), \( Z^z \) and every literal in the consequent of the implication containing antecedent literal \( Z^z \) are true under \( ^* \).

Let \( \Theta' \) be the formula that results from replacing in \( \Theta \) every occurrence of \( z \in X \), by its truth value under \( ^* \). For example, if \( z^\circ = \bot \), replace \( \neg z \) by \( \top \) and \( z \) by \( \bot \). It should be easy to see that \( \Theta' \) is a tautology. For any position \( \psi_w \) that contains \( z \) (resp. \( \neg z \)) in \( \Theta \) and \( \top \) in \( \Theta' \), the position \( \Psi_w \) in \( \Sigma \) must contain a single literal \( u_i^z \) (resp. \( \neg w_i^z \)) such that \( (w_i^z)^* = \top \) (resp. \( (\neg w_i^z)^* = \top \)) 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. every \( \neg w_i^z \) for \( 1 \leq i \leq t_z \)) must occur in the consequent of the implication in \( \Pi \) with non-isolated antecedent \( Z^z \). For reasons already discussed, such a literal must be true under any truth assignment \( ^* \) where \( \Pi^* = \top \). Define \( \Sigma' \) as the formula such that, for each occurrence of \( \top \) or \( \bot \) in position \( \psi_w \) of \( \Theta' \), the literal in position \( \Psi_w \) of \( \Sigma \) is replaced by the same value (\( \top \) or \( \bot \)) as \( \psi_w \). Note that \( \Sigma' \) only substitutes the logical atom in position \( \Psi_w \) of \( \Sigma \) when \( \Psi_w \) contains a positive or negative occurrence of an atom in \( X \). Since our goal is to show that \( \Sigma \) is true under truth assignments that make \( \Pi \) true, we need only show that \( \Sigma' \) is a tautology.

For a contradiction, assume \( \Sigma' \) is not a tautology. Then there exists some truth assignment \( ^\dagger \) defined on the variables of \( \Sigma' \) such that \( (\Sigma')^\dagger = \bot \). We define a truth assignment \( ^\dagger \) for \( \Theta' \) as follows. If a variable \( y \in Y \) is such that, for some \( i \) with \( 1 \leq i \leq r_y \), 

\[ (P_{i,1}^y \lor \ldots \lor P_{i,s}^y)^\dagger = \bot, \]

we let \( y_i^\dagger = \bot \); otherwise we let \( y_i^\dagger = \top \). It is not hard to see that, if a subformula of \( \Sigma' \) in a position \( \Psi_w \) is false under \( ^\dagger \), then the subformula of \( \Theta' \) in the corresponding position \( \psi_w \) is false under \( ^\dagger \) (but not necessarily vice versa). This, in view of the monotonicity of \( \lor \) and \( \land \), obviously implies that \( (\Theta')^\dagger = \bot \), because \( \Sigma' \) and \( \Theta' \) have the same (\( \lor, \land \))-structures. Now we are dealing with a contradiction, because the tautological \( \Theta' \) cannot be false.

“\( \leftrightarrow \)” Assume \( f(\phi) \) is provable in CL5. By Theorem 2.2 for some normal binary tautology \( \Phi \) there exists an atomic level substitution \( \sigma \) such that \( \sigma(\Phi) = f(\phi) \). Let \( \Phi = \Pi \rightarrow \Sigma \), where \( \Pi \) and \( \Sigma \) represent the antecedent and consequent of the outermost implication in \( \Phi \). We want to define a truth assignment \( ^\circ : X \rightarrow \{ \top, \bot \} \) such that \( \Theta^\circ \) 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 \( \Theta \) while concurrently defining a partial truth assignment \( ^* \) for \( \Phi \).

Procedure 1 - Consider a \( z \in X \) and let \( A \) be the atom of \( \Phi \) 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 \( \Phi \) 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 \( \Phi \)). Define \( z^\circ = \top \), \( A^\circ = \top \) and
$B^* = \bot$. The consequent\(^5^\) of the second occurrence of $A$ in $\Phi$ should take the form $a_1 \land a_2 \land \ldots \land a_{k^*}$ for some positive literals $a_1, a_2, \ldots, a_{k^*}$ in $\Phi$ and the consequent of $B$ should take the form $\neg b_1 \land \neg b_2 \land \ldots \land \neg b_{t^*}$ for some negative literals $\neg b_1, \neg b_2, \ldots, \neg b_{t^*}$. Let all of $a_1, a_2, \ldots, a_{k^*}$ and $b_1, b_2, \ldots, b_{t^*}$ be true under $\ast$.

**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 $\Phi$). Define $z^o = \bot$, $A^* = \top$ and $B^* = \bot$. The consequent of the second occurrence of $A$ in $\Phi$ must take the form $\neg a_1 \land \neg a_2 \land \ldots \land \neg a_{t^*}$ for some negative literals $\neg a_1, \neg a_2, \ldots, \neg a_{t^*}$ and the consequent of $B$ should take the form $b_1 \land b_2 \land \ldots \land b_{t^*}$ for some positive literals $b_1, b_2, \ldots, b_{t^*}$. Let all of $a_1, a_2, \ldots, a_{k^*}$ and $b_1, b_2, \ldots, b_{t^*}$ be false under $\ast$.

**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 $\Phi$ takes the form $a_1 \land a_2 \land \ldots \land a_{k^*}$ for some positive literals $a_1, a_2, \ldots, a_{k^*}$ then define $z^o = \top$ and $A^* = \bot$. Further, let all of $a_1, a_2, \ldots, a_{k^*}$ be true under $\ast$. If the consequent of the second occurrence of $A$ in $\Phi$ takes the form $\neg a_1 \land \neg a_2 \land \ldots \land \neg a_{t^*}$ for some negative literals $\neg a_1, \neg a_2, \ldots, \neg a_{t^*}$ then define $z^o = \bot$ and $A^* = \top$. Further, let all of $a_1, a_2, \ldots, a_{k^*}$ be false under $\ast$.

**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 $z^o = \bot$, $A^* = \top$ and $B^* = C^* = \bot$. Further, let every literal in the consequent of $B$ evaluate to $\bot$ under $\ast$ and every literal in the consequent of $C$ evaluate to $\top$ under $\ast$.

We again have $\Theta$ in disjunctive normal form and taking the form

$$(\psi_1 \land \psi_2 \land \ldots \land \psi_n) \lor \ldots \lor (\psi_{m+1} \land \psi_{m+2} \land \ldots \land \psi_{m+i})$$

where each $\psi_w$ is either a positive or negative literal. Its is also the case that $\Sigma$ takes the form

$$(\Psi_1 \land \Psi_2 \land \ldots \land \Psi_n) \lor \ldots \lor (\Psi_{m+1} \land \Psi_{m+2} \land \ldots \land \Psi_{m+i})$$

where each $\Psi_w$ is a disjunction of literals (possibly with just a single “disjunct”). Thus, we will use our previous convention in which $\psi_w$ represents a unique position in $\Theta$ and $\Psi_w$ represents the corresponding position in $\Sigma$.

Our construction guarantees that for each position $\psi_w$ in $\Theta$ containing some literal $z$ (resp. $\neg z$) such that $z \in X$, $\Psi_w$ contains positive (resp. negative) literal $a$ (resp. $\neg a$). By *procedure 1* we have $z^o = a^\ast$. As in the previous direction, let $\Theta'$ be the formula that results from replacing, in $\Theta$, every occurrence of $z \in X$ by its truth value under $\ast$. Let $\Sigma'$ be the formula such that, for each position $\psi_w$ in $\Theta'$ containing a logical atom $d \in \{ \top, \bot \}$, the literal in the corresponding position $\Psi_w$ of $\Sigma$ is replaced by $d$. The interpretation $\ast$ defined as part of *procedure 1* is such that $\Pi^\ast = \top$. Because $\Phi$ is a tautology, any extension of $\ast$ defined on all atoms of $\Sigma$ must make $\Sigma$ true. We also know that for each position

---

\(^5\)Here 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 $W$.”
Ψ_w in Σ' that contains a logical atom ⊤ (resp. ⊥), the literal in position Ψ_w of Σ evaluates to ⊤ (resp. ⊥) under •. 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 (Θ')* = ⊤. 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^y_{1,1}, q^y_{1,2} ∨ ... ∨ q^y_{1,s_y}) such that σ(q^y_{1,1}) = P^y_{1,1}, σ(q^y_{1,2}) = P^y_{1,2}, ..., σ(q^y_{1,s_y}) = P^y_{1,s_y} for some 1 ≤ i ≤ s_y. Similarly, for each position ψ_w in Θ' containing a negative literal ¬y, the corresponding position Ψ_w in Σ' contains a disjunction of negative literals (¬t^y_{1,j} ∨ ¬t^y_{2,j} ∨ ... ∨ ¬t^y_{s_y,j}) such that σ(t^y_{1,j}) = P^y_{1,j}, σ(t^y_{2,j}) = P^y_{2,j}, ..., σ(t^y_{s_y,j}) = P^y_{s_y,j} for some 1 ≤ j ≤ r_y. We now define a truth assignment † for Σ'. If y* = ⊤, let † be such that all of the corresponding literals ¬t^y_{1,j}, ¬t^y_{2,j}, ..., ¬t^y_{s_y,j}, for every 1 ≤ j ≤ r_y evaluate to false under †. If y* = ⊥, let † be such that all corresponding literals q^y_{1,1}, q^y_{1,2}, ..., q^y_{1,s_y} for every 1 ≤ i ≤ s_y evaluate to false under †. With some thought, one can see that whenever Θ' has a false (under *) literal in a position ψ_w, Σ' has a false (under †) disjunction of literals in the corresponding position Ψ_w. So, if (Θ')* = ⊥, then (Σ')† = ⊥. That is, (Σ')† = ⊤ (which is the case due to the tautologicity of Σ') implies (Θ')* = ⊤.

□

References

[1] M. Bauer. A PSPACE-complete first order fragment of computability logic. ACM Transactions on Computational Logic 15 (2014), N 1 (to appear).

[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. *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.

[9] G. Japaridze. *The taming of recurrences in computability logic through cirquent calculus, Part I*. *Archive for Mathematical Logic* 52 (2013), pp. 173-212.

[10] G. Japaridze. *The taming of recurrences in computability logic through cirquent calculus, Part II*. *Archive for Mathematical Logic* 52 (2013), pp. 213-259.

[11] P. Lincoln, J. Mitchell, A. Sevedov and N. Shankar. *Decision problems for propositional linear logic*. *Annals of Pure and Applied Logic* 56 (1992), pp. 239-311.

[12] I. Mezhir'ov and N. Vereshchagin. *On abstract resource semantics and computability logic*. *Journal of Computer and System Sciences* 76 (2010), pp. 356-372.

[13] 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.

[14] L. Stockmeyer. *The polynomial-time hierarchy*. *Theoretical Computer Science* 3 (1977), pp. 1-22.

[15] W. Xu and S. Liu. *Deduction theorem for symmetric cirquent calculus*. *Advances in Intelligent and Soft Computing* 82 (2010), pp. 121-126.

[16] 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.

[17] W. Xu and S. Liu. *The parallel versus branching recurrences in computability logic*. *Notre Dame Journal of Formal Logic* 54 (2013), pp. 61-78.