New Compilation Languages Based on Restricted Weak Decomposability

Petr Illner

Department of Theoretical Computer Science and Mathematical Logic, Faculty of Mathematics and Physics
Charles University, Czech Republic

illner3@gmail.com

Knowledge representation

How to represent propositional theories

Representations (aka languages) are studied from the following perspectives:

1) **Universality**: Can the language represent any Boolean function?

2) **Succinctness**[1]: How compactly can the language represent Boolean functions with respect to other languages?

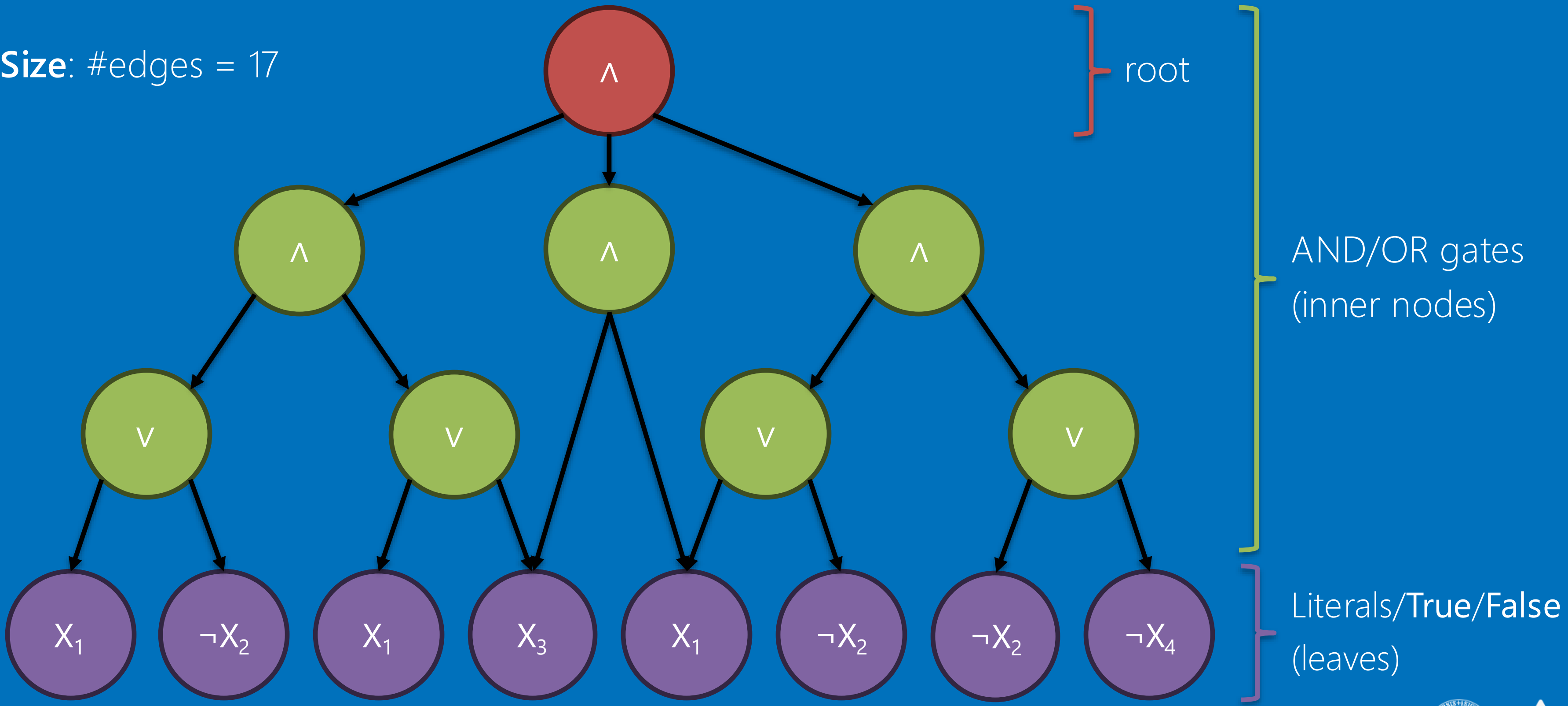
3) **Tractable operations**: What operations (that is, queries and transformations) does the language satisfy?

4) **Knowledge compiler**: Is there a knowledge compiler for the language?

[1] GOGIC, Goran, et al. The comparative linguistics of knowledge representation. In: IJCAI (1). 1995. p. 862-869.

Weak decomposable **negation normal form** (wD**NNF**) **circuits**[1]

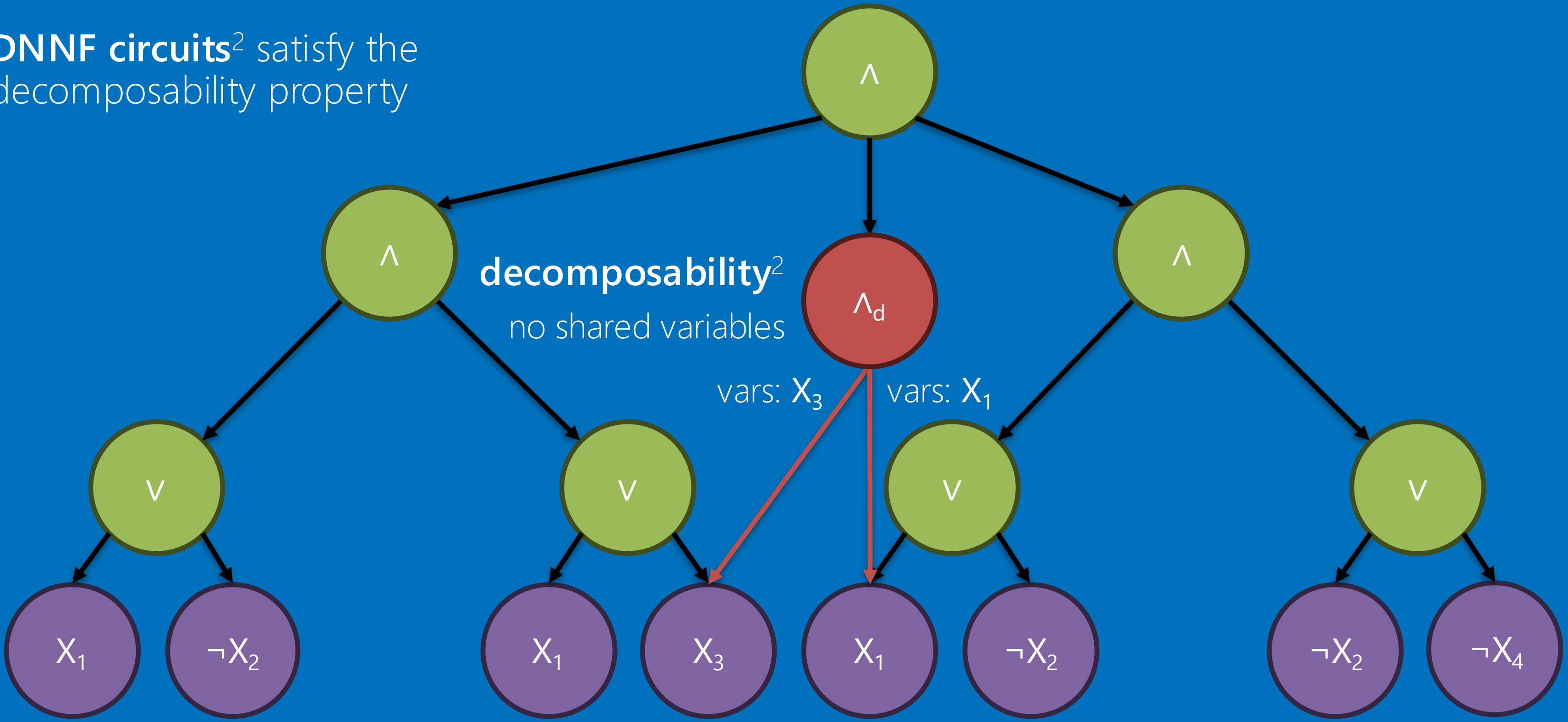
Size: #edges = 17



[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.

Weak **decomposable negation normal form** (wDNNF) **circuits**[1]

DNNF circuits[2] satisfy the decomposability property



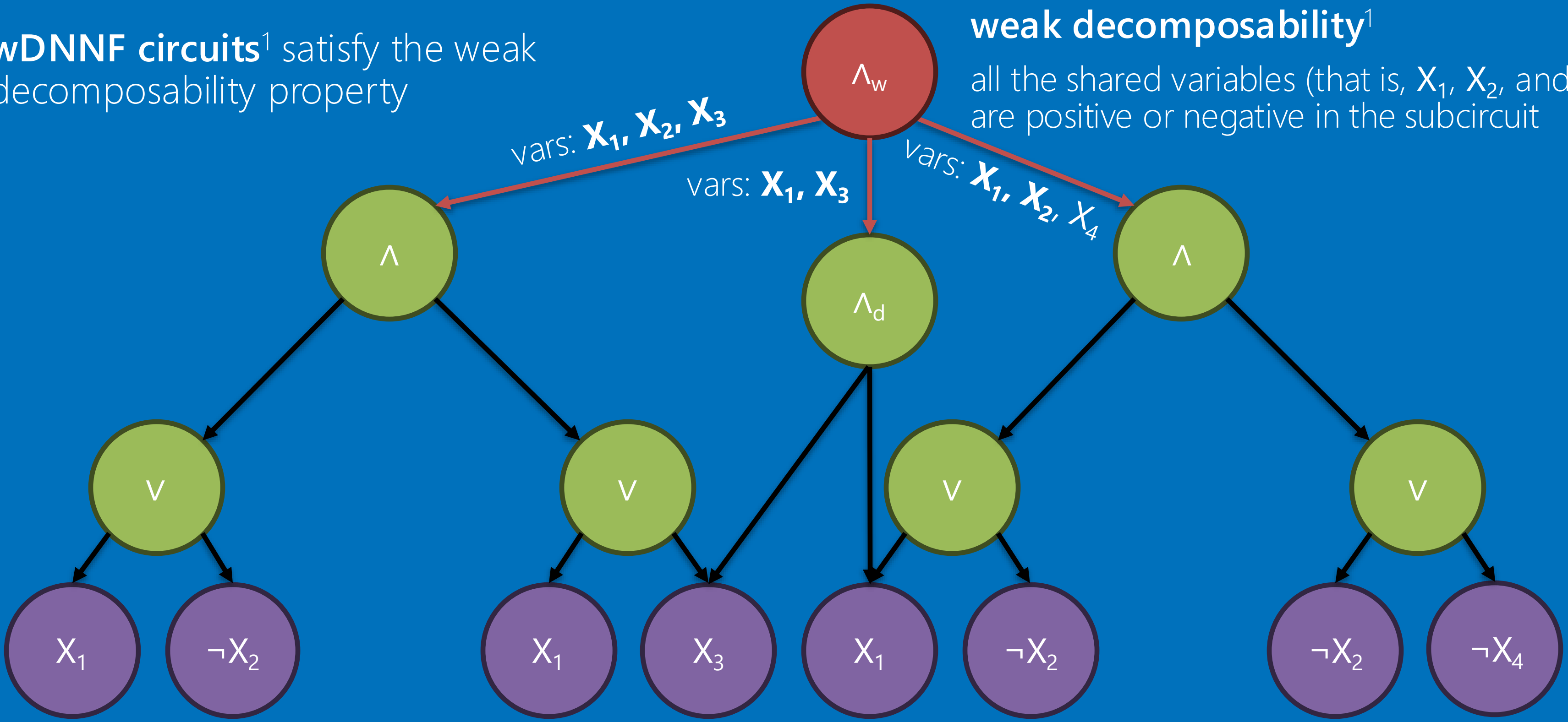
[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.
[2] DARWICHE, Adnan. Compiling knowledge into decomposable negation normal form. In: IJCAI. 1999. p. 284-289.

Weak decomposable negation normal form (wDNNF) circuits[1]

wDNNF circuits[1] satisfy the weak decomposability property

weak decomposability[1]

all the shared variables (that is, X_1, X_2, and X_3) are positive or negative in the subcircuit



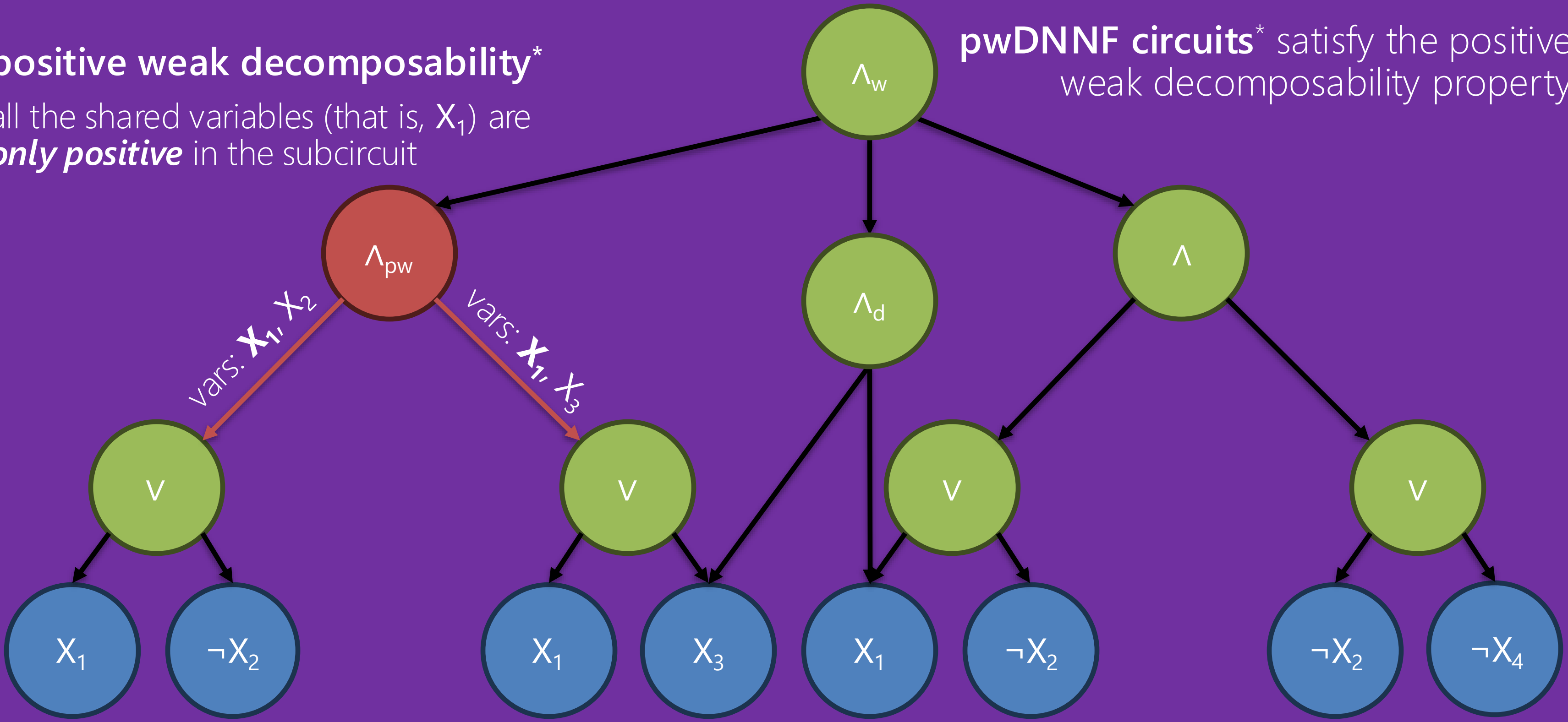
[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.

Weak decomposable negation normal form (wDNNF) circuits[1]

positive weak decomposability*

all the shared variables (that is, X_1) are ***only positive*** in the subcircuit

pwDNNF circuits* satisfy the positive weak decomposability property



[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.

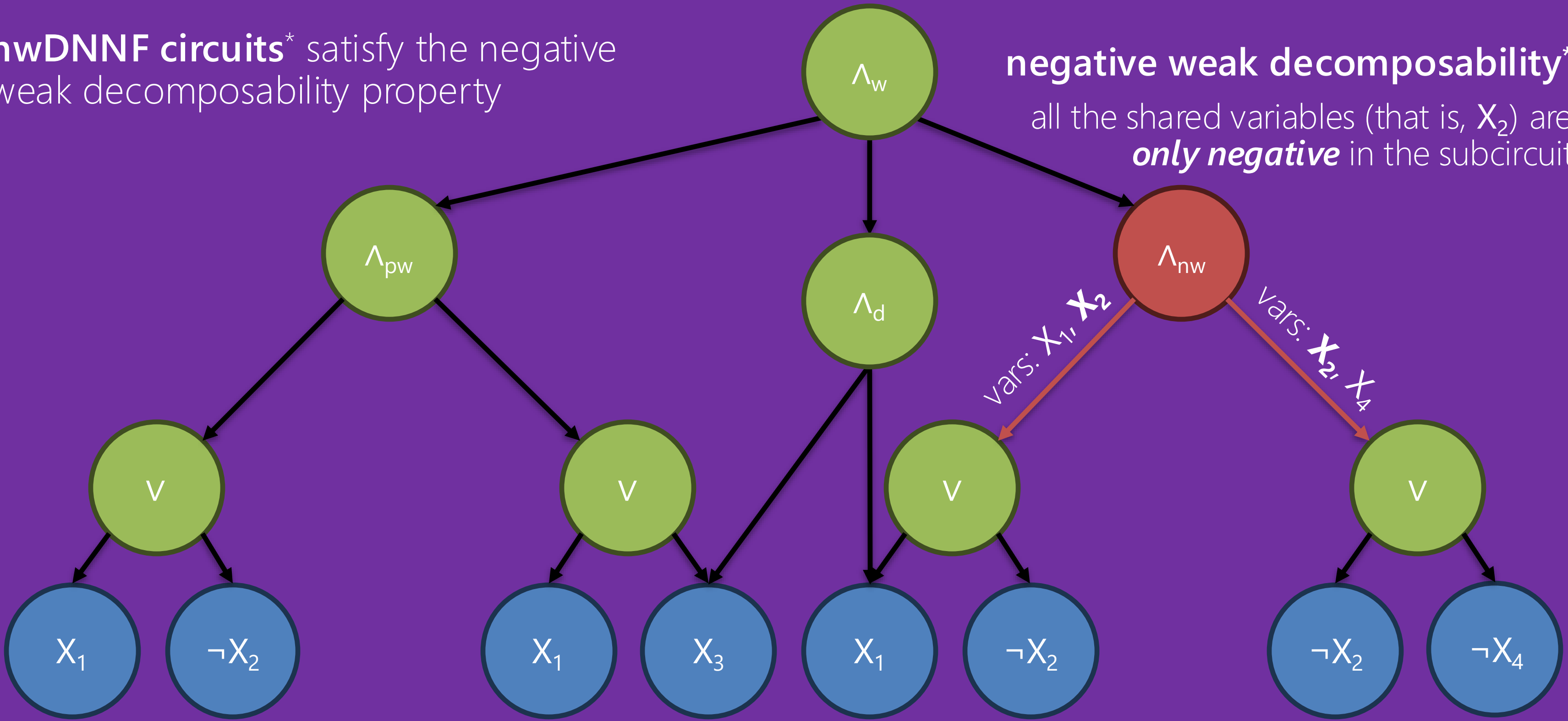
* **new**

Weak decomposable negation normal form (wDNNF) circuits[1]

nwDNNF circuits* satisfy the negative weak decomposability property

negative weak decomposability*

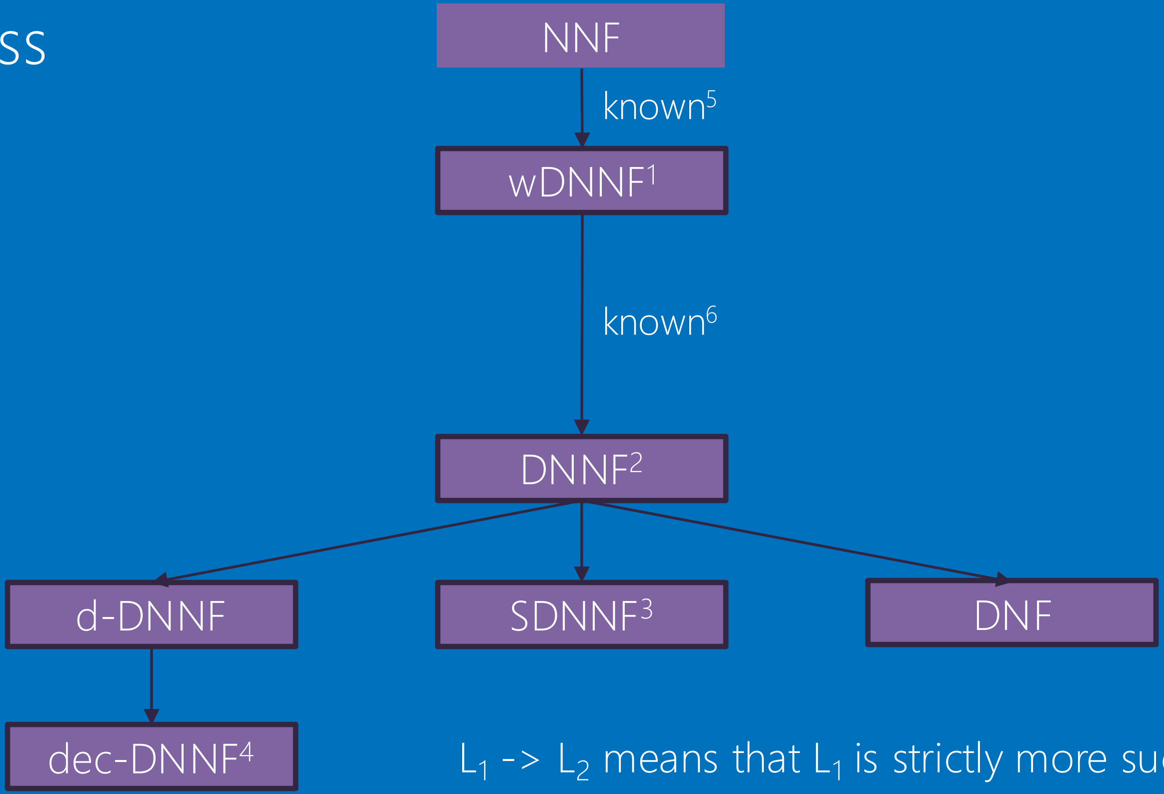
all the shared variables (that is, X_2) are ***only negative*** in the subcircuit



[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.

* new

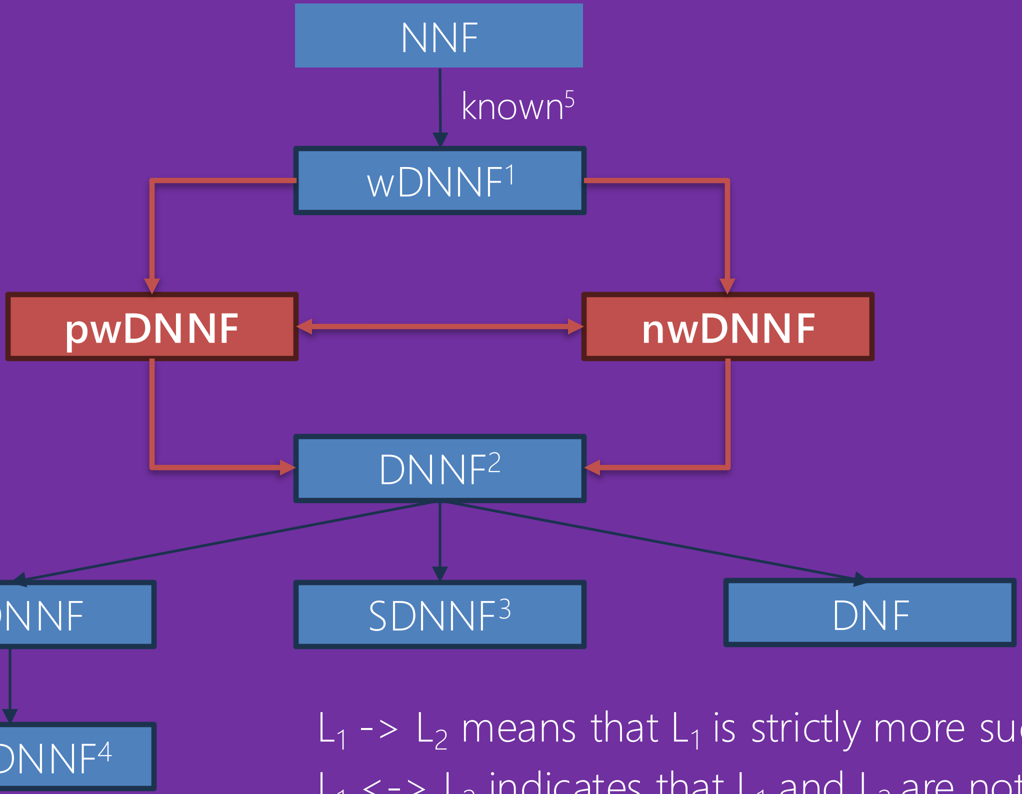
Succinctness



L_1 -> L_2 means that L_1 is strictly more succinct than L_2

[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.
[2] DARWICHE, Adnan. Compiling knowledge into decomposable negation normal form. In: IJCAI. 1999. p. 284-289.
[3] PIPATSRISAWAT, Knot; DARWICHE, Adnan. New Compilation Languages Based on Structured Decomposability. In: AAAI. 2008. p. 517-522.
[4] HUANG, Jinbo; DARWICHE, Adnan. The language of search. Journal of Artificial Intelligence Research, 2007, 29: 191-219.
[5] ILLNER, Petr; KUČERA, Petr. A Compiler for Weak Decomposable Negation Normal Form. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2024. p. 10562-10570.
[6] DE COLNET, Alexis; MENGEL, Stefan. A Compilation of Succinctness Results for Arithmetic Circuits. arXiv preprint arXiv:2110.13014, 2021.

Succinctness



L_1 -> L_2 means that L_1 is strictly more succinct than L_2

L_1 <-> L_2 indicates that L_1 and L_2 are not comparable

[1] AKSHAY, S., et al. Knowledge compilation for Boolean functional synthesis. In: 2019 Formal Methods in Computer Aided Design (FMCAD). IEEE, 2019. p. 161-169.
[2] DARWICHE, Adnan. Compiling knowledge into decomposable negation normal form. In: IJCAI. 1999. p. 284-289.
[3] PIPATSRISAWAT, Knot; DARWICHE, Adnan. New Compilation Languages Based on Structured Decomposability. In: AAAI. 2008. p. 517-522.
[4] HUANG, Jinbo; DARWICHE, Adnan. The language of search. Journal of Artificial Intelligence Research, 2007, 29: 191-219.
[5] ILLNER, Petr; KUČERA, Petr. A Compiler for Weak Decomposable Negation Normal Form. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2024. p. 10562-10570.

Standard queries

Query[1]	wDNNF	=	**pwDNNF**	=	**nwDNNF**	=	DNNF
Consistency *(CO)*	✓		✓*		✓*		✓
Validity *(VA)*	✗		✗*		✗*		✗
Clausal entailment *(CE)*	✓		✓*		✓*		✓
Implicant *(IM)*	✗		✗*		✗*		✗
Equivalence *(EQ)*	✗		✗*		✗*		✗
Sentential entailment *(SE)*	✗		✗*		✗*		✗
Model counting *(CT)*	✗		✗*		✗*		✗
Model enumeration *(ME)*	✓		✓*		✓*		✓

✓ polytime ✓ with polynomial delay ✗ not polytime unless $P = NP$ * **new result**

[1] DARWICHE, Adnan; MARQUIS, Pierre. A knowledge compilation map. Journal of Artificial Intelligence Research, 2002, 17: 229-264.

Transformations

Transformation[1]	wDNNF	=	**pwDNNF**	=	**nwDNNF**	=	DNNF
Conditioning *(CD)*	✓		✓*		✓*		✓
Singleton forgetting *(SFO)*	✓		✓*		✓*		✓
Forgetting *(FO)*	✓		✓*		✓*		✓
Conjunction *(∧C)*	✗		✗*		✗*		✗
Bounded conjunction *(∧BC)*	✗		✗*		✗*		✗
Disjunction *(∨C)*	✓		✓*		✓*		✓
Bounded disjunction *(∨BC)*	✓		✓*		✓*		✓
Negation *(¬C)*	✗		✗*		✗*		✗

✓ polytime ✗ not polytime unless $P = NP$ * **new result**

[1] DARWICHE, Adnan; MARQUIS, Pierre. A knowledge compilation map. Journal of Artificial Intelligence Research, 2002, 17: 229-264.

Minimum and maximum cardinalities

The cardinality of a model is the number of variables that are set to **True**.

The minimum (resp. maximum) cardinality of a circuit/formula is the minimum (resp. maximum) cardinality of all its models.

For an inconsistent circuit/formula, the minimum (resp. maximum) cardinality is $+\infty$ (resp. $-\infty$).

The minimum (resp. maximum) cardinality problem	
Instance:	Given a circuit/formula.
Task:	Get (a model with) the minimum (resp. maximum) cardinality of the circuit/formula.

$(X_1 \vee X_2) \wedge (X_2 \vee X_3) \wedge (\neg X_1 \vee \neg X_3)$

models			cardinality
$\neg X_1$,	X_2,	$\neg X_3$	**1** (min)
$\neg X_1$,	X_2,	X_3	**2** (max)
X_1,	X_2,	$\neg X_3$	**2** (max)

Weighted minimum and maximum cardinalities

The weighted minimum (resp. maximum) cardinality problem*

Instance:	Given a circuit and a non-negative weight for each variable.
Task:	Find a model that minimises (resp. maximises) the sum of the weights of the variables that are set to **True**.

variable	weight
X_1	1
X_2	2
X_3	3

$(X_1 \lor X_2) \land (X_2 \lor X_3) \land (\neg X_1 \lor \neg X_3)$

models			weighted cardinality	
$\neg X_1$,	X_2,	$\neg X_3$	**2**	(min)
$\neg X_1$,	X_2,	X_3	2 + 3 = **5**	(max)
X_1,	X_2,	$\neg X_3$	1 + 2 = **3**	

* For CNF formulae, it is called **the minimum-cost (resp. maximum-cost) satisfiability problem**[1]

[1] LI, Xiao Yu. Optimization algorithms for the minimum-cost satisfiability problem. North Carolina State University, 2004.

Non-standard queries

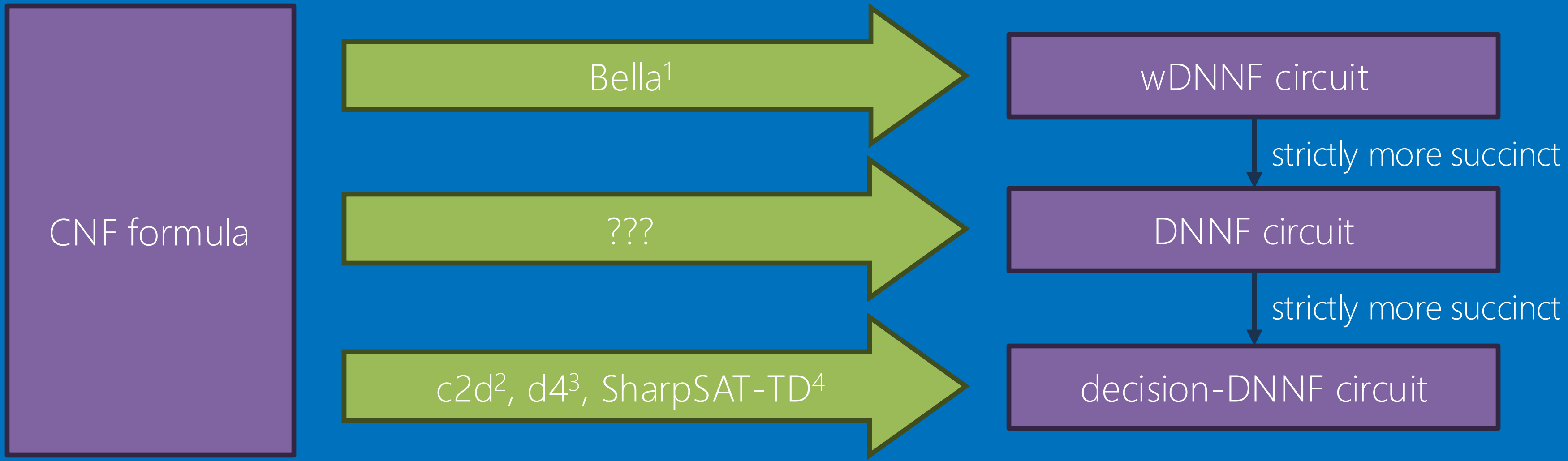
Query[1]	wDNNF	≠	pwDNNF	≠	nwDNNF	≠	DNNF
Minimum cardinality *(MinCard)*	✗*		✗*		✓*		✓
Weighted minimum cardinality *(wMinCard)*	✗*		✗*		✓*		✓
Maximum cardinality *(MaxCard)*	✗*		✓*		✗*		✓
Weighted maximum cardinality *(wMaxCard)*	✗*		✓*		✗*		✓

✓ polytime ✗ not polytime unless $P = NP$ * **new result**

[1] DARWICHE, Adnan. Decomposable negation normal form. Journal of the ACM (JACM), 2001, 48.4: 608-647.

Knowledge compilers

Currently, if we are interested in the (weighted) minimum or maximum cardinality queries, which are essential for many applications, we must use decision-DNNF circuits, which also satisfy harder operations (for example, model counting) and are strictly less succinct.



[1] ILLNER, Petr; KUČERA, Petr. A Compiler for Weak Decomposable Negation Normal Form. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2024. p. 10562-10570.
[2] DARWICHE, Adnan, et al. New advances in compiling CNF to decomposable negation normal form. In: Proc. of ECAI. Citeseer, 2004. p. 328-332.
[3] LAGNIEZ, Jean-Marie; MARQUIS, Pierre. An Improved Decision-DNNF Compiler. In: IJCAI. 2017. p. 667-673.
[4] KIESEL, Rafael; EITER, Thomas. Knowledge compilation and more with SharpSAT-TD. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning. 2023.

Open question:

Is there a knowledge compiler for DNNF circuits?

Open question:

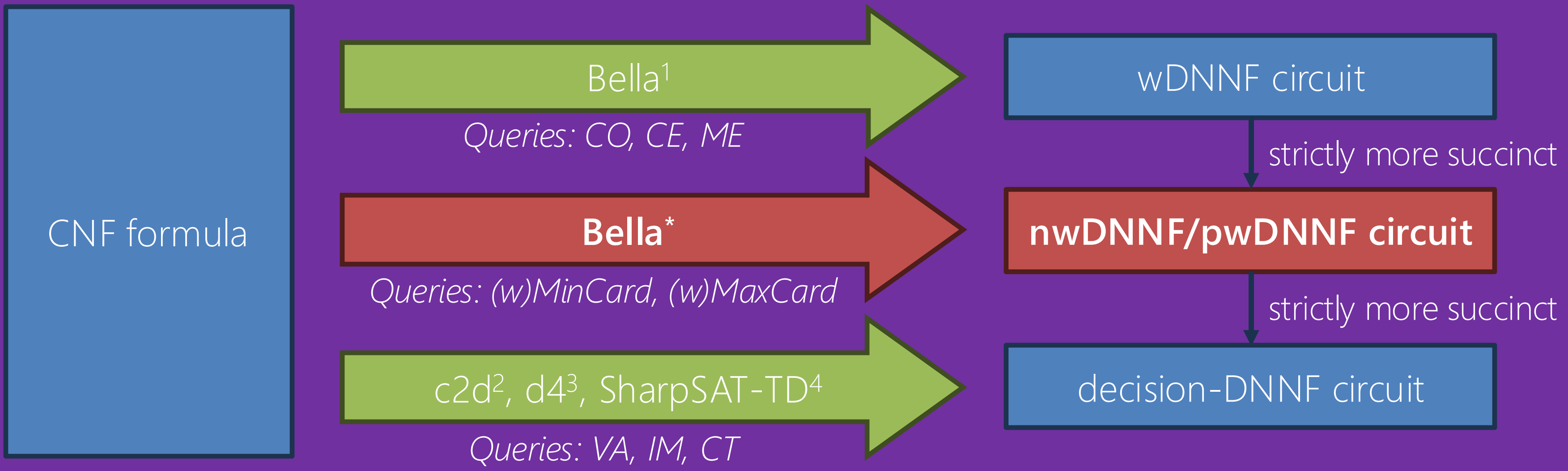
Is there a knowledge compiler for DNNF circuits?

can be reformulated as

Is there a knowledge compiler for a circuit type (resp. circuit types) with the same properties as DNNF circuits?

Knowledge compilers

Considering the (weighted) cardinality queries, we can use pwDNNF and nwDNNF circuits.



[*] our extended variant of Bella[1] for pwDNNF and nwDNNF circuits

[1] ILLNER, Petr; KUČERA, Petr. A Compiler for Weak Decomposable Negation Normal Form. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2024. p. 10562-10570.
[2] DARWICHE, Adnan, et al. New advances in compiling CNF to decomposable negation normal form. In: Proc. of ECAI. Citeseer, 2004. p. 328-332.
[3] LAGNIEZ, Jean-Marie; MARQUIS, Pierre. An Improved Decision-DNNF Compiler. In: IJCAI. 2017. p. 667-673.
[4] KIESEL, Rafael; EITER, Thomas. Knowledge compilation and more with SharpSAT-TD. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning. 2023.

Reducing MPE to minimum-cost satisfiability

The most probable explanation (MPE) problem

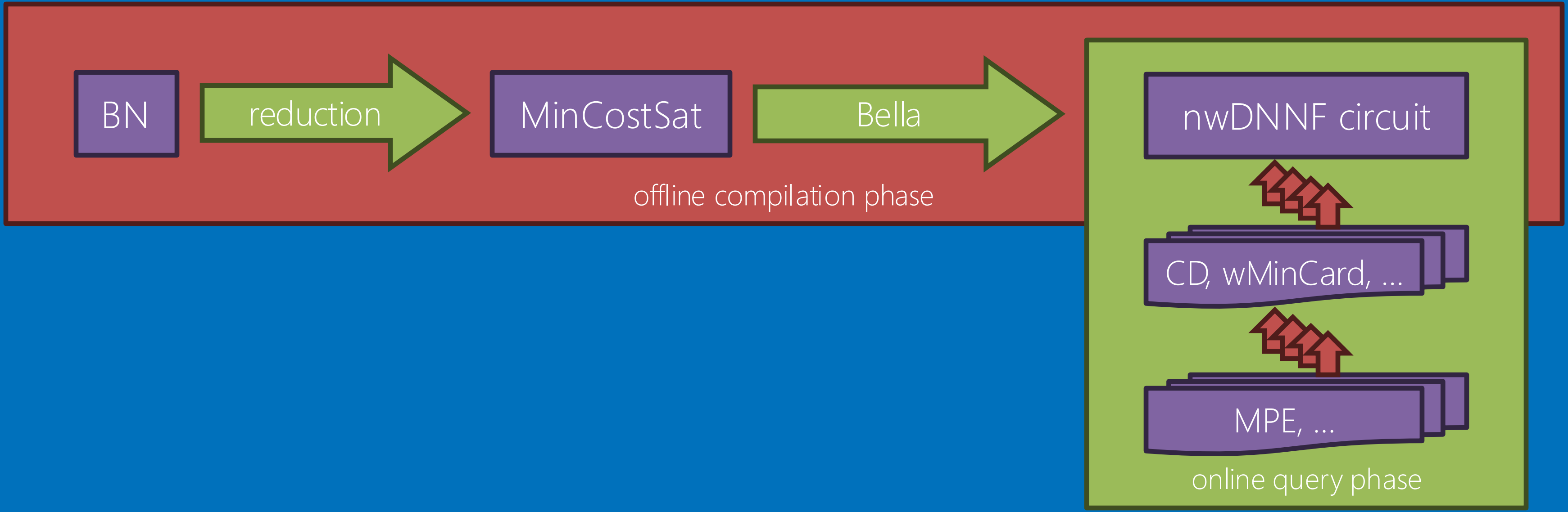
Instance:	Given a Bayesian network (BN) and some evidence.
Task:	Find a variable instantiation of the remaining variables (that is, an MPE instantiation) with the highest probability given that evidence (that is, the MPE probability).

[1] PARK, James D. Using weighted MAX-SAT engines to solve MPE. In: AAAI/IAAI. 2002. p. 682-687.
[2] PIPATSRISAWAT, Knot; DARWICHE, Adnan. Clone: Solving weighted Max-SAT in a reduced search space. In: AI 2007: Advances in Artificial Intelligence: 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, December 2-6, 2007. Proceedings 20. Springer Berlin Heidelberg, 2007. p. 223-233.

Applications

Problem: **Computing MPEs in two-layer BNs with large domains**

Required operations: Conditioning *(CD)*, and weighted minimum cardinality *(wMinCard)*



Experimental results

#nodes	density	domain size	**nwDNNF circuits (Bella)**			decision-DNNF circuits (D4)		
			time (s)	size	#	time (s)	size	#
5 : 5	100%	7	**108**	**1 909 132**	**1**	590	6 770 407	1
		8	**459**	**4 209 175**	**1**	3 296	15 836 293	1
		9	**1 884**	**8 471 321**	**1**	---	---	0
	80%	13	**664**	**19 509 851**	**10**	6 625	34 465 982	1
		14	**1 492**	**32 614 471**	**10**	---	---	0
		15	**3 255**	**49 867 671**	**10**	---	---	0
	60%	20	**856**	**29 159 870**	**10**	3 255	26 635 851	8
		25	**2 027**	**34 843 482**	**10**	---	---	0
		28	**3 796**	**52 154 815**	**10**	---	---	0

The compilation times (in seconds), the circuit sizes, and the number of successfully compiled instances - maximum is ten (resp. one) for 60% and 80% densities (resp. 100% density).

Main contributions

The main contributions of this paper are as follows:

1) We introduced pwDNNF and nwDNNF circuits and studied them in terms of the knowledge compilation map.

2) We related the new circuit types to wDNNF and DNNF circuits with respect to succinctness.

3) We showed a new caching scheme that exploits isomorphism and a new compilation method based on that caching scheme. **(not covered in this presentation)**

4) We presented our extended variant of Bella for pwDNNF and nwDNNF circuits.

5) We demonstrated that nwDNNF circuits are suitable for computing MPEs in two-layer BNs with large domains.