Checking the Size of Circumscribed Formulae

Paolo Liberatore

DIAG
Sapienza University of Rome
Rome, Italy

Abstract—The circumscription of a propositional formula $T$ may not be representable in polynomial space, unless the polynomial hierarchy collapses. This depends on the specific formula $T$, as some can be circumscribed in little space and others cannot. The problem considered in this article is whether this happens for a given formula or not. In particular, the complexity of deciding whether CIRC$(T)$ is equivalent to a formula of size bounded by $k$ is studied. This theoretical question is relevant as circumscription has applications in temporal logics, diagnosis, default logic and belief revision.

Keywords—Circumscription; computational complexity; belief revision.

I. INTRODUCTION

The circumscription reasoning mechanism requires a set of variables to be minimized [1], [2], that is, set to the logical value false whenever possible. Similarly to the closed world assumption [3], it formalizes the assumption that lack of information on certain conditions can be considered evidence that they do not hold. Applications include temporal domains [4], [5], diagnosis [6], induction [7] and belief revision [8]. Contrary to the basic closed world assumption, circumscription takes into account all possible ways variables can be set to false; for example, $x \lor y$ is consistent with either $\neg x$ and $\neg y$ but not both, leading to the two possible cases $(x \lor y) \land \neg x$ and $(x \lor y) \land \neg y$. These may be up to $2^n$, if the number of variables is $n$: a trivial representation of the circumscribed formula may be exponential. However, it may be equivalent to a smaller formula.

Expressing propositional circumscription as a formula of size bounded by a polynomial has been proved not possible in general [9], unless the polynomial hierarchy collapses [10], a condition generally deemed unlikely. As a result, the problem of whether propositional circumscription can be represented in space bounded by some number $k$ has not an obvious answer: it is possible in some cases but not in others. The problem considered in this article is whether this is possible; in particular, the complexity of this problem is studied. This is similar to the problem of minimizing propositional formulae: given a formula $F$, is there an equivalent formula of size bounded by $k$ [11]? For circumscription, the question is whether the circumscription of a formula is equivalent to some formula of size bounded by $k$. For example, the circumscription of $x \lor y$ accounts for both $(x \lor y) \land \neg x$ and $(x \lor y) \land \neg y$ to be possible; therefore, the result is the formula $((x \lor y) \land \neg x) \lor ((x \lor y) \land \neg y)$. However, this formula is equivalent to $(x \land \neg y) \lor (\neg x \land y)$. By the standard metric of formulae where size is defined as the number of variable occurrences, this formula has size 4. Therefore, the circumscription of $x \lor y$ is equivalent to a formula of size bounded by $k = 4$, but not for example $k = 1$ as no formula of a single variable is equivalent to $(x \land \neg y) \lor (\neg x \land y)$. The answer is not this easy when the formula is more complex than $x \lor y$. Indeed, it will be proved that the problem is hard for the complexity class $\Pi^p_2$, that is, harder than problems such as propositional satisfiability, vertex cover and Hamiltonian cycle [10].

The question of the size of the representation has an implementation impact. Indeed, verifying which conditions hold under the circumscription assumption amounts to CIRC$(T) \models C$, where $T$ represents the current information and $C$ the condition to check, and this is a hard problem [12], [13], [14]. However, if CIRC$(T)$ can be represented by a formula $F$ of bounded size, the problem can be solved by first finding $F$ and then solving the easier (coNP) problem $\bar{F} \models C$. Once $F$ is determined, any number of other conditions $C_1, C_2, \ldots$ can then be checked against $F$ at the same cost.

Since circumscription is also used as the target of translation of several belief revision operators, the question concerns the dynamic of logic. Indeed, changing a formula to accommodate for new information is generally expected to produce a result of bounded size.

The article is organized as follows: the next section contains the formal definition of circumscription and the notations used in this article, plus two preliminary lemmas; in the section afterwards, the complexity of the problem of whether the circumscription of a formula can be represented in size bounded by some number is studied; the final section comments the practical implications of this analysis and its open problems.

II. PRELIMINARY RESULTS

Propositional formulae are denoted by the capital letters $T$ and $F$, and are always assumed to be in Negation Normal Form (NNF). Sets of variables are denoted by $X$, $Y$ and $Z$. Notation $X \bar{\cdot}$ indicates the set $\{x \mid x \in X\}$. The shorthand $x \notin y$ indicates $(x \land \neg y) \lor (\neg x \land y)$.

Models are denoted by $\omega_X$, where the suffix $X$ indicates the set of variables: $\omega_X$ is a truth evaluation of the variables $X$, $\omega_Y$ is a truth evaluation of the variables $Y$, etc. Models are identified by the sets of variables they assign to true; this allows to write $\omega_X \subseteq \omega_X$ to mean that $\omega_X$ assigns true to all variables $\omega_X$ assigns true, but not necessarily the converse. The model assigning true to all variables $X$ is denoted $\omega_X^*$, the one assigning false to all $\omega_X$.

The following notation is used to denote a formula that represents a single model: $Form(\omega_X) = \land \{x \mid \omega_X \models x\} \cup \land \{x \mid \omega_X \models \neg x\}$. 

www.ijarai.thesai.org
\[ \{\neg x \mid \omega_X \models \neg x\} \text{. If } F \text{ is a formula over variables } X \cup Y \text{ and } \omega_X \text{ a truth evaluation over } X, \text{ the notation } F|_{\omega_X} \text{ indicates the formula obtained by replacing each variable } X \text{ in } F \text{ with its truth value according to } \omega_X. \]

In this article, circumscription is defined over propositional logic, and restricted to the case where all variables are minimized. This gives rise to the following definition.

**Definition 1:** Given a formula \( T \) over variables \( X \), its circumscription \( \text{CIRC}(T) \) is defined as follows, where \( X^\neg = \{\neg x \mid x \in X\} \).

\[
\text{CIRC}(T) = \bigvee \left\{ T \land S \mid \begin{array}{l}
S \subseteq X^\neg \\
T \land S \not\models \bot \\
\forall S' \subseteq X^\neg \ (S \subseteq S' \Rightarrow T \land S' \models \bot)
\end{array} \right\}
\]

Some formulae \( T \) have small circumscription. For example, \( T = \bigwedge X \) has a circumscription equal to itself, since \( S = \emptyset \) is the only subset of \( X^\neg \) satisfying the definition. Some other formulae have larger circumscription, such as \( T = \bigvee X \); indeed, for this formula \( S = X^\neg \setminus \{x\} \) satisfies the definition for every \( x \in X \). Some formulae do not even have polynomial-size equivalent representations of their circumscription [9].

Circumscription is simple to compute on formulae that imply either \( x, \neg x \), or \( x \neq x' \) for some variables \( x \) and \( x' \):

**Property 1:** The following equivalences hold:

\[
\begin{align*}
\text{CIRC}(T \land x) &= x \land \text{CIRC}(T|_{\omega_X}) \\
\text{CIRC}(T \land \neg x) &= \neg x \land \text{CIRC}(T|_{\omega_X}) \\
\text{CIRC}(T \land (x \neq x')) &= (x \neq x') \land \left( (\text{CIRC}(T)|_{\omega_X}) \lor \text{CIRC}(T|_{\omega_X}) \right)
\end{align*}
\]

These are well-known properties. The third equivalence allows evaluating \( \text{CIRC}(T) \) separately for \( x \) true and \( x \) false, if \( T \) does not contain \( x' \).

The size of formulae is defined by the following metrics.

**Definition 2:** The size of a formula \( F \), denoted \(|F|\), is the number of variable occurrences in \( F \).

For example, the size of \((a \land \neg b) \lor c \lor \neg (a \land b)\) is four, since the variable \( a \) occurs twice in it and \( b \) and \( c \) once each. According to this definition, the size of a formula and of its NNF form obtained by applying the De Morgan rules coincide. A bound on the size of a formula derives from its models.

**Lemma 1:** If a NNF formula \( F \) has a model that satisfies a literal \( l \) but not the modified model where the value of \( l \) is inverted, then \( F \) contains \( l \).

**Proof:** Let \( F \) be a formula and \( \omega_X \) its model satisfying \( l \). Let us assume, on the contrary, that \( F \) does not mention the literal \( l \). Since \( F \) is in NNF, no part of it is turned to false by changing the value of \( l \) from true to false. As a result, the model \( \omega_X \) obtained by changing the value of \( l \) in \( \omega_X \) satisfies \( F \), contradicting the assumption of the lemma. \( \square \)

As a consequence, if a formula is satisfied by a model where \( x \) is true but not by the same model where \( x \) is false, and vice versa, then any formula equivalent to it contains both \( x \) and \( \neg x \). Therefore, if a formula contains \( x \neq y \), either conjoined with a satisfiable formula not containing \( x \) and \( y \) or disjoined with a non-valid formula not containing \( x \) and \( y \), then it must contain at least two literal occurrences for \( x \) and two for \( y \). The following lemma shows a sufficient condition for the presence of a literal in a formula.

**Lemma 2:** Let \( F \) be a formula over \( X \cup Y \). For any truth evaluation \( \omega_X \), no formula equivalent to \( F \) is smaller than the smallest formula equivalent to \( F|_{\omega_X} \).

**Proof:** Let \( T \) be a formula equivalent to \( F \). Equivalence is preserved when replacing a variable with a truth value in both formulae. As a result, \( F|_{\omega_X} \equiv T|_{\omega_X} \). Furthermore, such a replacement does not increase the number of literal occurrences in \( T \), since it only replace some variables with either true or false. As a result, the size of \( T|_{\omega_X} \) is less than or equal to the size of \( T \). Since \( T|_{\omega_X} \) is a formula equivalent to \( F|_{\omega_X} \), it is at least as large as the smallest formula equivalent to \( F|_{\omega_X} \). Since \( T \) is larger or has the same size, the claim is proved. \( \square \)

This lemma is useful when formulae contain parts that are satisfiable only for a specific truth evaluation of some variables \( X \). Such formulae are built to the aim of generating a (relatively) large subformula whenever a condition is met.

**III. THE SIZE OF CIRCUMSCRIPTIVE FORMULAE**

In this section, we analyze the problem of deciding whether the circumscription of a formula can be represented by a formula of size bounded by an integer \( k \), in unary notation. The unary notation is used to avoid exponentially-sized formulae to be taken into account. Equivalently, the problem could be reformulated as: is there any formula that is equivalent to \( \text{CIRC}(T) \) and has size less or equal than another formula \( G \)?

**Theorem 1:** The problem of deciding whether \( \text{CIRC}(T) \) is equivalent to a formula \( F \) with \(|F| \leq k \), where \( k \) is a number in unary notation, is in \( \Sigma_2^p \).

**Proof:** The problem can be reformulated as follows: check whether there exists a formula \( F \) that is equivalent to \( \text{CIRC}(T) \) and \(|F| \leq k \). The problem \( F \models \text{CIRC}(T) \) is in coNP, since it amounts to check whether \( \omega \notin \omega' \) for every \( \omega \models T \) and \( \omega' \models F \). Since coNP is a subclass of \( \Pi_2^p \), this problem is also in \( \Pi_2^p \). The problem \( \text{CIRC}(T) \models F \) is instead \( \Pi_2^p \)-complete [12, 13, 14]; therefore, it is in \( \Pi_2^p \). The problem under consideration can be therefore solved by guessing a formula \( F \) of size bounded by \( k \) and then checking whether \( F \models \text{CIRC}(T) \) and \( \text{CIRC}(T) \models F \). Since both problems are in \( \Pi_2^p \), they can be checked by reversing the result of a \( \Sigma_2^p \) oracle. The problem can therefore be solved by a first nondeterministic step generating all formulae \( F \) with \(|F| \leq k \) and then by calling the oracle. It is therefore in \( \Sigma_3^p \). \( \square \)

The problem can be proved hard for the class \( \Pi_2^p \).

**Theorem 2:** The problem of deciding whether \( \text{CIRC}(T) \) is equivalent to a formula \( T' \) with \(|T'| \leq k \) is \( \Pi_2^p \)-hard.

**Proof:** Let \( F \) be a formula over variables \( X \cup Y \). The proof shows how to build in polynomial time a formula \( T \) and
a number \( k \) in unary notation such that \( \forall X \exists Y . F \) is valid if and only if \( \text{CIRC}(T) \) is equivalent to a formula of size \( \leq k \).

Let us assume, without loss of generality, that \( |X| = |Y| = n \). The reduction introduces a set of new variables \( X' \) in one-to-one correspondence with \( X \). It also introduces a set of new variables \( Y' \) in one-to-one correspondence with \( Y \) and a set of new variables \( Z \) of cardinality \( m = 3n + ||F|| + 1 \).

In this proof the following notations are used, where \( X \) and \( X' \) are sets of variables in one-to-one correspondence and each \( x \) corresponds to \( x' = c(x) \):

\[
\begin{align*}
X^\equiv & = \{ \neg x \mid x \in X \} \\
X \equiv X' & = \bigwedge \{ x \equiv x' \mid x \in X, \ x' = c(x) \} \\
X \not\equiv X' & = \bigwedge \{ x \not\equiv x' \mid x \in X, \ x' = c(x) \}
\end{align*}
\]

Formula \( T \) and number \( k \) are as follows:

\[
T = (X \not\equiv X') \land \left( (\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \right)
\]

\[
k = 14n + 3||F|| + 2
\]

The reduction works as follows: \( X \not\equiv X' \) allows expressing \( \text{CIRC}(T) \) in terms of the disjunction of \( \text{CIRC}(T|_{\omega_X}) \) for all possible \( \omega_X \); if \( \forall X \exists Y . F \) is true, all these formul\( \text{CIRC}(T|_{\omega_X}) \) can be expressed in the same way, so that a single formula equivalent to \( \text{CIRC}(T) \) exists with size bounded by \( k \); otherwise, for the evaluation \( \omega_X \) that makes \( F \) false \( \text{CIRC}(T|_{\omega_X}) \) alone has size greater than \( k \).

The first step employs the third equivalence of Property 1, when applied to every \( x \in X \) and its respective \( x' \in X' \), since \( T \) contains \( X \not\equiv X' \):

\[
\text{CIRC}(T) \equiv \bigvee_{\omega_X} \text{Form} (\omega_X) \text{CIRC}(T|_{\omega_X})
\]

The second step of the proof is to analyze \( \text{CIRC}(T|_{\omega_X}) \) for an evaluation \( \omega_X \). Formula \( T|_{\omega_X} \) can be rewritten as follows:

\[
T|_{\omega_X} \equiv \left( (X \not\equiv X') \land \left( (\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \right) \right|_{\omega_X}
\]

\[
= (X \not\equiv X'|_{\omega_X}) \land \left( (\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \right)_{|\omega_X}
\]

In this last formula, \( \omega_X \), is the evaluation of \( X' \) setting each variable in \( X' \) to the opposite value of the corresponding variable in \( X \). This formula does not contain any variable in \( X \). Therefore, \( \text{CIRC}(T|_{\omega_X}) \) is defined by taking into account only the other variables: \( X' \), \( Y \), \( Y' \) and \( Z \). Since \( X' \) has a fixed value, it holds:

\[
\text{CIRC}(T|_{\omega_X}) \equiv \text{Form}(\omega_X) \land \text{CIRC}(\left((\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \right)
\]

The first subformula of circumscription \( ((\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \) has only models \( \omega_Y \cup \omega'_Y, \omega_Z \cup \omega'_Z \), in which \( \omega_Y \) and \( \omega'_Y \) set all variables in \( Y \) and \( Y' \) to false. This model contains a model of the second subformula if \( F|_{\omega_X} \) is satisfiable. Indeed, let \( \omega_Y \) be the model that satisfies \( F|_{\omega_X} \). This model contains a model of \( \omega_Y \). The model \( \omega_Y \) assigns \( y' \in Y' \) to true if and only if the corresponding \( y \in Y \) is false in \( \omega_Y \) also satisfies \( (F|_{\omega_Y} \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \), and is contained in \( \omega_Y \). A model of the second subformula is therefore \( \omega_Y \cup \omega_Y', \omega_Z \cup \omega_Z', \omega_Z \cup \omega'_Z \), where \( \omega_Z \cup \omega'_Z \) set all variables to false and are therefore contained in \( \omega_Z \cup \omega'_Z \).

This proves that every model of the first subformula contains a model of the second, if \( F|_{\omega_X} \) is satisfiable. If this is the case, the first subformula is irrelevant to circumscription. Otherwise, the second subformula is unsatisfiable.

\[
\text{CIRC} \left( \left( (\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \right) \right)
\]

\[
= \text{CIRC}(F|_{\omega_X} \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z'))
\]

if \( F|_{\omega_X} \) is satisfiable

\[
= \text{CIRC}((\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \land (F \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \) otherwise
\]

The rest of the proof depends on whether \( F \) is satisfiable for every \( \omega_X \). If it is, then \( \text{CIRC}(T|_{\omega_X}) \) is equivalent to \( \omega_Y \land \text{CIRC}(F|_{\omega_X} \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z')) \) for every \( \omega_X \). As a result, \( \text{CIRC}(T) \) is equivalent to \( \text{CIRC}((X \not\equiv X') \land (\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z')) \), which is equivalent to \( (X \not\equiv X') \land (Y \not\equiv Y') \land (\bigwedge Z^\land \bigwedge Z') \) by Property 1. This formula has size \( 4n + ||F|| + 4n + 2 = 8n + ||F|| + 2 ||F|| + 2 = 14n + 3||F|| + 2 = k \).

If \( F \) is false for some \( \omega_X \), then \( \text{CIRC}(T|_{\omega_X}) \) is equivalent to \( \text{Form}(\omega_X) \land \text{CIRC}((\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z')) \), which is also equivalent to \( (\exists Y \land \bigwedge Y) \land (\exists Z \land \bigwedge Z') \) by applying the second and third equivalence of Property 1. For every \( z \in Z \), this formula describes a model that makes \( z \) true, but changing only the evaluation of \( z \) results in a model not satisfying this formula. The same applies to all variables in \( Z \) and \( Z' \) and their negation, and to all variables in \( Y \) and \( Y' \). By Lemma 1, every formula equivalent to this one has size greater than or equal to \( 4|Z| + 2|Y| = 4m + 2n = 4(3n + ||F|| + 1) + 2n = 12n + 4||F|| + 4 + 2n = 16n + 4||F|| + 4 > k \). By Lemma 2, every formula equivalent to \( \text{CIRC}(T) \) has size greater than or equal to this amount.
IV. Conclusions

The problem of checking whether the circumscription of a formula can be represented by a formula of size bounded by $k$ turned out to be $\Pi_2^p$-hard in $\Sigma_3^p$. These two classes are at the second and third level of the polynomial hierarchy, respectively. As a result, the problem cannot be solved by a propositional satisfiability solver. It can, however, be translated into a QBF and then passed as input to one of the existing QBF solvers [15].

An open question is how much complexity decreases if the formulae are in Horn form, and in particular if some additional restriction makes the problem tractable. If $k$ is in binary representation rather than unary, the question is whether $\text{CIRC}(T)$ can be represented by a formula that may be exponential, but still bounded by $k$. The necessity of considering such large formulae is likely to make this problem harder than with $k$ in unary notation: polynomial space may not be sufficient to solve it.

Indeed, assuming $k$ in unary notation amounts to requiring the equivalent formula to have size comparable to that of the input data. This is equivalent to ask whether $\text{CIRC}(T)$ is equivalent to a formula of the same size of another formula $G$, for example. Allowing $k$ to be stored in binary form with $n$ bit allows the bound be as large as $2^n - 1$. As a result, even formulae of exponential size are allowed as representations of $\text{CIRC}(T)$. What complicates the analysis is that the usual guess-and-check algorithm for finding such a formula does not work in polynomial space, as this may not be enough for even storing the formula. A cycle of the minimal models of $T$ is still feasible, but this may not allow determining the size of a formula satisfied exactly by all of them, unless such a formula is explicitly produced.

ACKNOWLEDGEMENTS

The author expresses his profound gratitude to Roberto Baldoni and Marco Schaerf for their invaluable suggestions during the writing of this article.

REFERENCES