Computing Short SAT Implicants via Ising/QUBO encodings

Giuseppe Spallitta, Leonardo Duenas-Osorio, Moshe Y. Vardi

23rd July, 2026

The 32nd International Conference on Principles and Practice of Constraint Programming

Implicants: partial satisfying assignments

Definition

Given a formula F, an **implicant** is a partial assignment π such that $\pi \models F$.
Equivalently, every complete assignment extending π satisfies F.

Example

$$F = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$$

A complete assignment satisfying F is:

$$x_1 = \top, \qquad x_2 = \bot, \qquad x_3 = \bot$$

Implicants: partial satisfying assignments

Definition

Given a formula F, an **implicant** is a partial assignment π such that $\pi \models F$.
Equivalently, every complete assignment extending π satisfies F.

Example

$$F = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$$

A complete assignment satisfying F is:

$$x_1 = \top, \qquad x_2 = \bot, \qquad x_3 = \bot$$

But a shorter partial assignment satisfying F is:

$$\pi = \{x_1 = \top\}$$

So an implicant is a satisfying assignment with don't-care variables.

Why short implicants matter

$$F = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$$

The short implicant

$$x_1 = \top$$

represents all these satisfying assignments:

x_1	x_2	x_3
$\top$	$\top$	$\top$
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\top$
$\top$	$\bot$	$\bot$

Message

A short implicant is a compact explanation: the **shorter**, the **better**.

Prime vs. minimum-cardinality implicants

Same formula

$$F = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$$

Prime implicant

$$\pi_{\text{prime}} = \{x_2 = \top,\ x_3 = \top\}$$

$$|\pi_{\text{prime}}| = 2$$

If I remove a literal, it is no longer implicant:

$$\{x_2 = \top\} \not\models F, \qquad \{x_3 = \top\} \not\models F.$$

Cannot be shrunk anymore.

Locally minimal, but not shortest.

Minimum-cardinality implicant

$$\pi_{\min} = \{x_1 = \top\}$$

$$|\pi_{\min}| = 1$$

It also implies F:

$$\{x_1 = \top\} \Rightarrow F.$$

No implicant can have size smaller than 1.

Globally shortest.

Computing short implicants is hard

Classical approaches

- **SAT-based shrinking:** iteratively remove literals while preserving implication[1].
  - As hard as finding a solution (NP-hard) + shrinking algorithms.
  - No guarantee to find the shortest; algorithms are heavily dependent on variable order.
- **Exact minimization:** search for a globally shortest implicant, often through PB/MaxSAT-style optimization integrated in SAT reasoning[2].
  - Finding one implicant of minimum cardinality is way harder than just a prime implicant due to the global size minimization.

[1] D. Déharbe, P. Fontaine, D. Le Berre, and B. Mazure (2013). *Computing prime implicants*. In: *2013 Formal Methods in Computer-Aided Design*

[2] D. Fried, A. Nadel, and Y. Shalmon (2023). *AllSAT for Combinational Circuits*. In: *26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023)*

What if implicants were not **computed**,
but **purely optimized**?

Ising machines as optimization framework

Ising machines are optimization platforms (HW or SW, including **quantum annealers**) that aim to **minimize** a quadratic penalty function over n variables:

$$H_{\text{QUBO}}(z) = \sum_i h_i z_i + \sum_{i<j} J_{ij} z_i z_j, \qquad z_i \in \{0, 1\}$$

$$H_{\text{Ising}}(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j, \qquad s_i \in \{-1, +1\}$$

- h: biases
- J: couplings
- z/s: binary variables or spins

Typical application domains [3]

- Combinatorial optimization problems
- Scheduling tasks
- **SAT-to-Ising encodings**[4]

A. Lucas (2014). *Ising formulations of many NP problems*. In: *Frontiers in Physics*

4 J. Nüßlein, S. Zielinski, T. Gabor, C. Linnhoff-Popien, and S. Feld (2023). *Solving (Max) 3-SAT via Quadratic Unconstrained Binary Optimization*. In: *Computational Science – ICCS 2023*

What is a SAT-to-Ising encoding?

The idea is to transform a SAT formula F into an Ising function H such that:

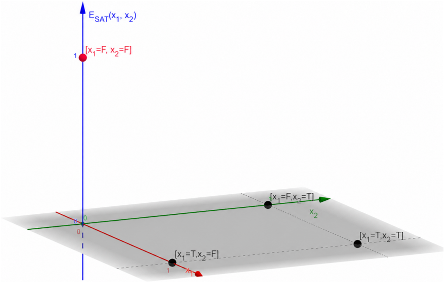
$$\text{satisfying assignments of } F \iff \text{minimum-energy states of } H$$

Example

$$F = (x_1 \vee x_2)$$

$$\Downarrow$$

$$H(z) = (1 - z_1)(1 - z_2)$$



Key idea

Solving SAT becomes an optimization problem: find the spin configuration with minimum energy.
It can be useful for problems which are hard for SAT solvers[5].

[5] J. Ding, G. Spallitta, and R. Sebastiani (2024). *Effective prime factorization via quantum annealing by modular locally-structured embedding*. In: *Scientific Reports*

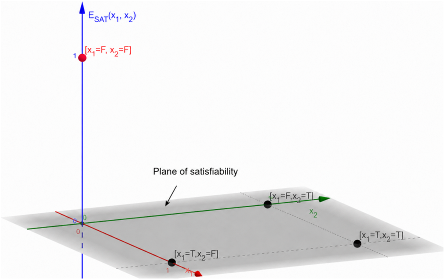
Main issue: Ising ground states are always total

SAT-to-Ising encodings usually keep a **one-to-one** mapping:

$$\text{Boolean } x_i \begin{cases} \top \text{ (True)} \\ \bot \text{ (False)} \end{cases} \Leftrightarrow \text{Qubits } z_i \begin{cases} 1 \\ 0 \end{cases}$$

Consequence

- Ground states are $(z_1, z_2) \in \{(1,1),(1,0),(0,1)\}$, which decode to total satisfying assignments $(x_1, x_2) \in \{(\top,\top),(\top,\bot),(\bot,\top)\}$.
- There is no way to output "$x_1 =$ true, $x_2 =$ don't care"!
- This is natural; **the annealer will assign a value $\{0,1\}$ to each qubit** in the encoding.



Research question

Can we integrate the concept of **“don’t care”** directly into the Ising model so that energy minimization returns *implicants*?

Research question

Can we integrate the concept of **"don't care"** directly into the Ising model so that energy minimization returns *implicants*?

Yes, via a **three valued semantic inside Ising model**!.

Core idea: from two-valued to three-valued semantics

Standard qubits represent $\{\top, \bot\}$. We want $\{\top, \bot, *\}$ where $*$ means *unassigned*.

Represent each Boolean variable A_i with two qubits (p_i, n_i) and decode as:

p_i	n_i	meaning	state
1	0	$A_i = \top$	assigned
0	1	$A_i = \bot$	assigned
0	0	$A_i = *$	**unassigned**
1	1	inconsistent	forbidden

Core idea: from two valued to three valued semantics

We can then generate novel SAT-to-Ising encodings:

$$H_{SAT2.0}(\mathbf{p}, \mathbf{n}) \;=$$

Core idea: from two valued to three valued semantics

We can then generate novel SAT-to-Ising encodings:

$$H_{SAT2.0}(\mathbf{p}, \mathbf{n}) \; = \; \lambda \cdot H_{\mathsf{SAT}}(\mathbf{p}, \mathbf{n})$$

- The λ-term (SAT term) pushes the optimizer to satisfy each clause of the formula F.
  - We can use state-of-the-art SAT-to-Ising encodings, replacing a single qubit per variable with three-valued semantics.

Core idea: from two valued to three valued semantics

We can then generate novel SAT-to-Ising encodings:

$$H_{SAT2.0}(\mathbf{p},\mathbf{n}) \;=\; \lambda \cdot H_{\mathsf{SAT}}(\mathbf{p},\mathbf{n}) \;+\; M\sum_i p_i n_i$$

- The λ-term (SAT term) pushes the optimizer to satisfy each clause of the formula F.
  - We can use state-of-the-art SAT-to-Ising encodings, replacing a single qubit per variable with three-valued semantics.
- The M-term (consistency term) pushes the optimizer to avoid forbidden semantics.

Core idea: from two valued to three valued semantics

We can then generate novel SAT-to-Ising encodings:

$$H_{SAT2.0}(\mathbf{p},\mathbf{n}) \;=\; \lambda \cdot H_{\mathsf{SAT}}(\mathbf{p},\mathbf{n}) \;+\; M\sum_i p_i n_i \;+\; \gamma\sum_i (p_i + n_i).$$

- The λ-term (SAT term) pushes the optimizer to satisfy each clause of the formula F.
  - We can use state-of-the-art SAT-to-Ising encodings, replacing a single qubit per variable with three-valued semantics.
- The M-term (consistency term) pushes the optimizer to avoid forbidden semantics.
- The γ-term (sparsity term) pushes the optimizer to set $(p_i, n_i) = (0, 0)$ unless forced by satisfiability.

Main theoretical guarantee (see proof in paper)

Main theorem

Let F be a CNF formula and let

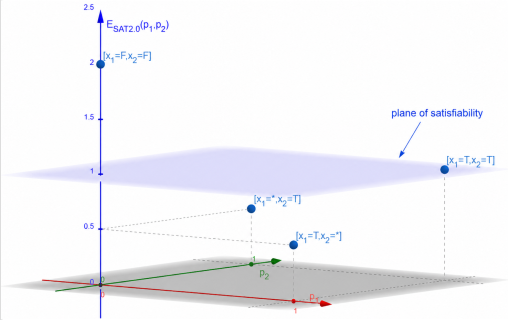
$$H_{SAT2.0}(p,n) = \lambda H_{\text{SAT}}(p,n) + M H_{\text{inc}}(p,n) + \gamma H_{\text{card}}(p,n).$$

If the weights satisfy suitable separation constraints:

$$\lambda > n\gamma, M > n\gamma$$

then:

- **All Ising models with energy lower than or equal to $n\gamma$ satisfy F.**
- Every global minimum of $H_{SAT2.0}$ is an implicant of minimum cardinality.



Main theoretical guarantee (see proof in paper)

Main theorem

Let F be a CNF formula and let

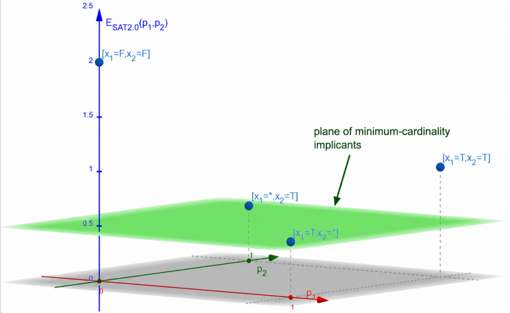
$$H_{SAT2.0}(p,n) = \lambda H_{\text{SAT}}(p,n) + M H_{\text{inc}}(p,n) + \gamma H_{\text{card}}(p,n)$$

If the weights satisfy suitable separation constraints:

$$\lambda > n\gamma, M > n\gamma$$

then:

- All Ising models with energy lower than or equal to $n\gamma$ satisfy F.
- **Every global minimum of $H_{SAT2.0}$ is a satisfying partial assignment of minimum cardinality.**



One H_{SAT} variant: Implicant shrinking from a SAT assignment

Assume we already have a satisfying assignment:

$$\alpha = (x_1 = \top,\ x_2 = \bot,\ x_3 = \top,\ \ldots).$$

Implicant shrinking

Starting from a complete satisfying assignment, we try to remove as many assigned literals as possible while still guaranteeing satisfiability.

$$(x_1 = \top,\ x_2 = \bot,\ x_3 = \top, \ldots) \quad \rightsquigarrow \quad (x_1 = \top,\ x_2 = *,\ x_3 = \top, \ldots)$$

The goal is to keep only the literals that are really needed.

Restricting $H_{SAT2.0}$ for implicant shrinking

Given a satisfying assignment α, we restrict the spin variables so that the optimizer can only keep literals consistent with α.

Fix incompatible polarities

If $\alpha(x_i) = \top$, set every occurrence of n_i in $H_{\mathrm{SAT2.0}}$ to 0.
If $\alpha(x_i) = \bot$, set every occurrence of p_i in $H_{\mathrm{SAT2.0}}$ to 0.

The optimizer can then choose, for each variable:

$$x_i = \alpha(x_i) \quad \text{or} \quad x_i = *.$$

Main theoretical guarantee (see proof in paper)

The Ising minimum gives the **shortest implicant contained in** α.

Another H_{SAT} variant: Projected implicants

Sometimes, not all variables are equally important. We care only about a set of relevant variables

$$R \subseteq \mathrm{Vars}(F),$$

while the remaining variables

$$I = \mathrm{Vars}(F) \setminus R$$

are hidden.

Projected implicant

A partial assignment π over R is a projected implicant of F if

$$\pi \models \exists I.\, F.$$

Equivalently, every assignment ρ to the remaining relevant variables can be extended with some assignment σ to the hidden variables so that F is satisfied:

$$\forall \rho : R \setminus \mathrm{vars}(\pi) \to \{\bot, \top\}, \qquad \exists \sigma : I \to \{\bot, \top\}, \qquad \pi \wedge \rho \wedge \sigma \models F.$$

Projected implicants in the $H_{SAT2.0}$ encoding

The encoding still enforces satisfiability over the full formula.

What stays the same

SAT and consistency terms are applied to all variables:

$$\lambda H_{\text{SAT}} + M H_{\text{inc}}.$$

What changes

The sparsity term is applied only to projected variables:

$$\gamma \sum_{x_i \in R} (p_i + n_i).$$

Thus:

$$H^{P}_{SAT2.0} = \lambda H_{\text{SAT}} + M H_{\text{inc}} + \gamma \sum_{x_i \in R} (p_i + n_i).$$

Experimenting with the encoding

We evaluate two Ising search strategies on two SAT-to-Ising tasks, resulting in four configurations:

	Basic	Iter
Full	Direct search of short implicant over F using one SA run	Direct search of short implicant over F through iterative refinement
Shrink	Shrink a given satisfying assignment using one SA run	Shrink a given satisfying assignment through iterative refinement

Search strategies

- Basic: one-shot simulated annealing run with 1000 samples.
- Iter: start from the best SA sample, freeze its polarity pattern, and re-run SA on the restricted model to push additional variables into the don't-care state [6].

[6] J. Ding, G. Spallitta, and R. Sebastiani (2024). *Effective prime factorization via quantum annealing by modular locally-structured embedding*. In: *Scientific Reports*

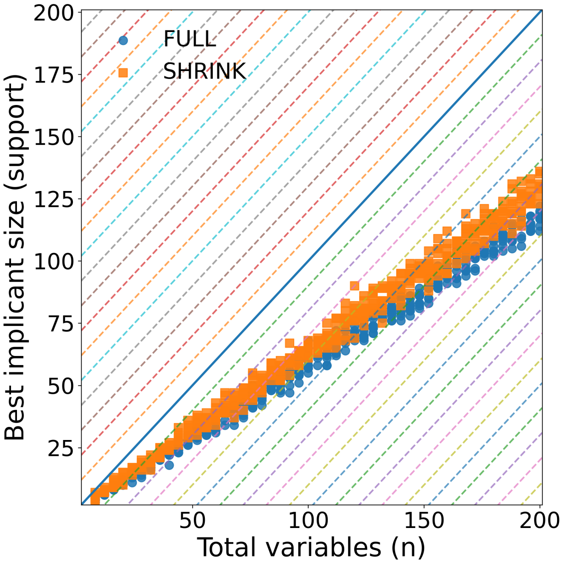
Preliminary results: does the encoding work?

What are we seeing?

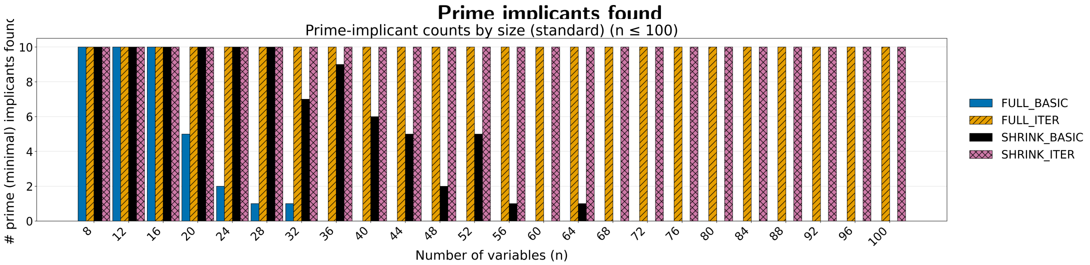
- Random 3-SAT instances from model-enumeration benchmarks.
- For each instance, we run simulated annealing and keep the **shortest implicant found** among 1000 samples (i..w. we use BASIC)
- Each point compares the formula size n with the best implicant size found.

Key findings

- For all instances, the best sample is a satisfying assignment.
- Around 30% of variables are left unassigned in all instances.
- Implicants from SHRINK are slightly longer than the ones from FULL because we have fewer degrees of freedom.



Random 3-SAT minimization: preliminary results



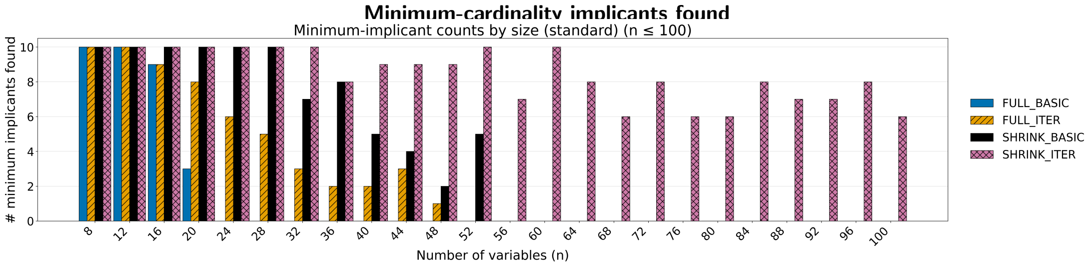
What are we seeing?

- Same random 3-SAT instances grouped by number of variables n (x-axis).
- Bars count how often each configuration finds the target implicant. 10 instances for each problem size n.

Key findings

- ITER constantly retrieves prime implicants for both tasks.
- BASIC implicant shrinking is more effective than full implicant retrieval.
- BASIC performances degrade fast

Random 3-SAT minimization: preliminary results



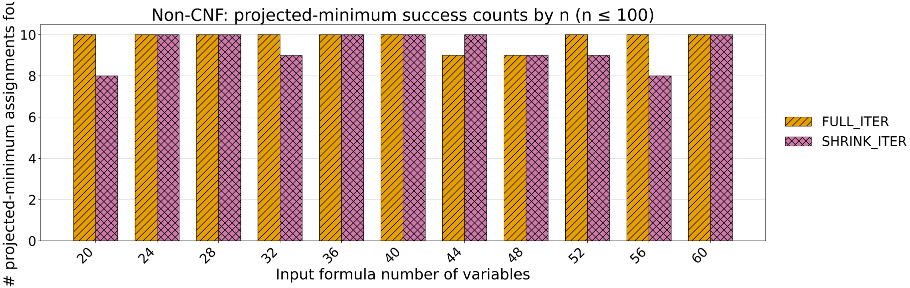
What are we seeing?

- Same random 3-SAT instances grouped by number of variables n (x-axis).
- Bars count how often each configuration finds the target implicant. 10 instances for each problem size n.

Key findings

- ITER on implicant shrinking is the only one that finds the minimum cardinality on large instances.
- It is more difficult to achieve minimum-cardinality.
- The tuning of simulated annealing is basilar, so it is expected (better engineering for future work).

Non-CNF formulas: projected minimization



What are we seeing?

- Non-CNF formulas from recent model-enumeration benchmarks, encoded into CNF through the NNF+PG transformation[7].
- Bars count how often each configuration finds the target implicant. 10 instances for each problem size n.

Key findings

- ITER consistently retrieves the minimum cardinality under projection for both tasks.
- Despite additional hidden variables (around 60% of the entire set of variables after transformation), we perform really well (!)

[7] G. Masina, G. Spallitta, and R. Sebastiani (2023). *On CNF Conversion for Disjoint SAT Enumeration*. In: *26th International Conference on Theory and Applications of Satisfiability Testing, SAT 2023, Alghero, Italy, July 4-8, 2023*

Summary and future work

Summary

- We discuss a **three-valued Ising/QUBO encoding**:

$$x_i = \top, \qquad x_i = \bot, \qquad x_i = *.$$

- Ground states can represent **partial assignments**, not only total SAT assignments.
- With suitable penalties, minima correspond to **minimum-cardinality implicants**.
- Preliminary results show that simple simulated annealing already finds implicants shorter than full assignments.

Future work

- **Reduce spins and couplings** by simplifying the encoding whenever possible.
- Test on stronger **Ising-machine solvers**, focusing on efficiency and scalability.
- Run experiments on **real quantum annealers**, studying embedding, penalty scaling, and hardware noise.

Thanks for listening!

Questions?

Paper: Computing Short SAT Implicants via SAT/QUBO Encodings

Scan the QR code to access the paper.

arXiv:
2605.07017

Backup: Ising $\leftrightarrow$ QUBO Transformation

Ising formulation

$$H_{\text{Ising}}(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j,$$

$$s_i \in \{-1, +1\}.$$

QUBO formulation

$$H_{\text{QUBO}}(z) = \sum_i a_i z_i + \sum_{i<j} b_{ij} z_i z_j,$$

$$z_i \in \{0, 1\}.$$

Variable transformation:

$$z_i = \frac{s_i + 1}{2}, \qquad s_i = 2z_i - 1.$$

Equivalent optimization problem:

$$\arg\min_s H_{\text{Ising}}(s) = \arg\min_z H_{\text{QUBO}}(z).$$

Coefficient mapping

$$b_{ij} = 4J_{ij}, \qquad a_i = 2h_i - 2\sum_{j\neq i} J_{ij}, \qquad c = -\sum_i h_i + \sum_{i<j} J_{ij} \text{ (constant)}.$$

What about clauses with more than two literals?

Three-literal clause

$$F = (x_1 \vee \neg x_2 \vee x_3)$$

Using polarity variables:

$$x_1 \mapsto p_1, \qquad \neg x_2 \mapsto n_2, \qquad x_3 \mapsto p_3.$$

The clause is violated only when no literal is selected:

$$p_1 = 0, \qquad n_2 = 0, \qquad p_3 = 0.$$

So the natural clause penalty is:

$$E_C = \lambda(1 - p_1)(1 - n_2)(1 - p_3).$$

Issue

This term is cubic, but Ising/QUBO models must be quadratic.

Quadratizing a three-literal clause

For $F = (x_1 \lor \neg x_2 \lor x_3)$, the cubic penalty is:

$$\lambda(1-p_1)(1-n_2)(1-p_3).$$

Introduce one auxiliary spin a to represent:

$$a = (1-p_1)(1-n_2).$$

The AND relation is enforced by the penalty:

$$Q_\land(a,u,v) = \lambda(uv - 2au - 2av + 3a),$$

where $u = 1 - p_1$ and $v = 1 - n_2$.
Then the clause penalty becomes quadratic:

$$\lambda a(1-p_3).$$

AND gadget: $a = uv$

u	v	a	$Q_\land(a,u,v)$
0	0	0	0
0	0	1	3λ
0	1	0	0
0	1	1	λ
1	0	0	0
1	0	1	λ
1	1	0	λ
1	1	1	0

Clause penalty

a	p_3	$\lambda a(1-p_3)$
0	0	0
0	1	0
1	0	λ
1	1	0

Result

A clause of size > 2 requires auxiliary variables to keep the Ising/QUBO encoding quadratic.

Projection meaning in the Ising model

Projected implicant

π is a projected implicant of F over P if it entails the existential projection of F:

$$\pi \models \exists H.F.$$

Equivalently:

$$\forall \rho : P \setminus \mathrm{vars}(\pi) \to \{\bot, \top\} \quad \exists \sigma : H \to \{\bot, \top\} \quad (\pi \wedge \rho \wedge \sigma) \models F.$$

Important limitation

An Ising solution gives one full spin configuration. Therefore, hidden variables are fixed once, not re-chosen for each visible completion.

So the Ising naturally enforces the stronger condition:

$$\exists \sigma^\star : H \to \{\bot, \top\} \quad \forall \rho : P \setminus \mathrm{vars}(\pi) \to \{\bot, \top\} \quad (\pi \wedge \rho \wedge \sigma^\star) \models F.$$

Projection: extending the guarantee beyond CNF

For real-world non-CNF formulas, we use the pipeline for effective short assignment generation:

$$F \quad \xrightarrow{\text{NNF+PG}} \quad F_{\text{CNF}}.$$

This transformation has the key property [8] that, for every short partial assignment μ over the original variables:

$$\mu \models F \quad \Rightarrow \quad \exists\sigma \text{ such that } \mu \wedge \sigma \models F_{\text{CNF}}.$$

It also uses the three-valued semantic, thus we can directly feed F_{CNF} to our algorithm.

Consequence

Under this condition, projection behaves like the ordinary implicant case: each projected implicant has one hidden auxiliary completion σ.

[8] G. Masina, G. Spallitta, and R. Sebastiani (2025). *On CNF Conversion for SAT and SMT Enumeration.* In: *J. Artif. Intell. Res.*