Counterexample Guided Abstraction Refinement Algorithm for Propositional Circumscription

Mikoláš Janota[1] Radu Grigore[2] Joao Marques-Silva[3]

[1]INESC-ID, Lisbon, Portugal

[2]Queen Mary, University of London

[3]University College Dublin, Ireland

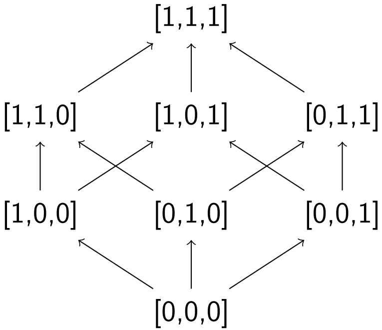
Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some 1-values to 0, yields a non-model.

Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some 1-values to 0, yields a non-model.

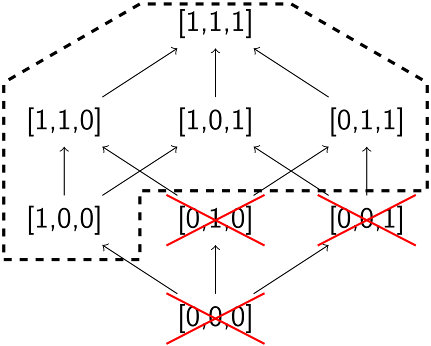
$\mathbf{x} \vee (\mathbf{y} \wedge \mathbf{z})$



Minimal Models

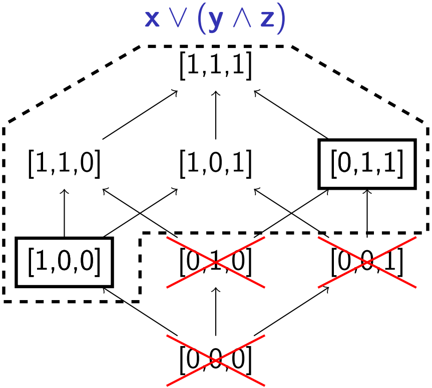
- A model of a formula is (point-wise) minimal *iff* flipping some 1-values to 0, yields a non-model.

$\mathbf{x} \vee (\mathbf{y} \wedge \mathbf{z})$



Minimal Models

- A model of a formula is (point-wise) minimal *iff* flipping some 1-values to 0, yields a non-model.



Entailment in Circumscription

- let τ and ψ be propositional formulas
- the problem of entailment in circumscription is to decide whether ψ holds in all minimal models of τ

$$\tau \models_{\min} \psi$$

Entailment in Circumscription

- let τ and ψ be propositional formulas
- the problem of entailment in circumscription is to decide whether ψ holds in all minimal models of τ

$$\tau \models_{\min} \psi$$

- Generalized Closed World Assumption (GCWA) means computing variables that are 0 in all minimal models

$$\{x \mid \tau \models_{\min} \neg x\}$$

Entailment in Circumscription

- let τ and ψ be propositional formulas
- the problem of entailment in circumscription is to decide whether ψ holds in all minimal models of τ

$$\tau \models_{\min} \psi$$

- Generalized Closed World Assumption (GCWA) means computing variables that are 0 in all minimal models

$$\{x \mid \tau \models_{\min} \neg x\}$$

- $x \vee y \models_{\min} \neg(x \wedge y)$

Entailment in Circumscription

- let τ and ψ be propositional formulas
- the problem of entailment in circumscription is to decide whether ψ holds in all minimal models of τ

$$\tau \models_{\min} \psi$$

- Generalized Closed World Assumption (GCWA) means computing variables that are 0 in all minimal models

$$\{x \mid \tau \models_{\min} \neg x\}$$

- $x \vee y \models_{\min} \neg(x \wedge y)$
- for $\tau = (x \vee y) \wedge (z \Rightarrow w)$

$$\mathsf{GCWA}(\tau) = \{z, w\}$$

Complexity

- How hard is to decide $\tau \models_{\min} \psi$?

Complexity

- How hard is to decide $\tau \models_{\min} \psi$?
- It is in the second level of polynomial hierarchy Π_2^P

Motivation

- It is complete for Π_2^P, so other problems can be converted to it.
- Circumscription is an important form of non-monotonic reasoning.
- Recently GCWA has been applied in interactive configuration.

Plan of Attack

$$\tau \models_{\min} \psi$$

- We are going to use a SAT solver — a tool that decides whether a formula is satisfiable or not.

Plan of Attack

$$\tau \models_{\min} \psi$$

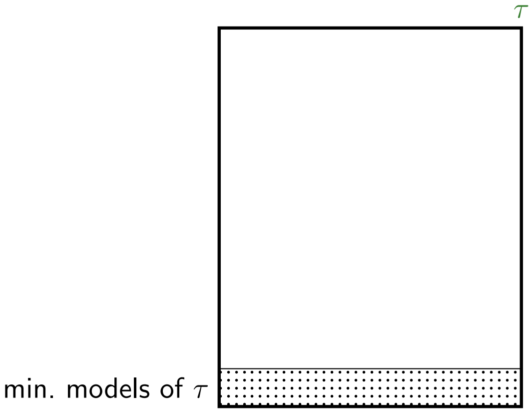
- We are going to use a SAT solver — a tool that decides whether a formula is satisfiable or not.
- We are going to construct a propositional formula expressing $\tau \models_{\min} \psi$.

Plan of Attack

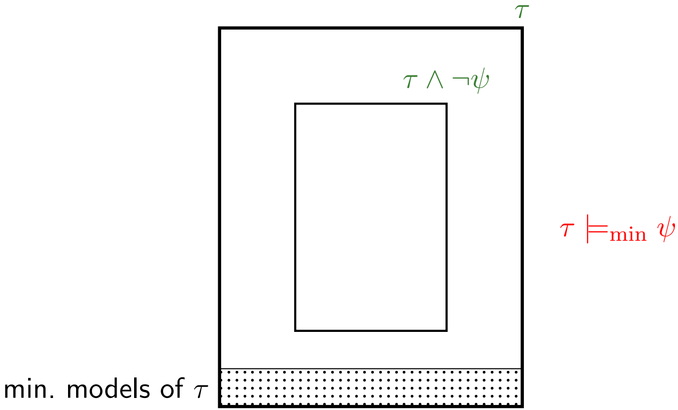
$$\tau \models_{\min} \psi$$

- We are going to use a SAT solver — a tool that decides whether a formula is satisfiable or not.
- We are going to construct a propositional formula expressing $\tau \models_{\min} \psi$.
- We are going to use abstraction to mitigate the size of the formula.

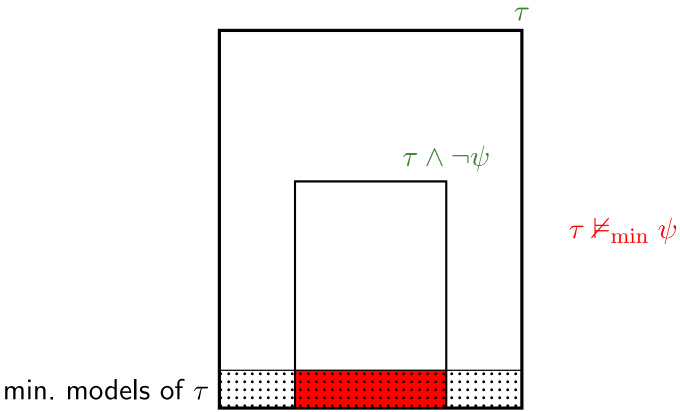
Propositional Form of $\models_{\min}$



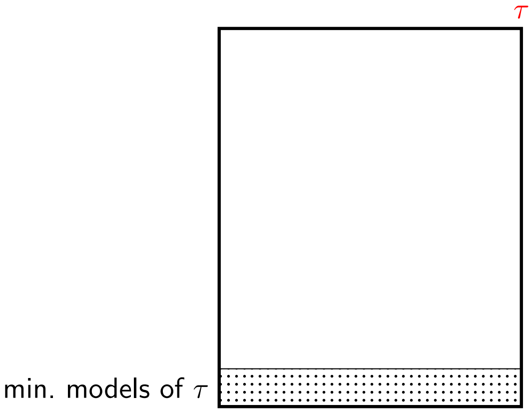
Propositional Form of $\models_{\min}$



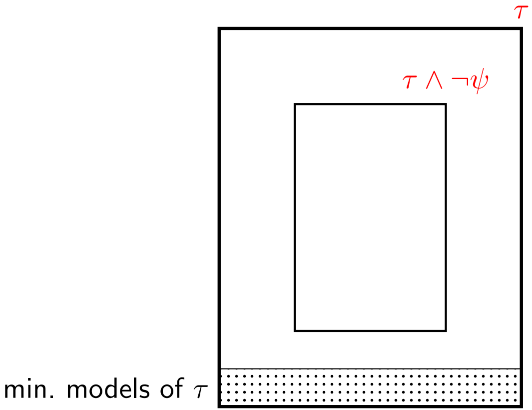
Propositional Form of $\models_{\min}$



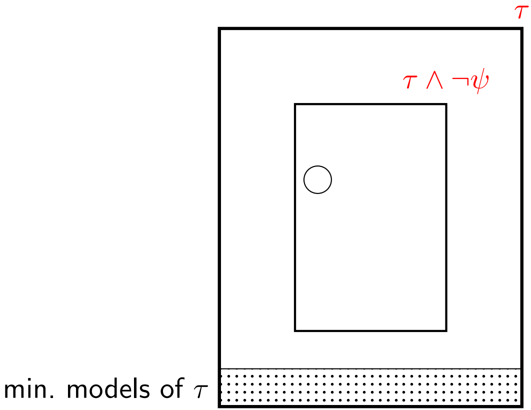
Propositional Form of $\models_{\min}$



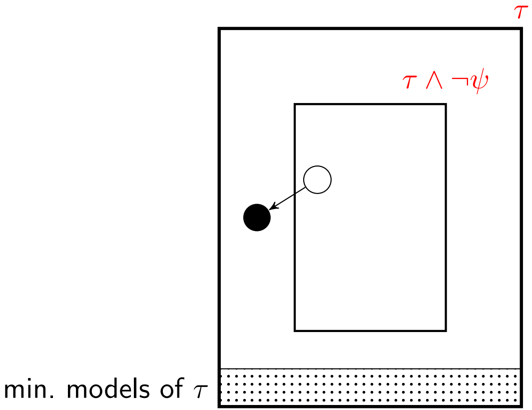
Propositional Form of $\models_{\min}$



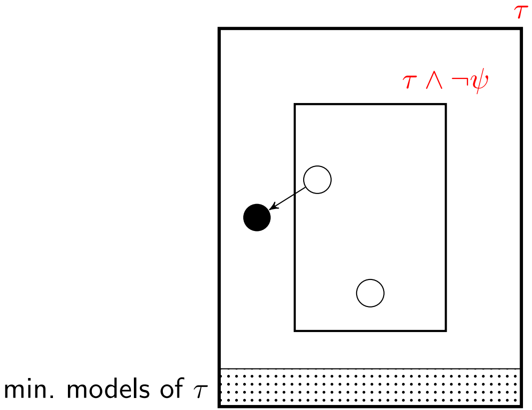
Propositional Form of $\models_{\min}$



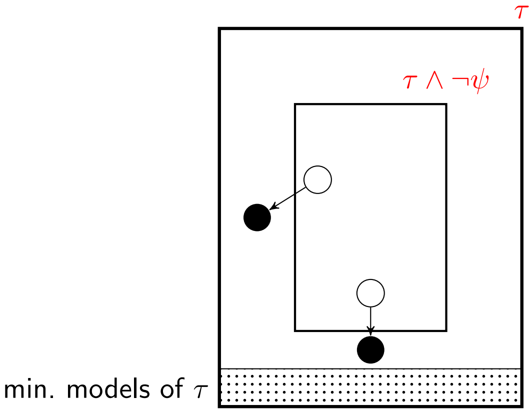
Propositional Form of $\models_{\min}$



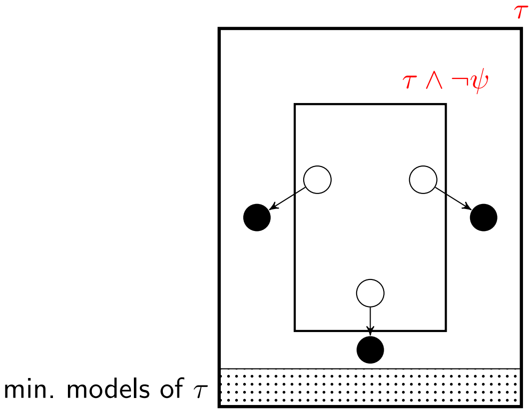
Propositional Form of $\models_{\min}$



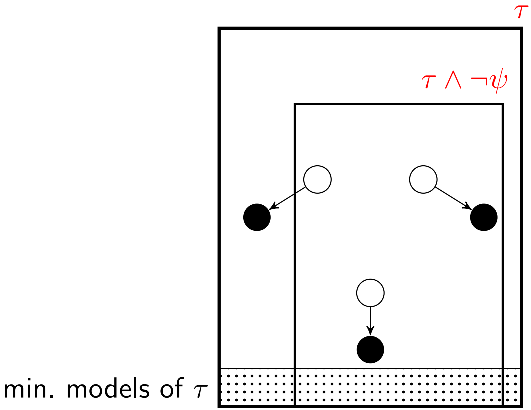
Propositional Form of $\models_{\min}$



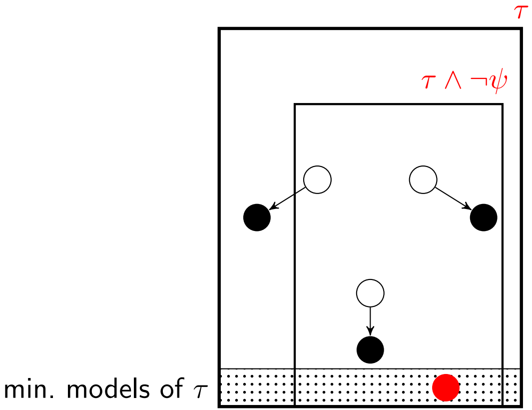
Propositional Form of $\models_{\min}$



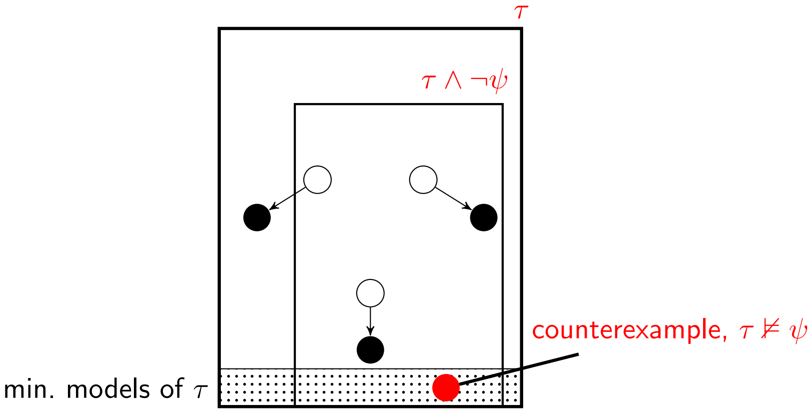
Propositional Form of $\models_{\min}$



Propositional Form of $\models_{\min}$



Propositional Form of $\models_{\min}$



Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$
- there exists a model ν' of ψ s.t. $\nu' < \nu$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$
- there exists a model ν' of ψ s.t. $\nu' < \nu$

- For $\nu \models \tau \wedge \neg\psi$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$
- there exists a model ν' of ψ s.t. $\nu' < \nu$

- For $\nu \models \tau \wedge \neg\psi$
- there exists a set of variables S s.t. $\nu \models \tau[S \rightarrow 0]$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$
- there exists a model ν' of ψ s.t. $\nu' < \nu$

- For $\nu \models \tau \wedge \neg\psi$
- there exists a set of variables S s.t. $\nu \models \tau[S \to 0]$

$$\tau = (x \vee y)$$
$$\psi = \neg(x \wedge y)$$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$
- there exists a model ν' of ψ s.t. $\nu' < \nu$

- For $\nu \models \tau \wedge \neg\psi$
- there exists a set of variables S s.t. $\nu \models \tau[S \rightarrow 0]$

$$\tau = (x \vee y)$$
$$\psi = \neg(x \wedge y)$$

$$\{x = 1, y = 1\} \models (x \vee y) \wedge (x \wedge y)$$

Propositional Form of $\models_{\min}$

- To prove $\tau \models_{\min} \psi$
- we show that for any model ν of $\tau \wedge \neg\psi$
- there exists a model ν' of ψ s.t. $\nu' < \nu$

- For $\nu \models \tau \wedge \neg\psi$
- there exists a set of variables S s.t. $\nu \models \tau[S \to 0]$

$$\tau = (x \vee y)$$
$$\psi = \neg(x \wedge y)$$

$$\{x = 1, y = 1\} \models (x \vee y) \wedge (x \wedge y)$$
$$\{x = 1, y = 1\} \models (0 \vee y)$$

Propositional Form of $\models_{\min}$

$$\tau \models_{\min} \psi$$

Propositional Form of $\models_{\min}$

$$\tau \models_{\min} \psi$$

iff

$$(\forall \nu) .\ (\nu \models \tau \wedge \neg\psi) \Rightarrow (\exists S \in \wp(V)) .\ \nu \models \tau[S \rightarrow 0]$$

Propositional Form of $\models_{\min}$

$$\tau \models_{\min} \psi$$

iff

$$(\forall \nu) . \ (\nu \models \tau \wedge \neg\psi) \Rightarrow (\exists S \in \wp(V)) . \nu \models \tau[S \to 0] \wedge \exists_{x \in S} . \nu(x) = 1$$

Propositional Form of $\models_{\min}$

$$\tau \models_{\min} \psi$$

iff

$$(\forall \nu) \,.\, (\nu \models \tau \wedge \neg\psi) \Rightarrow (\exists S \in \wp(V)) \,.\, \nu \models \tau[S \to 0] \wedge \exists_{x \in S} \,.\, \nu(x) = 1$$

iff

$$\text{TAUT: } \tau \wedge \neg\psi \Rightarrow \bigvee_{S \in \wp(V)} \left(\tau[S \to 0] \wedge \bigvee_{x \in S} x \right)$$

Propositional Form of $\models_{\min}$

$$\tau \models_{\min} \psi$$

iff

$$(\forall \nu) . (\nu \models \tau \wedge \neg\psi) \Rightarrow (\exists S \in \wp(V)) . \nu \models \tau[S \to 0] \wedge \exists_{x \in S} . \nu(x) = 1$$

iff

$$\text{TAUT: } \tau \wedge \neg\psi \Rightarrow \bigvee_{S \in \wp(V)} \left(\tau[S \to 0] \wedge \bigvee_{x \in S} x \right)$$

iff

$$\text{UNSAT: } \tau \wedge \neg\psi \wedge \bigwedge_{S \in \wp(V)} \left(\neg\tau[S \to 0] \vee \bigwedge_{x \in S} \neg x \right)$$

Abstraction

Abstract

$$\tau \wedge \neg\psi \wedge \bigwedge_{S \in \wp(V)} \left(\neg\tau[S \to 0] \vee \bigwedge_{x \in S} \neg x \right)$$

Abstraction

Abstract

$$\tau \wedge \neg\psi \wedge \bigwedge_{S\in\wp(V)} \left(\neg\tau[S \to 0] \vee \bigwedge_{x\in S} \neg x\right)$$

as

$$\tau \wedge \neg\psi \wedge \bigwedge_{S\in W} \left(\neg\tau[S \to 0] \vee \bigwedge_{x\in S} \neg x\right)$$

for some $W \in \wp(\wp(V))$

Abstraction

Abstract

$$\tau \wedge \neg\psi \wedge \bigwedge_{S \in \wp(V)} \left(\neg\tau[S \to 0] \vee \bigwedge_{x \in S} \neg x \right)$$

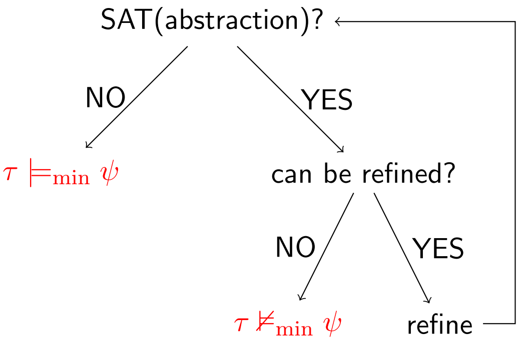
as

$$\tau \wedge \neg\psi \wedge \bigwedge_{S \in W} \left(\neg\tau[S \to 0] \vee \bigwedge_{x \in S} \neg x \right)$$

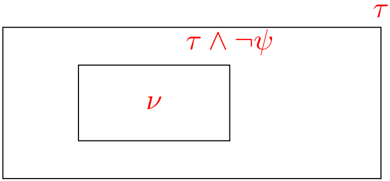
for some $W \in \wp(\wp(V))$

The abstraction is weaker. If the abstraction is shown UNSAT, the original formula is UNSAT.

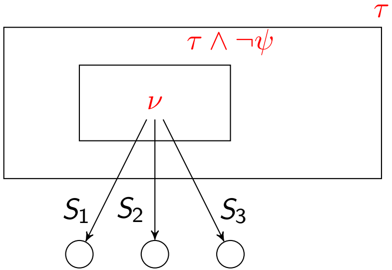
Abstraction-Refinement Loop



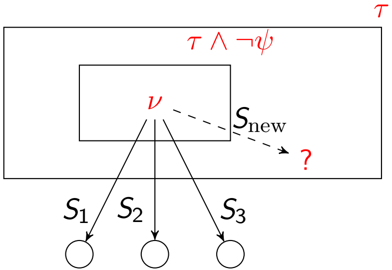
Refinement Test



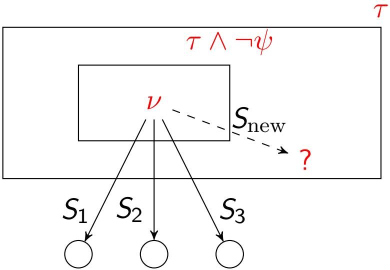
Refinement Test



Refinement Test

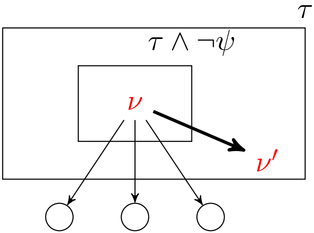


Refinement Test

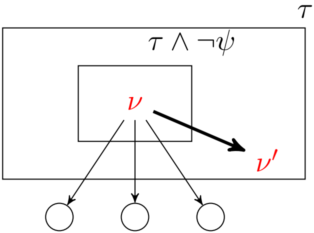


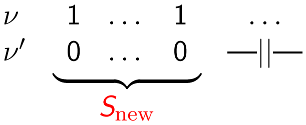
$$\mathsf{SAT}\left(\tau \wedge \bigwedge_{\nu(x)=0} \neg x \wedge \bigvee_{\nu(x)=1} \neg x\right)$$

Refinement

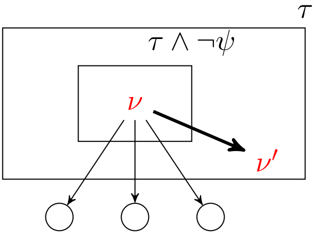


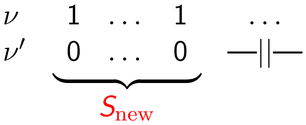
Refinement





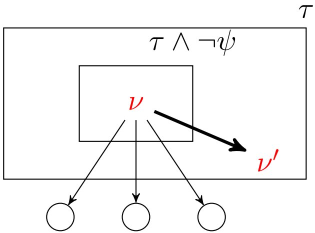
Refinement

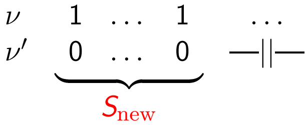




$$W' = W \cup \{S_{\text{new}}\}$$

Refinement

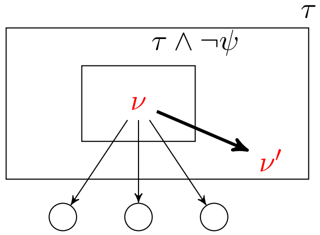




$$W' = W \cup \{S_{\mathrm{new}}\}$$

$$\tau \wedge \neg\psi \quad \wedge \quad \bigwedge_{S \in W} \left(\neg\tau[S \to 0] \vee \bigwedge_{x \in S} \neg x\right)$$

Refinement



$$\begin{array}{lccccc} \nu & 1 & \ldots & 1 & & \ldots \\ \nu' & 0 & \ldots & 0 & & \text{—||—} \end{array}$$

$$\underbrace{0 \ \ldots \ 0}_{S_{\mathrm{new}}}$$

$$W' = W \cup \{S_{\mathrm{new}}\}$$

$$\tau \wedge \neg\psi \quad \wedge \quad \bigwedge_{S \in W} \left(\neg\tau[S \to 0] \vee \bigwedge_{x \in S} \neg x\right)$$

$$\wedge \quad \neg\tau[S_{\mathrm{new}} \to 0] \vee \bigwedge_{x \in S_{\mathrm{new}}} \neg x$$

Algorithm

$\omega \leftarrow \tau \wedge \neg\psi$

while true **do**

    $(\text{outc}_1, \nu) \leftarrow \texttt{SAT}(\omega)$

    if $\text{outc}_1 = \text{false}$ **then**

        return true   `// no counterexample was found`

    end

    `// refine test`

    $(\text{outc}_2, \nu') \leftarrow \texttt{SAT}\left(\tau \wedge \bigwedge_{\nu(x)=0} \neg x \wedge \bigvee_{\nu(x)=1} \neg x\right)$

    if $\text{outc}_2 = \text{false}$ **then**   `//` ν `is minimal`

        return false   `// abstraction cannot be refined`

    end

    `// refine`

    $S \leftarrow \{x \in V \mid \nu(x) = 1 \wedge \nu'(x) = 0\}$

    $\omega \leftarrow \omega \wedge (\neg\tau[S \mapsto 0] \vee \bigwedge_{x \in S} \neg x)$

end

Experimental evaluation

		Our Approach		circ2dlp+gnt	
	tests	solved	time[s]	solved	time[s]
e-shop	174	174	2.1	95	2.4
BerkeleyDB	30	30	0.9	30	< 0.1
model-transf	41	41	1.1	35	2.8
SAT2009	15	3	7.6	2	2.5
TOTAL		248		162	

- We also tried a QBF solver but that has solved **none** of the 260 instances within the time limit.

Summary

- we tackled the problem of entailment in propositional circumscription using a SAT solver
- in order to do so, they express the problem as a propositional formula
- such formula is exponential a large
- we construct an abstraction of the formula, which enables us to decide the problem without constructing exponentially large one
- we are able to decide instances for which it was previously not possible