Quantified Boolean Formulas: Complexity and Expressiveness

Uwe Bubeck

University of Paderborn

20.11.2012

Outline

- Introduction
- Free Variables and Equivalence
- Complexity
- Expressiveness
- Conclusion

Section 1

Introduction

Quantified Boolean Formulas 1/2

QBF extends propositional logic by allowing universal and existential quantifiers over propositional variables.

Inductive definition:

1. Every propositional formula is a QBF.
2. If Φ is a QBF then $\forall x \Phi$ and $\exists y \Phi$ are also QBFs.
3. If Φ_1 and Φ_2 are QBFs then $\neg\Phi_1$, $\Phi_1 \land \Phi_2$ and $\Phi_1 \lor \Phi_2$ are also QBFs.

Quantified Boolean Formulas 2/2

In a closed QBF, every variable is quantified.

Semantics definition for closed QBF:

$\exists y\ \Phi(y)$ is true if and only if

$\Phi[y/0]$ is true or $\Phi[y/1]$ is true.

$\forall x\ \Phi(x)$ is true if and only if

$\Phi[x/0]$ is true and $\Phi[x/1]$ is true.

$\Rightarrow$ A closed QBF is either true or false.

Prenexing

Closed QBFs Φ and Ψ are logically equivalent ($\Phi \approx \Psi$)

$\Leftrightarrow$ they are satisfiability equivalent

$\Leftrightarrow$ they both evaluate to the same truth value.

Every QBF can be transformed in linear time into a logically equivalent prenex formula $Q_1 v_1 \ldots Q_k v_k \, \phi$ by:

1. renaming quantified variables,
2. transformation into negation normal form (NNF),
3. moving quantifiers to the front by maxiscoping:

   $(Qv \, \Phi) \circ \Psi \approx Qv \, (\Phi \circ \Psi)$ for $\circ \in \{\wedge, \vee\}$ and $Q \in \{\forall, \exists\}$.

Tree Models: Definition

A closed prenex QBF $Q_1 v_1 \dots Q_k v_k\ \phi$ is true if and only if there exists a tree such that: [Samulowitz et. al., 2006]

1. Each inner node is labeled with a variable v_i, $1 \leq i \leq k$, its outgoing edges are labeled with $v_i = 0$ or $v_i = 1$.
2. Inner nodes labeled with v_i have two children if and only if $Q_i = \forall$, and one child otherwise.
3. For each path from root to leaf labeled with $(v_{i_1}, \dots, v_{i_j})$, we have $1 \leq i_1 < \cdots < i_j \leq k$ (i.e. order as in the prefix).
4. On each path from root to leaf, the edge labels are a satisfying assignment to the propositional matrix ϕ.

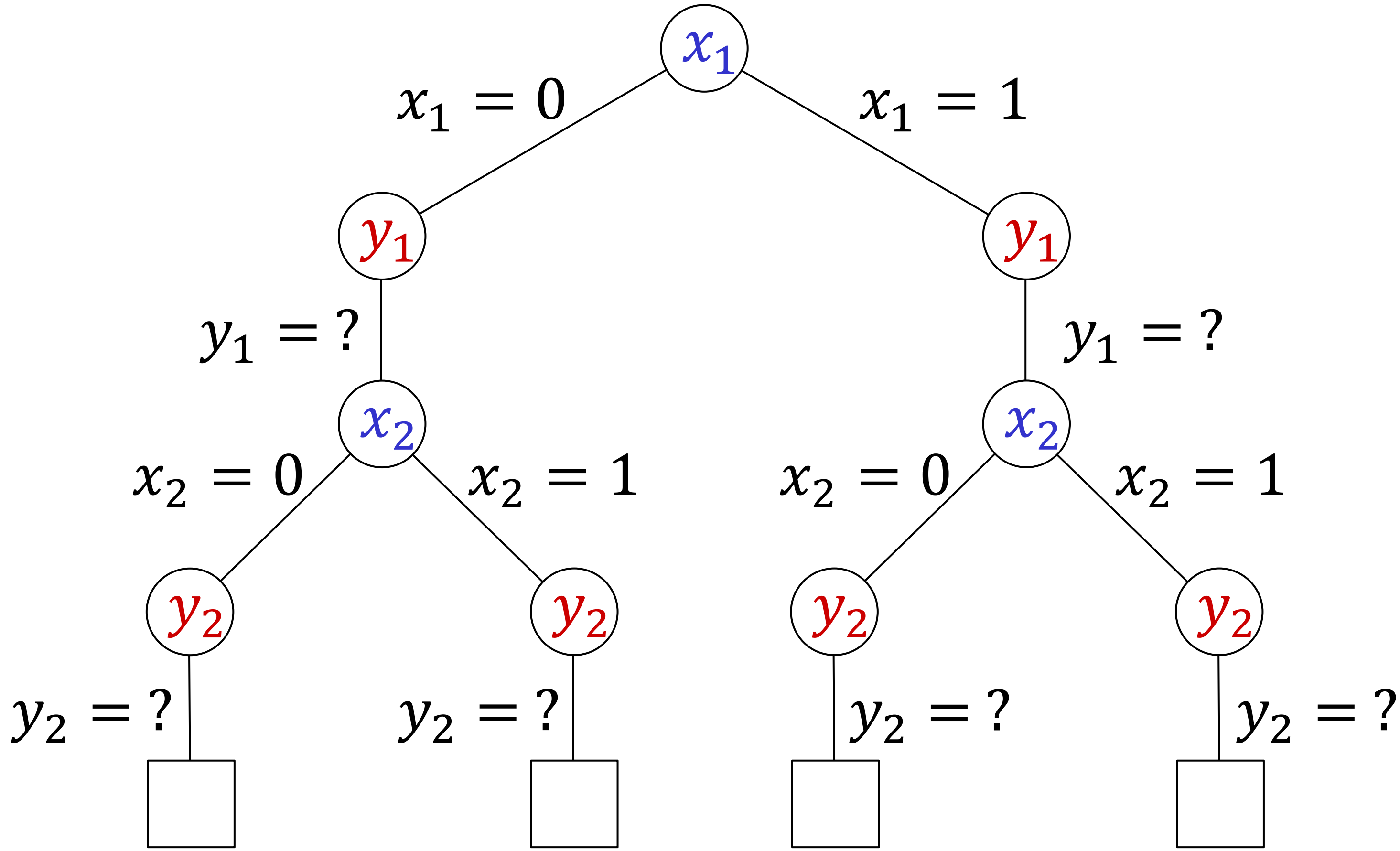
Tree Models: Definition

A closed prenex QBF $Q_1 v_1 \dots Q_k v_k\ \phi$ is true if and only if there exists a tree such that: [Samulowitz et. al., 2006]

1. Each inner node is labeled with a variable v_i, $1 \le i \le k$, its outgoing edges are labeled with $v_i = 0$ or $v_i = 1$.
2. Inner nodes labeled with v_i have two children if and only if $Q_i = \forall$, and one child otherwise.
3. For each path from root to leaf labeled with $(v_{i_1}, \dots, v_{i_j})$, we have $1 \le i_1 < \cdots < i_j \le k$ (i.e. order as in the prefix).
4. On each path from root to leaf, the edge labels are a satisfying assignment to the propositional matrix ϕ.

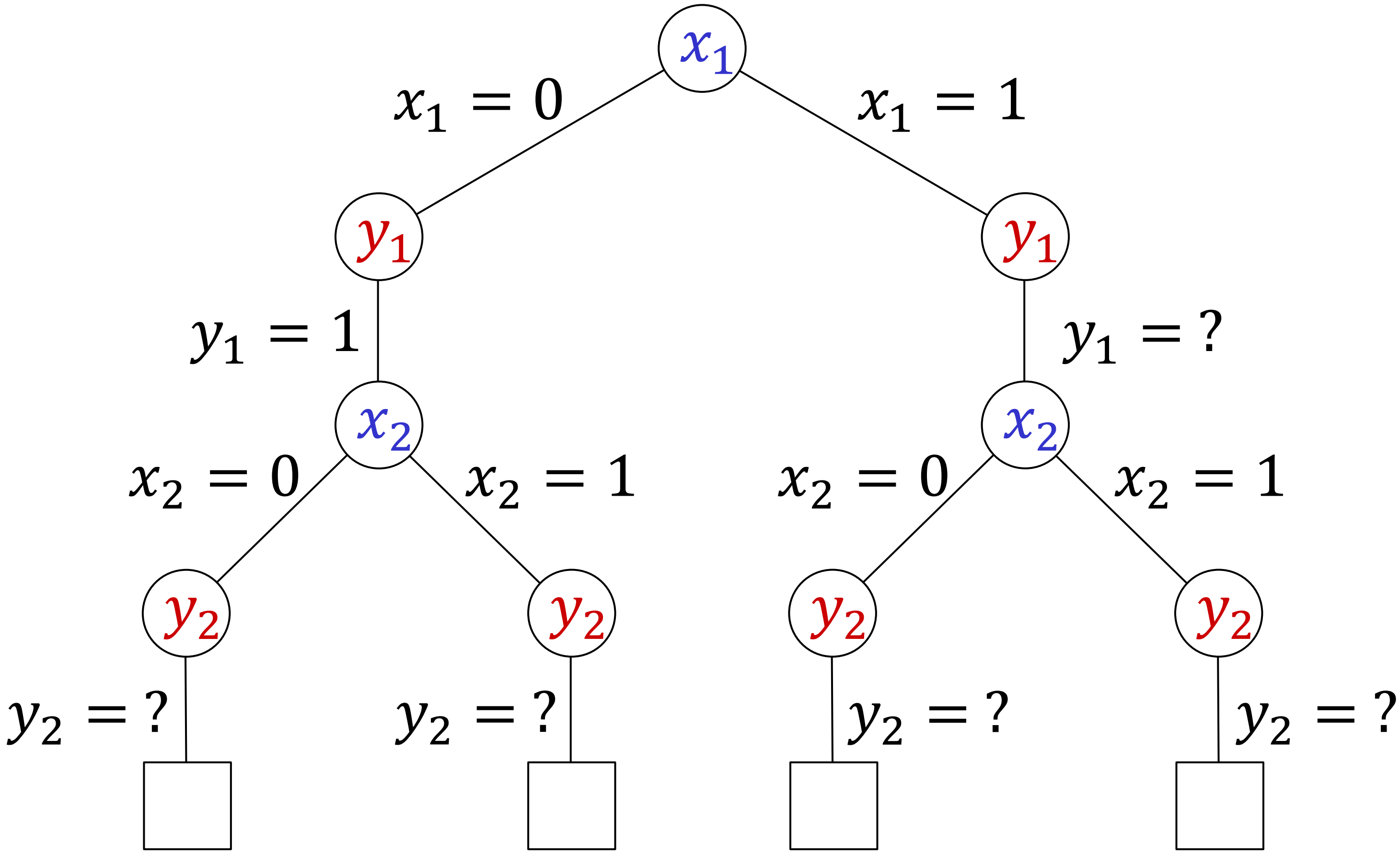
Tree Models: Example

Example: $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \; (x_1 \vee \neg y_2) \wedge (\neg x_1 \vee y_2) \wedge (y_1 \vee x_2)$



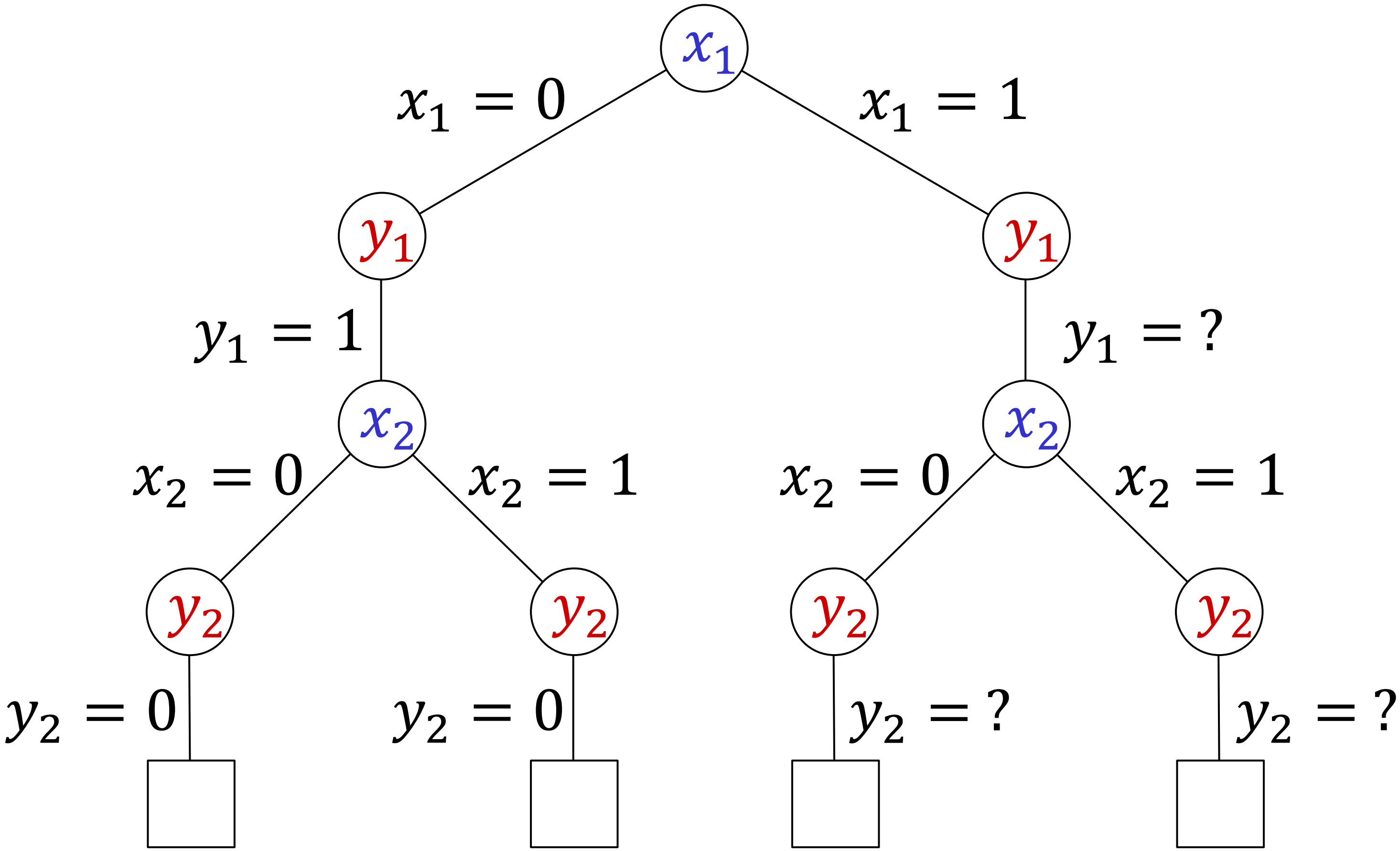
Tree Models: Example

Example: $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \, (x_1 \vee \neg y_2) \wedge (\neg x_1 \vee y_2) \wedge (y_1 \vee x_2)$



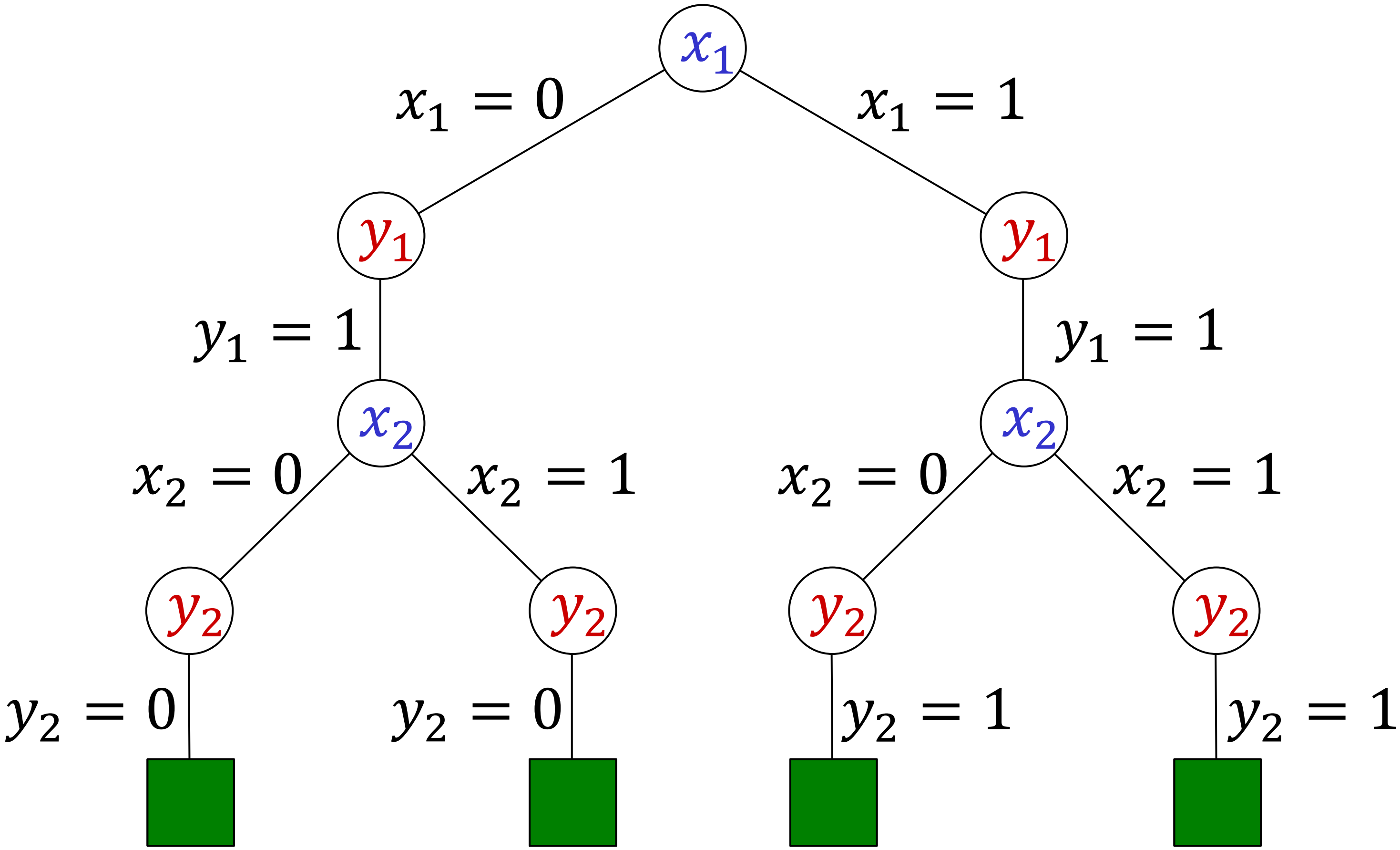
Tree Models: Example

Example: $\forall x_1 \exists y_1 \forall x_2 \exists y_2\ (x_1 \vee \neg y_2) \wedge (\neg x_1 \vee y_2) \wedge (y_1 \vee x_2)$



Tree Models: Example

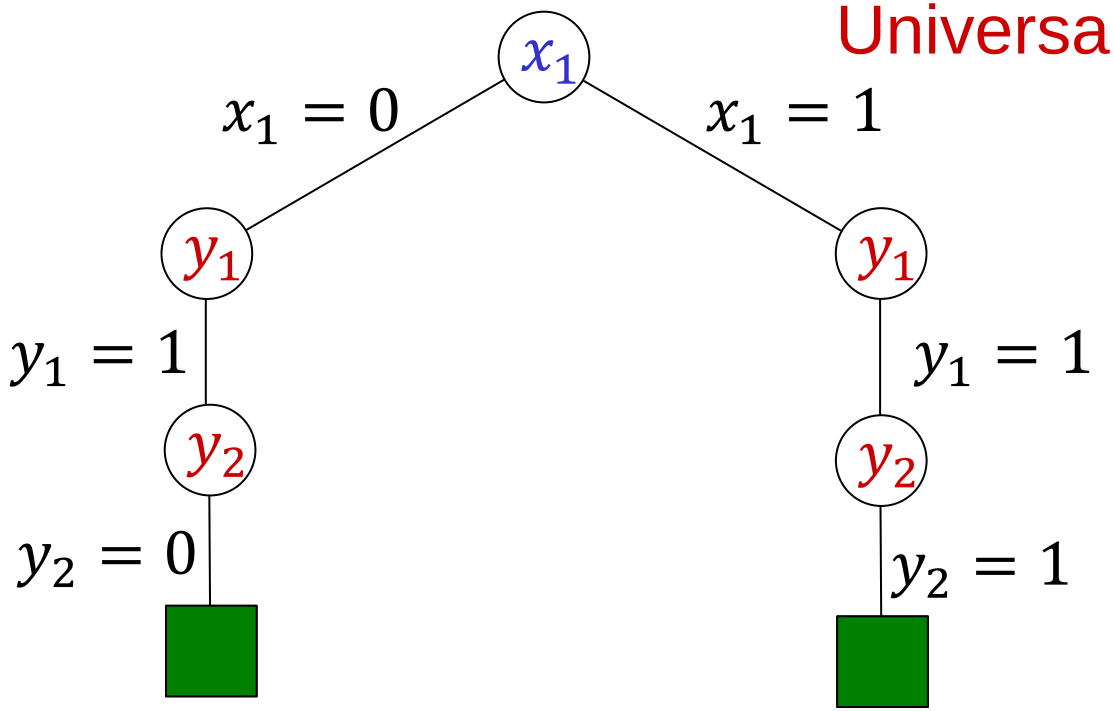
Example: $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \; (x_1 \vee \neg y_2) \wedge (\neg x_1 \vee y_2) \wedge (y_1 \vee x_2)$



Tree Models: Example

Example: $\forall x_1 \exists y_1 \cancel{\forall x_2} \exists y_2 \; (x_1 \lor \neg y_2) \land (\neg x_1 \lor y_2) \land (y_1 \lor \cancel{x_2})$

Universal Reduction



Tree Models: Generalization

The requirement that variables must appear in the same order as in the prefix (rule 3) can be further relaxed:

- by grouping similar quantifiers into quantifier blocks.

$$\underbrace{\forall v_1 \dots \forall v_{n_1}}_{S_1} \underbrace{\exists v_{n_1+1} \dots \exists v_{n_2}}_{S_2} \underbrace{\forall v_{n_2+1} \dots \forall v_{n_3}}_{S_3} \dots \phi$$

- by considering variable dependency schemes.

Dependency Schemes 1/2

Consider the following formula:

$$\forall x \exists y_1 \forall u \exists y_2 \exists y_3 \, [(x \vee u \vee y_3) \wedge (y_1 \vee y_3) \wedge (\neg y_1 \vee \neg y_3) \wedge (\neg u \vee \neg y_3) \wedge (\neg y_1 \vee \neg y_2)]$$

Which variables must be above y_2 in the tree without altering the truth value of the formula?

In general:

dependency decision problem is PSPACE-complete.

[Samer / Szeider, 2007]

Dependency Schemes 2/2

$$\forall x \exists y_1 \forall u \exists y_2 \exists y_3 \, [(x \vee u \vee y_3) \wedge (y_1 \vee y_3) \wedge (\neg y_1 \vee \neg y_3) \wedge (\neg u \vee \neg y_3) \wedge (\neg y_1 \vee \neg y_2)]$$

One simple heuristic: recovering non-prenex structure:

$$\forall x \exists y_1 [[\forall u \exists y_3 (x \vee u \vee y_3) \wedge (y_1 \vee y_3) \wedge (\neg y_1 \vee \neg y_3) \wedge (\neg u \vee \neg y_3)] \wedge [\exists y_2 (\neg y_1 \vee \neg y_2)]]$$

- Identified informally by [Biere, 2004],
- Formalized in [Samer / Szeider, 2007], [Bubeck / Kleine Büning, 2007],
- Currently tightest scheme by [van Gelder, 2011].

Still no practicable algorithm for computing tight schemes!

Free Variables and Equivalence

Semantics of Free Variables 1/2

A focus of this talk is to allow free variables.

Notation:

- $\Phi(z_1, \ldots, z_n)$ for formula Φ with free variables $z_1, \ldots, z_n$.
- QBF* is the class of quantified Boolean formulas with free variables.

The valuation of a QBF* depends on the values of the free variables:

$\Phi(z_1, \ldots, z_n) \in \mathrm{QBF}^*$ is satisfied by a truth assignment $t: \{z_1, \ldots, z_n\} \to \{0,1\}$ if and only if $\Phi(t(z_1), \ldots, t(z_n)) \coloneqq \Phi[z_1/t(z_1), \ldots, z_n/t(z_n)]$ is true.

Semantics of Free Variables 2/2

Example:

$\Phi(a, b, c, d) := \exists y\ (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee c \vee d)$

Then

$\Phi(0,1,0,1) = \exists y\ (0 \vee y) \wedge (1 \vee y) \wedge (\neg y \vee 0 \vee 1)$ is true,

$\Phi(0,1,0,0) = \exists y\ (0 \vee y) \wedge (1 \vee y) \wedge (\neg y \vee 0 \vee 0)$ is false,

etc.

$\rightarrow$ $\exists y\ (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee c \vee d)$ is true if and only if $(a \vee c \vee d) \wedge (b \vee c \vee d)$ is true.

$\rightarrow$ Every QBF* is equivalent to a propositional formula.

Logical Equivalence

Propositional or quantified Boolean formulas α and β with (free) variables $z_1, \dots, z_n$ are logically equivalent, written as $\alpha(z_1, \dots, z_n) \approx \beta(z_1, \dots, z_n)$, if and only if $\alpha(t(z_1), \dots, t(z_n)) = \beta(t(z_1), \dots, t(z_n))$ for each truth assignment t to $z_1, \dots, z_n$.

$\rightarrow$ Quantified variables are not directly considered for logical equivalence. They can be seen as local within the respective formula.

Satisfiability Equivalence

$(a \vee \underline{c \vee d}) \wedge (b \vee \underline{c \vee d}) \approx \exists y\ (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee \underline{c \vee d})$

Without quantifier:

$(a \vee c \vee d) \wedge (b \vee c \vee d) \not\approx (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee c \vee d)$

Problem:

$a = b = 1, c = d = 0, y = 1$ satisfies left, but not right side.

Relaxation: satisfiability equivalence

$(a \vee c \vee d) \wedge (b \vee c \vee d) \approx_{SAT} (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee c \vee d)$

Existence of a satisfying assignment for one side implies there is <u>some</u> satisfying assignment for the other side.

Equivalence and Rewriting

For propositional formulas α, β, γ, it holds that

$\beta \approx \gamma \;\Rightarrow\; \alpha \approx \alpha[\beta/\gamma]$.

→ Parts of a formula can be replaced with logically equivalent formulas.

But: $\beta \approx_{SAT} \gamma \;\not\Rightarrow\; \alpha \approx_{SAT} \alpha[\beta/\gamma]$.

→ Satisfiability equivalence is sometimes too weak.

Complexity

QBF* Complexity 1/2

Well known: the decision problem for QBF and the satisfiability problem for QBF* are PSPACE-complete.

[Meyer / Stockmeyer, 1973]

→ QBF* consequence and equivalence problems are also PSPACE-complete.

→ QDNF* remains PSPACE-complete.

Some verification problems are also PSPACE-complete:

- Propositional LTL satisfiability [Sistla / Clarke, 1985]
- Symbolic reachability in sequential circuits [Savitch, 1970]

QBF* Complexity 2/2

Typical approaches to reduce the complexity:

- Restriction of the matrix to special classes of propositional formulas

- Bounding of quantifier alternations in the prefix:

$$\underbrace{\exists \ldots \exists \forall \ldots \forall \exists \ldots \exists \ldots}_{} \ \phi \qquad \underbrace{\forall \ldots \forall \exists \ldots \exists \forall \ldots \forall \ldots}_{} \ \phi$$

k quantifier blocks, outermost existential:

k quantifier blocks, outermost universal:

prefix type Σ_k

prefix type Π_k

Polynomial-time Hierarchy

Well-known relationship with polynomial-time hierarchy:

For fixed $k \geq 1$, the satisfiability problem for QBF* with prefix type Σ_k is Σ_k^P-complete, and Π_k^P-complete for prefix type Π_k.

[Stockmeyer, 1976]

$$\Delta_0^P := \Sigma_0^P := \Pi_0^P := P$$

$$\Sigma_{k+1}^P := NP^{\Sigma_k^P}, \qquad \Pi_{k+1}^P := co - \Sigma_{k+1}^P, \qquad \Delta_{k+1}^P := P^{\Sigma_k^P}$$

Schaefer‘s Dichotomy Theorem

Consider a quantified constraint expression

$$Q_1 y_1 \ldots Q_n y_n \;\; f_1\left(x_{1,1}, \ldots, x_{1,m_1}\right) \wedge \cdots \wedge f_k\left(x_{k,1}, \ldots, x_{k,m_k}\right)$$

with $Q_1, \ldots, Q_n \in \{\forall, \exists\}$, Boolean functions $f_1, \ldots, f_k \in C$ and arguments $x_{i,j} \in \{y_1, \ldots, y_n\} \cup \{0,1\}$.

Dichotomy Theorem:

Let C be a finite set of constraints. If C is *Horn*, *anti-Horn*, *bijunctive / Krom* (equiv. to 2-CNF) or *affine* (equiv. to XOR-CNF) then $QSAT_c(C)$ is in P.

Otherwise, $QSAT_c(C)$ is PSPACE-complete.

[Schaefer, 1978], [Dalmau, 1997] [Creignou / Khanna et al., 2001]

QHORN* and Q2-CNF*

- QHORN* is the class of QCNF* formulas with at most one positive literal per clause.

  QHORN* satisfiability is decidable in quadratic time $O(|\forall| \cdot |\Phi|)$, where $|\forall|$ is the number of universal variables and $|\Phi|$ the formula length.

  Algorithm: by unit propagation [Flögel et al. 1995]
  or by universal expansion [Bubeck / Kleine Büning, 2008]

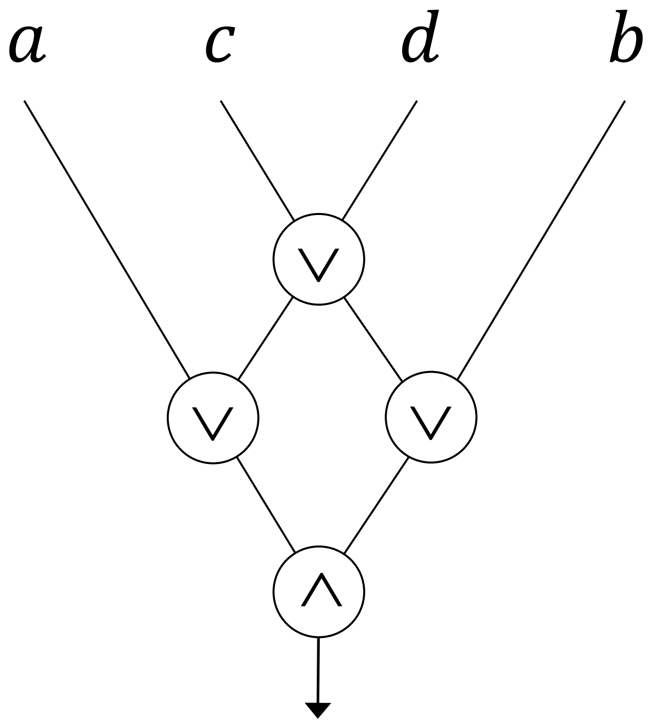
- Q2-CNF* satisfiability is decidable in linear time.

  Algorithm: by strongly connected components [Aspvall et al., 1979]

Expressiveness

Other Representations

We can also define logical equivalence with other representations of Boolean functions:



$\approx (a \vee c \vee d) \wedge (b \vee c \vee d)$

$\approx \exists y\, (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee c \vee d)$

For each assignment to a, b, c, d, all three representations evaluate to the same truth value.

Quantified Encodings 1/2

Main Theme:

How compact are encodings in QBF* (or subclasses) versus other logically equivalent representations?

Encoding Techniques:

1. Abbreviate exact repetitions by existential variables:

$$(\underline{A \lor \neg B \lor C} \lor D) \land (\underline{A \lor \neg B \lor C} \lor \neg E) \land (\underline{A \lor \neg B \lor C} \lor F)$$

$$\approx \exists y \left(y \leftrightarrow (\underline{A \lor \neg B \lor C})\right) \land (y \lor D) \land (y \lor \neg E) \lor (y \lor F)$$

Simple implication if term occurs only in one polarity:

$$\exists y \left(y \rightarrow (A \lor \neg B \lor C)\right) \land (y \lor D) \land (y \lor \neg E) \lor (y \lor F)$$

(quantified versions of [Tseitin, 1970] and [Plaisted, 1986])

Quantified Encodings 2/2

2. Compress conjunctions of renamings / instantiations

   by universal variables:

   $\phi(A_1, B_1) \wedge \phi(A_2, B_2) \wedge \phi(A_3, B_3)$

   $\approx \forall u \forall v \left(\bigvee_{i=1\ldots3} \left((u \leftrightarrow A_i) \wedge (v \leftrightarrow B_i) \right) \right) \rightarrow \phi(u, v)$

   [Dershowitz et al., 2005], [Meyer/Stockmeyer, 1973]

3. Iterative Squaring (Extension of 2.):

   $\Phi(x_0, x_n) = \exists x_1 \ldots \exists x_{n-1}\ \phi(x_0, x_1) \wedge \phi(x_1, x_2) \wedge \cdots \wedge \phi(x_{n-1}, x_n)$

   Encoding:

   $\Phi_n(x_0, x_n) \coloneqq \exists y \left(\Phi_{n/2}(x_0, y) \wedge \Phi_{n/2}(y, x_n) \right)$

   $\approx \exists y \forall u \forall v \left(\left((u \leftrightarrow x_0) \wedge (v \leftrightarrow y) \right) \vee \left((u \leftrightarrow y) \wedge (v \leftrightarrow x_n) \right) \right) \rightarrow \Phi_{n/2}(u, v)$

   [Meyer/Stockmeyer, 1973]

Nested Instantiations 1/3

Instantiations might also be nested:

$$\big((A_1 \wedge A_2) \vee (\neg A_1 \wedge \neg A_2)\big) \rightarrow \big((B_1 \wedge B_2) \vee (\neg B_1 \wedge \neg B_2)\big)$$

is of the form $\psi(\phi(A_1, A_2), \phi(B_1, B_2))$.

Other example: Parity of n Boolean variables

$$f_0(p_1, p_2) \coloneqq (\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)$$

$$f_1(p_1, p_2, p_3, p_4) \coloneqq f_0(f_0(p_1, p_2), f_0(p_3, p_4))$$

$$f_2(p_1, \ldots, p_{16}) \coloneqq f_1(f_1(p_1, \ldots, p_4), \ldots, f_1(p_{13}, \ldots, p_{16}))$$

$\log_2 \log_2 n + 1$ definitions of size $O(n)$.

Nested Instantiations 2/3

General Definition: A Nested Boolean Function (NBF) is a sequence of functions $\mathrm{D}(f_k) = (f_0, \dots, f_k)$ with

- initial functions $f_0, \dots, f_t$ defined by $f_i(\boldsymbol{x}^i) \coloneqq \alpha_i(\boldsymbol{x}^i)$ for a propositional formula α_i over $\boldsymbol{x}^i \coloneqq (x^{i,1}, \dots, x^{i,n_i})$
- compound functions $f_{t+1}, \dots, f_k$ of the form $f_i(\boldsymbol{x}^i) \coloneqq f_{j_0}(f_{j_1}(\boldsymbol{x}_1^i), \dots, f_{j_r}(\boldsymbol{x}_r^i))$ with previously defined functions $f_{j_0}, \dots, f_{j_r}$ and matching tuples $\boldsymbol{x}_1^i, \dots, \boldsymbol{x}_r^i$ over variables from $\boldsymbol{x}^i$ or Boolean constants.

[Bubeck / Kleine Büning, 2012], [Cook / Soltys, 1999]

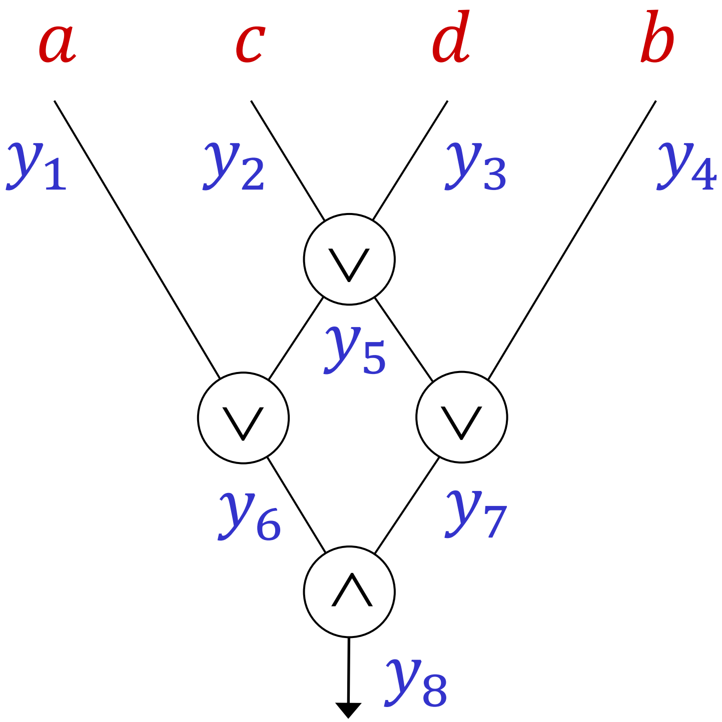
Nested Instantiations 3/3

- By clever combination of previously presented encoding techniques, every NBF can be transformed in linear time into a logically equivalent prenex QBF*.
  [Bubeck / Kleine Büning, 2012]
- The inverse direction is very simple by simulating quantifier expansion, but length increases slightly to $O(|v| \cdot |\Phi|)$ for a QBF* Φ with $|v|$ quantified variables.
- **Application:** Configuration Problems (→ Talk by Hans)
- **Future Work:** Solvers, interesting NBF subclasses

Circuits and Existential Quantifiers

There is a close connection between fan-out in Boolean circuits and existential quantification:

1. Transformation from circuit to formula



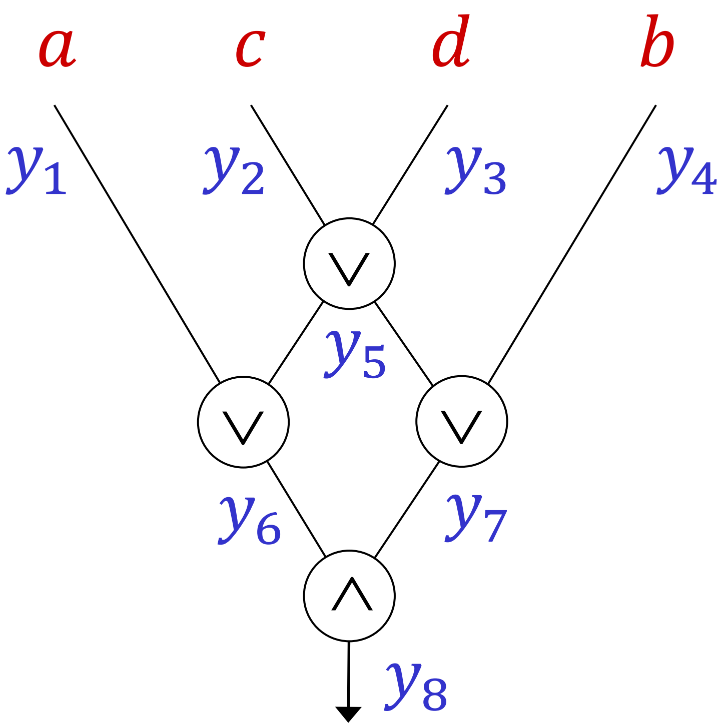
$$
\begin{aligned}
&\exists y_1 \ldots \exists y_8 \\
&(a \vee y_1) \wedge (c \vee y_2) \wedge \\
&(d \vee y_3) \wedge (b \vee y_4) \wedge \\
&(y_2 \wedge y_3 \rightarrow y_5) \wedge \\
&(y_1 \wedge y_5 \rightarrow y_6) \wedge \\
&(y_5 \wedge y_4 \rightarrow y_7) \wedge \\
&(y_6 \vee y_7 \rightarrow y_8) \wedge \\
&\neg y_8
\end{aligned}
$$

[Bauer / Brand et al., 1973] [Anderaa / Börger, 1981]

Circuits and Existential Quantifiers

There is a close connection between fan-out in Boolean circuits and existential quantification:

1. Transformation from circuit to formula



$$\exists y_1 \dots \exists y_8$$
$$(a \vee y_1) \wedge (c \vee y_2) \wedge$$
$$(d \vee y_3) \wedge (b \vee y_4) \wedge$$
$$(\neg y_2 \vee \neg y_3 \vee y_5) \wedge$$
$$(\neg y_1 \vee \neg y_5 \vee y_6) \wedge$$
$$(\neg y_5 \vee \neg y_4 \vee y_7) \wedge$$
$$(\neg y_6 \vee y_8) \wedge (\neg y_7 \vee y_8) \wedge$$
$$\neg y_8$$
$$\approx \exists y \, (a \vee y) \wedge (b \vee y) \wedge (\neg y \vee c \vee d)$$

The Class $\exists$HORN^b

- The previous linear transformation produces existentially quantified formulas in CNF with at most one positive bound variable per clause (i.e. the bound variables respect the Horn property).
  We call such formulas $\exists$HORN^b.

- **General:** for QK $\subseteq$ QCNF, QK^b is the class of formulas $\Phi(z_1, \ldots, z_n) = Q_1 v_1 \ldots Q_m v_m \wedge_i \left(\phi_i^b(v_1, \ldots, v_m) \vee \phi_i^f(z_1, \ldots, z_n)\right)$ where $Q_1 v_1 \ldots Q_m v_m \ \wedge_i \phi_i^b(v_1, \ldots, v_m)$ is a formula in QK, and $\phi_i^f(z_1, \ldots, z_n)$ an arbitrary clause over free variables.

Circuits and Existential Quantifiers

2. Transformation from formula to circuit

Any $\exists\text{HORN}^b$ formula can be transformed in polynomial time into a logically equivalent Boolean circuit.

[Anderaa / Börger, 1981] [Kleine Büning / Zhao / Bubeck, 2009]

$\rightarrow$ $\exists\text{HORN}^b$ and circuits have similar expressiveness.

$\exists$HORN^b vs Propositional Formulas

There are $\exists$HORN^b (even $\exists$HORN*) formulas for which there is no logically equivalent propositional CNF of polynomial length.

[Kleine Büning / Lettmann, 1999]

Is a polynomial-length transformation from $\exists$HORN^b to arbitrary propositional formulas possible?

Equivalent: Are circuits with unrestricted fan-out more expressive than with fan-out 1?

$\exists\mathrm{HORN}^b$ vs QHORN^b

Can the expressive power of $\exists\mathrm{HORN}^b$ be enhanced by universal quantification?

Not significantly. For every $\Phi \in \mathrm{QHORN}^b$ with $|\forall|$ universal quantifiers, there exists a logically equivalent $\exists\mathrm{HORN}^b$ formula of quadratic length $O(|\forall| \cdot |\Phi|)$ which can be computed also in time $O(|\forall| \cdot |\Phi|)$.
[Bubeck, 2010]

That means QHORN^b satisfiability is NP-complete.

Minimal Falsity and Quantification

A closed QCNF formula is minimal false (MF) iff it is false and removing an arbitrary clause makes it true.

Example:

$\forall x \exists y_0 \exists y_1\ (y_0 \vee z_0) \wedge (\neg y_0 \vee z_1) \wedge (\neg y_0 \vee z_2) \wedge (y_1 \vee z_3) \wedge (x \vee z_4)$

Bound parts: $\forall x \exists y_0 \exists y_1\ (y_0) \wedge (\neg y_0) \wedge (\neg y_0) \wedge (y_1) \wedge (x)$

MF subformulas:

1. $\exists y_0\ (y_0) \wedge (\neg y_0)$
2. $\exists y_0\ (y_0) \wedge (\neg y_o)$
3. $\forall x\ (x)$

Minimal Falsity and Quantification

$\forall x \exists y_0 \exists y_1\ (y_0 \vee z_0) \wedge (\neg y_0 \vee z_1) \wedge (\neg y_0 \vee z_2) \wedge (y_1 \vee z_3) \wedge (x \vee z_4)$

Bound parts: $\forall x \exists y_0 \exists y_1\ (y_0) \wedge (\neg y_0) \wedge (\neg y_0) \wedge (y_1) \wedge (x)$

MF subformulas:

1. $\exists y_0\ (y_0) \wedge (\neg y_0)$
2. $\exists y_0\ (y_0) \wedge (\neg y_o)$
3. $\forall x\ (x)$

Corresponding bound parts:

1. z_0, z_1
2. z_0, z_2
3. z_4

For each MF subformula, one of the corresponding free parts must be satisfied.

So the formula is equivalent to $(z_0 \vee z_1) \wedge (z_0 \vee z_2) \wedge z_4$.

Minimal Falsity and Quantification

MF subformulas of the bound parts determine the role of the free parts in QCNF* formulas:

$$Q \bigwedge_{1 \le i \le q} \left(\phi_i^b \vee \phi_i^f\right) \approx \underbrace{\bigwedge_{\left(Q\ \phi_{i_1}^b \wedge \cdots \wedge \phi_{i_r}^b\right) \in MF}}_{\text{all MF subformulas of the bound parts}} \left(\phi_{i_1}^f \vee \cdots \vee \phi_{i_r}^f\right)$$

[Bubeck / Kleine Büning, 2010]

How do restrictions on the structure of the MF skeleton influence the expressiveness?

Conclusion

Conclusion

- Free variables are nothing to be afraid of. ☺
- Compared to propositional logic, there is a very rich set of QBF* subclasses.
- How do these subclasses differ in expressiveness?

  Is there a hierarchy
  PROP $<_{poly-len}$ ∃HORN^b $<_{poly-len}$ ∃CNF* $<_{poly-len}$ ∀∃CNF* ... ?
- How are these classes related to other representations, e.g. circuits?

References 1/6

- S. Anderaa and E. Börger.
  The Equivalence of Horn and Network Complexity for Boolean Functions. 1981
- B. Aspvall, M. Plass and R. Tarjan.
  A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas. 1979
- M. Bauer, D. Brand, M. Fischer, A. Meyer and M. Paterson.
  A Note on Disjunctive Form Tautologies. 1973
- A. Biere.
  Resolve and Expand. 2004
- U. Bubeck.
  Model-Based Transformations for Quantified Boolean Formulas. 2010

References 2/6

- U. Bubeck and H. Kleine Büning.
  Bounded Universal Expansion for Preprocessing QBF. 2007
- U. Bubeck and H. Kleine Büning.
  Models and Quantifier Elimination for Quantified Horn Formulas. 2008
- U. Bubeck and H. Kleine Büning.
  The Power of Auxiliary Variables for Propositional and Quantified Boolean Formulas. 2010
- U. Bubeck and H. Kleine Büning.
  Encoding Nested Boolean Functions as Quantified Boolean Formulas. 2012
- S. Cook and M. Soltys.
  Boolean Programs and Quantified Propositional Proof Systems. 1999

References 3/6

- N. Creignou, S. Khanna and M. Sudan.
  Complexity Classifications of Boolean Constraint Satisfaction Problems. 2001
- V. Dalmau
  Some Dichotomy Theorems on Constant-free Boolean Formulas. 1997
- N. Dershowitz, Z. Hanna and J. Katz.
  Bounded Model Checking with QBF. 2005
- U. Egly, M. Seidl, H. Tompits, S. Woltran and M. Zolda.
  Comparing Different Prenexing Strategies for Quantified Boolean Formulas. 2004
- A. Flögel, M. Karpinski and H. Kleine Büning.
  Resolution for Quantified Boolean Formulas. 1995

References 4/6

- H. Kleine Büning and U. Bubeck.
  Theory of Quantified Boolean Formulas. 2009
- H. Kleine Büning and T. Lettmann.
  Propositional Logic: Deduction and Algorithms. 1999
- H. Kleine Büning, X. Zhao and U. Bubeck.
  Resolution and Expressiveness of Subclasses of Quantified Boolean Formulas and Circuits. 2009
- A. Meyer and L. Stockmeyer.
  Word Problems Requiring Exponential Time. 1973
- D. Plaisted and S. Greenbaum.
  A Structure-preserving Clause Form Translation. 1986

References 5/6

- M. Samer and S. Szeider.
  Backdoor Sets of Quantified Boolean Formulas. 2007
- H. Samulowitz, J. Davies, F. Bacchus.
  Preprocessing QBF. 2006
- W. Savitch.
  Relationships Between Nondeterministic and Deterministic Tape Complexities. 1970
- T. Schaefer.
  The Complexity of Satisfiability Problems. 1978
- A. Sistla and E. Clarke.
  The Complexity of Propositional Linear Time Logics. 1985

References 6/6

- L. Stockmeyer.

  The Polynomial-Time Hierarchy. 1976

- G. Tseitin.

  On the Complexity of Derivation in Propositional Calculus. 1970

- A. van Gelder.

  Variable Independence and Resolution Paths. 2011