Autarkies for DQCNF

Oliver Kullmann [1] **Ankit Shukla** [2]

[1] Swansea University [2] JKU, Linz

FMCAD 2019, San Jose, California, USA

Der Wissenschaftsfonds.

LOGICAL METHODS IN COMPUTER SCIENCE

DQCNF: Dependency Quantified Boolean CNF

Formula

$F =$

Quantifier Prefix

$\forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y)\, :$

Matrix

F_0

Matrix

$F_0 =$

Propositional formula in CNF

$$(a \vee b \vee x) \wedge (\bar{a} \vee \bar{b} \vee \bar{x}) \wedge (\bar{a} \vee b \vee y) \wedge (a \vee \bar{b} \vee \bar{y}) \wedge$$
$$(c \vee x \vee y) \wedge (c \vee \bar{x} \vee \bar{y} \vee a) \wedge (\bar{c} \vee x \vee \bar{y}) \wedge (\bar{c} \vee \bar{x} \vee y) \wedge$$
$$(d \vee e \vee x) \wedge (\bar{d} \vee \bar{e} \vee \bar{x} \vee c) \wedge (\bar{d} \vee e \vee y) \wedge (d \vee \bar{e} \vee \bar{y})$$

Dependency set

$D_a = \{x\},\ D_b = \{y\},\ D_c = \{x, y\},\ D_d = \{x\},\ D_e = \{y\}$

Solve DQCNF: SAT or UNSAT?

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y)$:
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Solve DQCNF: SAT or UNSAT?

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y)$:
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

DQCNF Satisfiability

F is satisfiable, if there exists a total assignment: map each existential variable v to a boolean function over the dependency-set of v,
$f = \{fa_x,\ fb_y,\ fc_{x,y},\ fd_x,\ fe_y\}$
such that the matrix after substitution and simplification becomes satisfiable.

Solve DQCNF: SAT or UNSAT?

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

DQCNF Satisfiability

F is satisfiable, if there exists a total assignment: map each existential variable v to a boolean function over the dependency-set of v,
$f = \{fa_x,\ fb_y,\ fc_{x,y},\ fd_x,\ fe_y\}$
such that the matrix after substitution and simplification becomes satisfiable.

Solve DQCNF: SAT or UNSAT?

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, ⓒ\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

DQCNF Satisfiability

F is satisfiable, if there exists a total assignment: map each existential variable v to a boolean function over the dependency-set of v,
$f = \{fa_x,\ fb_y,\ fc_{x,y},\ fd_x,\ fe_y\}$
such that the matrix after substitution and simplification becomes satisfiable.

Rules of the game

Given a DQCNF formula F:

1. Pick existential variables, assign it a boolean function of universal variables and substitute it in the matrix.

Rules of the game

Given a DQCNF formula F:

1. Pick existential variables, assign it a boolean function of universal variables and substitute it in the matrix.
2. The corresponding clauses becomes tautology.

Rules of the game

Given a DQCNF formula F:

1. Pick existential variables, assign it a boolean function of universal variables and substitute it in the matrix.
2. The corresponding clauses becomes tautology.

Ques. What are allowed values?

Rules of the game

Given a DQCNF formula F:

1. Pick existential variables, assign it a boolean function of universal variables and substitute it in the matrix.
2. The corresponding clauses becomes tautology.

Ques. What are allowed values?
Ans. Any boolean function based on the **dependency set**.

Rules of the game

Given a DQCNF formula F:

1. Pick existential variables, assign it a boolean function of universal variables and substitute it in the matrix.
2. The corresponding clauses becomes tautology.

Ques. What are allowed values?
Ans. Any boolean function based on the **dependency set**.

Values of boolean functions

$\forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y)$
$a \mapsto 0, 1, x, \neg x$
$b \mapsto 0, 1, y, \neg y$
$c \mapsto 0, 1, x, \neg x, y, \neg y, x \vee y, x \wedge y, \ldots$

Solve DQCNF: piece by piece

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y):$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Solve the DQCNF

Choices a, b, c, d, e

Solve DQCNF: piece by piece

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y):$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Solve the DQCNF

Choices a, b, c, d, e

Solve DQCNF: piece by piece

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Solve the DQCNF

- Pick d, e

Solve DQCNF: piece by piece

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Solve the DQCNF

- Pick d, e
  $d \mapsto \neg x$, $e \mapsto \neg y$

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{\bar{x}, \bar{y}, x\}, \{x, y, \bar{x}, c\}, \{x, \bar{y}, y\}, \{\bar{x}, y, \bar{y}\}$

Solve the DQCNF

- Pick d, e
  $d \mapsto \neg x$, $e \mapsto \neg y$

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\cancel{\{\bar{x}, e, x\}}, \cancel{\{x, y, \bar{x}, c\}}, \cancel{\{x, \bar{y}, y\}}, \cancel{\{\bar{x}, y, \bar{y}\}}$

Solve the DQCNF

- Pick d, e
  $d \mapsto \neg x$, $e \mapsto \neg y$

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Solve the DQCNF

- Pick d, e
  $d \mapsto \neg x$, $e \mapsto \neg y$

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Solve the DQCNF

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Solve the DQCNF

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Rule 2. Corresponding clauses becomes tautology!

Solve the DQCNF

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Solve the DQCNF

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Solve the DQCNF

- Pick c

Solve DQCNF: piece by piece

DQCNF formula

$F_1 := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Solve the DQCNF

- Pick c

  $c = x \leftrightarrow y \qquad [(x \vee \neg y) \wedge (\neg x \vee y)]$

Solve DQCNF: piece by piece

DQCNF formula

$F_2 := \forall x, y\, \exists a(x)\, \exists b(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{x \leftrightarrow y, x, y\}, \{x \leftrightarrow y, \bar{x}, \bar{y}, a\}, \{\overline{x \leftrightarrow y}, x, \bar{y}\}, \{\overline{x \leftrightarrow y}, \bar{x}, y\}$

Solve the DQCNF

- Pick c

  $c = x \leftrightarrow y \qquad [(x \vee \neg y) \wedge (\neg x \vee y)]$

Solve DQCNF: piece by piece

DQCNF formula

$F_2 := \forall x, y\, \exists a(x)\, \exists b(y)$:
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
~~$\{x \leftrightarrow y, x, y\}$~~, ~~$\{x \leftrightarrow y, \bar{x}, \bar{y}, a\}$~~, ~~$\{\overline{x \leftrightarrow y}, x, \bar{y}\}$~~, ~~$\{\overline{x \leftrightarrow y}, \bar{x}, y\}$~~

Solve the DQCNF

- Pick c

  $c = x \leftrightarrow y$ $\quad [(x \vee \neg y) \wedge (\neg x \vee y)]$

Solve DQCNF: piece by piece

DQCNF formula

$F_2 := \forall x, y \, \exists a(x) \, \exists b(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\}$

Solve the DQCNF

- Pick c
  $c = x \leftrightarrow y$ $\quad [(x \vee \neg y) \wedge (\neg x \vee y)]$

Solve DQCNF piece by piece

DQCNF formula

$F_2 := \forall x, y \, \exists a(x) \, \exists b(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\}$

Solve the DQCNF

Solve DQCNF piece by piece

DQCNF formula

$F_2 := \forall x, y\, \exists a(x)\, \exists b(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\}$

Solve the DQCNF

Solve DQCNF piece by piece

DQCNF formula

$F_2 := \forall x, y \, \exists a(x) \, \exists b(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\}$

Solve the DQCNF

- Pick a, b

Solve DQCNF piece by piece

DQCNF formula

$F_2 := \forall x, y \, \exists a(x) \, \exists b(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\}$

Solve the DQCNF

- Pick a, b
  $a = \neg x, b = \neg y$

Solve DQCNF piece by piece

DQCNF formula

$F_3 := \{\bar{x}, \bar{y}, x\}, \{x, y, \bar{x}\}, \{x, \bar{y}, y\}, \{\bar{x}, y, \bar{y}\}$

Solve the DQCNF

- Pick a, b

  $a = \neg x, b = \neg y$

Solve DQCNF piece by piece

DQCNF formula

$F_3 := \{\bar{x}, \bar{y}, x\}, \{x, y, \bar{x}\}, \{x, \bar{y}, y\}, \{\bar{x}, y, \bar{y}\}$

Solve the DQCNF

- Pick a, b

  $a = \neg x, b = \neg y$

Solve DQCNF piece by piece

DQCNF formula

$F_3 :=$ true

Solve the DQCNF

- Pick a, b

  $a = \neg x, b = \neg y$

DQCNF F is SAT

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

DQCNF F is SAT

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Total satisfying assignment

$a \mapsto \neg x$, $b \mapsto \neg y$, $c \mapsto (x \leftrightarrow y)$, $d \mapsto \neg x$, $e \mapsto \neg y$

Autarkies for DQCNF

Autarkies for SAT [Büning and Kullmann(2009)]

A **partial assignment** $\varphi : \mathrm{var}(F) \mapsto \{0, 1\}$ is an autarky iff

- for every clause $C \in F$ either φ does not "touch" C, i.e., $\mathrm{var}(\varphi) \cap \mathrm{var}(C) = \emptyset$, or
- φ satisfies C i.e. $\varphi * C$ is true.

Autarkies for DQCNF

Autarkies for SAT [Büning and Kullmann(2009)]

A **partial assignment** $\varphi : \mathrm{var}(F) \mapsto \{0, 1\}$ is an autarky iff

- for every clause $C \in F$ either φ does not "touch" C, i.e., $\mathrm{var}(\varphi) \cap \mathrm{var}(C) = \emptyset$, or
- φ satisfies C i.e. $\varphi * C$ is true.

Example

For $F = \{\, \{a\}, \{a, b\}\, \}$;

Autarkies for DQCNF

Autarkies for SAT [Büning and Kullmann(2009)]

A **partial assignment** $\varphi : \text{var}(F) \mapsto \{0,1\}$ is an autarky iff

- for every clause $C \in F$ either φ does not "touch" C, i.e., $\text{var}(\varphi) \cap \text{var}(C) = \emptyset$, or
- φ satisfies C i.e. $\varphi * C$ is true.

Example

For $F = \{\, \{a\}, \{a, b\} \,\}$; the partial assignment $b \mapsto 1$ is an autarky.

Autarkies for DQCNF

Autarkies for SAT [Büning and Kullmann(2009)]

A **partial assignment** $\varphi : \mathrm{var}(F) \mapsto \{0, 1\}$ is an autarky iff

- for every clause $C \in F$ either φ does not "touch" C, i.e., $\mathrm{var}(\varphi) \cap \mathrm{var}(C) = \emptyset$, or
- φ satisfies C i.e. $\varphi * C$ is true.

Example

For $F = \{\, \{a\}, \{a, b\} \,\}$; the partial assignment $b \mapsto 1$ is an autarky.

Autarkies for DQCNF

- φ assign existential variables v of F with **boolean functions** of variables of the dependency-set.
- Making a clause "*true*" now means making it a tautology.

DQCNF Autarky

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

DQCNF Autarky

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Partial assignment,
$d \mapsto \neg x$, $e \mapsto \neg y$ is an autarky.

Extreme cases

1. The empty partial assignment is an autarky for every F (trivial autarky).

2. A satisfying assignment for F is also an autarky for F.

Extreme cases

1. The empty partial assignment is an autarky for every F (trivial autarky).
2. A satisfying assignment for F is also an autarky for F.
3. Elimination of **pure literals** is a special case of an "autarky reduction".

Pure literal

$\forall x, y \exists a(x) b(x, y) : \{x, \bar{y}, \bar{a}\}, \{\bar{x}, a, b\}, \{y, b\}$
Assign $b \mapsto 1$.

The basic lemmas

Lemma (satisfiability-equivalence)

For an autarky φ of F, $\varphi * F$ is satisfiability-equivalent to F.

The basic lemmas

Lemma (satisfiability-equivalence)

For an autarky φ of F, $\varphi * F$ is satisfiability-equivalent to F.

Proof.

If there is a satisfying assignment of F it satisfies also $\varphi * F$, since just clauses have been removed.
If ϕ is a total satisfying assignment for $\varphi * F$, then $\varphi \cup \phi$ is a (partial) satisfying assignment for F. □

The basic lemmas

Lemma (satisfiability-equivalence)

For an autarky φ of F, $\varphi * F$ is satisfiability-equivalent to F.

Lemma (confluence)

Autarky reduction is confluent.

Lemma (composition)

The composition of two autarkies is again an autarky.

The basic lemmas

- A DQCNF F is called **lean** if it has no non-trivial autarkies.

The basic lemmas

- A DQCNF F is called **lean** if it has no non-trivial autarkies.
- **Lean kernel** (unique) is obtained by repeatedly applying autarky-reduction on F as long as possible.

The basic lemmas

- A DQCNF F is called **lean** if it has no non-trivial autarkies.
- **Lean kernel** (unique) is obtained by repeatedly applying autarky-reduction on F as long as possible.

Lemma (autarky decomposition)

*A DQCNF F can always be decomposed into the largest (**unique**) autark sub-DQCNF (satisfiable part by autarky) and the largest lean sub-DQCNF (lean kernel).*

Every chain of autarky reductions starting with F can be extended to it's lean kernel (where it necessarily ends).

Finding autarkies

Challenge

Finding an autarky for DQCNF is as hard as finding a satisfying assignment.

Finding autarkies

Challenge

Finding an autarky for DQCNF is as hard as finding a satisfying assignment.

Our solution: Autarky Systems

1. Restricting the range of autarkies to a more feasible domain.
   - Restrict the structure of the boolean function.
   - Restrict the number of existential variable used.
2. Maintain the good general properties of arbitrary autarkies.

A- and E-systems: A_1 Autarky system

A_1 allow the boolean functions to depend on 1 universal variable.

DQCNF formula

F $:= \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y)$:
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}\{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Example

Exactly one A_1-autarky
$a \mapsto \neg x$, $b \mapsto \neg y$
F is E_1-lean.

Deciding whether a DQCNF has a non-trivial A_1-autarky is **NP-complete**.

A- and E-systems: E_1 Autarky system

E_1 only uses one existential variable.

DQCNF formula

$F := \forall x, y\, \exists c(x,y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\}$

Example

Exactly one E_1-autarky
$c \mapsto (x \vee \neg y) \wedge (\neg x \vee y)$
F A_1-lean.

Deciding the existence of E_1-autarky can be done in **polynomial time**.

A- and E-systems: E_1 Autarky system: $E_1 + A_1$

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x, y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

A- and E-systems: E_1 Autarky system: $E_1 + A_1$

DQCNF formula

$F := \forall x, y\, \exists a(x)\, \exists b(y)\, \exists c(x,y)\, \exists d(x)\, \exists e(y) :$
$\{a, b, x\}, \{\bar{a}, \bar{b}, \bar{x}\}, \{\bar{a}, b, y\}, \{a, \bar{b}, \bar{y}\},$
$\{c, x, y\}, \{c, \bar{x}, \bar{y}, a\}, \{\bar{c}, x, \bar{y}\}, \{\bar{c}, \bar{x}, y\},$
$\{d, e, x\}, \{\bar{d}, \bar{e}, \bar{x}, c\}, \{\bar{d}, e, y\}, \{d, \bar{e}, \bar{y}\}$

Total satisfying assignment

$a \mapsto \neg x,\ b \mapsto \neg y,\ c \mapsto (x \leftrightarrow y),\ d \mapsto \neg x,\ e \mapsto \neg y$
Exactly has 4 autarkies.

Deciding the existence of $E_1 + A_1$-autarky can be done is **NP-complete**.

Translation ($\mathbf{t}$) to SAT: Finding A_1 via compilation

1. **Selected (boolean) functions**: explicitly list the possible boolean functions as values of the existential variables.

   `S(c)` $= t(c, 0), t(c, 1), t(c, x), t(c, \neg x), t(c, y), t(c, \neg y)$

Translation (**t**) to SAT: Finding A_1 via compilation

1. **Selected (boolean) functions**: explicitly list the possible boolean functions as values of the existential variables.

   $\texttt{S(c)} = t(c, 0), t(c, 1), t(c, x), t(c, \neg x), t(c, y), t(c, \neg y)$

2. **Admissible partial assignment**: compile for each clause $C \in F$ the minimal possibilities for C to become a tautology.

   $\texttt{M}(\{\texttt{c}, \bar{\texttt{x}}, \bar{\texttt{y}}, \texttt{a}\}) = p(c, 1), p(a, 1),$

Translation (**t**) to SAT: Finding A_1 via compilation

1. **Selected (boolean) functions**: explicitly list the possible boolean functions as values of the existential variables.

   $\texttt{S(c)} = t(c,0), t(c,1), t(c,x), t(c,\neg x), t(c,y), t(c,\neg y)$

2. **Admissible partial assignment**: compile for each clause $C \in F$ the minimal possibilities for C to become a tautology.

   $\texttt{M}(\{\texttt{c}, \bar{\texttt{x}}, \bar{\texttt{y}}, \texttt{a}\}) = p(c,1), p(a,1), p(c,x), p(c,y), p(a,x),$

Translation (**t**) to SAT: Finding A_1 via compilation

1. **Selected (boolean) functions**: explicitly list the possible boolean functions as values of the existential variables.

   $$\texttt{S(c)} = t(c,0), t(c,1), t(c,x), t(c,\neg x), t(c,y), t(c,\neg y)$$

2. **Admissible partial assignment**: compile for each clause $C \in F$ the minimal possibilities for C to become a tautology.

   $$\texttt{M}(\{\texttt{c},\bar{\texttt{x}},\bar{\texttt{y}},\texttt{a}\}) = p(c,1), p(a,1), p(c,x), p(c,y), p(a,x), p(c,x,a,\neg x), p(c,\neg x,a,x)$$

Translation (**t**) to SAT: Finding A_1 via compilation

1. **Selected (boolean) functions**: explicitly list the possible boolean functions as values of the existential variables.

   $$\texttt{S(c)} = t(c,0), t(c,1), t(c,x), t(c,\neg x), t(c,y), t(c,\neg y)$$

2. **Admissible partial assignment**: compile for each clause $C \in F$ the minimal possibilities for C to become a tautology.

   $$\texttt{M}(\{\texttt{c}, \bar{\texttt{x}}, \bar{\texttt{y}}, \texttt{a}\}) = p(c,1), p(a,1), p(c,x), p(c,y), p(a,x), \\ p(c,x,a,\neg x), p(c,\neg x,a,x)$$

3. **Selector-variable**: if C is touched (selected), at least one of the minimal possibilities for C is fulfilled.

   $$\texttt{ALO}(\texttt{M}(\{\texttt{c}, \bar{\texttt{x}}, \bar{\texttt{y}}\}))$$

Numbers: Computing normalforms in DQBF track

334 instances in the DQBF track of QBFEVAL'18 in 9000s.
(finding an autarky is quick, proving UNSAT: time consuming.)

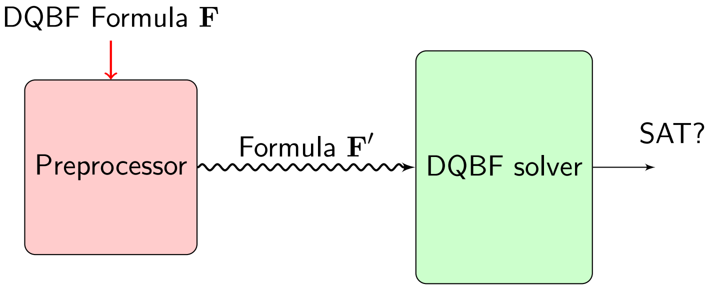
330 instances are $E_1{+}A_1$-lean (have no non-trivial E_1- or A_1-autarky).

No.	Instances	Autarky type	Reduction	
			c(F)	**c(F')**
1.	BLOEM_EQ1.DQDIMACS	**A1-satisfiable**	16	-
2.	TENTRUP17_LTL2DBA_THETA _ENVIRONMENT_1.DQDIMACS	E1+A1-satisfiable	**732**	-
3.	BLOEM_EX1.DQDIMACS	A1: non-trivial autarky	52	**18**
4.	BLOEM_EX2.DQDIMACS	A1: non-trivial autarky	139	99

Autarkies use and applications

"Preprocessing can be extremely beneficial." - Armin Biere [Biere(2011)]

1. `Preprocessing`: reduce the input formula by simplification procedures before the formula is passed to the actual solving algorithm.
2. `Inprocessing`: use the formula simplification procedures during the search process of the solver.



Conclusion

- Autarky theory for DQBF.
- Two basic autarky systems A_1, E_1 and their combination $E_1 + A_1$.
- A SAT translation.

Future Work:

- Determining the (unique) normalforms for A_1, E_1, $E_1{+}A_1$ for all over 12,000 instances in QBFLIB.
- Consider more stronger autarky systems A_2, E_2.

Thanks!

Bibliography

Armin Biere.
Preprocessing and inprocessing techniques in SAT.
In *Haifa Verification Conference*, volume 1, 2011.

Hans Kleine Büning and Oliver Kullmann.
Minimal unsatisfiability and autarkies.
Handbook of Satisfiability, 185:339–401, 2009.