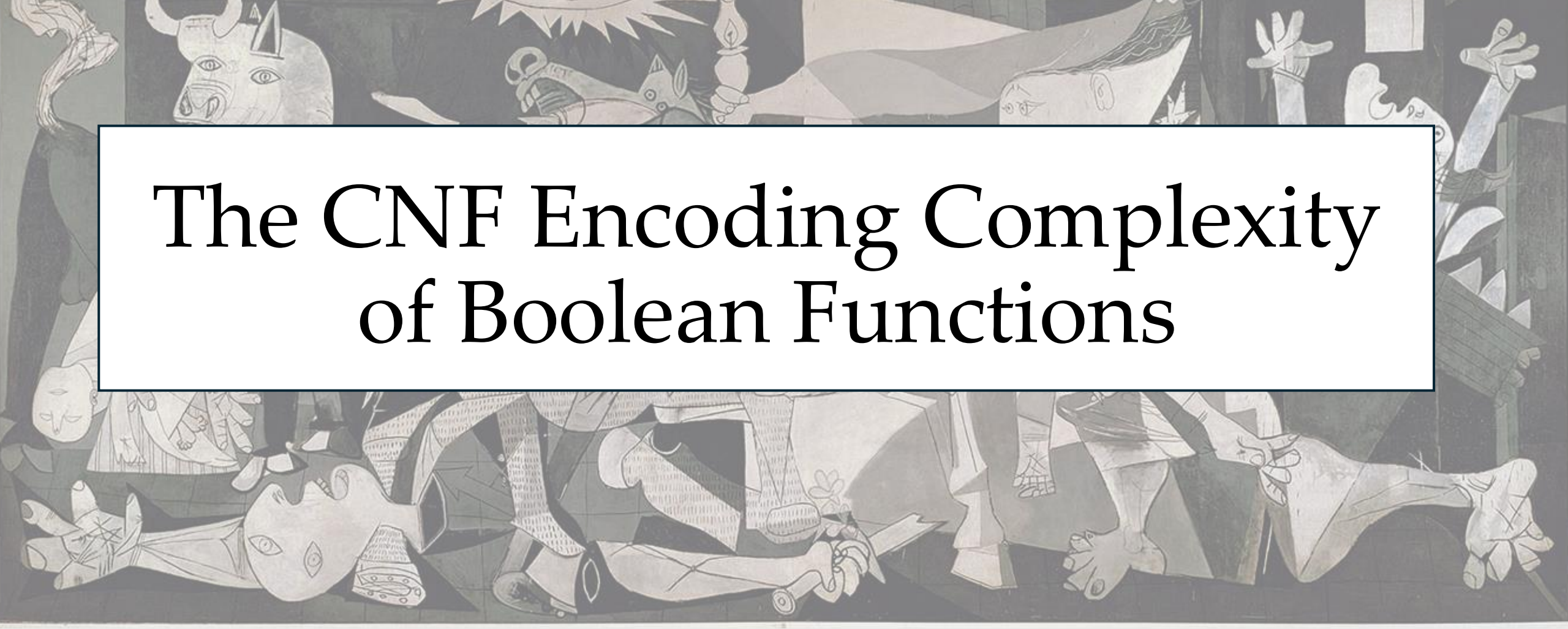


The CNF Encoding Complexity of Boolean Functions

Bernardo Subercaseaux Carnegie Mellon University

joint work with Andrew Krapivin and Ben Przybocki

CNF Formula

$$F := (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5})$$
$$\wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5})$$
$$\wedge (\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5})$$
$$\wedge (\overline{x_4} \vee \overline{x_5})$$

CNF Formula

$$F := (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5}) \wedge (\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_4} \vee \overline{x_5})$$

$$\mathrm{AMO}\colon \{0,1\}^5 \to \{0,1\}$$

$$\mathrm{AMO}(x_1, x_2, x_3, x_4, x_5) \equiv \left(\sum_{i=1}^{5} x_i \leq 1 \right)$$

Boolean function

CNF Formula

$$F := (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5}) \wedge (\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_4} \vee \overline{x_5})$$

Encodes

$$\mathrm{AMO}\colon \{0,1\}^5 \to \{0,1\}$$

$$\mathrm{AMO}(x_1, x_2, x_3, x_4, x_5) \equiv \left(\sum_{i=1}^{5} x_i \leq 1 \right)$$

Boolean function

CNF Formula

$$F := (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5}) \wedge (\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_4} \vee \overline{x_5})$$

$\binom{5}{2} = 10$ clauses

Encodes

$$\text{AMO}: \{0,1\}^5 \to \{0,1\}$$

$$\text{AMO}(x_1, x_2, x_3, x_4, x_5) \equiv \left(\sum_{i=1}^{5} x_i \leq 1 \right)$$

Boolean function

CNF Formula

$$F := (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5}) \wedge (\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_4} \vee \overline{x_5})$$

$\binom{5}{2} = 10$ clauses

Can we do better?

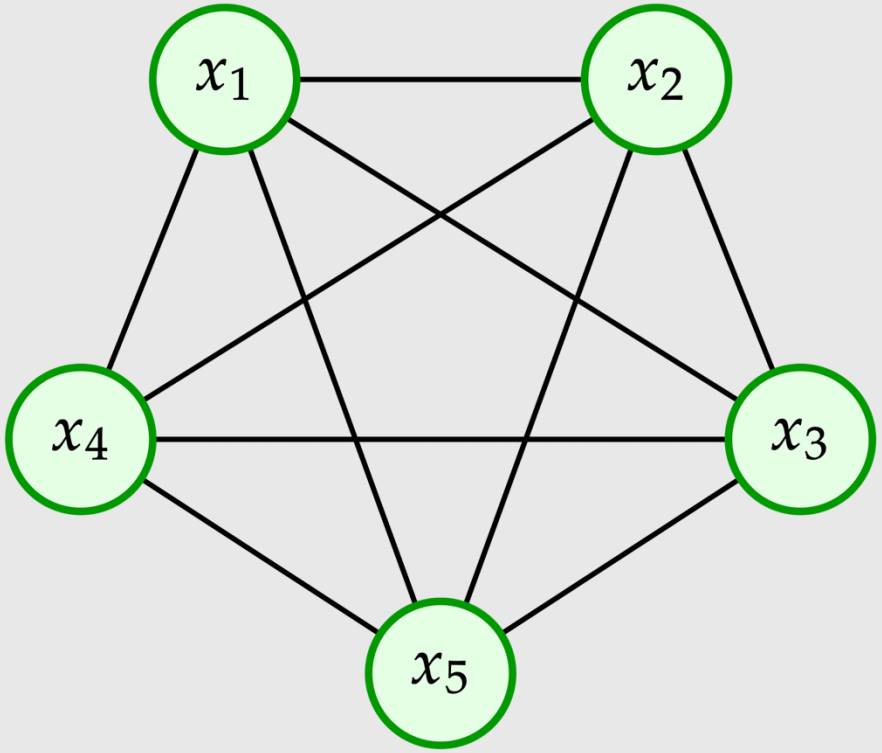
Encodes

$$\text{AMO}: \{0,1\}^5 \to \{0,1\}$$

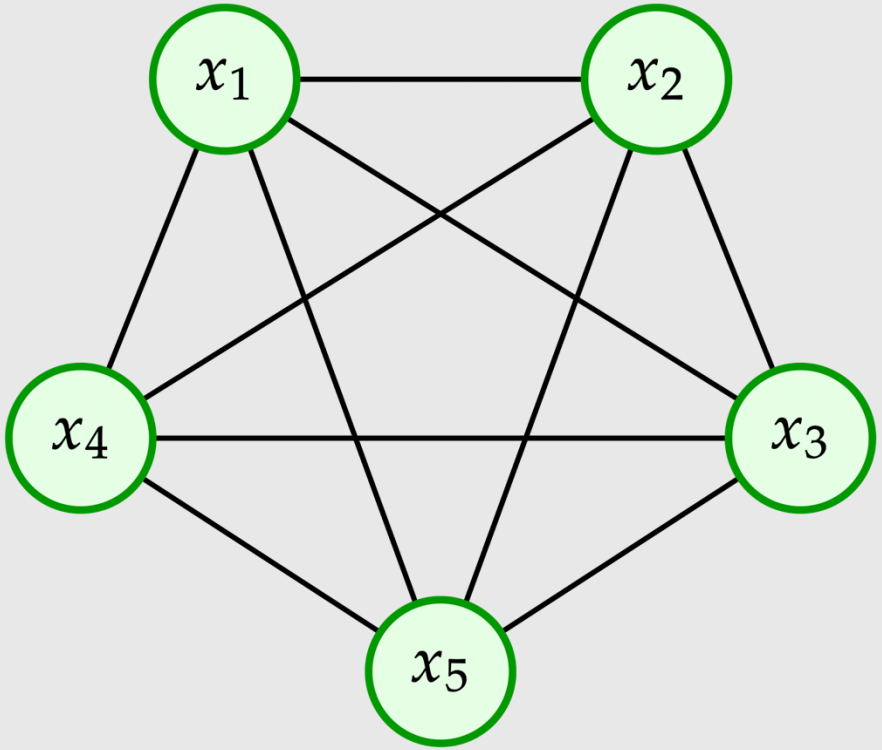
$$\text{AMO}(x_1, x_2, x_3, x_4, x_5) \equiv \left(\sum_{i=1}^{5} x_i \leq 1 \right)$$

Boolean function

Claim: the AMO function corresponds to **independent sets** in a clique



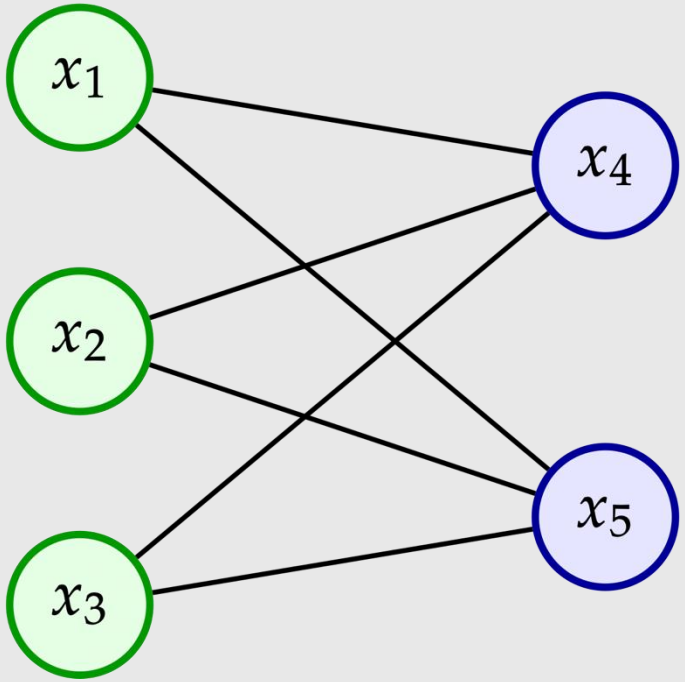
Claim: the AMO function corresponds to **independent sets** in a clique



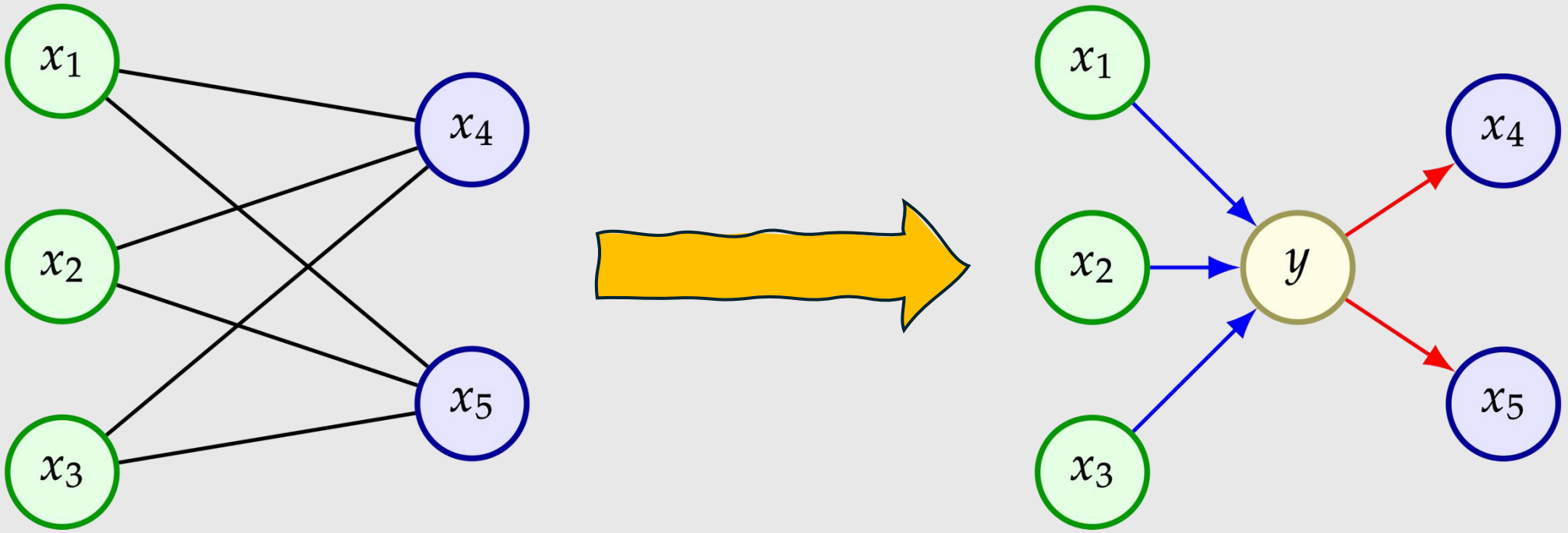
$$F := (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5}) \wedge (\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5}) \wedge (\overline{x_4} \vee \overline{x_5})$$

$$\textsc{IndSet}_G := \bigwedge_{\{i,j\} \in E(G)} (\overline{x_i} \vee \overline{x_j})$$

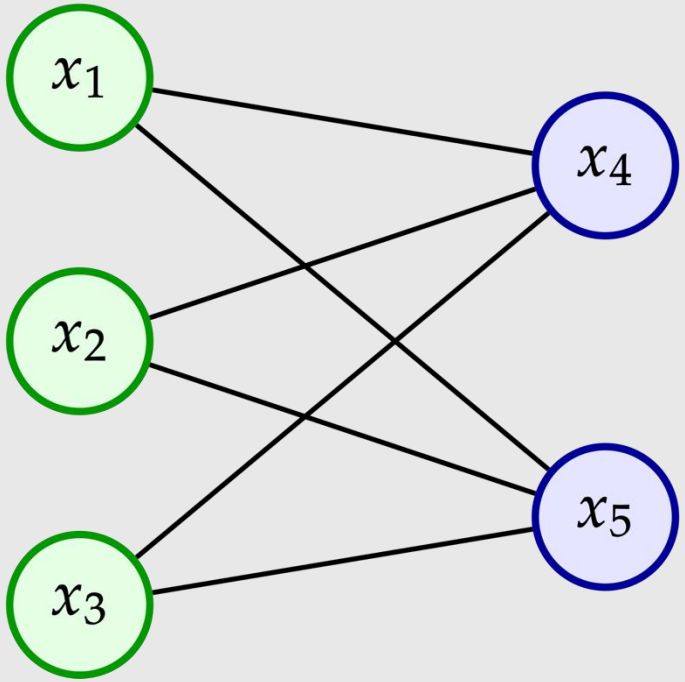
The Magic Step

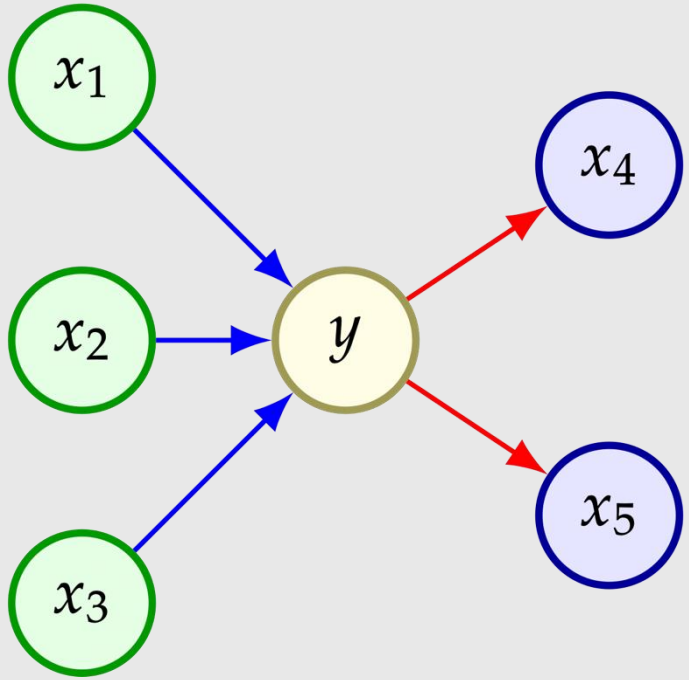


The Magic Step

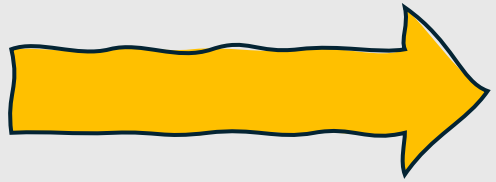


The Magic Step

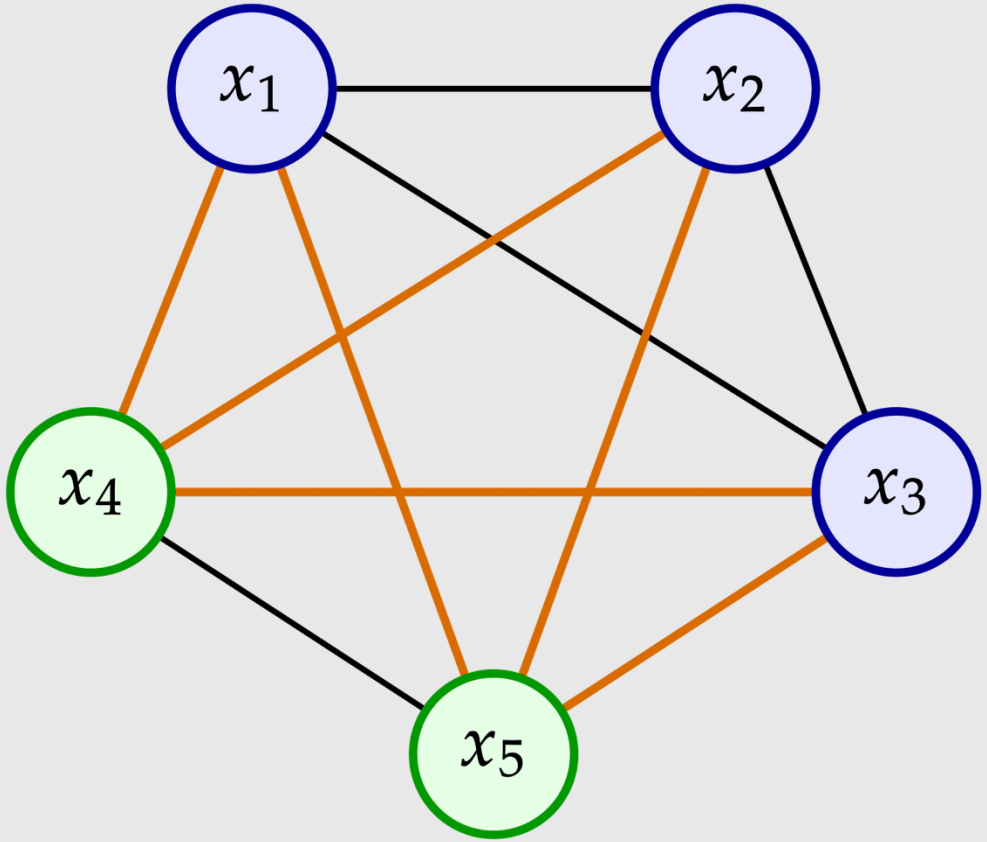




$$(\overline{x_1} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_5})$$
$$(\overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5})$$
$$(\overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_3} \vee \overline{x_5})$$



$$(\overline{x_1} \vee y) \wedge (\overline{x_2} \vee y) \wedge (\overline{x_3} \vee y)$$
$$(\overline{y} \vee \overline{x_4}) \wedge (\overline{y} \vee \overline{x_5})$$



$$|E| = \binom{5}{2} = 10 \text{ clauses}$$

Can we do better?

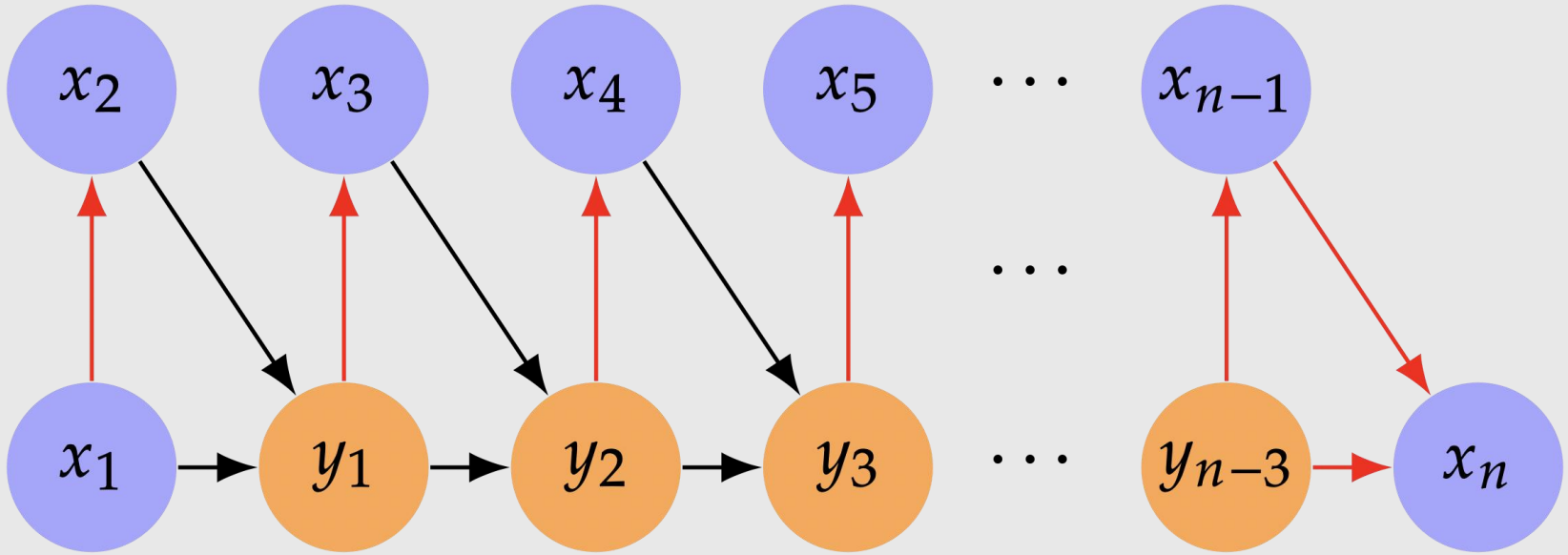
Yes! 9 clauses

AtMostOne function

1. Appears **everywhere**! From Sudokus to Hamiltonian paths

AtMostOne function

1. Appears **everywhere**! From Sudokus to Hamiltonian paths



Sequential encoding: $3n$-6 clauses

AtMostOne function

1. Appears **everywhere**! From Sudokus to Hamiltonian paths

2. Chen's encoding (2010): $2n + 4\sqrt{n}$ clauses.

AtMostOne function

1. Appears **everywhere**! From Sudokus to Hamiltonian paths
2. Chen's encoding (2010): $2n + 4\sqrt{n}$ clauses.
3. Kučera, Savický, Vorel (2017): $2n + 2\sqrt{n}$ lower bound for 2-CNF

AtMostOne function

1. Appears **everywhere**! From Sudokus to Hamiltonian paths
2. Chen's encoding (2010): $2n + 4\sqrt{n}$ clauses.
3. Kučera, Savický, Vorel (2017): $2n + 2\sqrt{n}$ lower bound for 2-CNF
4. Krapivin, Przybocki, **S.** (2026): $2n + 2\sqrt{2}\sqrt{n}$ upper bound, wider!

AtMostOne function

1. Appears **everywhere**! From Sudokus to Hamiltonian paths
2. Chen's encoding (2010): $2n + 4\sqrt{n}$ clauses.
3. Kučera, Savický, Vorel (2017): $2n + 2\sqrt{n}$ lower bound for 2-CNF
4. Krapivin, Przybocki, **S.** (2026): $2n + 2\sqrt{2}\sqrt{n}$ upper bound, wider!

Applications to circuit complexity:

- Improved circuit for Th_2 (from Adleman 1976)
- Solves question from Bloniarz (1979) on single-level circuits
- Simplified refutation of the single-level conjecture (Wegener, 1991)

General progress on cardinality constraints

Towards an Optimal CNF Encoding of Boolean Cardinality Constraints
(Technical Report)

Carsten Sinz

Symbolic Computation Group, WSI for Computer Science
University of Tübingen, 72076 Tübingen, Germany
sinz@informatik.uni-tuebingen.de

Abstract. We consider the problem of encoding Boolean cardinality constraints in conjunctive normal form (CNF). Boolean cardinality constraints are formulae expressing that at most (resp. at least) k out of n propositional variables or formulae are true. We present a unifying framework for a whole family of such encodings encompassing previously proposed solutions. We give two novel encodings that improve upon existing results, one which requires only $7n$ clauses and $2n$ additional variables, and another one demanding $\mathcal{O}(n \cdot k)$ clauses, but with the advantage that inconsistencies can be detected in linear time by unit propagation alone. Moreover, we prove a linear lower bound on the number of required clauses for any such encoding.

Sinz (2005)

Near-Optimal Encodings of Cardinality Constraints

Andrew Krapivin, Benjamin Przybocki, and Bernardo Subercaseaux

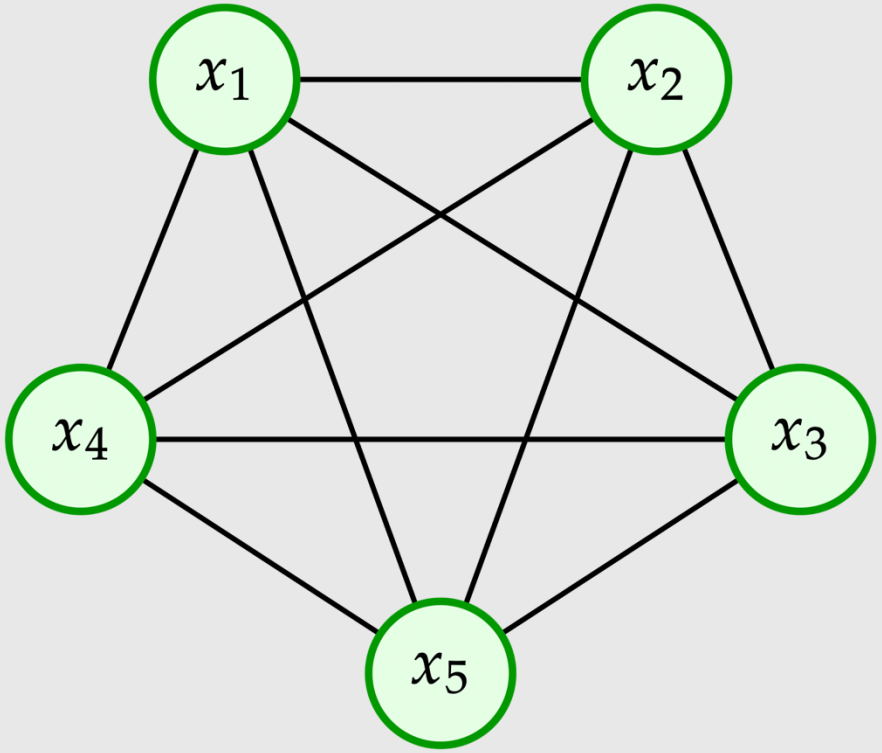
Carnegie Mellon University
[akrapivi,bprzyboc,bsuberca]@andrew.cmu.edu

Abstract

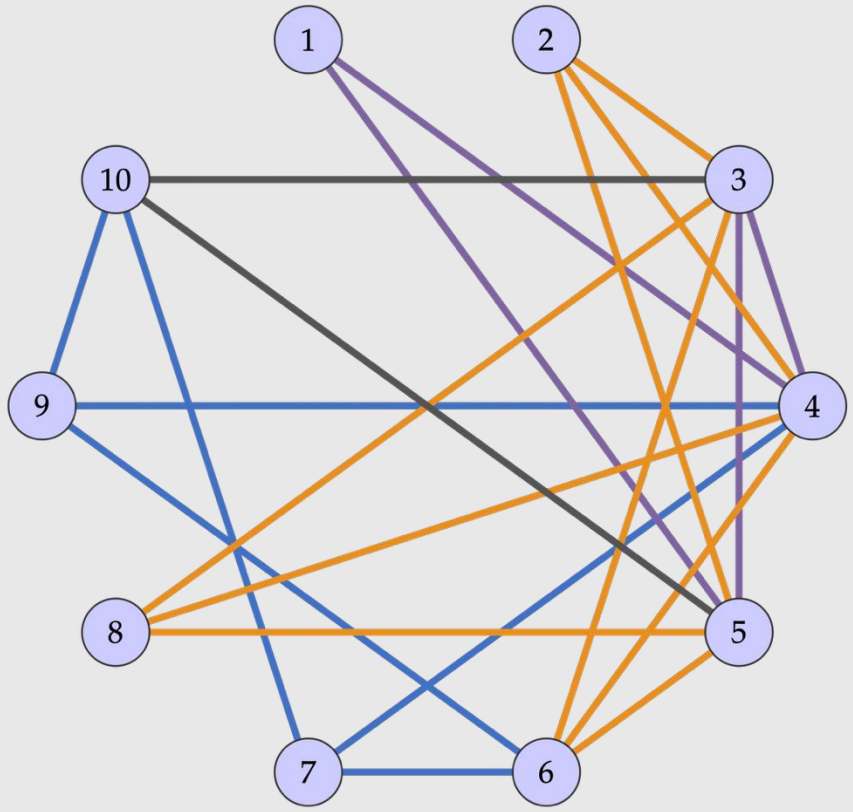
We present several novel encodings for cardinality constraints, which use fewer clauses than previous encodings and, more importantly, introduce new generally applicable techniques for constructing compact encodings. First, we present a CNF encoding for the $\mathsf{AtMostOne}(x_1, \ldots, x_n)$ constraint using $2n + 2\sqrt{2n} + O(\sqrt[3]{n})$ clauses, thus refuting the conjectured optimality of Chen's product encoding. Our construction also yields a smaller monotone circuit for the threshold-2 function, improving on a 50-year-old construction of Adleman and incidentally solving a long-standing open problem in circuit complexity. On the other hand, we show that any encoding for this constraint requires at least $2n + \sqrt{n+1} - 2$ clauses, which is the first nontrivial unconditional lower bound for this constraint and answers a question of Kučera, Savický, and Vorel. We then turn our attention to encodings of $\mathsf{AtMost}_k(x_1, \ldots, x_n)$, where we introduce *grid compression*, a technique inspired by hash tables, to give encodings using $2n + o(n)$ clauses as long as $k = o(\sqrt[3]{n})$ and $4n + o(n)$ clauses as long as $k = o(n)$. Previously, the smallest known encodings were of size $(k+1)n + o(n)$ for $k \leq 5$ and $7n - o(n)$ for $k \geq 6$.

K. P. S. (2026)

Claim: the AMO function corresponds to **independent sets** in a clique



For more general graphs, **partition** the edges into **bicliques**



$B_1 = \{4, 6, 10\} \times \{7, 9\}$

$B_2 = \{1, 3\} \times \{4, 5\}$

$B_3 = \{2, 6, 8\} \times \{3, 4, 5\}$

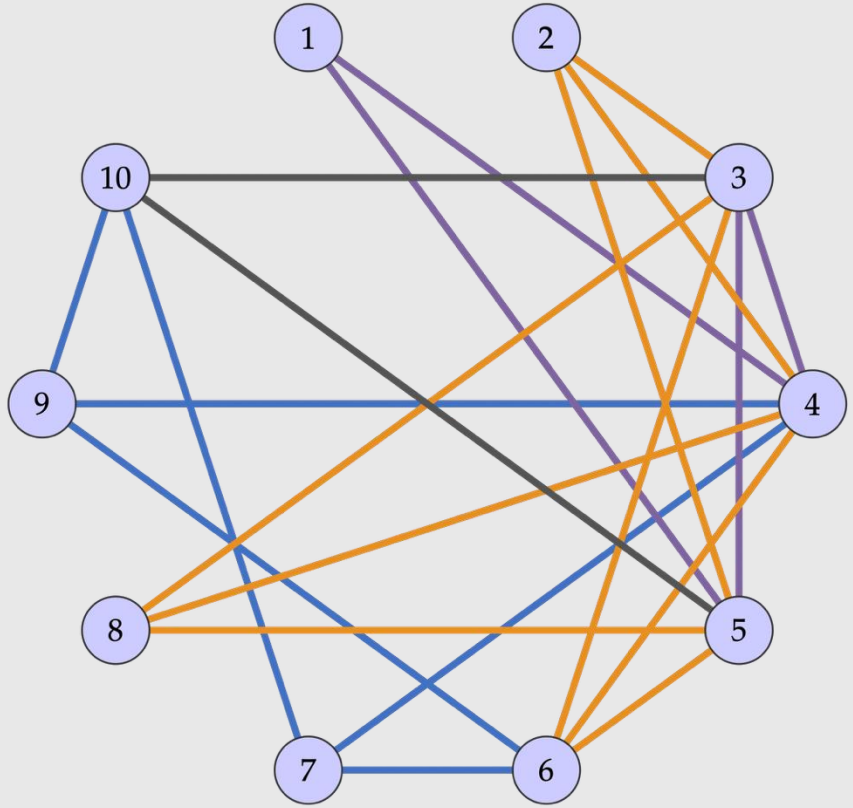
$B_4 = \{3, 5\} \times \{10\}$

For more general graphs, **partition** the edges into **bicliques**

Theorem:
Every 2-CNF function on n bits can be encoded with

$$\frac{\lg(3)}{4} \cdot \frac{n^2}{\lg n} \approx 0.395 \frac{n^2}{\lg n}$$

binary clauses



$B_1 = \{4, 6, 10\} \times \{7, 9\}$
$B_2 = \{1, 3\} \times \{4, 5\}$
$B_3 = \{2, 6, 8\} \times \{3, 4, 5\}$
$B_4 = \{3, 5\} \times \{10\}$

CNF Encodings

Given a boolean function $f : \{0,1\}^n \to \{0,1\}$,
a CNF encoding for f is a tuple (φ, X, Y) such that:

1. $X = (x_1, \ldots, x_n)$ are 'input' variables
2. Y is a set of 'auxiliary' (non-deterministic) variables
3. $f(X) = 1 \iff \exists Y, \varphi(X, Y) = 1$

CNF Encoding Complexity

$$\textsc{enc}(f) = \min_{\varphi \text{ encodes } f} |\varphi|$$

Minimum number of clauses to encode a Boolean function

CNF Encoding Complexity

$$\textsc{enc}(f) = \min_{\varphi \text{ encodes } f} |\varphi|$$

Minimum number of clauses to encode a Boolean function

E.g., what's the **worst** case over all functions on n bits?

CNF Encoding Complexity

$$\textsc{enc}(f) = \min_{\varphi \text{ encodes } f} |\varphi|$$

Minimum number of clauses to encode a Boolean function

E.g., what's the **worst** case over all functions on n bits?

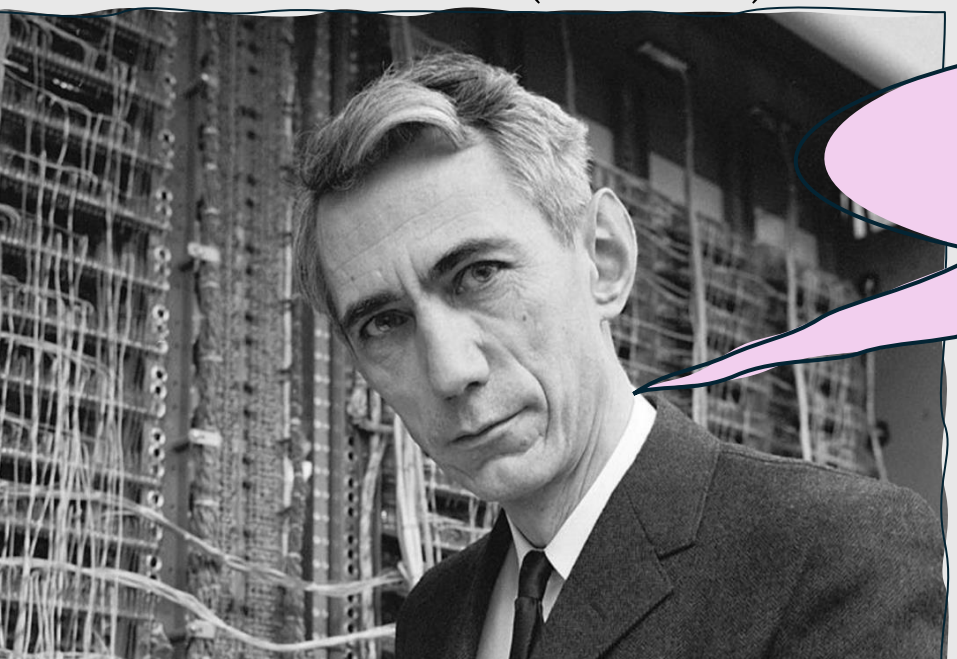
$$\Theta\left(\sqrt{2^n}\right)$$

CNF Encoding Complexity

E.g., what's the **worst** case over all functions on n bits?

$$\Theta\left(\sqrt{2^n}\right)$$

The real Claude (Shannon)



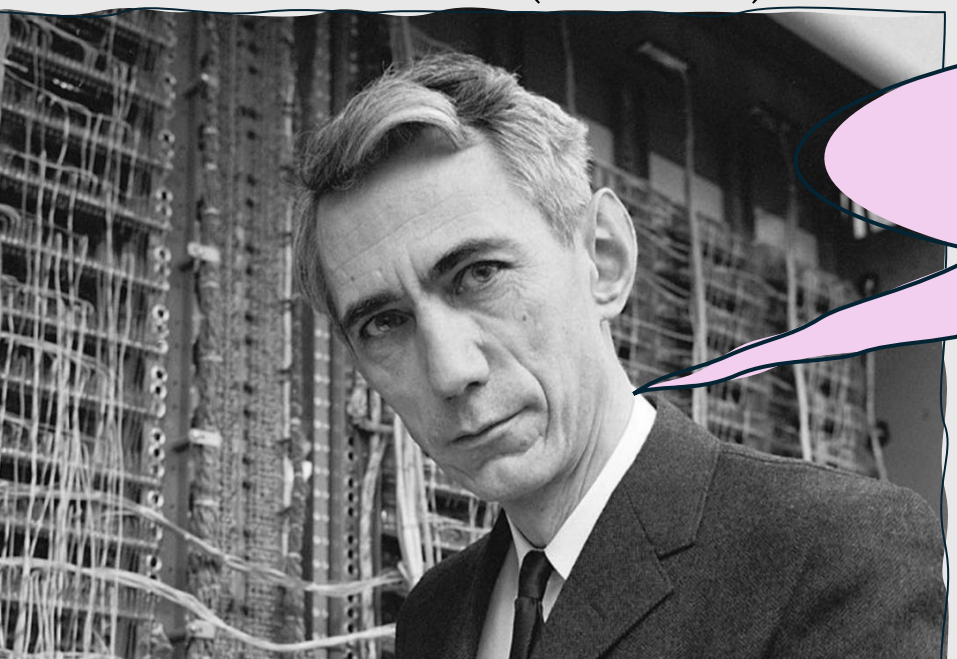
But there are 2^{2^n} boolean functions !

CNF Encoding Complexity

E.g., what's the **worst** case over all functions on n bits?

$$\Theta\left(\sqrt{2^n}\right)$$

The real Claude (Shannon)

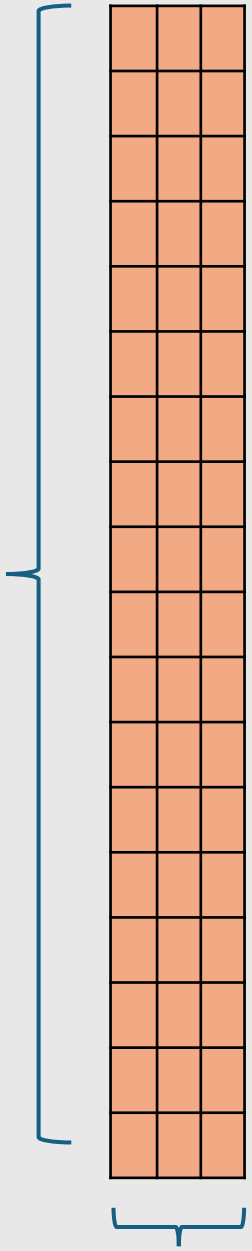


But there are 2^{2^n} boolean functions !

Yes Claude, so some CNF must use $\sim \frac{2^n}{n}$ literals

Tseitin Transform
of Shannon–Lupanov circuits

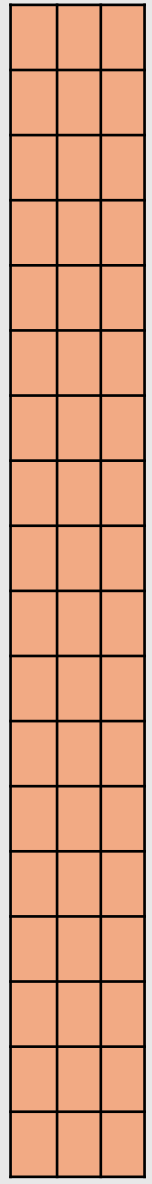
$$\frac{2^n}{n}$$



3-CNF

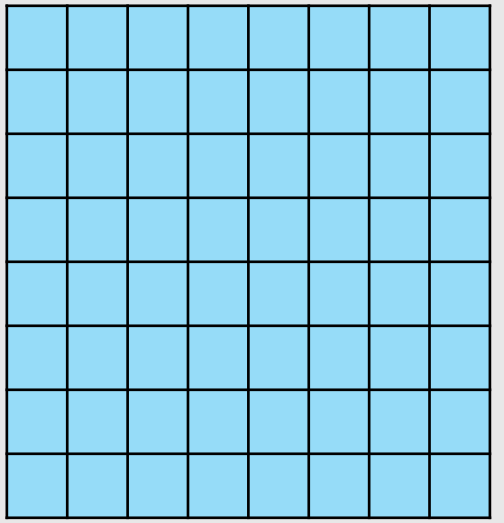
Tseitin Transform
of Shannon–Lupanov circuits

$$\frac{2^n}{n}$$



3-CNF

Our encodings



$\sim \sqrt{2^n}$

$\sim \sqrt{2^n}$

General results

Arbitrary functions

Lower bound: $0.794 \cdot \sqrt{2^n}$

Upper bound: $2.82 \cdot \sqrt{2^n}$

General results

Arbitrary functions

Lower bound: $0.794 \cdot \sqrt{2^n}$

Upper bound: $2.82 \cdot \sqrt{2^n}$

Sparse functions $M := |f^{-1}(1)|$

Lower bound: $\sqrt{M} \cdot \sqrt{n - \lg M}$

Upper bound: $\sqrt{M} \cdot (n - \lg M)$

General results

Arbitrary functions

Lower bound: $0.794 \cdot \sqrt{2^n}$

Upper bound: $2.82 \cdot \sqrt{2^n}$

***k*-CNF functions**

Lower bound: $n^{k/2}$

Upper bound: $o_k(n^k/\lg n)$

Sparse functions $M := |f^{-1}(1)|$

Lower bound: $\sqrt{M} \cdot \sqrt{n - \lg M}$

Upper bound: $\sqrt{M} \cdot (n - \lg M)$

General results

Arbitrary functions

Lower bound: $0.794 \cdot \sqrt{2^n}$

Upper bound: $2.82 \cdot \sqrt{2^n}$

***k*-CNF functions**

Lower bound: $n^{k/2}$

Upper bound: $o_k(n^k/\lg n)$

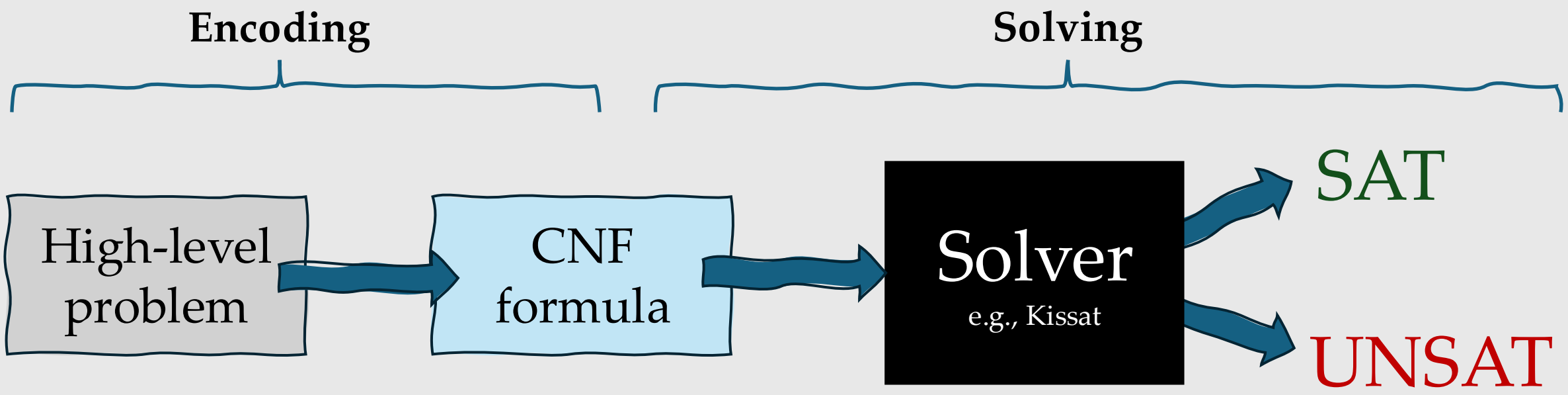
Sparse functions $M := |f^{-1}(1)|$

Lower bound: $\sqrt{M} \cdot \sqrt{n - \lg M}$

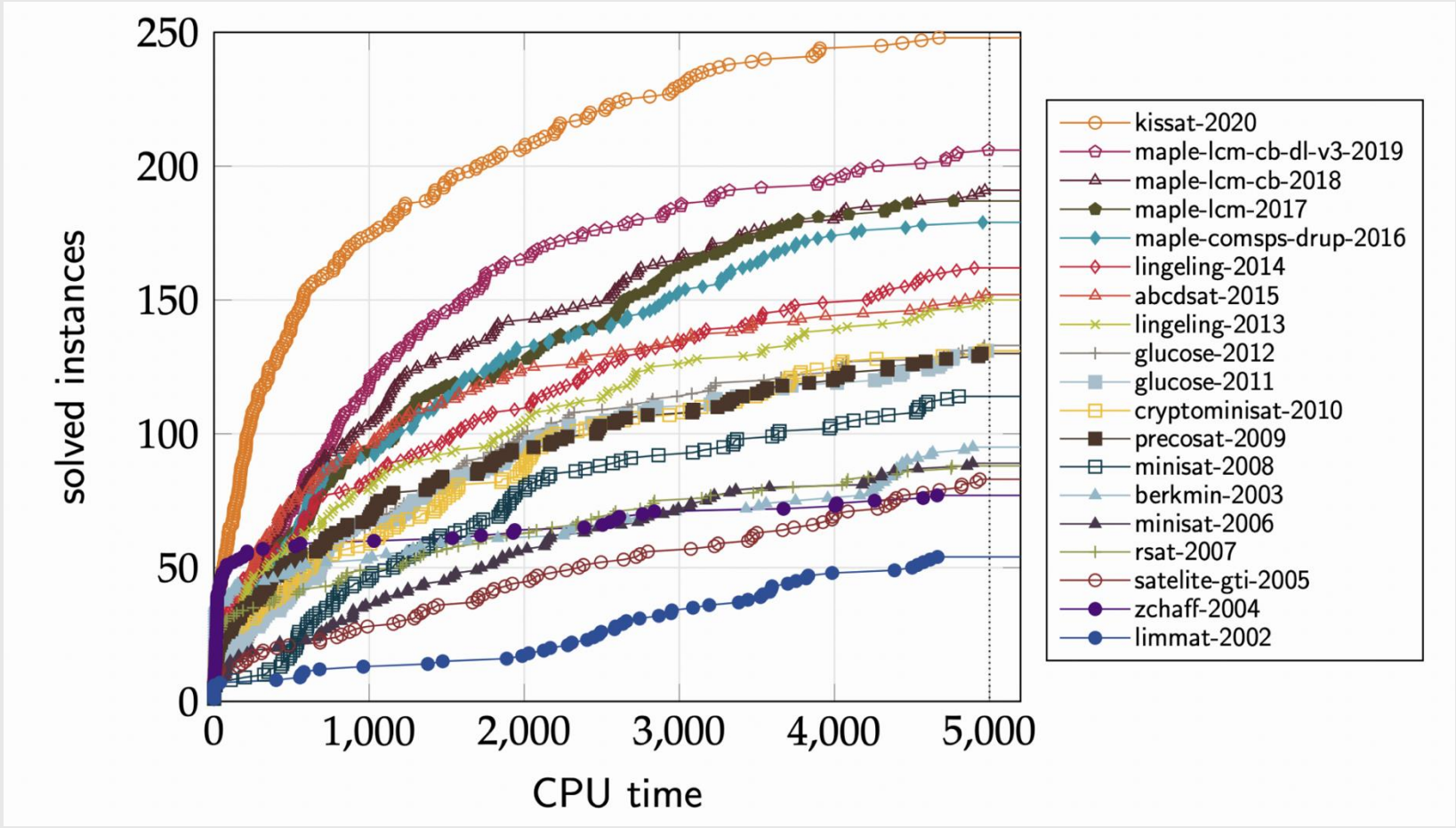
Upper bound: $\sqrt{M} \cdot (n - \lg M)$

Functions computable by NRAM in time ***t*** with ***m*** total bits of advice, ***b*** used per input:

$$\widetilde{O}\left(\sqrt{mb} + t\right)$$


Encoding
Solving
High-level problem
CNF formula
Solver
e.g., Kissat
SAT
UNSAT

There's been tremendous progress on **solving**!



There's been tremendous progress on **solving**!

Software vs Hardware

	Jerusat (2003)	siege_v3 (2003)	Forklift (2003)	Glucose (2016)	CaDiCal (2019)	MapleSat19 (2019)
old HW (1999)	38	43	43	105	100	72
new HW (2019)	71	96	105	188	184	187

Team SW

Team HW

Fichte, Hecher, Szeider (2021)

Encodings can matter more than software + hardware

We have gotten x4000 speed-ups on many concrete problems by changing the encoding

Encodings can matter more than software + hardware

We have gotten x4000 speed-ups on many concrete problems by changing the encoding

An $O(n^4)$ encoding for $n = 500$ cannot even be parsed by Kissat! So getting $O(n^2)$ makes all the difference

Encodings can matter more than software + hardware

We have gotten x4000 speed-ups on many concrete problems by changing the encoding

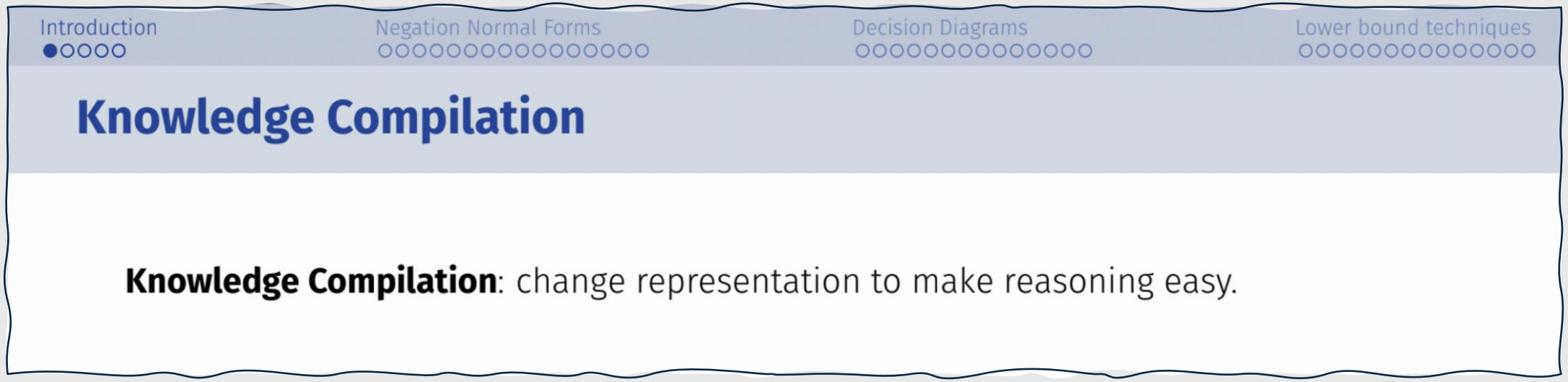
An $O(n^4)$ encoding for $n = 500$ cannot even be parsed by Kissat! So getting $O(n^2)$ makes all the difference

Design principles of effective encodings are still unclear; art based on hard-won experience

Is succinctness always better?

Is succinctness always better?

How does this fit in Knowledge Compilation?



(De Colnet, 2026, Dagstuhl)

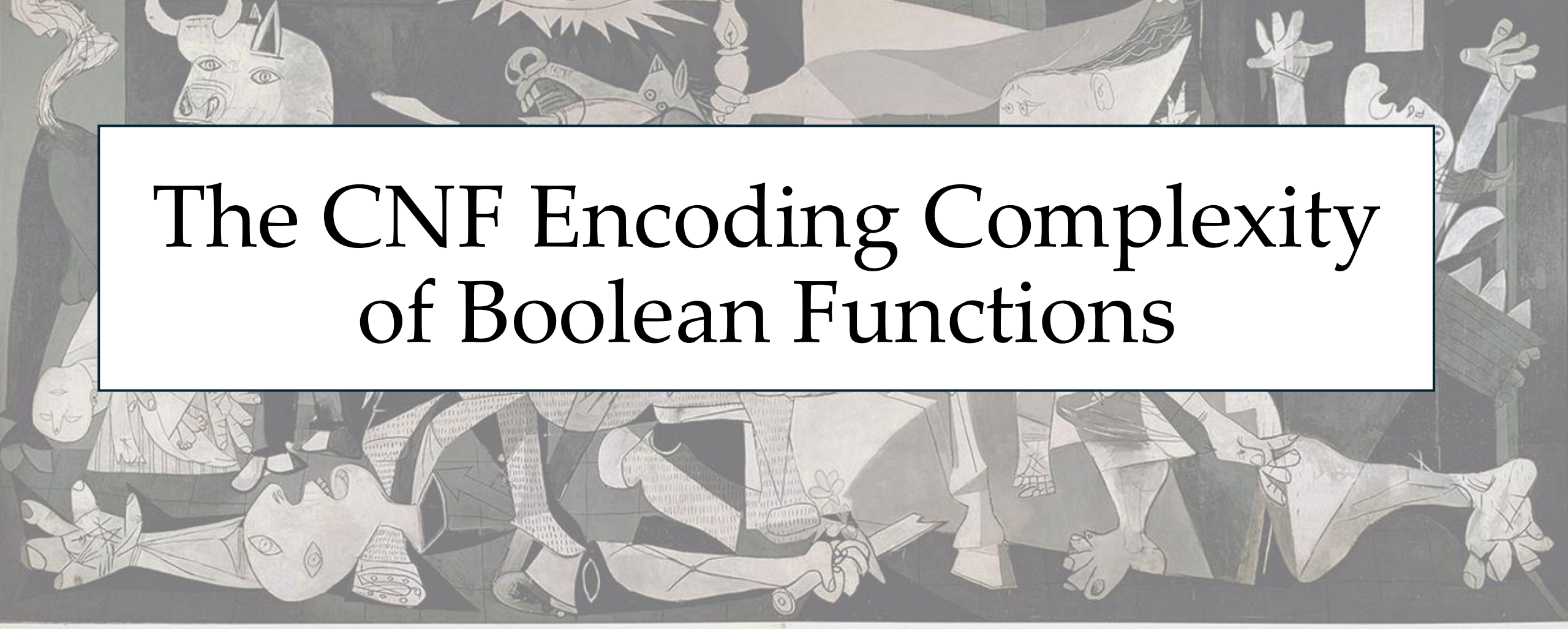
How does this fit in Knowledge Compilation?

Knowledge Compilation

Knowledge Compilation: change representation to make reasoning easy.

(De Colnet, 2026, Dagstuhl)

CNF Encodings make reasoning **easier** (but still hard!)



The CNF Encoding Complexity of Boolean Functions

Bernardo Subercaseaux Carnegie Mellon University

joint work with Andrew Krapivin and Ben Przybocki