Logic and Probability

The Computational Connection

Adnan Darwiche
UCLA

Deduction at Scale Seminar, March, 2011

Inference

- **Probabilistic Graphical Models:**
  Marginal and conditional probabilities
  Most likely instantiations…

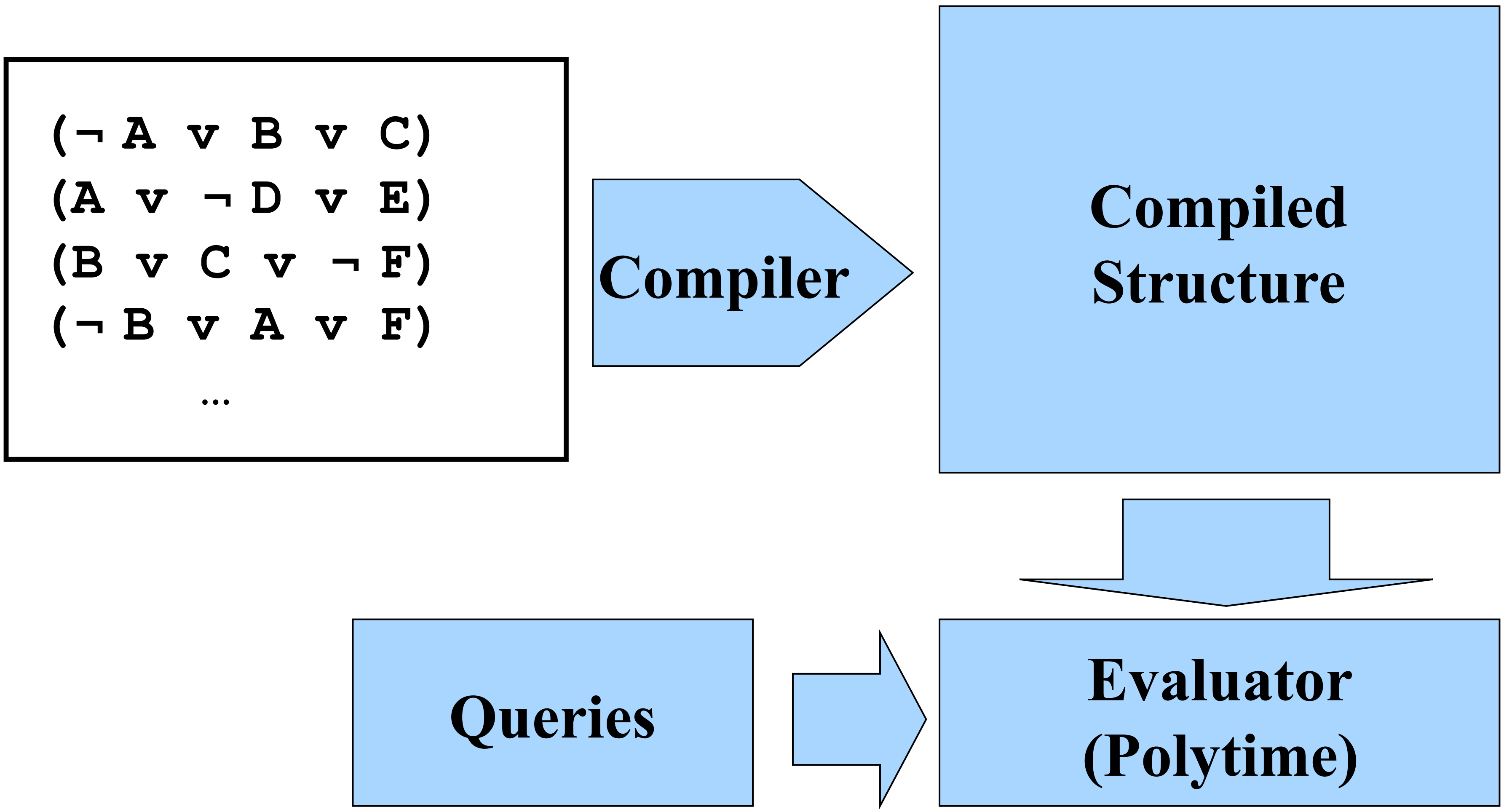
- **Propositional Knowledge Bases:**
  Logical entailment
  Existential quantification
  Model counting…

Two Main Themes

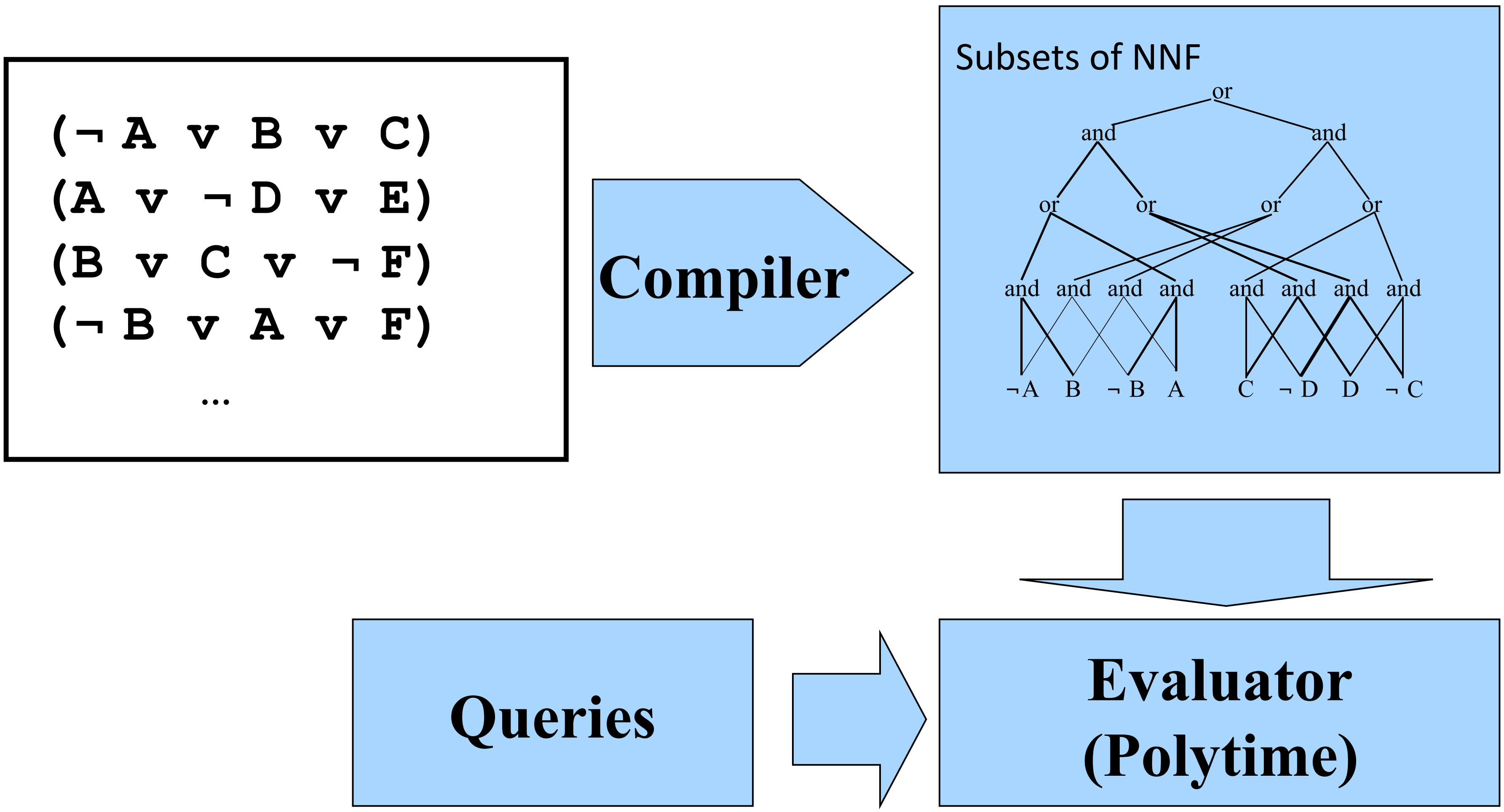
- **Exact inference as:**
  Enforcing decomposability and determinism on propositional knowledge bases

- **Approximate inference as:**
  Relaxing, compensating for, and recovering equivalence constraints (equalities)

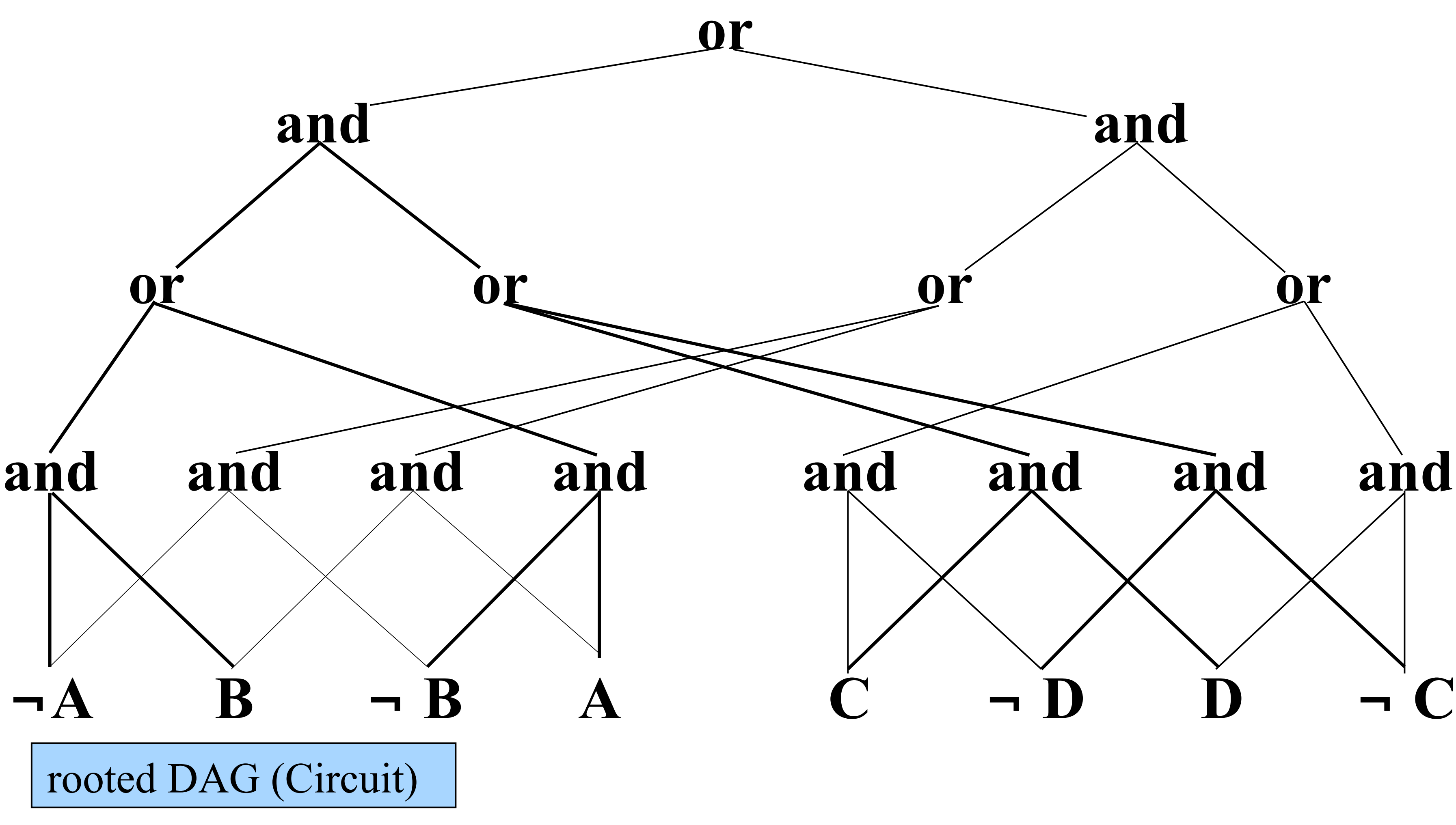
Knowledge Compilation



Knowledge Compilation

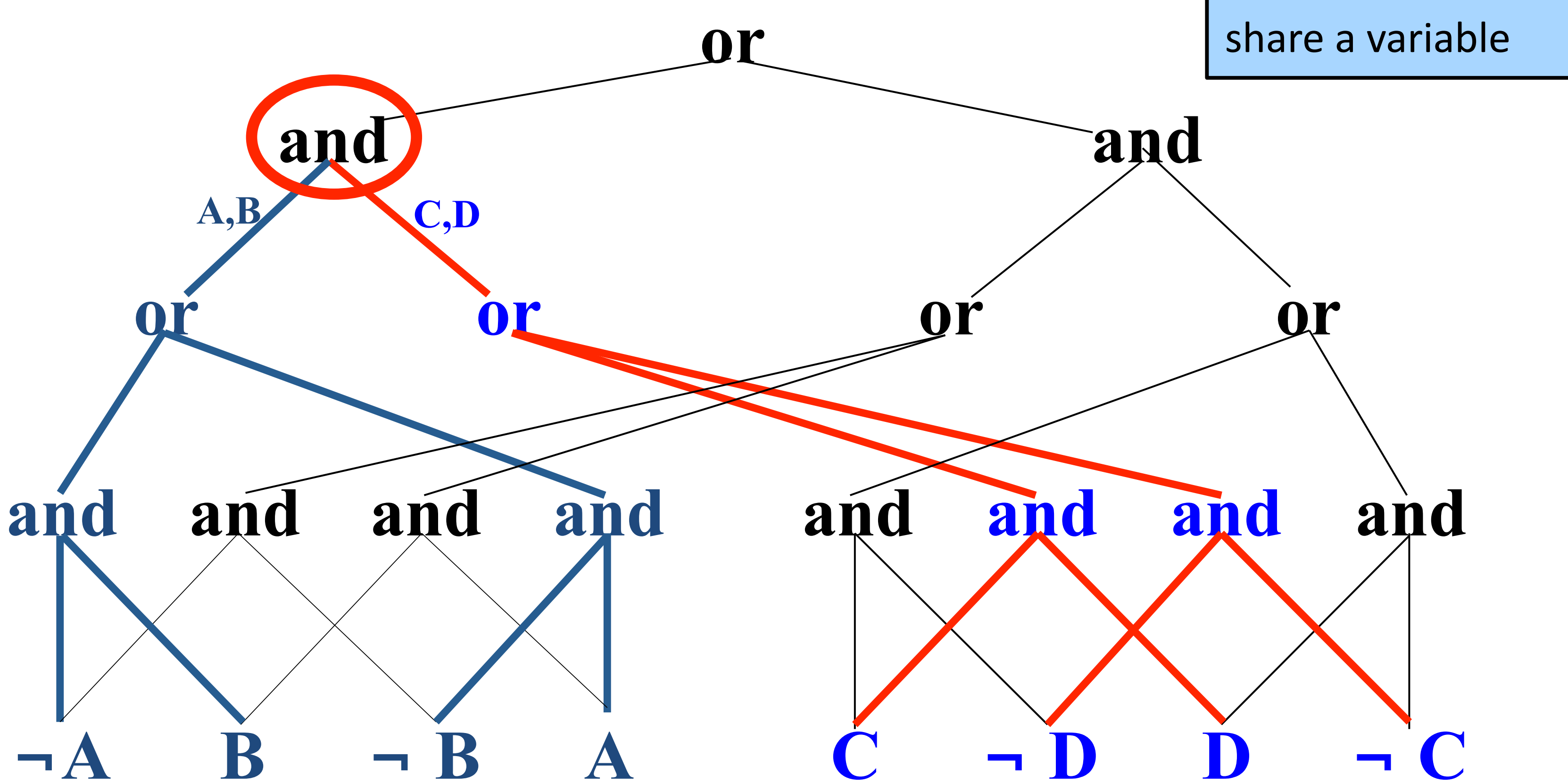


Negation Normal Form



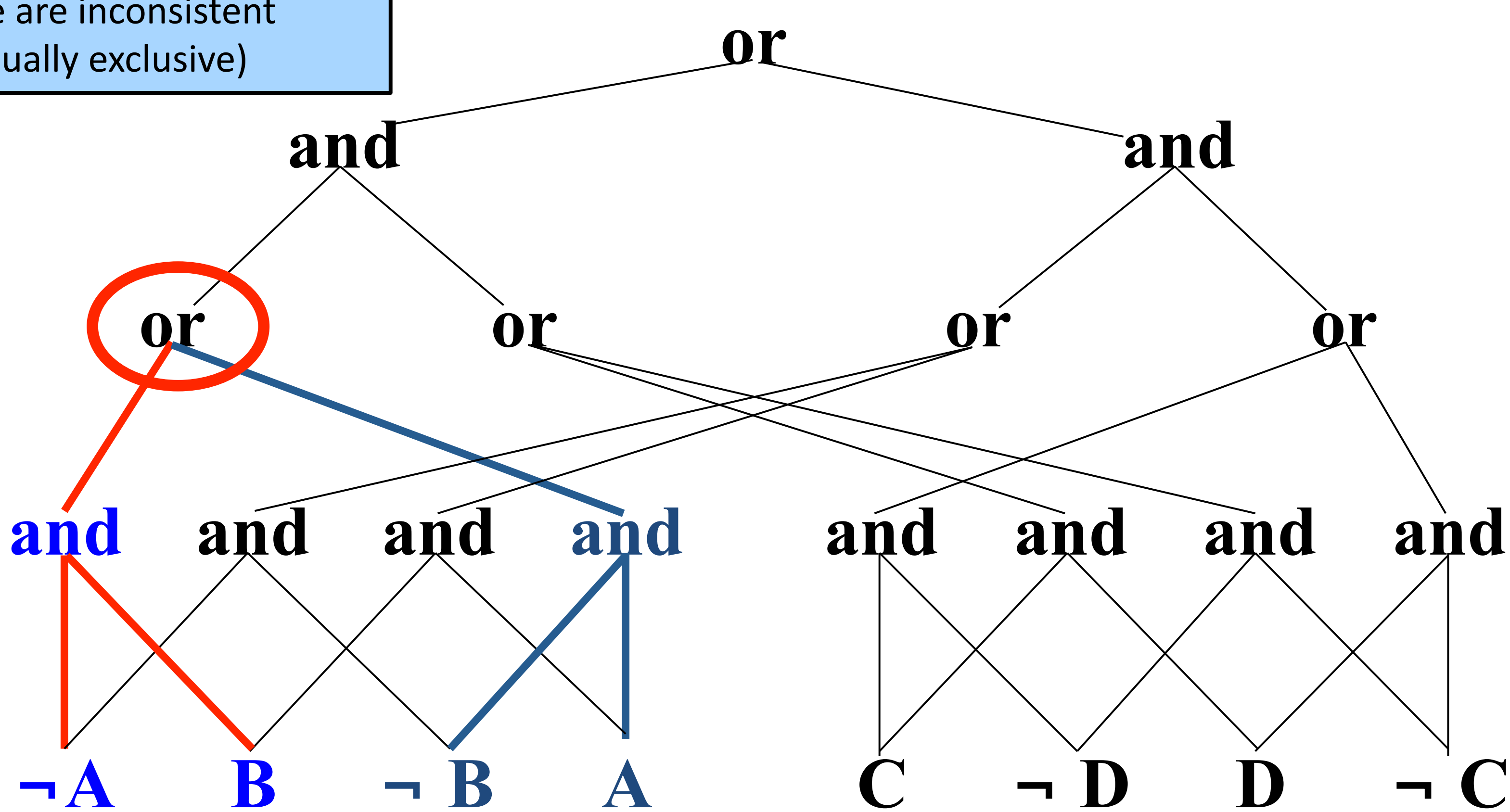
Decomposability (DNNF)

No two children of AND share a variable

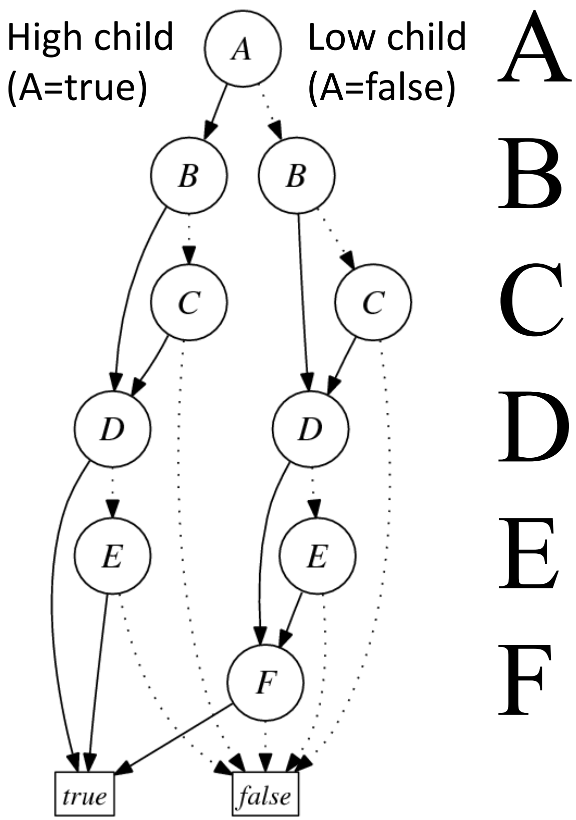


Determinism (d-DNNF)

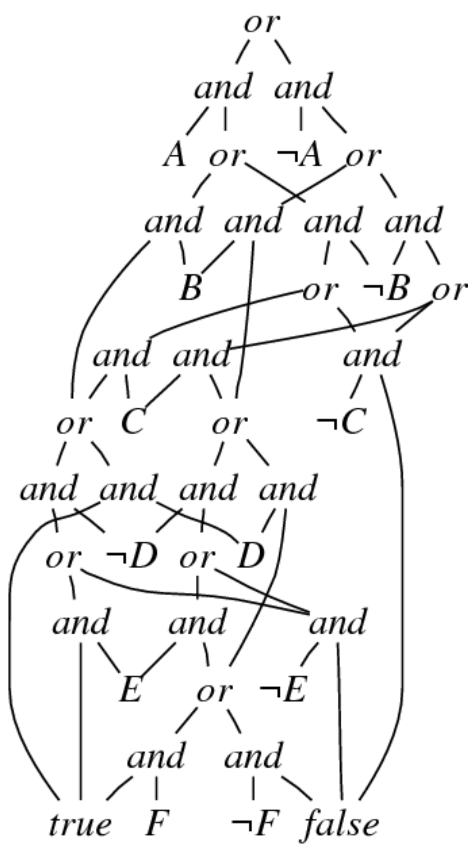
Every pair of children of or-node are inconsistent (mutually exclusive)



OBDD:
d-DNNF + Additional Properties



OBDD (traditional form)

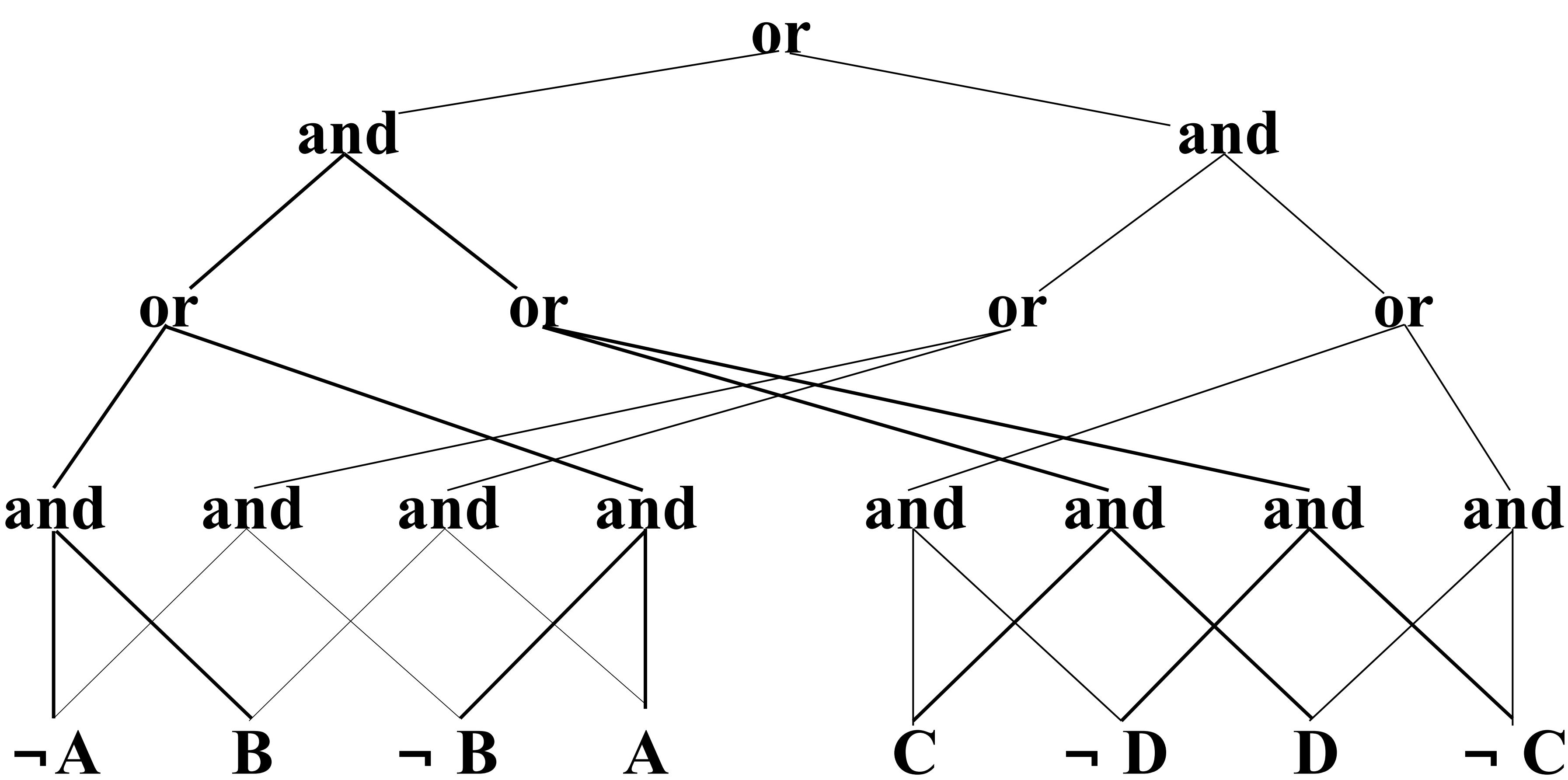


OBDD (NNF)

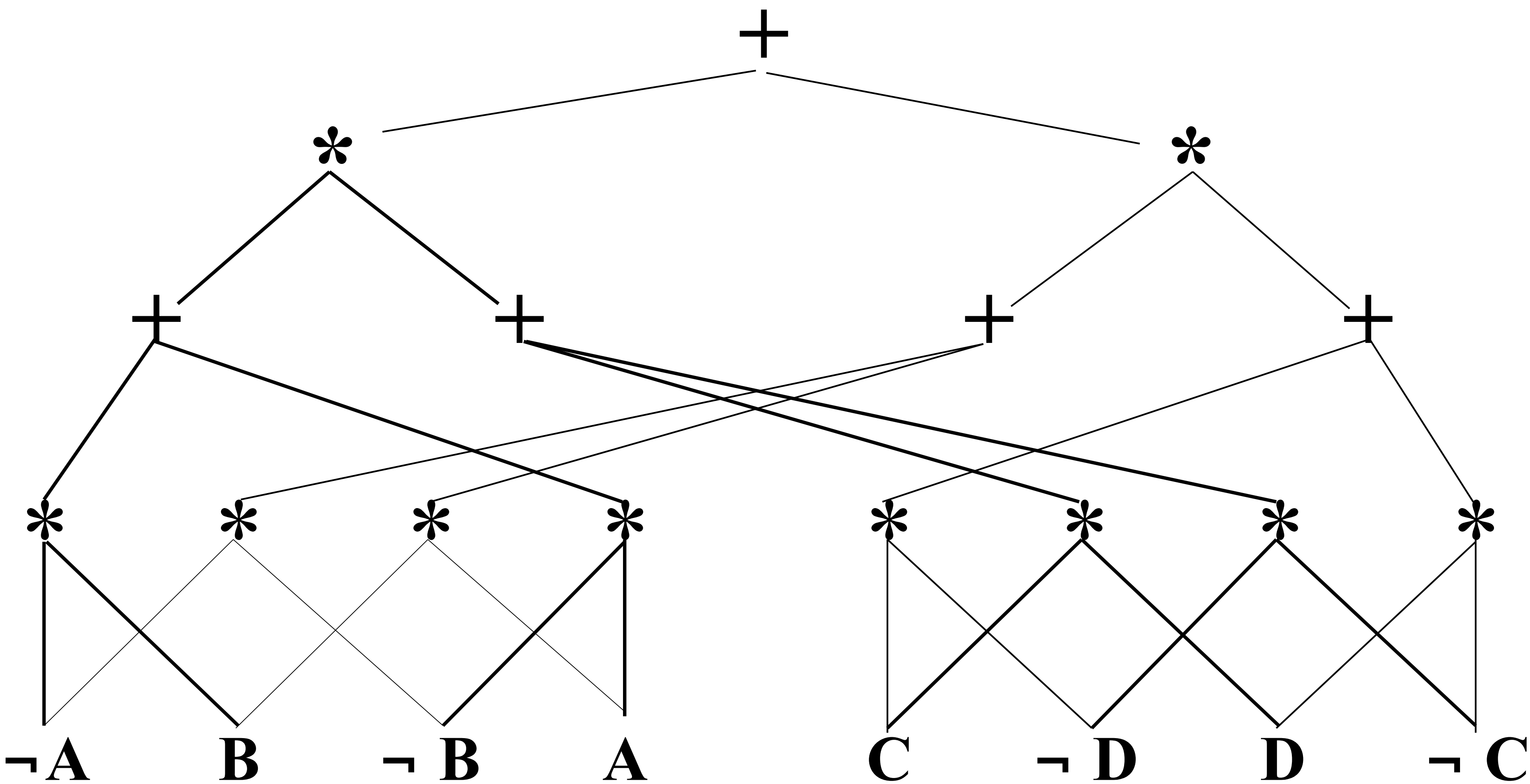
Queries and Transformations

- **Queries**
  SAT, MAXSAT, logical entailment, equivalence testing, model counting,…
- **Transformations:**
  Existential quantification, conjunction, disjunction, negation…
- More properties imply more polytime queries and transformations, but less succinctness

Counting Models (d-DNNF)

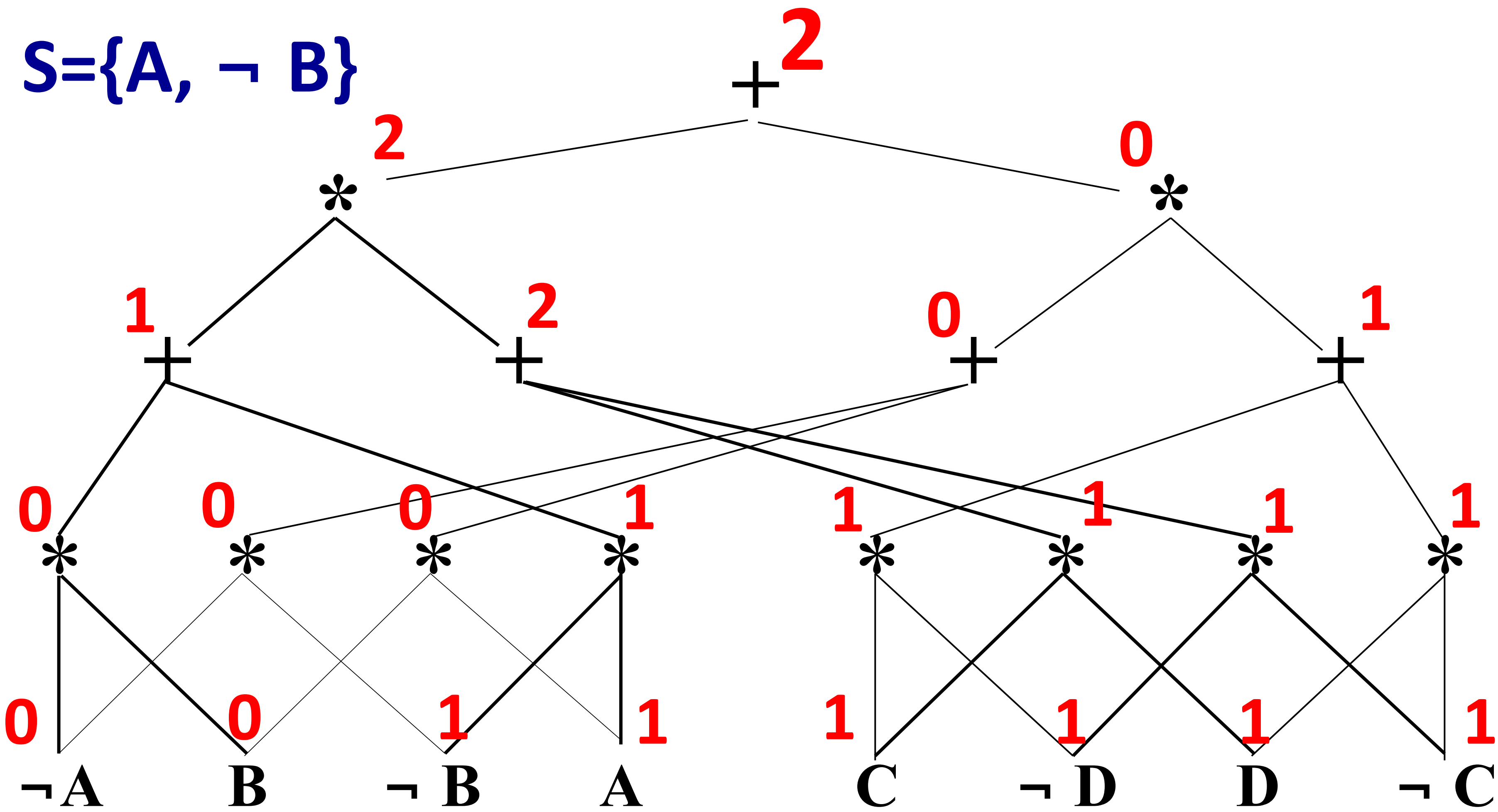


Counting Graph

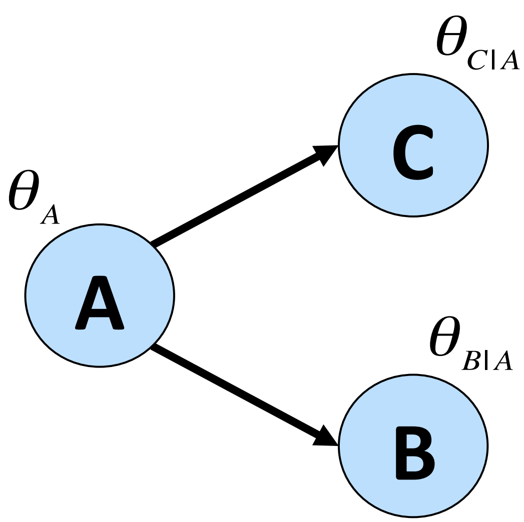


Counting Graph

S={A, ¬ B}



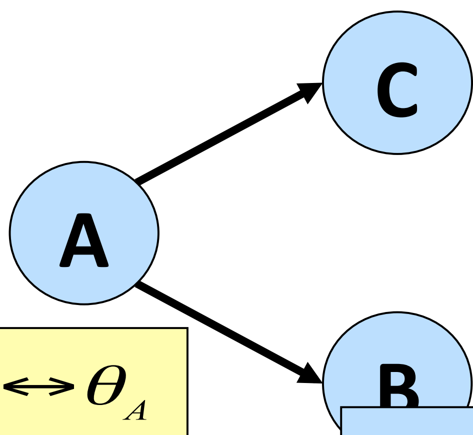
Probabilistic Inference by Weighted Model Counting



$\Pr(\neg B)$

A	B	C	Pr(.)
T	T	T	$\theta_A \theta_{B\|A} \theta_{C\|A}$
T	T	F	$\theta_A \theta_{B\|A} \theta_{\neg C\|A}$
T	F	T	$\theta_A \theta_{\neg B\|A} \theta_{C\|A}$
T	F	F	$\theta_A \theta_{\neg B\|A} \theta_{\neg C\|A}$
F	T	T	$\theta_{\neg A} \theta_{B\|\neg A} \theta_{C\|\neg A}$
F	T	F	$\theta_{\neg A} \theta_{B\|\neg A} \theta_{\neg C\|\neg A}$
F	F	T	$\theta_{\neg A} \theta_{\neg B\|\neg A} \theta_{C\|\neg A}$
F	F	F	$\theta_{\neg A} \theta_{\neg B\|\neg A} \theta_{\neg C\|\neg A}$

Probabilistic Inference by Weighted Model Counting



$A \leftrightarrow \theta_A$
$\neg A \leftrightarrow \theta_{\neg A}$

$A \wedge B \leftrightarrow \theta_{B|A}$

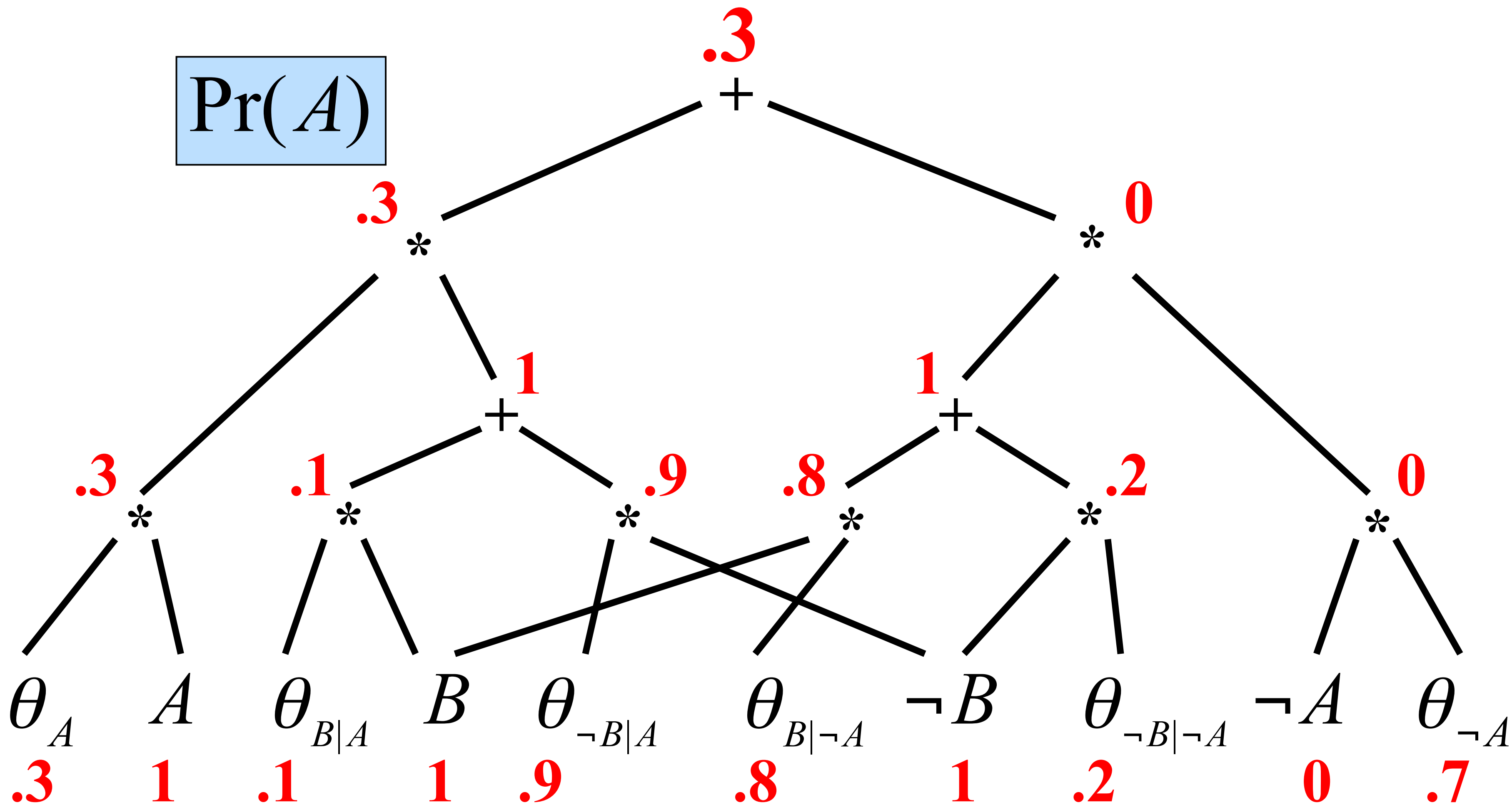
$A \wedge C \leftrightarrow \theta_{C|A}$
$A \wedge \neg C \leftrightarrow \theta_{\neg C|A}$
$\vdots$

A	B	C	Pr(.)
T	T	T	$\theta_A \theta_{B|A} \theta_{C|A}$
T	T	F	$\theta_A \theta_{B|A} \theta_{\neg C|A}$
T	F	T	$\theta_A \theta_{\neg B|A} \theta_{C|A}$
T	F	F	$\theta_A \theta_{\neg B|A} \theta_{\neg C|A}$

$$\Pr(\alpha) = wmc(\Delta \wedge \alpha)$$

$$A, B, \neg C \Rightarrow \underset{.3}{\theta_A}, \underset{.6}{\theta_{B|A}}, \underset{.8}{\theta_{\neg C|A}}, \underset{1}{\neg \theta_{\neg A}}, \ldots, \underset{1}{\neg \theta_{\neg C|\neg A}}$$

Weighted Model Counting (Arithmetic Circuits)



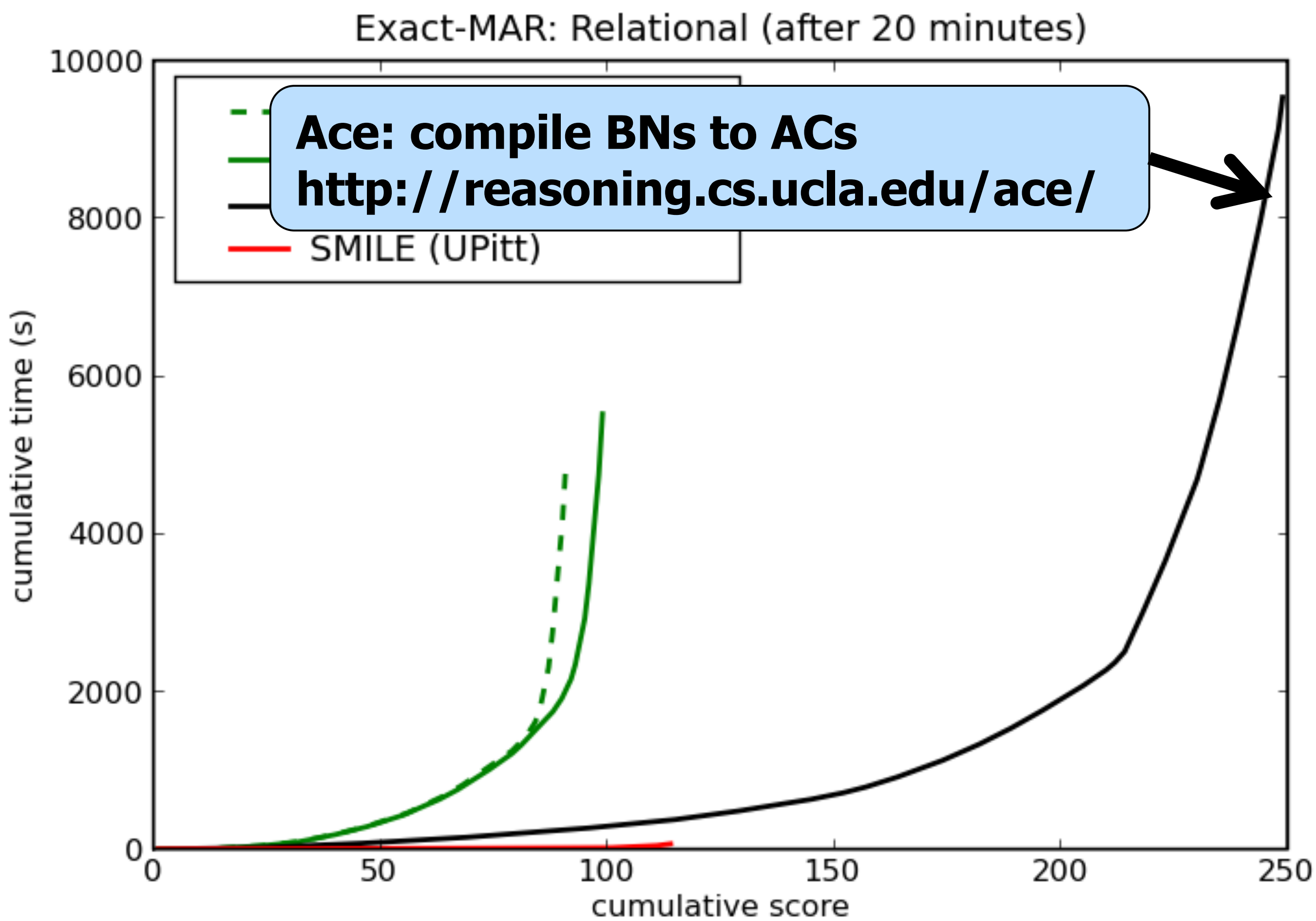
Why Logic?

- Encoding local structure is easy:
  - Zero-parameters encoded by adding clauses:

$$\boxed{\theta_{C|A}=0} \quad \boxed{\neg A \vee \neg C}$$

  - Context-specific independence encoded by collapsing variables:

$$\boxed{\theta_{C|AB}=\theta_{C|A\neg B}}$$



- Relational networks (251 networks)
  - Average clique size is 50

Alchemy - Open Source AI

Home

Documentation
Tutorial
User's Manual
Developer's Manual
APIs
Change Log
FAQs

Run
Requ
Dow

Mailing Lists
Alchemy
Alchemy-announce
Alchemy-update
Alchemy-discuss

Repositories
Code
Datasets
MLNs
Publications

Contributors

Welcome to the Alchemy system! Alchemy is a software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation. Alchemy allows you to easily develop a wide range of AI applications, including:

- Collective classification
- Link prediction
- Entity resolution
- Social network modeling
- Information extraction

If you are not already familiar with Markov logic, we recommend that you first read the paper Unifying Logical and Statistical AI.

Alchemy is a software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on Markov logic representations.

- Generative weight learning
- Structure learning
- MAP/MPE inference (including memory efficient)
- Probabilistic inference: MC-SAT, Gibbs Sampling, Simulated Tempering, Belief Propagation (including lifted)
- Support for native and linked-in functions
- Block inference and learning over variables with mutually exclusive and exhaustive values
- EM (to handle ground atoms with unknown truth values during learning)
- Specification of indivisible formulas (i.e. formulas that should not be broken up into separate clauses)
- Support of continuous features and domains
- Online inference
- Decision Theory

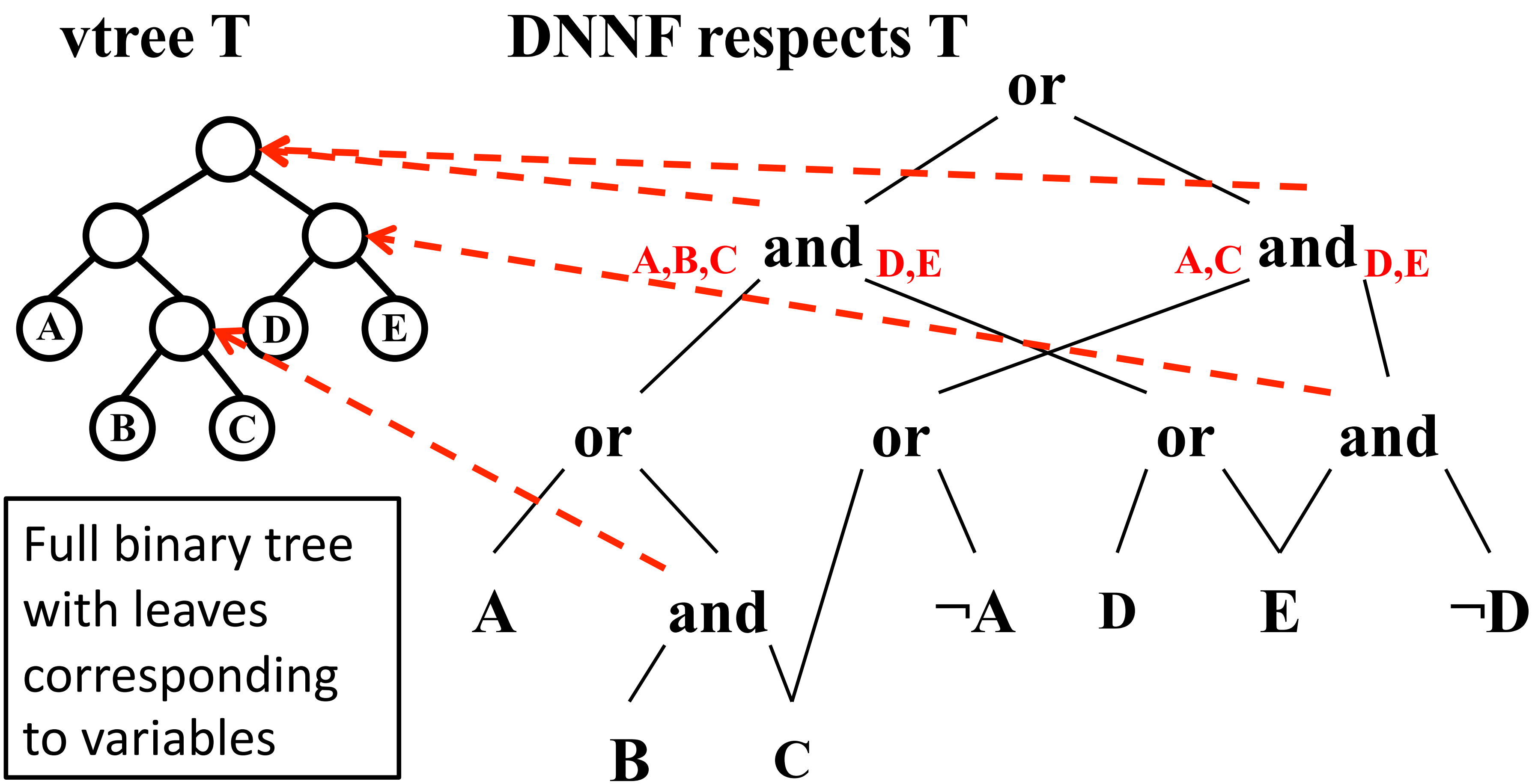
In the next release we plan to include:

- Online learning
- Exact inference for small domains

Current Challenges

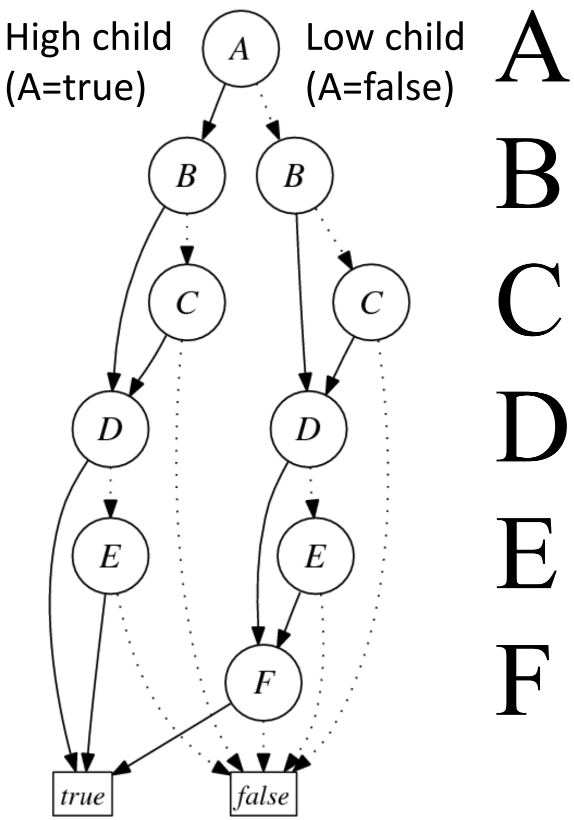
- **Incremental compilation:**
  - What? Current compilers monolithic: c2d (UCLA) and DSharp (Toronto)
  - Need:
    - Logic: planning and verification applications
    - Probability: approximate inference
  - Main insight:
    - Structured decomposability & vtrees (AAAI-08, AAAI-10)
- **Guarantees and Complexity results:**
  - Upper & lower bounds on size of compilation (AAAI-10, ECAI-10)
  - Main insights:
    - The notion of a decomposition (AAAI-10)
    - The notion of an interaction function (ECAI-10)

Structured Decomposability

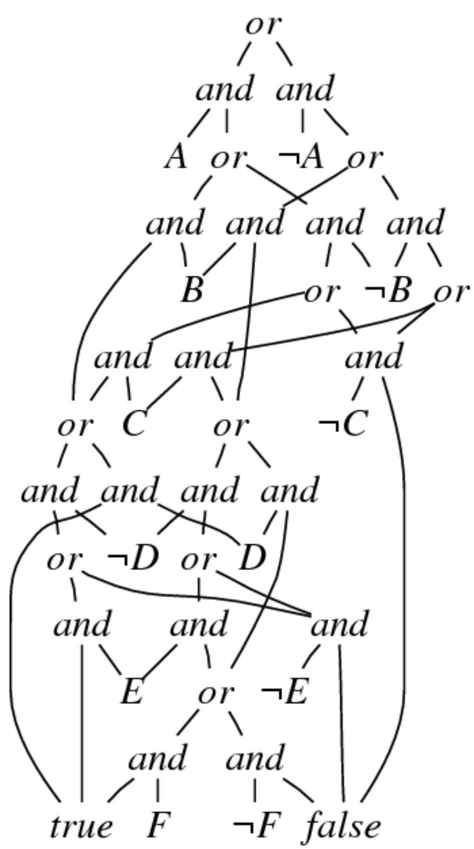


Full binary tree with leaves corresponding to variables

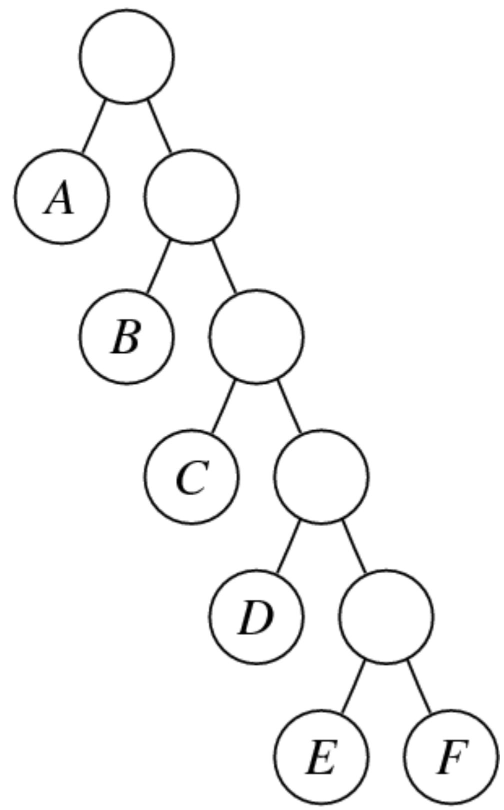
OBDD: DNNF that Respects Linear vtree



OBDD (traditional form)



OBDD (NNF)



Linear vtree

Decomposition of Boolean Functions (AAAI-10)

- Examples: $f = (X_1 \vee X_2) \wedge (Y_1 \vee X_2) \wedge (X_1 \vee Y_2) \wedge (Y_1 \vee Y_2) \vee (X_2 \wedge Y_3)$
  - $\mathbf{X} = \{X_1, X_2\}$, $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$:

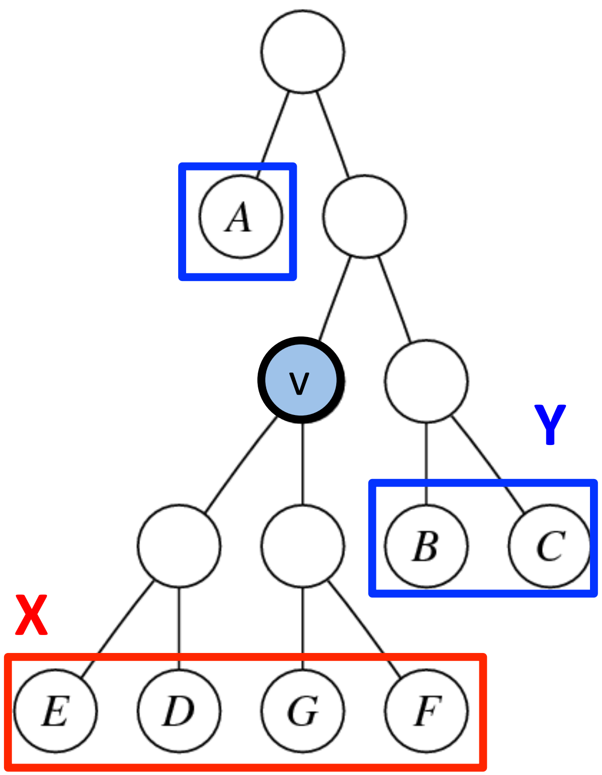
$$\cancel{f(\mathbf{X},\mathbf{Y}) = g(\mathbf{X}) \wedge h(\mathbf{Y})}$$

$$f(\mathbf{X},\mathbf{Y}) = \underbrace{f_1}_{g_1(\mathbf{X}) \wedge h_1(\mathbf{Y})} \vee \underbrace{f_2}_{g_2(\mathbf{X}) \wedge h_2(\mathbf{Y})} \vee \underbrace{f_3}_{g_3(\mathbf{X}) \wedge h_3(\mathbf{Y})} \vee \dots \vee \underbrace{f_m}_{g_m(\mathbf{X}) \wedge h_m(\mathbf{Y})}$$

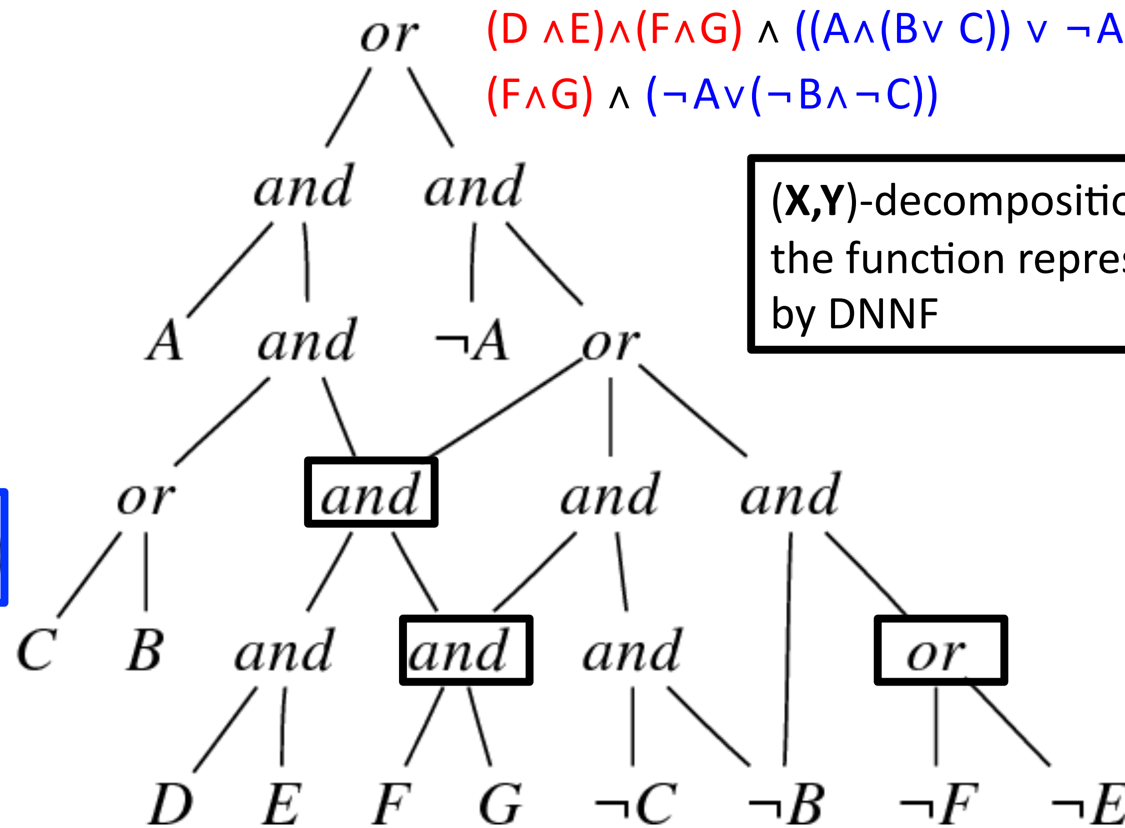
$$\boxed{(\mathbf{X},\mathbf{Y})\text{-decomposition of } f}$$

Lower Bounds (AAAI-10)

Vtree



DNNF



$(\neg E \vee \neg F) \wedge (\neg A \wedge \neg B)$ **∨**

$(D \wedge E) \wedge (F \wedge G) \wedge ((A \wedge (B \vee C)) \vee \neg A)$ **∨**

$(F \wedge G) \wedge (\neg A \vee (\neg B \wedge \neg C))$

(**X,Y**)-decomposition of the function represented by DNNF

The Interaction Function (ECAI-10)

$$f(\mathbf{X},\mathbf{Y}) = g(\mathbf{X}) \wedge h(\mathbf{Y}) \wedge I(\mathbf{X},\mathbf{Y})$$

Captures precisely knowledge about variables in **X**

$\exists \mathbf{Y}\, f$

Captures precisely knowledge about variables in **Y**

$\exists \mathbf{X}\, f$

Captures precisely interaction between variables **X** and **Y**

$\boxed{f \vee \neg(\exists \mathbf{X}\, f) \vee \neg(\exists \mathbf{Y}\, f)}$

The Interaction Function (ECAI-10)

$$f(\mathbf{X},\mathbf{Y}) = \quad g(\mathbf{X}) \wedge h(\mathbf{Y}) \wedge I(\mathbf{X},\mathbf{Y})$$

Captures precisely knowledge about variables in **X**

Captures precisely knowledge about variables in **Y**

Captures precisely interaction between variables **X** and **Y**

$\exists \mathbf{Y}\ f$

true

$\exists \mathbf{X}\ f$

(B v C)

A => (C => B)
¬A => (B => C)

(A => B) (¬A => C) **X**={A} **Y**={B,C}

Current Research

- Searching for good vtrees (on-going)
- Characterizing and searching for optimal decompositions
- Upper and lower bounds on size of DNNF

- Key objective: incremental compiler for DNNF and d-DNNF

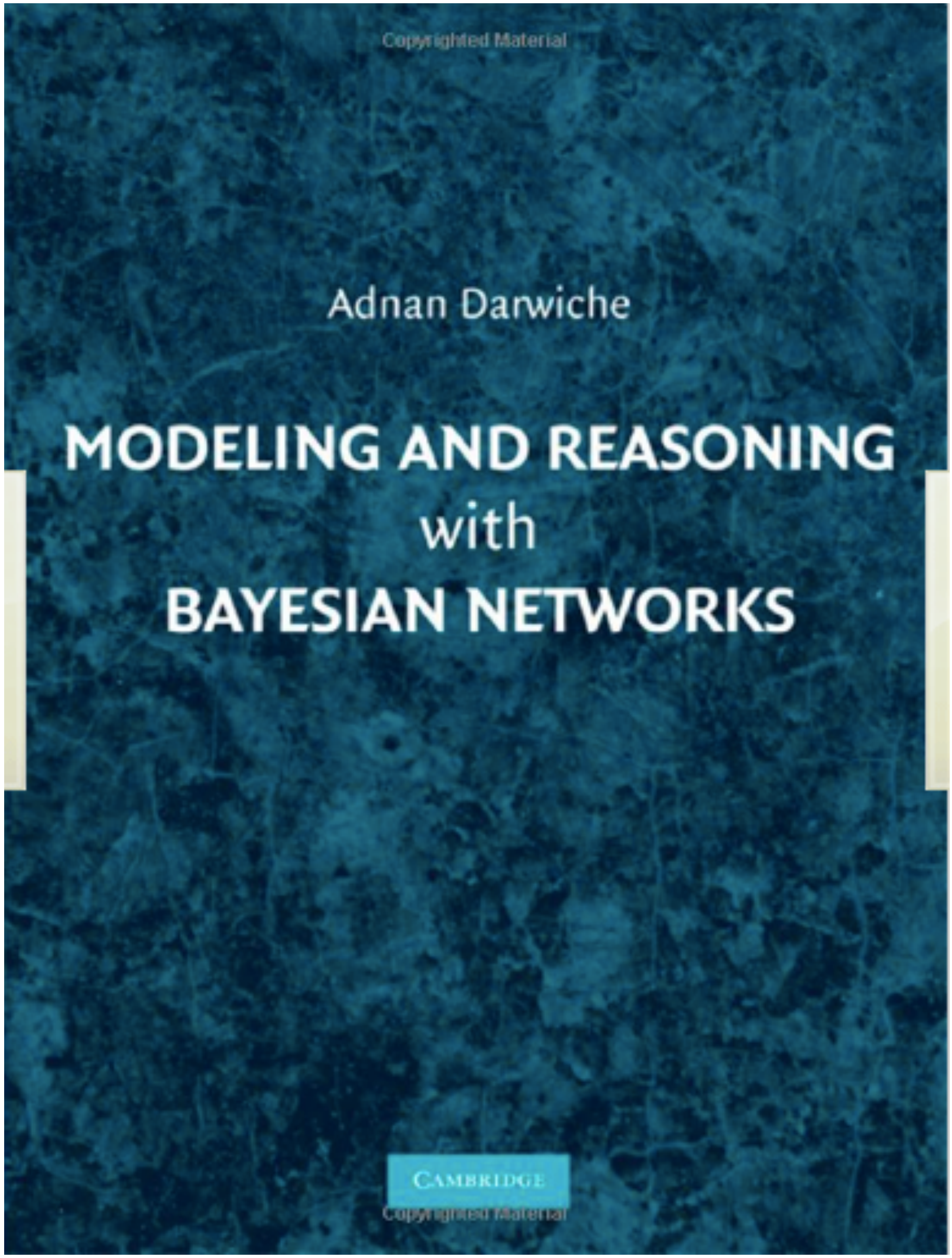
- ???

Two Main Themes

- **Exact inference as:**
  Enforcing decomposability and determinism on propositional knowledge bases

- **Approximate inference as:**
  Relaxing, compensating for, and recovering equivalence constraints (equalities)

http://reasoning.cs.ucla.edu/samiam/


Copyrighted Material
Adnan Darwiche
MODELING AND REASONING
with
BAYESIAN NETWORKS
CAMBRIDGE
Copyrighted Material

http://reasoning.cs.ucla.edu

Two Main Themes

- **Exact inference as:**
  Enforcing decomposability and determinism on propositional knowledge bases

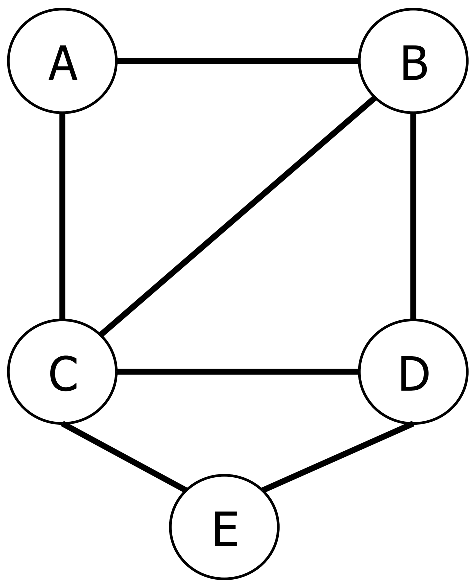
- **Approximate inference as:**
  Relaxing, compensating for, and recovering equivalence constraints (equalities)

Treewidth

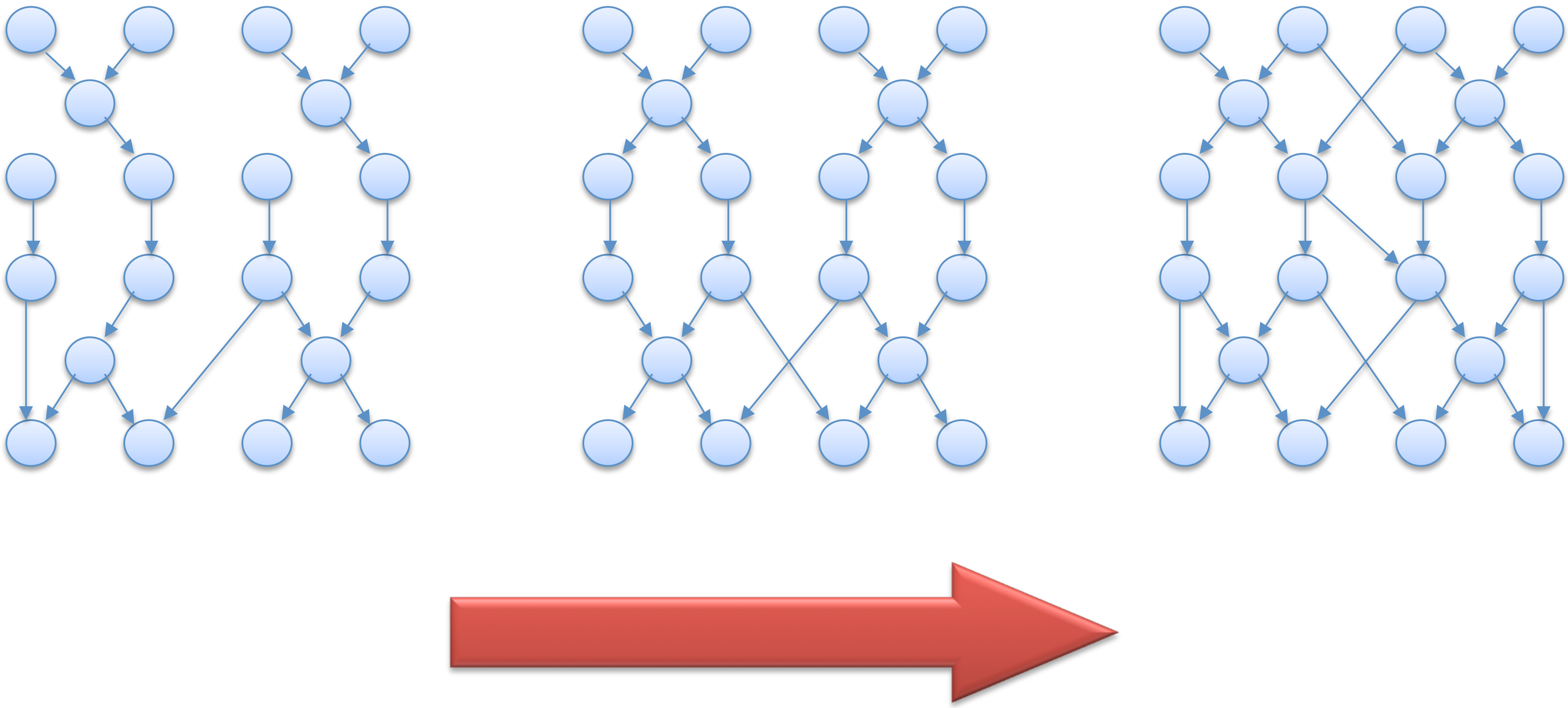
CNF

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

Constraint Graph



Treewidth



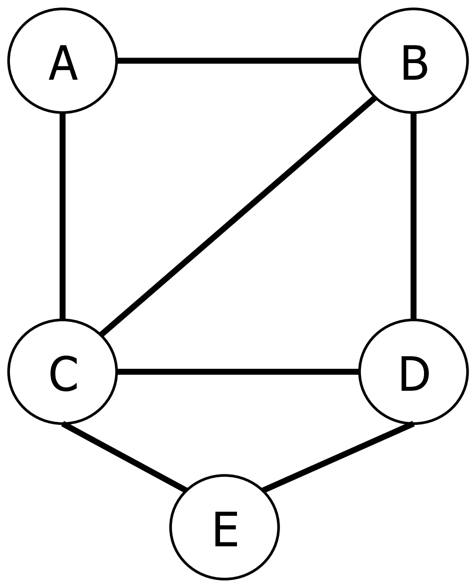
$$O(n \exp\{w\})$$

Treewidth

CNF

(A v B v ¬ C) ∧
(A v ¬ B v C) ∧
(C v ¬ D v E) ∧
(B v D) ∧
(D v E)

Constraint Graph

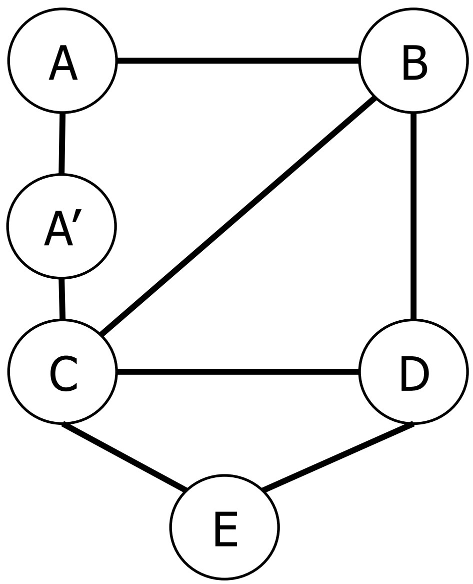


Treewidth

CNF

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

Constraint Graph

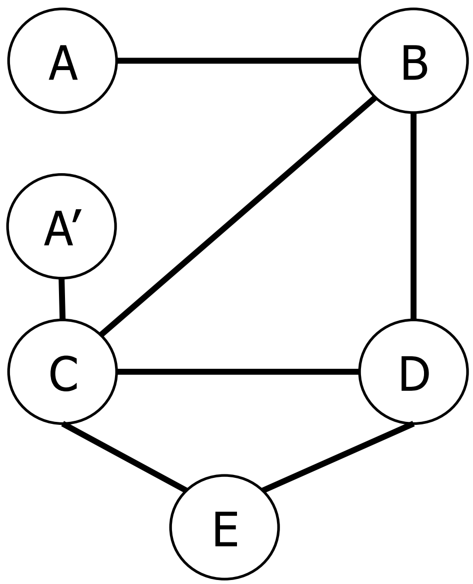


Treewidth

CNF

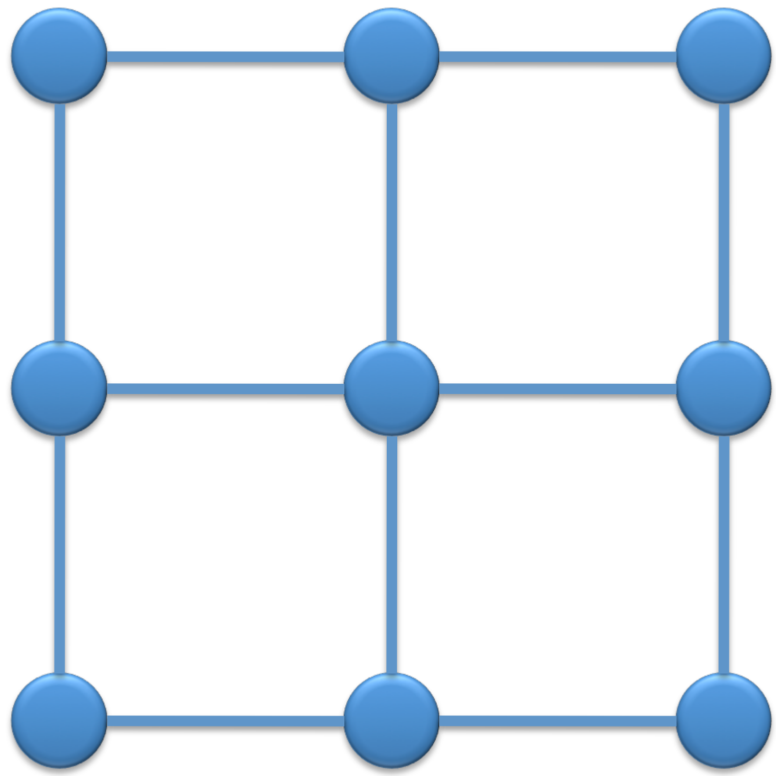
$(\mathbf{A} \vee B \vee \neg C) \wedge$
$(\mathbf{A'} \vee \neg B \vee C) \wedge$
$(C \vee \neg D \vee E) \wedge$
$(B \vee D) \wedge$
$(D \vee E)$

Constraint Graph



Equivalence Constraints

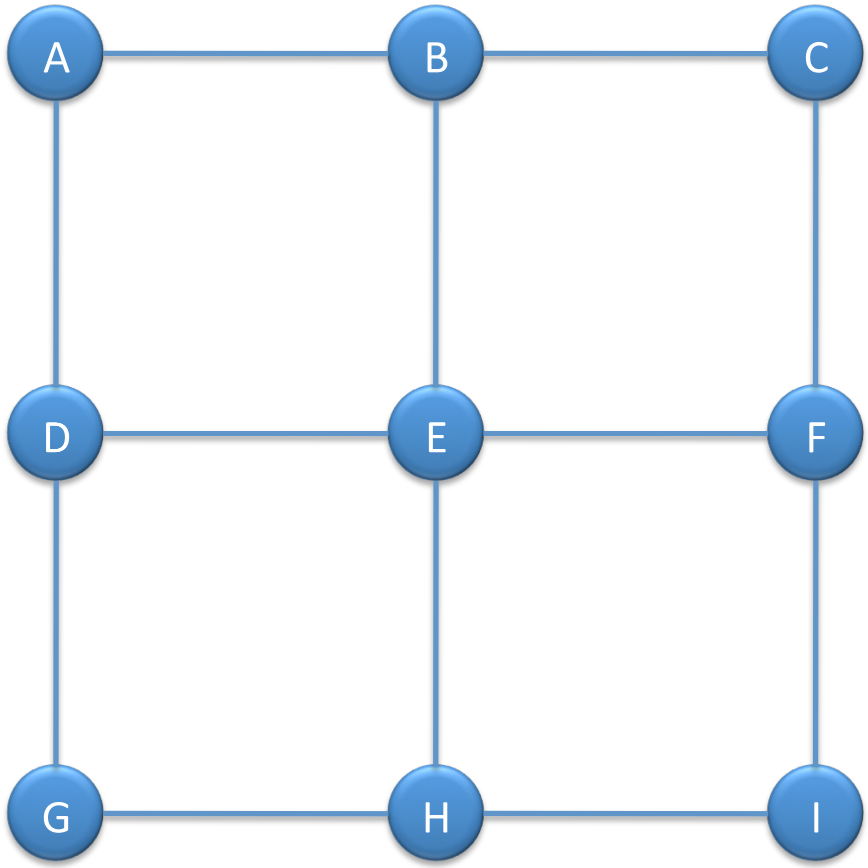
$$\psi_{eq}(X_i{=}x_i, X_j{=}x_j) = \begin{cases} 1 & \text{if } x_i = x_j \\ 0 & \text{otherwise} \end{cases}$$



$$\Pr(x_1, \ldots, x_n) = \frac{1}{Z} \psi(x_1, x_5) \ldots \psi(x_2, x_4) \ldots \theta(x_1) \ldots \theta(x_n)$$

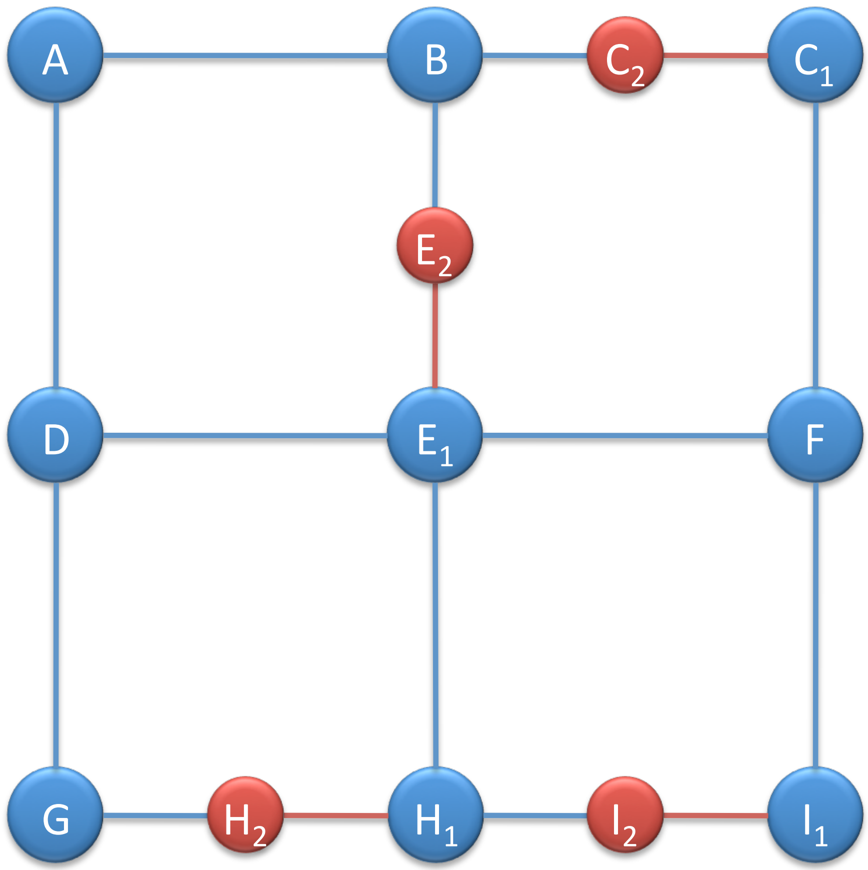
Relaxing Equivalence Constraints

- Model $\mathcal{M}$



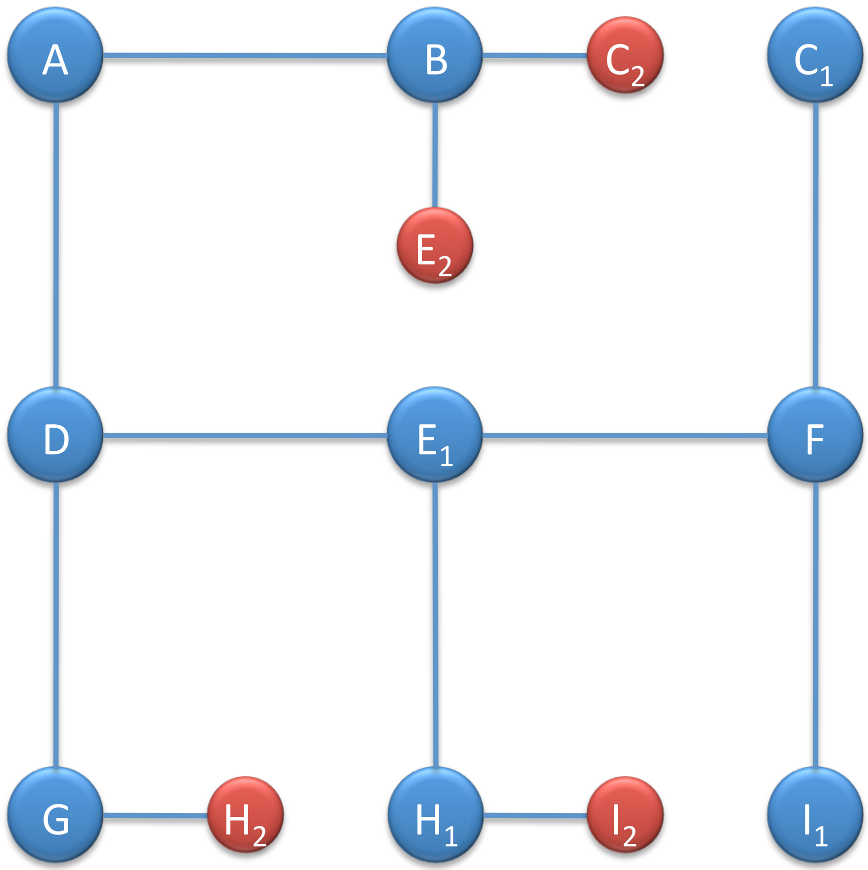
Relaxing Equivalence Constraints

- Model + Eq.



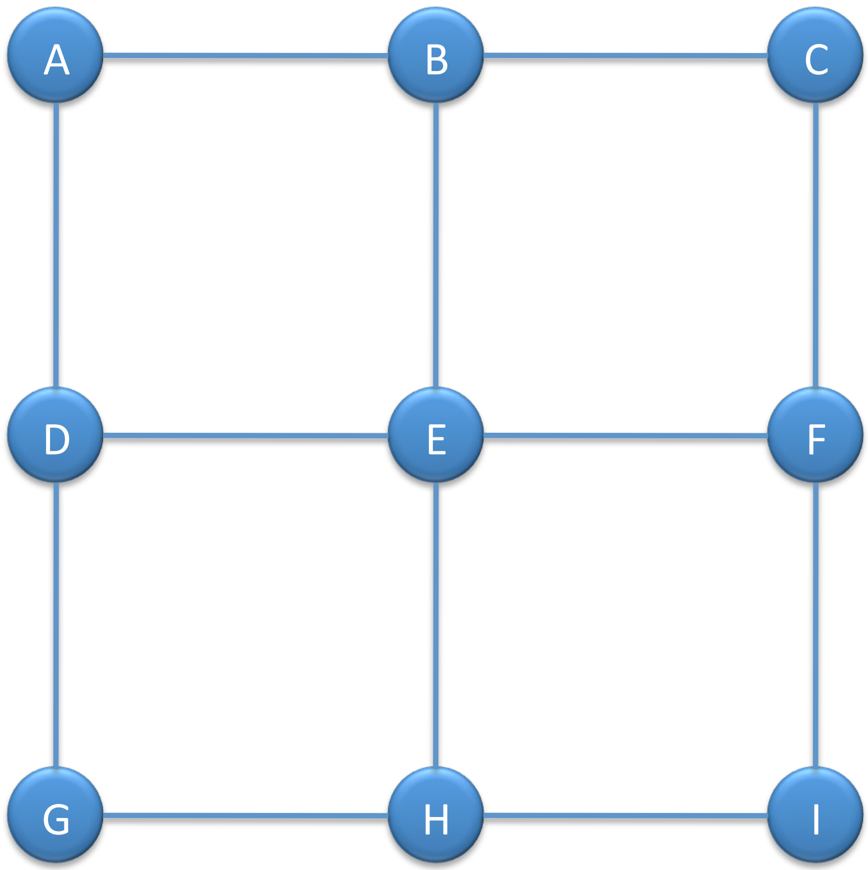
Relaxing Equivalence Constraints

- Relaxed
- Treewidth 1



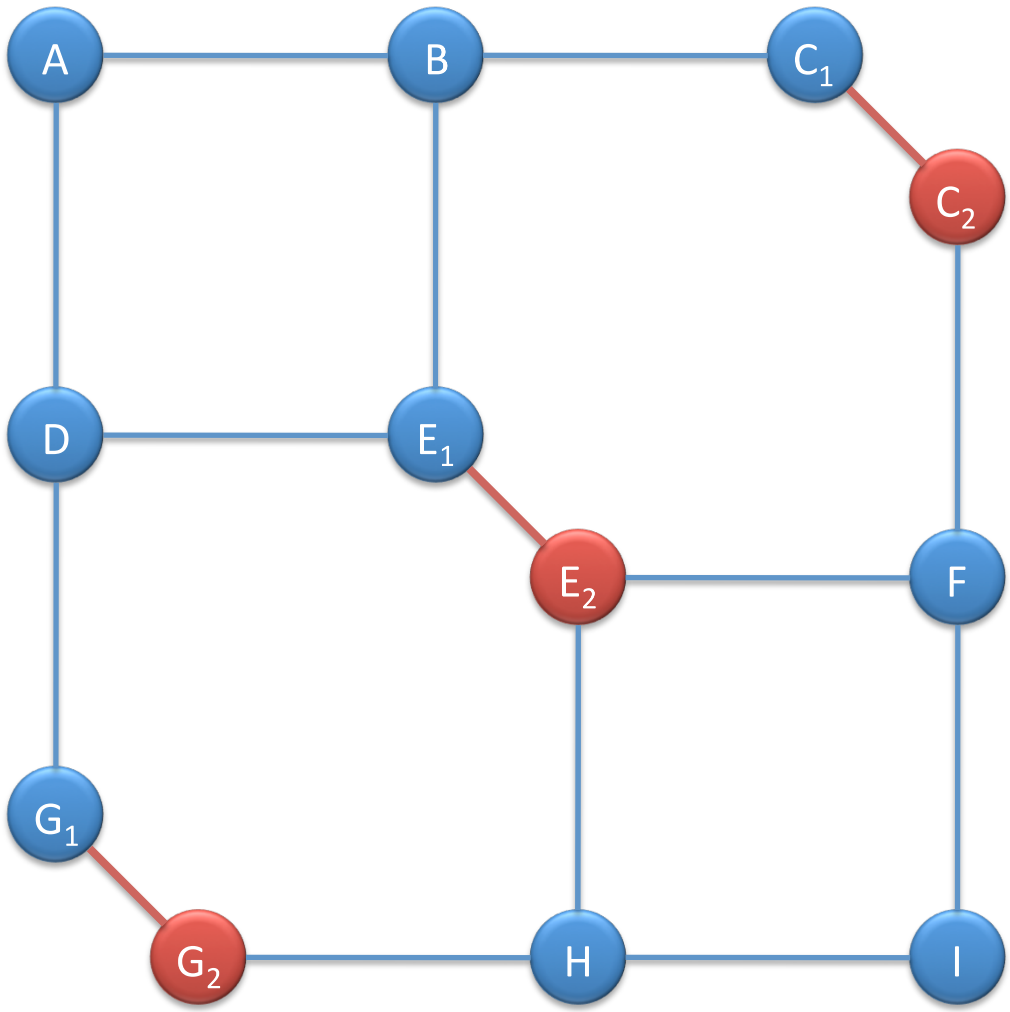
Relaxing Equivalence Constraints

- Model $\mathcal{M}$



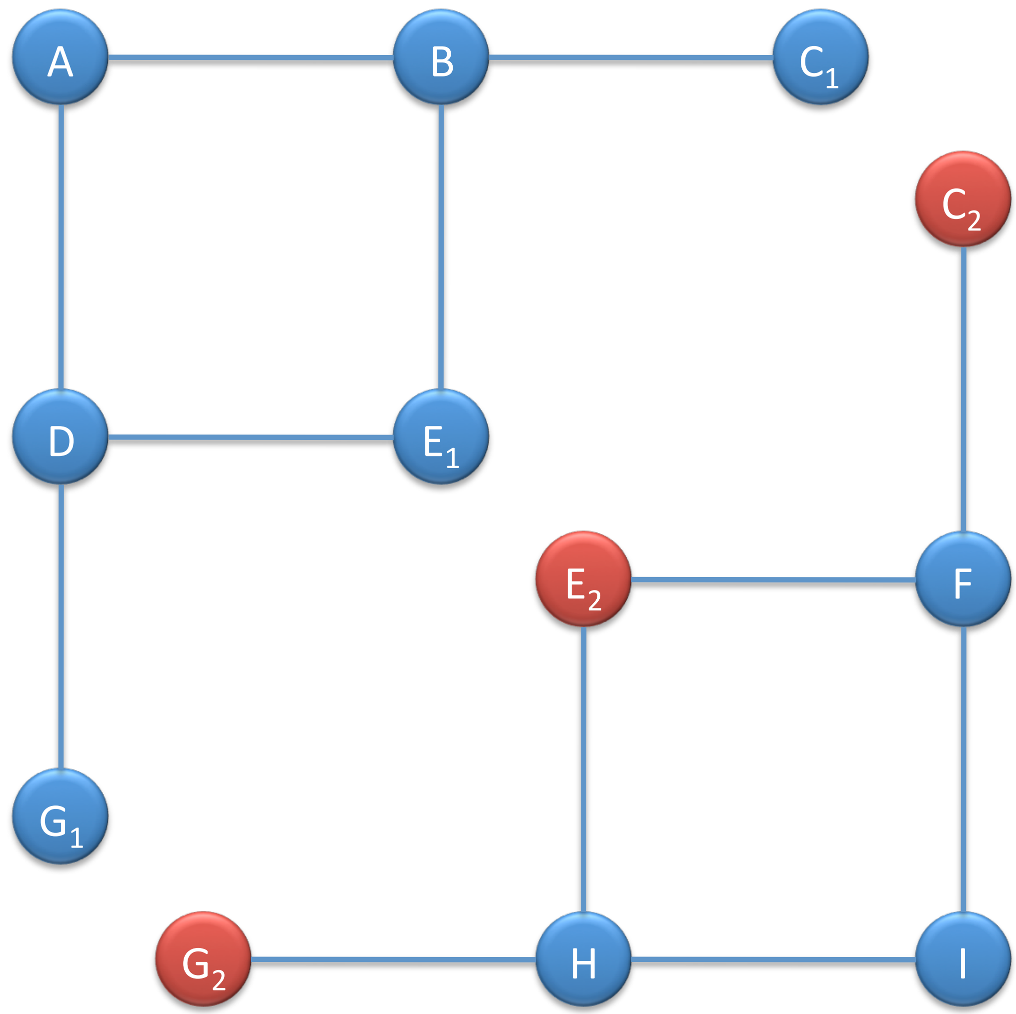
Relaxing Equivalence Constraints

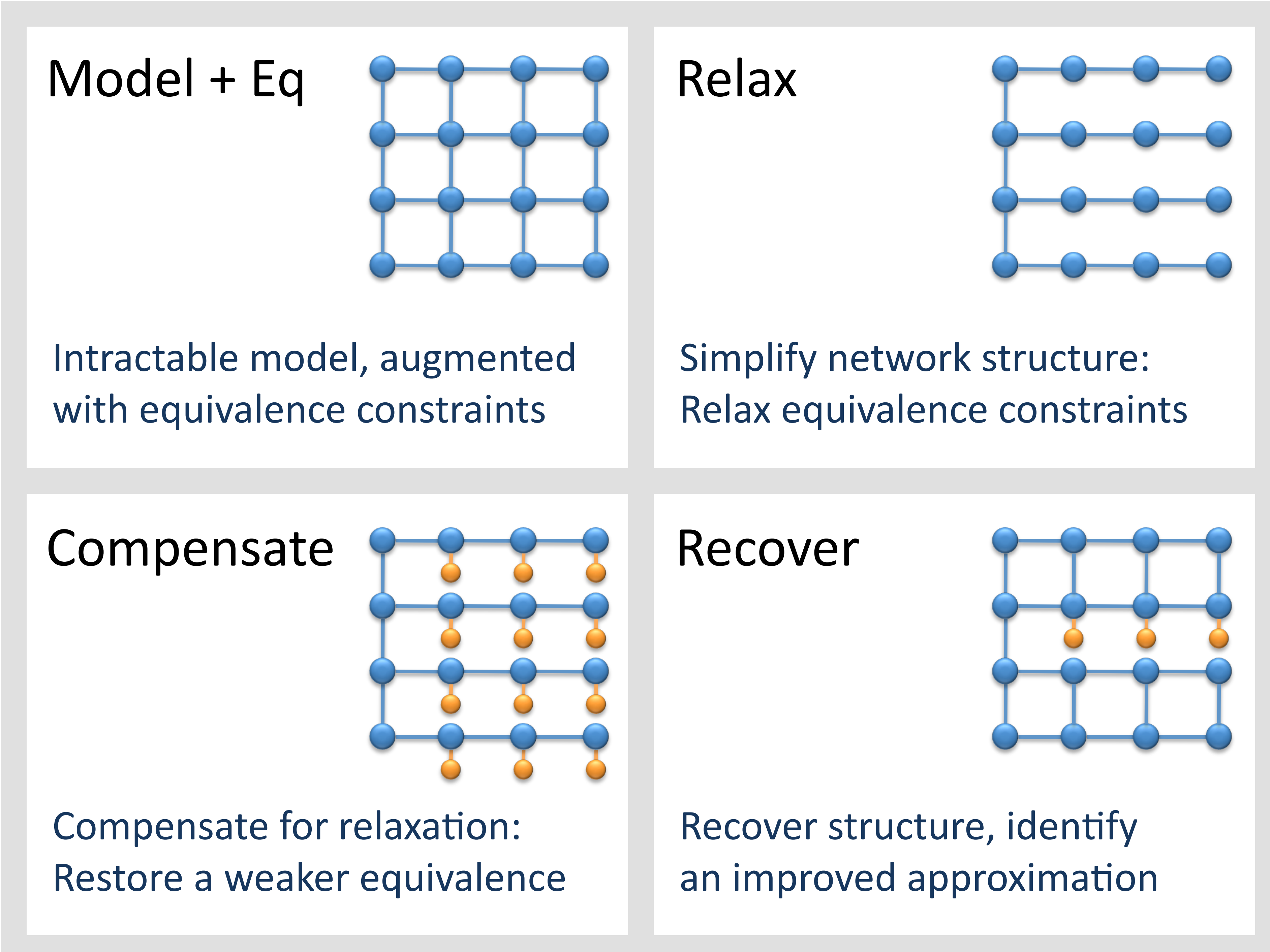
- Model + Eq.



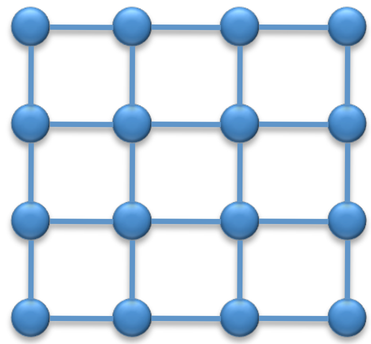
Relaxing Equivalence Constraints

- Relaxed
- Decomposed



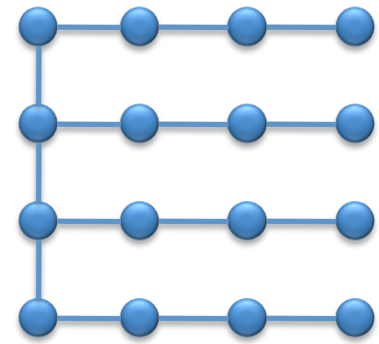

Model + Eq
Intractable model, augmented
with equivalence constraints
Relax
Simplify network structure:
Relax equivalence constraints
Compensate
Compensate for relaxation:
Restore a weaker equivalence
Recover
Recover structure, identify
an improved approximation

Model + Eq



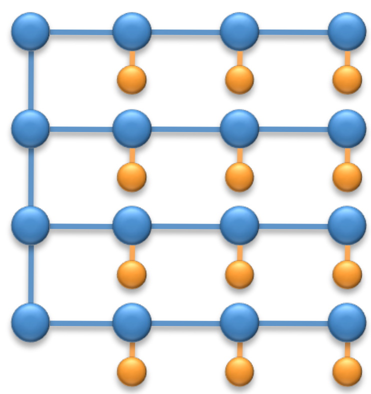
Intractable model, augmented with equivalence constraints

Relax



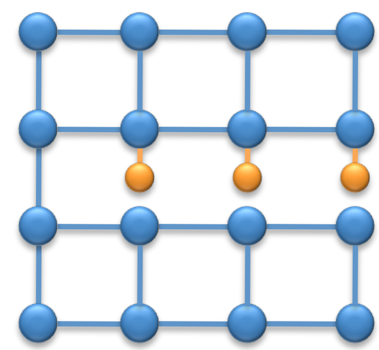
Usually gives upper/lower bounds: mini-buckets, MAXSAT

Compensate



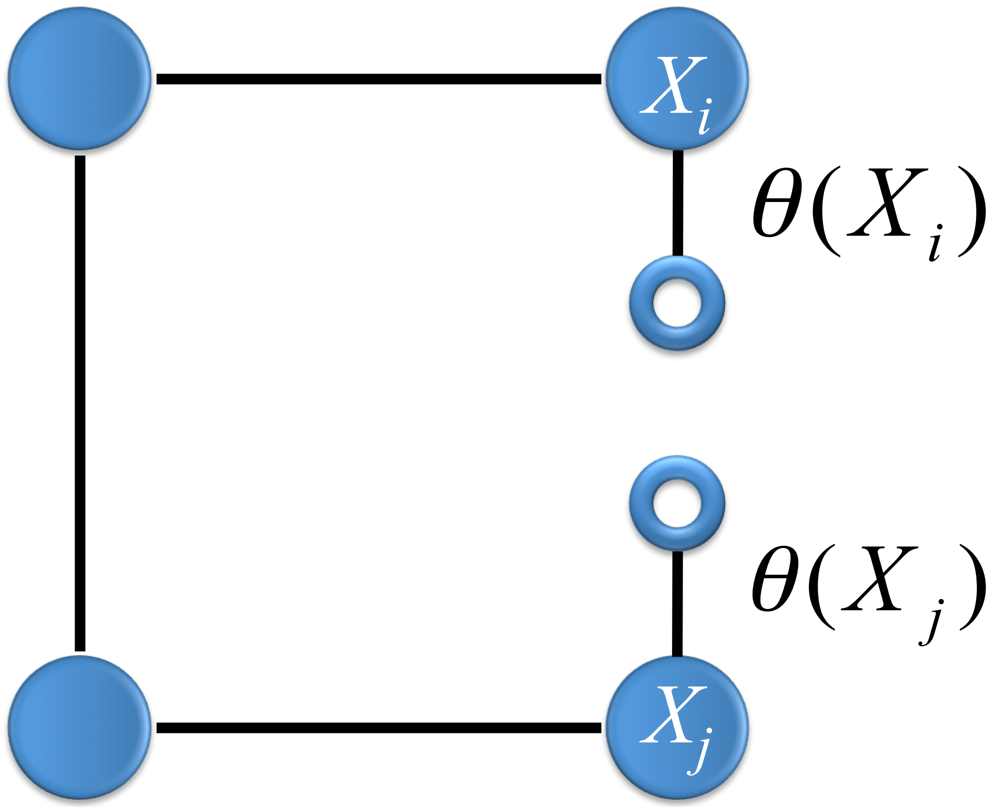
Compensate for relaxation: Restore a weaker equivalence

Recover



Recover structure, identify an improved approximation

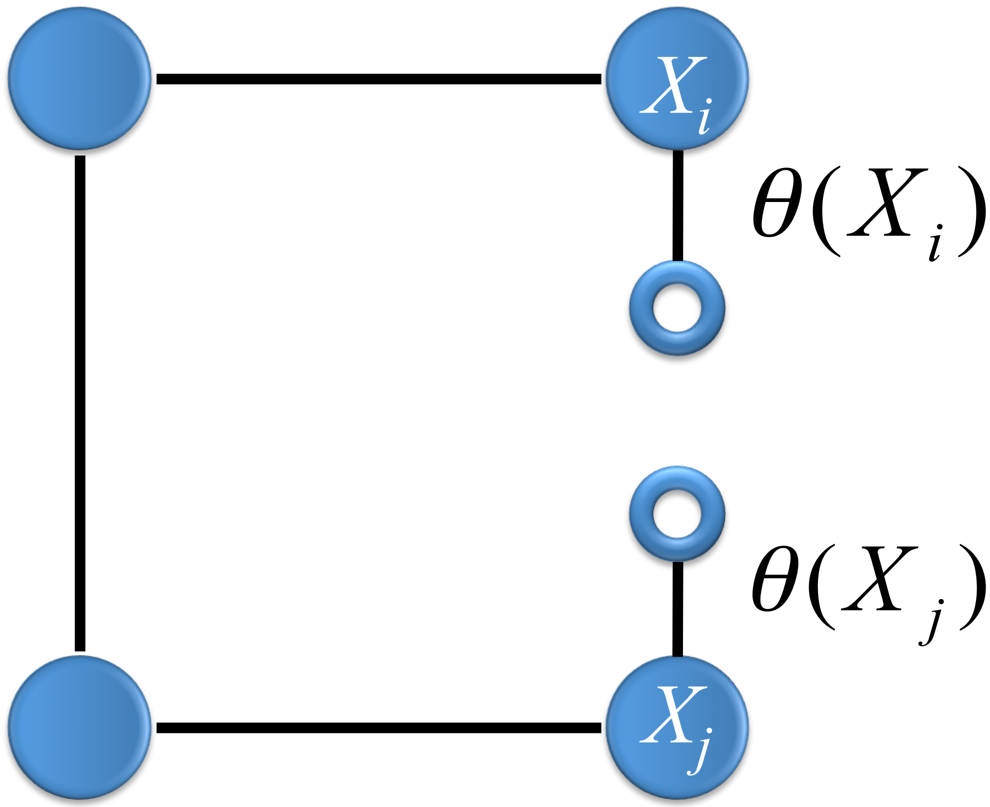
Compensating for an Equivalence



X_i	$\theta(X_i)$
T	.4789
F	.5211

X_j	$\theta(X_j)$
T	.8273
F	.1727

Compensating for an Equivalence

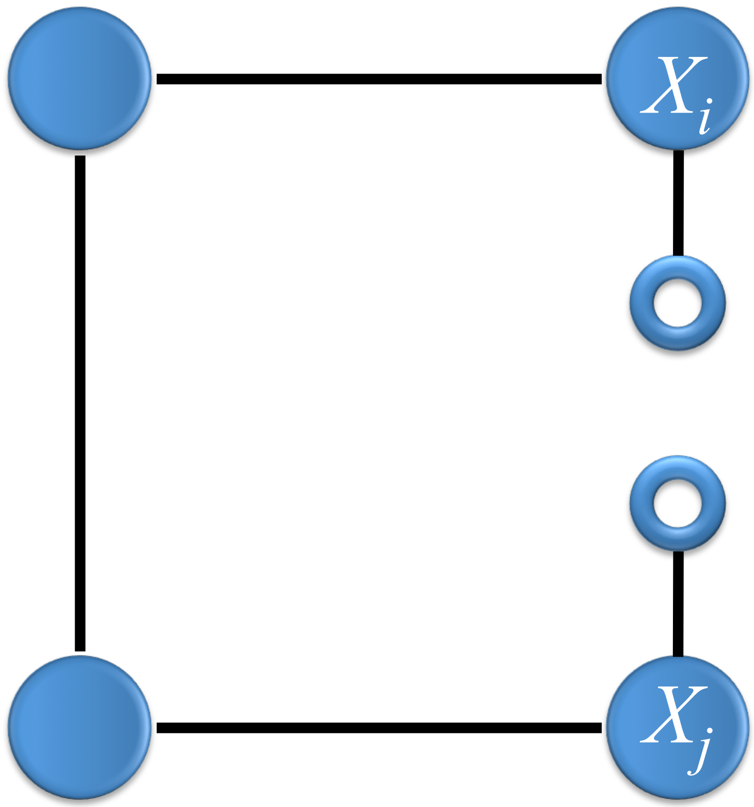


X_i	$\theta(X_i)$
T	.4789
F	.5211

X_j	$\theta(X_j)$
T	.8273
F	.1727

$$\Pr(X_i = x) = \Pr(X_j = x)$$

Compensating for an Equivalence

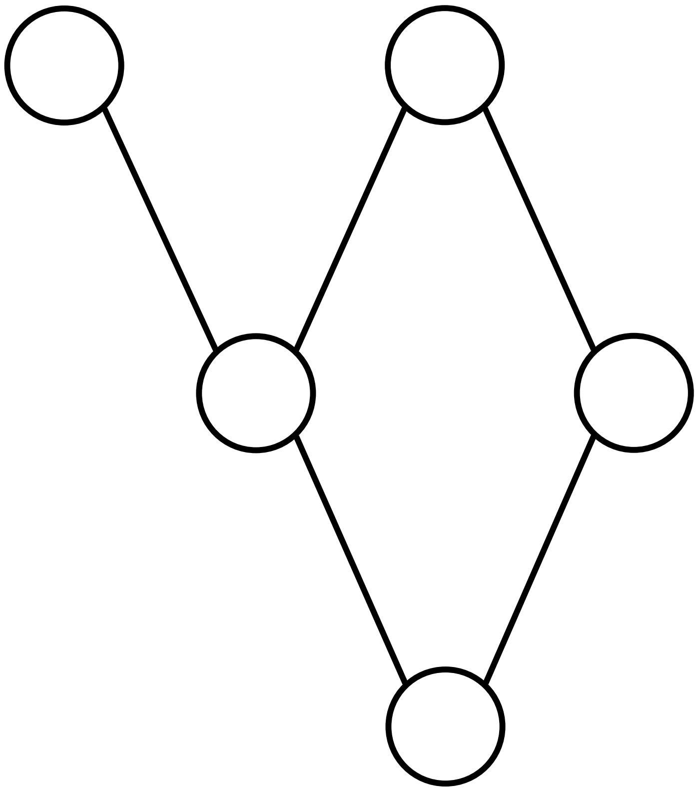


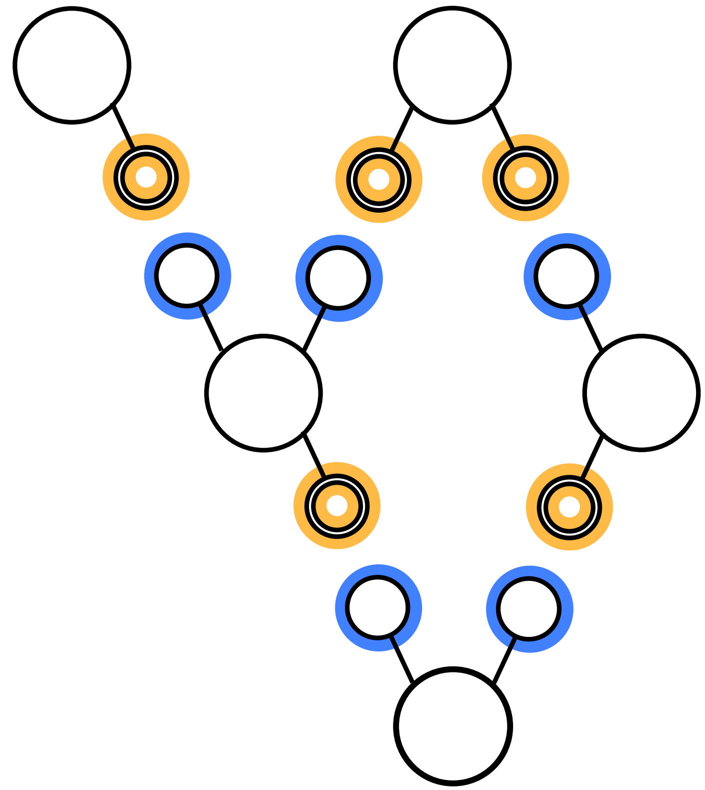
$$\theta(X_i) = \alpha \frac{\partial Z}{\partial \theta(X_j)}$$

$$\theta(X_j) = \alpha \frac{\partial Z}{\partial \theta(X_i)}$$

$$\Pr(X_i = x) = \Pr(X_j = x)$$

Parametrizing Edges Iteratively

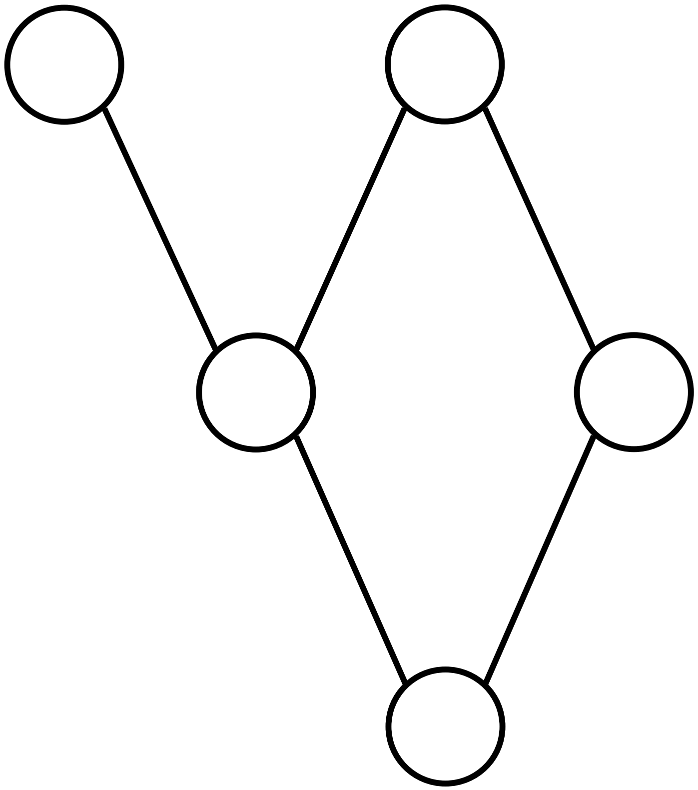


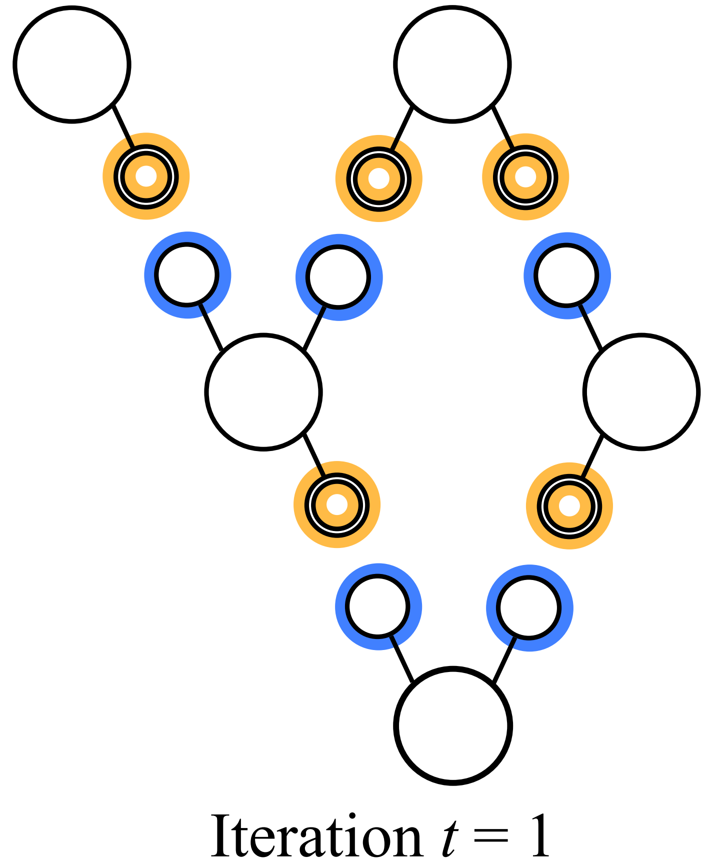


Iteration $t = 0$

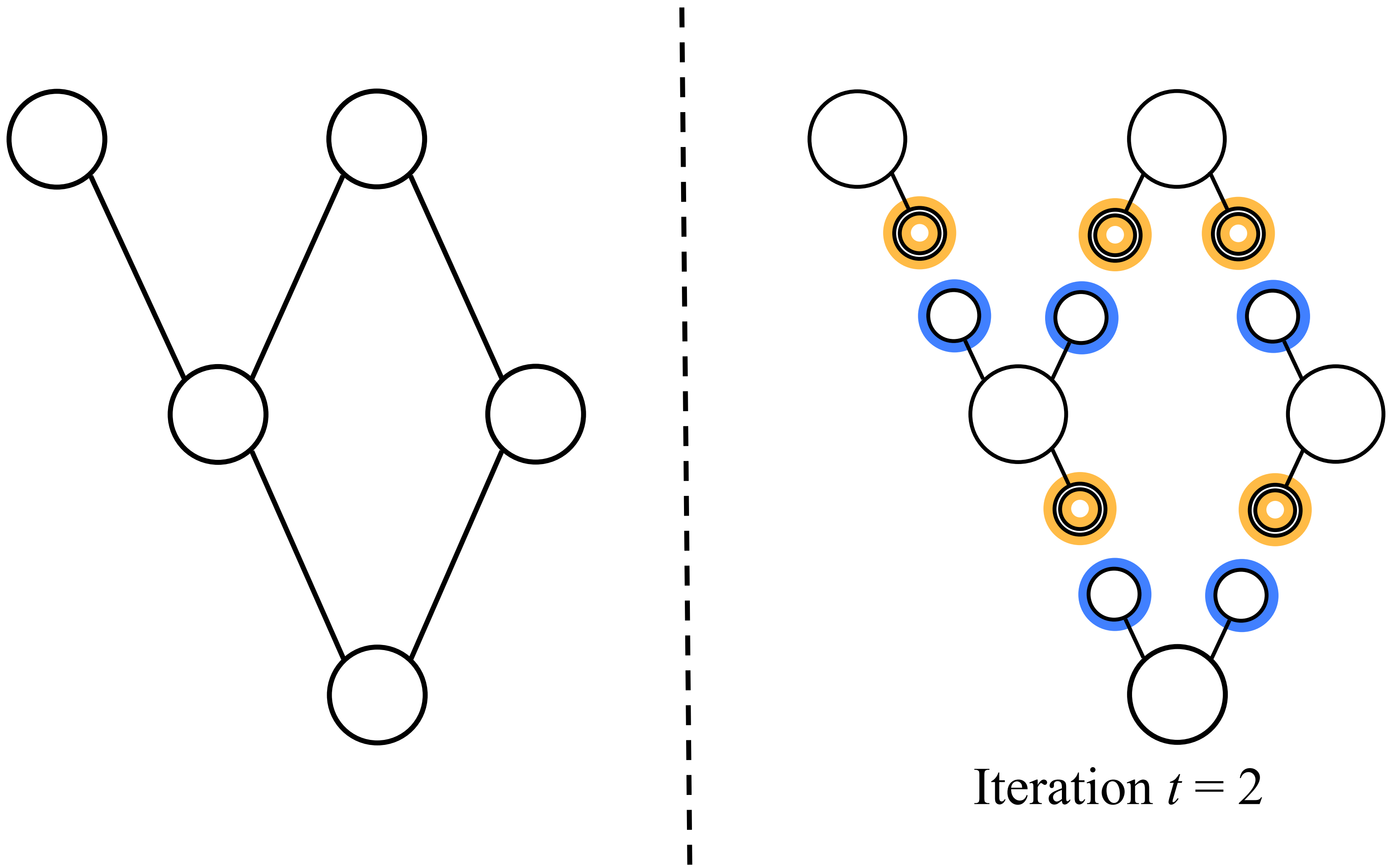
Initialization

Parametrizing Edges Iteratively



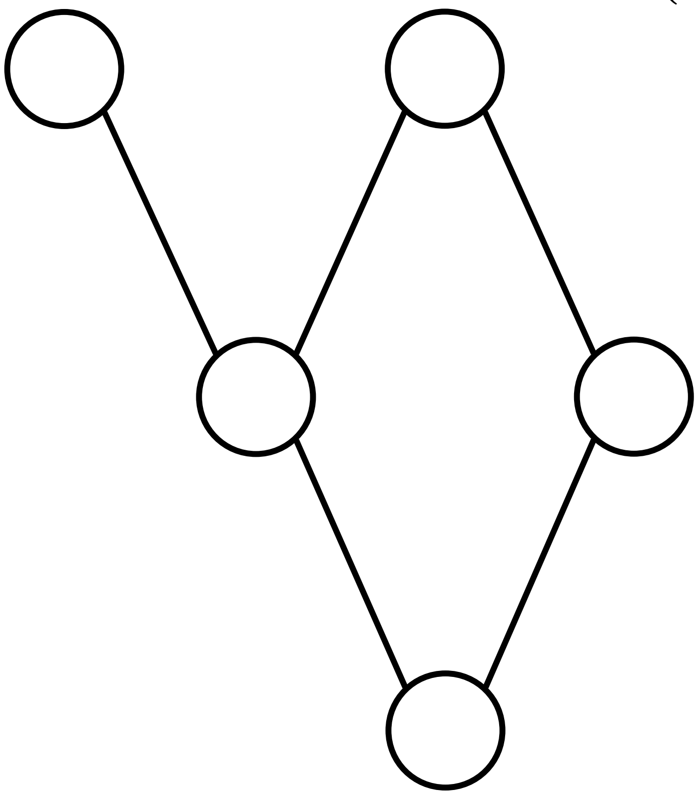


Parametrizing Edges Iteratively

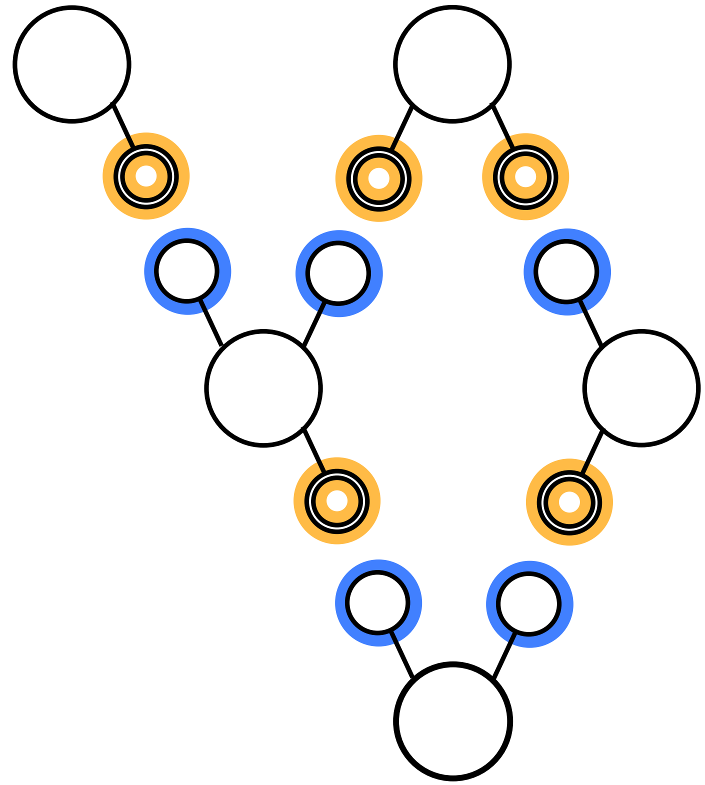


Parametrizing Edges Iteratively

$$\theta(X_i) = \alpha \frac{\partial Z}{\partial \theta(X_j)}$$



$$\theta(X_j) = \alpha \frac{\partial Z}{\partial \theta(X_i)}$$

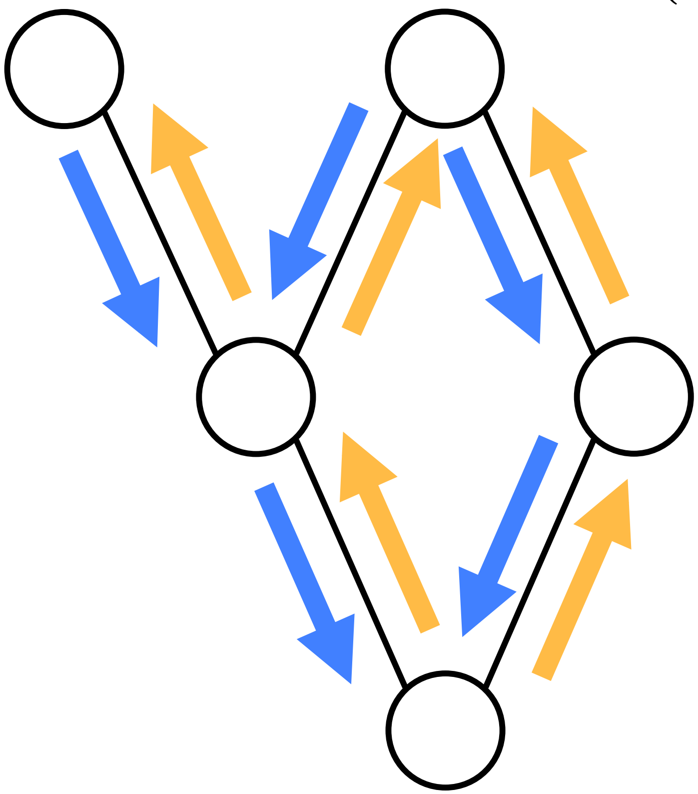


Iteration t

Convergence

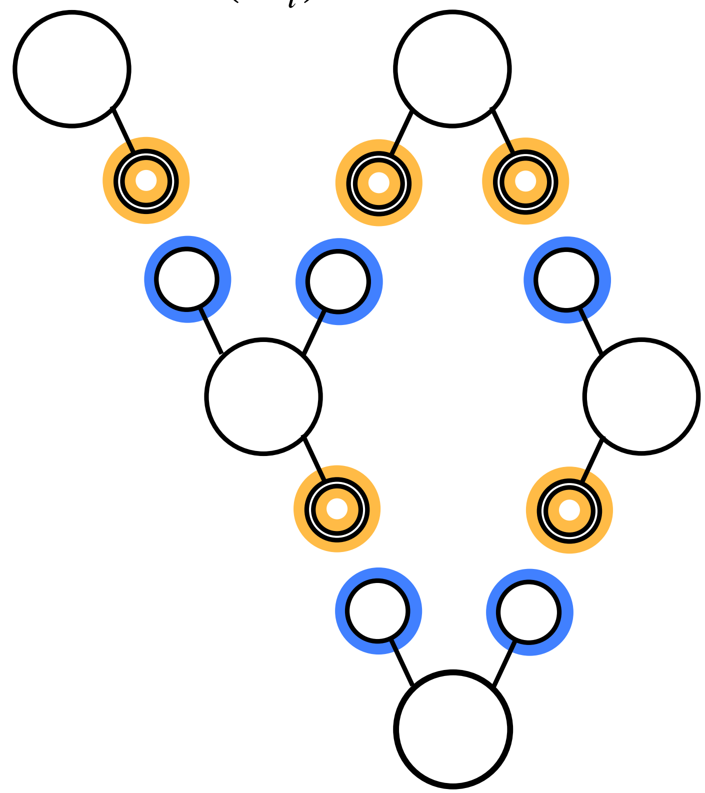
Characterizing Loopy Belief Propagation

$$\theta(X_i) = \alpha \frac{\partial Z}{\partial \theta(X_j)}$$



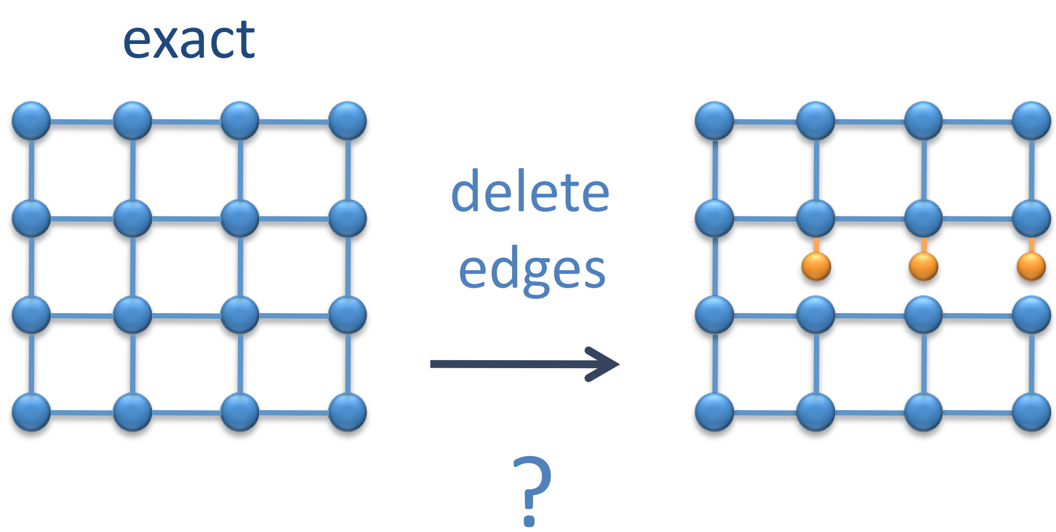
Iteration t

$$\theta(X_j) = \alpha \frac{\partial Z}{\partial \theta(X_i)}$$

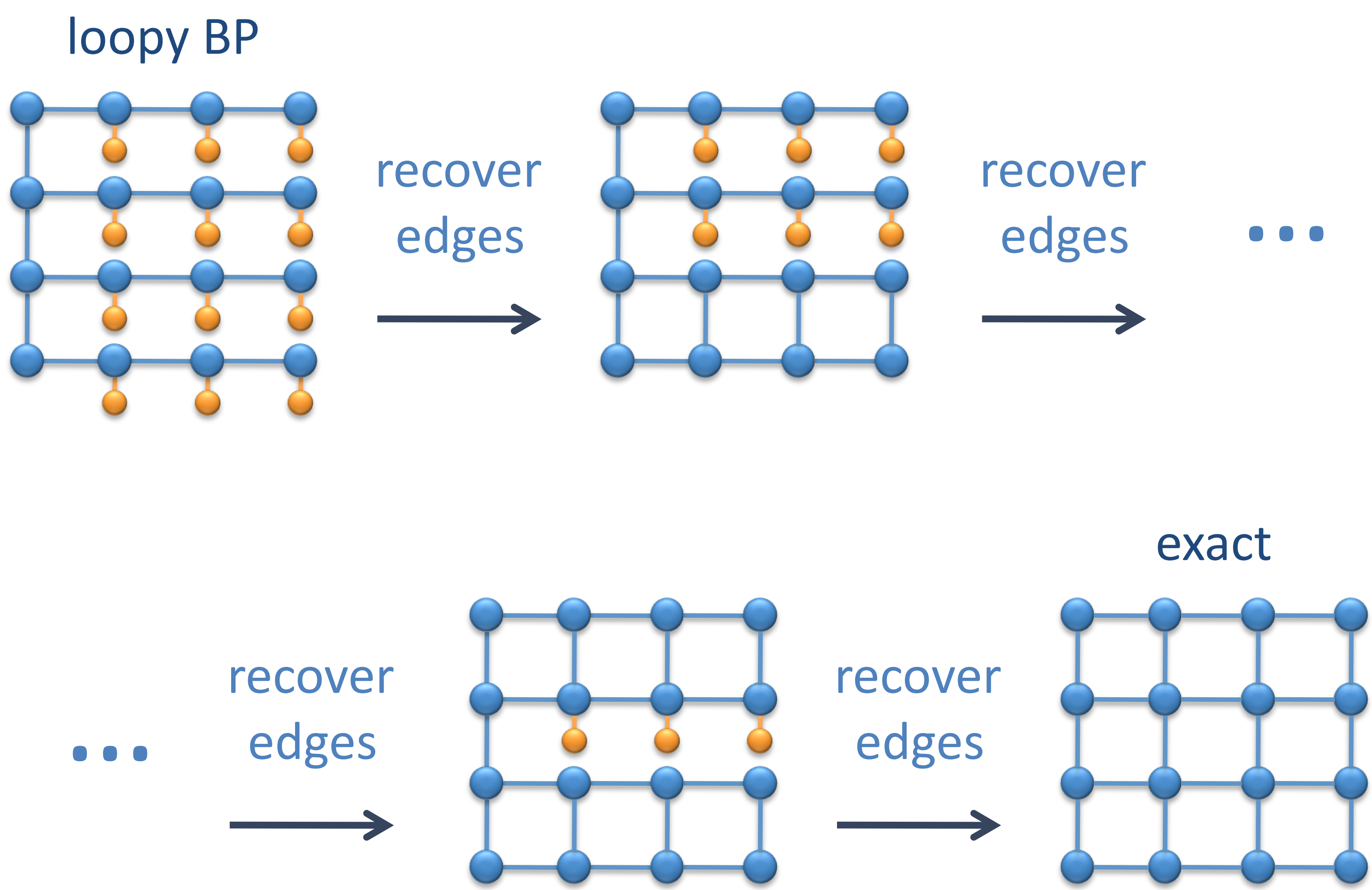


Iteration t

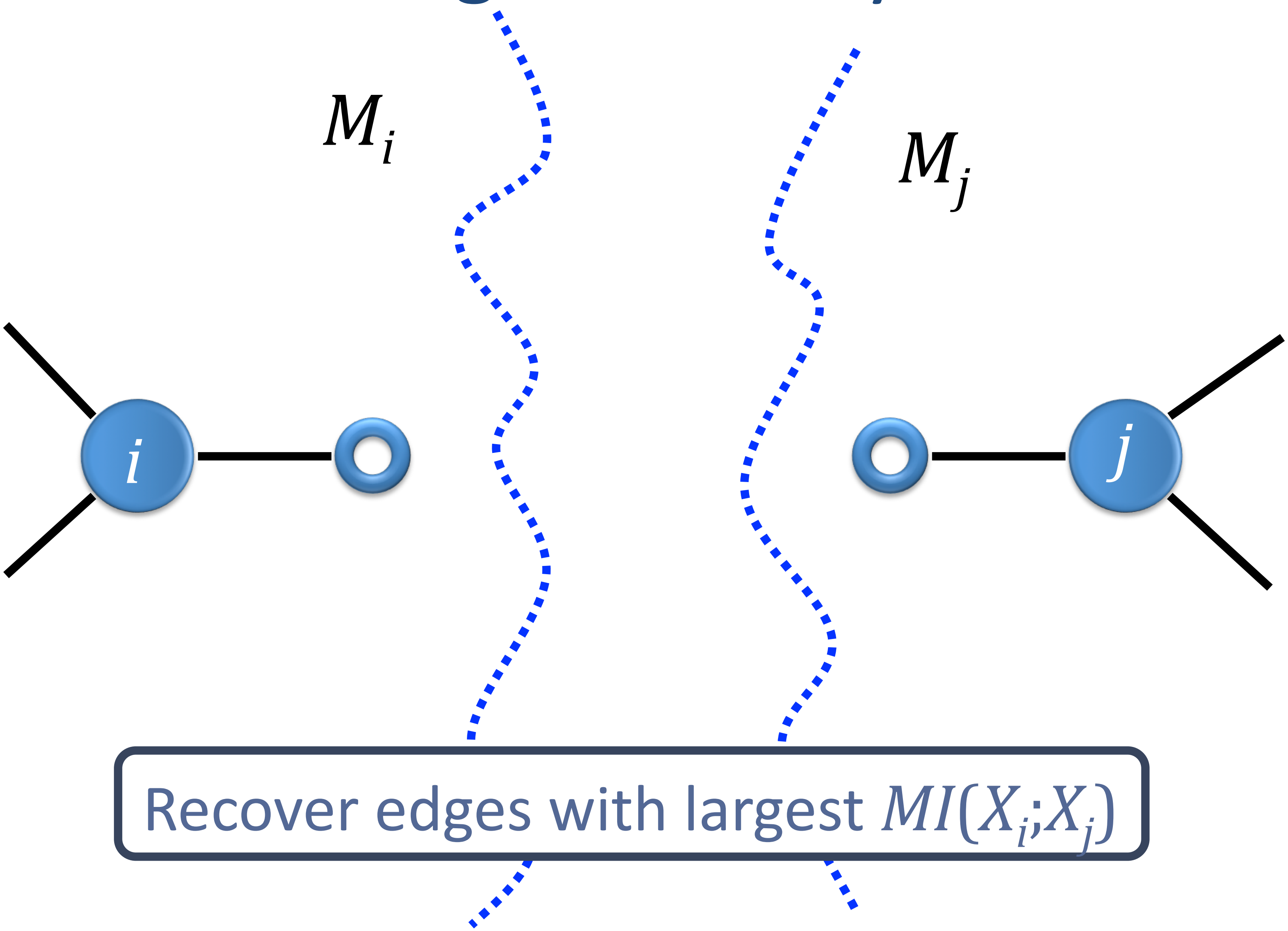
Which Edges to Delete?



Edge Recovery

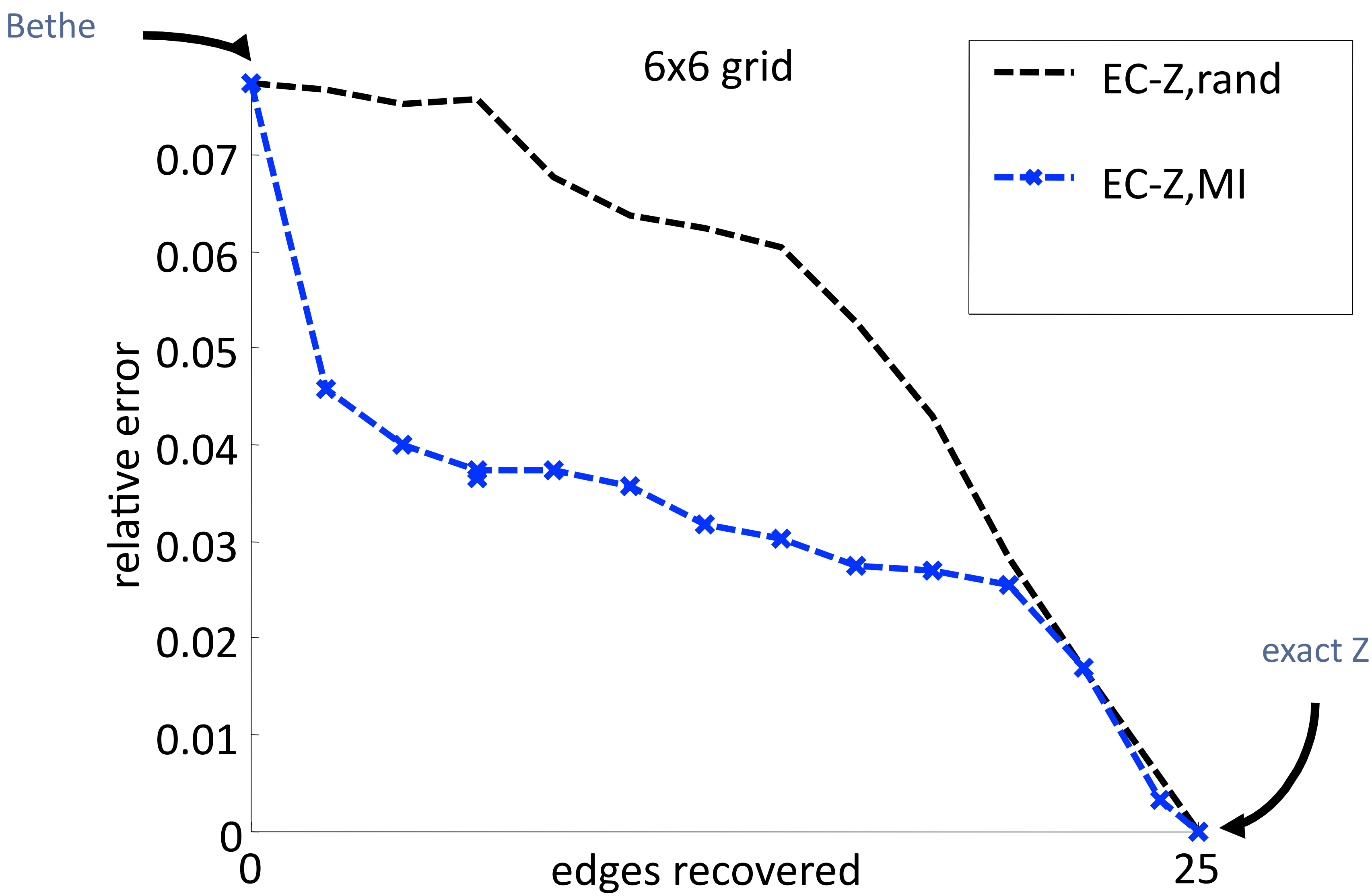


Edge Recovery



Recover edges with largest $MI(X_i;X_j)$

Edge Recovery



Evaluation Benchmarks

Benchmark	PR	MAR	MPE
CSP	8	8	55
Grids	20	20	40
Image Alignment			10
Medical Diagnosis	26	26	
Object Detection	96	96	92
Pedigree	4	4	
Protein Folding			21
Protein-Protein Interaction			8
Segmentation	50	50	50
TOTAL	204	204	287

Overall Results

PR Task: 20 Seconds

Solver	Score
edbr	1.7146
vgogate	2.1620
libDai	2.2775

MAR Task: 20 Seconds

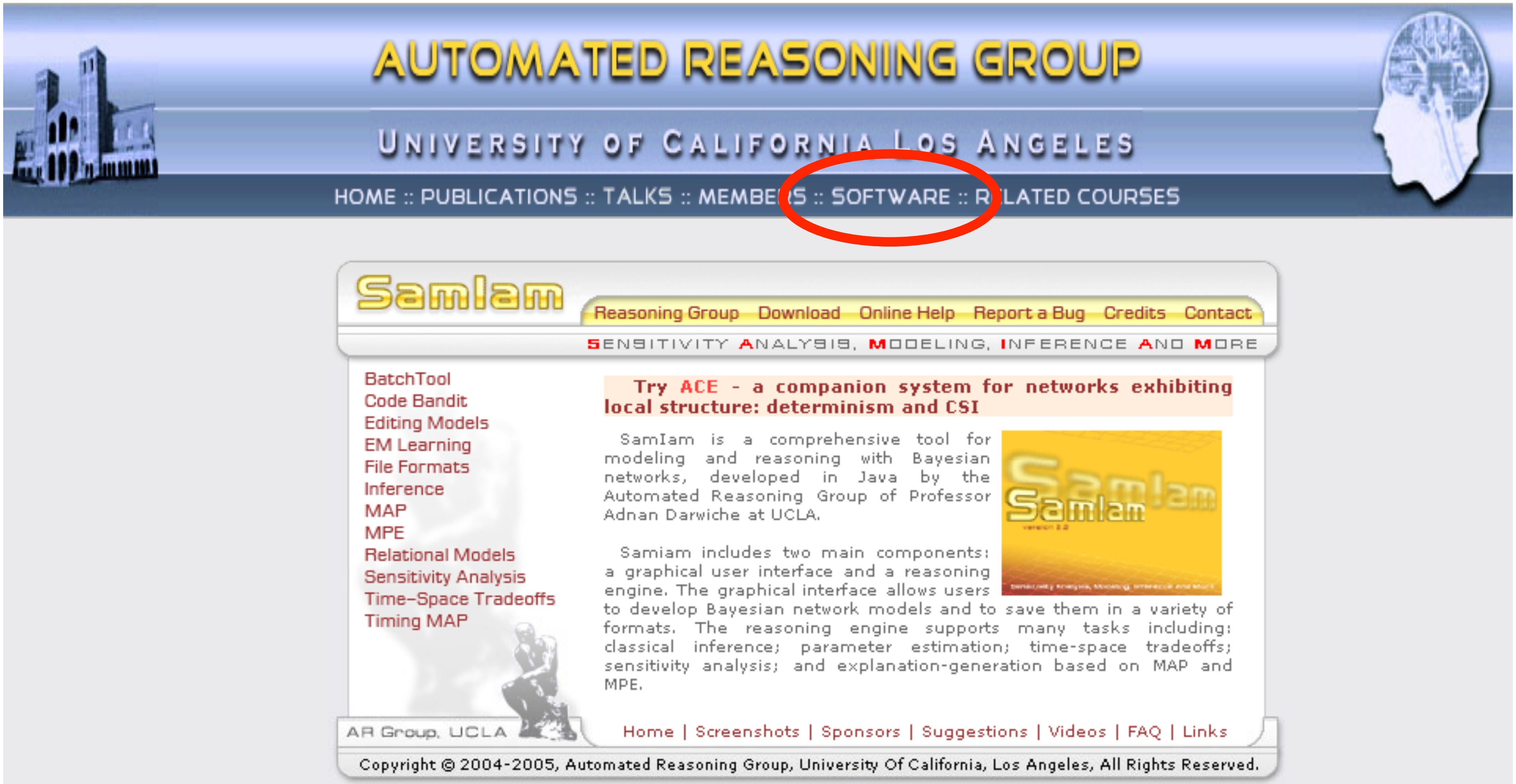
Solver	Score
edbq	0.2390
libDai2	0.3064
vgogate	0.4409

Ideally…

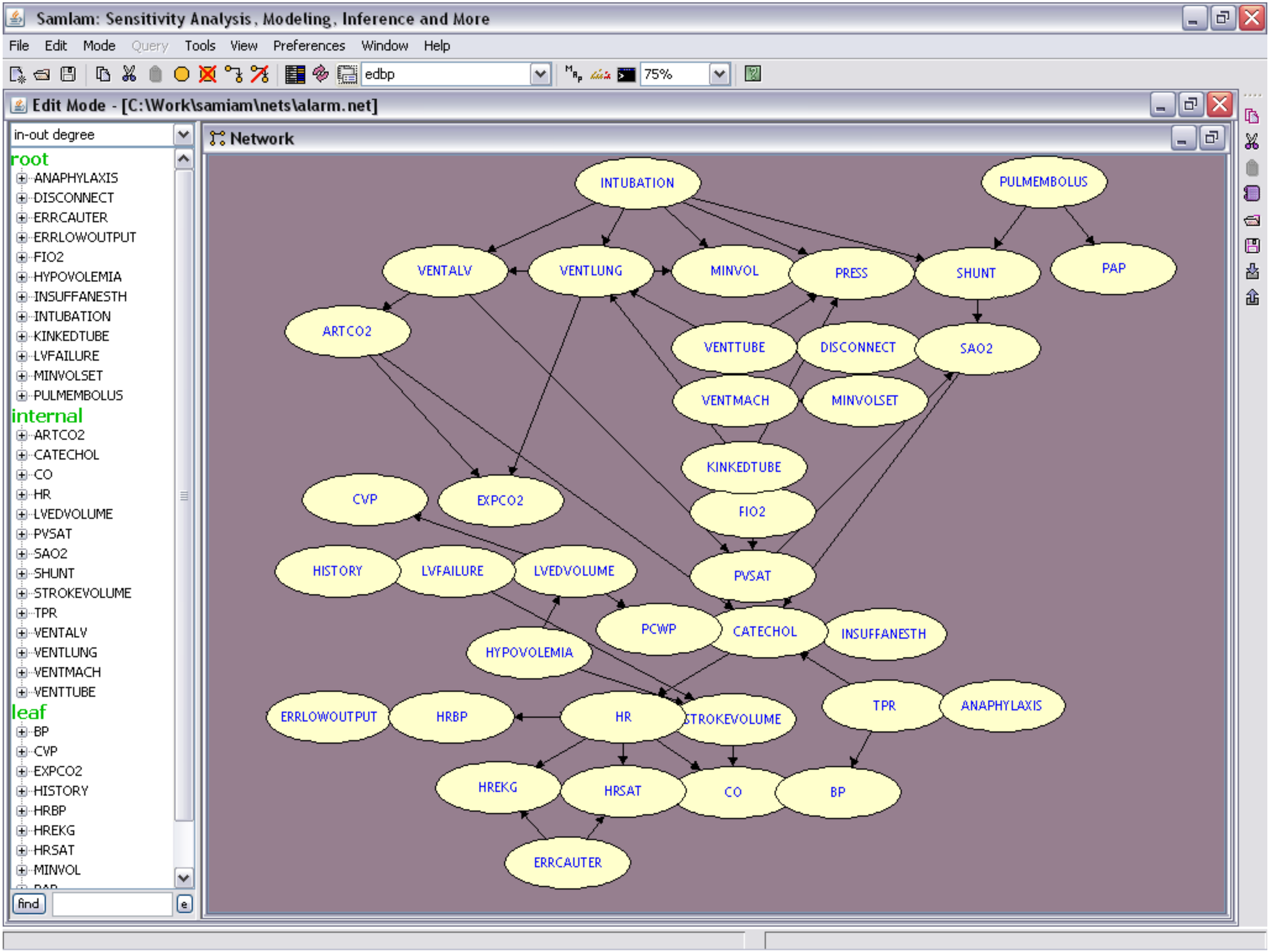
- Exact inference based on compiling CNFs
- Edge recovery using incremental compilation:
  - conjoin recovered equivalence constraint with current compilation
- Not there yet: more engineering needed!

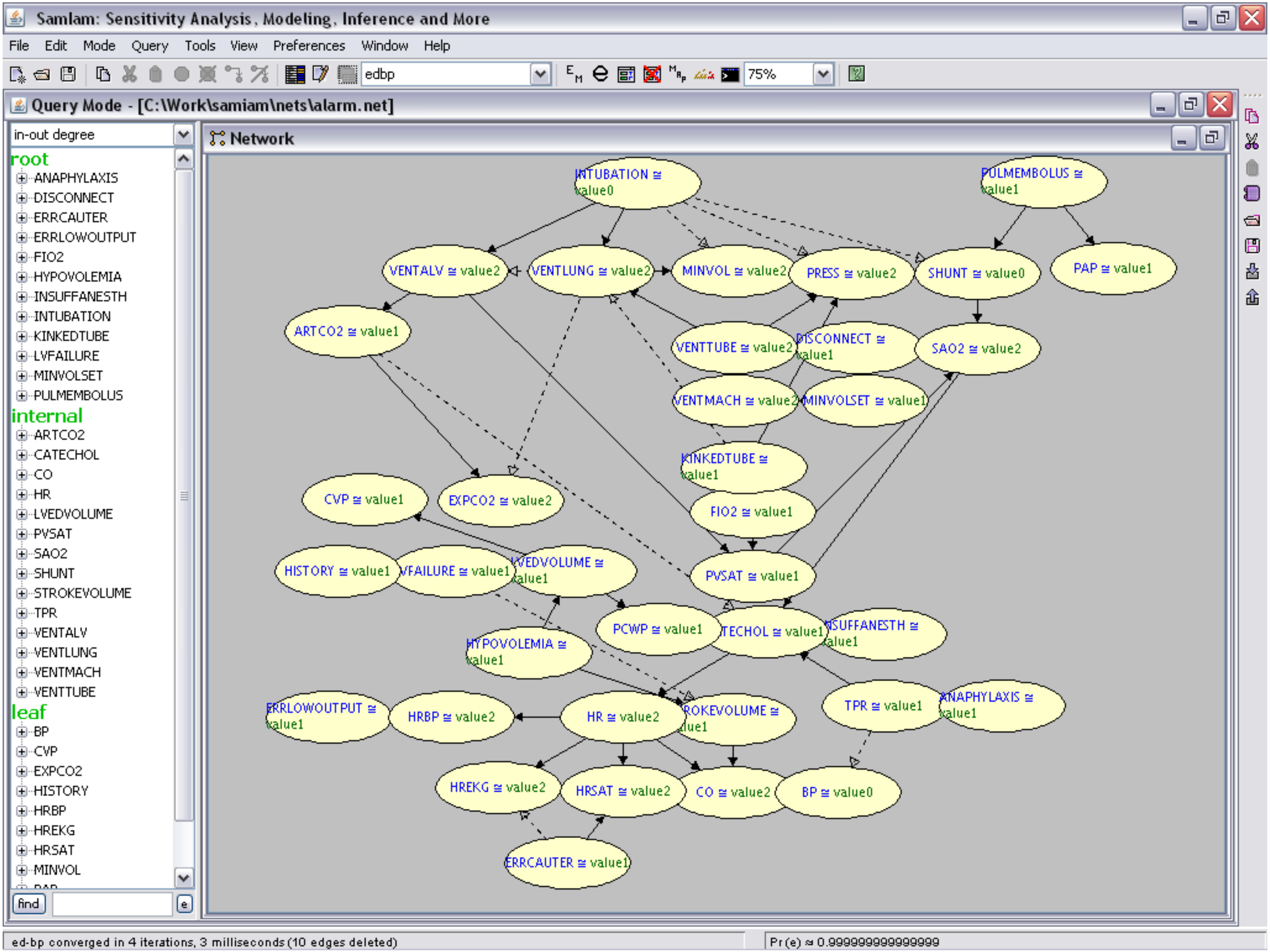
Key Ideas

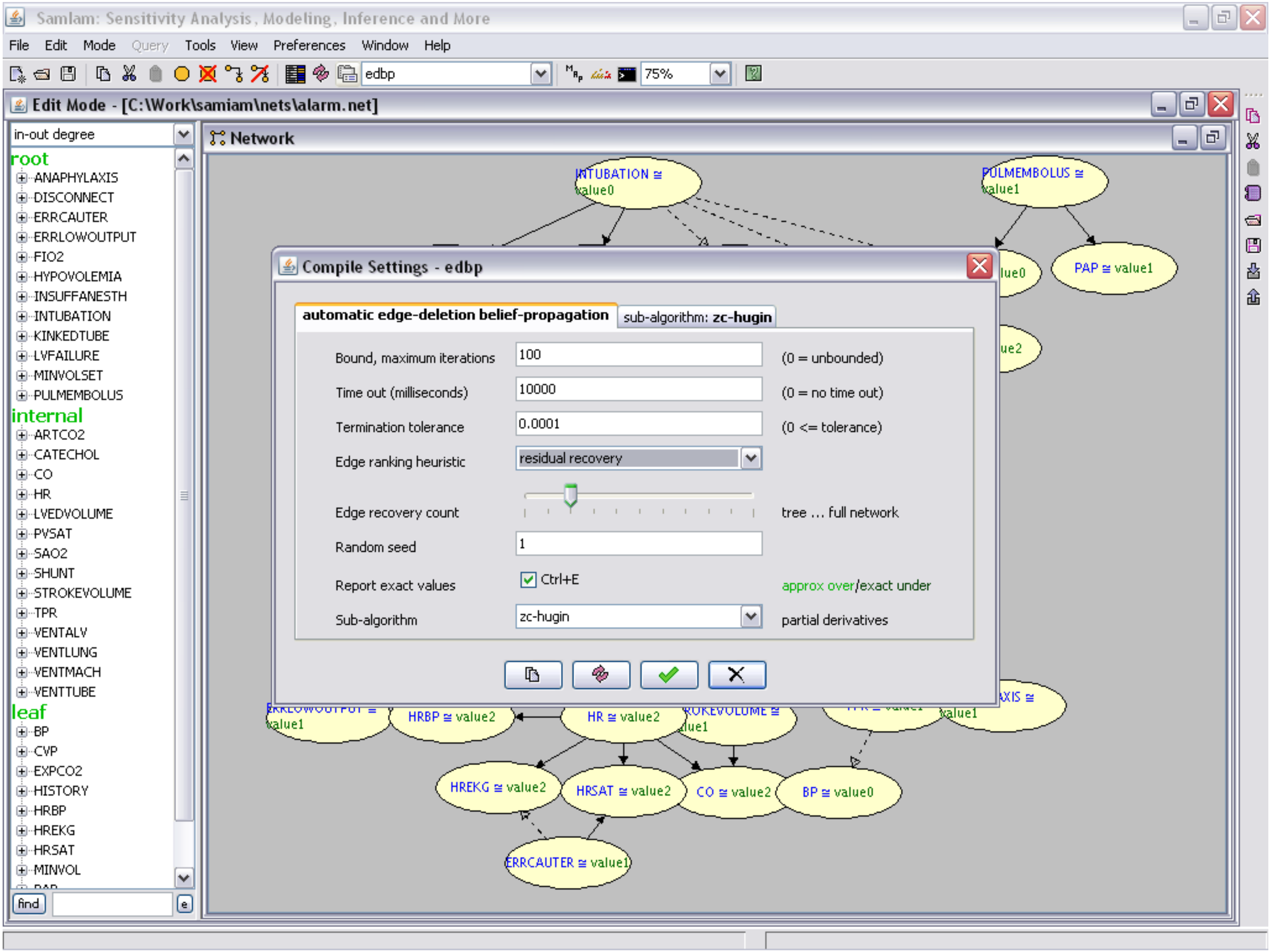
- **Approximate inference:** formulated as exact inference in an approximate model
- **Approximate models**: obtained by relaxing and compensating for equivalence constraints
- **Anytime inference:** selective recovery of equivalence constraints
- **Exact inference:** formulated in terms of enforcing decomposability and determinism of propositional knowledge bases

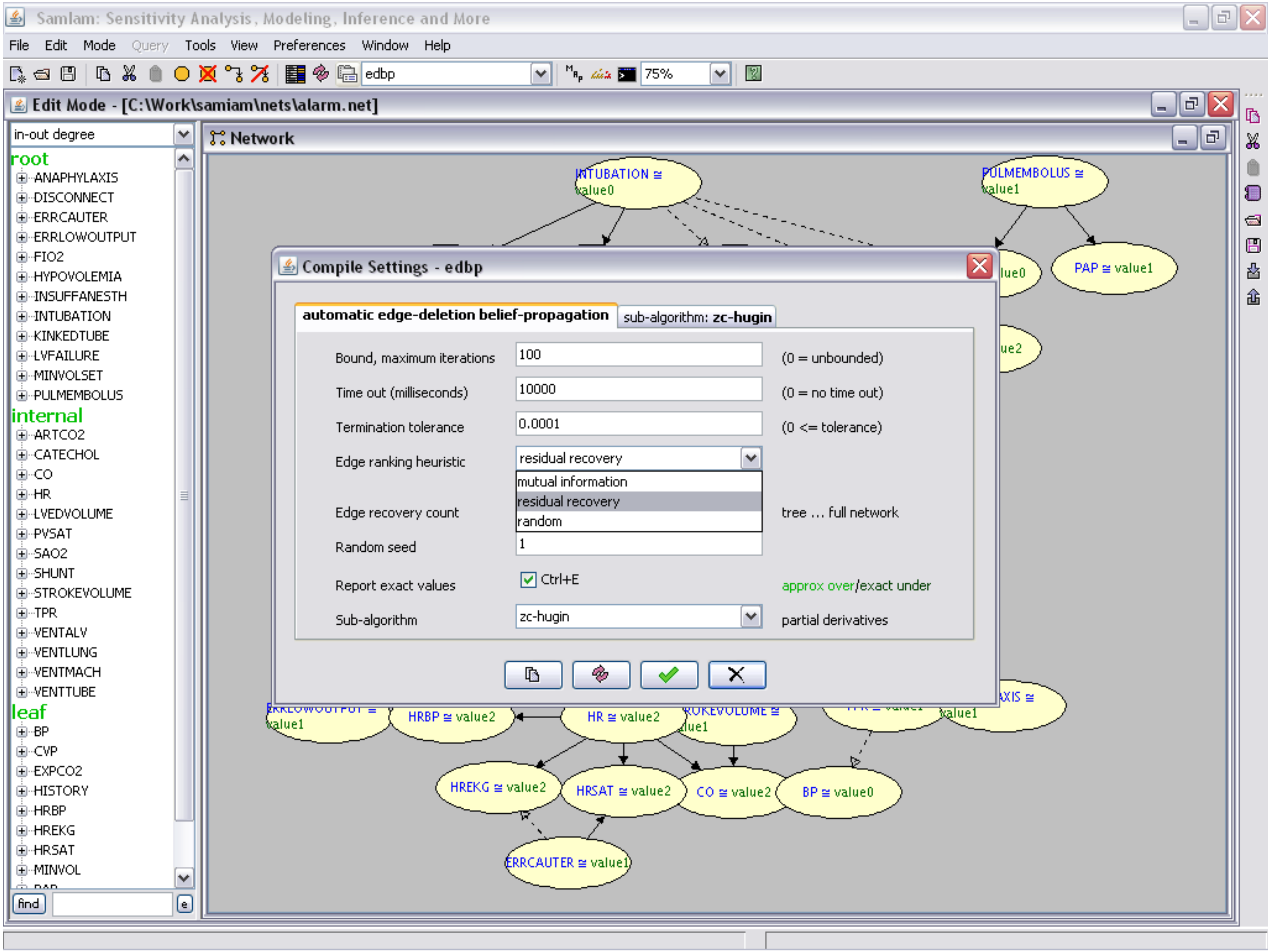


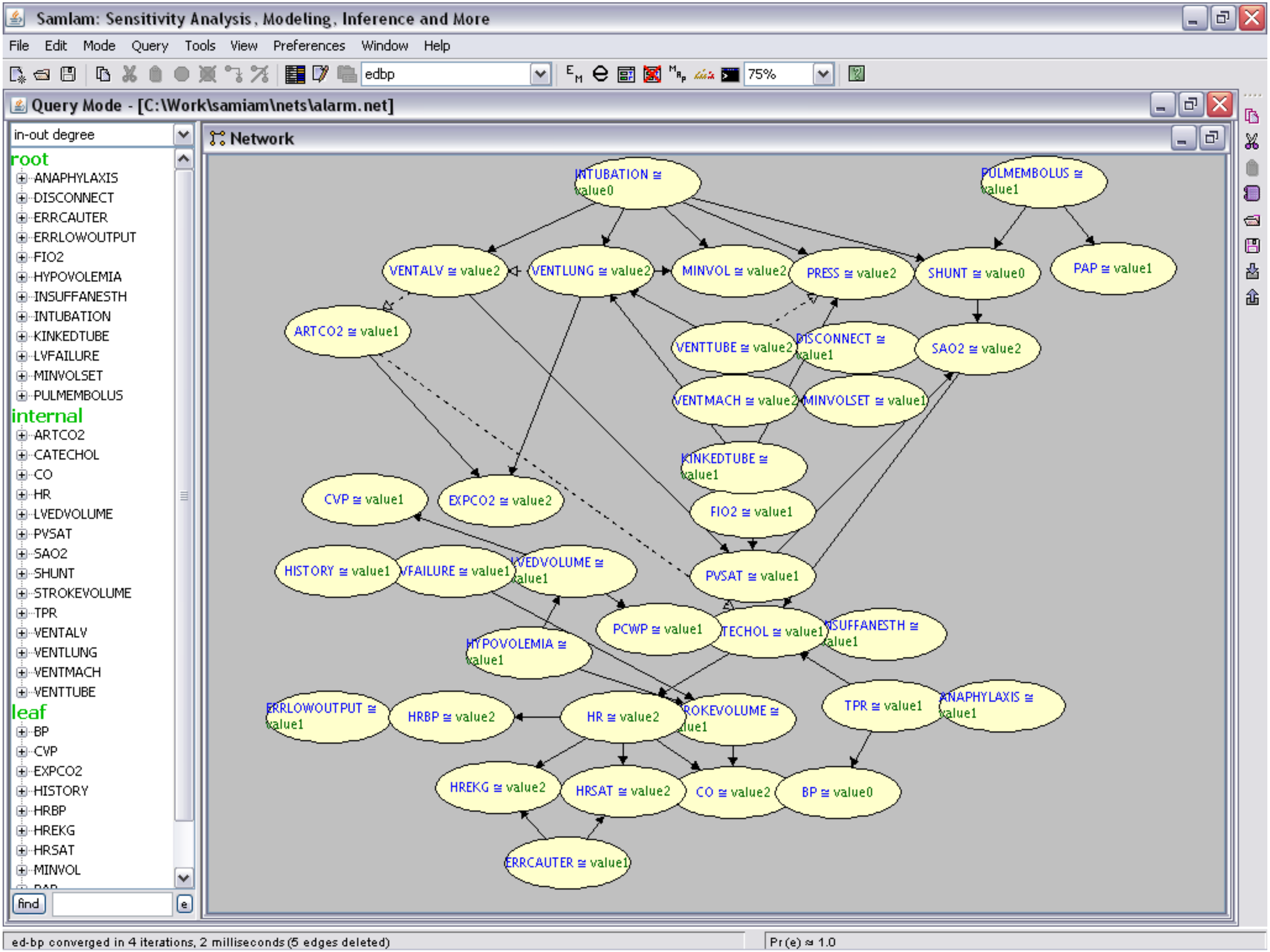
http://reasoning.cs.ucla.edu/samiam/


Samlam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
edbp
75%
Edit Mode - [C:\Work\samiam\nets\alarm.net]
in-out degree
root
ANAPHYLAXIS
DISCONNECT
ERRCAUTER
ERRLOWOUTPUT
FIO2
HYPOVOLEMIA
INSUFFANESTH
INTUBATION
KINKEDTUBE
LVFAILURE
MINVOLSET
PULMEMBOLUS
internal
ARTCO2
CATECHOL
CO
HR
LVEDVOLUME
PVSAT
SAO2
SHUNT
STROKEVOLUME
TPR
VENTALV
VENTLUNG
VENTMACH
VENTTUBE
leaf
BP
CVP
EXPCO2
HISTORY
HRBP
HREKG
HRSAT
MINVOL
find
e
Network
INTUBATION
PULMEMBOLUS
VENTALV
VENTLUNG
MINVOL
PRESS
SHUNT
PAP
ARTCO2
VENTTUBE
DISCONNECT
SAO2
VENTMACH
MINVOLSET
KINKEDTUBE
CVP
EXPCO2
FIO2
HISTORY
LVFAILURE
LVEDVOLUME
PVSAT
PCWP
CATECHOL
INSUFFANESTH
HYPOVOLEMIA
ERRLOWOUTPUT
HRBP
HR
STROKEVOLUME
TPR
ANAPHYLAXIS
HREKG
HRSAT
CO
BP
ERRCAUTER


Samlam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
edbp
75%
Query Mode - [C:\Work\samiam\nets\alarm.net]
in-out degree
Network
root
ANAPHYLAXIS
DISCONNECT
ERRCAUTER
ERRLOWOUTPUT
FIO2
HYPOVOLEMIA
INSUFFANESTH
INTUBATION
KINKEDTUBE
LVFAILURE
MINVOLSET
PULMEMBOLUS
internal
ARTCO2
CATECHOL
CO
HR
LVEDVOLUME
PVSAT
SAO2
SHUNT
STROKEVOLUME
TPR
VENTALV
VENTLUNG
VENTMACH
VENTTUBE
leaf
BP
CVP
EXPCO2
HISTORY
HRBP
HREKG
HRSAT
MINVOL
find
INTUBATION ≅ value0
PULMEMBOLUS ≅ value1
VENTALV ≅ value2
VENTLUNG ≅ value2
MINVOL ≅ value2
PRESS ≅ value2
SHUNT ≅ value0
PAP ≅ value1
ARTCO2 ≅ value1
VENTTUBE ≅ value2
DISCONNECT ≅ value1
SAO2 ≅ value2
VENTMACH ≅ value2
MINVOLSET ≅ value1
KINKEDTUBE ≅ value1
CVP ≅ value1
EXPCO2 ≅ value2
FIO2 ≅ value1
HISTORY ≅ value1
LVFAILURE ≅ value1
LVEDVOLUME ≅ value1
PVSAT ≅ value1
PCWP ≅ value1
CATECHOL ≅ value1
INSUFFANESTH ≅ value1
HYPOVOLEMIA ≅ value1
ERRLOWOUTPUT ≅ value1
HRBP ≅ value2
HR ≅ value2
STROKEVOLUME ≅ value1
TPR ≅ value1
ANAPHYLAXIS ≅ value1
HREKG ≅ value2
HRSAT ≅ value2
CO ≅ value2
BP ≅ value0
ERRCAUTER ≅ value1
ed-bp converged in 4 iterations, 3 milliseconds (10 edges deleted)
Pr(e) ≈ 0.999999999999999


Samlam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
edbp
75%
Edit Mode - [C:\Work\samiam\nets\alarm.net]
in-out degree
root
ANAPHYLAXIS
DISCONNECT
ERRCAUTER
ERRLOWOUTPUT
FIO2
HYPOVOLEMIA
INSUFFANESTH
INTUBATION
KINKEDTUBE
LVFAILURE
MINVOLSET
PULMEMBOLUS
internal
ARTCO2
CATECHOL
CO
HR
LVEDVOLUME
PVSAT
SAO2
SHUNT
STROKEVOLUME
TPR
VENTALV
VENTLUNG
VENTMACH
VENTTUBE
leaf
BP
CVP
EXPCO2
HISTORY
HRBP
HREKG
HRSAT
MINVOL
find
e
Network
INTUBATION ≅ value0
PULMEMBOLUS ≅ value1
PAP ≅ value1
HRBP ≅ value2
HR ≅ value2
HREKG ≅ value2
HRSAT ≅ value2
CO ≅ value2
BP ≅ value0
ERRCAUTER ≅ value1
Compile Settings - edbp
automatic edge-deletion belief-propagation
sub-algorithm: zc-hugin
Bound, maximum iterations
100
(0 = unbounded)
Time out (milliseconds)
10000
(0 = no time out)
Termination tolerance
0.0001
(0 <= tolerance)
Edge ranking heuristic
residual recovery
Edge recovery count
tree … full network
Random seed
1
Report exact values
Ctrl+E
approx over/exact under
Sub-algorithm
zc-hugin
partial derivatives


Samiam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
edbp
75%
Edit Mode - [C:\Work\samiam\nets\alarm.net]
in-out degree
root
ANAPHYLAXIS
DISCONNECT
ERRCAUTER
ERRLOWOUTPUT
FIO2
HYPOVOLEMIA
INSUFFANESTH
INTUBATION
KINKEDTUBE
LVFAILURE
MINVOLSET
PULMEMBOLUS
internal
ARTCO2
CATECHOL
CO
HR
LVEDVOLUME
PVSAT
SAO2
SHUNT
STROKEVOLUME
TPR
VENTALV
VENTLUNG
VENTMACH
VENTTUBE
leaf
BP
CVP
EXPCO2
HISTORY
HRBP
HREKG
HRSAT
MINVOL
find
e
Network
INTUBATION ≅ value0
PULMEMBOLUS ≅ value1
PAP ≅ value1
Compile Settings - edbp
automatic edge-deletion belief-propagation
sub-algorithm: zc-hugin
Bound, maximum iterations
100
(0 = unbounded)
Time out (milliseconds)
10000
(0 = no time out)
Termination tolerance
0.0001
(0 <= tolerance)
Edge ranking heuristic
residual recovery
mutual information
residual recovery
random
Edge recovery count
tree ... full network
Random seed
1
Report exact values
Ctrl+E
approx over/exact under
Sub-algorithm
zc-hugin
partial derivatives
HRBP ≅ value2
HR ≅ value2
HREKG ≅ value2
HRSAT ≅ value2
CO ≅ value2
BP ≅ value0
ERRCAUTER ≅ value1


Samiam: Sensitivity Analysis, Modeling, Inference and More
File
Edit
Mode
Query
Tools
View
Preferences
Window
Help
edbp
75%
Query Mode - [C:\Work\samiam\nets\alarm.net]
in-out degree
Network
root
ANAPHYLAXIS
DISCONNECT
ERRCAUTER
ERRLOWOUTPUT
FIO2
HYPOVOLEMIA
INSUFFANESTH
INTUBATION
KINKEDTUBE
LVFAILURE
MINVOLSET
PULMEMBOLUS
internal
ARTCO2
CATECHOL
CO
HR
LVEDVOLUME
PVSAT
SAO2
SHUNT
STROKEVOLUME
TPR
VENTALV
VENTLUNG
VENTMACH
VENTTUBE
leaf
BP
CVP
EXPCO2
HISTORY
HRBP
HREKG
HRSAT
MINVOL
find
e
INTUBATION ≅ value0
PULMEMBOLUS ≅ value1
VENTALV ≅ value2
VENTLUNG ≅ value2
MINVOL ≅ value2
PRESS ≅ value2
SHUNT ≅ value0
PAP ≅ value1
ARTCO2 ≅ value1
VENTTUBE ≅ value2
DISCONNECT ≅ value1
SAO2 ≅ value2
VENTMACH ≅ value2
MINVOLSET ≅ value1
KINKEDTUBE ≅ value1
CVP ≅ value1
EXPCO2 ≅ value2
FIO2 ≅ value1
HISTORY ≅ value1
VFAILURE ≅ value1
VEDVOLUME ≅ value1
PVSAT ≅ value1
PCWP ≅ value1
TECHOL ≅ value1
NSUFFANESTH ≅ value1
HYPOVOLEMIA ≅ value1
ERRLOWOUTPUT ≅ value1
HRBP ≅ value2
HR ≅ value2
ROKEVOLUME ≅ lue1
TPR ≅ value1
ANAPHYLAXIS ≅ value1
HREKG ≅ value2
HRSAT ≅ value2
CO ≅ value2
BP ≅ value0
ERRCAUTER ≅ value1
ed-bp converged in 4 iterations, 2 milliseconds (5 edges deleted)
Pr(e) ≈ 1.0

Adnan Darwiche

MODELING AND REASONING with BAYESIAN NETWORKS

CAMBRIDGE

http://reasoning.cs.ucla.edu