Prime Compilation of Non-Clausal Formulae

Alessandro Previti[1], Alexey Ignatiev[2], Antonio Morgado[2], Joao Marques-Silva[2]

[1]HIIT, Department of Computer Science, University of Helsinki

[2]IST/INESC-ID, Portugal

CL Day 2016

Motivation

- Knowledge Compilation [DM2002,CD1997]

Motivation

- Knowledge Compilation [DM2002,CD1997]
- Formula Minimization [Q1952,Q1959,M1956]

Motivation

- Knowledge Compilation [DM2002,CD1997]
- Formula Minimization [Q1952,Q1959,M1956]
- Model-based diagnosis [dK1992]

Motivation

- Knowledge Compilation [DM2002,CD1997]
- Formula Minimization [Q1952,Q1959,M1956]
- Model-based diagnosis [dK1992]
- Inductive generalization in model checking [BM2007]
- Modal logic [B2009]
- ...

Contributions

- A new approach that can compile **non-clausal formulae**
- Can compile formulae with **thousands of variables**
- It's completely based on **SAT technology**

Basic Definitions

A **literal** is a variable or its negation

- **Clause**: A **disjunction** of literals

$$(c \vee \neg a)$$

Satisfied clause: at least one literal is **true** under the given assignment to variables

- **Term**: A **conjunction** of literals

$$(c \wedge \neg a)$$

Satisfied term: all of its literals are **true** under the given assignment to variables

Propositional formulae

- **Clausal**:
  - **CNF**: conjunction of clauses

$$(c \vee a) \wedge (c \vee \neg a)$$

  - **DNF**: disjunction of terms

$$(c \wedge a) \vee (c \wedge \neg a)$$

- **Non-clausal**:
  - Non-CNF and Non-DNF
  - **Propositional formulae**: well-formed formulae built with standard connectives $\neg, \wedge, \vee$

$$((c \wedge a) \vee (c \wedge \neg a)) \wedge d$$

Prime Implicants and Prime Implicates

- A **term** I_n is called an **implicant** of F if $I_n \vDash F$.
  - An implicant I_n of F is called **prime** if any subset $I_n' \subsetneq I_n$ is not an implicant of F.

Prime Implicants and Prime Implicates

- A **term** I_n is called an **implicant** of F if $I_n \vDash F$.
  - An implicant I_n of F is called **prime** if any subset $I'_n \subsetneq I_n$ is not an implicant of F.
    - A **satisfying assignment** (**model**) expressed as a conjunction of literals is an **implicant** (i.e if $p = 1$, $s = 0$, $t = 1$ is a satisfying assignment then $p \wedge \neg s \wedge t$ is an implicant)

Prime Implicants and Prime Implicates

- A **term** I_n is called an **implicant** of F if $I_n \models F$.
  - An implicant I_n of F is called **prime** if any subset $I'_n \subsetneq I_n$ is not an implicant of F.
    - A **satisfying assignment** (**model**) expressed as a conjunction of literals is an **implicant** (i.e if $p = 1$, $s = 0$, $t = 1$ is a satisfying assignment then $p \wedge \neg s \wedge t$ is an implicant)

- A **clause** I_e is called an **implicate** of F if $F \models I_e$.
  - An implicate I_e of F is called **prime** if any subset $I'_e \subsetneq I_e$ is not an implicate of F.

Example Prime Implicants/Implicates

$$F = (p \overset{C_1}{\vee} s) \wedge (r \vee \overset{C_2}{t} \vee \neg s) \wedge (r \overset{C_3}{\vee} \neg t)$$

Example Prime Implicants/Implicates

$$F = (\overset{C_1}{p \vee s}) \wedge (\overset{C_2}{r \vee t \vee \neg s}) \wedge (\overset{C_3}{r \vee \neg t})$$

- Implicate: $\overset{C_4}{p \vee r \vee t}$ $\Rightarrow$ obtained by resolution of C_1 and C_2

Example Prime Implicants/Implicates

$$F = (\overset{C_1}{p \vee s}) \wedge (\overset{C_2}{r \vee t \vee \neg s}) \wedge (\overset{C_3}{r \vee \neg t})$$

- Implicate: $\overset{C_4}{p \vee r \vee t}$ $\Rightarrow$ obtained by resolution of C_1 and C_2
  - **Prime implicate**: $p \vee r$ $\Rightarrow$ obtained by resolution of C_3 and C_4

Example Prime Implicants/Implicates

$$F = (\overset{C_1}{p \vee s}) \wedge (\overset{C_2}{r \vee t \vee \neg s}) \wedge (\overset{C_3}{r \vee \neg t})$$

- Implicate: $\overset{C_4}{p \vee r \vee t} \Rightarrow$ obtained by resolution of C_1 and C_2
  - **Prime implicate**: $p \vee r \Rightarrow$ obtained by resolution of C_3 and C_4
- Implicant: $p \wedge \neg s \wedge r \wedge \neg t$ $(p = 1, s = 0, r = 1, t = 0)$

Example Prime Implicants/Implicates

$$F = (\overset{C_1}{p \lor s}) \land (\overset{C_2}{r \lor t \lor \neg s}) \land (\overset{C_3}{r \lor \neg t})$$

- Implicate: $\overset{C_4}{p \lor r \lor t}$ $\Rightarrow$ obtained by resolution of C_1 and C_2
  - **Prime implicate**: $p \lor r$ $\Rightarrow$ obtained by resolution of C_3 and C_4
- Implicant: $p \land \neg s \land r \land \neg t$ $(p = 1, s = 0, r = 1, t = 0)$
  - **Prime implicant**: $p \land r$ $(p = 1, r = 1)$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= (\neg a \lor \overset{C_1}{b} \lor c) \land (a \lor \overset{C_2}{d}) \land (\neg d \lor \overset{C_3}{e} \lor f) \\
I_n &= \neg a \land b \land c \land d \land e \land \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= (\neg a \overset{C_1}{\vee} b \vee c) \wedge (a \overset{C_2}{\vee} d) \wedge (\neg d \overset{C_3}{\vee} e \vee f) \\
I_n &= \neg a \wedge b \wedge c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$F = \overset{C_1}{(\neg a \vee b \vee c)} \wedge \overset{C_2}{(a \vee d)} \wedge \overset{C_3}{(\neg d \vee e \vee f)}$$
$$I_n = b \wedge c \wedge d \wedge e \wedge \neg f$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= (\neg a \lor \overset{C_1}{b} \lor c) \land (a \overset{C_2}{\lor} d) \land (\neg d \overset{C_3}{\lor} e \lor f) \\
I_n &= b \land c \land d \land e \land \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= \overbrace{(\neg a \vee b \vee c)}^{C_1} \wedge \overbrace{(a \vee d)}^{C_2} \wedge \overbrace{(\neg d \vee e \vee f)}^{C_3} \\
I_n &= c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= \overbrace{(\neg a \vee b \vee c)}^{C_1} \wedge \overbrace{(a \vee d)}^{C_2} \wedge \overbrace{(\neg d \vee e \vee f)}^{C_3} \\
I_n &= c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= \overbrace{(\neg a \vee b \vee c)}^{C_1} \wedge \overbrace{(a \vee d)}^{C_2} \wedge \overbrace{(\neg d \vee e \vee f)}^{C_3} \\
I_n &= c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= (\neg a \vee \overset{C_1}{b} \vee c) \wedge (a \overset{C_2}{\vee} d) \wedge (\neg d \overset{C_3}{\vee} e \vee f) \\
I_n &= c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= \overbrace{(\neg a \lor b \lor c)}^{C_1} \land \overbrace{(a \lor d)}^{C_2} \land \overbrace{(\neg d \lor e \lor f)}^{C_3} \\
I_n &= c \land d \land e \land \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= \overbrace{(\neg a \vee b \vee c)}^{C_1} \wedge \overbrace{(a \vee d)}^{C_2} \wedge \overbrace{(\neg d \vee e \vee f)}^{C_3} \\
I_n &= c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= \overset{C_1}{(\neg a \vee b \vee c)} \wedge \overset{C_2}{(a \vee d)} \wedge \overset{C_3}{(\neg d \vee e \vee f)} \\
I_n &= c \wedge d \wedge e \wedge \neg f
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$\begin{aligned} F &= \overset{C_1}{(\neg a \vee b \vee c)} \wedge \overset{C_2}{(a \vee d)} \wedge \overset{C_3}{(\neg d \vee e \vee f)} \\ I_n &= c \wedge d \wedge e \wedge \neg f \end{aligned}$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= (\neg a \lor \overset{C_1}{b} \lor c) \land (a \overset{C_2}{\lor} d) \land (\neg d \overset{C_3}{\lor} e \lor f) \\
I_n &= c \land d \land e
\end{aligned}
$$

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$
\begin{aligned}
F &= (\neg a \vee \overset{C_1}{b} \vee c) \wedge (a \overset{C_2}{\vee} d) \wedge (\neg d \overset{C_3}{\vee} e \vee f) \\
I_n &= c \wedge d \wedge e
\end{aligned}
$$

- **Propositional** formulae:
  1. Shannon expansion: Worst-case **exponential grow** of the formula

Reduction of Implicants/Implicates

- **CNF** formulae:
  - **Polynomial time** procedure

$$\begin{aligned} F &= \overbrace{(\neg a \vee b \vee c)}^{C_1} \wedge \overbrace{(a \vee d)}^{C_2} \wedge \overbrace{(\neg d \vee e \vee f)}^{C_3} \\ I_n &= c \wedge d \wedge e \end{aligned}$$

- **Propositional** formulae:
  1. Shannon expansion: Worst-case **exponential grow** of the formula

OR

  1. Convert F to F_{CNF} using a **Tseitin encoding**
  2. Reduce an implicant I_n using the following procedure:

input : Formula F_{CNF}, I_n and $Var(F)$
output: Prime Implicant in I_n
foreach $l \in I_n$ *and* $var(l) \in Var(F)$ **do**
    $I'_n = I_n \setminus \{l\}$
    if $I'_n \wedge \neg F$ *unsat* **then**
        $I_n = I'_n$
    else
        continue
return I_n
end

Related work and drawback

- Iterated consensus or resolution [Q1952,Q1959,T1967]
- Unionist product [C1996]
- Based on dual rail encoding [P1999,J2014]
- Semantic resolution [S1970]
- SE-trees [R1994]
- BDD-based (i.e ZRes) [SD2001]

Related work and drawback

- Iterated consensus or resolution [Q1952,Q1959,T1967]
- Unionist product [C1996]
- Based on dual rail encoding [P1999,J2014]
- Semantic resolution [S1970]
- SE-trees [R1994]
- BDD-based (i.e ZRes) [SD2001]

They all assume the formula in CNF (DNF) or they are limited to formulae with few variables

Hitting Set Duality (1)

- Minimal Hitting Set (MHS):
  Given a collection Γ of sets, a hitting set H for Γ is a set such that $\forall S \in \Gamma, H \cap S \neq \emptyset$.
  - A hitting set H is *minimal* if none of its subsets is a hitting set.

Hitting Set Duality (1)

- Minimal Hitting Set (MHS):
  Given a collection Γ of sets, a hitting set H for Γ is a set such that $\forall S \in \Gamma, H \cap S \neq \emptyset$.
  - A hitting set H is *minimal* if none of its subsets is a hitting set.

- Example Minimal Hitting Set:

$$\Gamma = \begin{bmatrix} \{a, b, c\} \\ \{b, d\} \\ \{e\} \end{bmatrix}$$

$H_1 = \{b, e\}$, $H_2 = \{a, d, e\}$, $H_3 = \{c, d, e\}$.
Note that instead $\{a, b, e\}$ is not a Minimal Hitting Set.

Hitting Set Duality (2)

- Prime Implicants and Implicates are related by a hitting set duality
  - $PI_n(F)$: set of all prime implicants of F
  - $PI_e(F)$: set of all prime implicates of F

> A term (clause) I is a prime implicant (implicate) of F if and only if I is a minimal hitting set of $PI_e(F)$ ($PI_n(F)$)

This remains true for any subset of $PI_e(F)(PI_n(F))$ that is equivalent to F (cover)

MHS on subsets of $PI_e(F)(PI_n(F))$

- Suppose $PI_e^{'}(F) \subset PI_e(F)$ and $PI_e^{'}(F)$ not equivalent to F
  - A MHS p of $PI_e^{'}(F)$ does not necessarily corresponds to a prime implicant

A term p is a prime implicant of F if

1. p is a MHS of $PI_e^{'}(F)$
2. $p \wedge \neg F$ is unsatisfiable

- When sets are represented as clauses with positive literals, minimal models correspond to MHS
  - A minimal model is a model containing a minimal number of variables assigned to true

Prime Compilation of Non-Clausal Formulae

- An approach completely based on SAT technology

Prime Compilation of Non-Clausal Formulae

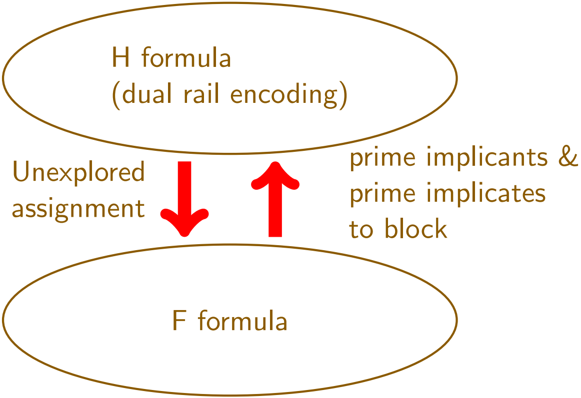
- An approach completely based on SAT technology
- Exploits the existing duality between prime implicants and prime implicates in order to find new prime implicants/implicates

Prime Compilation of Non-Clausal Formulae

- An approach completely based on SAT technology
- Exploits the existing duality between prime implicants and prime implicates in order to find new prime implicants/implicates
- Complements existing approaches (i.e ZRes) [SD'01]

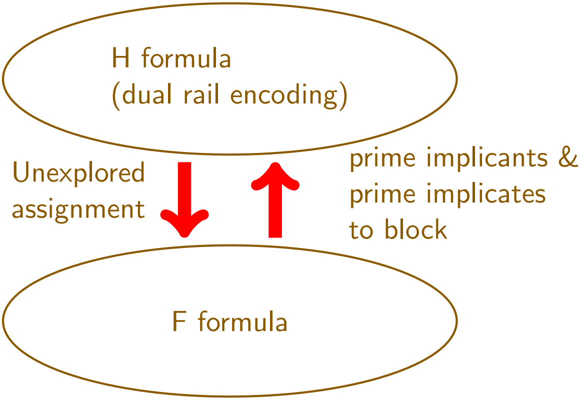
H formula

- We use a CNF formula H to keep track of the already computed prime implicants/implicates



H formula

- We use a CNF formula H to keep track of the already computed prime implicants/implicates



When H is unsatisfiable either all the $PI_n(F)$ or all the $PI_e(F)$ have been computed

Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$

$$H = (b \vee \neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee \neg d) \wedge (b \vee c) \wedge d \wedge (a \vee \neg b)$$

$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$

$$H = (b \vee \neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee \neg d) \wedge (b \vee c) \wedge d \wedge (a \vee \neg b)$$

$$(b \vee \neg c)$$

$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$

$$H = (b \vee \neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee \neg d) \wedge (b \vee c) \wedge d \wedge (a \vee \neg b)$$

$(b \vee \neg c)$ $(\neg a \vee \neg b)$

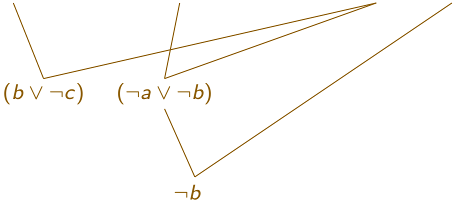
$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$

$$H = (b \vee \neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee \neg d) \wedge (b \vee c) \wedge d \wedge (a \vee \neg b)$$

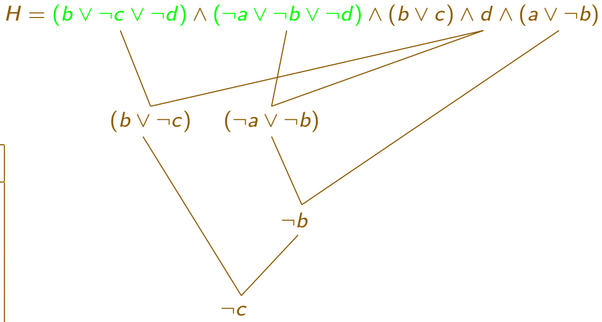


$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$



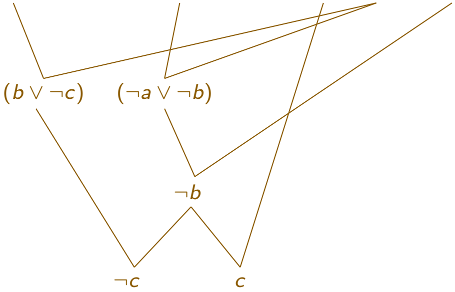
$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$

$$H = (b \vee \neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee \neg d) \wedge (b \vee c) \wedge d \wedge (a \vee \neg b)$$



$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

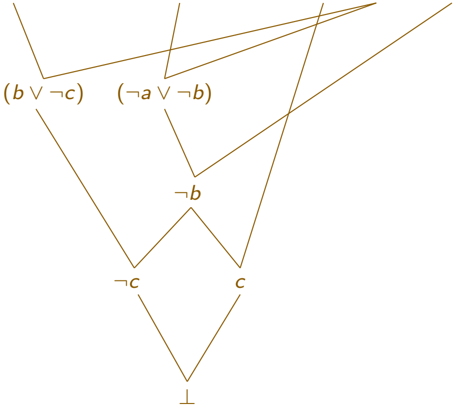
Dual Rail Encoding (1)

- Prime implicants/implicates can be more than 2^n
- Example without dual rail encoding:

$$F = d \wedge (a \vee \neg b \vee \neg d) \wedge (c \vee b)$$

$PI_n(F)$	$PI_e(F)$
$\neg b \wedge c \wedge d$	$b \vee c$
$a \wedge c \wedge d$	d
$a \wedge b \wedge d$	$a \vee c$
	$a \vee \neg b$

$$H = (b \vee \neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee \neg d) \wedge (b \vee c) \wedge d \wedge (a \vee \neg b)$$



Dual Rail Encoding (2)

- For each variable v in $var(F)$ create two variables x_v and $x_{\neg v}$:
  1. $(x_v = 1$ and $x_{\neg v} = 0) \Rightarrow v = 1$
  2. $(x_v = 0$ and $x_{\neg v} = 1) \Rightarrow v = 0$
  3. $(x_v = 0$ and $x_{\neg v} = 0) \Rightarrow v$ is a don't care
  4. $(x_v = 1$ and $x_{\neg v} = 1) \Rightarrow$ forbidden

  In order to achieve the requirement of point 4 add the clause $\{(\neg x_v \vee \neg x_{\neg v}) \mid v \in var(F)\}$

Algorithm primer-b

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \mathrm{var}(F)\}$
while true **do**
  $(\mathrm{st}, A^H) \leftarrow \mathrm{MinModel}(H)$
  if not st **then return**
  $A^F \leftarrow \mathrm{Map}(A^H)$
  $(\mathrm{st}, M^{\neg F}) \leftarrow \mathtt{SAT}(A^F \wedge \neg F)$
  if st **then** # $F \models \neg M^{\neg F}$; i.e. $\neg M^{\neg F}$ is an implicate
    $I_e \leftarrow \mathtt{ReduceImplicate}(M^{\neg F}, F)$
    $\mathtt{ReportImplicate}(I_e)$
    $b \leftarrow \{x_l \mid l \in I_e\}$
  else # $A^F \models F$; i.e. A^F is an implicant
    $I_n \leftarrow A^F$
    $\mathtt{ReportImplicant}(I_n)$
    $b \leftarrow \{\neg x_l \mid l \in I_n\}$
  $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
  $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
  if not st **then return**
  $A^F \leftarrow \text{Map}(A^H)$
  $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
  if st **then**
    $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
    $\texttt{ReportImplicate}(I_e)$
    $b \leftarrow \{x_l \mid l \in I_e\}$
  else
    $I_n \leftarrow A^F$
    $\texttt{ReportImplicant}(I_n)$
    $b \leftarrow \{\neg x_l \mid l \in I_n\}$
  $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

$$B = \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in var(F)\}$$

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

$$H = B \wedge (x_b \vee x_c) \wedge x_d \wedge (x_a \vee x_c) \wedge (x_a \vee x_{\neg b})$$

	$x_a\ x_{\neg a}$	$x_b\ x_{\neg b}$	$x_c\ x_{\neg c}$	$x_d\ x_{\neg d}$
$A^H =$	10	00	10	10

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \mathrm{var}(F)\}$
while true **do**
    $(\mathrm{st}, A^H) \leftarrow \mathrm{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \mathrm{Map}(A^H)$
    $(\mathrm{st}, M^{\neg F}) \leftarrow \mathtt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \mathtt{ReduceImplicate}(M^{\neg F}, F)$
        $\mathtt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\mathtt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

If unsatisfiable then all the prime implicants have been computed!!

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \mathrm{var}(F)\}$
while true **do**
  $(\mathrm{st}, A^H) \leftarrow \mathrm{MinModel}(H)$
  if not st **then return**
  $A^F \leftarrow \mathrm{Map}(A^H)$
  $(\mathrm{st}, M^{\neg F}) \leftarrow \mathtt{SAT}(A^F \wedge \neg F)$
  if st **then**
    $I_e \leftarrow \mathtt{ReduceImplicate}(M^{\neg F}, F)$
    $\mathtt{ReportImplicate}(I_e)$
    $b \leftarrow \{x_l \mid l \in I_e\}$
  else
    $I_n \leftarrow A^F$
    $\mathtt{ReportImplicant}(I_n)$
    $b \leftarrow \{\neg x_l \mid l \in I_n\}$
  $H \leftarrow H \cup \{b\}$
end

	$(x_a\ x_{\neg a})$	$(x_b\ x_{\neg b})$	$(x_c\ x_{\neg c})$	$(x_d\ x_{\neg d})$
$A^H =$	10	00	10	10
		$\Downarrow$		
$A^F =$	1 (a)	D (b)	1 (c)	1 (d)

Assignment to test: $a \wedge c \wedge d$

Algorithm

input : Formula F

output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$

while true **do**

    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$

    if not st **then return**

    $A^F \leftarrow \text{Map}(A^H)$

    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$

    if st **then**

        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$

        $\texttt{ReportImplicate}(I_e)$

        $b \leftarrow \{x_l \mid l \in I_e\}$

    else

        $I_n \leftarrow A^F$

        $\texttt{ReportImplicant}(I_n)$

        $b \leftarrow \{\neg x_l \mid l \in I_n\}$

    $H \leftarrow H \cup \{b\}$

end

$$(st, M^{\neg F}) \leftarrow SAT(a \wedge c \wedge d \wedge \neg F)$$

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

$a \wedge c \wedge d \wedge \neg F$ unsatisfiable

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

input : Formula F_{CNF}, I_e and $Var(F)$
output: Prime Implicate in I_e

foreach $l \in I_e$ *and* $var(l) \in Var(F)$ **do**
    $I'_e = I_e \setminus \{l\}$
    if $I'_e \wedge F$ *unsat* **then**
        $I_e = I'_e$
    else
        continue
return I_e
end

Algorithm

input : Formula F

output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \mathrm{var}(F)\}$
while true **do**
    $(\mathrm{st}, A^H) \leftarrow \mathrm{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \mathrm{Map}(A^H)$
    $(\mathrm{st}, M^{\neg F}) \leftarrow \mathtt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \mathtt{ReduceImplicate}(M^{\neg F}, F)$
        $\mathtt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\mathtt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \mathrm{var}(F)\}$
while true **do**
  $(\mathrm{st}, A^H) \leftarrow \mathrm{MinModel}(H)$
  if not st **then return**
  $A^F \leftarrow \mathrm{Map}(A^H)$
  $(\mathrm{st}, M^{\neg F}) \leftarrow \mathtt{SAT}(A^F \wedge \neg F)$
  if st **then**
    $I_e \leftarrow \mathtt{ReduceImplicate}(M^{\neg F}, F)$
    $\mathtt{ReportImplicate}(I_e)$
    $b \leftarrow \{x_l \mid l \in I_e\}$
  else
    $I_n \leftarrow A^F$
    $\mathtt{ReportImplicant}(I_n)$
    $b \leftarrow \{\neg x_l \mid l \in I_n\}$
  $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

$I_n \leftarrow a \wedge c \wedge d$

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
  $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
  if not st **then return**
  $A^F \leftarrow \text{Map}(A^H)$
  $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
  if st **then**
    $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
    $\texttt{ReportImplicate}(I_e)$
    $b \leftarrow \{x_l \mid l \in I_e\}$
  else
    $I_n \leftarrow A^F$
    $\texttt{ReportImplicant}(I_n)$
    $b \leftarrow \{\neg x_l \mid l \in I_n\}$
  $H \leftarrow H \cup \{b\}$
end

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

$$b \leftarrow (\neg x_a \vee \neg x_c \vee \neg x_d)$$

Algorithm

input : Formula F
output: $PI_n(F)$ and prime implicate cover of F

$H \leftarrow \{(\neg x_v \vee \neg x_{\neg v}) \mid v \in \text{var}(F)\}$
while true **do**
    $(\text{st}, A^H) \leftarrow \text{MinModel}(H)$
    if not st **then return**
    $A^F \leftarrow \text{Map}(A^H)$
    $(\text{st}, M^{\neg F}) \leftarrow \texttt{SAT}(A^F \wedge \neg F)$
    if st **then**
        $I_e \leftarrow \texttt{ReduceImplicate}(M^{\neg F}, F)$
        $\texttt{ReportImplicate}(I_e)$
        $b \leftarrow \{x_l \mid l \in I_e\}$
    else
        $I_n \leftarrow A^F$
        $\texttt{ReportImplicant}(I_n)$
        $b \leftarrow \{\neg x_l \mid l \in I_n\}$
    $H \leftarrow H \cup \{b\}$
end

Example using dual rail encoding and minimal models

$H = B \wedge (\neg x_{\neg b} \vee \neg x_c \vee \neg x_d) \wedge (\neg x_a \vee \neg x_b \vee \neg x_d) \wedge (x_b \vee x_c) \wedge x_d \wedge (x_a \vee x_c) \wedge (x_a \vee x_{\neg b})$

	$(x_a\ x_{\neg a})$	$(x_b\ x_{\neg b})$	$(x_c\ x_{\neg c})$	$(x_d\ x_{\neg d})$
$A^H =$	10	00	10	10

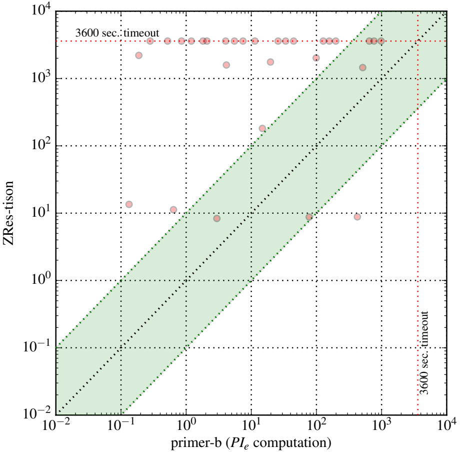
$$\Downarrow$$

$A^F =$	1	D	1	1
	(a)	(b)	(c)	(d)

$a \wedge c \wedge d$ **is the last remaining prime implicant and is returned as a model!**

Results

	QG6	Geffe gen.	F+PHP	F+GT	Total
ZRes-tison	0	0	11	0	11
primer-a (PI_n)	53	**596**	**30**	26	705
primer-a (PI_e)	28	588	**30**	27	673
primer-b (PI_n)	**64**	595	**30**	**30**	**719**
primer-b (PI_e)	30	577	**30**	27	664



primer-b vs Zres on F+PHP family

Conclusion & Future Work

- Presented a new approach that can compile **non-clausal formulae**
- Can compile formulae with **thousands of variables**
- It's completely based on **SAT technology**
- Complements existing approaches

- Future work
  - Applications of prime enumeration
    - SAT-Based Formula Simplification [IPM15]
  - Preferred prime implicants/implicates
    - Horn LUB [MPM15]

Thank You