On Quantifying Literals in Boolean Logic and Its Applications to Explainable AI
(Extended Abstract)

Adnan Darwiche¹, Pierre Marquis²,³
¹Computer Science Department, UCLA
²Univ. Artois, CNRS, CRIL
³Institut Universitaire de France
darwiche@cs.ucla.edu, marquis@cril.fr

Abstract

Quantified Boolean logic results from adding operators to Boolean logic for existentially and universally quantifying variables. This extends the reach of Boolean logic by enabling a variety of applications that have been explored over the decades. The existential quantification of literals (variable states) and its applications have also been studied in the literature. We complement this by studying universal literal quantification and its applications, particularly to explainable AI. We also provide a novel semantics for quantification and discuss the interplay between variable/literal and existential/universal quantification. We further identify classes of Boolean formulas and circuits that allow efficient quantification. Literal quantification is more fine-grained than variable quantification, which leads to a refinement of quantified Boolean logic with literal quantification as its primitive.

1 Introduction

To quantify a variable from a Boolean formula is to eliminate that variable from the formula. This quantification process can be performed either existentially or universally, each leading to a different semantics and a different set of applications. The use of quantification in Boolean logic dates back at least to George Boole’s work [1854] where he used universal quantification to perform some forms of logical deduction. Existential quantification has a particularly intuitive interpretation as it can be viewed as a process of removing from a formula all and only the information which pertains to the quantified variable. Due to this semantics, which is referred to as forgetting [Lin and Reiter, 1994], existential quantification has received much attention in AI and database theory, particularly in the management of inconsistent information. This use dates back at least to [Weber, 1986] who employed existential quantification to combine pieces of information that may contradict each other. Variable quantification plays a prominent role in complexity theory too since deciding the validity of quantified Boolean formulas (QBFs) is the canonical PSPACE-complete problem; see, for example, [Papadimitriou, 1994]. The validity of QBFs has also been used to characterize the polynomial hierarchy [Stockmeyer, 1977]. In contrast to AI where existential quantification has had a more dominant role than universal quantification, the role of these two forms of quantification has been more symmetrical in other areas of computer science particularly in complexity theory.

The interpretation of existential variable quantification as a process of forgetting information prompted a refinement in which literals (states of variables) are also existentially quantified [Lang et al., 2003], allowing one to remove information that pertains to a literal from a formula. Existential literal quantification is more fine-grained than existential variable quantification as the latter can be formulated in terms of the former. This has led to further applications which are reviewed in [Darwiche and Marquis, 2021].

The dominance of existential quantification within AI is worth noting. We attribute this to the lack of an intuitive enough interpretation of universal quantification, which is a gap that we address in this work. We started our study by an observation that some recent work on explainable AI [Darwiche and Hirth, 2020] can be formulated in terms of universal quantification, particularly universal literal quantification which has not been discussed in the literature before this work. This prompted us to formalize this notion and to elaborately investigate its connections to explainable AI. Our investigation led us to interpret universal quantification as a selection process (in contrast to a forgetting process), which gave rise to many implications. It also led us to some results on the efficient computation of universal literal quantification (e.g., being tractable on CNFs) which has further implications on the efficient computation of explainable AI queries.

In explainable AI, one is typically interested in reasoning about the behavior of classifiers which make decisions on instances. For example, one may wish to understand why a classifier made a particular decision. One may also wish to determine whether a decision is biased (i.e., would be different if we were to only change some protected features of the instance). Such classifiers can be represented using Boolean formulas, even in some cases where they have a numeric form that is learned from data.¹ A major insight underlying our results is that many explainable AI queries correspond to a pro-

¹One can capture the input-output behavior of machine learning classifiers with discrete features using Boolean formulas, either through an encoding or a compilation process. This has been
cess of selecting instances with particular properties, where the
Boolean formula characterizing such instances constitutes
the answer to the explainable AI query. Moreover, the
universal quantification of literals can be used to select such in-
stances. We use this insight to formulate and generalize some
of the recently introduced notions in explainable AI.

We limit the scope of this extended abstract to the defini-
tion of universal literal quantification, some of its properties
and to only one of its applications to explainable AI. We will
also discuss the selection semantics of universal literal quan-
tification. For a more comprehensive and in-depth treatment,
including proofs of results, we defer the reader to the corre-
sponding journal article [Darwiche and Marquis, 2021].

2 Classifiers as Boolean Formulas

We assume a finite set of Boolean variables \( \Sigma \) where \( x \) and \( \bar{x} \) denote the positive and negative literals of Boolean variable \( X \), respectively. A world is a set of literals which contains
exactly one literal for each variable in \( \Sigma \). If \( \Sigma = \{ X, Y, Z \} \),
then \( \{ x, y, z \} \) and \( \{ \bar{x}, \bar{y}, \bar{z} \} \) would be examples of worlds.
We will sometimes simply use \( xy\bar{z} \) and \( x\bar{y}\bar{z} \) to denote such worlds.
We use \( \top \) and \( \bot \) to denote the Boolean constants true and false, and \( \phi = \phi \) to mean that Boolean formula \( \phi \) implies
the Boolean formula \( \phi \). Finally, \( \phi \mid x \) denotes the formula obtained
by replacing every occurrence of variable \( X \) in \( \phi \) with \( x \).
Similarly, \( \phi \mid \bar{x} \) is the formula obtained by replacing every
occurrence of variable \( X \) in \( \phi \) with \( \bar{x} \).

We will restrict our discussion to classifiers over binary features and two classes (positive and negative). A classifier
is then represented by a Boolean formula \( \Delta \), where variable
assignments that satisfy \( \Delta \) correspond to positive instances
and those that satisfy \( \neg \Delta \) correspond to negative instances.

Definition 1. Let \( \Delta \) be a Boolean formula over variables
\( X_1, \ldots, X_n \). We call \( \Delta \) a classifier, variable \( X_i \) a feature,
literal \( \ell_i \) a characteristic, and \( \delta = \{ \ell_1, \ldots, \ell_n \} \) an instance.
The decision of classifier \( \Delta \) on instance \( \delta \) is denoted \( \Delta(\delta) \).
It is defined as \( \Delta(\delta) = 1 \) if \( \delta \models \Delta \) (positive decision) and
\( \Delta(\delta) = 0 \) if \( \delta \models \neg \Delta \) (negative decision). We further define
\( \Delta_\delta = \Delta \) when the instance \( \delta \) is positive and \( \Delta_\delta = \neg \Delta \) when
the instance \( \delta \) is negative.

Since \( \Delta \) captures positive instances and \( \neg \Delta \) captures neg-ative instances, we work with \( \Delta \) when reasoning about post-
itive decisions and with \( \neg \Delta \) when reasoning about negative
decisions. This explains the significance of the notation \( \Delta_\delta \).

Consider the following classifier for admitting students
into an academic program [Darwiche and Hirth, 2020]. This
classifier makes its decision based on five features: whether
an applicant passed the entrance exam (\( E \)), is a first time ap-
plicant (\( F \)), has good grades (\( G \)), has work experience (\( W \))
and comes from a rich hometown (\( R \)). The classifier is spec-
ified by the following Conjunctive Normal Forms (CNFs):

\[
\Delta = (e \lor g) \land (e \lor r) \land (e \lor w) \land (f \lor r) \land (f \lor g \lor w)
\]
\[
\neg \Delta = (e \lor f \lor r) \land (e \lor f \lor g) \land (e \lor f \lor w) \land (g \lor f \lor w)
\]

Consider now an applicant who does not come from a rich
hometown but satisfies all other requirements. This appli-
cant will be admitted, \( \Delta(e, f, g, w, \bar{r}) = 1 \). The applicant
will not be admitted if they did not pass the entrance exam,
\( \Delta(e, f, g, w, r) = 0 \). Instance \( \delta_1 = \{ e, f, g, w, \bar{r} \} \) is positive and instance \( \delta_2 = \{ e, f, g, w, r \} \) is negative. Using our nota-
tion, \( \Delta_{\delta_1} = \Delta \) since \( \delta_1 \models \Delta \) and \( \Delta_{\delta_2} = \neg \Delta \) since \( \delta_2 \models \neg \Delta \).

3 Universal Literal Quantification

We next define the universal quantification of literal \( \ell \) from
a Boolean formula \( \phi \) and show how it can be used to reason
about the decisions of classifiers (explain them in particular).
We first note that \( \phi \) can be expanded as \( \phi = (\ell \lor (\phi(\ell))) \lor
(\ell \lor (\phi(\ell))) \), which is equivalent to what is known as Boole’s
or Shannon’s expansion, \( \phi = (\ell \lor (\phi(\ell))) \lor (\ell \lor (\phi(\ell))) \).

Definition 2. Universally quantifying literal \( \ell \) from formula
\( \phi \) is defined as follows: \( \forall \ell \cdot \phi = (\ell \lor (\phi(\ell))) \lor (\ell \lor (\phi(\ell))) \).

That is, the operator \( \forall \ell \) drops literal \( \ell \) from the expansion
\( \phi = (\ell \lor (\phi(\ell))) \lor (\ell \lor (\phi(\ell))) \). Universal quantification is
classically applied to variables: \( \forall X \cdot \phi = (\phi[x] \lor (\phi[x])) \).
Consider the formula \( \phi = (x \lor y) \land (y \lor x) \) which says that
variables \( X \) and \( Y \) are equivalent. We have \( \forall x \cdot \phi = x \land y \)
and \( \forall x \cdot \phi = x \lor y \). Note, however, that \( \forall X \cdot \phi = \bot \).
Moreover, \( \forall x \cdot (\forall x \cdot \phi) = \forall x \cdot (\forall x \cdot \phi) = \top \). For the formula
\( \phi = (x \lor y) \), we have \( \forall x \cdot \phi = y \), \( \forall x \cdot \phi = (x \lor y) \) and
\( \forall x \cdot \phi = y \). Moreover, we have \( \forall x \cdot (\forall x \cdot \phi) = \forall x \cdot (\forall x \cdot \phi) = \top \).
[Darwiche and Marquis, 2021] studied various properties of
universal literal quantification, including the following ones.

- Universal literal quantification is commutative: For literals \( \ell_1, \ell_2 \) and formula \( \phi \), \( \forall \ell_1 \cdot (\forall \ell_2 \cdot \phi) = \forall \ell_2 \cdot (\forall \ell_1 \cdot \phi) \).
Thus, the order of quantifying literals does not matter so we
can write \( \forall \ell_1, \ldots, \ell_n \cdot \phi \) or \( \forall \ell_1, \ldots, \ell_n \cdot \phi \).
- Universal literal quantification strengthens a formula:
For literal \( \ell \) and formula \( \phi \), we have \( \forall \ell \cdot \phi \models \phi \).
- Universal literal quantification is more fine-grained than
universal variable quantification: For variable \( X \) and
formula \( \phi \), we have \( \forall X \cdot \phi \models \forall x \cdot \phi = (\forall x \cdot \phi) \).
- Universal literal quantification preserves logical implication:
For literal \( \ell \), formulas \( \phi \) and \( \phi \), we have \( \phi \models \phi \) only
if \( \forall \ell \cdot \phi \models \forall \ell \cdot \phi \).
- Universal literal quantification is the dual of existential
literal quantification: For literal \( \ell \) and formula \( \phi \), we have
\( \forall \ell \cdot \phi = \neg (\exists \ell \cdot (\neg \phi)) \).

4 An Application to Explainable AI

We next consider an application of universal literal quantifi-
cation to explaining the decisions of classifiers. Consider a
decision \( \Delta(\delta) \) made by classifier \( \Delta \) on instance \( \delta \). One way to
explain this decision is to identify a minimal set of characteristics \(\gamma \subseteq \delta\) that is sufficient to trigger the decision, \(\gamma \models \Delta\). This notion of explanation was introduced in [Shih et al., 2018] under the name of a \(PI\)-explanation and later called a sufficient reason for the decision [Darwiche and Hirth, 2020] and an abductive explanation [Ignatiev et al., 2019a].

Consider the admission classifier above and an applicant who does not come from a rich hometown but satisfies all other requirements. This applicant will be admitted, \(\Delta(e, f, g, w, r) = 1\), and the decision has two sufficient reasons: \(\{e, f, g\}\) and \(\{e, f, w\}\). Passing the entrance exam \(e\), being a first-time applicant \(f\) and having good grades \(g\) guarantee admission yet no strict subset of these characteristics will. Having work experience instead of good grades will also guarantee admission but again no strict subset of characteristics \(\{e, f, w\}\) guarantees this. If this applicant were to come from a rich hometown, then there would be more sufficient reasons for the admission decision, \(\Delta(e, f, g, w, r) = 1\): \(\{e, f, g\}, \{e, f, w\}, \{e, g, r\}, \{e, r, w\}\) and \(\{g, r, w\}\).

A decision may have an exponential number of sufficient reasons. Moreover, some explainable AI queries, such as ones relating to decision bias, are based on checking whether the sufficient reasons for a decision satisfy some properties. These sufficient reasons can sometimes be represented compactly using the notion of a complete reason introduced in [Darwiche and Hirth, 2020], which showed a number of results relating to this notion. For example, the sufficient reasons for a decision correspond to the prime implicants of its complete reason. Moreover, some properties of sufficient reasons can be checked in time linear in the size of a complete reason when it is represented in an appropriate form. However, the process for computing complete reasons in [Darwiche and Hirth, 2020] was procedural and restricted to a specific type of classifiers. Universal literal quantification can be used to provide a semantical definition of complete reasons which decouples them from the type of underlying classifier.

**Theorem 1.** For classifier \(\Delta\) and instance \(\delta = \{\ell_1, \ldots, \ell_n\}\), \(\forall \delta \cdot \Delta_\delta\) is equivalent to the complete reason for decision \(\Delta(\delta)\).

As shown in [Darwiche and Marquis, 2021], a number of logical forms allow a linear-time computation of universal literal quantification, including Conjunctive Normal Form (CNF). In particular, if \(\Delta\) is a CNF, then \(\forall \ell \cdot \Delta\) can be obtained by dropping every occurrence of literal \(\ell\) from the CNF.

Consider the admission classifier above and an applicant who passed the entrance exam, has good grades and work experience, comes from a rich hometown but is not a first time applicant. The classifier will admit this applicant, \(\Delta(e, f, g, w, r) = 1\). The complete reason for this decision is \(\forall e, f, g, r, w \cdot \Delta = (e \lor g) \land (e \lor w) \land (r \lor (f \lor g \lor w))\), which has four prime implicants representing the sufficient reasons for the decision: \(\{e, g, r\}, \{e, r, w\}, \{e, f, r\}, \{g, r, w\}\).

[Darwiche and Marquis, 2021] discussed further applications of universal literal quantification, which include determining decision bias and identifying characteristics that are irrelevant to a decision. It also showed how this notion can be used to reason about populations, which are groups of instances that are decided similarly.

### 5 Selection Semantics

As mentioned earlier, the popularity of existential quantification in AI is due to its forgetting semantics. In particular, to existentially quantify a variable \(X\) from a formula \(\varphi\) is to remove any information, and only information, that the formula \(\varphi\) contains about variable \(X\). Similarly, to existentially quantify a literal from a formula is to remove any, and only, information that the formula contains about the literal. We will next provide a selection semantics for universal literal quantification; see [Darwiche and Marquis, 2021] for the corresponding semantics of universal variable quantification.

Our semantics is based on the notion of \(\alpha\)-independent models for a formula \(\varphi\), where \(\alpha\) is a set of literals. These are models of formula \(\varphi\) that do not depend on literals \(\alpha\); that is, they continue to be models of \(\varphi\) if we flip any of the literals they may have in \(\alpha\) (a model of \(\varphi\) is a world that satisfies it).

**Definition 3.** Let \(\alpha\) be a set of literals for distinct variables. A world \(\omega\) is said to be an \(\alpha\)-independent model of formula \(\varphi\) iff \(\alpha \subseteq \omega\) and \(\omega \setminus \alpha \models \varphi\).

Consider a world \(\omega\) and literals \(\alpha\) such that \(\alpha \subseteq \omega\). If world \(\omega\) is an \(\alpha\)-independent model of formula \(\varphi\), we can flip any literals of \(\alpha\) in world \(\omega\) while maintaining the world as a model of \(\varphi\). Consider formula \(\varphi = (x \lor y) \land z\) and its model \(\omega = \{x, y, z\}\). This model is not an \((x, y)\)-independent model of \(\varphi\) since \(\{x, \bar{y}, z\}\) is not a model of \(\varphi\). On the other hand, world \(\{x, y, \bar{z}\}\) is an \((x, y)\)-independent model of \(\neg \varphi\) as we can flip literals \(x\) and \(y\) in any manner while maintaining the world as a model for \(\neg \varphi\).

We are now ready to present our selection semantics for universally quantifying a set of (possibly conflicting) literals \(\ell_1, \ldots, \ell_n\), from a formula \(\varphi\). According to this semantics, universally quantifying these literals is a process of selecting models of \(\varphi\) that do not depend on literals \(\ell_1, \ldots, \ell_n\).

**Theorem 2.** Let \(\varphi\) be a formula, \(\ell_1, \ldots, \ell_n\) be literals, \(\omega\) be a world and \(\alpha = \omega \cap \{\ell_1, \ldots, \ell_n\}\). Then \(\omega \models \forall \ell_1, \ldots, \ell_n \cdot \varphi\) iff \(\alpha\) is an \(\alpha\)-independent model of \(\varphi\).

Consider \(\varphi = (x \lor y \lor z) \land (\bar{x} \lor y \lor v \lor t)\), literals \(x, y, z\) and world \(\omega = \bar{x} \lor y \lor t\), leading to \(\alpha = \omega \cap \{x, \bar{y}, \bar{z}\} = \bar{y} \lor \bar{z}\). Then \(\omega\) is an \(\alpha\)-independent model of \(\varphi\) since \(zt \models \varphi\). Hence, \(\omega \models \forall x, y, z \cdot \varphi\) which can be verified since \(\forall x, y, z \cdot \varphi = (x \lor y \lor z) \land (\neg y \lor \neg t)\). For literals \(x, \bar{y}, \bar{z}\), we get \(\alpha = \omega \cap \{\bar{x}, y, z\} = \bar{x} \lor \bar{z}\) so \(\omega\) is not an \(\alpha\)-independent model of \(\varphi\) since \(\forall t \not\models \varphi\) (indeed, \(\bar{x} \lor y \lor t \models \neg \varphi\)). Hence, \(\omega \not\models \forall x, \bar{y}, \bar{z} \cdot \varphi\) which can be verified since \(\forall x, \bar{y}, \bar{z} \cdot \varphi = x \lor \bar{t}\).

According to Theorem 2, when universally quantifying literals \(\ell_1, \ldots, \ell_n\), from formula \(\varphi\), we are selecting all (and only) models of \(\varphi\) that do not depend on literals \(\ell_1, \ldots, \ell_n\). That is, the models of \(\forall \ell_1, \ldots, \ell_n \cdot \varphi\) are precisely the models of \(\varphi\) which continue to be models of \(\varphi\) if we were to flip any literals they may have in \(\ell_1, \ldots, \ell_n\). We next discuss this selection semantics in the context of classification.

Consider a classifier whose positive instances are specified by formula \(\Delta\) and suppose we universally quantify a set of characteristics \(\ell_1, \ldots, \ell_n\) from \(\Delta\) to yield \(\forall \ell_1, \ldots, \ell_n \cdot \Delta\). In this case, \(\forall \ell_1, \ldots, \ell_n \cdot \Delta\) characterizes all positive instances \(\delta\) whose positiveness does not depend on characteristics \(\ell_1, \ldots, \ell_n\). That is, flipping any characteristics in
\( \delta \cap \{ \ell_1, \ldots, \ell_n \} \) will yield a positive instance. A symmetric situation arises when universally quantifying characteristics \( \ell_1, \ldots, \ell_n \) from \( \neg \Delta \): We select negative instances whose negativeness does not depend on characteristics \( \ell_1, \ldots, \ell_n \).

For a concrete example, consider the following classifier which decides whether an applicant should be granted a loan based on four features: whether they defaulted on a previous loan \( (D) \), have a guarantor \( (G) \), own a home \( (H) \) or have a high income \( (I) \). This classifier is specified by the following formula \( \Delta \) and its negation \( \neg \Delta \), both in CNF:

\[
\Delta = (h \lor i) \land (\bar{d} \lor g) \land (d \lor i)
\]

\[
\neg \Delta = (d \lor \bar{i}) \land (d \lor \bar{h}) \land (\bar{g} \lor \bar{i})
\]

This classifier will grant a loan to an applicant who never defaulted on a loan, owns a home, has a high income but does not have a guarantor, \( \Delta(d, g, h, i) = 1 \). This positive instance is not a model of \( \forall \delta \Delta = i \land g \). Thus, characteristic \( \bar{d} \) is essential for the positiveness of this instance (instance \( \{d, \bar{g}, h, i\} \) is negative). However, the positive instance \( \{d, g, h, i\} \) is a model of \( \forall \delta \Delta \) so characteristic \( d \) is not essential for its positiveness (instance \( \{d, g, \bar{h}, \bar{i}\} \) is also positive).

If the characteristics \( \ell_1, \ldots, \ell_n \) correspond to a positive instance \( \delta \), then \( \delta = \{ \ell_1, \ldots, \ell_n \} \models \Delta \). In this case, \( \forall \ell_1, \ldots, \ell_n \cdot \Delta \) characterizes all positive instances \( \delta' \) whose positiveness is due only to the characteristics they have in common with \( \delta \); that is, independently of their characteristics \( \delta' \cap \{ \ell_1, \ldots, \ell_n \} \). Hence, \( \forall \ell_1, \ldots, \ell_n \cdot \Delta \) will characterize all subsets of instance \( \delta \) which are sufficient to trigger a positive decision (these subsets correspond to the sufficient reasons for the decision). This explains why universal literal quantification can be used to compute the complete reason behind a decision as shown earlier. Due to this selection semantics of universal literal quantification, [Darwiche and Marquis, 2021] showed that it can be used to answer further queries of interest to explainable AI. This includes queries relating to decision bias and to the irrelevance of certain characteristics and features to decisions.

Acknowledgments

This work has benefited from the support of the International Research Project MAKC (“Modern Approaches to Knowledge Compilation”) shared between the Automated Reasoning Group of the University of California at Los Angeles (UCLA) and the Centre de Recherche en Informatique de Lens (CRIL, UMR 8188 CNRS – Artois University). The work has been partially supported by NSF grant IIS-1910317, DARPA grant N66001-17-2-4032, AI Chair EXPEKCTATION (ANR-19-CHIA-0005-01) of the French National Research Agency (ANR) and by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

[Audemard et al., 2020] Gilles Audemard, Frédéric Koriche, and Pierre Marquis. On tractable XAI queries based on compiled representations. In Proc. of KR’20, pages 838–849, 2020.

[Boole, 1854] G. Boole. An investigation of the laws of thought. Walton and Maberley, London, 1854.

[Chan and Darwiche, 2003] Hei Chan and Adnan Darwiche. Reasoning about Bayesian network classifiers. In Proc. of UAI’03, pages 107–115, 2003.

[Choi et al., 2020] Arthur Choi, Andy Shih, Anchal Goyanka, and Adnan Darwiche. On symbolically encoding the behavior of random forests. In Proc. of FoMAS’20, 3rd Workshop on Formal Methods for ML-Enabled Autonomous Systems, 2020.

[Darwiche and Hirth, 2020] Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In Proc. of ECAI’20, pages 712–720, 2020.

[Darwiche and Marquis, 2021] Adnan Darwiche and Pierre Marquis. On quantifying literals in boolean logic and its applications to explainable AI. J. Artif. Intell. Res., 72:285–328, 2021.

[Ignatiev et al., 2019a] Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. Abduction-based explanations for machine learning models. In Proc. of AAAI’19, pages 1511–1519, 2019.

[Ignatiev et al., 2019b] Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. On relating explanations and adversarial examples. In Proc. of NeurIPS’19, pages 15857–15867, 2019.

[Lang et al., 2003] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence: Formulavariable independence and forgetting. J. Artif. Intell. Res., 18:391–443, 2003.

[Lin and Reiter, 1994] F. Lin and R. Reiter. Forget it! In Proc. of AAAI Fall Symposium on Relevance, pages 154–159, 1994.

[Narodytska et al., 2018] Nina Narodytska, Shiva Prasad Kasiviswanathan, Leonid Ryzhyk, Mooy Sagiv, and Toby Walsh. Verifying properties of binarized deep neural networks. In Proc. of AAAI’18, pages 6615–6624, 2018.

[Papadimitriou, 1994] Christos H. Papadimitriou. Computational complexity. Addison–Wesley, 1994.

[Shi et al., 2020] Weijia Shi, Andy Shih, Adnan Darwiche, and Arthur Choi. On tractable representations of binary neural networks. In Proc. of KR’20, pages 882–892, 2020.

[Shih et al., 2018] Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining Bayesian network classifiers. In Proc. of IJCAI’18, pages 5103–5111, 2018.

[Shih et al., 2019] Andy Shih, Arthur Choi, and Adnan Darwiche. Compiling Bayesian network classifiers into decision graphs. In Proc. of AAAI’19, pages 7966–7974, 2019.

[Stockmeyer, 1977] Larry J. Stockmeyer. The polynomial-time hierarchy. Theoretical Computer Science, 3:1–22, 1977.

[Weber, 1986] Andreas Weber. Updating propositional formulas. In Proc. of EDS’86, 1st Int. Conf. on Expert Database Systems, pages 487–500, 1986.