PI is not at least as succinct as MODS

Nikolay Kaleyski

July 7, 2017

"A Knowledge Compilation Map", Adnan Darwiche & Pierre Marquis (2002)

L	NNF	DNNF	d-DNNF	sd-DNNF	FBDD	OBDD	$OBDD_<$	DNF	CNF	PI	IP	MODS
NNF	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
DNNF	$\not\leq^*$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\not\leq^*$	?	$\leq$	$\leq$
d-DNNF	$\not\leq^*$	$\not\leq^*$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\not\leq^*$	$\not\leq^*$	?	?	$\leq$
sd-DNNF	$\not\leq^*$	$\not\leq^*$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\not\leq^*$	$\not\leq^*$	?	?	$\leq$
FBDD	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$	$\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$
OBDD	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$
$OBDD_<$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$
DNF	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$
CNF	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$	$\not\leq$	$\leq$
PI	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\not\leq$	?
IP	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$
MODS	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$

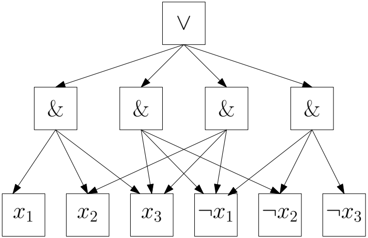
Known results in knowledge compilation (continued)

"A Knowledge Compilation Map", Adnan Darwiche & Pierre Marquis (2002)

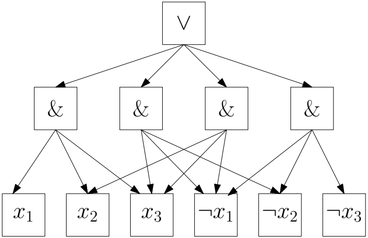
L	NNF	DNNF	d-DNNF	sd-DNNF	FBDD	OBDD	$OBDD_<$	DNF	CNF	PI	IP	MODS
NNF	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
DNNF	$\not\leq^*$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\not\leq^*$	?	$\leq$	$\leq$
d-DNNF	$\not\leq^*$	$\not\leq^*$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\not\leq^*$	$\not\leq^*$	?	?	$\leq$
sd-DNNF	$\not\leq^*$	$\not\leq^*$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\not\leq^*$	$\not\leq^*$	?	?	$\leq$
FBDD	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$	$\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$
OBDD	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$
$OBDD_<$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$
DNF	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$
CNF	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$	$\not\leq$	$\leq$
PI	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\not\leq$	?
IP	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$	$\leq$
MODS	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\not\leq$	$\leq$

Background: Sentences and Formulas

- A **sentence** is a directed acyclic graph with Boolean operations in the internal nodes and literals in the leaves.



- A **sentence** is a directed acyclic graph with Boolean operations in the internal nodes and literals in the leaves.

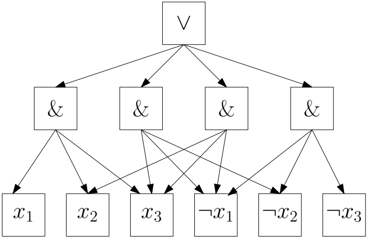


- Every sentence has an **equivalent Boolean formula**.

$$x_1x_2x_3 \vee \overline{x_1x_2}x_3 \vee \overline{x_1}x_2x_3 \vee \overline{x_1x_2x_3}$$

Background: Sentences and Formulas

- A **sentence** is a directed acyclic graph with Boolean operations in the internal nodes and literals in the leaves.



- Every sentence has an **equivalent Boolean formula**.

$$x_1x_2x_3 \vee \overline{x_1}\,\overline{x_2}x_3 \vee \overline{x_1}x_2x_3 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}$$

- We will work with formulas.

Background: Languages and Succinctness

- A **language** is a class of formulas having some given property.

Background: Languages and Succinctness

- A **language** is a class of formulas having some given property.
- Examples of languages: CNF, DNF, NNF, etc.

Background: Languages and Succinctness

- A **language** is a class of formulas having some given property.
- Examples of languages: CNF, DNF, NNF, etc.
- A language L_1 is **at least as succinct** as a language L_2 ($L_1 \leq L_2$) if there is a polynomial p such that

$$(\forall \varphi_2 \in L_2)(\exists \varphi_1 \in L_1)(\varphi_1 \equiv \varphi_2 \ \& \ |\varphi_1| \leq p(|\varphi_2|))$$

Background: MODS and PI

- A variable assignment can be expressed as a term containing all pertinent variables, e.g.

$$x_1\overline{x_2x_3}x_4x_5$$

for

$$f = \{(x_1, 1), (x_2, 0), (x_3, 0), (x_4, 1), (x_5, 1)\}$$

Background: MODS and PI

- A variable assignment can be expressed as a term containing all pertinent variables, e.g.

$$x_1 \overline{x_2 x_3} x_4 x_5$$

for

$$f = \{(x_1, 1), (x_2, 0), (x_3, 0), (x_4, 1), (x_5, 1)\}$$

- An **implicate** of a formula φ is a clause π such that

$$(\forall v : Vars(\varphi) \to \{0, 1\})(\varphi(v) = 1 \implies \pi(v) = 1)$$

for any variable assignment v.

Background: MODS and PI

- A variable assignment can be expressed as a term containing all pertinent variables, e.g.

$$x_1 \overline{x_2 x_3} x_4 x_5$$

for

$$f = \{(x_1, 1), (x_2, 0), (x_3, 0), (x_4, 1), (x_5, 1)\}$$

- An **implicate** of a formula φ is a clause π such that

$$(\forall v : Vars(\varphi) \rightarrow \{0, 1\})(\varphi(v) = 1 \implies \pi(v) = 1)$$

for any variable assignment v.

- A **prime implicate** is an implicate from which no literal can be removed without it ceasing to be an implicate.

Background: MODS and PI (continued)

- An **implicant** of a formula φ is a term τ such that

$$(\forall v : \mathit{Vars}(\varphi) \rightarrow \{0, 1\})(\tau(v) = 1 \implies \varphi(v) = 1)$$

for any variable assignment v.

Background: MODS and PI (continued)

- An **implicant** of a formula φ is a term τ such that

$$(\forall v : \mathit{Vars}(\varphi) \to \{0,1\})(\tau(v) = 1 \implies \varphi(v) = 1)$$

  for any variable assignment v.
- A **prime implicant** is an implicant from which no literal can be removed without it ceasing to be an implicant.

Background: MODS and PI (continued)

- An **implicant** of a formula φ is a term τ such that

$$(\forall v : \mathit{Vars}(\varphi) \to \{0,1\})(\tau(v) = 1 \implies \varphi(v) = 1)$$

  for any variable assignment v.
- A **prime implicant** is an implicant from which no literal can be removed without it ceasing to be an implicant.
- A formula in the **MODS language** is a list (disjunction) of all of its models (terms).

Background: MODS and PI (continued)

- An **implicant** of a formula φ is a term τ such that

$$(\forall v : Vars(\varphi) \to \{0,1\})(\tau(v) = 1 \implies \varphi(v) = 1)$$

  for any variable assignment v.
- A **prime implicant** is an implicant from which no literal can be removed without it ceasing to be an implicant.
- A formula in the **MODS language** is a list (disjunction) of all of its models (terms).
- A sentence in the **PI language** is a list (conjunction) of all of its *prime implicates*.

Background: MODS and PI (continued)

- An **implicant** of a formula φ is a term τ such that

$$(\forall v : Vars(\varphi) \to \{0,1\})(\tau(v) = 1 \implies \varphi(v) = 1)$$

  for any variable assignment v.
- A **prime implicant** is an implicant from which no literal can be removed without it ceasing to be an implicant.
- A formula in the **MODS language** is a list (disjunction) of all of its models (terms).
- A sentence in the **PI language** is a list (conjunction) of all of its *prime implicates*.
- The MODS language is not at least as succinct as PI as witnessed by

$$\Sigma = \bigvee_{i=1}^{n} x_i$$

Overview

- Inductive construction of a sequence of Boolean functions $\{\varphi_i\}_i$.

Overview

- Inductive construction of a sequence of Boolean functions $\{\varphi_i\}_i$.
- Lower bound on the number of prime **implicants** of φ_i: super-polynomial in the number of false points of φ_i.

Overview

- Inductive construction of a sequence of Boolean functions $\{\varphi_i\}_i$.
- Lower bound on the number of prime **implicants** of φ_i: super-polynomial in the number of false points of φ_i.
- The negated functions $\overline{\varphi_i}$ witness $PI \not\leq MODS$.

Overview

- Inductive construction of a sequence of Boolean functions $\{\varphi_i\}_i$.
- Lower bound on the number of prime **implicants** of φ_i: super-polynomial in the number of false points of φ_i.
- The negated functions $\overline{\varphi_i}$ witness $PI \not\leq MODS$.
- Upper bound: separation cannot be improved by better analysis.

Overview

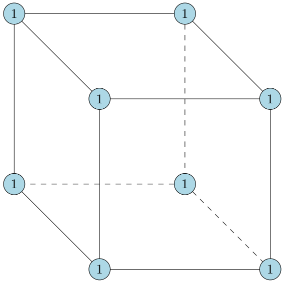
- Inductive construction of a sequence of Boolean functions $\{\varphi_i\}_i$.
- Lower bound on the number of prime **implicants** of φ_i: super-polynomial in the number of false points of φ_i.
- The negated functions $\overline{\varphi_i}$ witness $PI \not\leq MODS$.
- Upper bound: separation cannot be improved by better analysis.
- Exact formula.

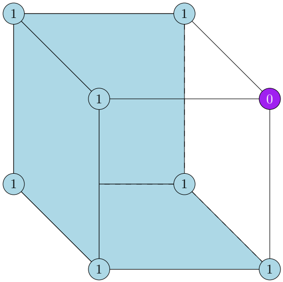
Overview

- Inductive construction of a sequence of Boolean functions $\{\varphi_i\}_i$.
- Lower bound on the number of prime **implicants** of φ_i: super-polynomial in the number of false points of φ_i.
- The negated functions $\overline{\varphi_i}$ witness $PI \not\leq MODS$.
- Upper bound: separation cannot be improved by better analysis.
- Exact formula.
- Thesis available at *Charles University's Thesis Repository*.

Finding a counterexample

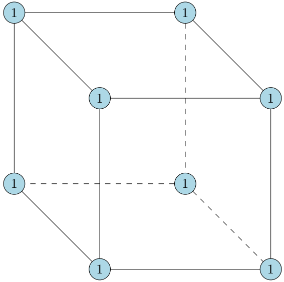
- Sequence of Boolean functions φ_i with "many" prime implicates and few models...

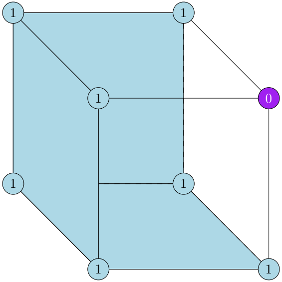




Finding a counterexample

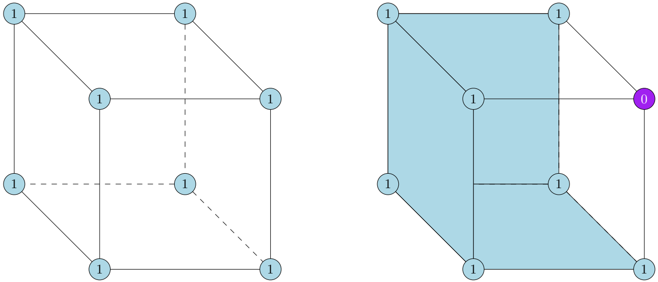
- Sequence of Boolean functions φ_i with "many" prime implicates and few models. . .
- or a sequence with "many" prime *implicants* and few *false points*.



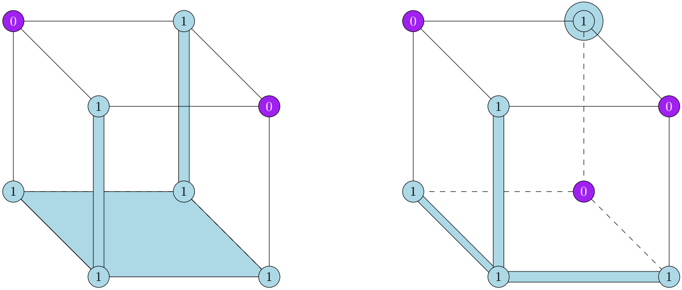


Finding a counterexample

- Sequence of Boolean functions φ_i with "many" prime implicates and few models...
- or a sequence with "many" prime *implicants* and few *false points*.
- Geometric view: inserting false points into a hypercube

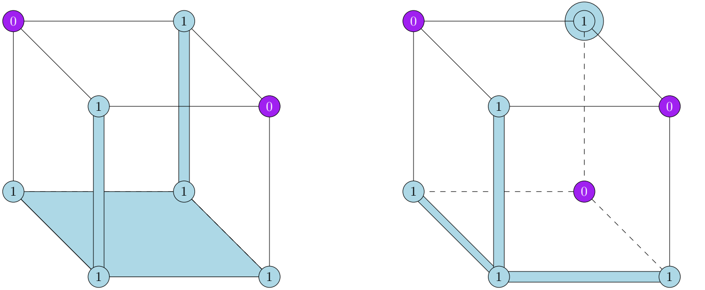


Finding a counterexample (continued)



- Intuition: insert false points, maximize Hamming distance between true points

Finding a counterexample (continued)



- Intuition: insert false points, maximize Hamming distance between true points
- Suggestion: linear code

Construction

- A sequence of matrices $\{\mathbf{A}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{A}_0 = (0)$$

$$\mathbf{A}_{i+1} = \begin{pmatrix} \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \overline{\mathbf{A}}_i \end{pmatrix}$$

Construction

- A sequence of matrices $\{\mathbf{A}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{A}_0 = (0)$$

$$\mathbf{A}_{i+1} = \begin{pmatrix} \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \overline{\mathbf{A}}_i \end{pmatrix}$$

- From these, another sequence $\{\mathbf{B}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{B}_i = \begin{pmatrix} \mathbf{A}_i \\ \overline{\mathbf{A}}_i \end{pmatrix}$$

Construction

- A sequence of matrices $\{\mathbf{A}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{A}_0 = (0)$$

$$\mathbf{A}_{i+1} = \begin{pmatrix} \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \overline{\mathbf{A}}_i \end{pmatrix}$$

- From these, another sequence $\{\mathbf{B}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{B}_i = \begin{pmatrix} \mathbf{A}_i \\ \overline{\mathbf{A}}_i \end{pmatrix}$$

- The Boolean function φ_i for $i \in \mathcal{N}$ is now defined as the function having precisely the rows of $\mathbf{B}_i$ as false points.

Construction

- A sequence of matrices $\{\mathbf{A}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{A}_0 = (0)$$

$$\mathbf{A}_{i+1} = \begin{pmatrix} \mathbf{A}_i & \mathbf{A}_i \\ \mathbf{A}_i & \overline{\mathbf{A}}_i \end{pmatrix}$$

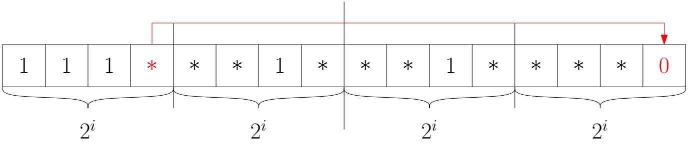
- From these, another sequence $\{\mathbf{B}_i\}_{i \in \mathcal{N}}$ is defined as

$$\mathbf{B}_i = \begin{pmatrix} \mathbf{A}_i \\ \overline{\mathbf{A}}_i \end{pmatrix}$$

- The Boolean function φ_i for $i \in \mathcal{N}$ is now defined as the function having precisely the rows of $\mathbf{B}_i$ as false points.
- It is shown that φ_i has many prime *implicants*; then its negation $\overline{\varphi}_i$ has many prime *imlicates*.

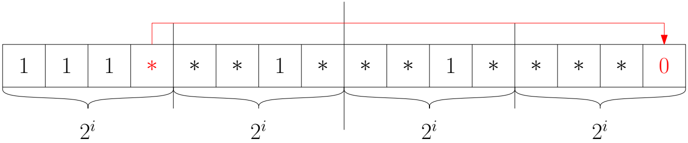
Number of prime implicants: Lower bound

- Given a prime implicant p of φ_i with a single negative literal whose positive part agrees with precisely one row of $\mathbf{B}_i$, we can construct 2^i different prime implicants of φ_{i+2} via the following construction step:



Number of prime implicants: Lower bound

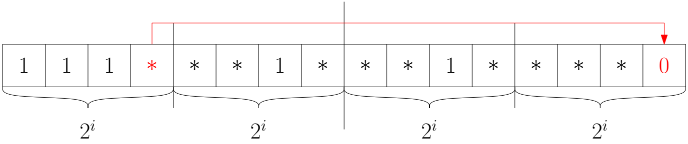
- Given a prime implicant p of φ_i with a single negative literal whose positive part agrees with precisely one row of $\mathbf{B}_i$, we can construct 2^i different prime implicants of φ_{i+2} via the following construction step:



- By induction, φ_i has $\Omega\left(2^{\frac{(i-1)^2-1}{4}}\right)$ prime implicants ...

Number of prime implicants: Lower bound

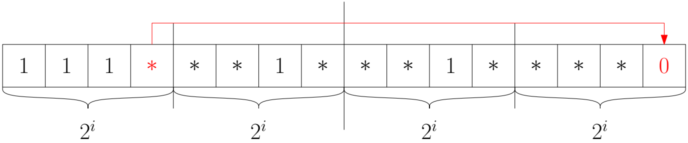
- Given a prime implicant p of φ_i with a single negative literal whose positive part agrees with precisely one row of $\mathbf{B}_i$, we can construct 2^i different prime implicants of φ_{i+2} via the following construction step:



- By induction, φ_i has $\Omega\left(2^{\frac{(i-1)^2-1}{4}}\right)$ prime implicants ...
- ... but "only" $\Theta(2^i)$ false points.

Number of prime implicants: Lower bound

- Given a prime implicant p of φ_i with a single negative literal whose positive part agrees with precisely one row of $\mathbf{B}_i$, we can construct 2^i different prime implicants of φ_{i+2} via the following construction step:



- By induction, φ_i has $\Omega\left(2^{\frac{(i-1)^2-1}{4}}\right)$ prime implicants ...
- ...but "only" $\Theta(2^i)$ false points.
- Hence $\overline{\varphi_i}$ has "many" prime *implicates* w.r.t. to its number of *true* points, or models.

Number of prime implicants: Upper bound

- The prime implicants considered are very specific.

Number of prime implicants: Upper bound

- The prime implicants considered are very specific.
- Is an exponential separation of PI and MODS possible via φ_i?

Number of prime implicants: Upper bound

- The prime implicants considered are very specific.
- Is an exponential separation of PI and MODS possible via φ_i?
- A generalization of the above construction is needed.

Number of prime implicants: Upper bound

- The prime implicants considered are very specific.
- Is an exponential separation of PI and MODS possible via φ_i?
- A generalization of the above construction is needed.
- Fixing the values of variables has "global effects" and affects other variables as well:

1	1	*	1	*	*	*	*	*	*	*	*	*	*	*	*
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}

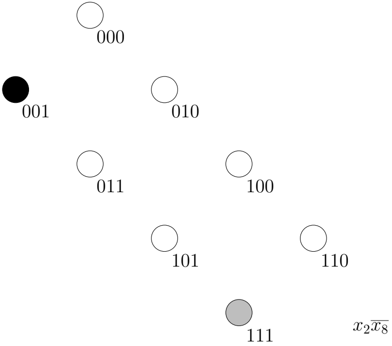
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1

Number of prime implicants: Upper bound

- A graph-based representation is used to express the polar relations between pairs of variables.

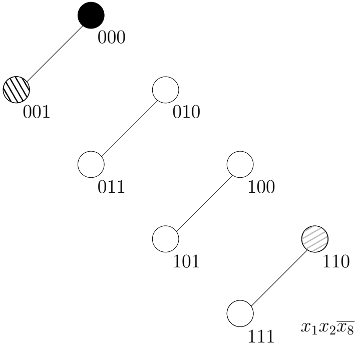
Number of prime implicants: Upper bound

- A graph-based representation is used to express the polar relations between pairs of variables.
- Each non-redundant fixation halves the number of connected components.



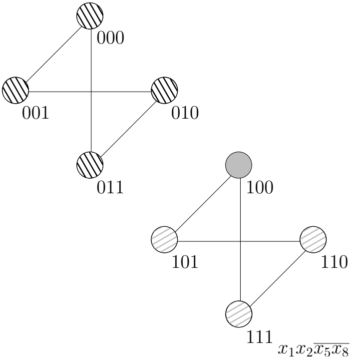
Number of prime implicants: Upper bound

- A graph-based representation is used to express the polar relations between pairs of variables.
- Each non-redundant fixation halves the number of connected components.



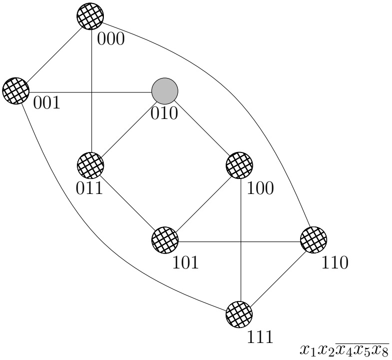
Number of prime implicants: Upper bound

- A graph-based representation is used to express the polar relations between pairs of variables.
- Each non-redundant fixation halves the number of connected components.



Number of prime implicants: Upper bound

- A graph-based representation is used to express the polar relations between pairs of variables.
- Each non-redundant fixation halves the number of connected components.



Number of prime implicants: Upper bound

- The proof is somewhat technical.

Number of prime implicants: Upper bound

- The proof is somewhat technical.
- Two variables can be fixed without merging any connected components (one positive, one negative).

Number of prime implicants: Upper bound

- The proof is somewhat technical.
- Two variables can be fixed without merging any connected components (one positive, one negative).
- Every further fixation halves the number of components or introduces redundancy.

Number of prime implicants: Upper bound

- The proof is somewhat technical.
- Two variables can be fixed without merging any connected components (one positive, one negative).
- Every further fixation halves the number of components or introduces redundancy.
- After $2 + i$ fixations, the graph becomes connected and no further non-redundant fixations can be performed.

Number of prime implicants: Upper bound

- The proof is somewhat technical.
- Two variables can be fixed without merging any connected components (one positive, one negative).
- Every further fixation halves the number of components or introduces redundancy.
- After $2 + i$ fixations, the graph becomes connected and no further non-redundant fixations can be performed.
- Thus no prime implicant of φ_i can have more than $(i + 2)$ literals, and their number is at most

$$\sum_{l=1}^{i+2} \binom{n}{l} 2^l \leq 3^{n+2} \in \Theta(3^n)$$

Number of prime implicants: Upper bound

- The proof is somewhat technical.
- Two variables can be fixed without merging any connected components (one positive, one negative).
- Every further fixation halves the number of components or introduces redundancy.
- After $2 + i$ fixations, the graph becomes connected and no further non-redundant fixations can be performed.
- Thus no prime implicant of φ_i can have more than $(i + 2)$ literals, and their number is at most

$$\sum_{l=1}^{i+2} \binom{n}{l} 2^l \leq 3^{n+2} \in \Theta(3^n)$$

- The number of false points of φ_n is exponential in n!

Number of prime implicants: Exact formula

- The polar graphs are analyzed further by observing a correspondence between variable fixations yielding connected graphs and generating sets of the vector space $\{0,1\}^i$.

Number of prime implicants: Exact formula

- The polar graphs are analyzed further by observing a correspondence between variable fixations yielding connected graphs and generating sets of the vector space $\{0,1\}^i$.
- The exact number of prime implicants of φ_i is

$$\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \frac{2^{n+2i+1} \cdot \alpha_n^{2i}}{(2i+1)(2i+2)}$$

where α_n^i is the number of i-element linearly independent sets in $\{0,1\}^n$.