Learning Probabilistic Sentential Decision Diagrams by Sampling

KDMiLe 2020

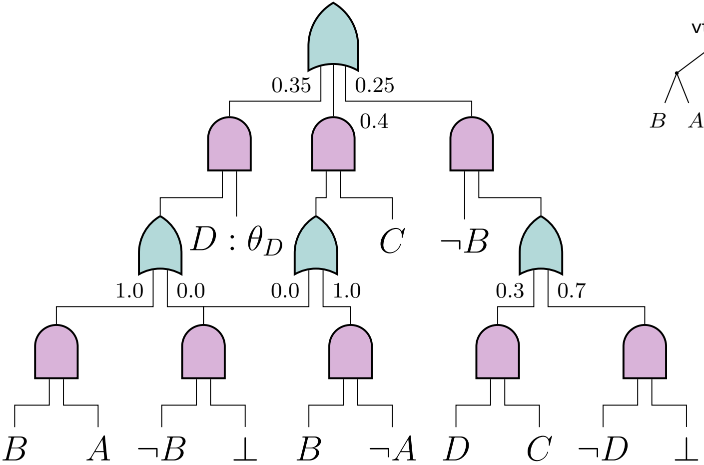
Renato Geh[1] Denis Mauá[1] Alessandro Antonucci[2]

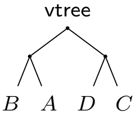
[1]Institute of Mathematics and Statistics, University of São Paulo, Brazil
`{renatolg,ddm}@ime.usp.br`

[2]Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Switzerland
`alessandro@idsia.ch`

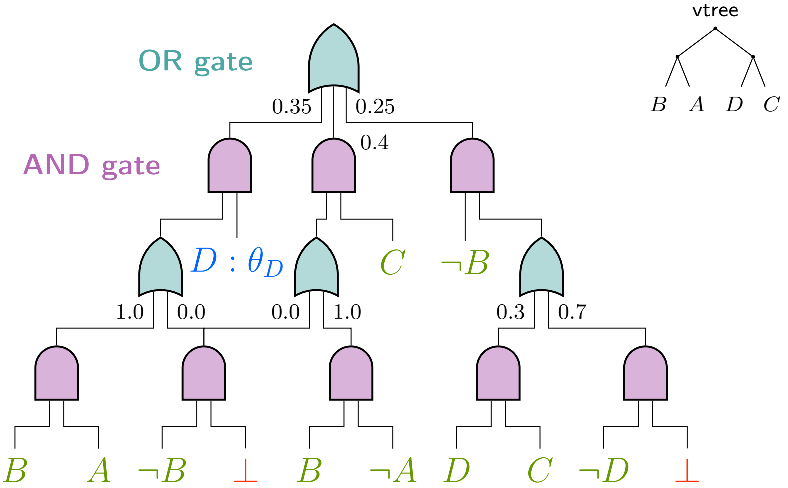
Background

Probabilistic Sentential Decision Diagrams (PSDDs)



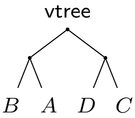


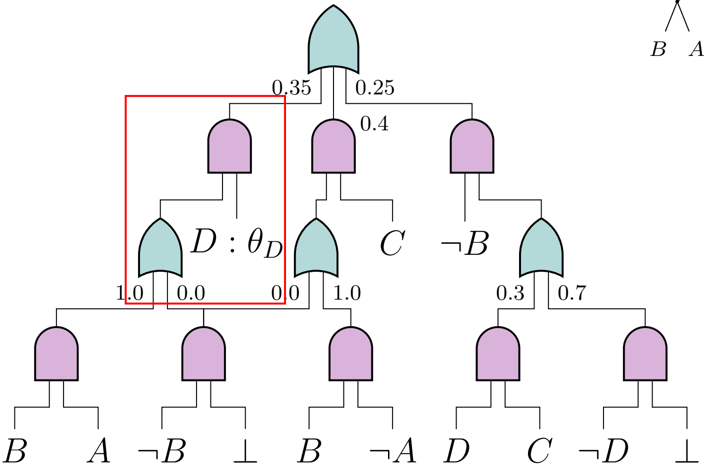
Kisa et al. 2014



Leaves are either literals, constants or Bernoulli distributions.

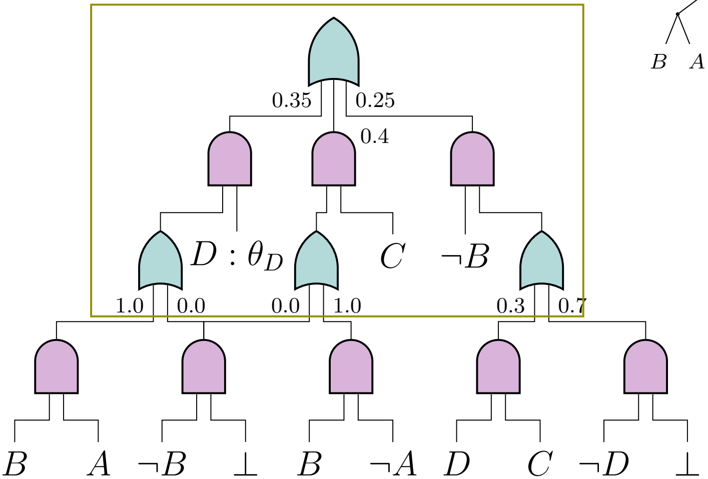
Elements...

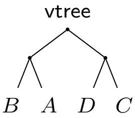




$$\underbrace{(A \wedge B)}_{\text{prime}} \wedge \underbrace{(D \vee \neg D)}_{\text{sub}}$$

Partition...





$$\underbrace{(A \wedge B)}_{e_1} \vee \underbrace{((\neg A \wedge B) \wedge C)}_{e_2} \vee \underbrace{((C \wedge D) \wedge \neg B)}_{e_3}$$

Related Works

LearnPSDD

Given. A vtree and a circuit.

Idea. Learn a PSDD as an expansion of initial circuit.

How?

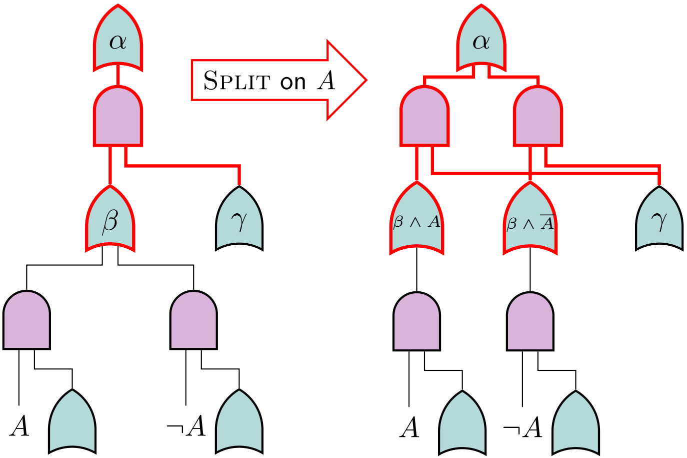
1. Recursively apply small changes;
2. Evaluate score;
3. Greedily accept changes.

$$\text{Score}(\mathcal{S}, \mathcal{S}'|D) = \frac{\log p(\mathcal{S}'|D) - \log p(\mathcal{S}|D)}{|\mathcal{S}'| - |\mathcal{S}|}$$

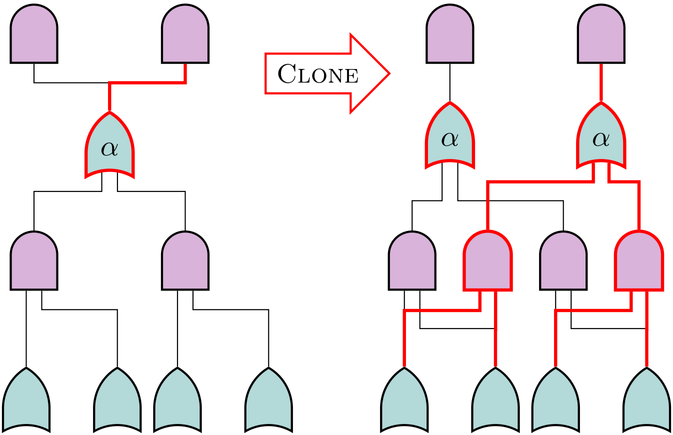
What are these "small" changes?

Liang, Bekker, and G. V. d. Broeck 2017

SPLIT an element...



CLONE a partition...



Strudel

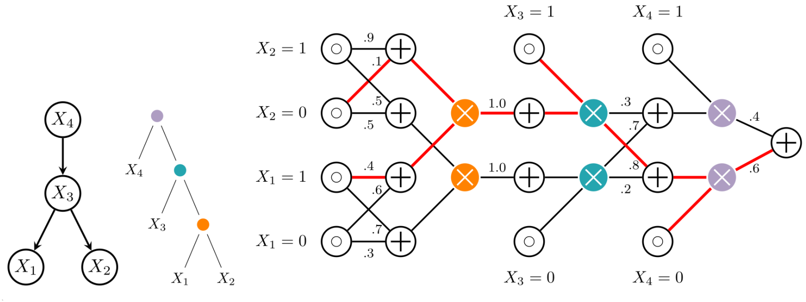
Given. A Chow-Liu Tree (CLT).

Idea. Learn a PSDD and its vtree.

How?

1. Learn a CLT.
2. Extract a vtree from CLT.
3. Compile CLT into circuit.
4. "Grow" circuit with SPLIT.

Dang, Vergari, and G. v. d. Broeck 2020



Dang, Vergari, and G. v. d. Broeck 2020

Learning by Sampling

Monte-Carlo Structure Learning

Previous attempts *grew* or *compiled* existing models.

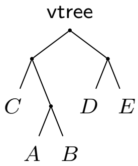
Instead, we want to *build* a circuit solely from knowledge.

Our approach. Start off with a logic formula:

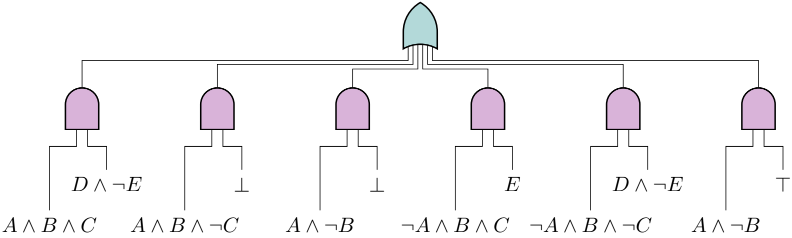
$$\phi(A,B,C,D,E)=(B\land C\land((A\land D\land\neg E)\lor(\neg A\land E)))\lor(\neg A\land((\neg C\land D\land\neg E)\lor\neg B))$$

Recursively decompose ϕ top-down through subsequent partitions.

How to decompose formula into elements...



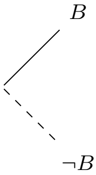
$$(B \wedge C \wedge ((A \wedge D \wedge \neg E) \vee (\neg A \wedge E))) \vee (\neg A \wedge ((\neg C \wedge D \wedge \neg E) \vee \neg B))$$



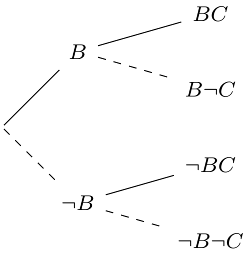
...while ensuring primes are exhaustive and mutually exclusive?

Given a prime ordering, such as $\mathbf{O} = (B, C, A)$, return a set of primes.

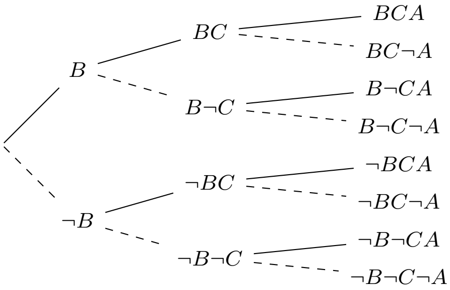
Given a prime ordering, such as $\mathbf{O} = (B, C, A)$, return a set of primes.



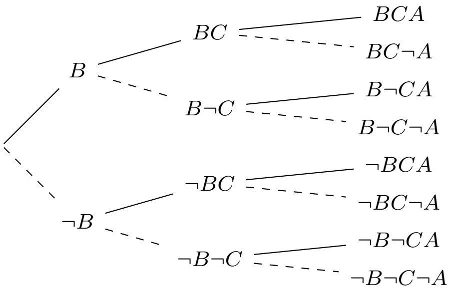
Given a prime ordering, such as $\mathbf{O} = (B, C, A)$, return a set of primes.



Given a prime ordering, such as $\mathbf{O} = (B, C, A)$, return a set of primes.



Given a prime ordering, such as $\mathbf{O} = (B, C, A)$, return a set of primes.



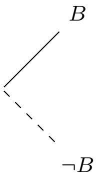
This gives an exponential number of elements!

Instead, let's "forget" a variable when it does not "matter", i.e. when

$$\phi|_x = \phi|_{\neg x}.$$

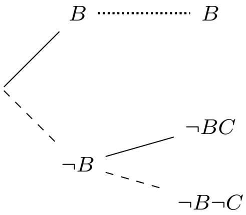
Instead, let's "forget" a variable when it does not "matter", i.e. when

$$\phi|_x = \phi|_{\neg x}.$$



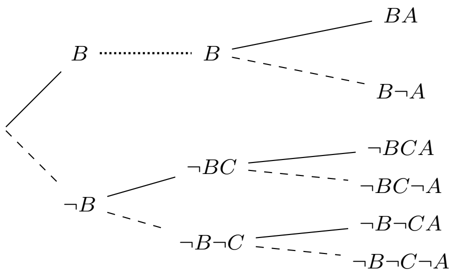
Instead, let's "forget" a variable when it does not "matter", i.e. when

$$\phi|_x = \phi|_{\neg x}.$$



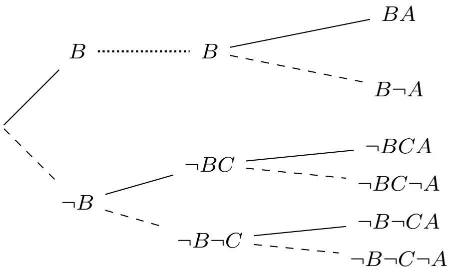
Instead, let's "forget" a variable when it does not "matter", i.e. when

$$\phi|_x = \phi|_{\neg x}.$$



Instead, let's "forget" a variable when it does not "matter", i.e. when

$$\phi|_x = \phi|_{\neg x}.$$



We avoid computing an exponential number of primes!

In other words...

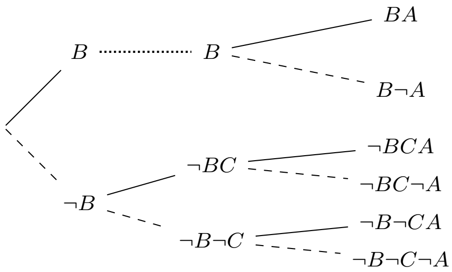
Given.

- A random prime ordering $\mathbf{O} = \{X_1, \ldots, X_n\}$;
- A formula ϕ.

Generate a set of primes $\{p_i\}_{i=1}^{n}$, and then subs $s_i = \phi|_{p_i}$.

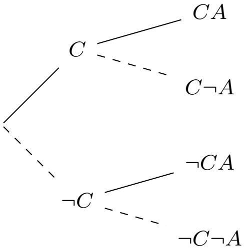
But what if $\phi \equiv \top$ or $\forall X_i \in \mathbf{O}$, $X_i \notin \phi$?

Then we have even more freedom! Stochastically marginalize with some probability p.



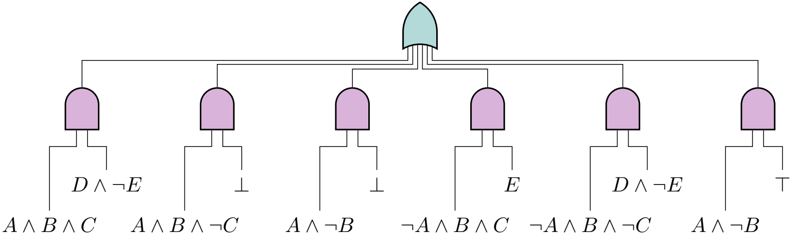
We can marginalize any variable we want, since any operation on ϕ with X_i is idempotent.

Then we have even more freedom! Stochastically marginalize with some probability p.

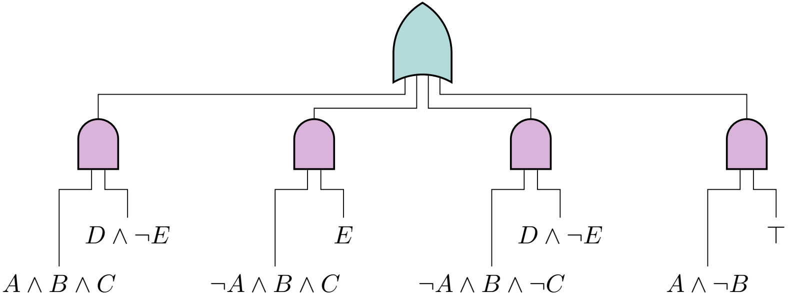


We can marginalize any variable we want, since any operation on ϕ with X_i is idempotent.

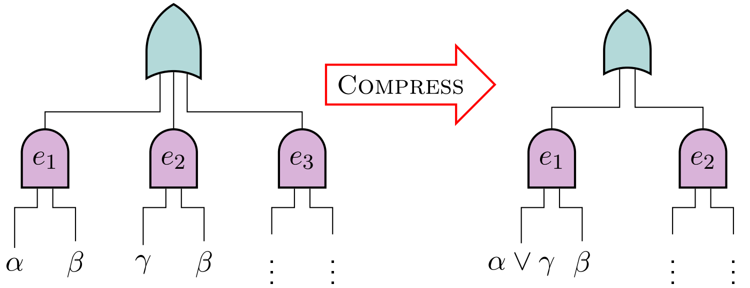
$$(B \wedge C \wedge ((A \wedge D \wedge \neg E) \vee (\neg A \wedge E))) \vee (\neg A \wedge ((\neg C \wedge D \wedge \neg E) \vee \neg B))$$



$$(B \wedge C \wedge ((A \wedge D \wedge \neg E) \vee (\neg A \wedge E))) \vee (\neg A \wedge ((\neg C \wedge D \wedge \neg E) \vee \neg B))$$



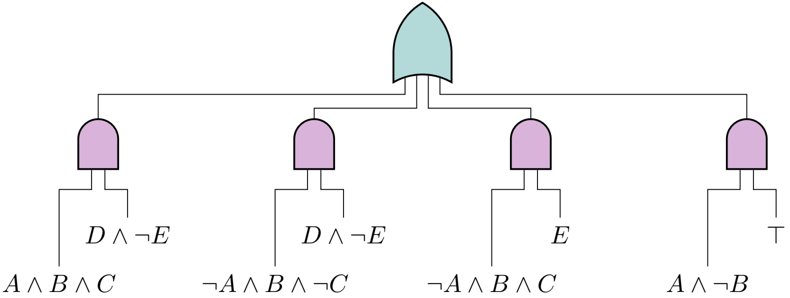
Compression



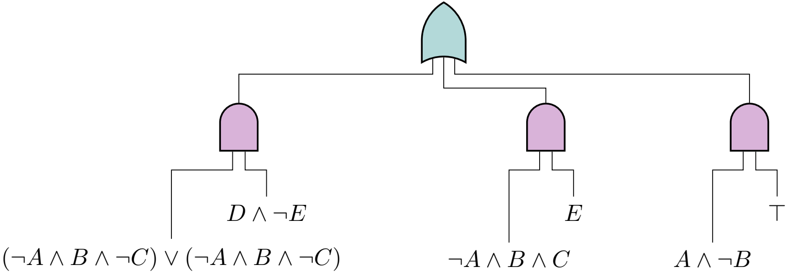
Compress elements (p_1, s) and (p_2, s) by replacing them with $(p_1 \vee p_2, s)$.

Compression generates different distributions consistent with formula.

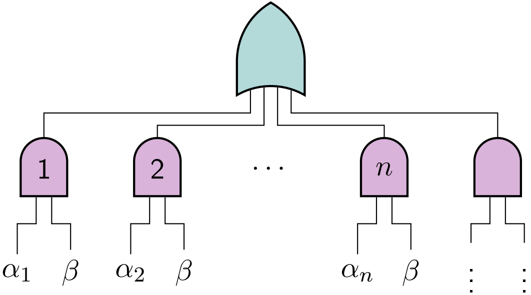
$$(B \wedge C \wedge ((A \wedge D \wedge \neg E) \vee (\neg A \wedge E))) \vee (\neg A \wedge ((\neg C \wedge D \wedge \neg E) \vee \neg B))$$



$$(B \wedge C \wedge ((A \wedge D \wedge \neg E) \vee (\neg A \wedge E))) \vee (\neg A \wedge ((\neg C \wedge D \wedge \neg E) \vee \neg B))$$

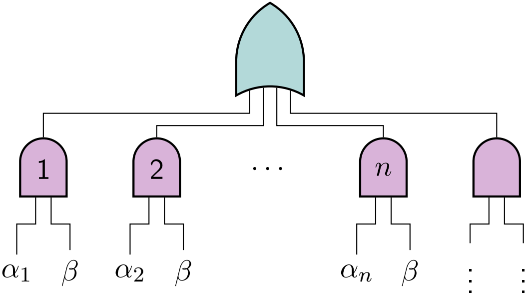


If there are n compressable elements...



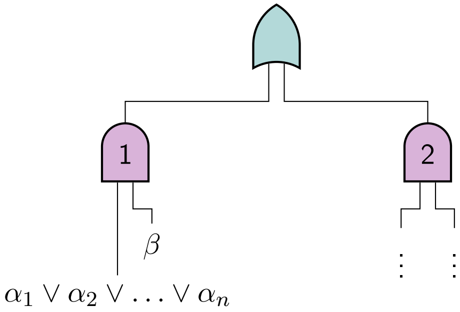
We can generate $\sum_{k=1}^{n} \binom{n}{k}$ different circuits.

If there are n compressable elements...



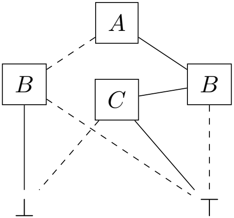
Uniformly sample a circuit from Pascal Triangle's k-th row.

If there are n compressable elements...



But how to efficiently compute potentially complex disjunctions?

Binary Decision Diagrams (BDDs)



Operation	Description	Complexity
Reduce	canonical form of ϕ	$\mathcal{O}(n \cdot \log n)$
Apply	$\phi_1 \oplus \phi_2$	$\mathcal{O}(n_1 \cdot n_2)$
Restrict	$\phi\|_x$	$\mathcal{O}(n \cdot \log n)$
Forget	$\phi\|_x \vee \phi\|_{\neg x}$	$\mathcal{O}(n^2)$

$$\phi(A, B, C) = (A \vee \neg B) \wedge (\neg B \vee C)$$

Results

Likelihood

Data	#vars	Worst	Best	Average	LSPN-Opt	LSPN-CV	CLT	CNet
1	10	-449.89	**-413.51**	-415.60	-422.37	-444.62	-456.49	-585.23
2	15	-745.20	**-693.51**	-695.82	-695.09	-739.49	-803.41	-1070.29
3	20	-1024.01	-969.51	-971.80	**-959.03**	-1003.93	-1075.31	-1855.05
4	25	-1268.82	-1208.22	-1210.27	**-1185.28**	-1254.47	-1290.94	-2033.58
5	30	-1548.92	**-1440.27**	-1442.58	-1441.90	-1543.35	-1535.54	-2048.16
6	100	-5169.14	-4995.53	-4997.83	**-4958.06**	-5232.04	-5712.73	-10326.73
7	100	-5329.17	-5153.51	-5155.82	**-4900.65**	-5206.54	-5710.64	-9948.88
8	14	-472.15	**-423.32**	-425.62	-490.21	-506.62	-486.06	-601.31

Data	Logic formula
1	$\phi_1 = (X_1 \wedge X_3) \vee (X_4 \wedge \neg X_2) \vee (X_5 \wedge \neg X_{10})$
2	$\phi_2 = (X_1 \wedge X_3) \vee (X_4 \wedge \neg X_2) \vee (X_5 \wedge \neg X_{10}) \vee (X_{12} \wedge \neg X_{13} \wedge X_{15} \wedge \neg X_{14})$
3	$\phi_3 = (X_1 \wedge X_3) \vee (X_4 \wedge \neg X_2) \vee (X_5 \wedge \neg X_{10}) \vee (X_{12} \wedge \neg X_{13} \wedge X_{15} \wedge \neg X_{14})$
4	$\phi_4 = (X_1 \wedge X_3) \vee (X_4 \wedge \neg X_2) \vee (X_5 \wedge \neg X_{10}) \vee (X_{12} \wedge \neg X_{13} \wedge X_{15} \wedge \neg X_{14})$
5	$\phi_5 = (X_2 \vee X_{30}) \wedge (\neg X_{15} \vee \neg X_{10}) \wedge (\neg X_{25} \vee X_5 \vee X_{15} \vee \neg X_1) \wedge (X_1 \vee X_{15} \vee \neg X_{30})$
6	$\phi_6 = (X_{10} \vee X_{30}) \wedge (\neg X_1 \vee X_5) \wedge (\neg X_{10} \vee X_{14} \vee X_{23}) \wedge (X_2 \vee \neg X_{27} \vee X_{35}) \wedge (X_{98} \vee \neg X_{78} \vee \neg X_{27} \vee X_8)$
7	$\phi_7 = \top$
8	$\phi_8 = d_0 \vee d_1 \vee d_2 \vee d_3 \vee d_4 \vee d_5 \vee d_6 \vee d_7 \vee d_8 \vee d_9$

Thank You!

Questions?

References

References I

Dang, Meihua, Antonio Vergari, and Guy van den Broeck (2020). "Strudel: Learning Structured-Decomposable Probabilistic Circuits". In: *Proceedings of the Tenth International Conference on Probabilistic Graphical Models*.

Kisa, Doga et al. (2014). "Probabilistic Sentential Decision Diagrams". In: *Knowledge Representation and Reasoning Conference*. URL: https://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8005.

Liang, Yitao, Jessa Bekker, and Guy Van den Broeck (2017). "Learning the Structure of Probabilistic Sentential Decision Diagrams". In: *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. Ed. by Gal Elidan, Kristian Kersting, and Alexander T. Ihler. AUAI Press. URL: http://auai.org/uai2017/proceedings/papers/291.pdf.