Tractable Learning in Structured Probability Spaces

Guy Van den Broeck

UCLA

DTAI Seminar - KU Leuven

Dec 20, 2016

Structured probability spaces?

Running Example

Courses:

- Logic (L)
- Knowledge Representation (K)
- Probability (P)
- Artificial Intelligence (A)

Data

L	K	P	A	Students
0	0	1	0	6
0	0	1	1	54
0	1	1	1	10
1	0	0	0	5
1	0	1	0	1
1	0	1	1	0
1	1	0	0	17
1	1	1	0	4
1	1	1	1	3

Constraints

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

Probability Space

unstructured

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

Structured Probability Space

unstructured

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

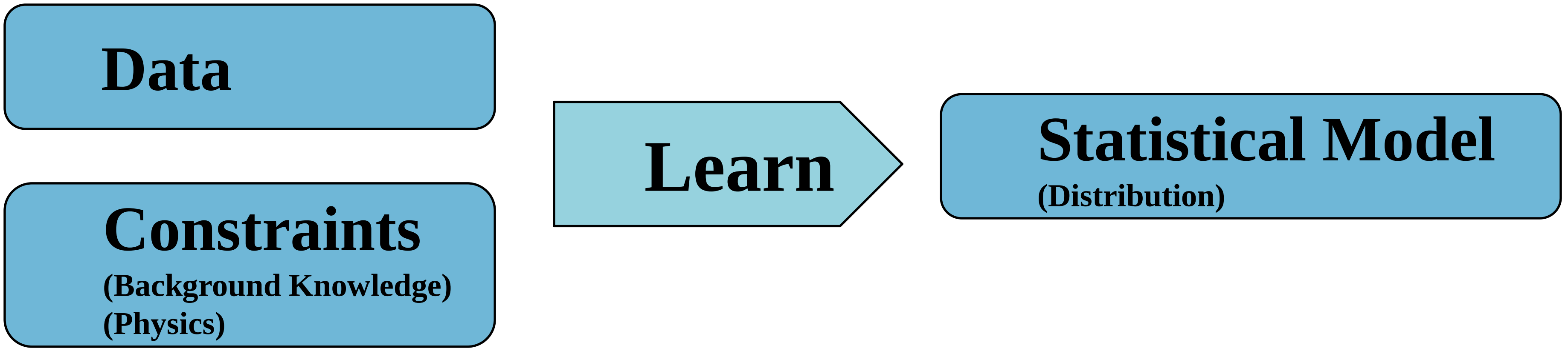
- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

7 out of 16 instantiations are impossible

structured

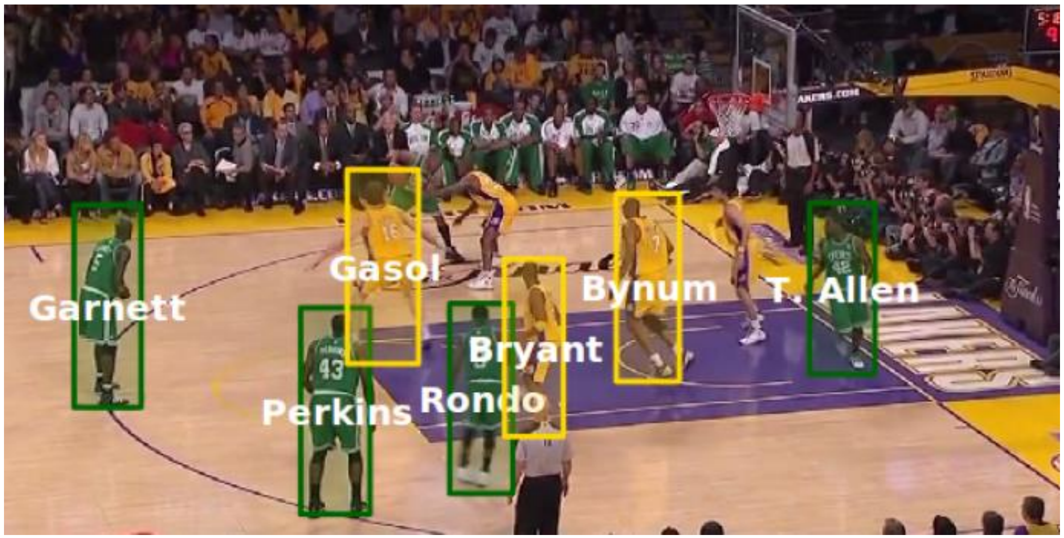
L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Learning with Constraints



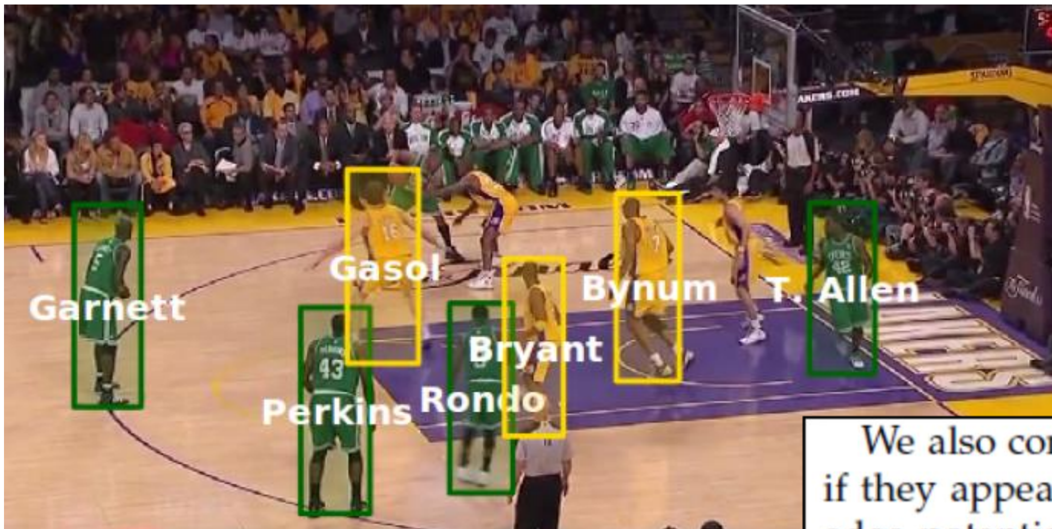
Learn a statistical model that assigns
zero probability
to instantiations that violate the constraints.

Example: Video



[Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos.]

Example: Video



We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time t. We then introduce an edge potential that enforces mutual exclusion:

$$\psi_{\text{mutex}}(y_{t,i}, y_{t,j}) = \begin{cases} 1 & \text{if } y_{t,i} \neq y_{t,j} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This potential specifies the constraint that a player can be appear only *once* in a frame. For example, if the i-th detection $y_{t,i}$ has been assign to Bryant, $y_{t,j}$ cannot have the same identity because Bryant is impossible to appear twice in a frame.

[Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos.]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],…, [Chang, M. W., Ratinov, L., & Roth, D. (2012). Structured learning with constrained conditional models.], [https://en.wikipedia.org/wiki/Constrained_conditional_model]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

  If a modifier is kept, its subject is also kept

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],…, [Chang, M. W., Ratinov, L., & Roth, D. (2012). Structured learning with constrained conditional models.], [https://en.wikipedia.org/wiki/Constrained_conditional_model]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

  If a modifier is kept, its subject is also kept

- Information extraction

Citations	
Start	The citation must start with author or editor.
AppearsOnce	Each field must be a consecutive list of words, and can appear at most once in a citation.
Punctuation	State transitions must occur on punctuation marks.
BookJournal	The words *proc, journal, proceedings, ACM* are *JOURNAL* or *BOOKTITLE*.
...	...
TechReport	The words *tech, technical* are *TECH_REPORT*.
Title	Quotations can appear only in titles.
Location	The words *CA, Australia, NY* are *LOCATION*.

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],…, [Chang, M. W., Ratinov, L., & Roth, D. (2012). Structured learning with constrained conditional models.], [https://en.wikipedia.org/wiki/Constrained_conditional_model]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

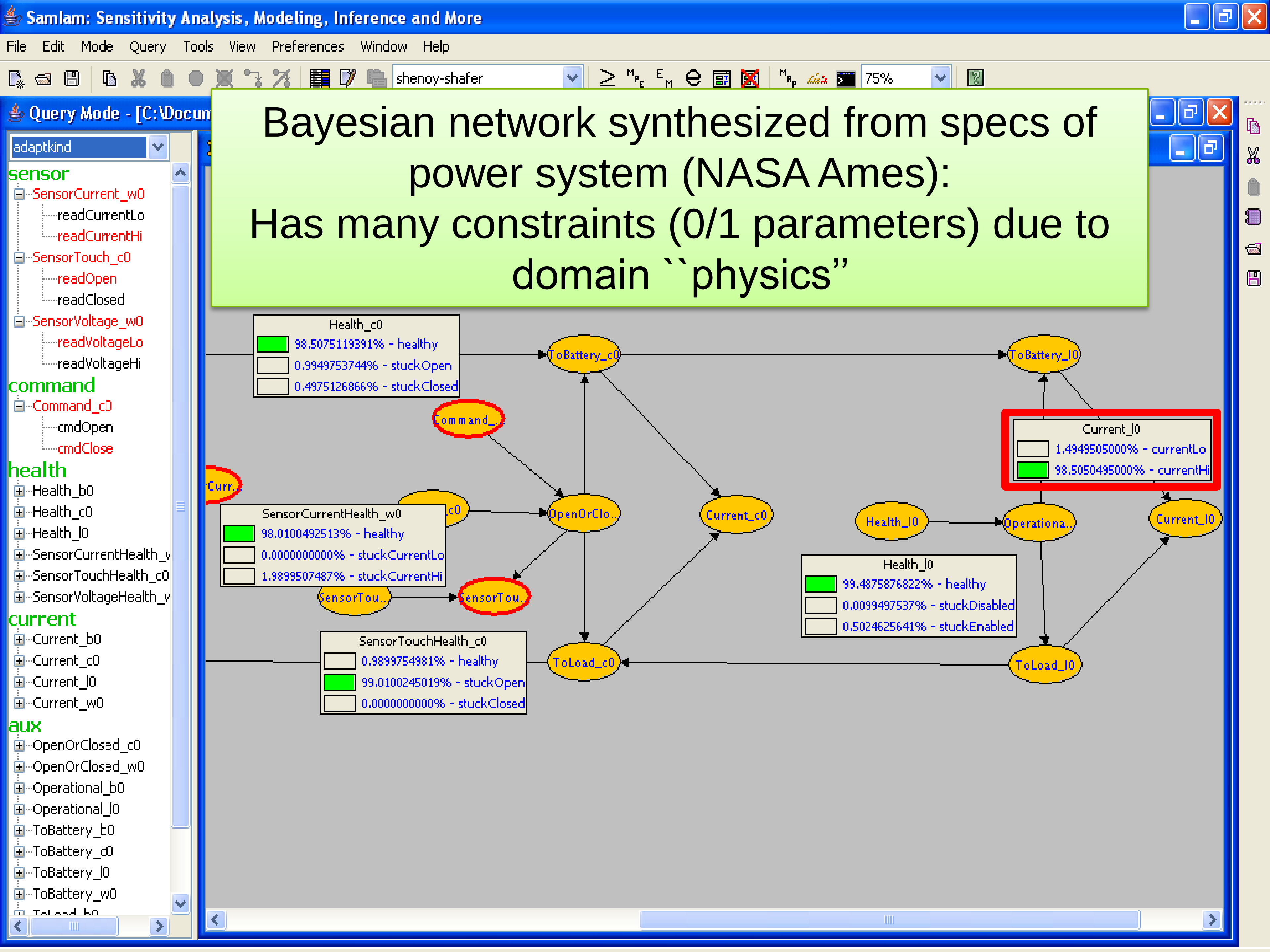
  If a modifier is kept, its subject is also kept

- Information extraction
- Semantic role labeling
- … and many more!

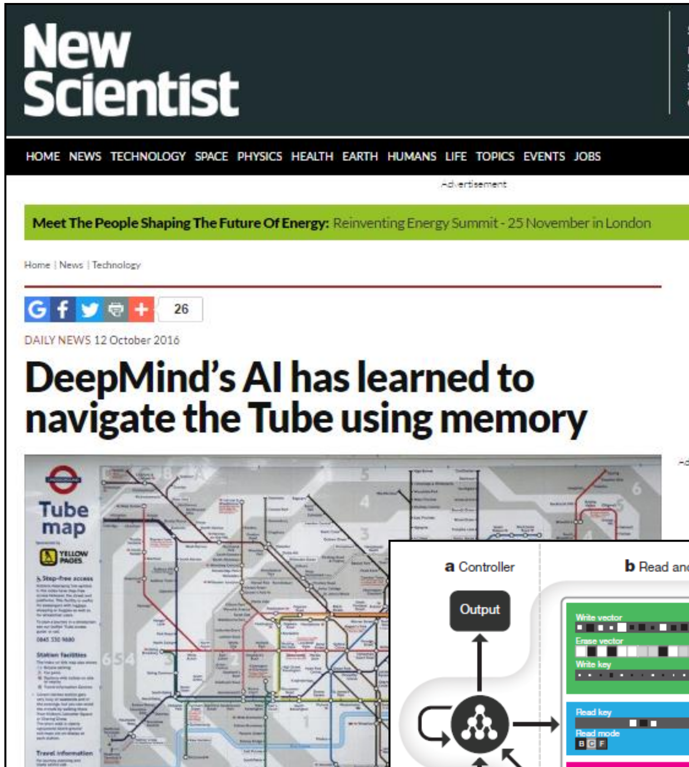
Citations	
Start	The citation must start with author or editor.
AppearsOnce	Each field must be a consecutive list of words, and can appear at most once in a citation.
Punctuation	State transitions must occur on punctuation marks.
BookJournal	The words *proc*, *journal*, *proceedings*, *ACM* are *JOURNAL* or *BOOKTITLE*.
...	...
TechReport	The words *tech*, *technical* are *TECH_REPORT*.
Title	Quotations can appear only in titles.
Location	The words *CA*, *Australia*, *NY* are *LOCATION*.

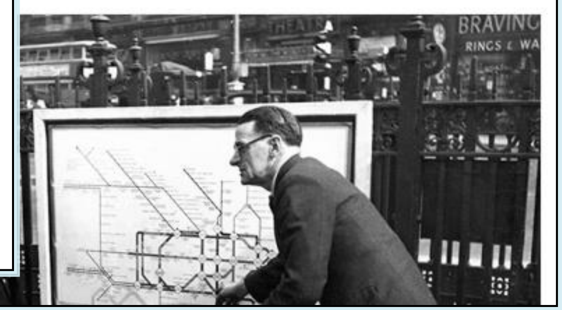
[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],…, [Chang, M. W., Ratinov, L., & Roth, D. (2012). Structured learning with constrained conditional models.], [https://en.wikipedia.org/wiki/Constrained_conditional_model]

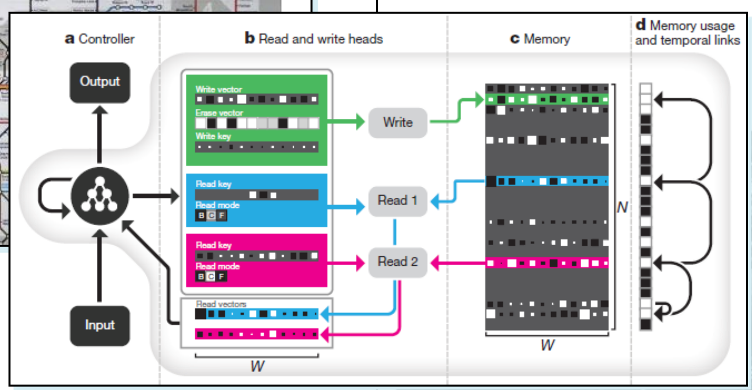
Bayesian network synthesized from specs of power system (NASA Ames):
Has many constraints (0/1 parameters) due to domain ``physics"



Example: Deep Learning





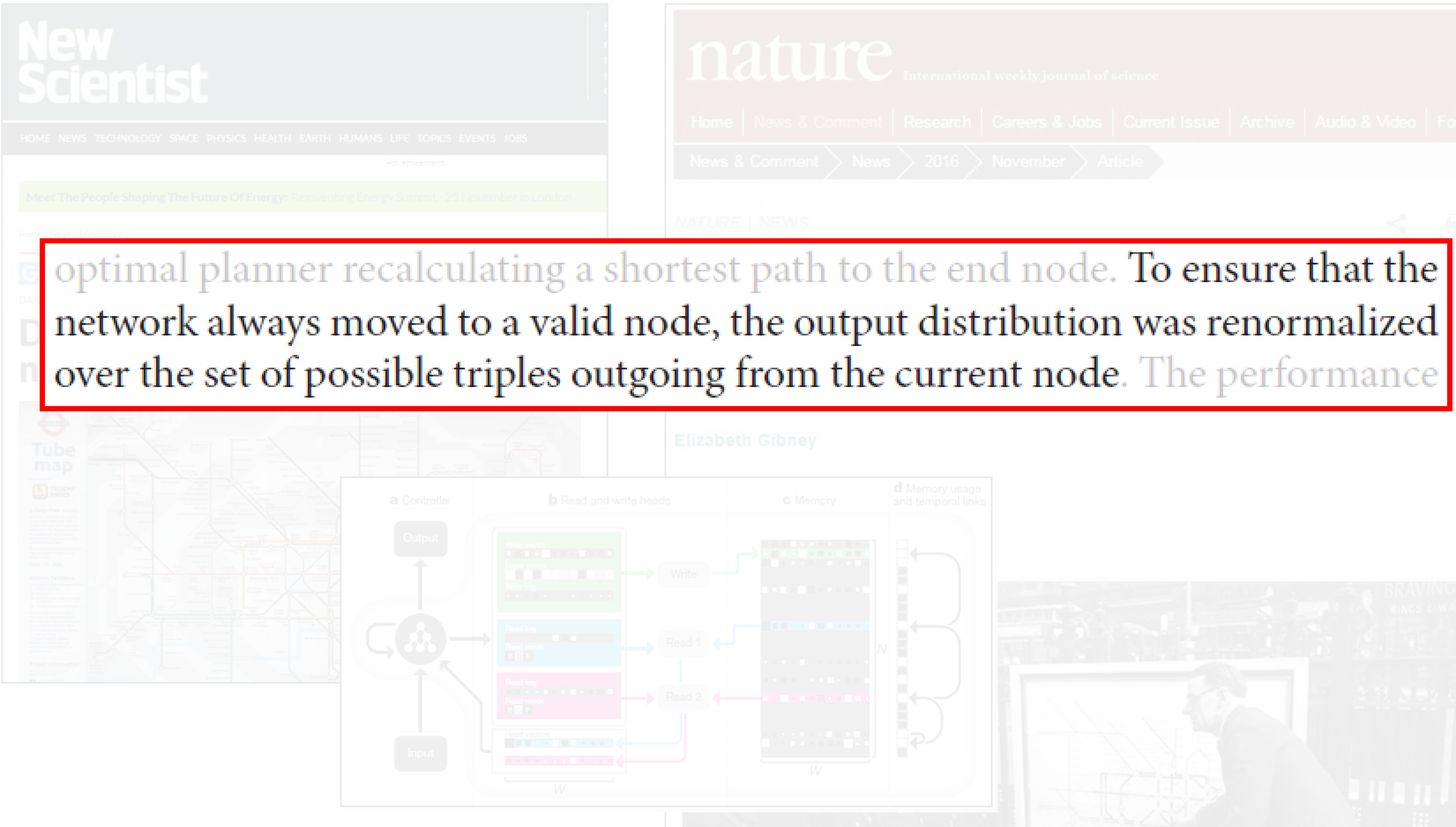


[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Example: Deep Learning

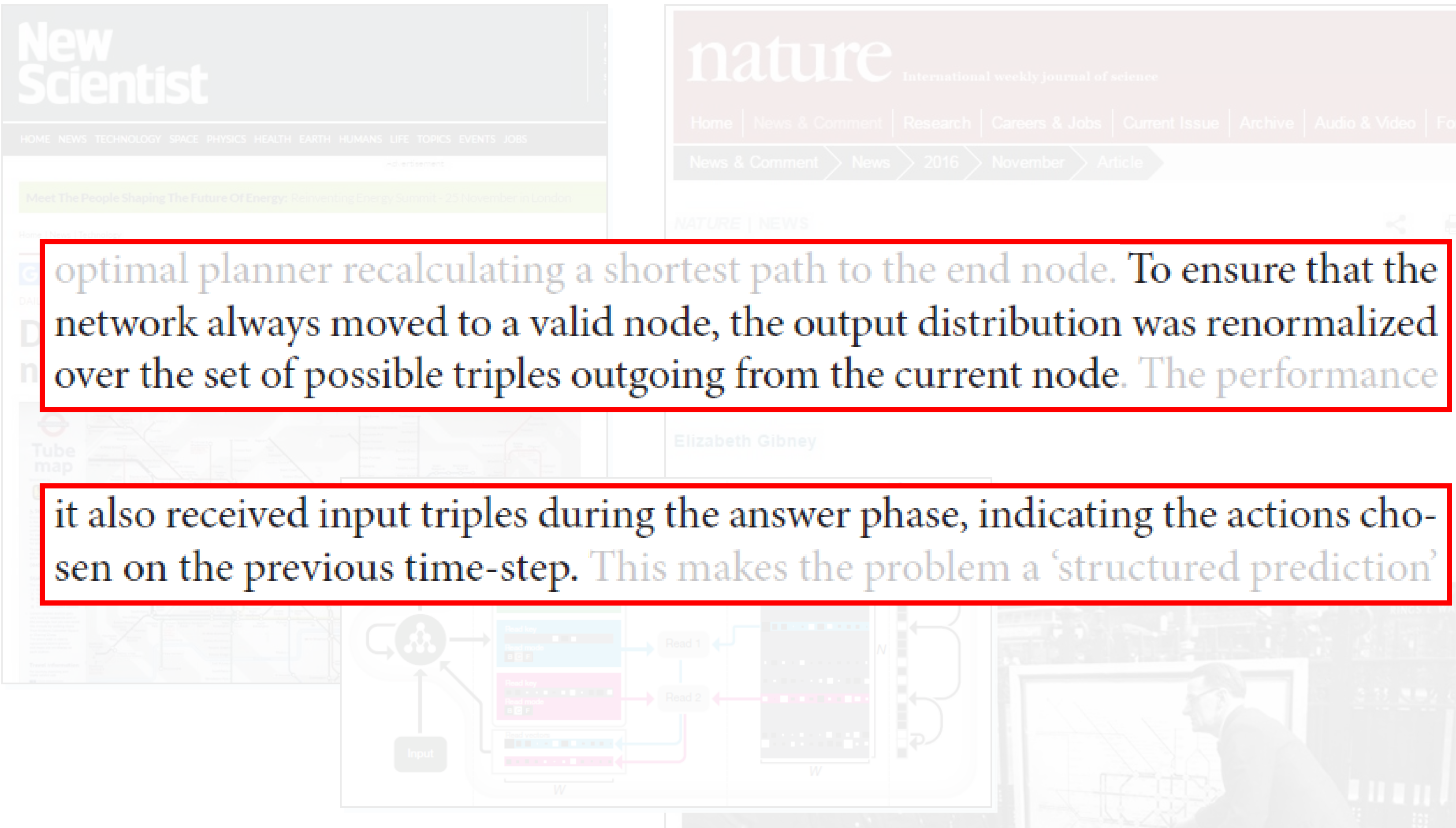
[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Example: Deep Learning



[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

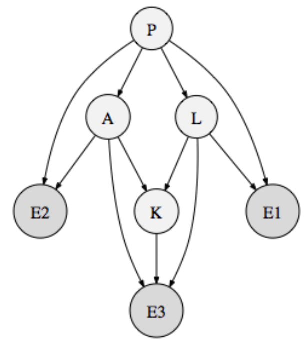
Example: Deep Learning



[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

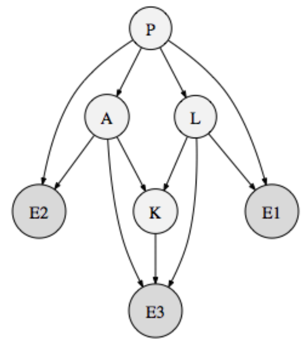
What are people doing now?

- Ignore constraints
- Handcraft into models
- Use specialized distributions
- Find non-structured encoding
- Try to learn constraints
- Hack your way around



What are people doing now?

- Ignore constraints
- Handcraft into models
- Use specialized distributions
- Find non-structured encoding
- Try to learn constraints
- Hack your way around



Accuracy ?
Specialized skill ?
Intractable inference ?
Intractable learning ?
Waste parameters ?
Risk predicting out of space ?

\+

you are on your own ☹

Structured Probability Spaces

- Everywhere in ML!
  - Configuration problems, inventory, video, text, deep learning
  - Planning and diagnosis (physics)
  - Causal models: cooking scenarios (interpreting videos)
  - Combinatorial objects: parse trees, rankings, directed acyclic graphs, trees, simple paths, game traces, etc.

Structured Probability Spaces

- Everywhere in ML!
  - Configuration problems, inventory, video, text, deep learning
  - Planning and diagnosis (physics)
  - Causal models: cooking scenarios (interpreting videos)
  - Combinatorial objects: parse trees, rankings, directed acyclic graphs, trees, simple paths, game traces, etc.
- Some representations: constrained conditional models, mixed networks, probabilistic logics.

Structured Probability Spaces

- Everywhere in ML!
  - Configuration problems, inventory, video, text, deep learning
  - Planning and diagnosis (physics)
  - Causal models: cooking scenarios (interpreting videos)
  - Combinatorial objects: parse trees, rankings, directed acyclic graphs, trees, simple paths, game traces, etc.
- Some representations: constrained conditional models, mixed networks, probabilistic logics.

No ML boxes out there that take constraints as input! ☹

Structured Probability Spaces

- Everywhere in ML!
  - Configuration problems, inventory, video, text, deep learning
  - Planning and diagnosis (physics)
  - Causal models: cooking scenarios (interpreting videos)
  - Combinatorial objects: parse trees, rankings, directed acyclic graphs, trees, simple paths, game traces, etc.
- Some representations: constrained conditional models, mixed networks, probabilistic logics.

No ML boxes out there that take constraints as input! ☹

<u>Goal</u>: Constraints as important as data! General purpose!

Specification Language: Logic

Structured Probability Space

unstructured

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

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

7 out of 16 instantiations are impossible

structured

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

Boolean Constraints

unstructured

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

$$P \vee L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \vee L)$$

7 out of 16 instantiations are impossible

structured

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

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

10 items:
3,628,800
rankings

20 items:
2,432,902,008,176,640,000
rankings

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

A_{ij} item i at position j (n items require n^2 Boolean variables)

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

A_{ij} item i at position j (n items require n^2 Boolean variables)

An item may be assigned to more than one position

A position may contain more than one item

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item i assigned to a unique position (n constraints)

$$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item i assigned to a unique position (n constraints)

$$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

constraint: each position j assigned a unique item (n constraints)

$$\bigvee_i A_{ij} \wedge \left(\bigwedge_{k \neq i} \neg A_{kj} \right)$$

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item i assigned to a unique position (n constraints)

$$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

constraint: each position j assigned a unique item (n constraints)

$$\bigvee_i A_{ij} \wedge \left(\bigwedge_{k \neq i} \neg A_{kj} \right)$$

total constraints	$2n$
unstructured space	2^{n^2}
structured space	$n!$

Structured Space for Paths

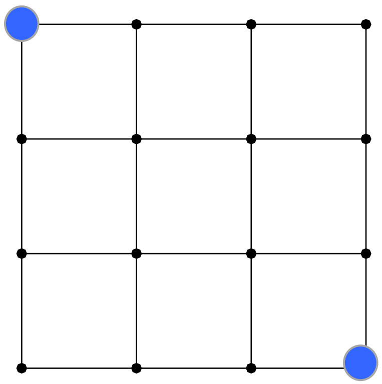

NATURE | NEWS

Google's AI reasons its way around the London Underground

DeepMind's latest technique uses external memory to solve tasks that require logic and reasoning — a step toward more human-like AI.

Elizabeth Gibney

13 October 2016



Structured Space for Paths

nature

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

News & Comment > News > 2016 > November > Article

NATURE | NEWS

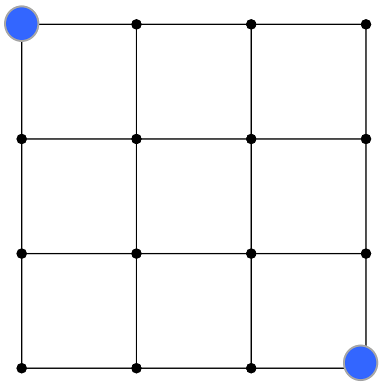
Google's AI reasons its way around the London Underground

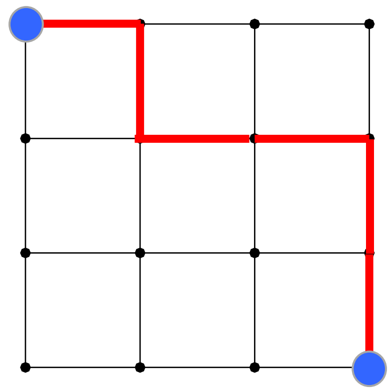
DeepMind's latest technique uses external memory to solve tasks that require logic and reasoning — a step toward more human-like AI.

Elizabeth Gibney

13 October 2016

Rights & Permissions

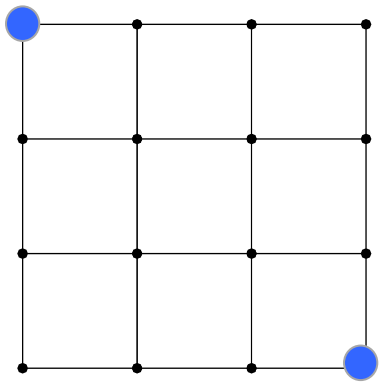


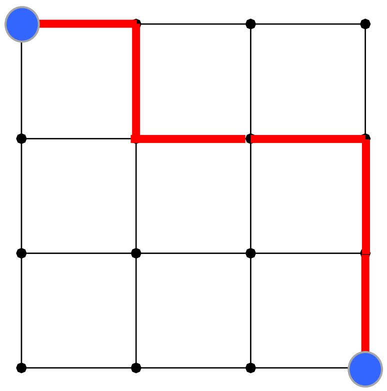


Good variable assignment
(represents route)

184

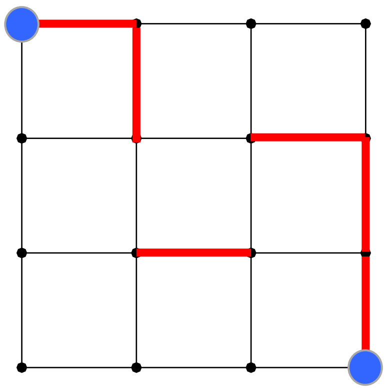
Structured Space for Paths





Good variable assignment (represents route)

184



Bad variable assignment (does not represent route)

16,777,032

Structured Space for Paths


nature

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

News & Comment > News > 2016 > November > Article

NATURE | NEWS

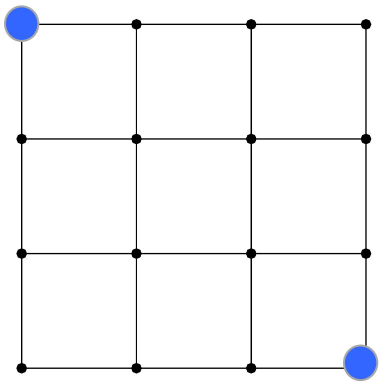
Google's AI reasons its way around the London Underground

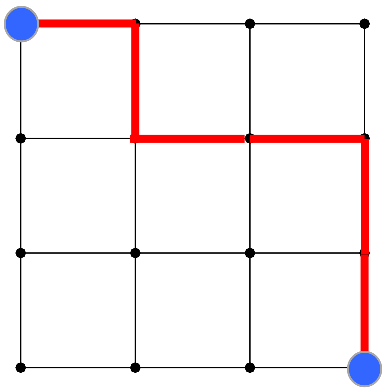
DeepMind's latest technique uses external memory to solve tasks that require logic and reasoning — a step toward more human-like AI.

Elizabeth Gibney

13 October 2016

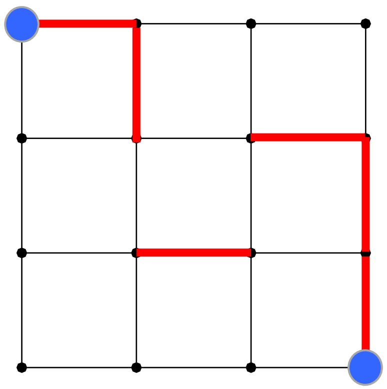
Rights & Permissions





Good variable assignment (represents route)

184



Bad variable assignment (does not represent route)

16,777,032

Space easily encoded in logical constraints ☺

Structured Space for Paths


nature

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

News & Comment > News > 2016 > November > Article

NATURE | NEWS

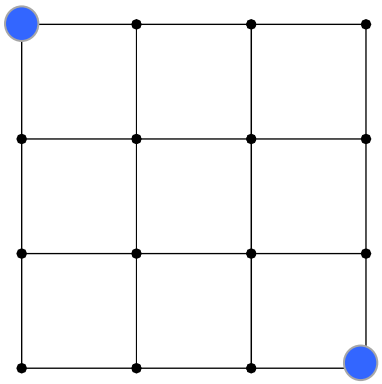
Google's AI reasons its way around the London Underground

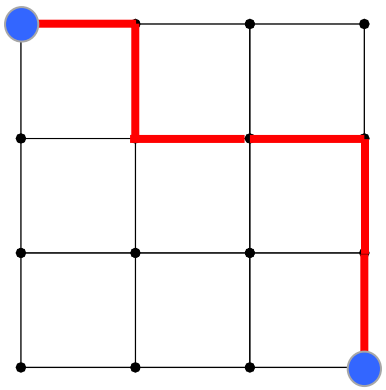
DeepMind's latest technique uses external memory to solve tasks that require logic and reasoning — a step toward more human-like AI.

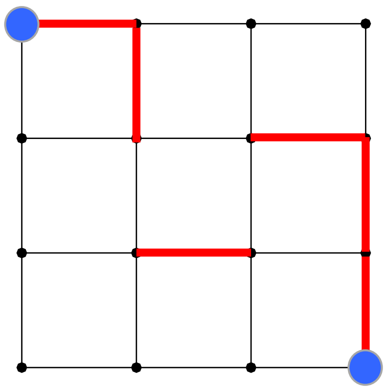
Elizabeth Gibney

13 October 2016

Rights & Permissions







Good variable assignment (represents route)

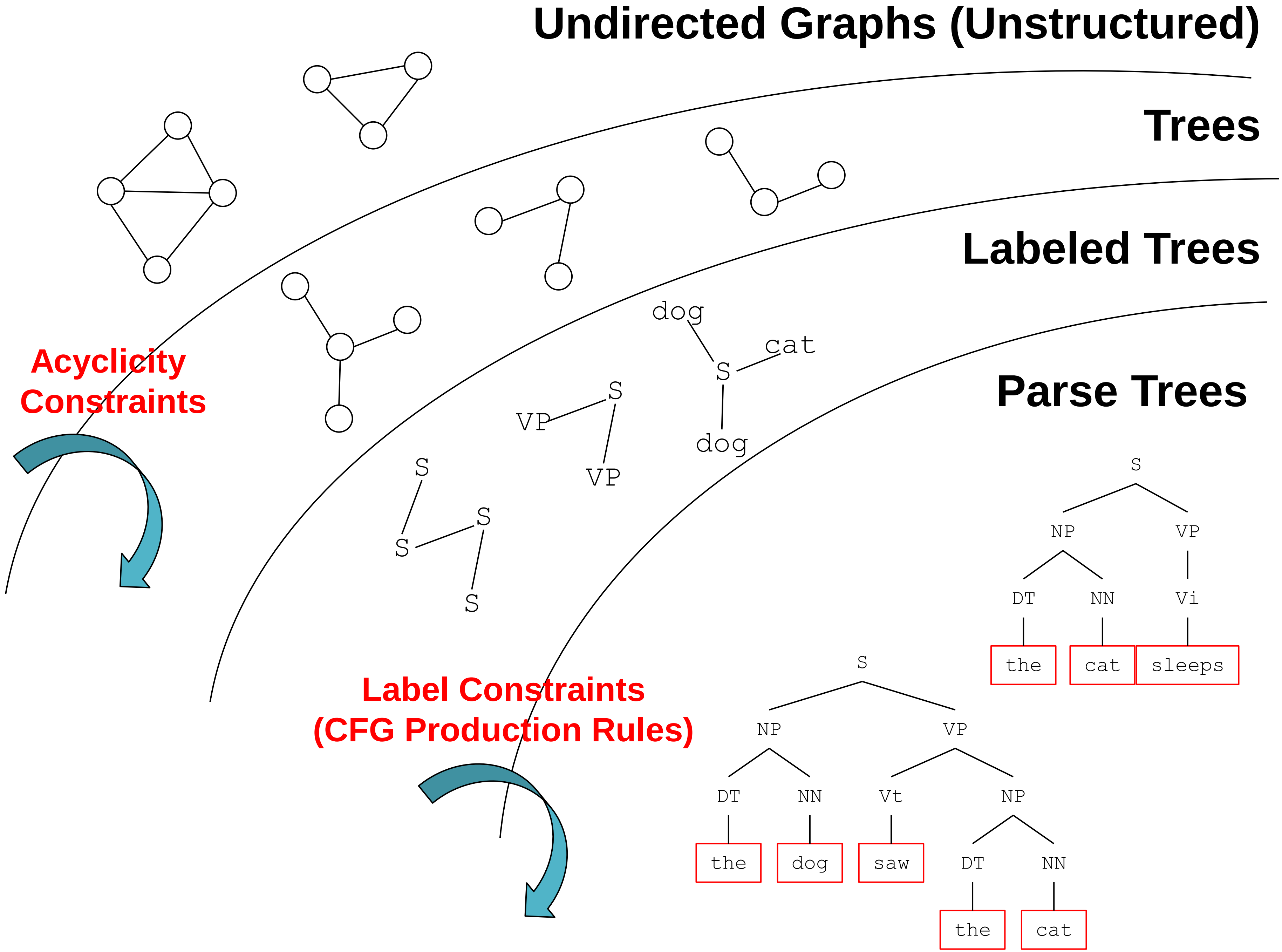
184

Bad variable assignment (does not represent route)

16,777,032

Space easily encoded in logical constraints ☺

Unstructured probability space: 184+16,777,032 = 2^{24}



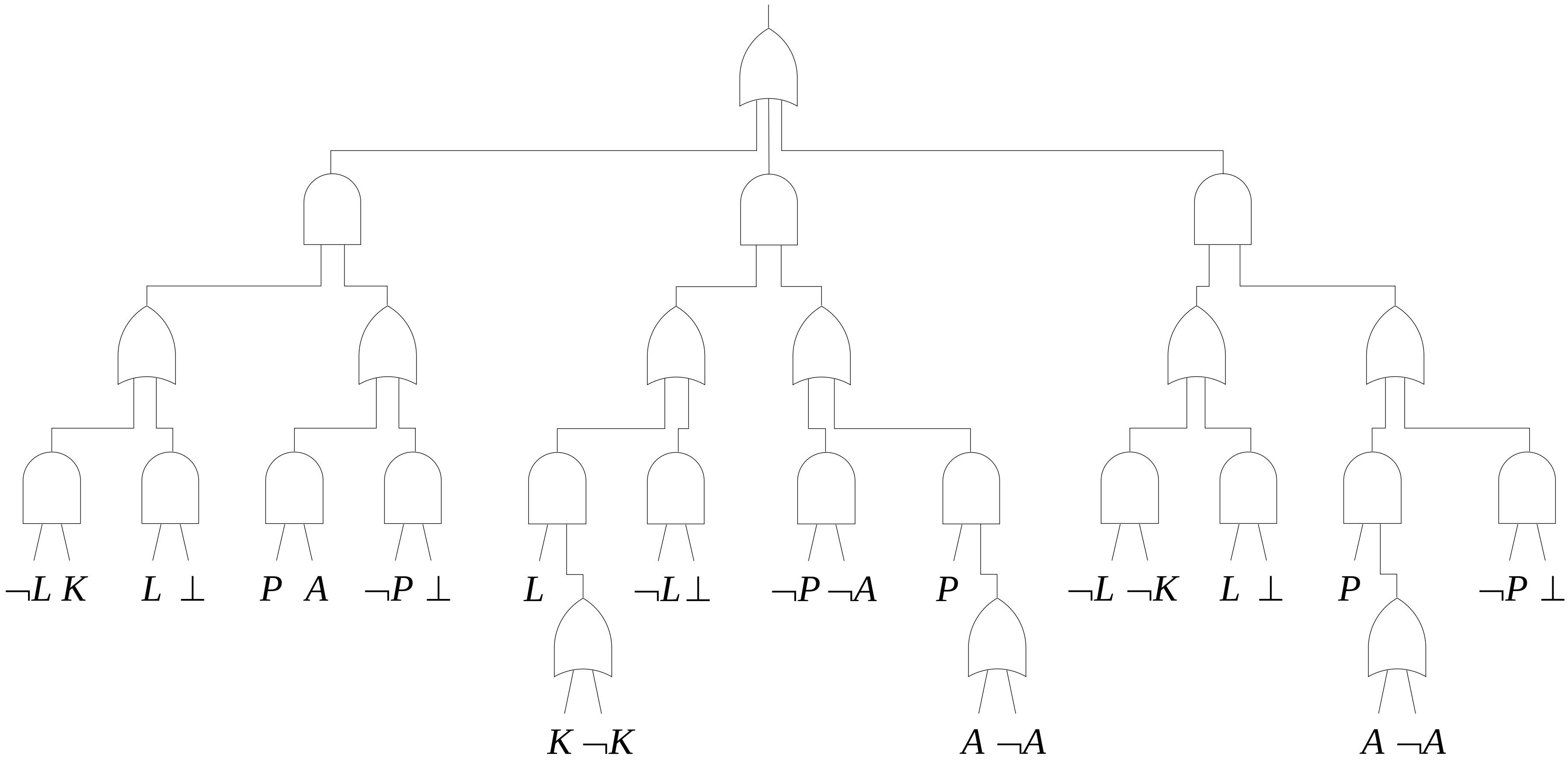
Undirected Graphs (Unstructured)
Trees
Labeled Trees
Parse Trees
Acyclicity Constraints
Label Constraints
(CFG Production Rules)
dog
cat
S
dog
VP
S
VP
S
S
S
S
S
NP
VP
DT
NN
Vi
the
cat
sleeps
S
NP
VP
DT
NN
Vt
NP
the
dog
saw
DT
NN
the
cat

“Deep Architecture”

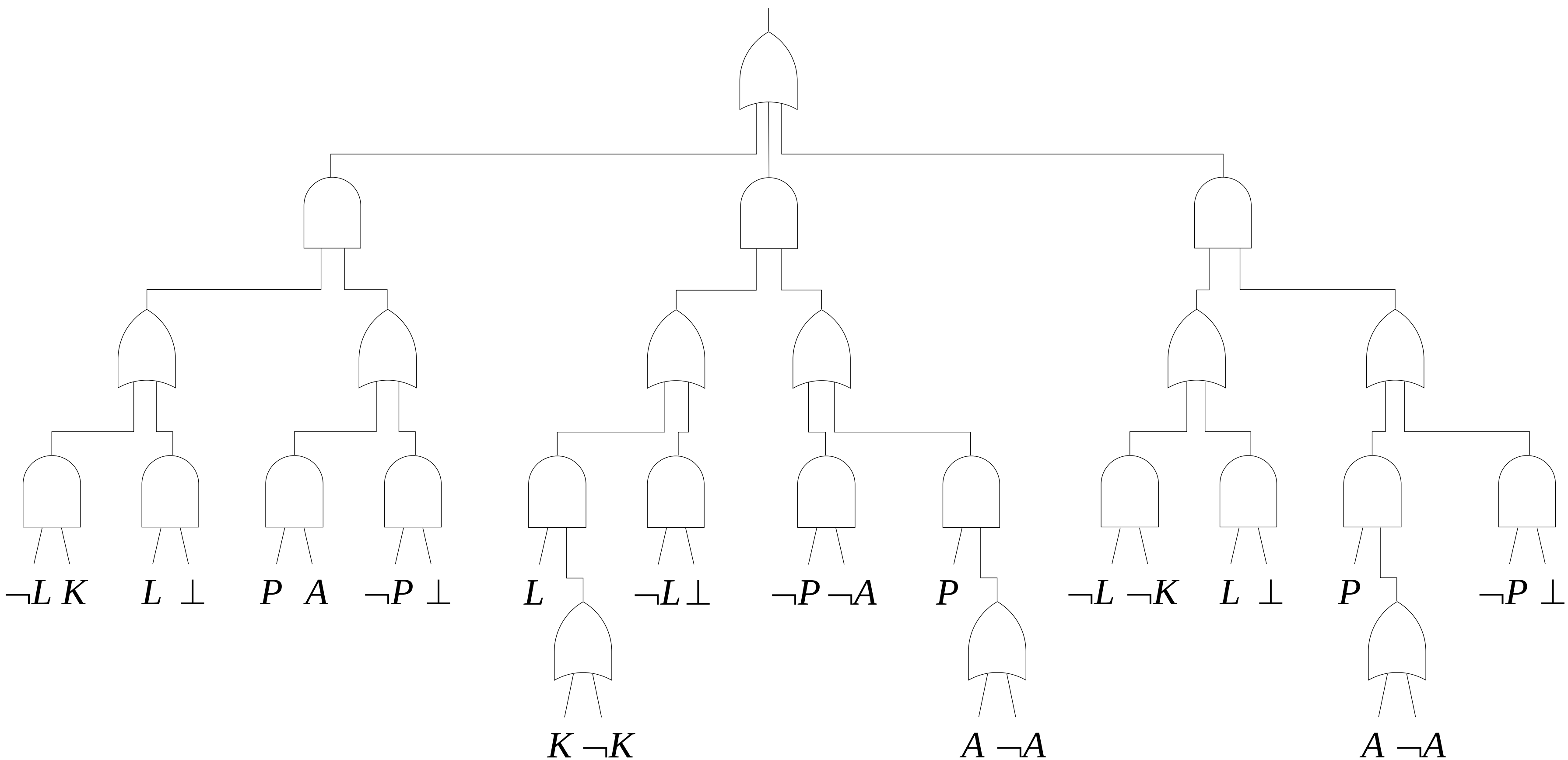
Logic + Probability

Logical Circuits

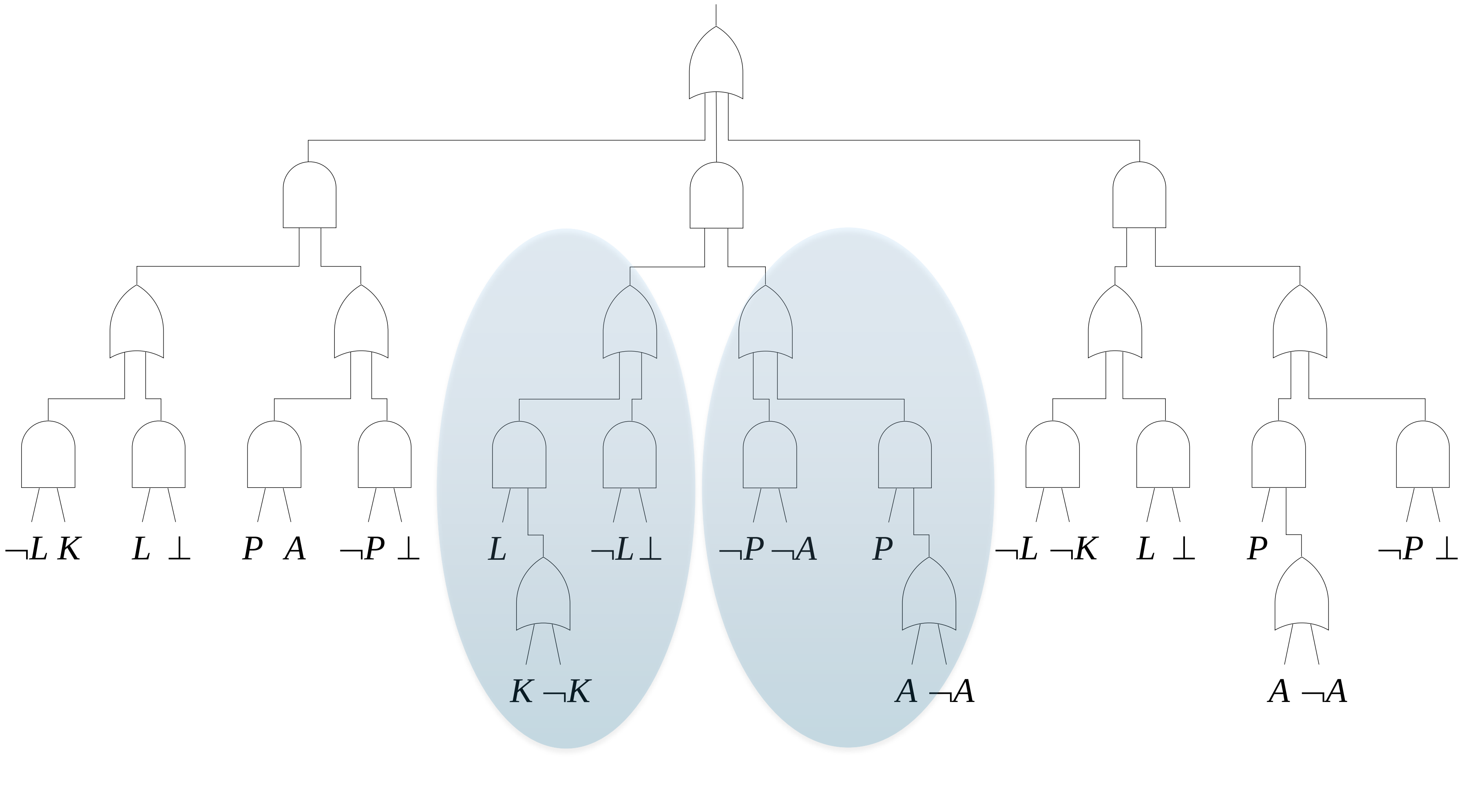
$P \vee L$
$A \Rightarrow P$
$K \Rightarrow (P \vee L)$



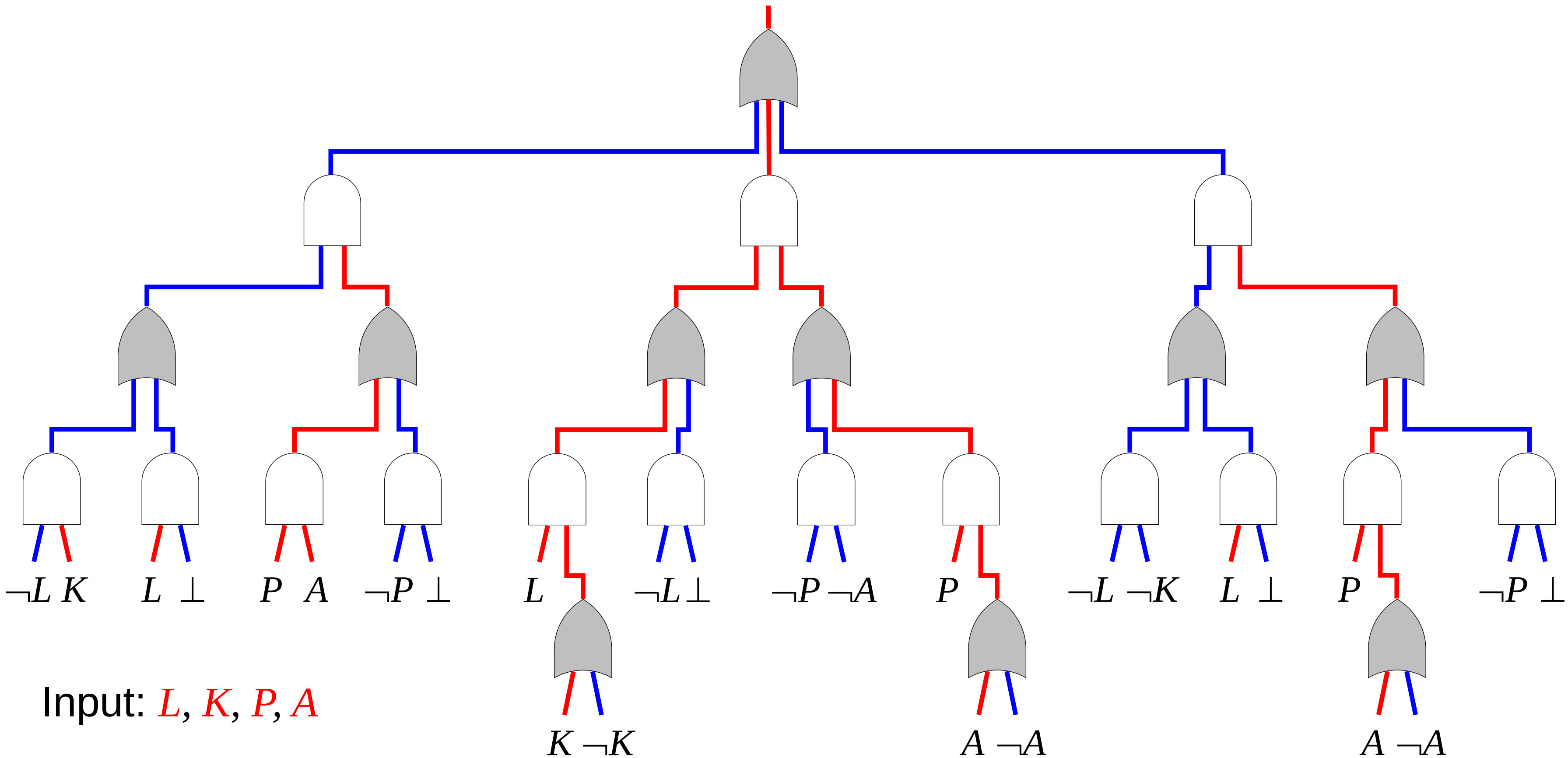
Property: Decomposability



Property: Decomposability

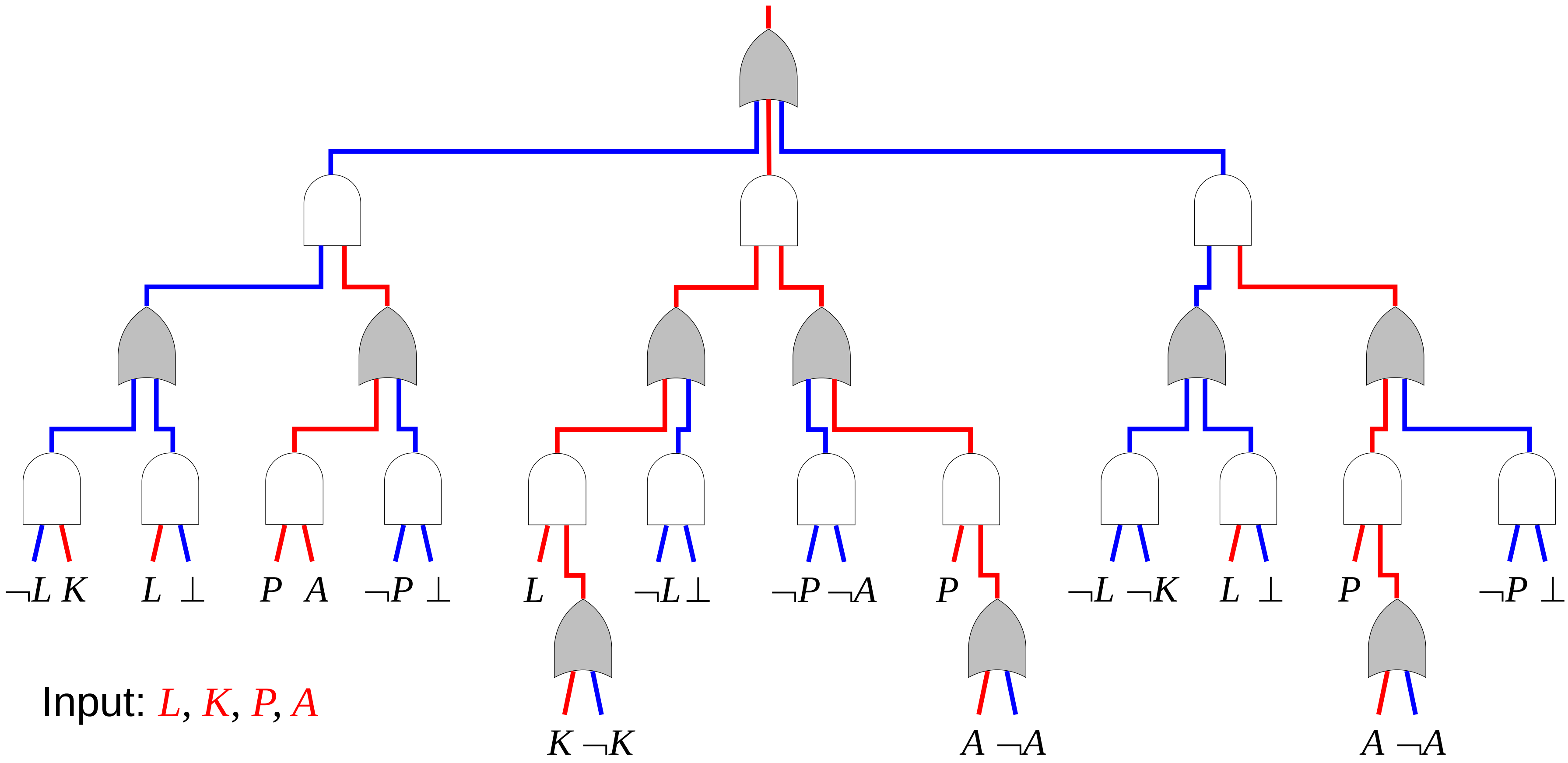


Property: Determinism



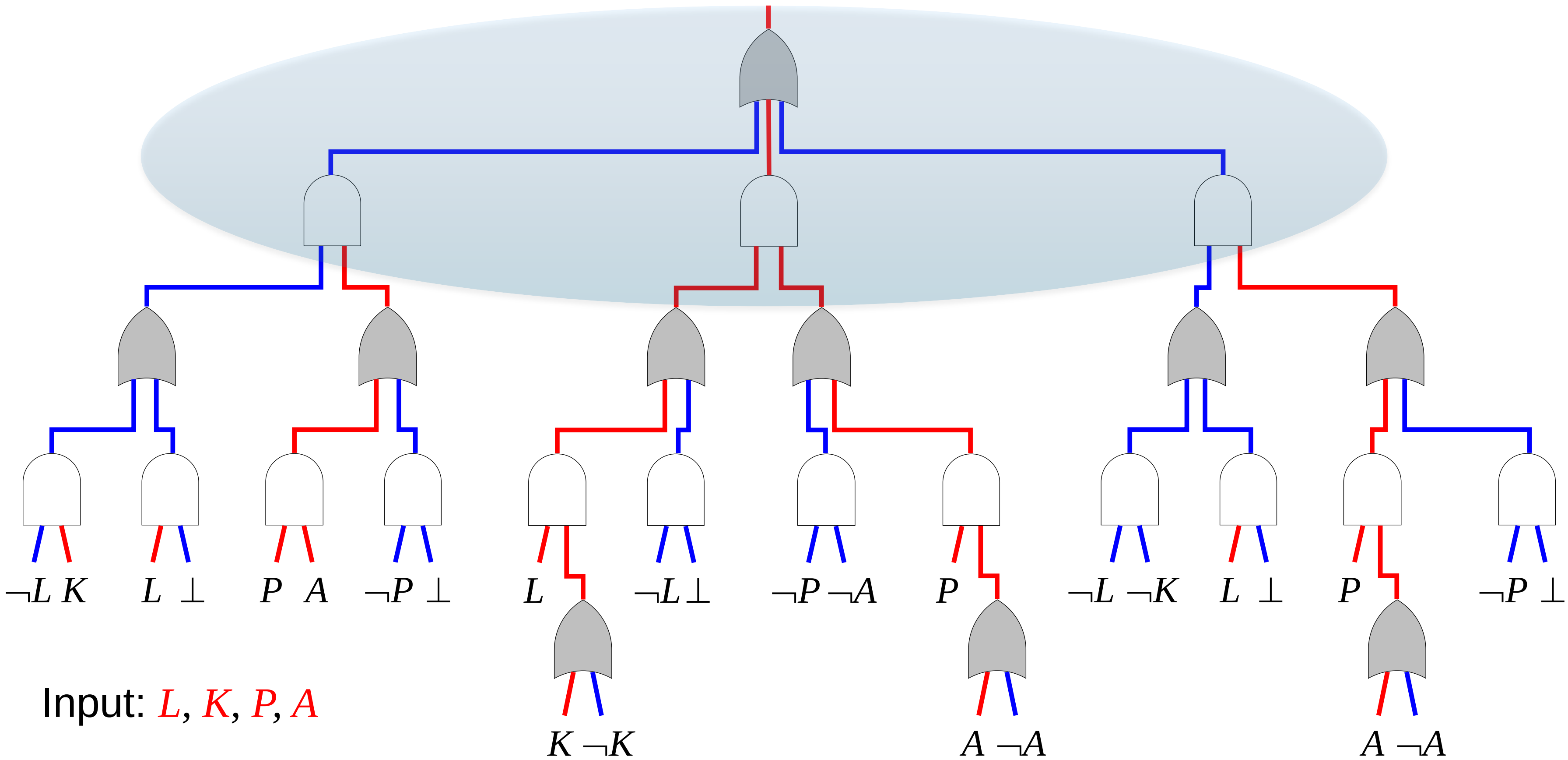
Input: L, K, P, A

Sentential Decision Diagram (SDD)



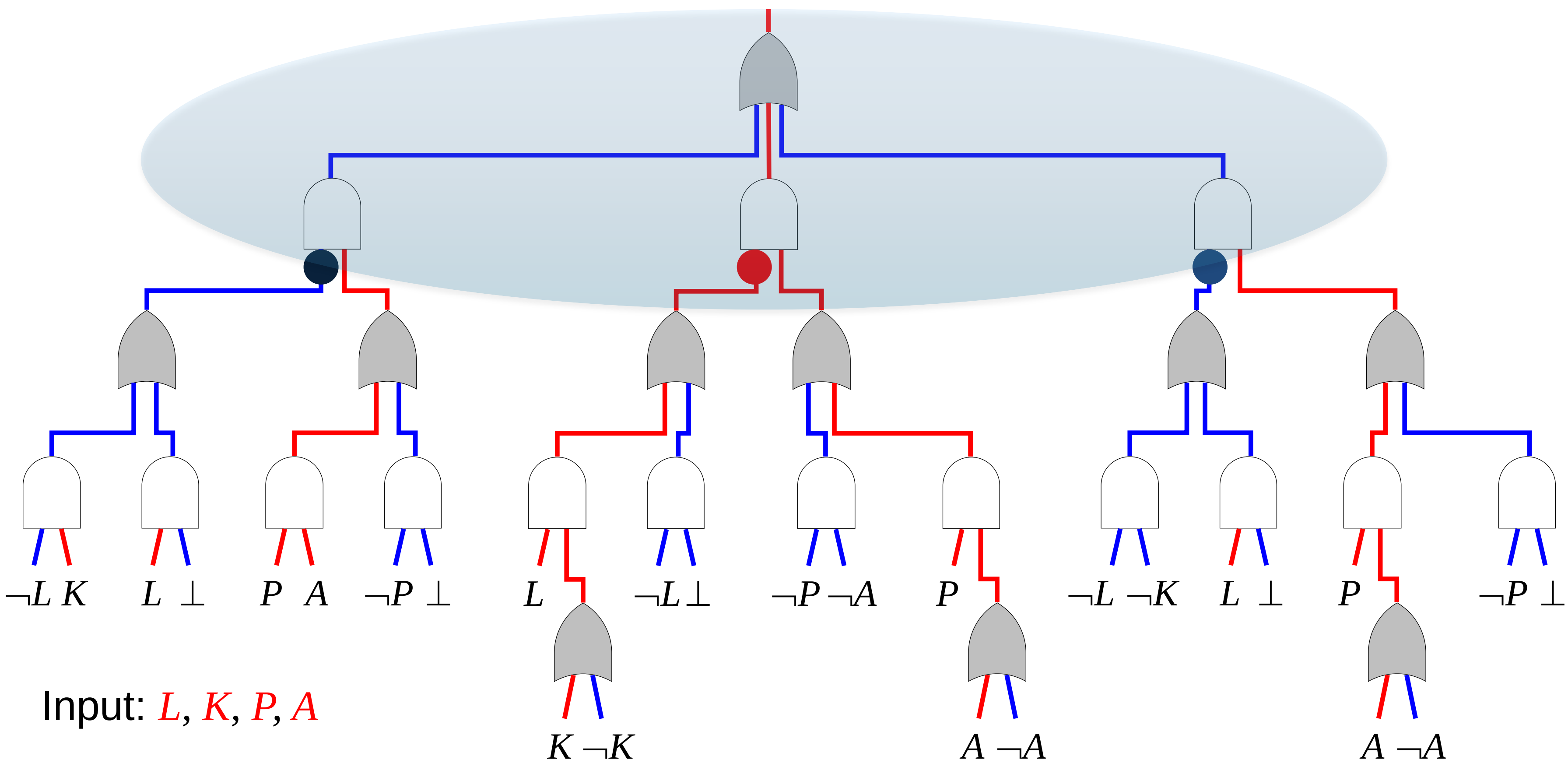
Input: L, K, P, A

Sentential Decision Diagram (SDD)



Input: L, K, P, A

Sentential Decision Diagram (SDD)



Tractable for Logical Inference

- Is structured space empty? (SAT)
- Count size of structured space (#SAT)
- Check equivalence of spaces
- Algorithms linear in circuit size ☺
  (pass up, pass down, similar to backprop)

Tractable for Logical Inference

- Is structured space empty? (SAT)
- Count size of structured space (#SAT)
- Check equivalence of spaces
- Algorithms linear in circuit size ☺
  (pass up, pass down, similar to backprop)

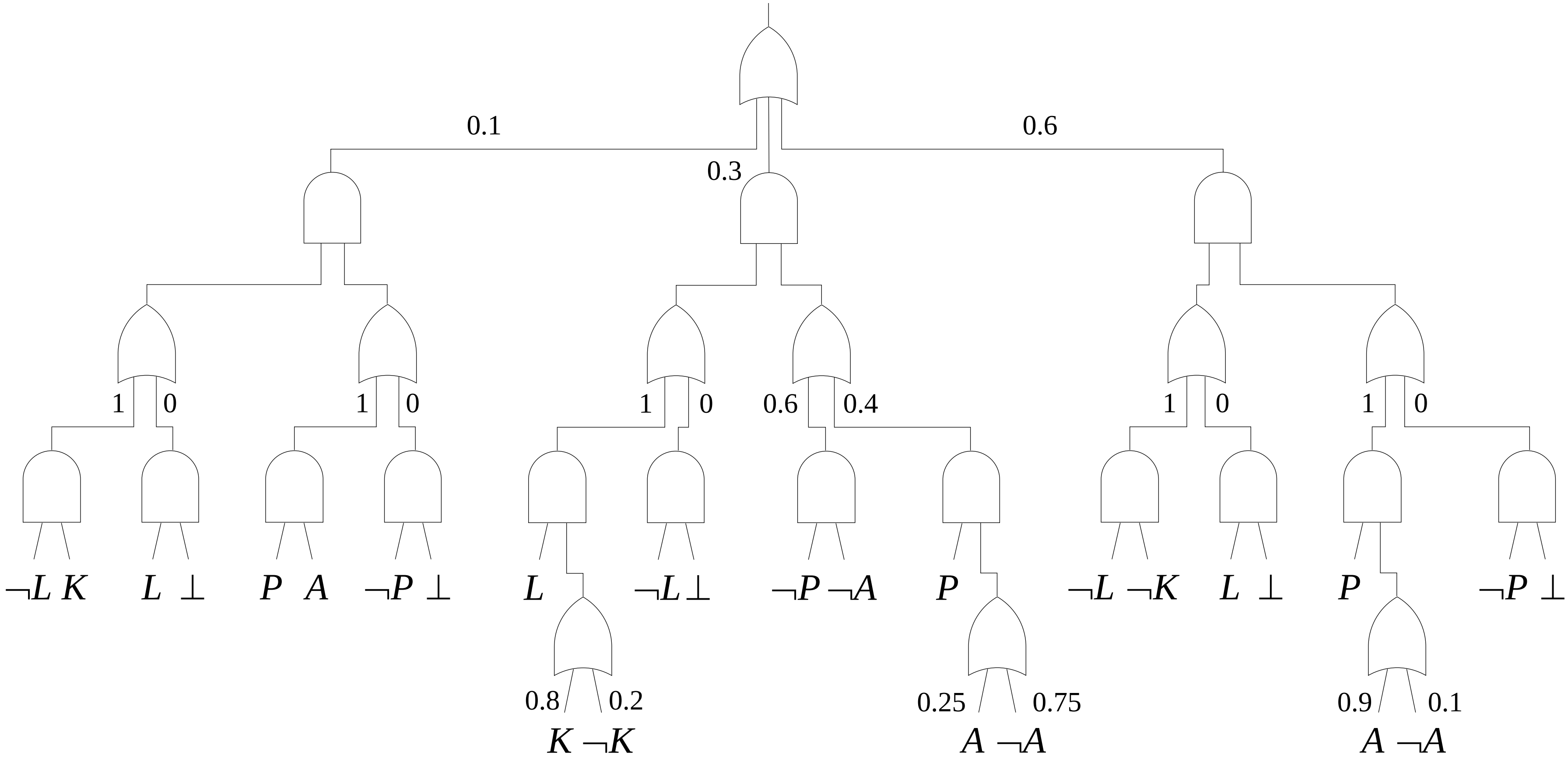
SCIENCE + TECHNOLOGY

Artificial intelligence framework developed by UCLA professor now powers Toyota websites

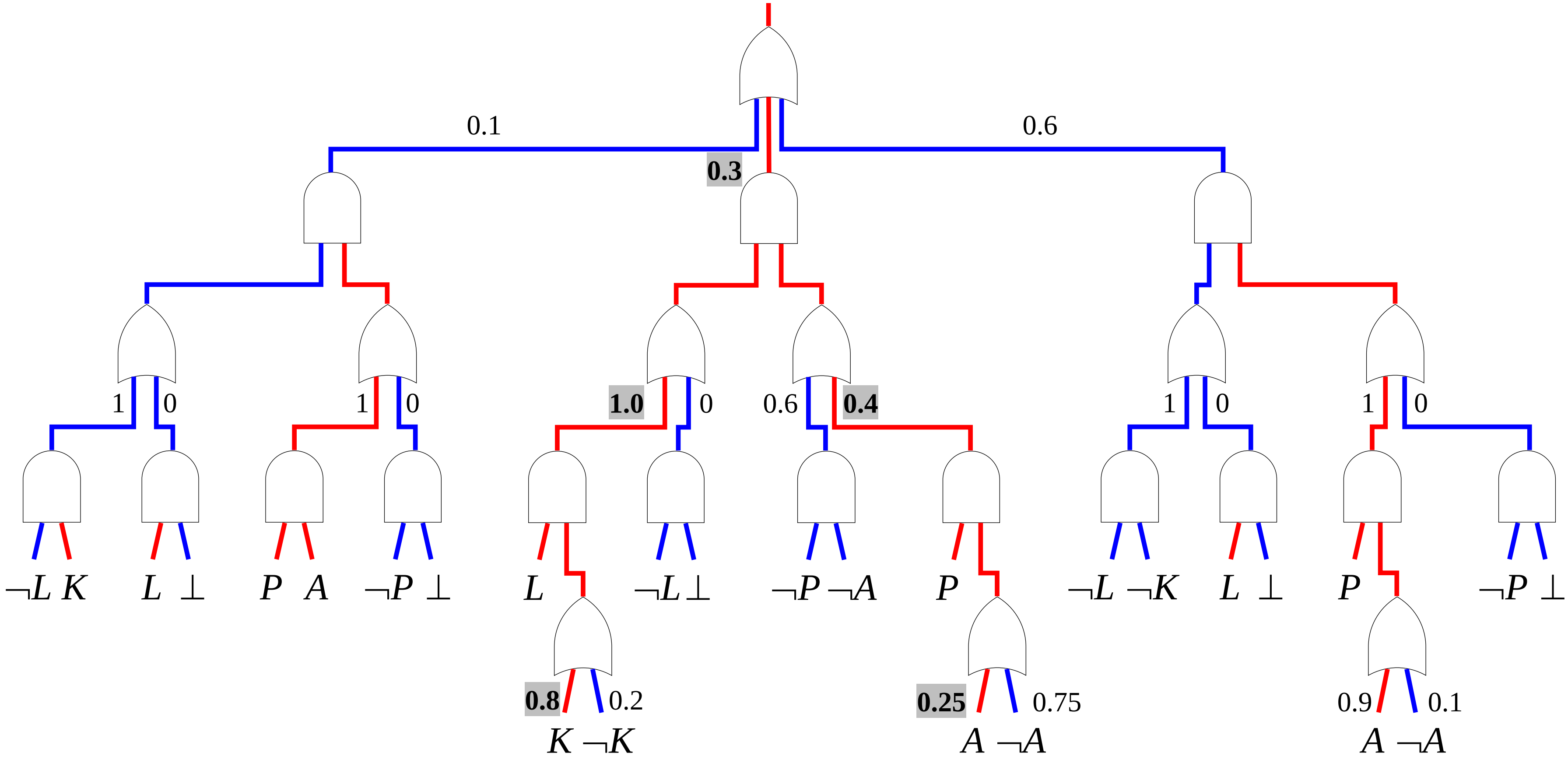
Adnan Darwiche's invention helps consumers customize their vehicles online

Matthew Chin | May 12, 2016

PSDD: Probabilistic SDD

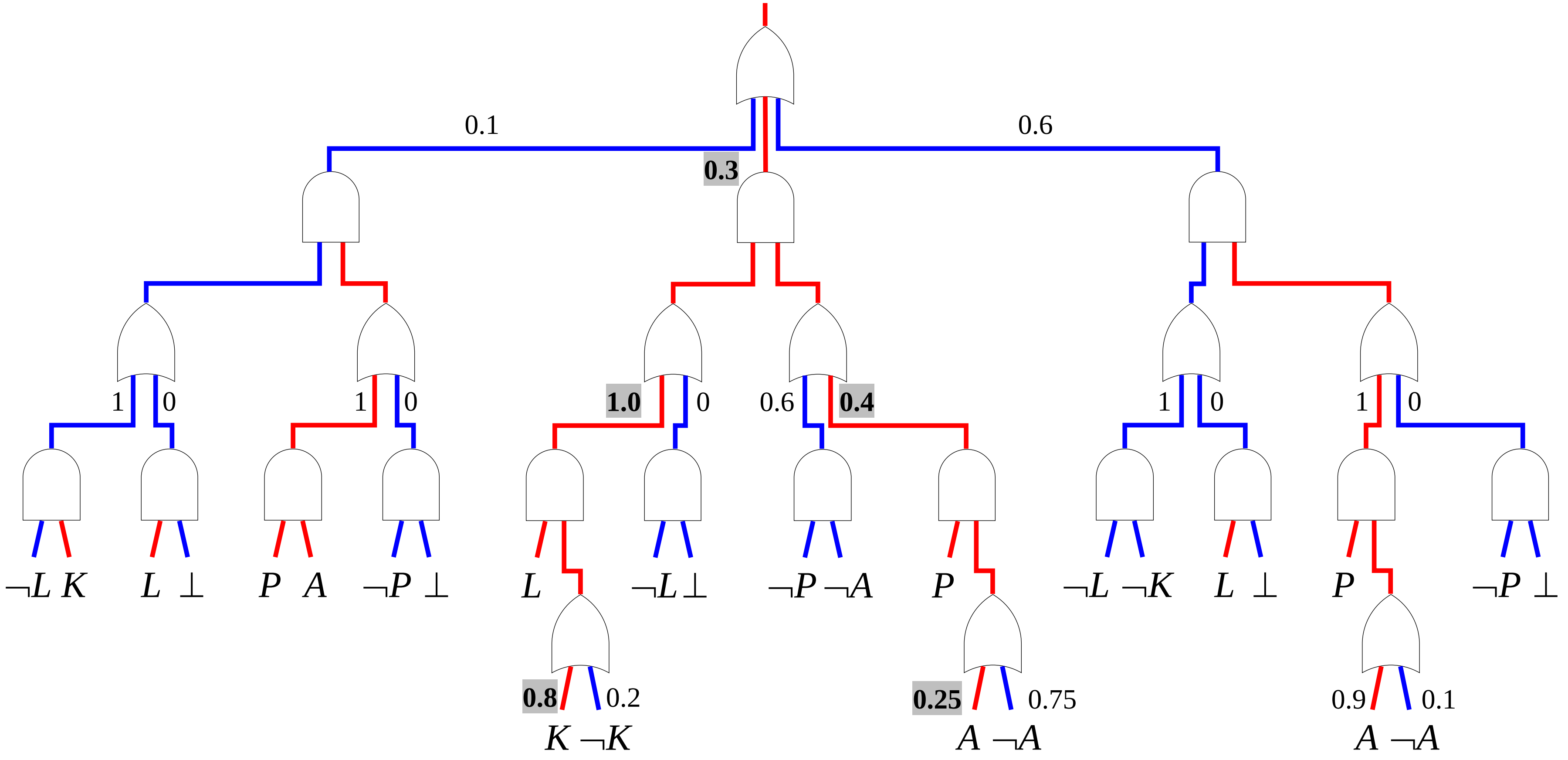


PSDD: Probabilistic SDD



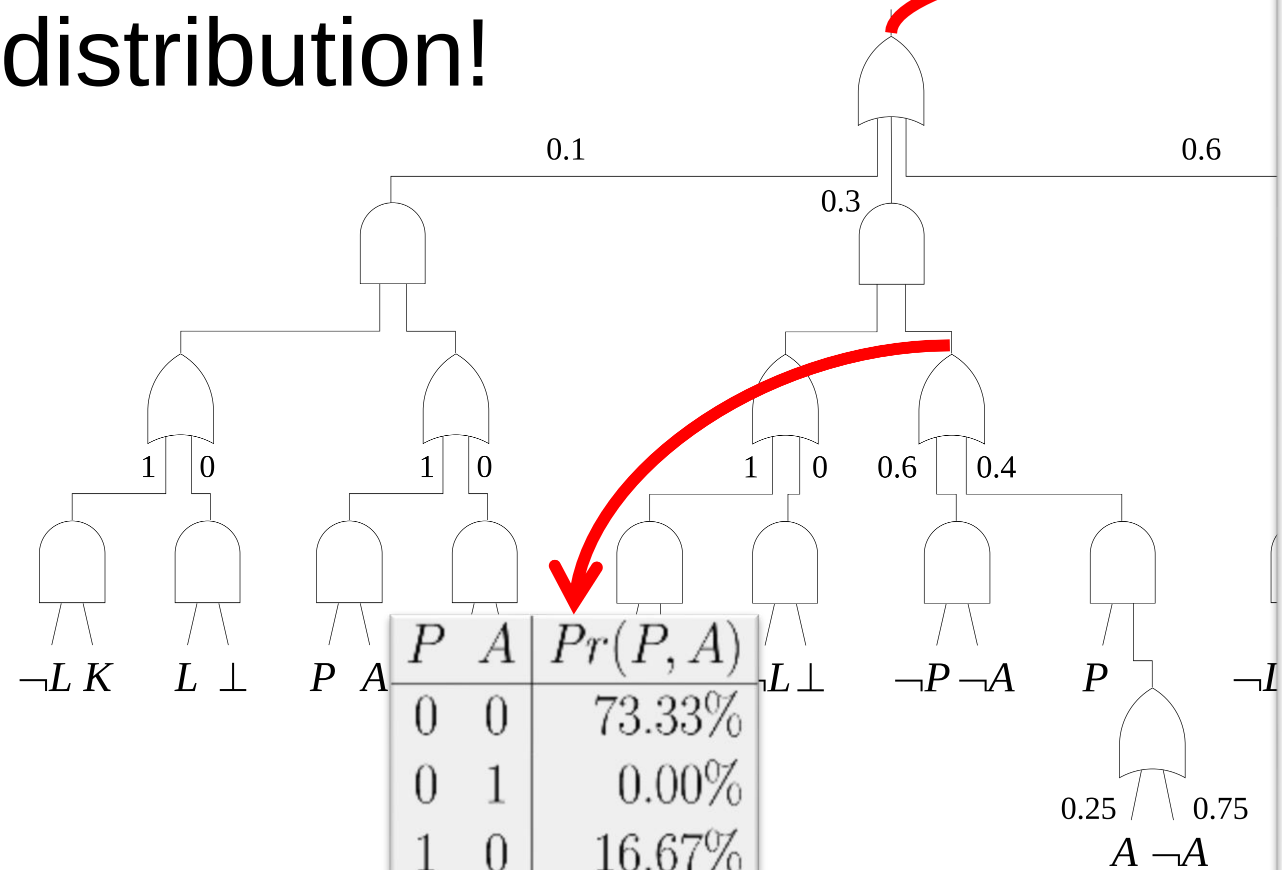
Input: L, K, P, A

PSDD: Probabilistic SDD



Input: L, K, P, A Pr(L,K,P,A) = 0.3 x 1.0 x 0.8 x 0.4 x 0.25 = 0.024

PSDD nodes induce a normalized distribution!



L	K	P	A	Pr(L, K, P, A)
0	0	0	0	0.00%
0	0	0	1	0.00%
0	0	1	0	6.00%
0	0	1	1	54.00%
0	1	0	0	0.00%
0	1	0	1	0.00%
0	1	1	0	0.00%
0	1	1	1	10.00%
1	0	0	0	4.40%
1	0	0	1	0.00%
1	0	1	0	1.00%
1	0	1	1	0.60%
1	1	0	0	17.6%
1	1	0	1	0.00%
1	1	1	0	4.00%
1	1	1	1	2.40%

P	A	Pr(P, A)
0	0	73.33%
0	1	0.00%
1	0	16.67%
1	1	10.00%

Can read independences off the circuit structure

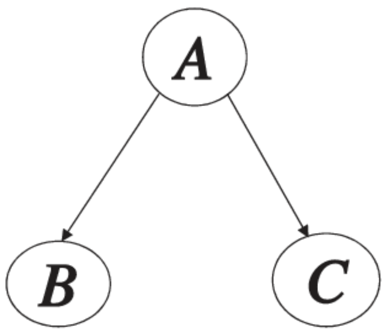
Tractable for Probabilistic Inference

- **MAP inference**: Find most-likely assignment (otherwise NP-complete)
- Computing **conditional probabilities** Pr(x|y) (otherwise PP-complete)
- **Sample** from Pr(x|y)

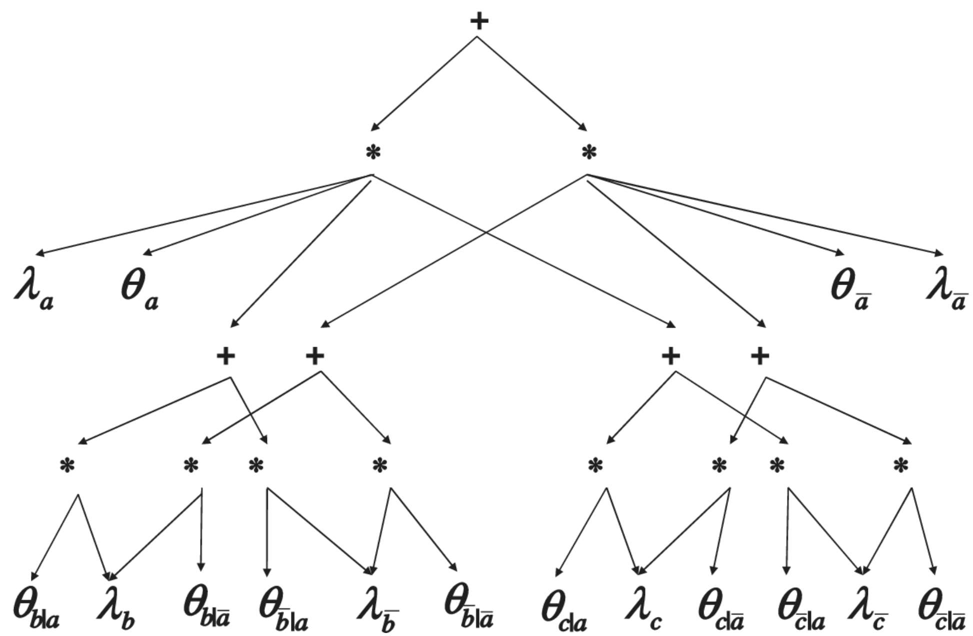
- Algorithms linear in circuit size ☺
  (pass up, pass down, similar to backprop)

PSDDs are Arithmetic Circuits (ACs)

[Darwiche, JACM 2003]



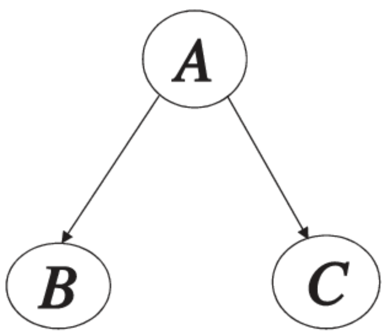
Bayesian Network (BN)



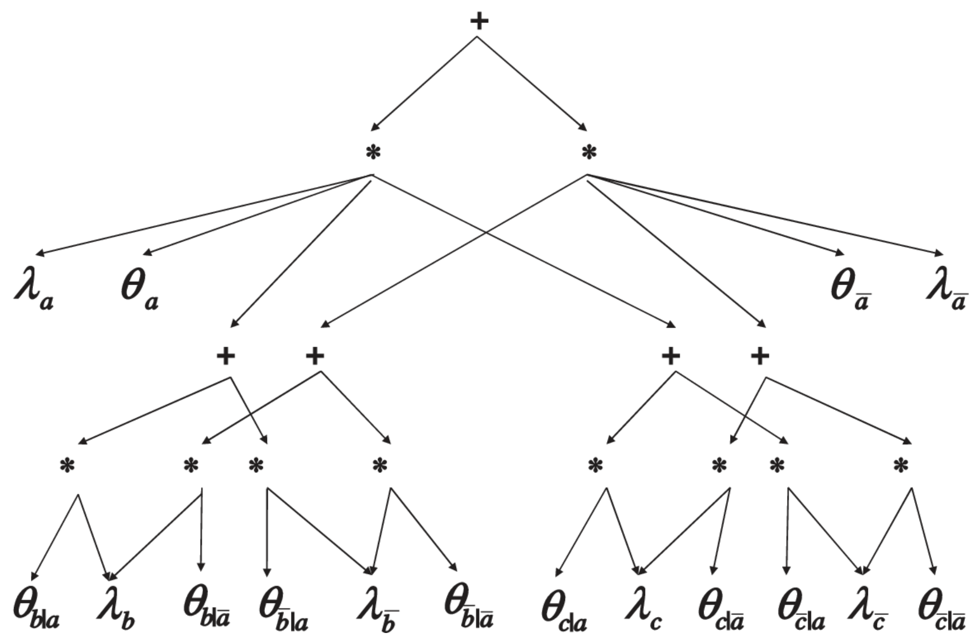
Arithmetic Circuit (AC)

PSDDs are Arithmetic Circuits (ACs)

[Darwiche, JACM 2003]



Bayesian Network (BN)

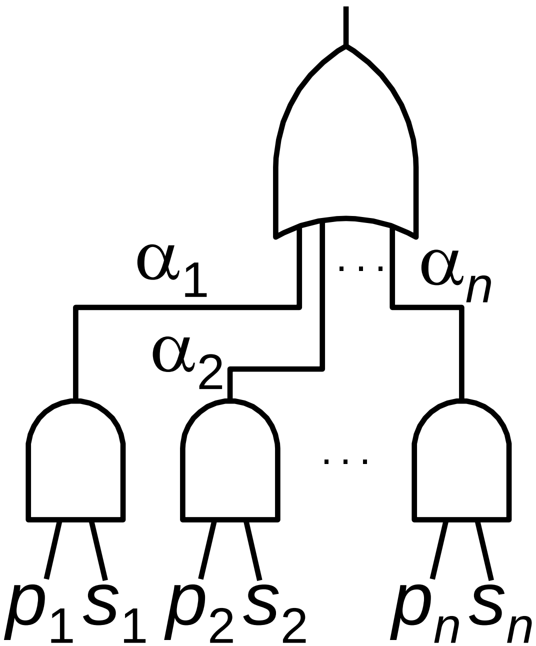


Arithmetic Circuit (AC)

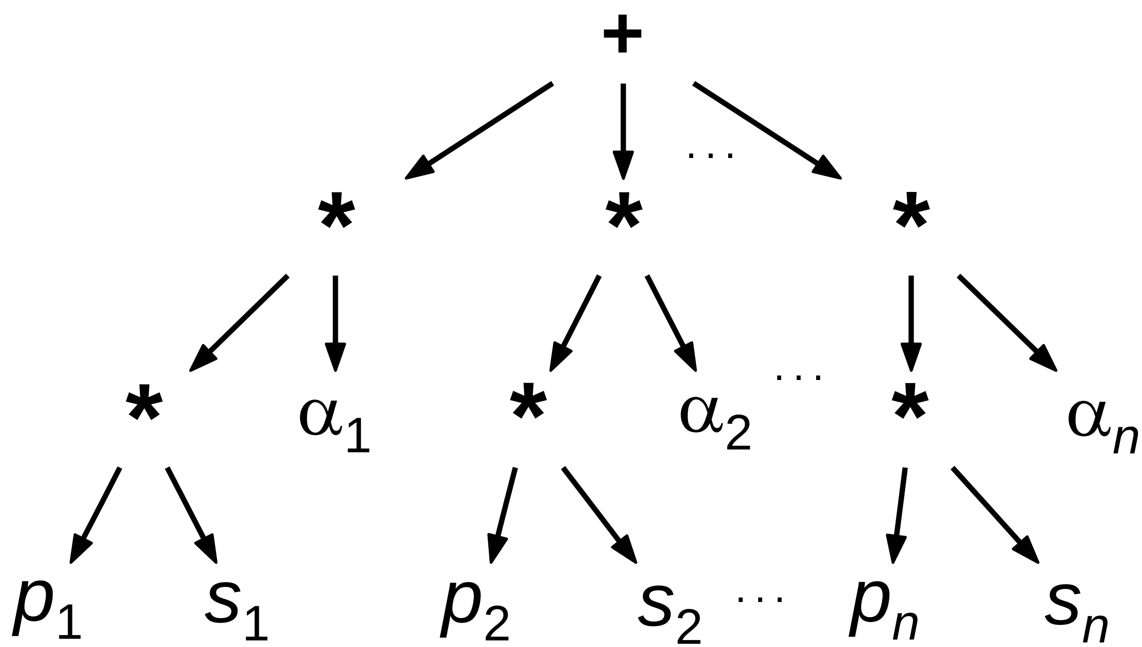
Known in the ML literature as SPNs
UAI 2011, NIPS 2012 best paper awards

[ICML 2014] (SPNs equivalent to ACs)

Result: PSDDs are ACs



PSDD



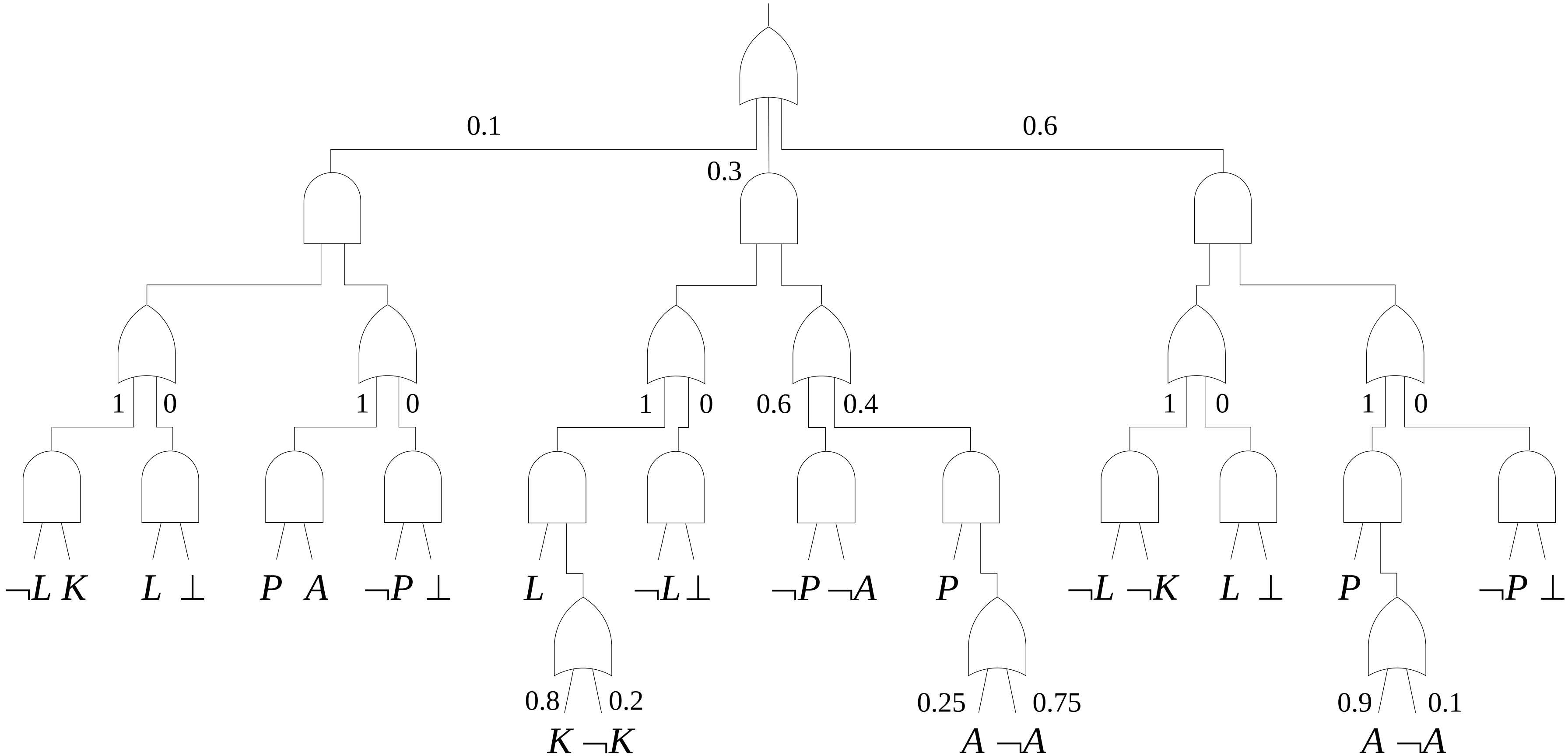
AC

decomposable+ and deterministic+ ACs
(over a structured space)

Learning PSDDs

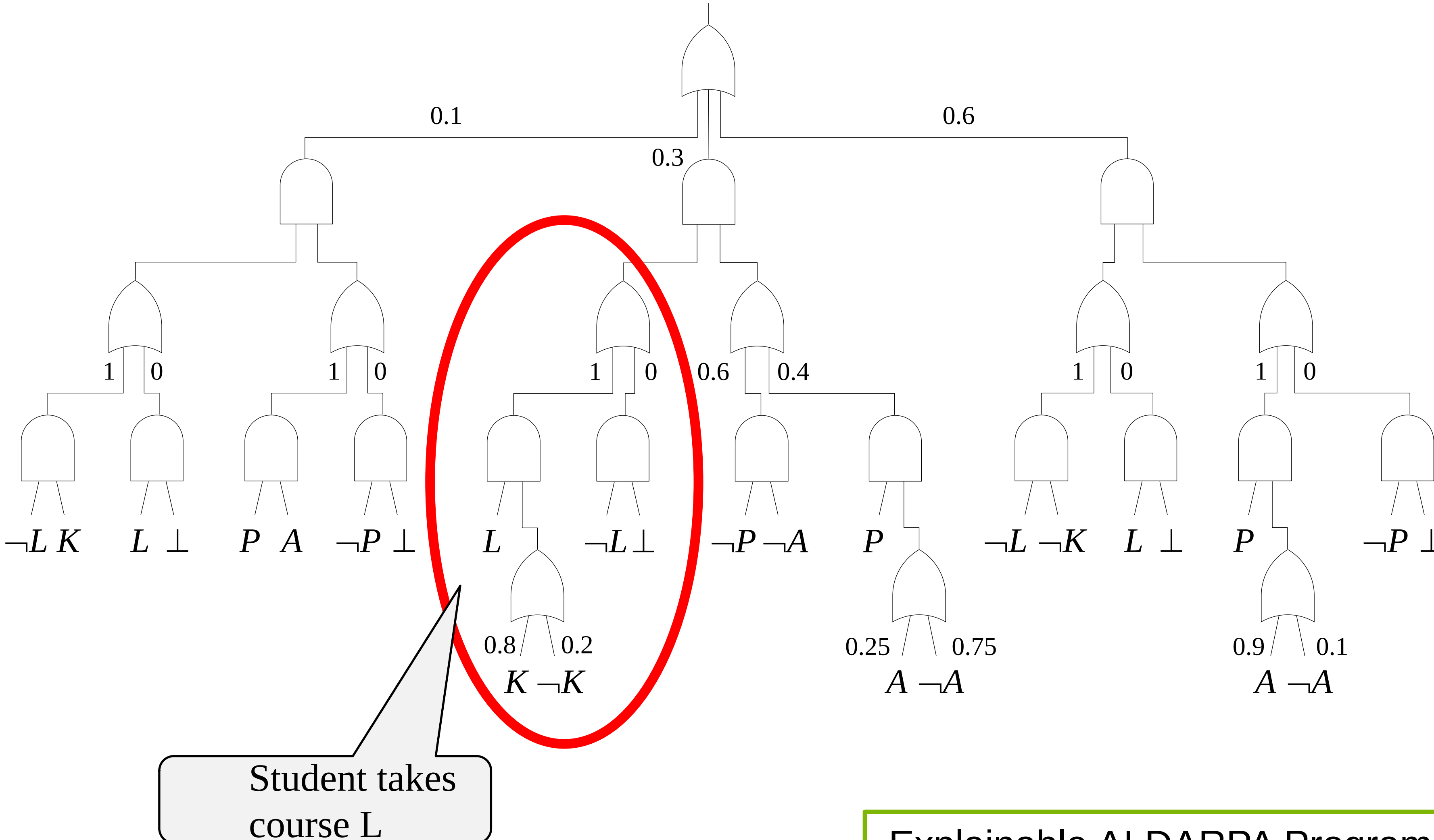
Logic + Probability + ML

Parameters are Interpretable



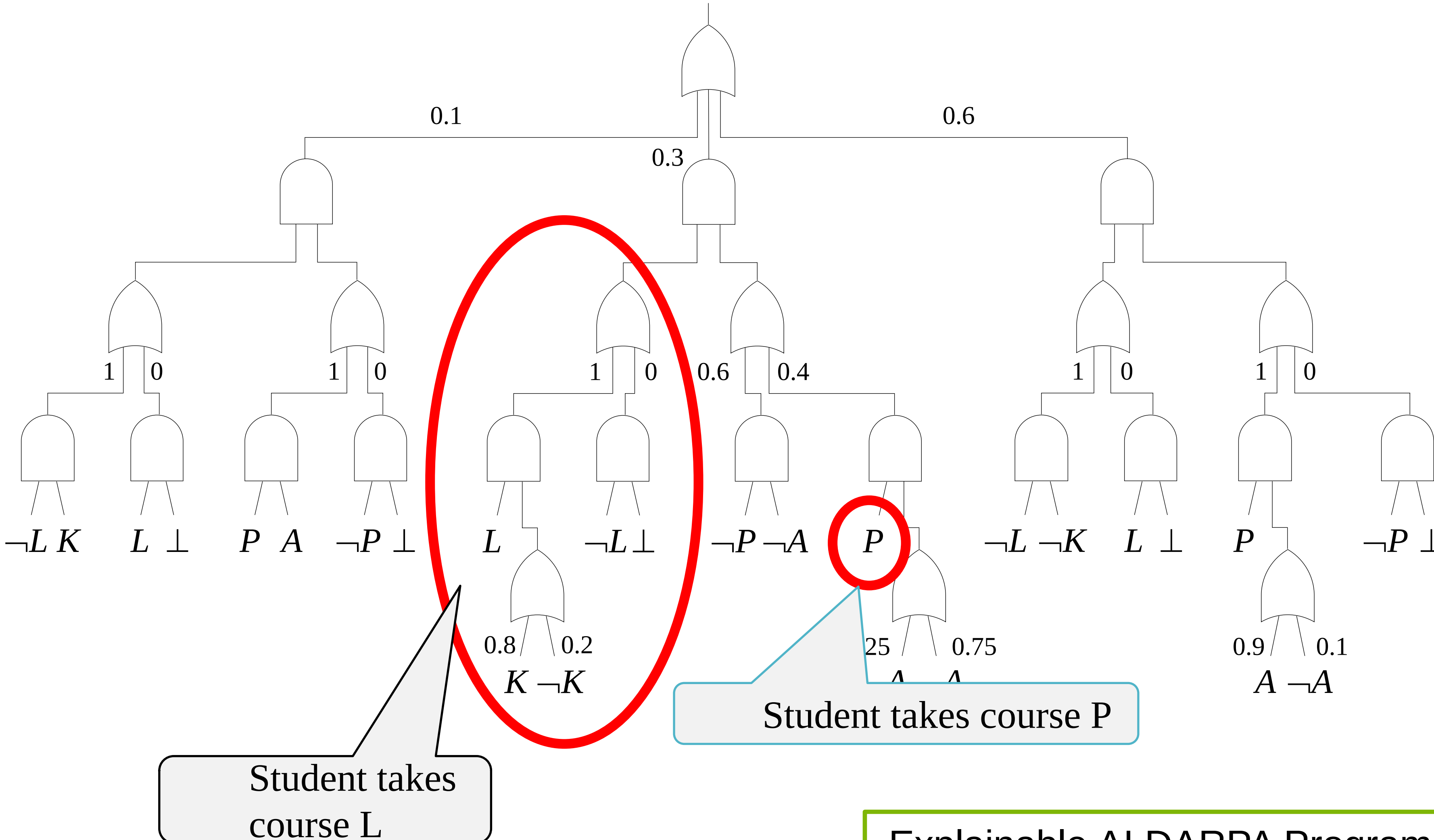
Explainable AI DARPA Program

Parameters are Interpretable



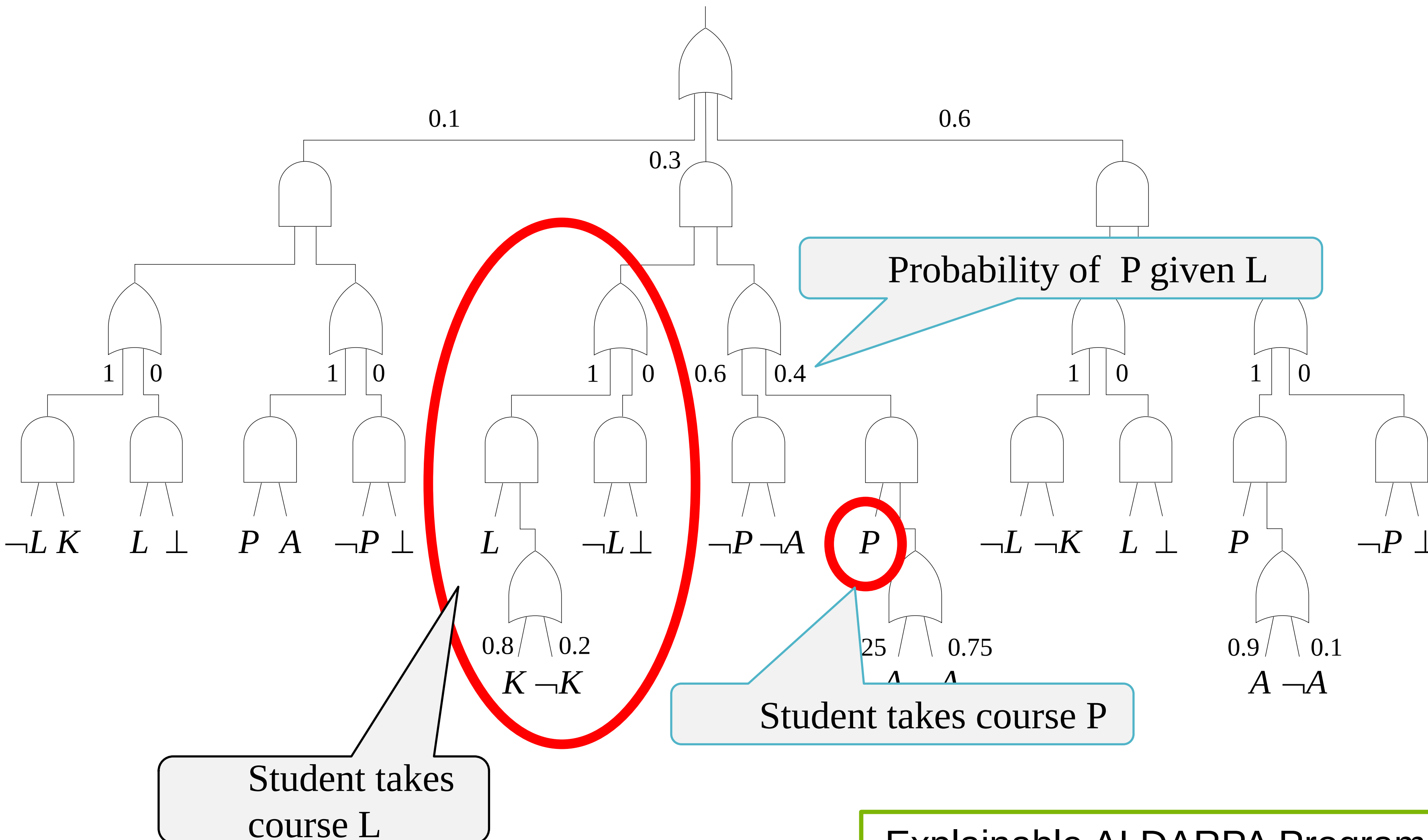
Explainable AI DARPA Program

Parameters are Interpretable



Explainable AI DARPA Program

Parameters are Interpretable



Explainable AI DARPA Program

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

  One pass over data to estimate Pr(x|y)

Note a lot to say: very easy!

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

  One pass over data to estimate Pr(x|y)

Note a lot to say: very easy!

- Structure learning:
  - Compile constraints to SDD

    Use SAT solver technology

    (naive? see later)

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

  One pass over data to estimate Pr(x|y)

Note a lot to say: very easy!

- Structure learning:
  - Compile constraints to SDD

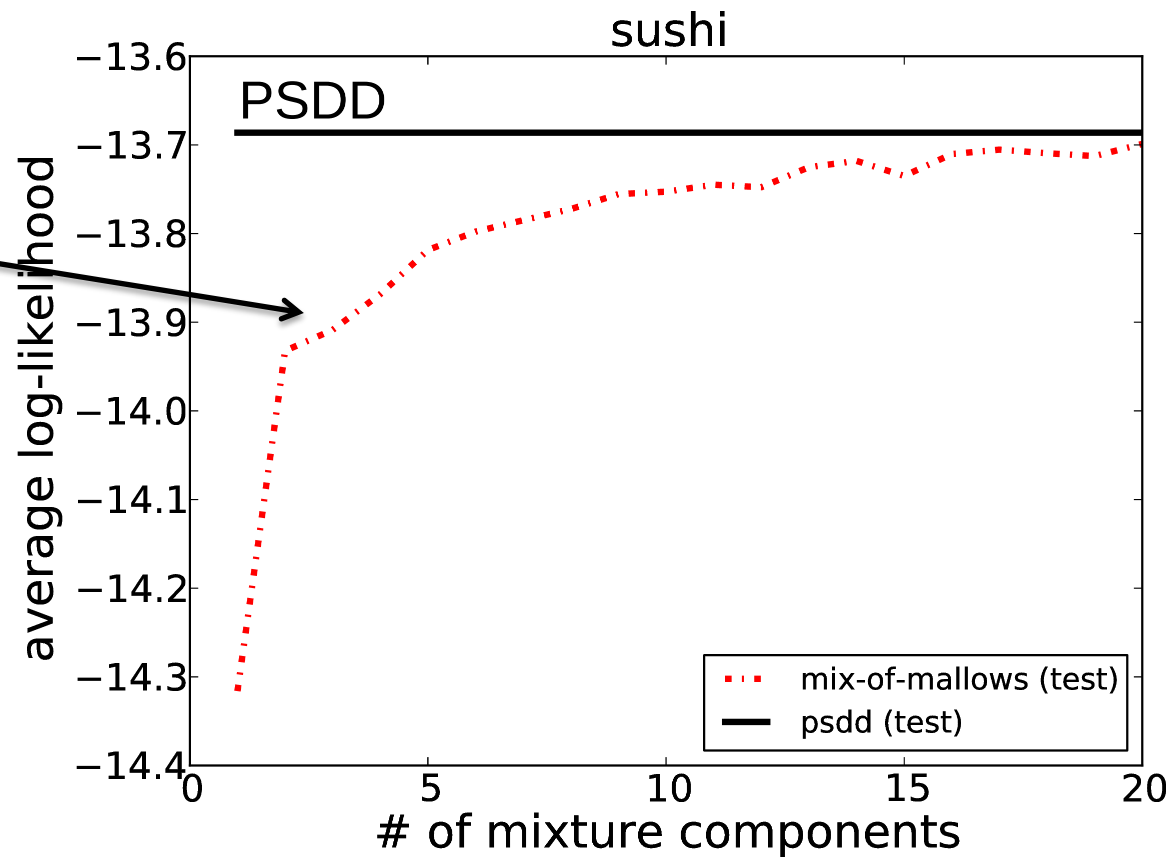
    Use SAT solver technology

    (naive? see later)
  - Search for structure to fit data (ongoing work)

Learning Preference Distributions

Special-purpose distribution: Mixture-of-Mallows

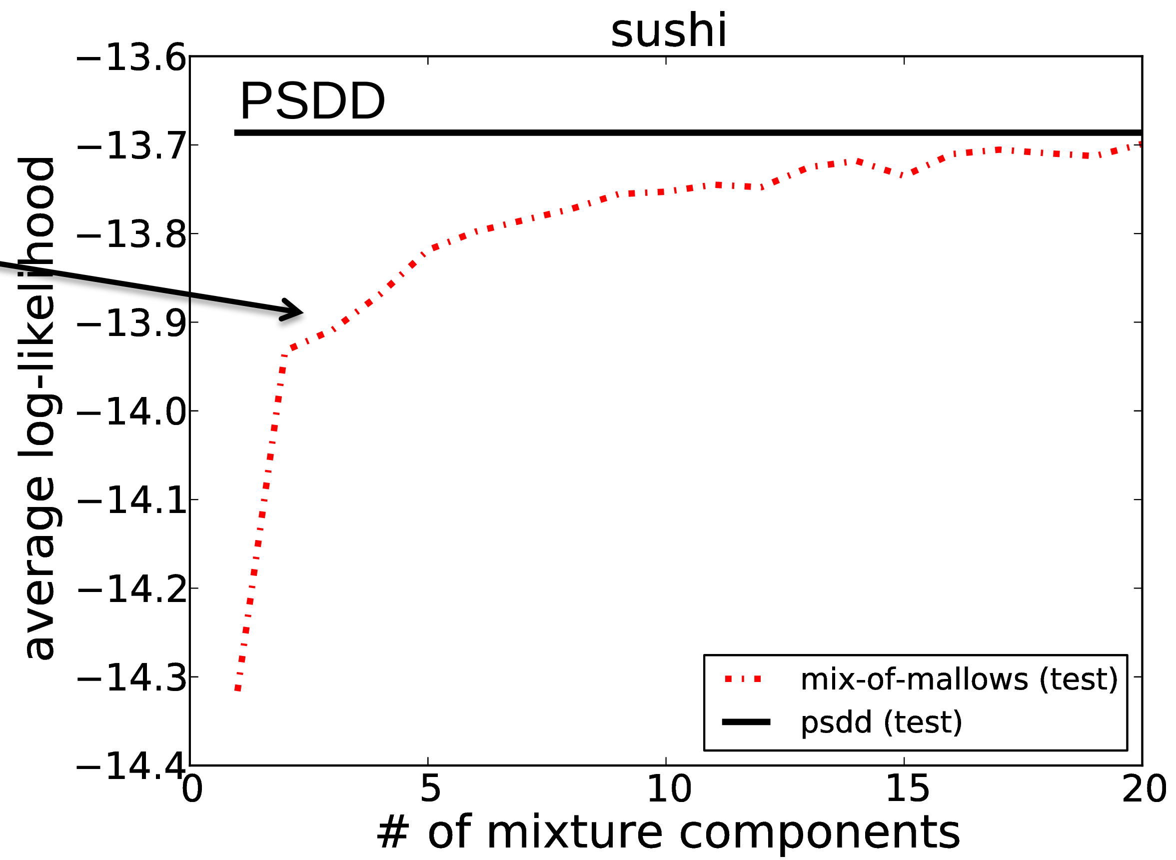
- # of components from 1 to 20
- EM with 10 random seeds
- implementation of Lu & Boutilier



Learning Preference Distributions

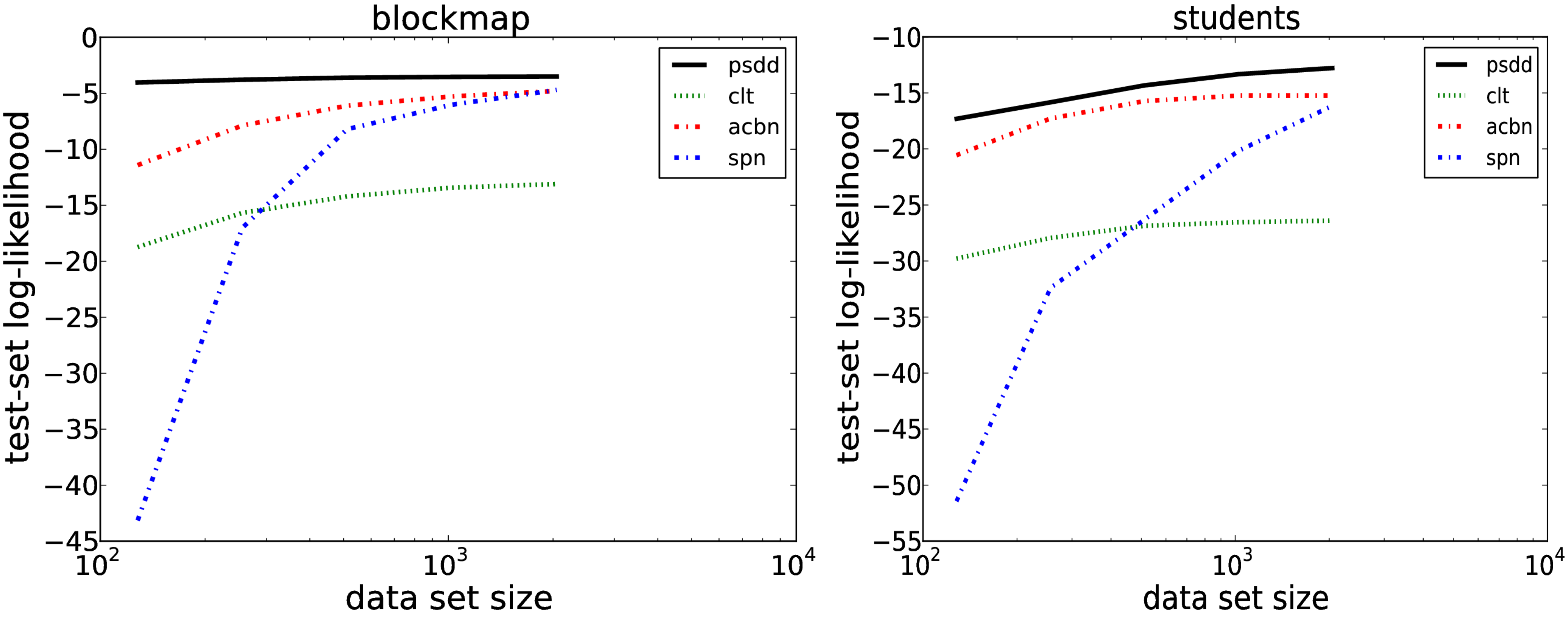
Special-purpose distribution: Mixture-of-Mallows

- # of components from 1 to 20
- EM with 10 random seeds
- implementation of Lu & Boutilier

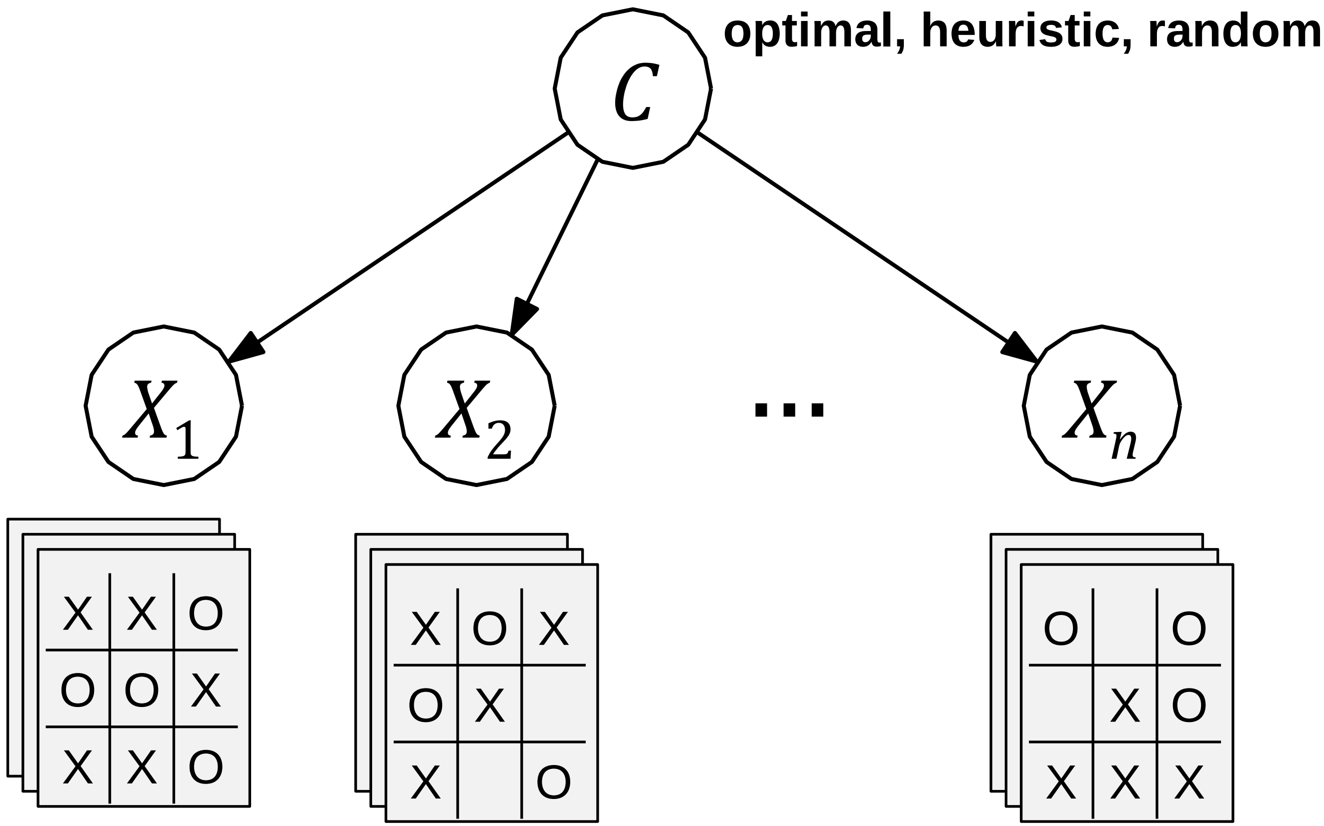


This is the naive approach, without real structure learning!

What happens if you ignore constraints?

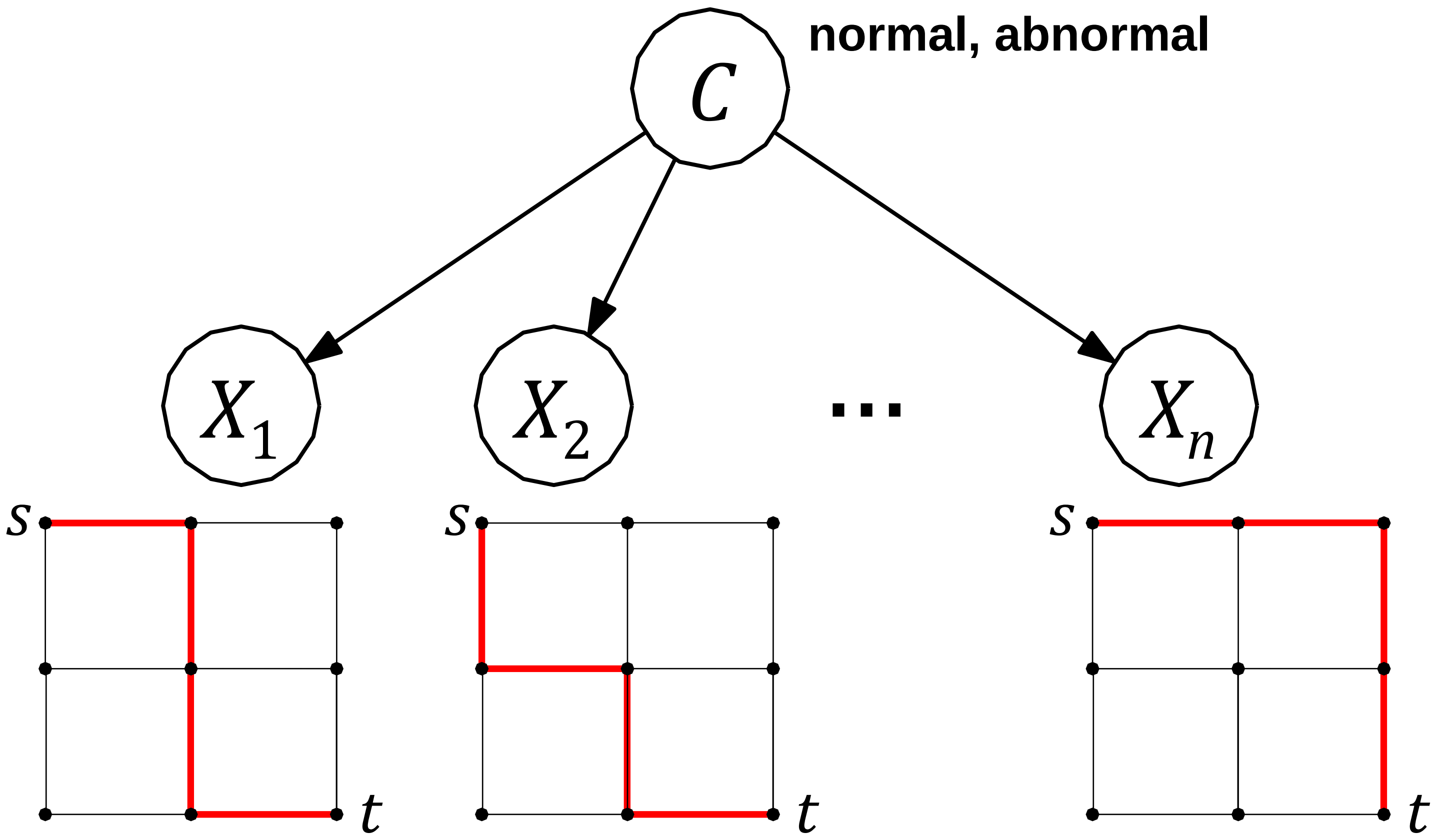


Structured Naïve Bayes Classifier



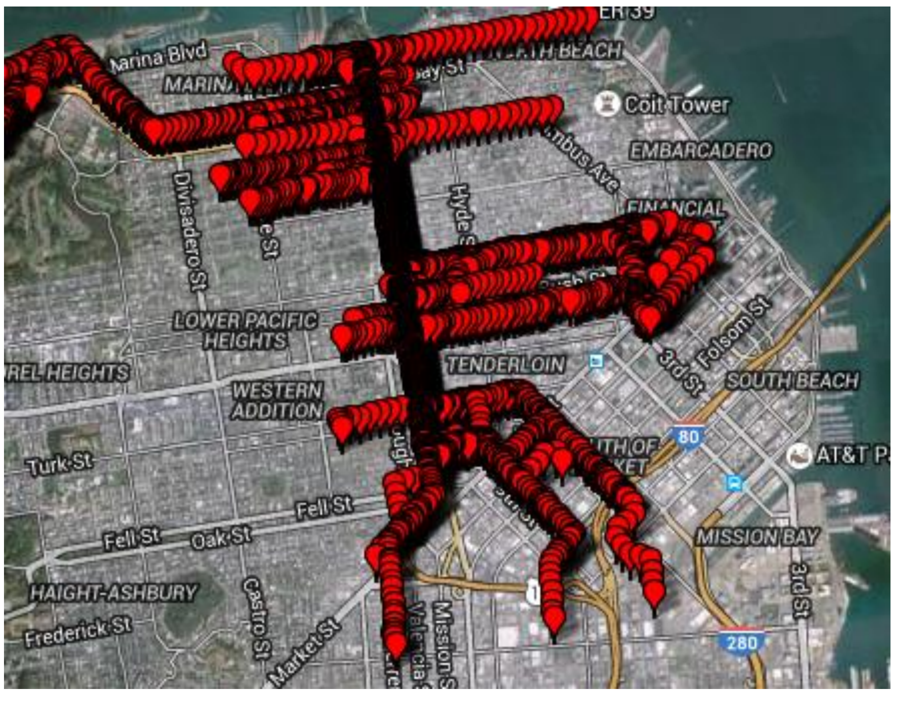
Attribute with 362,880 values (possible game traces)

Structured Naïve Bayes Classifier



Attribute with 789,360,053,252 values (routes in 8 × 8 grid)

Learning Route Distributions (ongoing)



- Uber GPS data in SF
- Project GPS coordinates onto a graph, then learn distributions over routes
- Applications:
  - Detect anomalies
  - Given a partial route, predict its most likely completion

Parameter Estimation

a classical
complete dataset

id	X	Y	Z
1	x_1	y_2	z_1
2	x_2	y_1	z_2
3	x_2	y_1	z_2
4	x_1	y_1	z_1
5	x_1	y_2	z_2

closed-form
(maximum-likelihood
estimates are unique)

a classical
incomplete dataset

id	X	Y	Z
1	x_1	y_2	?
2	x_2	y_1	?
3	?	?	z_2
4	?	y_1	z_1
5	x_1	y_2	z_2

EM algorithm

Parameter Estimation

a classical
complete dataset

id	X	Y	Z
1	x_1	y_2	z_1
2	x_2	y_1	z_2
3	x_2	y_1	z_2
4	x_1	y_1	z_1
5	x_1	y_2	z_2

closed-form
(maximum-likelihood
estimates are unique)

a classical
incomplete dataset

id	X	Y	Z
1	x_1	y_2	**?**
2	x_2	y_1	**?**
3	**?**	**?**	z_2
4	**?**	y_1	z_1
5	x_1	y_2	z_2

EM algorithm

a new type of
incomplete dataset

id	X	Y	Z
1	$X \equiv Z$		
2	x_2 and (y_2 or z_2)		
3	$x_2 \Rightarrow y_1$		
4	$X \oplus Y \oplus Z \equiv 1$		
5	x_1 and y_2 and z_2		

Missed in the
ML literature

Structured Datasets

a classical **complete** dataset
(e.g., total rankings)

id	1^{st} sushi	2^{nd} sushi	3^{rd} sushi	…
1	fatty tuna	sea urchin	salmon roe	…
2	fatty tuna	tuna	shrimp	…
3	tuna	tuna roll	sea eel	…
4	fatty tuna	salmon roe	tuna	…
5	egg	squid	shrimp	…

a classical **incomplete** dataset
(e.g., top-*k* rankings)

id	1^{st} sushi	2^{nd} sushi	3^{rd} sushi	…
1	fatty tuna	sea urchin	?	…
2	fatty tuna	?	?	…
3	tuna	tuna roll	?	…
4	fatty tuna	salmon roe	?	…
5	egg	?	?	…

Structured Datasets

a classical **complete** dataset
(e.g., total rankings)

id	1^{st} sushi	2^{nd} sushi	3^{rd} sushi	…
1	fatty tuna	sea urchin	salmon roe	…
2	fatty tuna	tuna	shrimp	…
3	tuna	tuna roll	sea eel	…
4	fatty tuna	salmon roe	tuna	…
5	egg	squid	shrimp	…

a new type of **incomplete** dataset
(e.g., **partial** rankings)

id	1^{st} sushi	2^{nd} sushi	3^{rd} sushi	…
1	(fatty tuna > sea urchin) and (tuna > sea eel)			…
2	(fatty tuna is 1^{st}) and (salmon roe > egg)			…
3	tuna > squid			…
4	egg is last			…
5	egg > squid > shrimp			…

(represents constraints on possible *total rankings*)

Learning from Incomplete Data

- Movielens Dataset:
  - 3,900 movies, 6,040 users, 1m ratings
  - take ratings from 64 most rated movies
  - ratings 1-5 converted to pairwise prefs.
- PSDD for **partial** rankings
  - 4 tiers
  - 18,711 parameters

movies by expected tier

rank	movie
1	The Godfather
2	The Usual Suspects
3	Casablanca
4	The Shawshank Redemption
5	Schindler’s List
6	One Flew Over the Cuckoo’s Nest
7	The Godfather: Part II
8	Monty Python and the Holy Grail
9	Raiders of the Lost Ark
10	Star Wars IV: A New Hope

PSDD Sizes

items	tier size	Size		
n	k	SDD	Structured Space	Unstructured Space
8	2	443	840	$1.84 \cdot 10^{19}$
27	3	4,114	$1.18 \cdot 10^{9}$	$2.82 \cdot 10^{219}$
64	4	23,497	$3.56 \cdot 10^{18}$	$1.04 \cdot 10^{1233}$
125	5	94,616	$3.45 \cdot 10^{31}$	$3.92 \cdot 10^{4703}$
216	6	297,295	$1.57 \cdot 10^{48}$	$7.16 \cdot 10^{14044}$
343	7	781,918	$4.57 \cdot 10^{68}$	$7.55 \cdot 10^{35415}$

Structured Queries

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

Structured Queries

- no other Star Wars movie in top-5
- at least one comedy in top-5

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

Structured Queries

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

- no other Star Wars movie in top-5
- at least one comedy in top-5

rank	movie
1	Star Wars V: The Empire Strikes Back
2	American Beauty
3	The Godfather
4	The Usual Suspects
5	The Shawshank Redemption

Structured Queries

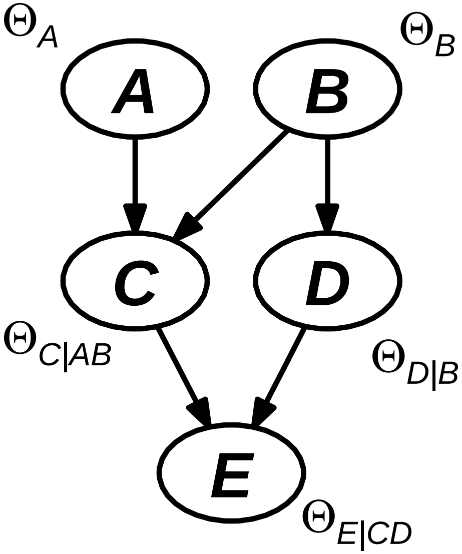
rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

- no other Star Wars movie in top-5
- at least one comedy in top-5

rank	movie
1	Star Wars V: The Empire Strikes Back
2	American Beauty
3	The Godfather
4	The Usual Suspects
5	The Shawshank Redemption

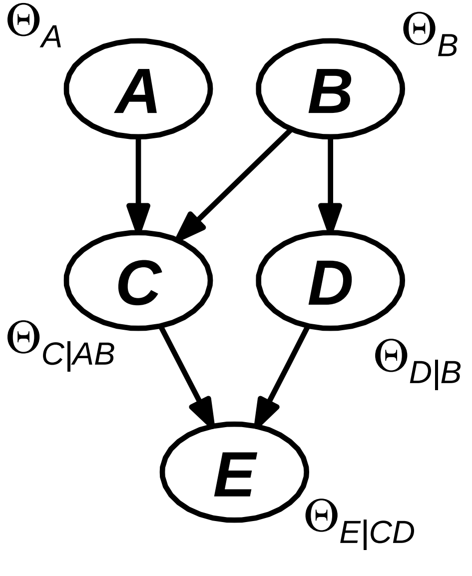
diversified recommendations via
logical constraints

Compiling PGMs into PSDDs



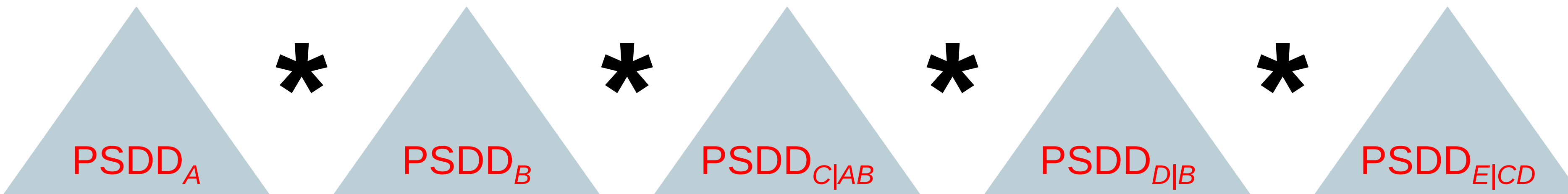
Compiling PGMs into PSDDs

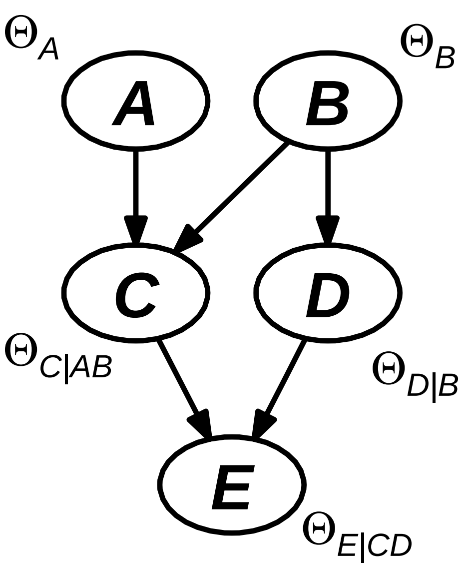
$$\Pr(A,B,C,D,E) = \Theta_A\ \Theta_B\ \Theta_{C|AB}\ \Theta_{D|B}\ \Theta_{E|CD}$$



Compiling PGMs into PSDDs

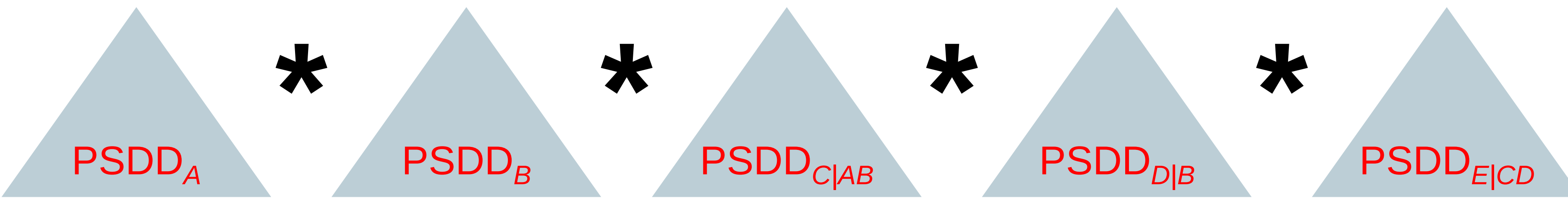
$$\Pr(A,B,C,D,E) = \Theta_A\ \Theta_B\ \Theta_{C|AB}\ \Theta_{D|B}\ \Theta_{E|CD}$$

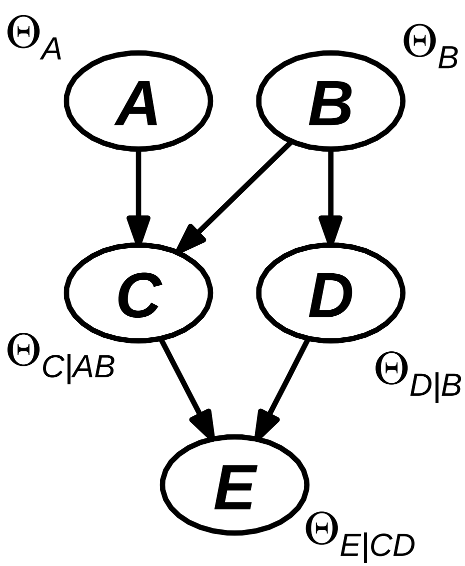




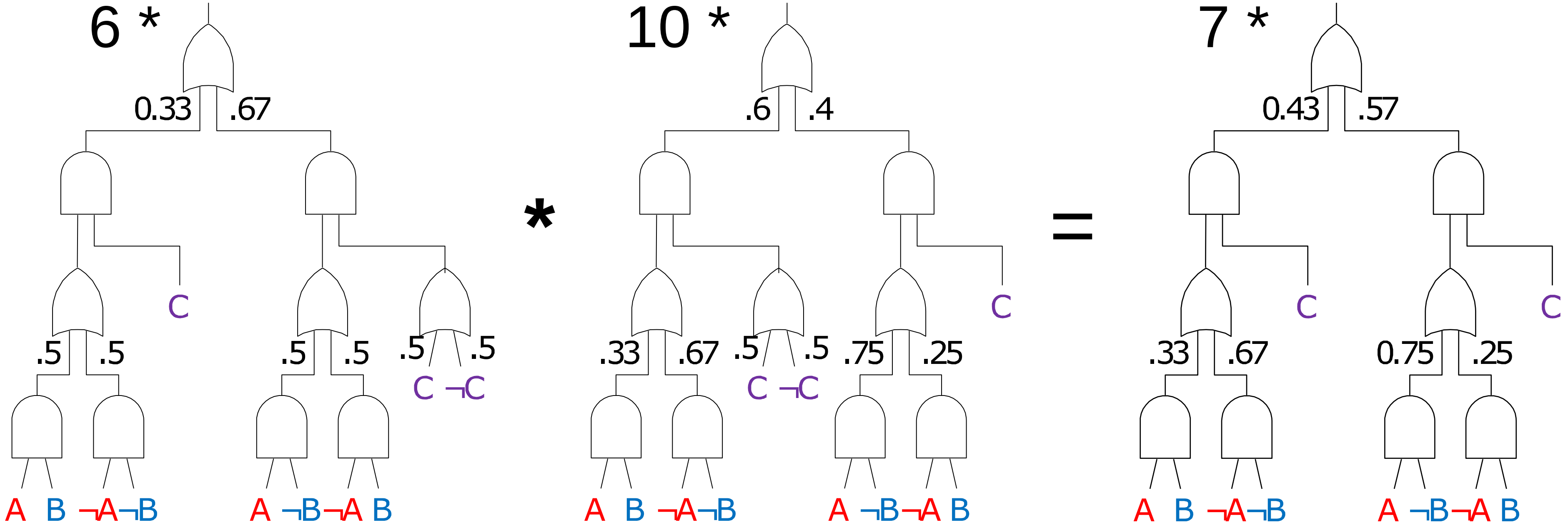
Compiling PGMs into PSDDs

$$\Pr(A,B,C,D,E) = \Theta_A\ \Theta_B\ \Theta_{C|AB}\ \Theta_{D|B}\ \Theta_{E|CD}$$





Sparse tables [Larkin & Decther 2003], ADDs [Bahar, et al. 1993], AOMDDs [Mateescu, et al., 2008], PDGs [Jaeger, 2004]



A	B	C	f
T	T	T	1
T	T	F	0
T	F	T	1
T	F	F	1
F	T	T	1
F	T	F	1
F	F	T	1
F	F	F	0

*

A	B	C	g
T	T	T	1
T	T	F	1
T	F	T	3
T	F	F	0
F	T	T	1
F	T	F	0
F	F	T	2
F	F	F	2

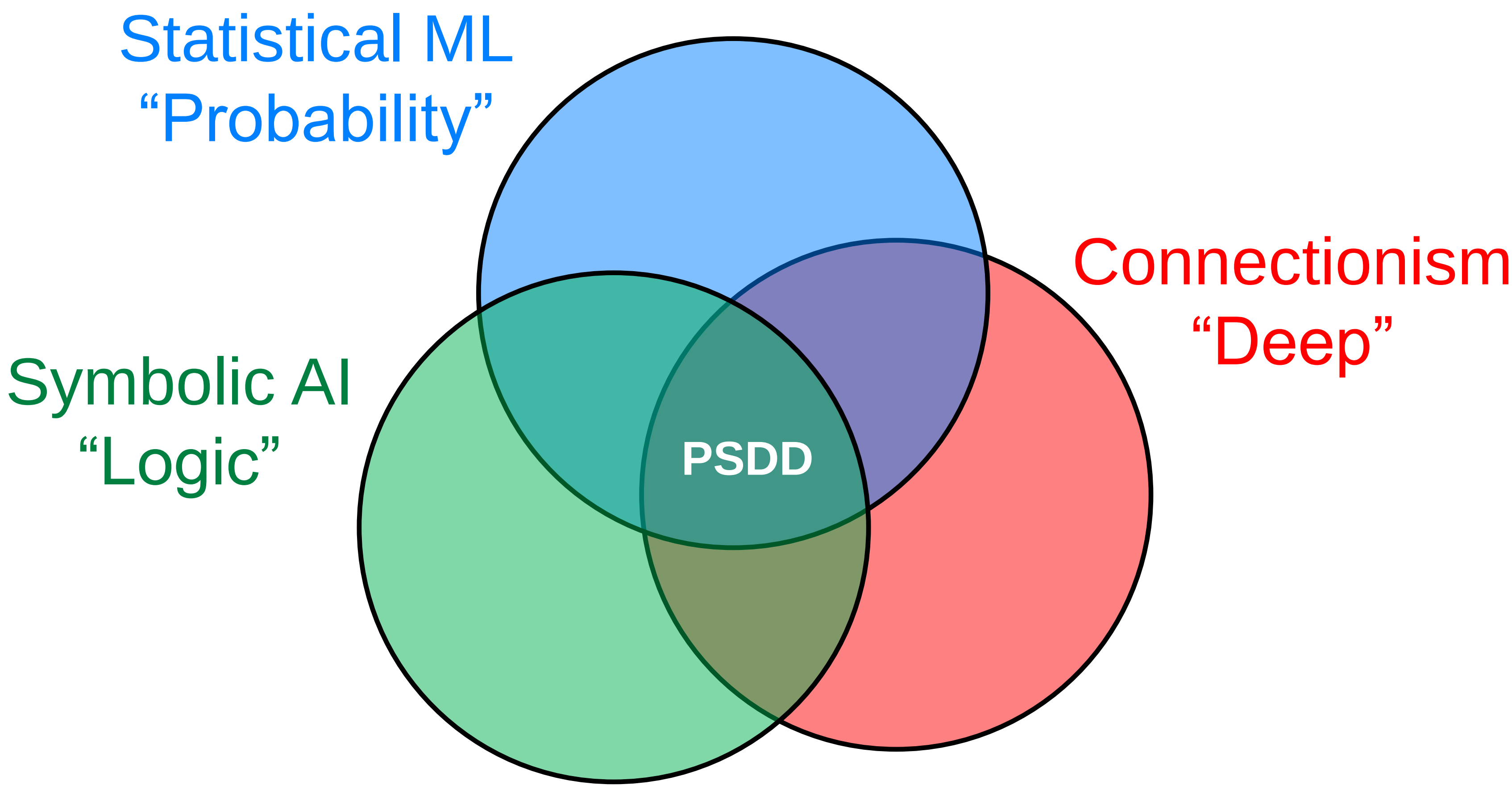
=

A	B	C	f*g
T	T	T	1
T	T	F	0
T	F	T	3
T	F	F	0
F	T	T	1
F	T	F	0
F	F	T	2
F	F	F	0

Conclusions

- Structured spaces are everywhere ☺
- Roles of Boolean constraints in ML
  - Domain constraints and combinatorial objects (**structured probability space**)
  - Incomplete examples (**structured datasets**)
  - Questions and evidence (**structured queries**)
- Learn distributions over combinatorial objects
- Strong properties for inference and learning

Conclusions



References

Probabilistic Sentential Decision Diagrams
Doga Kisa, Guy Van den Broeck, Arthur Choi and Adnan Darwiche
KR, 2014

Learning with Massive Logical Constraints
Doga Kisa, Guy Van den Broeck, Arthur Choi and Adnan Darwiche
ICML 2014 workshop

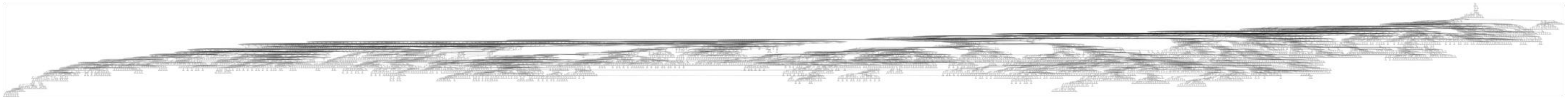
Tractable Learning for Structured Probability Spaces
Arthur Choi, Guy Van den Broeck and Adnan Darwiche
IJCAI, 2015

Tractable Learning for Complex Probability Queries
Jessa Bekker, Jesse Davis, Arthur Choi, Adnan Darwiche, Guy Van den Broeck.
NIPS, 2015

Structured Features in Naive Bayes Classifiers
Arthur Choi, Nazgol Tavabi and Adnan Darwiche
AAAI, 2016

Tractable Operations on Arithmetic Circuits
Jason Shen, Arthur Choi and Adnan Darwiche
NIPS, 2016

Questions?



PSDD with 15,000 nodes