ISIPTA 2019
GHENT
03/07

ALESSANDRO ANTONUCCI,
ALESSANDRO FACCHINI &
LILITH MATTEI

CREDAL SENTENTIAL DECISION DIAGRAMS

- Alessandro Antonucci, a senior researcher in probabilistic graphical models and machine learning

Istituto Dalle Molle di Studi per l’Intelligenza Artificiale

- Alessandro Facchini, a convenience* logician

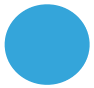
*Concept and formulation by Yoichi Hirai

- Lilith Mattei, research assistant, wannabe PhD student

WHAT ARE CSDD? FIRST SOME ZOOLOGY

WHAT ARE CSDD? FIRST SOME ZOOLOGY

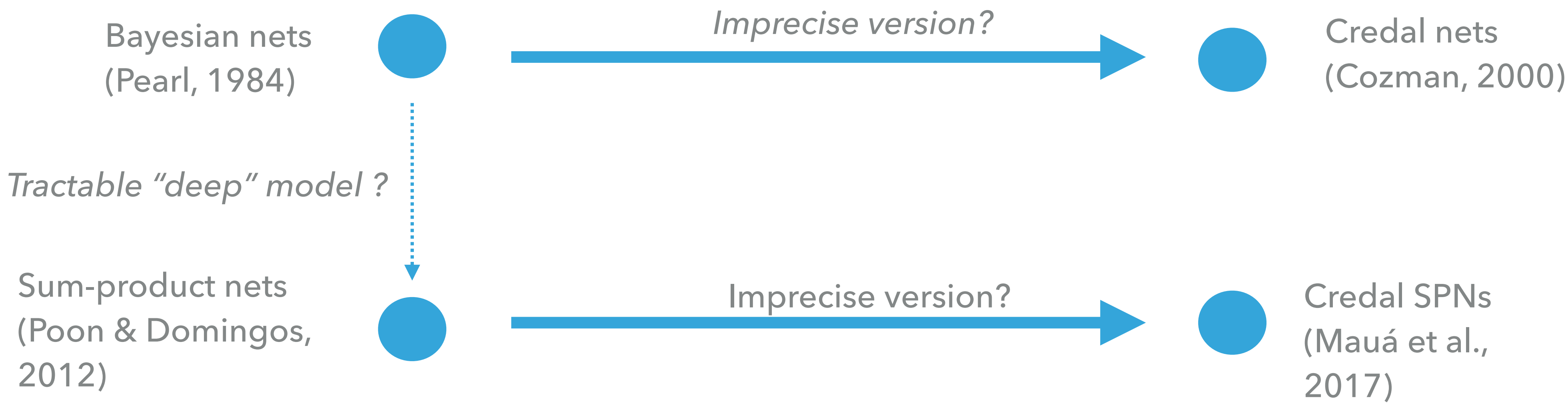
Bayesian nets
(Pearl, 1984)



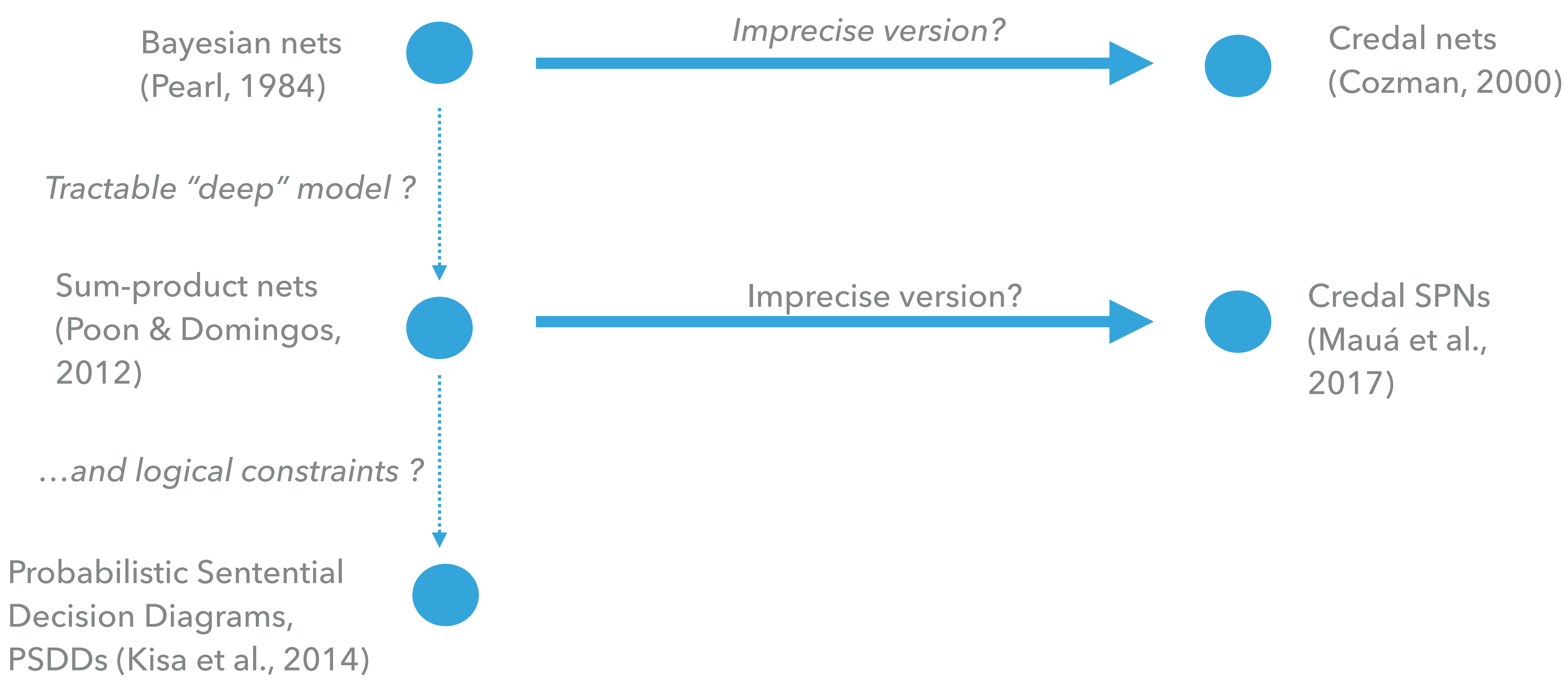
WHAT ARE CSDD? FIRST SOME ZOOLOGY

WHAT ARE CSDD? FIRST SOME ZOOLOGY

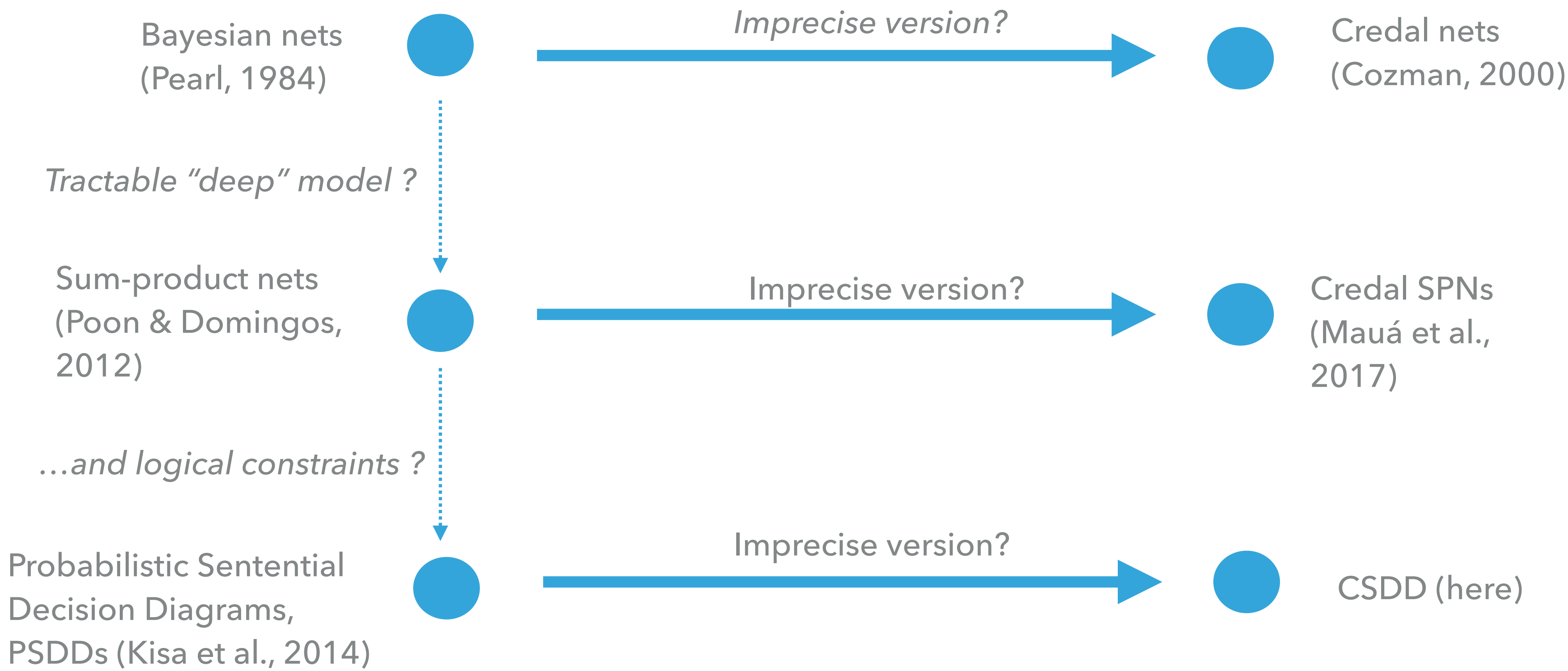
WHAT ARE CSDD? FIRST SOME ZOOLOGY



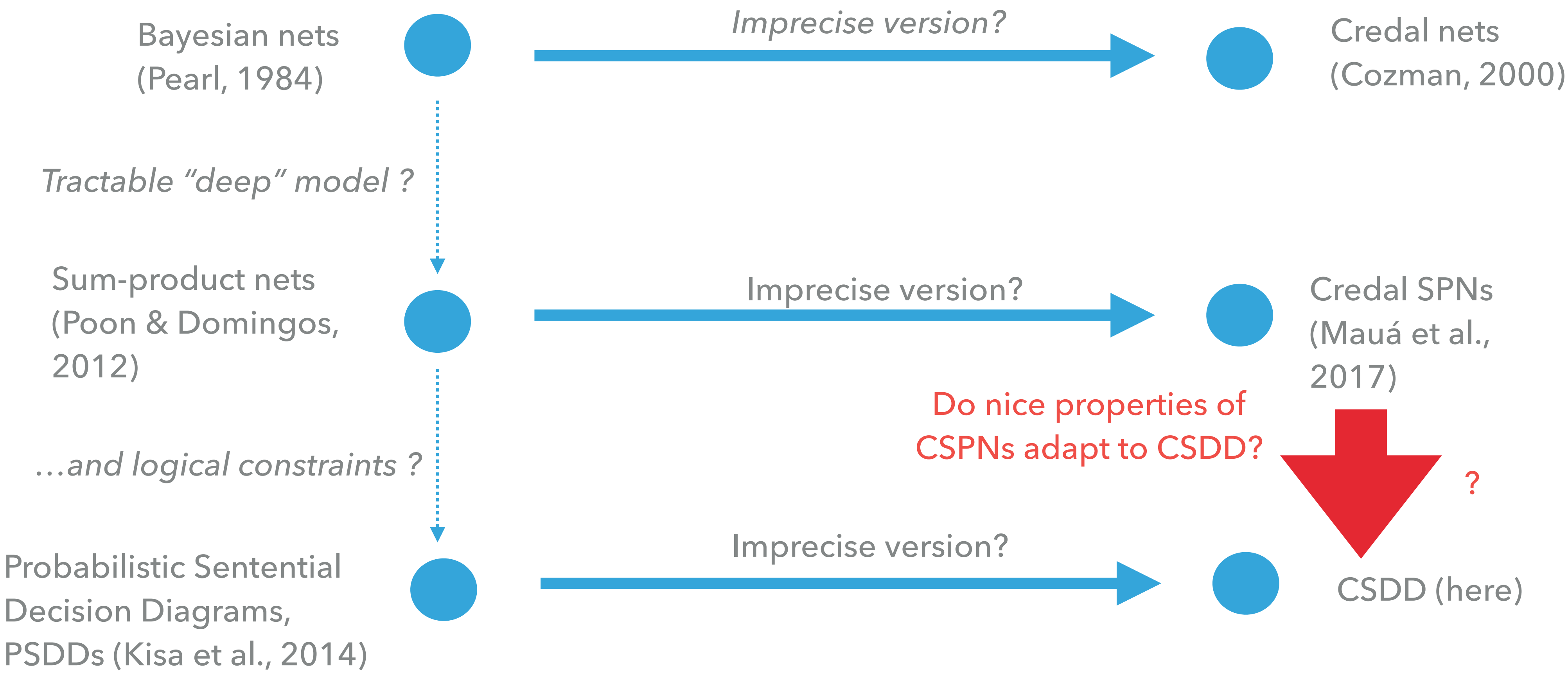
WHAT ARE CSDD? FIRST SOME ZOOLOGY



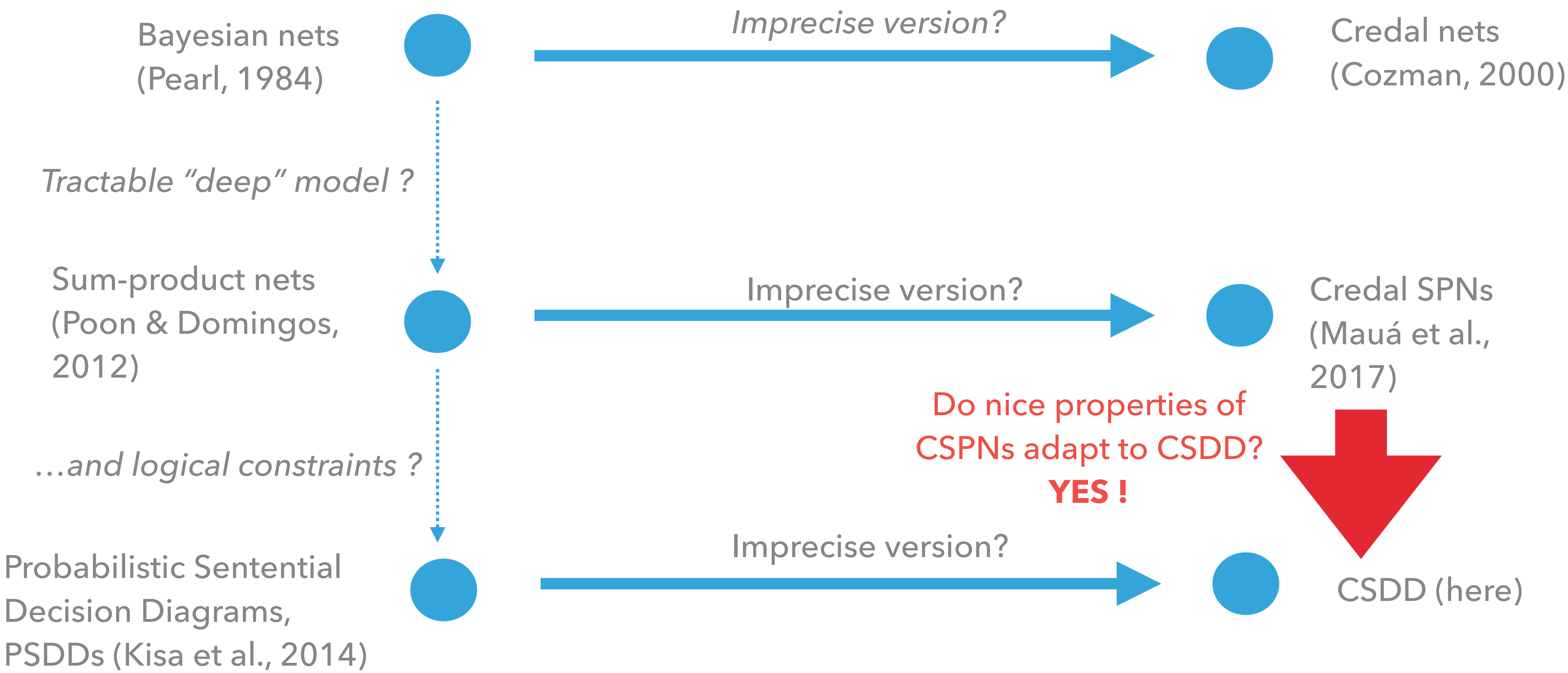
WHAT ARE CSDD? FIRST SOME ZOOLOGY



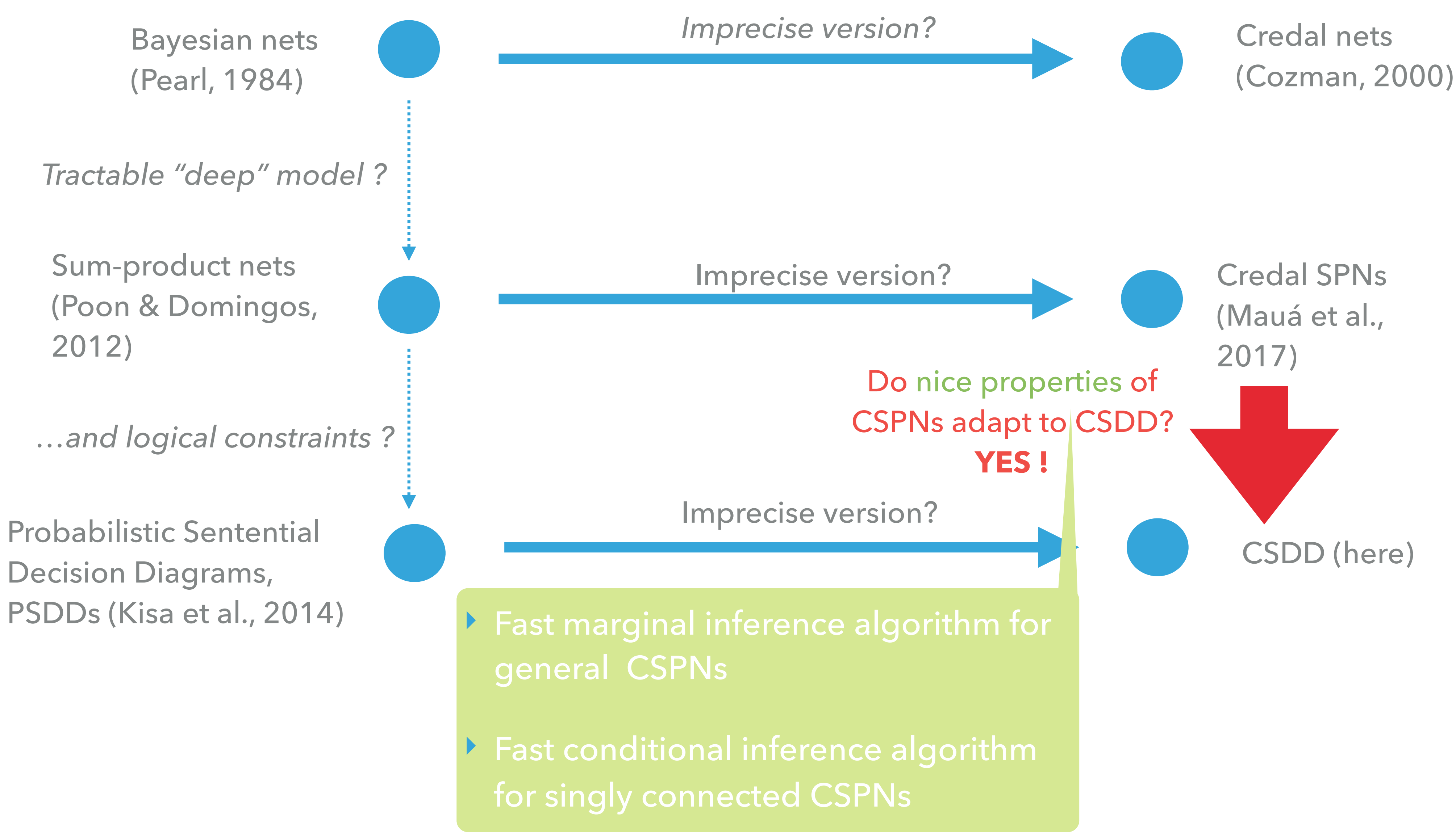
WHAT ARE CSDD? FIRST SOME ZOOLOGY



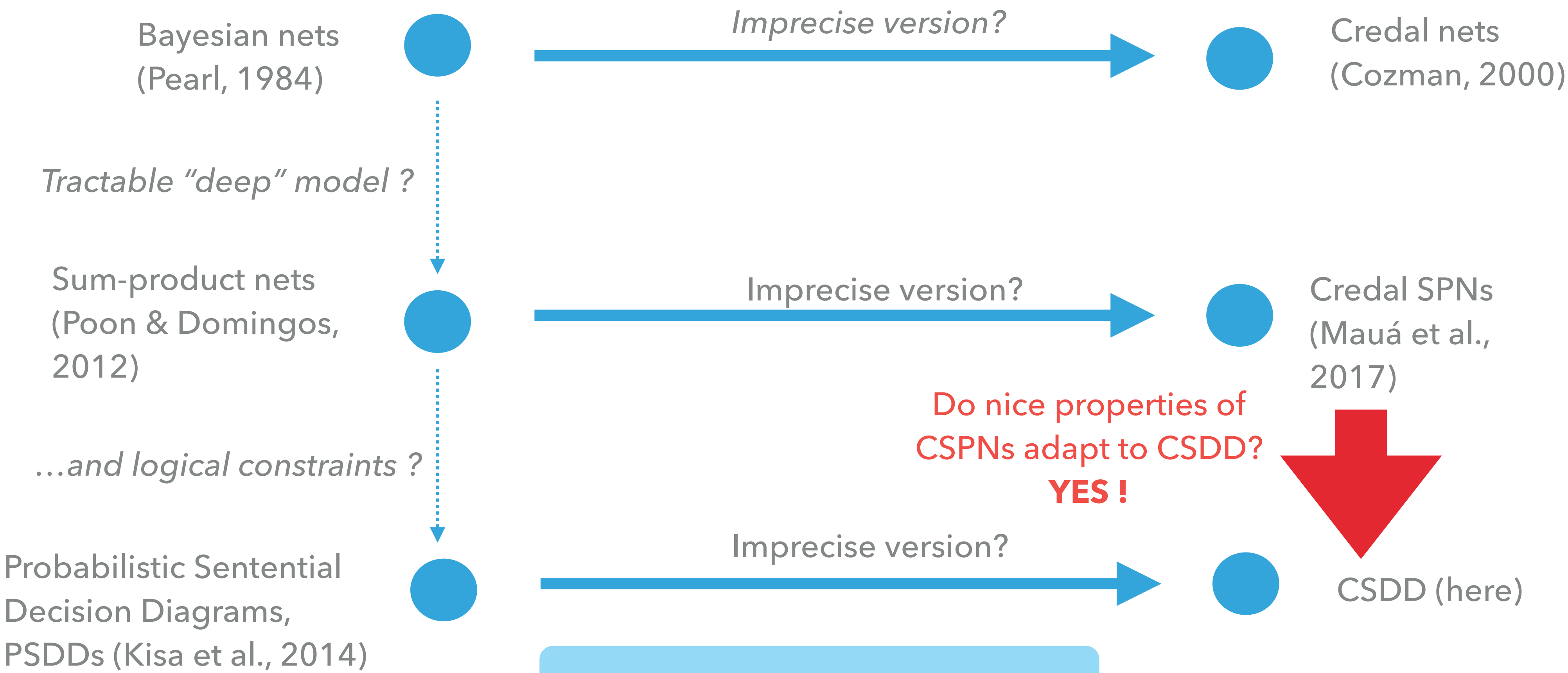
WHAT ARE CSDD? FIRST SOME ZOOLOGY



WHAT ARE CSDD? FIRST SOME ZOOLOGY



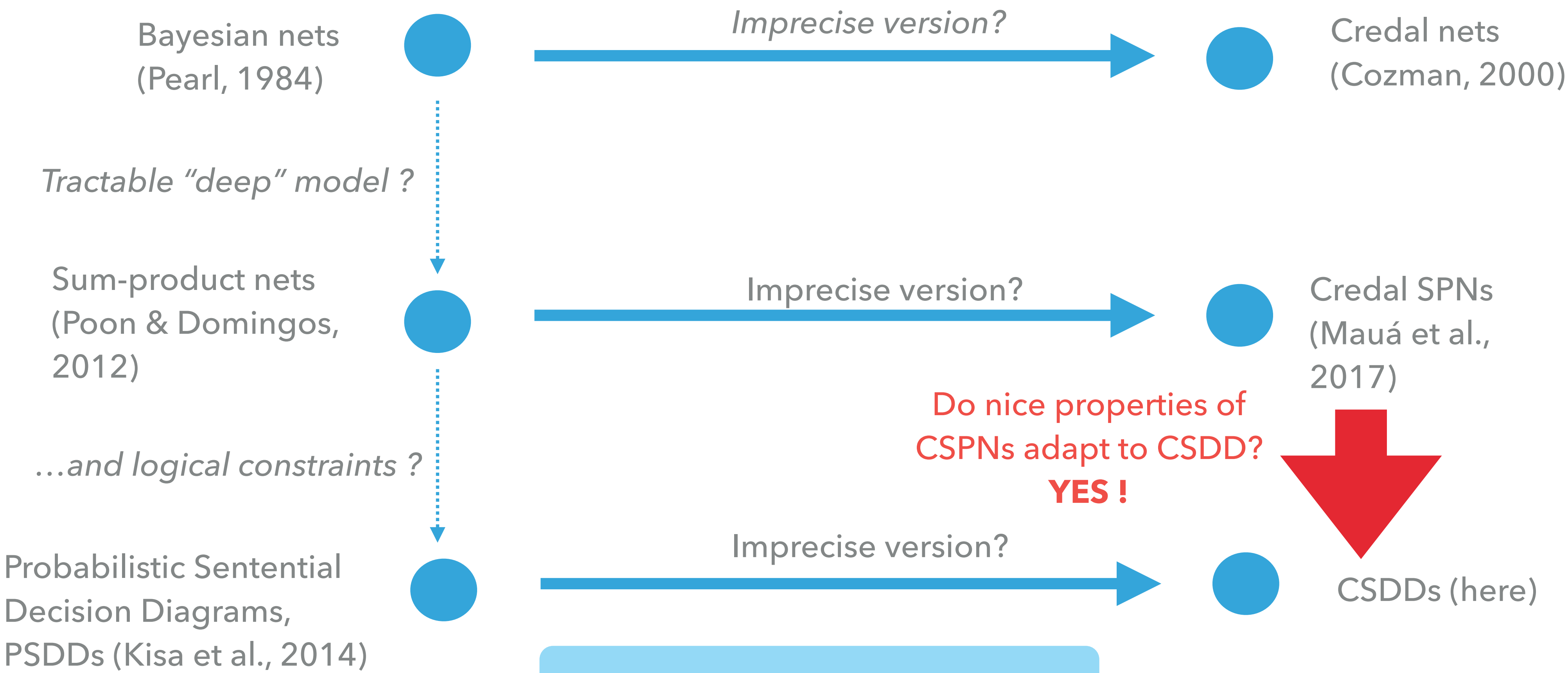
WHAT ARE CSDD? FIRST SOME ZOOLOGY



message of this work:

CSDD's stand to PSDD's as CSPN's stand to SPN's

WHAT ARE CSDD? FIRST SOME ZOOLOGY



message of this work:

CSDD's stand to PSDD's as CSPN's stand to SPN's

…but what are CSDDs ?

A FIRST GLIMPSE TO CSDD

A FIRST GLIMPSE TO CSDD

- CSDD = ***Credal version of Probabilistic Sentential Decision Diagrams***

A FIRST GLIMPSE TO CSDD

- CSDD = ***Credal version of Probabilistic Sentential Decision Diagrams***

    - so, what are PSDDs?

A FIRST GLIMPSE TO CSDD

- CSDD = ***Credal version of Probabilistic Sentential Decision Diagrams***

    - so, what are PSDDs?
    - actually, what are SDDs?

100 STUDENTS ENROLLING IN 4 CLASSES: LOGIC (L), KNOWLEDGE REPRESENTATION (K), PROBABILITY (P), AI (A)

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- 16 joint states
- Three logical constraints

$$(P \vee L), \quad (A \rightarrow P), \quad (K \rightarrow A \vee L)$$

100 STUDENTS ENROLLING IN 4 CLASSES: LOGIC (L), KNOWLEDGE REPRESENTATION (K), PROBABILITY (P), AI (A)

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- 16 joint states
- Three logical constraints

$$\phi := (P \vee L) \wedge (A \rightarrow P) \wedge (K \rightarrow A \vee L)$$

100 STUDENTS ENROLLING IN 4 CLASSES: LOGIC (L), KNOWLEDGE REPRESENTATION (K), PROBABILITY (P), AI (A)

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- 16 joint states
- Three logical constraints

$$\phi := (P \vee L) \wedge (A \rightarrow P) \wedge (K \rightarrow A \vee L)$$

- 7 states not satisfying the logical constraints (hence never observed)

100 STUDENTS ENROLLING IN 4 CLASSES: LOGIC (L), KNOWLEDGE REPRESENTATION (K), PROBABILITY (P), AI (A)

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- 16 joint states
- Three logical constraints

$$\phi := (P \vee L) \wedge (A \rightarrow P) \wedge (K \rightarrow A \vee L)$$

- 7 states not satisfying the logical constraints (hence never observed)
- 1 state logically possible but never observed

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$

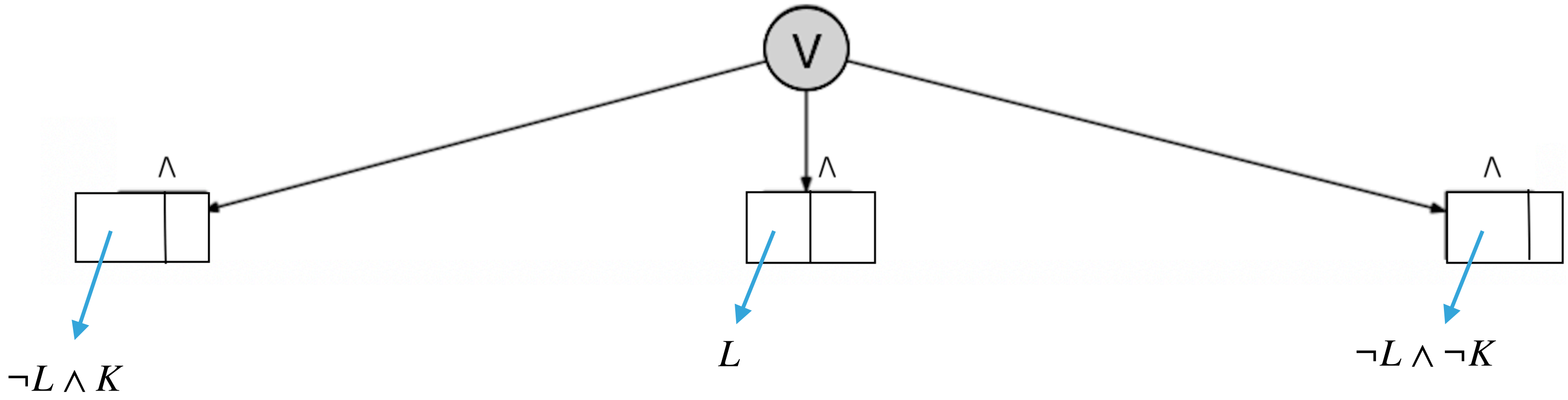
MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$

Partition

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

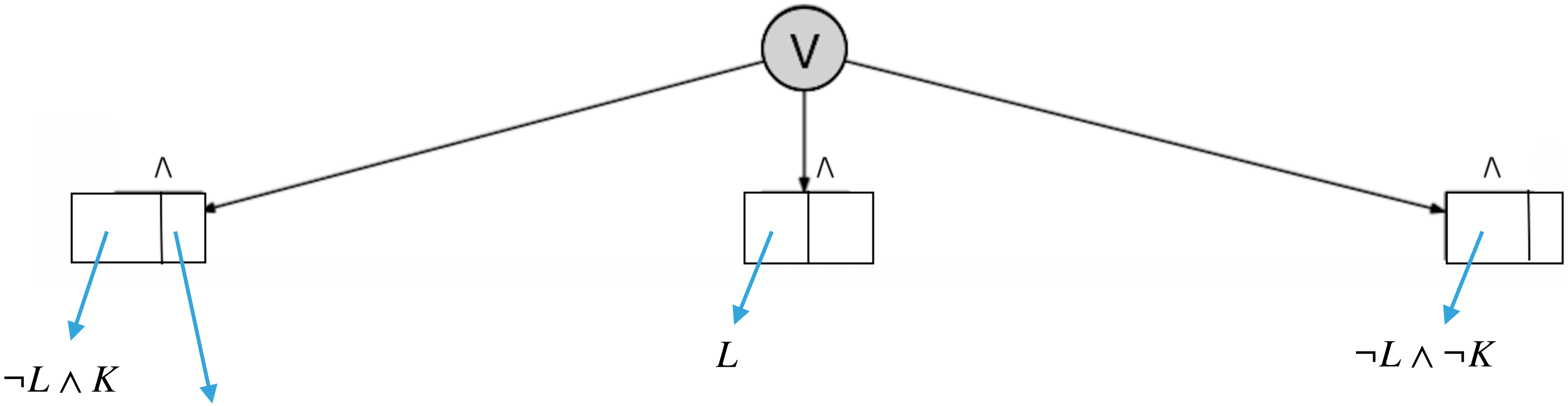
- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g., $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$



$$\phi = (P \vee L) \wedge (P \vee \neg A) \wedge (A \vee L \vee \neg K)$$

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g.,
  $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$

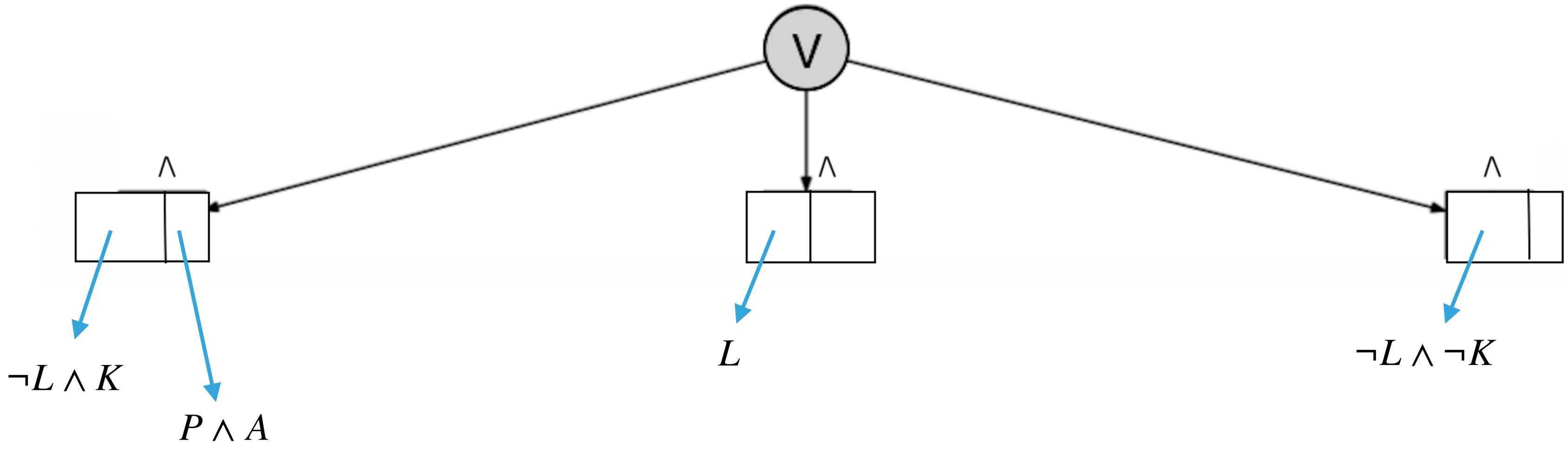


What does ϕ becomes when $L = \bot, K = \top$?

$$\phi = (P \vee L) \wedge (P \vee \neg A) \wedge (A \vee L \vee \neg K)$$

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

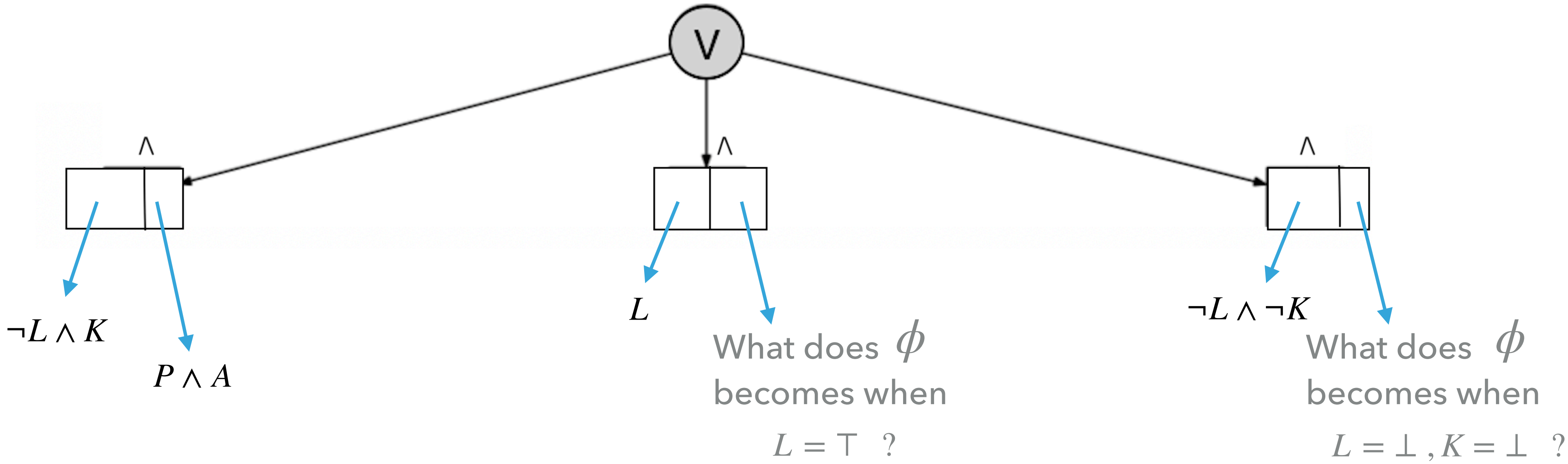
- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g.,
  $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$



$$\phi = (P \vee L) \wedge (P \vee \neg A) \wedge (A \vee L \vee \neg K)$$

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

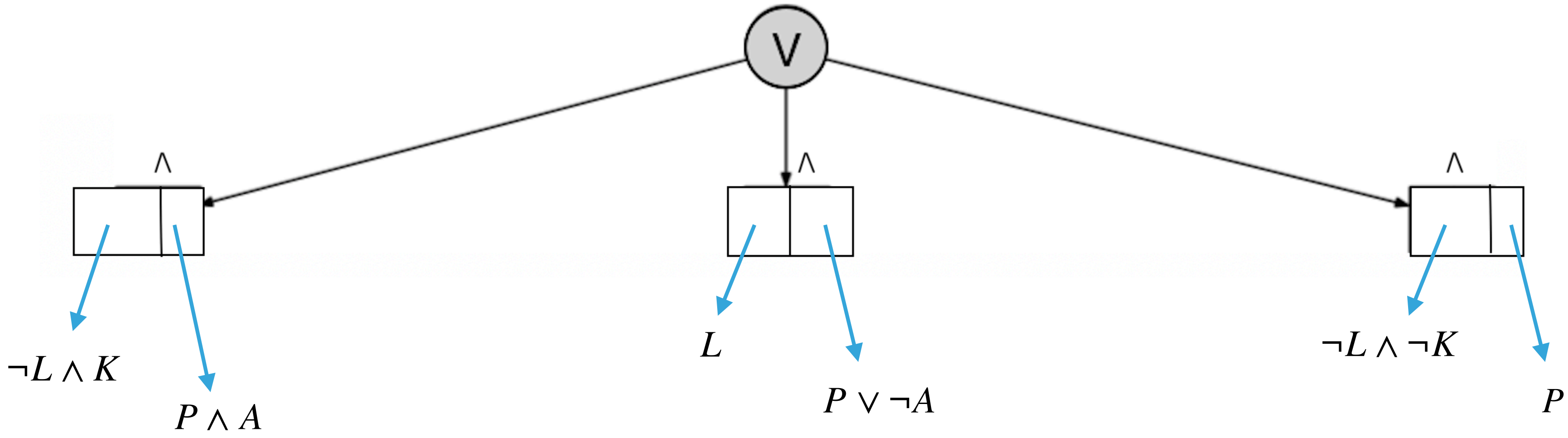
- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g.,
  $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$



$$\phi = (P \vee L) \wedge (P \vee \neg A) \wedge (A \vee L \vee \neg K)$$

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g.,
  $\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$

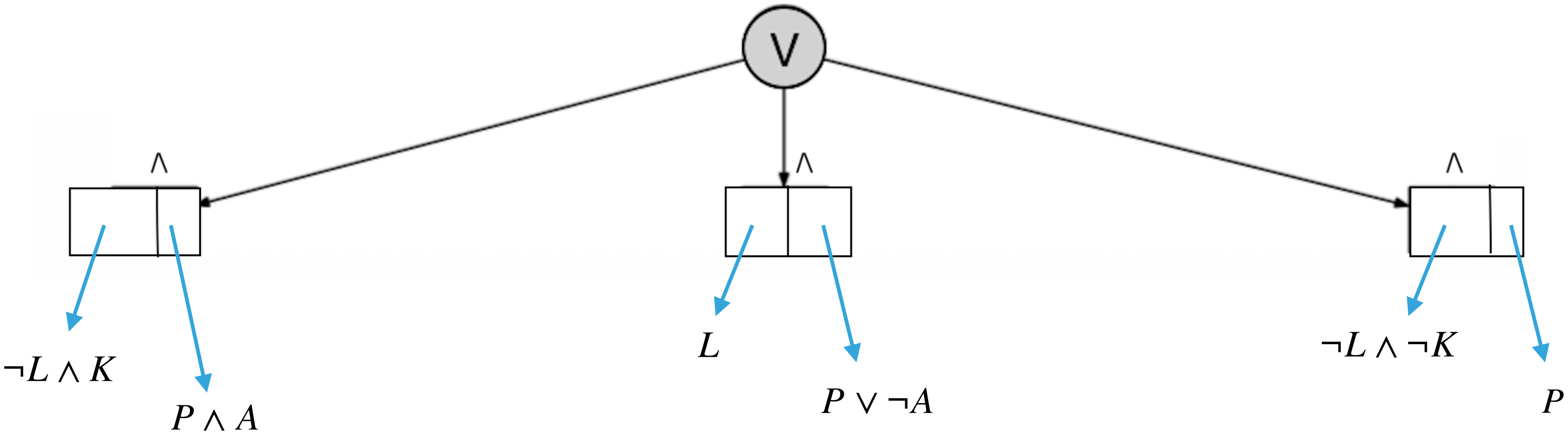


$$\phi = (P \vee L) \wedge (P \vee \neg A) \wedge (A \vee L \vee \neg K)$$

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g.,

$$\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$$

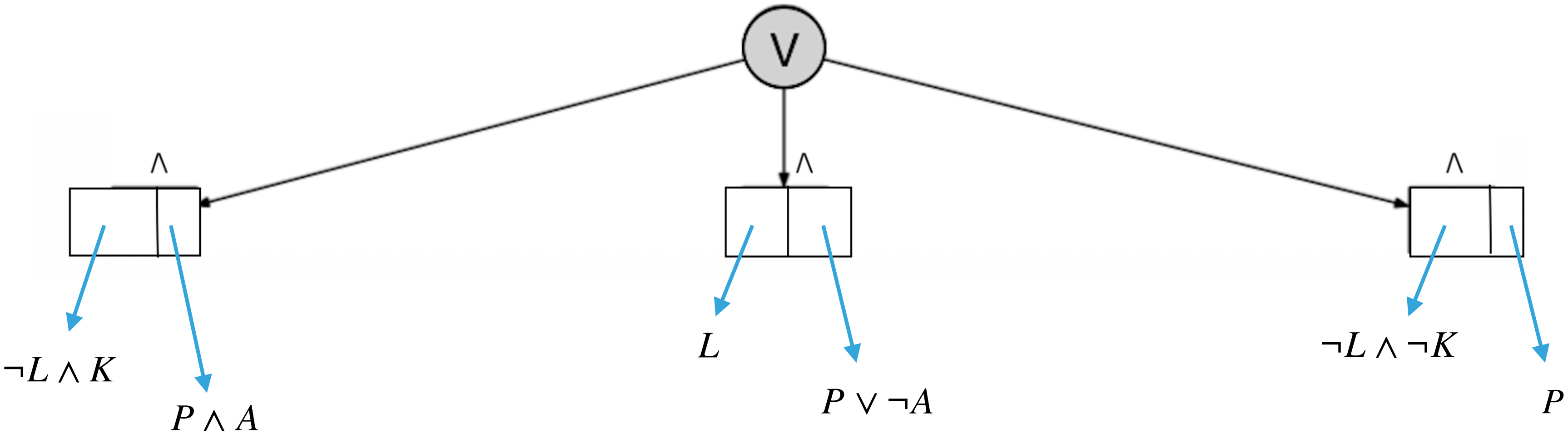


- $(\neg L \wedge K) \wedge (P \wedge A) \bigvee L \wedge (P \vee \neg A) \bigvee (\neg L \wedge \neg K) \wedge P = \phi$

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

- **A Sentential Decision Diagram** representing ϕ is a "deterministic" logic circuit
- take a subset of the variables, form a partition of the tautology, e.g.,

$$\top = (\neg L \wedge K) \vee L \vee (\neg L \wedge \neg K)$$

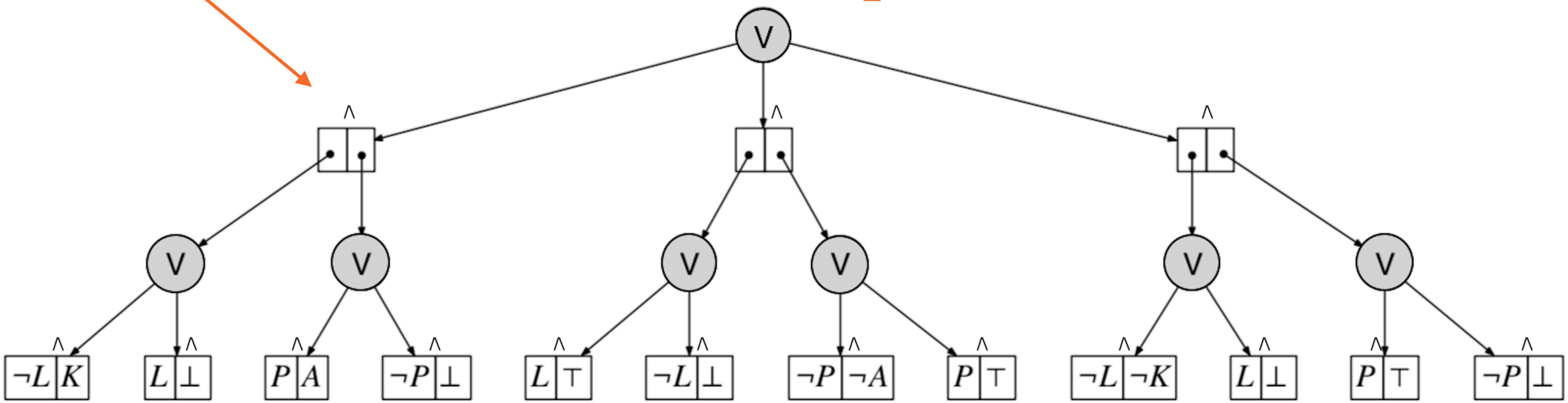


- $(\neg L \wedge K) \wedge (P \wedge A) \bigvee L \wedge (P \vee \neg A) \bigvee (\neg L \wedge \neg K) \wedge P = \phi$
- **Proceed recursively…**

MODELING CONSTRAINTS WITH CIRCUITS: SDD'S (DARWICHE 2011)

Paired boxes: AND gates

Decision nodes: OR gates



$$(\neg L \wedge K \bigvee L \wedge \bot) \bigwedge (P \wedge A \bigvee \neg P \wedge \bot)$$

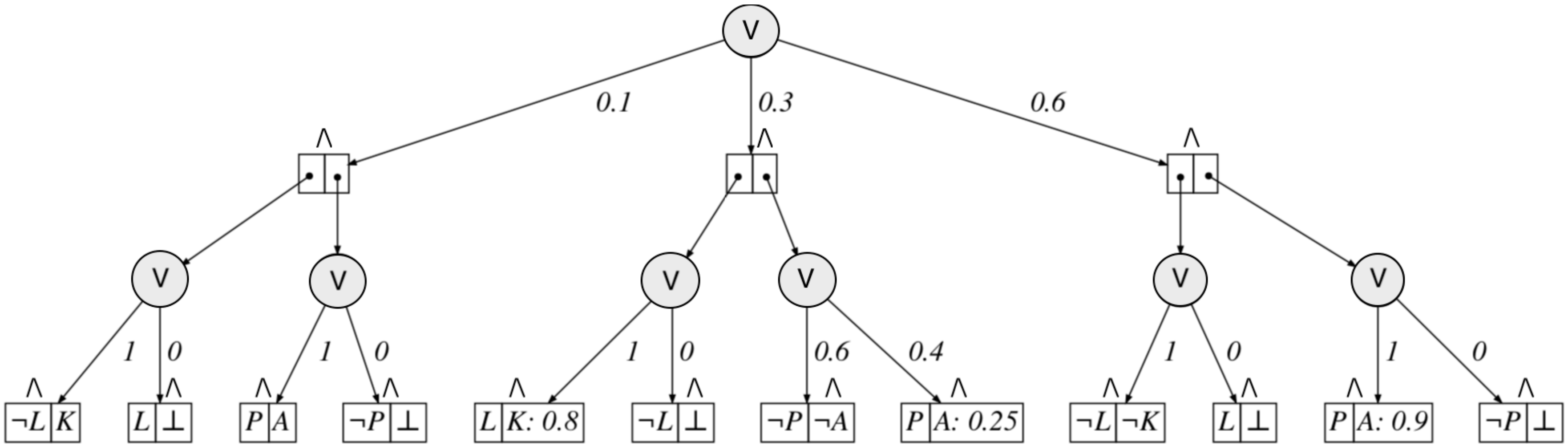
$$\bigvee (L \wedge \top \bigvee \neg L \wedge \bot) \bigwedge (\neg P \wedge \neg A \bigvee P \wedge \top)$$

$$\bigvee (\neg L \wedge \neg K \bigvee L \wedge \bot) \bigwedge (P \wedge \top \bigvee \neg P \wedge \bot)$$

$$= \phi$$

MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

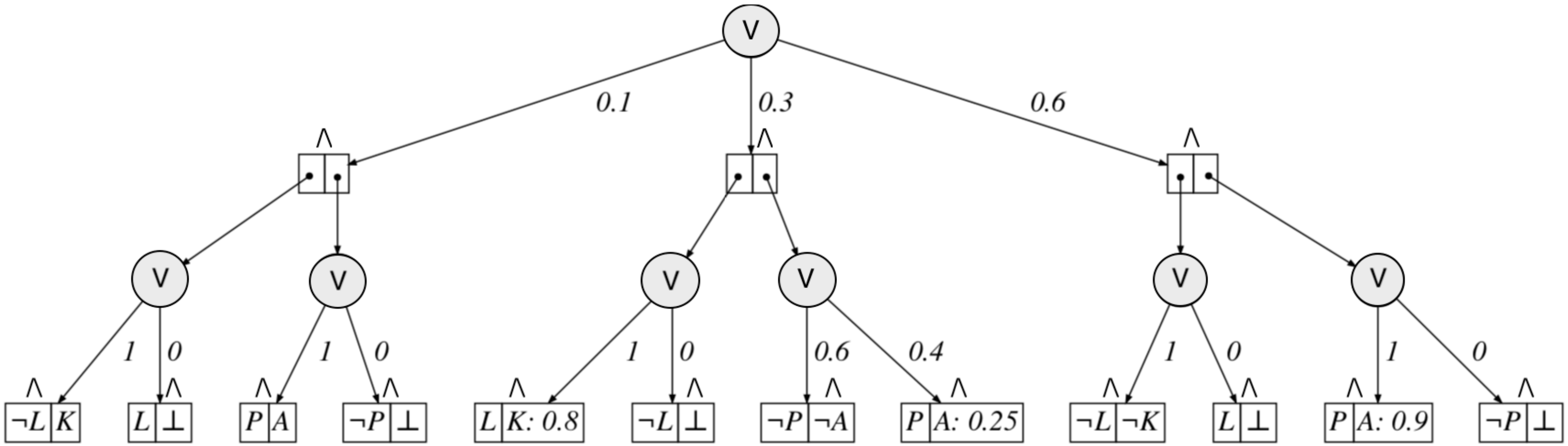
- **A Probabilistic Sentential Decision Diagrams (PSDDs)** for. ϕ is a parametrized SDD:



MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

- **A Probabilistic Sentential Decision Diagrams** (**PSDDs**) for. ϕ is a parametrized SDD:

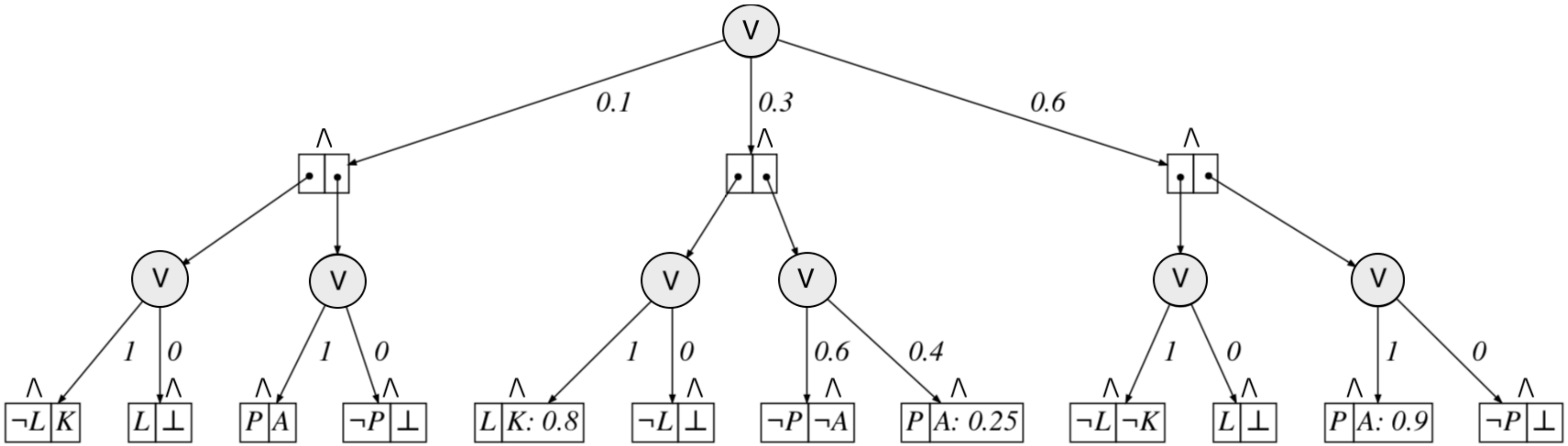
- Parameters learned from data



MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

- **A Probabilistic Sentential Decision Diagrams (PSDDs)** for. ϕ is a parametrized SDD:

- Parameters learned from data

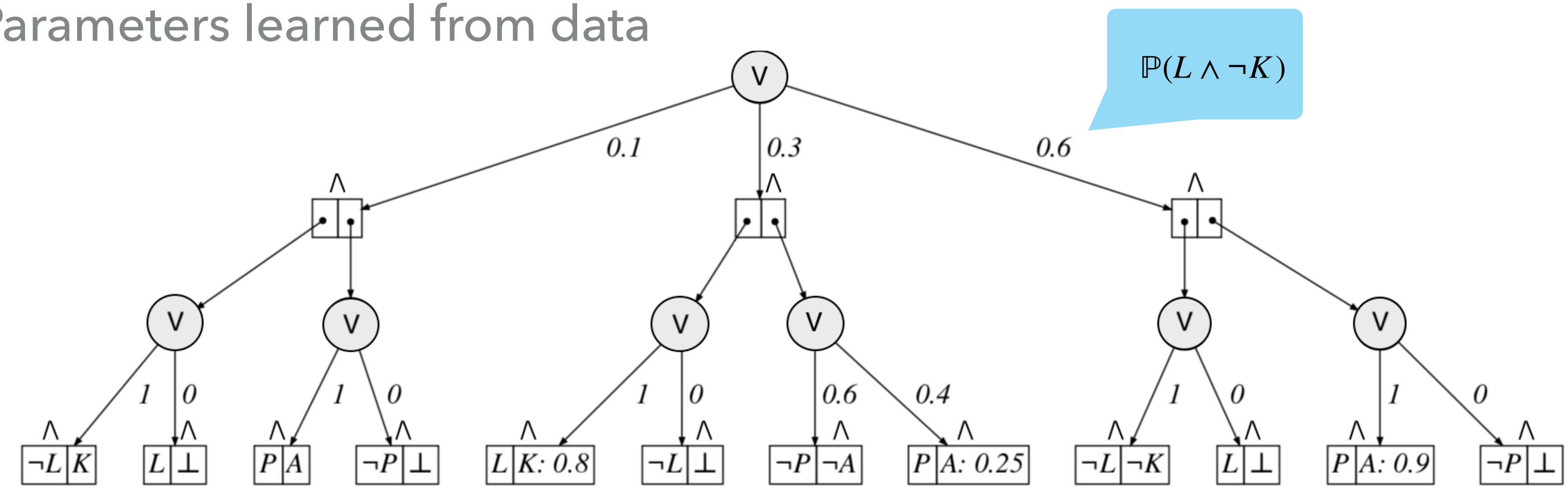


- Inducing a joint probability $\mathbb{P}(A, L, P, K)$

MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

- **A Probabilistic Sentential Decision Diagrams** (**PSDDs**) for. ϕ is a parametrized SDD:

- Parameters learned from data

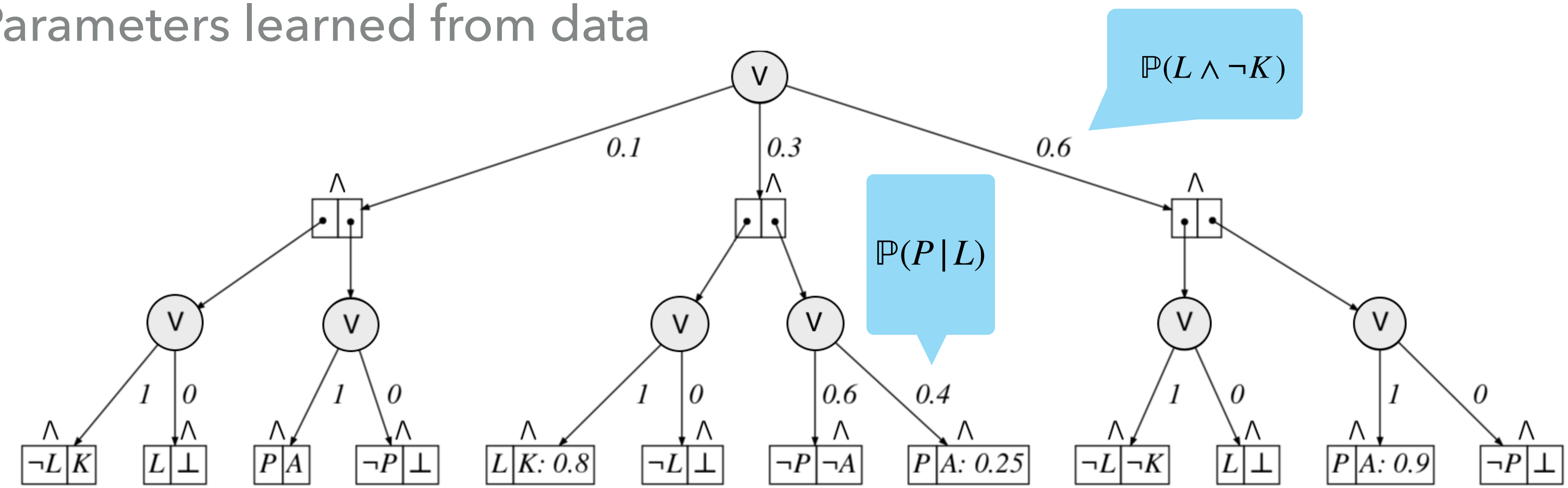


- Inducing a joint probability $\mathbb{P}(A, L, P, K)$

MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

- **A Probabilistic Sentential Decision Diagrams** (**PSDDs**) for. ϕ is a parametrized SDD:

- Parameters learned from data

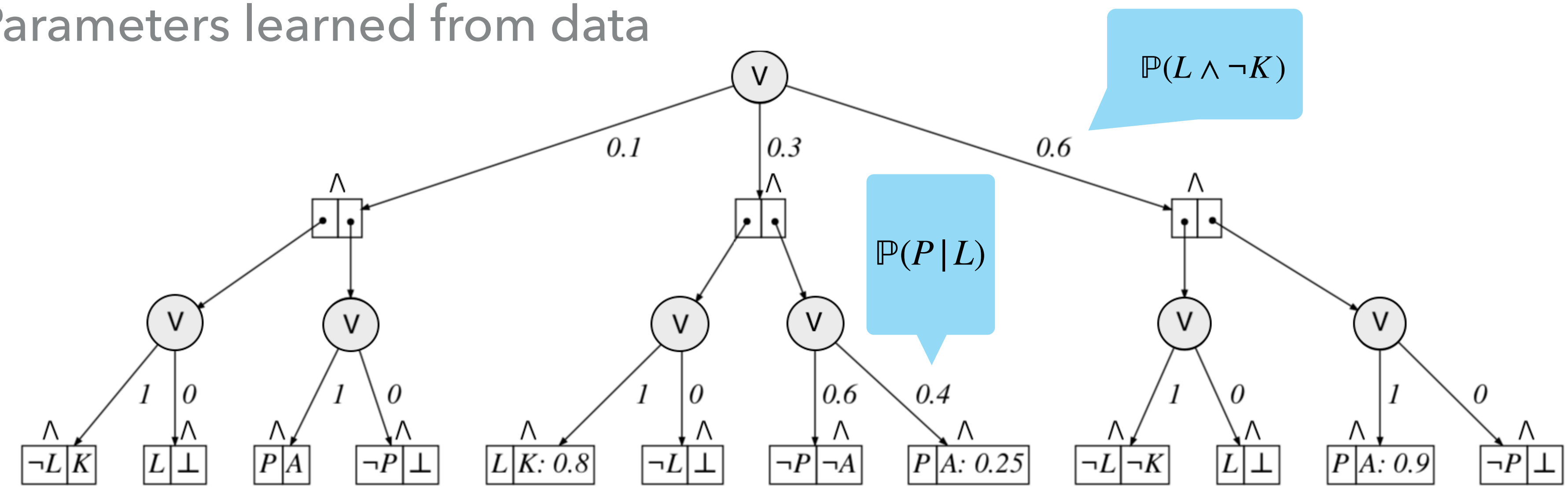


- Inducing a joint probability $\mathbb{P}(A, L, P, K)$

MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

- **A Probabilistic Sentential Decision Diagrams** (**PSDDs**) for. ϕ is a parametrized SDD:

- Parameters learned from data



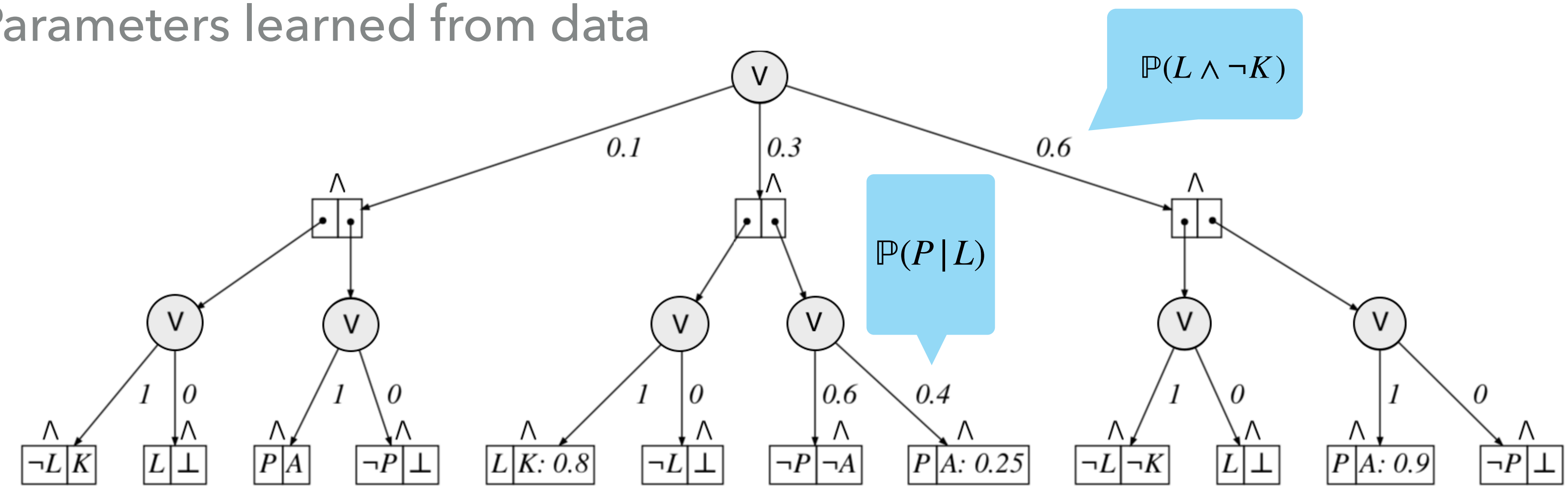
- Inducing a joint probability $\mathbb{P}(A, L, P, K)$

- context-specific independences wrt $\mathbb{P}$ derived from the structure

MODELING DATA + CONSTRAINTS WITH CIRCUITS: PSDD'S (KISA, 2014)

- **A Probabilistic Sentential Decision Diagrams** (**PSDDs**) for. ϕ is a parametrized SDD:

- Parameters learned from data



- Inducing a joint probability $\mathbb{P}(A, L, P, K)$

- context-specific independences wrt $\mathbb{P}$ derived from the structure

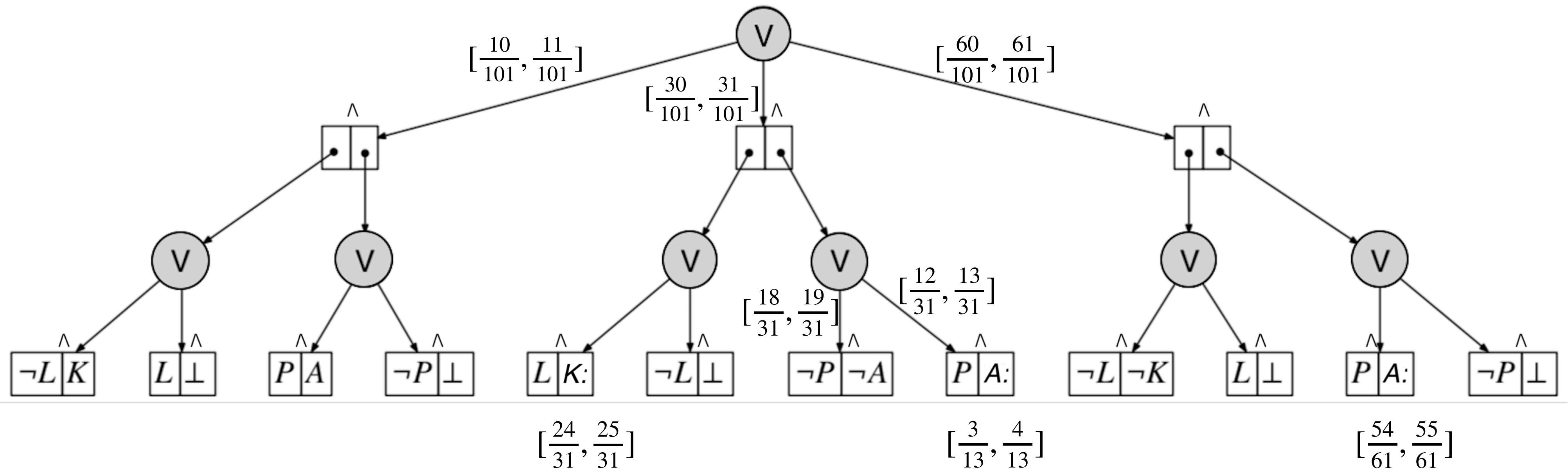
- Logically impossible events have zero probability: $\mathbb{P}(\mathsf{x}) > 0 \leftrightarrow \mathsf{x} \vDash \phi$

CREDAL VERSION OF PSDD'S:

CREDAL VERSION OF PSDD'S: REPLACE PMF'S WITH CS'S

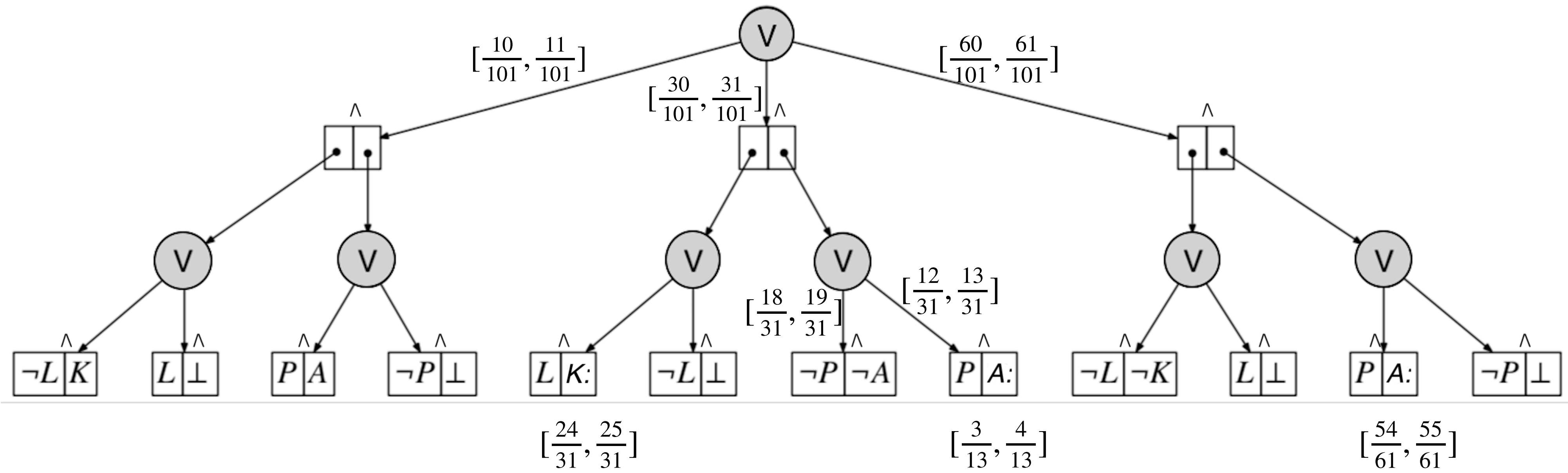
CREDAL VERSION OF PSDD'S: REPLACE PMF'S WITH CS'S

- **Credal Sentential Decision Diagrams** (**CSDD**s) for ϕ



CREDAL VERSION OF PSDD'S: REPLACE PMF'S WITH CS'S

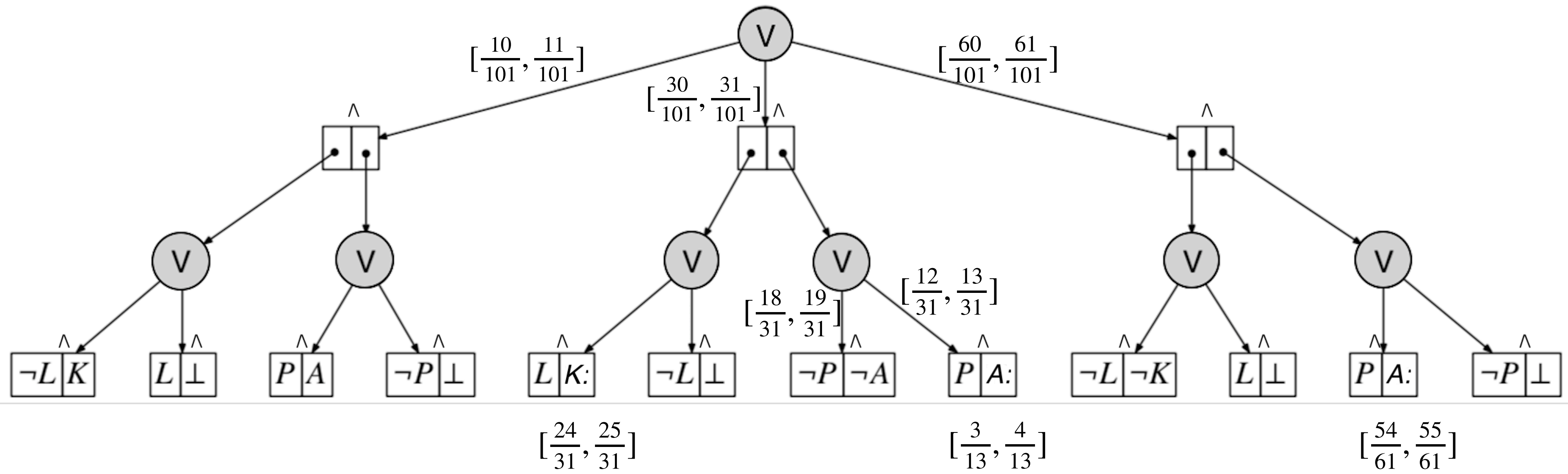
- **Credal Sentential Decision Diagrams** (**CSDD**s) for ϕ



- Syntax: **CS** attached to each decision node and to each terminal node $\top$

CREDAL VERSION OF PSDD'S: REPLACE PMF'S WITH CS'S

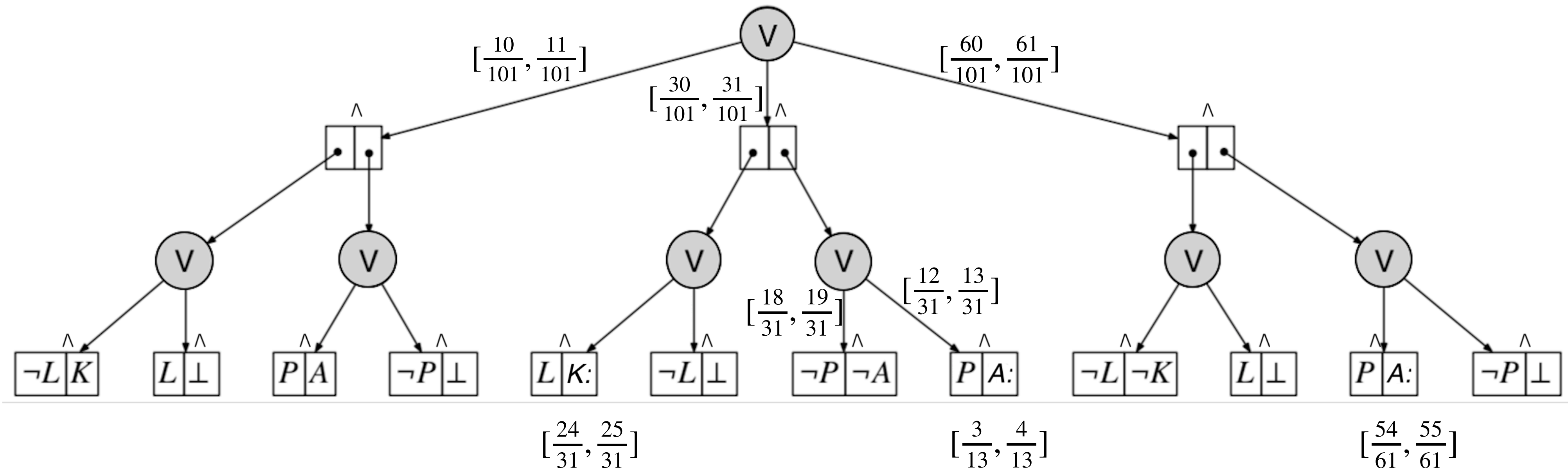
- **Credal Sentential Decision Diagrams** (**CSDD**s) for ϕ



- Syntax: **CS** attached to each decision node and to each terminal node T
- Semantics: collection of consistent PSDDs

CREDAL VERSION OF PSDD'S: REPLACE PMF'S WITH CS'S

- **Credal Sentential Decision Diagrams** (**CSDD**s) for ϕ



- Syntax: **CS** attached to each decision node and to each terminal node T
- Semantics: collection of consistent PSDDs
- PSDD induces joint P, CSDD induces joint CS ("Strong extension")

CSDD'S INFERENCE

CSDD'S INFERENCE

Marginal queries:

Given evidence e, calculate

$$\underline{\mathbb{P}}(e) = \min_{\mathbb{P}(\mathbf{X}) \in \mathbb{K}(\mathbf{X})} \mathbb{P}(e)$$

CSDD'S INFERENCE

Marginal queries:

Given evidence e, calculate

$$\underline{\mathbb{P}}(e) = \min_{\mathbb{P}(\mathbf{X}) \in \mathbb{K}(\mathbf{X})} \mathbb{P}(e)$$

Conditional queries:

Given available evidence e and queried variabile, calculate

$$\underline{\mathbb{P}}(x \,|\, e) = \min_{\mathbb{P}(\mathbf{X}) \in \mathbb{K}(\mathbf{X})} \frac{\mathbb{P}(x, e)}{\mathbb{P}(e)}$$

TWO POLYTIME ALGORITHMS

- Adaptation of CSPNs algorithms (Mauá et al.) to CSDDs:

Marginal queries:

- Bottom-up propagation of LP task's results
- Coefficients of each LP task are computed in the lower level
- Feasible regions are the local CSs

Conditional queries:

- Decisional version of original task
- Bottom-up propagation of LP task's results
- Coefficients of each LP task are computed in the lower level, **depending on evidence**
- Feasible regions are the local CSs

TWO POLYTIME ALGORITHMS

- Adaptation of CSPNs algorithms (Mauá et al.) to CSDDs:

Marginal queries:

- Bottom-up propagation of LP task's results
- Coefficients of each LP task are computed in the lower level
- Feasible regions are the local CSs

Conditional queries:

- Decisional version of original task
- Bottom-up propagation of LP task's results
- Coefficients of each LP task are computed in the lower level, **depending on evidence**
- Feasible regions are the local CSs

Needs singly connected topology

CONCLUSIONS AND FUTURE WORK

- CSDDs as a new tool for sensitivity analysis in PSDD
- Robust marginalisation and conditioning (for singly connected circuits) with poly complexity
- Application to “credal” ML with structured spaces
- Complexity and approximations results for multiply connected CSDDs
- Hybrid (structured/unstructured) models
- Structural learning (trade-off small SDD / likelihood / independences)
- CNs vs. CSDDs ?

CONCLUSIONS AND FUTURE WORK

- CSDDs as a new tool for sensitivity analysis in PSDD
- Robust marginalisation and conditioning (for singly connected circuits) with poly complexity
- Application to “credal” ML with structured spaces
- **Complexity and approximations results for multiply connected CSDDs**
- Hybrid (structured/unstructured) models
- Structural learning (trade-off small SDD / likelihood / independences)
- **CNs vs. CSDDs ?**

Credal Sentential Decision Diagrams (CSDDs)

Alessandro Antonucci, Alessandro Facchini, Lilith Mattei

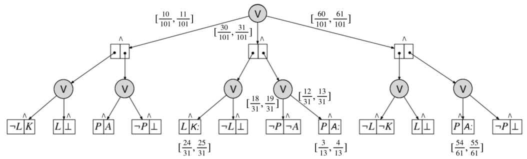
{alessandro, alessandro.facchini, lilith}@idsia.ch

USI/SUPSI Istituto Dalle Molle di studi sull'intelligenza artificiale IDSIA

A NEW CLASS OF (CREDAL) GRAPHICAL MODELS

- Bayesian nets as classical (precise) probabilistic graphical models (BNs)
- With imprecise probabilities? Credal networks (CNs, Cozman, 2000)
- With deep structure (and tractable inference)?
  Sum-product networks (SPNs, Poon & Domingos, 2011)
- With deep structure and imprecise probabilities?
  Credal sum-product networks (CSPNs, Mauá et al., 2017)
- With deep structure and embedding logical constraints?
  Probabilistic sentential decision diagrams (PSDDs, Kisa et al., 2014)
- Deep structure, imprecise probabilities and logical constraints?
  Credal sentential decision diagrams (CSDDs, this paper)

FROM SDDs TO CSDDs (THROUGH PSDDs)



- Logical skeleton? ϕ as a circuit alternating OR and AND gates
- This is a sentential decision diagram, (SDD, Choi & Darwiche, 2013)
- Probabilistic model? Probability mass functions annotating the OR gates of the SDD (PSDDs)
- PSDD is a joint probability mass function consistent with the constraints

$$\mathbb{P}(L,K,P,A) : \mathbb{P}(l,k,p,a) = 0 \text{ iff } (l,k,p,a) \not\models \phi$$

- CSDD? Credal version of PSDD: credal sets instead of mass functions
- Credal sets on OR gates and terminal nodes $\top$
- Semantics: all PSDDs with parameters consistent with the local credal sets
- Strong extension $\mathbb{K}(L,K,P,A)$ as the joint credal set of **all** the joint mass functions induced by the consistent PSDDs
- CSDD Inference? Lower/upper bounds wrt the strong extension
- Base theorem: for each $\mathbf{z}$: $\underline{\mathbb{P}}(\mathbf{z}) > 0$ iff $\mathbf{z} \models \phi$ and $\overline{\mathbb{P}}(\mathbf{z}) = 0$ iff $\mathbf{z} \not\models \phi$
- Learning CSDD? Parameters are conditional probabilities, *Imprecise Dirichlet Model* to learn local (conditional) credal sets
- Data scarcity issue on the leaves.justifies imprecise approach!

CONCLUSIONS & OUTLOOKS

- CSDDs as a new tool for sensitivity analysis in PSDD
- Fast robust marginalisation and conditioning
  (but conditioning works for singly connected circuits only)
- Complexity results and approximated algorithm are needed
- CNs vs. CSDDs? Credal classification with CSDDs?

REFERENCES

- Hoifung Poon & Pedro Domingos. Sum-product networks: a new deep architecture. In IEEE ICCV Workshops, pages 689-690. IEEE, 2011.
- Denis Mauá, Fabio Cozman, Diarmaid Conaty, and Cassio de Campos. Credal sum-product networks. In Proceedings of ISIPTA '17, pages 205-216, 2017.
- Denis Deratani Mauá, Diarmaid Conaty, Fabio Cozman, Katja Poppenhaeger, and Cassio de Campos. Robustifying sum-product networks. International Journal of Approximate Reasoning, 2018.
- Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In Proceedings of the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning, 2014.
- Fabio Cozman. Credal networks. Artificial Intelligence, 120:199-233, 2000.
- Arthur Choi and Adnan Darwiche. Dynamic minimization of sentential decision diagrams. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.

TOY EXAMPLE: CLASSES ENROLLMENT

- Data about 100 students in four classes
- **L**ogic, **K**nowledge, **P**robability and **A**rtificial Intelligence
- Logical constraints for classes:
  $\phi := (P \vee L) \wedge (A \to P) \wedge (K \to A \vee L)$
- Out of $2^4 = 16$ joint configurations,
  only eight in the data set
  seven are logically impossible,
  one possible but observed)
- Robust learning of a model over (L,K,P,A)?
- Consistent with the logical constraints ϕ?
- The solution is a CSSD!

L	K	P	A	#
0	0	0	0	0
0	0	0	1	0
0	0	1	0	6
0	0	1	1	54
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	10
1	0	0	0	5
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	13
1	1	0	1	0
1	1	1	0	8
1	1	1	1	3

MARGINAL QUERIES

- Circuit traversal from leaves in reverse topological order
- Every time a decision node is processed, a LP task whose feasible region are the local credal sets of the node should be solved.
- Analogous to Mauá et al. (2017) for CSPNs, with additionally support to logical constraints

Algorithm 2 Lower probability of evidence

```
input: CSDD, evidence e
for n ← N,...,1 do
  π(n) ← 0
  if node n is terminal then
    v ← leaf vtree node that n is normalized for
    π(n) ← P_n(e_v)
  else
    ((p_i, s_i)_{i=1}^k, K_n(P)) ← n (decision node)
    π(n) ← min_{[θ_1,...,θ_k] ∈ K_n(P)} Σ_{i=1}^k π(p_i) · π(s_i) · θ_i
  end if
end for
output: P(e) ← π(1)
```

CONDITIONAL QUERIES

- Conditional queries solved by *generalized Bayes' rule* (GBR)
- Associated decision problem is deciding whether or not, for a given $\mu \in [0,1]$: $\underline{\mathbb{P}}(x|\boldsymbol{e}) > \mu$
- As $\mathbb{P}(x|\boldsymbol{e}) + \mathbb{P}(\neg x|\boldsymbol{e}) = 1$ for each $\mathbb{P}(\boldsymbol{X}) \in \mathbb{K}^r(\boldsymbol{X})$, and assuming that $\mathbb{P}(\boldsymbol{e}) > 0$, this corresponds to: $\min_{\mathbb{P}(\boldsymbol{X}) \in \mathbb{K}^r(\boldsymbol{X})} [(1-\mu)\mathbb{P}(x,\boldsymbol{e}) - \mu\mathbb{P}(\neg x, \boldsymbol{e})] > 0$
- Recursive formulation (for singly connected circuits): $\min_{[\theta_1,\ldots,\theta_k] \in \mathbb{K}_r(P)} \sum_{i=1}^k \underline{\pi}(p_i)\, \underline{\sigma}(s_i)\, \theta_i > 0$
- where $\underline{\pi}(p_i)$ is equal to $\min_{\mathbb{P}_{p_i}(\boldsymbol{Z}) \in \mathbb{K}^{p_i}(\boldsymbol{Z})} [(1-\mu)\mathbb{P}_{p_i}(x, \boldsymbol{e}_l) - \mu \mathbb{P}_{p_i}(\neg x, \boldsymbol{e}_l)]$
- and $\underline{\sigma}(s_i)$ is equal to $\begin{cases} \overline{\mathbb{P}}_{s_i}(\boldsymbol{e}_r) & \text{if } \underline{\pi}(p_i) < 0 \\ \underline{\mathbb{P}}_{s_i}(\boldsymbol{e}_r) & \text{otherwise.} \end{cases}$

- Circuit traversal from leaves (as for marginal queries)
- LP tasks on decision nodes whose coefficients are computed with marginal queries
- Bracketing scheme to solve GBR
- Again analogous to Mauá et al. (2017) result for CSPNs

Algorithm 3 Lower conditional probability

```
input: CSDD, μ, X = x, e
for n ← N,...,1 do
  π(n) ← 0
  v ← vtree node that n is normalized for
  if node n is terminal then
    π(n) ← Λ_n(μ) as in Eq. (12)
  else
    ((p_i, s_i)_{i=1}^k, K_n(P)) ← n (decision node)
    if X occurs in v then
      if X occurs in v^l then
        u ← v^l and w ← v^r
        u_i ← p_i and w_i ← s_i for 1 ≤ i ≤ k
      else if X occurs in v^r then
        w ← v^l and u ← v^r
        u_i ← s_i and w_i ← p_i for 1 ≤ i ≤ k
      end if
      π(n) ← min_{[θ_1,...,θ_k] ∈ K_n(P)} Σ_{i=1}^k π(u_i) · σ(w_i) · θ_i
      with σ as in Eq. (10)
    end if
  end if
end for
output: sign[P(x|e) − μ] ← sign[π(1)]
```