Lifted Junction Tree Algorithm

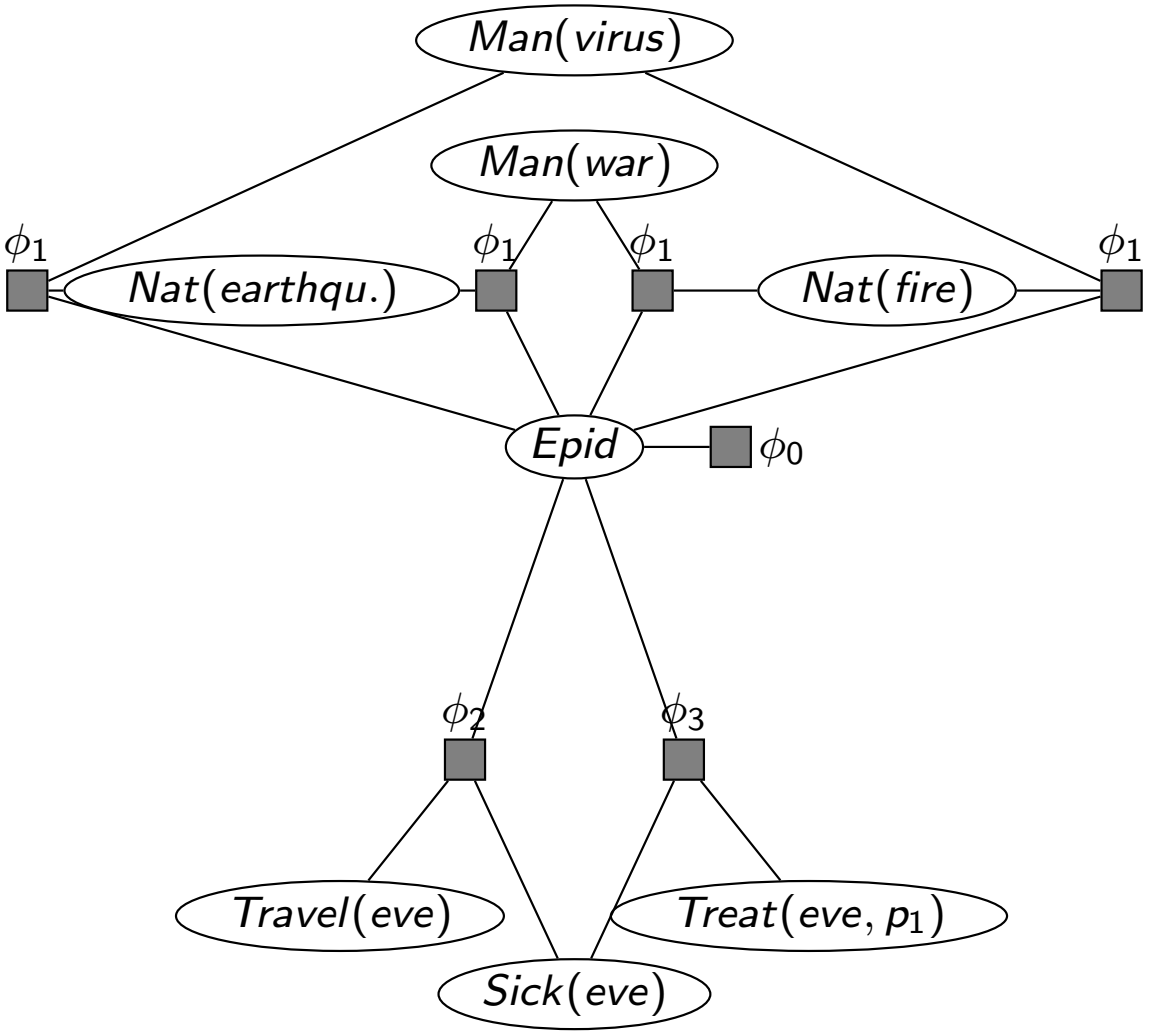
Most Probable Explanation

Tanya Braun, Ralf Möller

Institute of Information Systems
University of Lübeck

June 20, 2018

Probabilistic Graphical Models

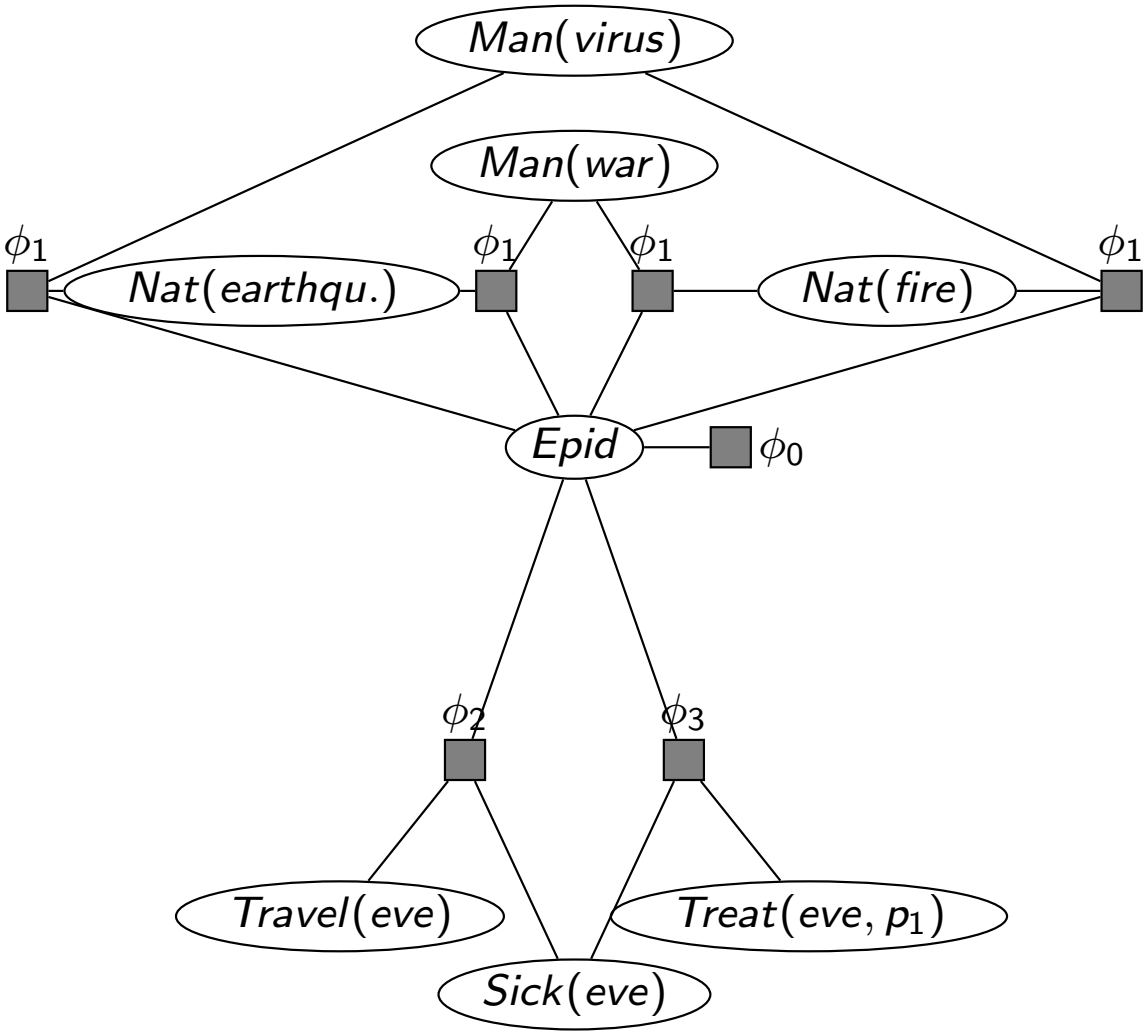


Query answering (QA): Eliminate all non-query variables

Problem: Practical QA

Large models

Many queries



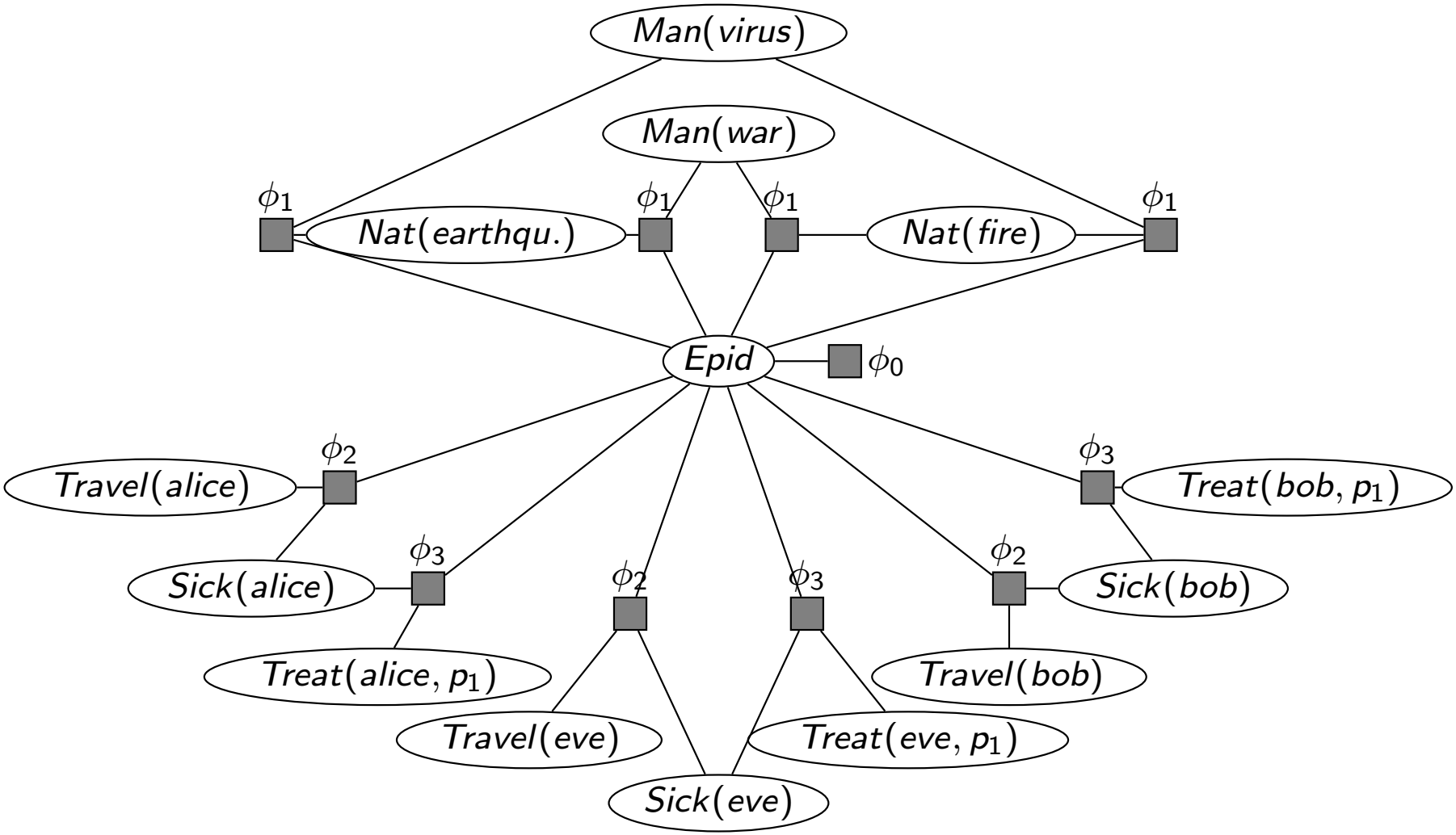
Symmetries

Clusters

Problem: Practical QA

Large models

Many queries



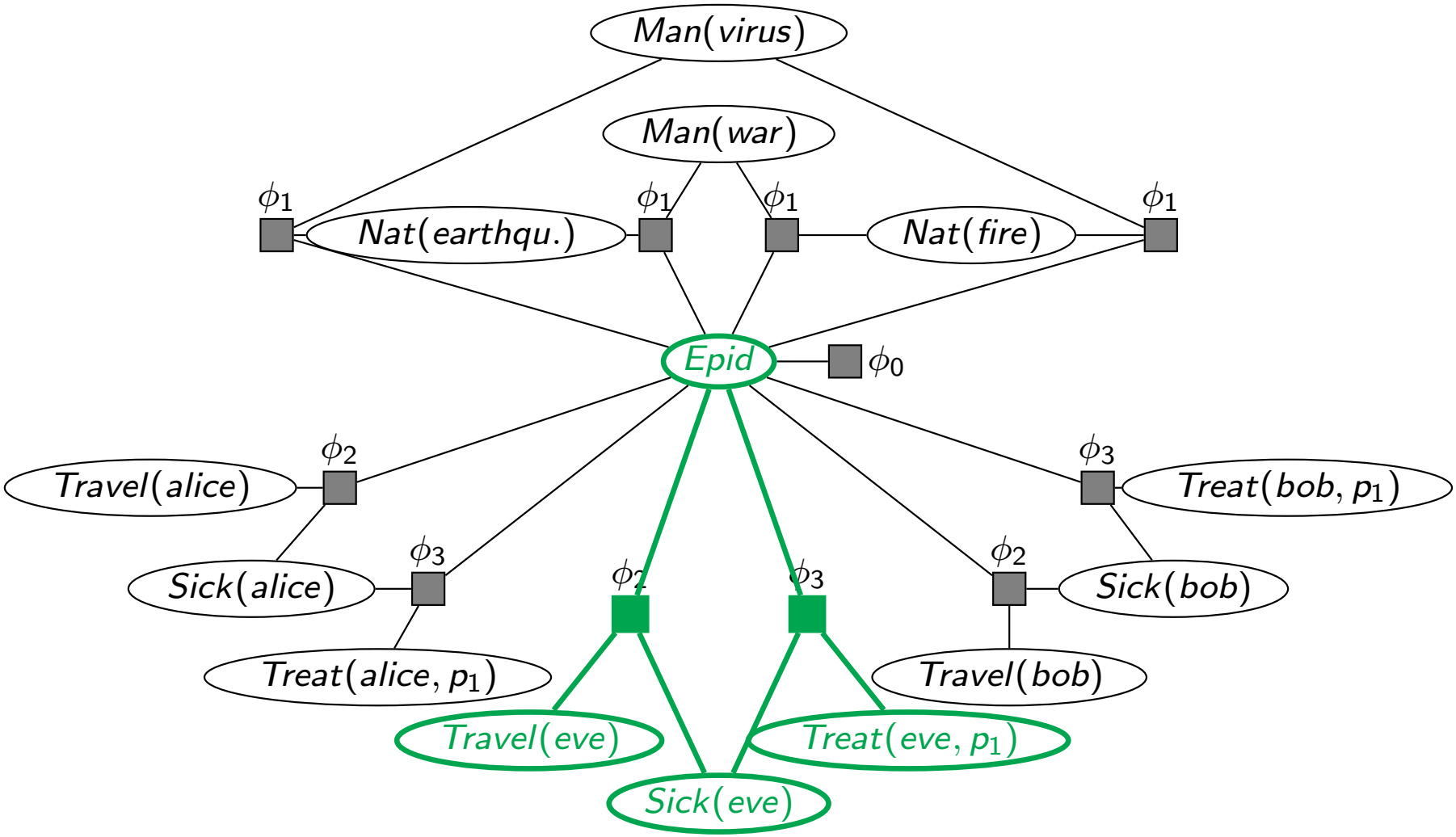
Symmetries

Clusters

Problem: Practical QA

Large models

Many queries



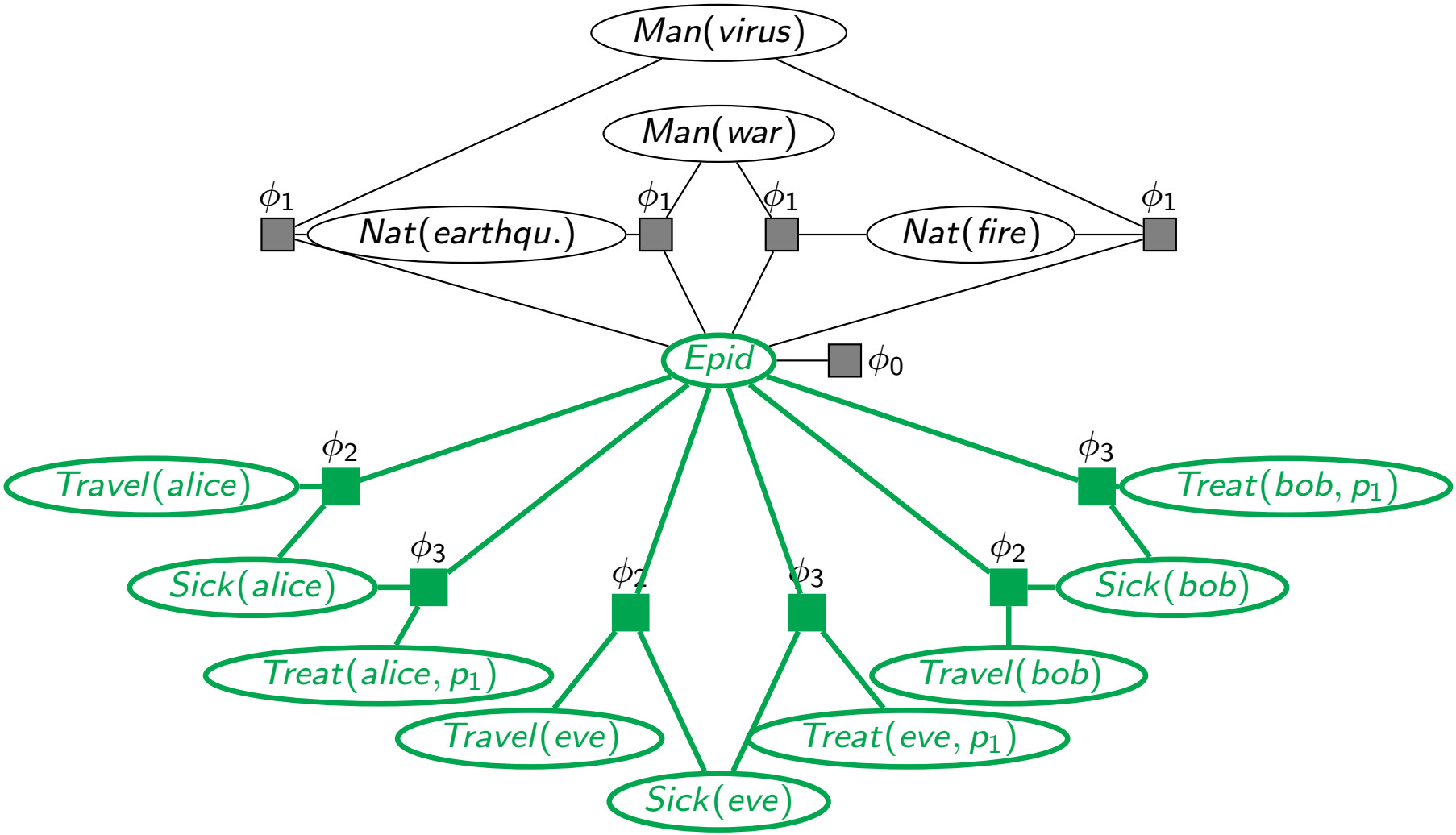
Symmetries

Clusters

Problem: Practical QA

Large models

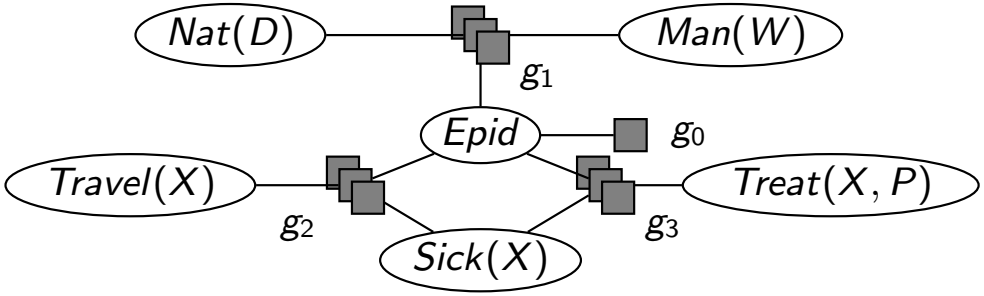
Many queries

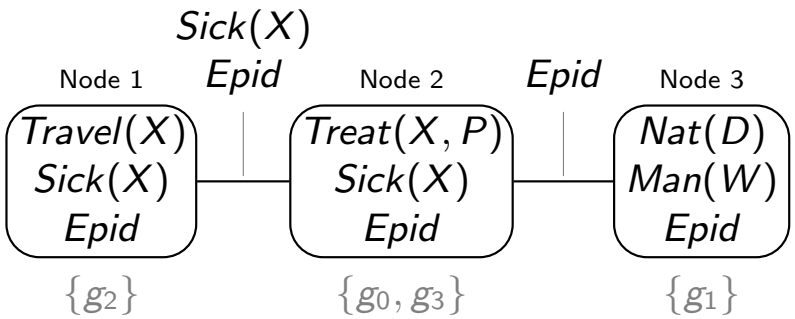


Symmetries

Clusters

Approach: Parameters and Clusters



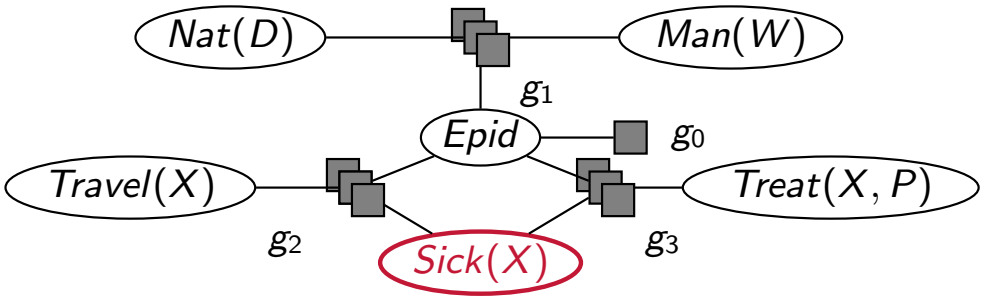


- Parameterisation
  - → Avoid explosion of nodes
- Lifted variable elimination (LVE)[1]
  - → Elimination: $\sum$
  - → Save computations
  - → Each query in isolation
- Multiple queries: Junction Tree[2]
  - → Cluster representation, messages
- Lifted Junction Tree Algorithm (LJT)[3]
  - → LVE as subroutine
  - → Save computations
  - → Ground probability queries

[1] Poole (2003), de Salvo Braz et al. (2006), Milch et al. (2008), Apsel & Brafman (2011), Taghipour et al. (2013)
[2] Lauritzen & Spiegelhalter (1988) [3] Braun & Möller (2016, 2017)

Marginal Distribution Ground Queries

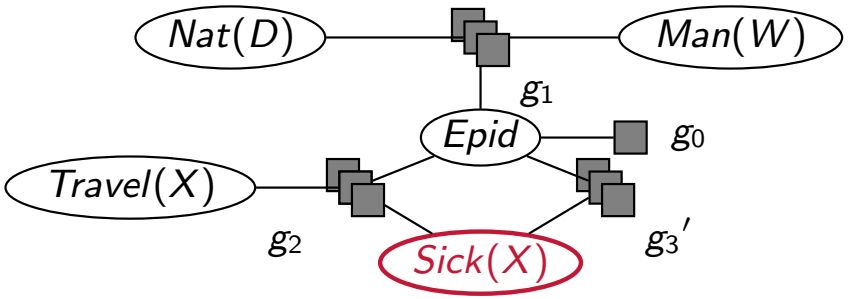
$P(Sick(eve))$



QA: LVE eliminates

Marginal Distribution Ground Queries

$P(Sick(eve))$

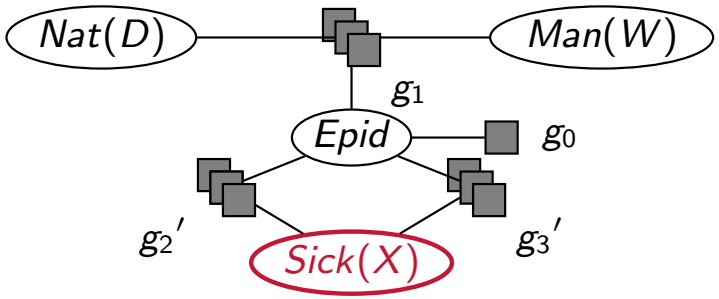


QA: LVE eliminates

- $Treat(X, P)$ from g_3

Marginal Distribution Ground Queries

$P(Sick(eve))$

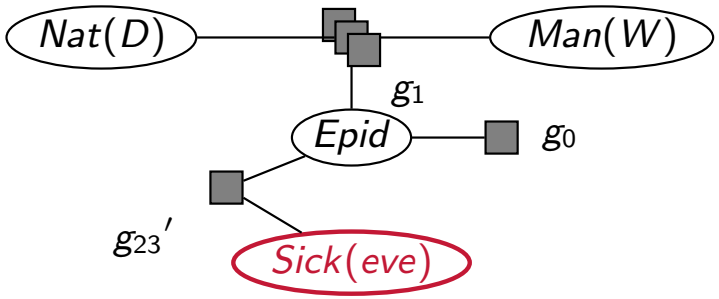


QA: LVE eliminates

- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2

Marginal Distribution Ground Queries

$P(Sick(eve))$

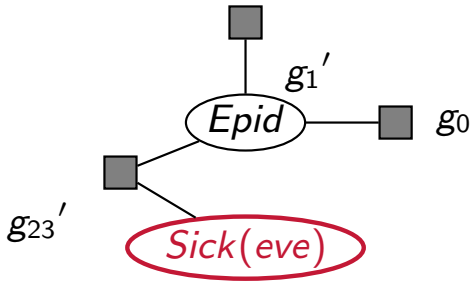


QA: LVE eliminates

- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$

Marginal Distribution Ground Queries

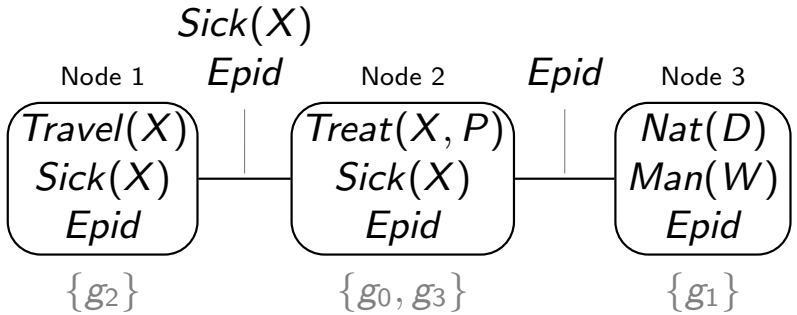
$P(Sick(eve))$

QA: LVE eliminates

- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

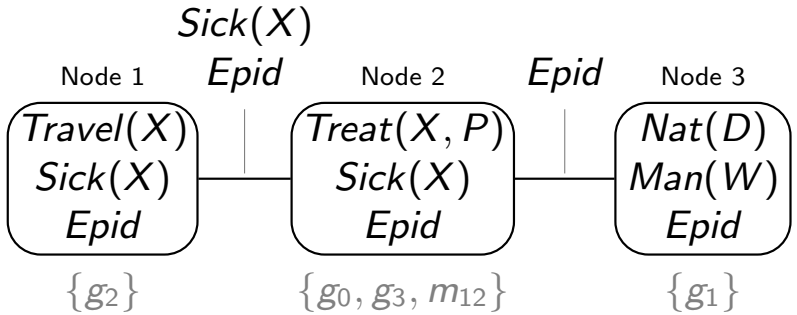
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT passes messages (indep. of query)

- Message = query on edge variables

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

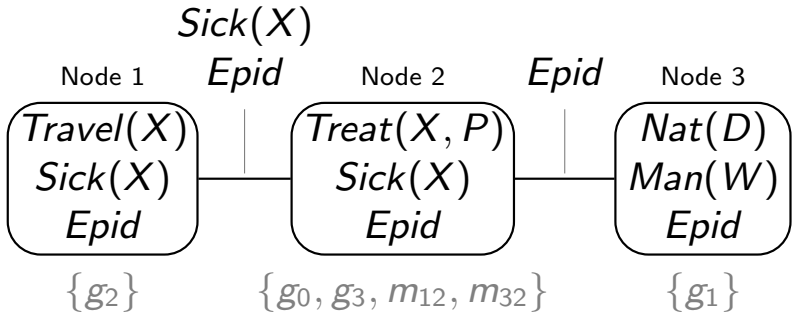
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT passes messages (indep. of query)

- Message = query on edge variables
- $1 \longrightarrow 2$: $P(Sick(X), Epid)$
  - Eliminate $Travel(X)$

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

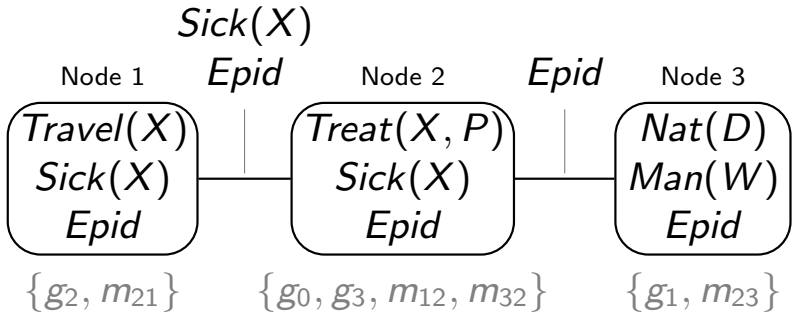
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT passes messages (indep. of query)

- Message = query on edge variables
- $1 \longrightarrow 2$: $P(Sick(X), Epid)$
  - Eliminate $Travel(X)$
- $3 \longrightarrow 2$: $P(Epid)$
  - Eliminate $Nat(D)$, $Man(W)$

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

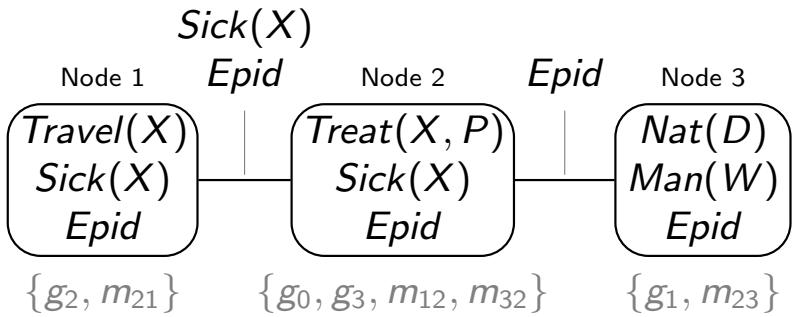
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT passes messages (indep. of query)

- Message = query on edge variables
- $1 \longrightarrow 2$: $P(Sick(X), Epid)$
  - Eliminate $Travel(X)$
- $3 \longrightarrow 2$: $P(Epid)$
  - Eliminate $Nat(D)$, $Man(W)$
- Messages from node 2

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

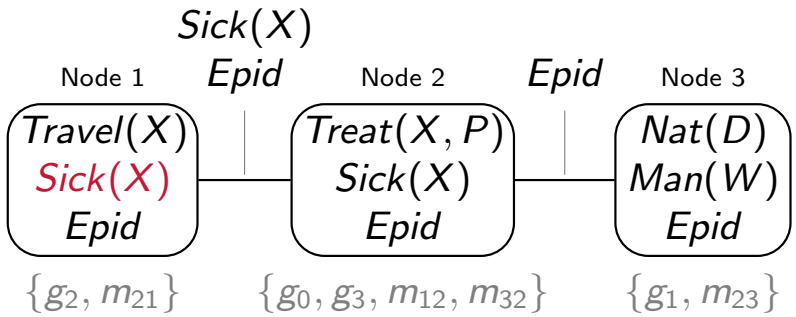
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT answers queries

- Node where query term appears

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

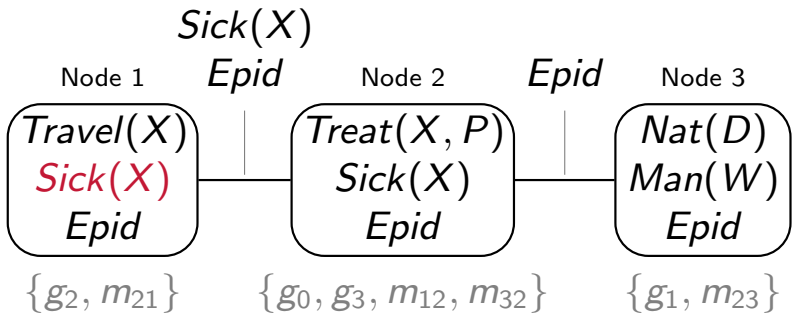
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT answers queries

- Node where query term appears
- Node 1: g_2, m_{12}

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

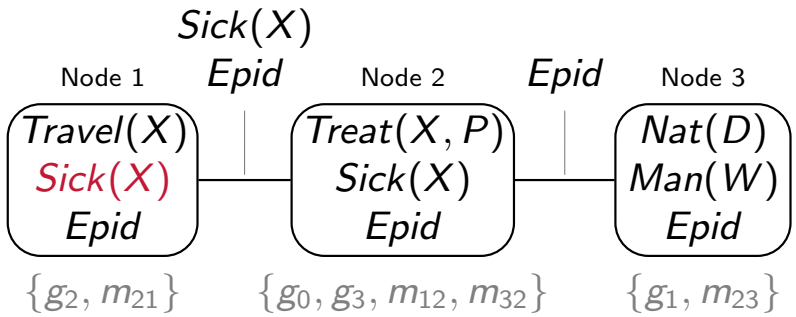
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT answers queries

- Node where query term appears
- Node 1: g_2, m_{12}
  - Eliminate $Travel(X)$ from g_2

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

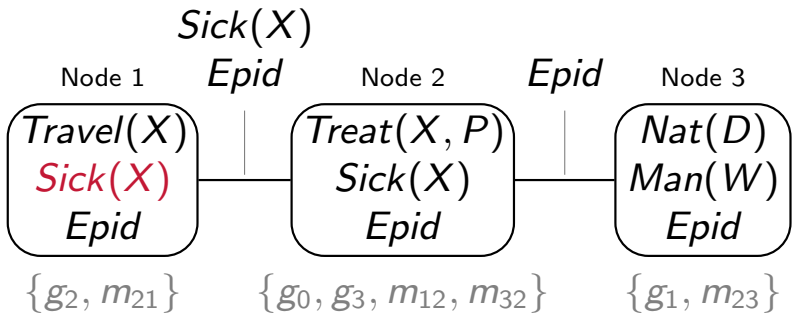
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT answers queries

- Node where query term appears
- Node 1: g_2, m_{12}
  - Eliminate $Travel(X)$ from g_2
  - Eliminate $Sick(X)$, $X \neq eve$, from $g_2' \cdot m_{21}$

Marginal Distribution Ground Queries

$P(Sick(eve))$



QA: LVE eliminates

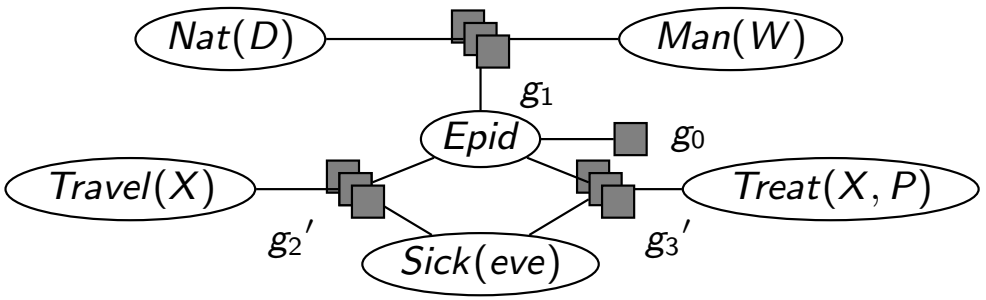
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

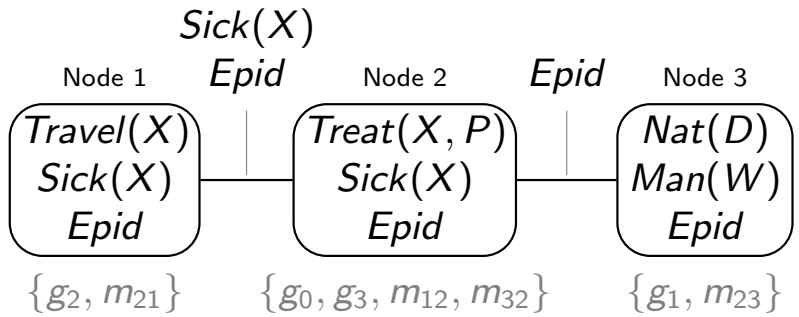
LJT answers queries

- Node where query term appears
- Node 1: g_2, m_{12}
  - Eliminate $Travel(X)$ from g_2
  - Eliminate $Sick(X)$, $X \neq eve$, from $g_2' \cdot m_{21}$
  - Eliminate $Epid$ from g_{23}'

Marginal Distribution Ground Queries

$P(Sick(eve)), P(Epid), \ldots$





QA: LVE eliminates

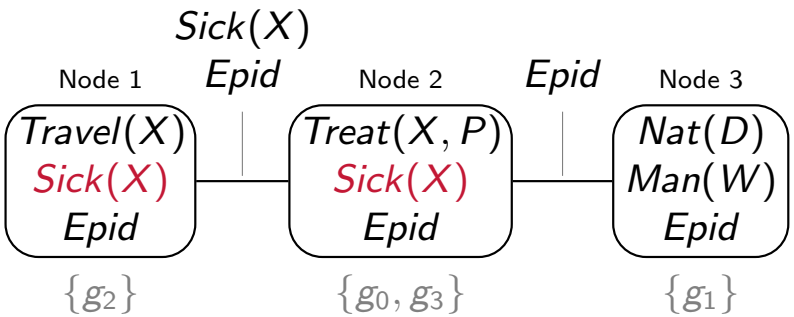
- $Treat(X, P)$ from g_3
- $Travel(X)$ from g_2
- $Sick(X), X \neq eve$ from $g_2' \cdot g_3'$
- $Nat(D)$, $Man(W)$
- $Epid$

LJT answers queries

- Node where query term appears
- Node 1: g_2, m_{12}
  - Eliminate $Travel(X)$ from g_2
  - Eliminate $Sick(X)$, $X \neq eve$, from $g_2' \cdot m_{21}$
  - Eliminate $Epid$ from g_{23}'

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$

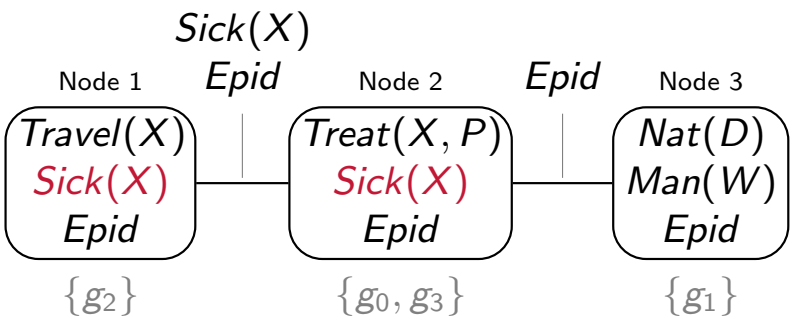


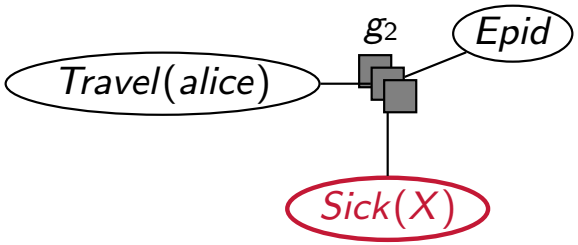
Evidence: LJT

- Evidence at each node

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$





Node 1

Evidence: LJT

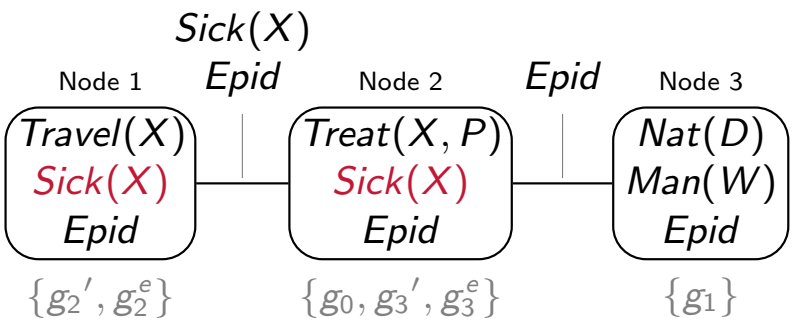
- Evidence at each node

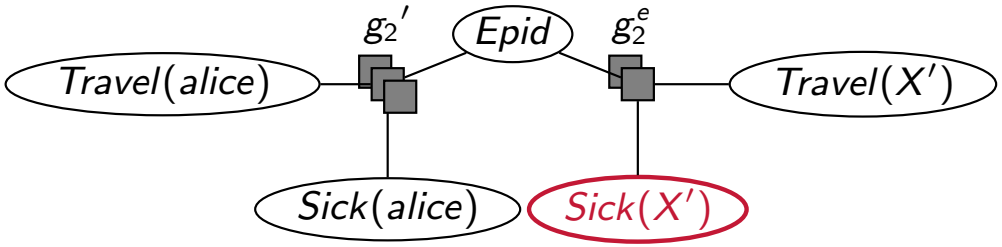
Evidence at node: use LVE

- Each factor containing $Sick(X)$

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$





Node 1

Evidence: LJT

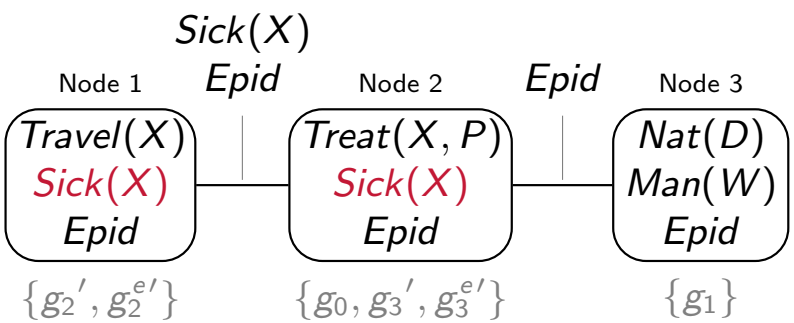
- Evidence at each node

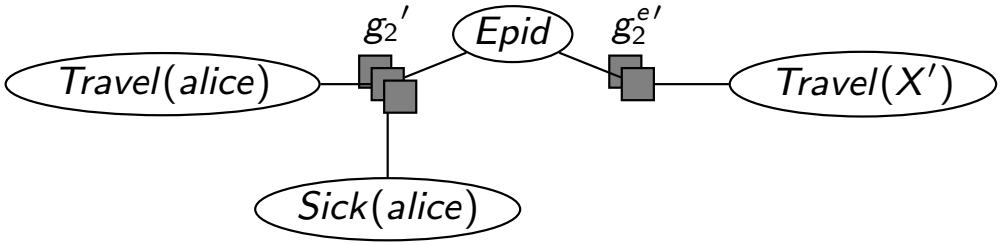
Evidence at node: use LVE

- Each factor containing $Sick(X)$
  - Splits X w.r.t. $X \in \{eve, bob\}$

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$





Node 1

Evidence: LJT

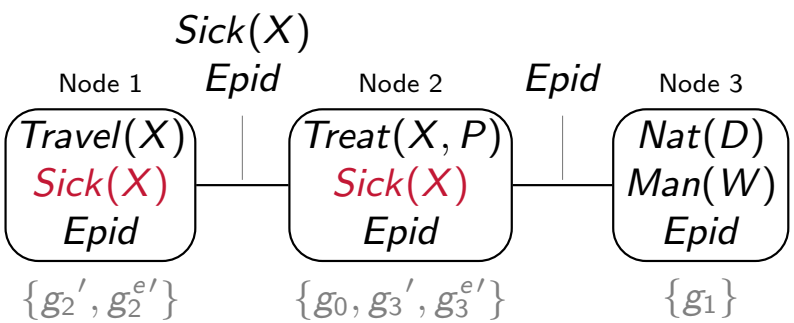
- Evidence at each node

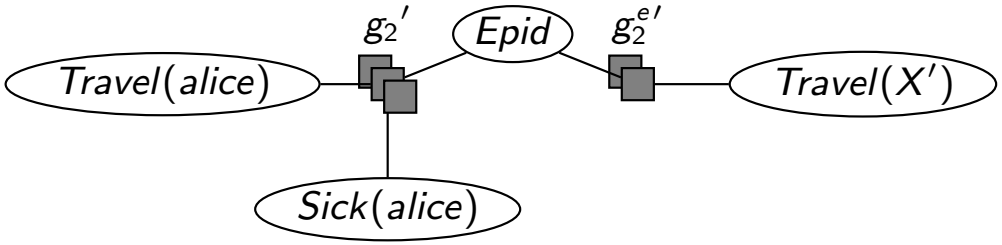
Evidence at node: use LVE

- Each factor containing $Sick(X)$
  - Splits X w.r.t. $X \in \{eve, bob\}$
  - Absorbs $Sick(X) = true$, $X \in \{eve, bob\}$

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$





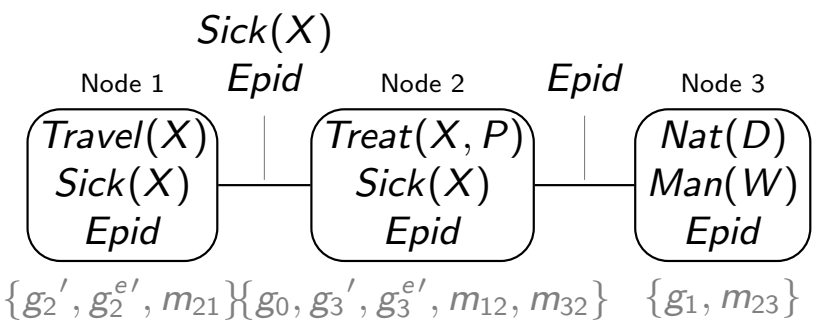
Node 1

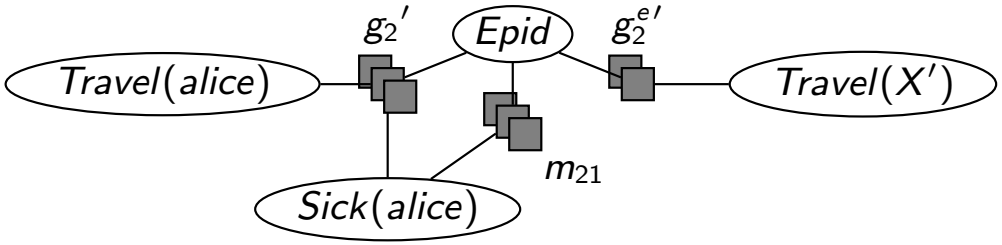
Evidence: LJT

- Evidence at each node
- Continue with LJT

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$





Node 1

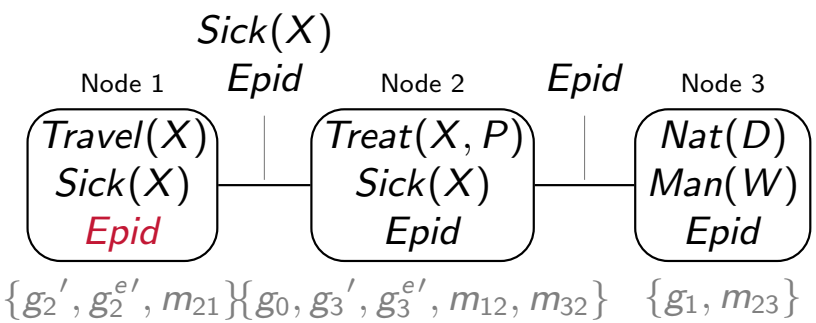
Evidence: LJT

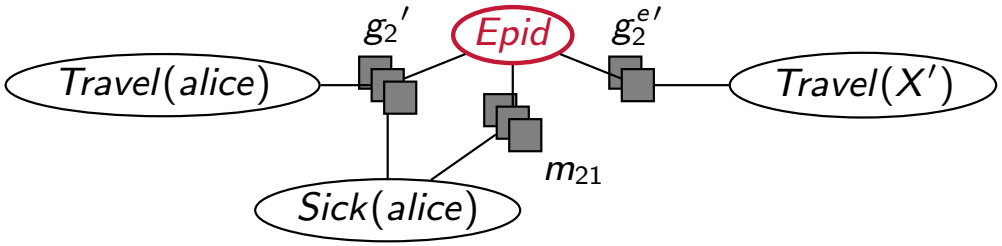
- Evidence at each node
- Continue with LJT

  → Pass messages

Conditional Distribution Ground Queries

$P(Epid|Sick(eve) = true, Sick(bob) = true)$





Node 1

Evidence: LJT

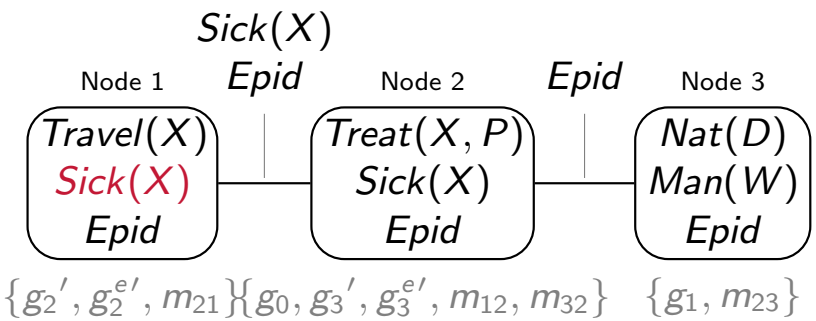
- Evidence at each node
- Continue with LJT

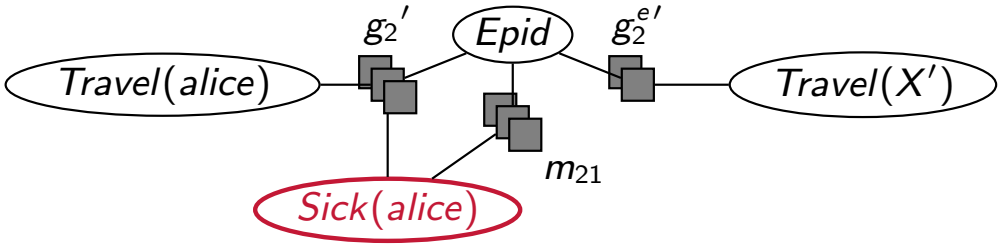
  → Pass messages

  → Answer queries $P(Epid)$

Conditional Distribution Ground Queries

$P(Sick(alice)|Sick(eve) = true, Sick(bob) = true), \ldots$





Node 1

Evidence: LJT

- Evidence at each node
- Continue with LJT

  → Pass messages

  → Answer queries

  $P(Epid)$

  $P(Sick(alice))$

  $\ldots$

Summary: LJT

Algorithm steps

1. Construction
2. Evidence entering
3. Message passing
4. Query answering

Reasoning Algorithm

Static overhead

$\rightarrow$ Construction, message passing

Efficient query answering

$\rightarrow$ QA on smaller models

Query language expressiveness

$\rightarrow$ Marginal, conditional distributions

$\rightarrow$ Likelihood of observations

Summary: LJT

Algorithm steps

1. Construction
2. Evidence entering
3. Message passing
4. Query answering

Reasoning Algorithm

Static overhead

→ Construction, message passing

Efficient query answering

→ QA on smaller models

Query language expressiveness

→ Marginal, conditional distributions

→ Likelihood of observations

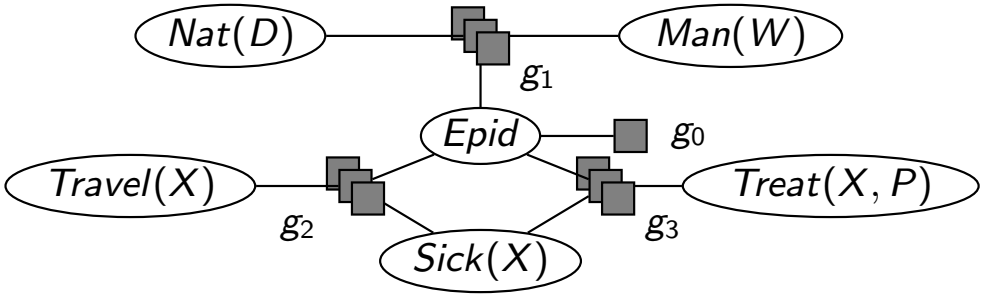
→ Most probable explanation?

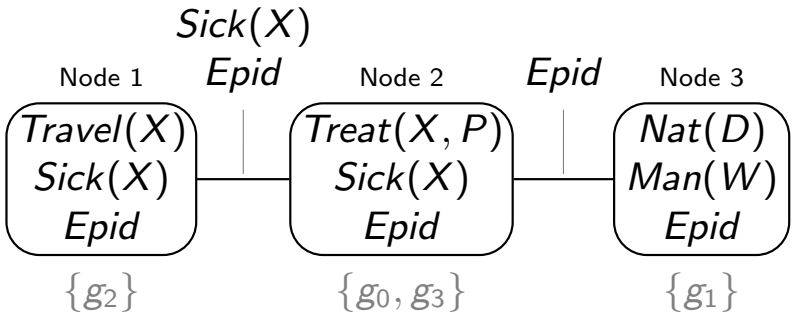
Conference Contribution

Extensions to LJT [Braun & Möller 2016]

- Expressivity extension for query language
  - Most probable explanation (MPE)
  - Maximum a posteriori assignment (MAP)
- Adapted LJT for queries of different types
  - Assignment queries: MPE, MAP
  - Probability queries: Marginal, Conditional, Likelihood

Assignment Queries: MPE





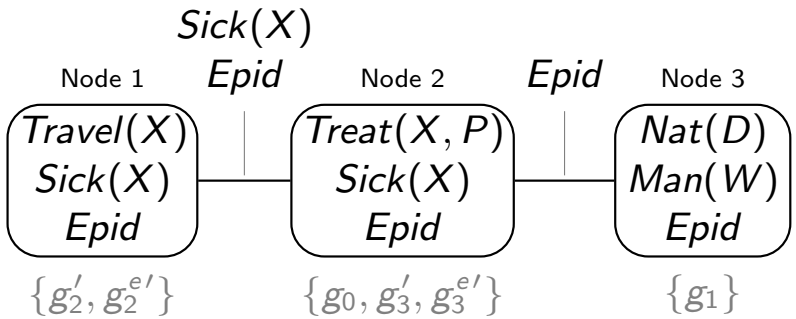
Most probable assignment in a model G given evidence $\mathbf{E}$

$$\arg\max_{\mathbf{V}} P(\mathbf{V}|\mathbf{E})$$

- $\mathbf{V}$ = set of all random variables in G without evidence
- LVE: replace $\sum$ with arg max for elimination (LVE^{MPE})
- LJT: use LVE^{MPE} during message calculation (LJT^{MPE})

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$

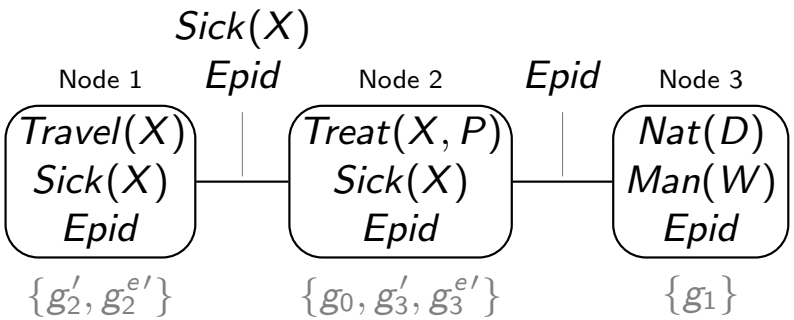


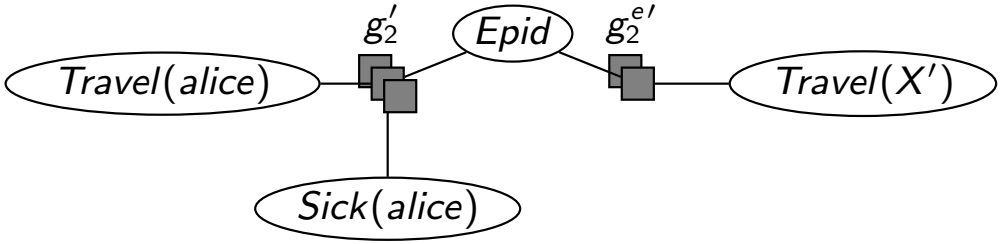
LJT$^{\text{MPE}}$

- Absorb evidence at nodes

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V} | Sick(eve) = true, Sick(bob) = true)$





Node 1

LJT$^{\text{MPE}}$

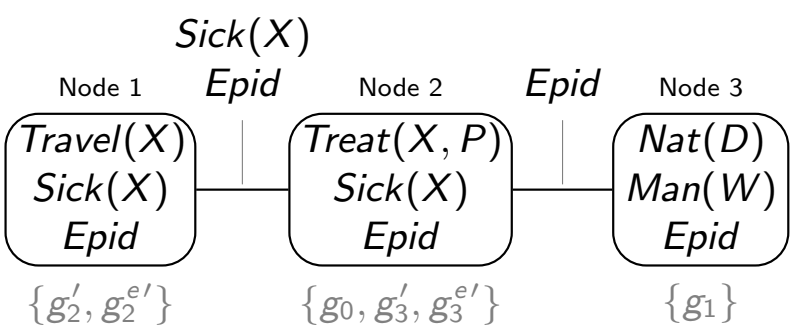
- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$

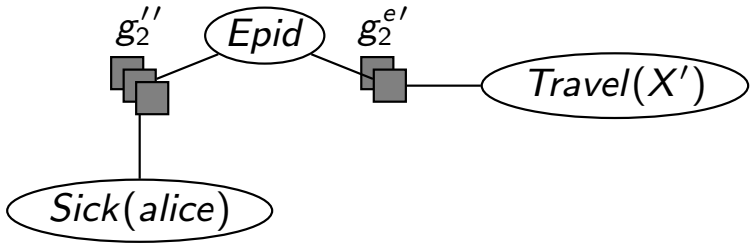
Pass message

- Eliminate using LVE$^{\text{MPE}}$
- Depending on $Epid, Sick(X)$, choose most probable assignment

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$





Node 1

LJT$^{\text{MPE}}$

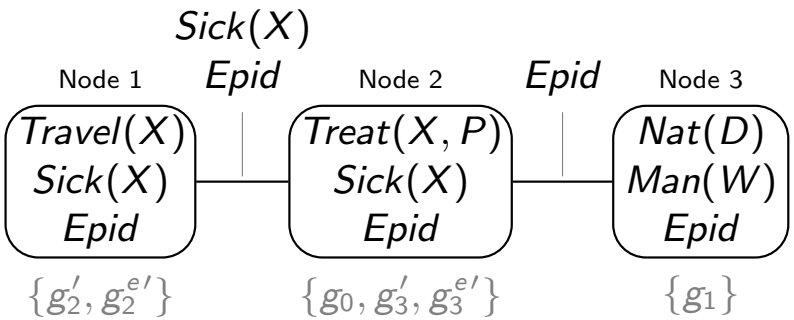
- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$

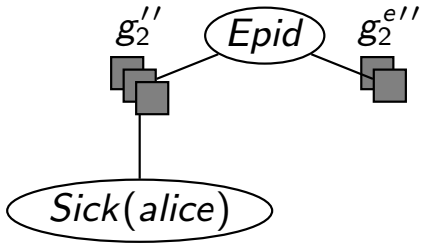
Pass message

- Eliminate using LVE$^{\text{MPE}}$
- Depending on $Epid, Sick(X)$, choose most probable assignment
  - $Travel(alice)$

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$





Node 1

LJT$^{\text{MPE}}$

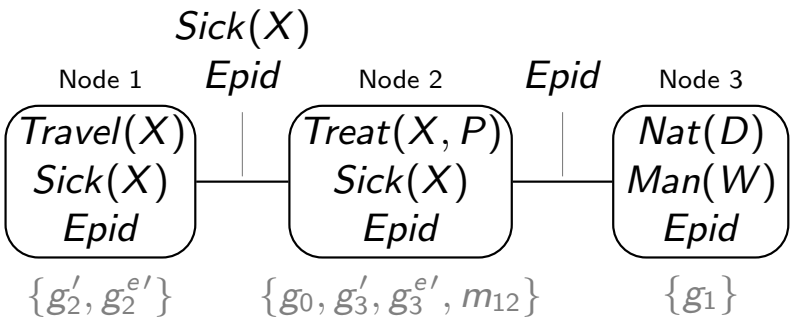
- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$

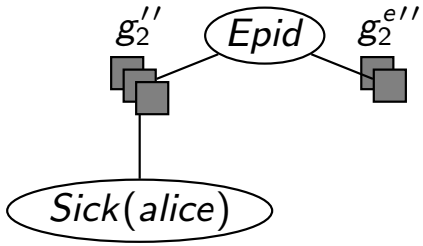
Pass message

- Eliminate using LVE$^{\text{MPE}}$
- Depending on $Epid, Sick(X)$, choose most probable assignment
  - $Travel(alice)$
  - $Travel(X')$

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$





Node 1

LJT$^{\text{MPE}}$

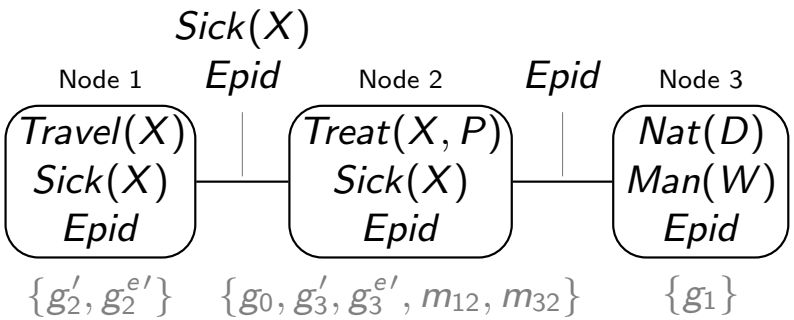
- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$

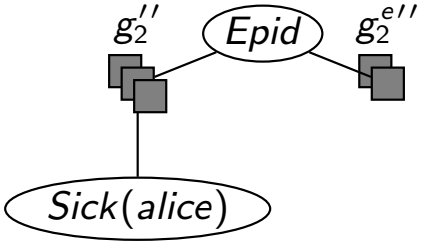
Pass message

- Eliminate using LVE$^{\text{MPE}}$
- Depending on $Epid, Sick(X)$, choose most probable assignment
  - $Travel(alice)$
  - $Travel(X')$
- Send information to node 2

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$





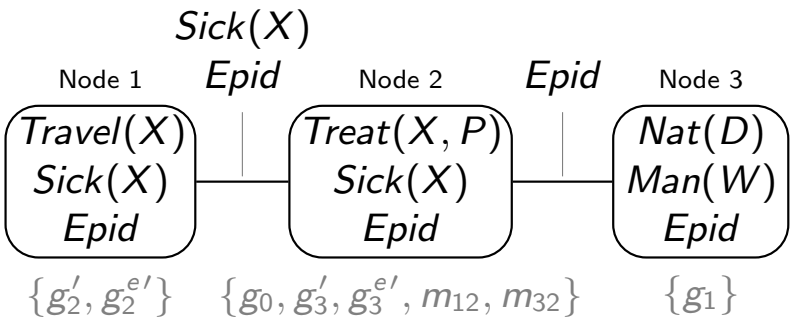
Node 1

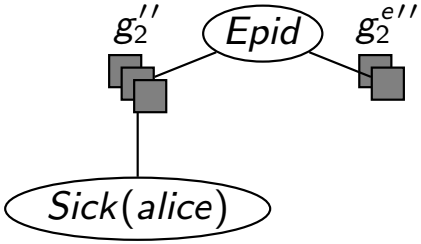
LJT$^{\text{MPE}}$

- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$
  - $3 \longrightarrow 2$

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$





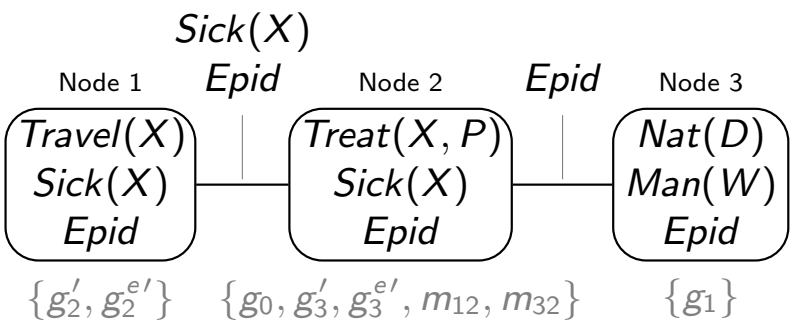
Node 1

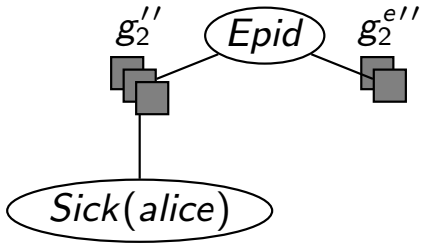
LJT$^{\text{MPE}}$

- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$
  - $3 \longrightarrow 2$
- (Answer query)
  - Eliminate variables

Assignment Queries: MPE

$\arg\max_{\mathbf{V}} P(\mathbf{V}|Sick(eve) = true, Sick(bob) = true)$





Node 1

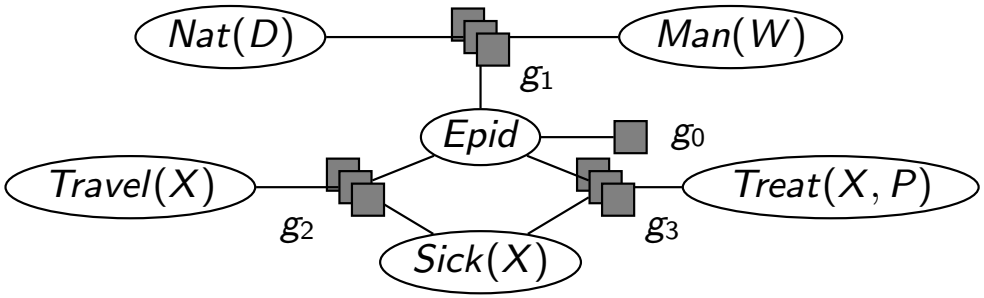
LJT$^{\text{MPE}}$

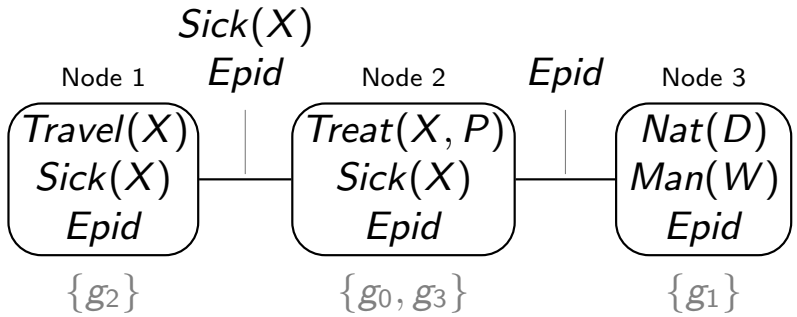
- Absorb evidence at nodes
- Pass messages (inward)
  - $1 \longrightarrow 2$
  - $3 \longrightarrow 2$
- (Answer query)
  - Eliminate variables
  - Output MPE

MPE

- $Epid = false$
- $\forall X' \in \{eve, bob\}, P:$
  $Treat(X', P) = true,\ Travel(X') = false$
- $Sick(alice) = false,\ Travel(alice) = true,$
  $\forall P : Treat(alice, P) = false$
- $\forall D : Nat(D) = false$
- $\forall W : Man(W) = false$

Assignment Queries: MAP





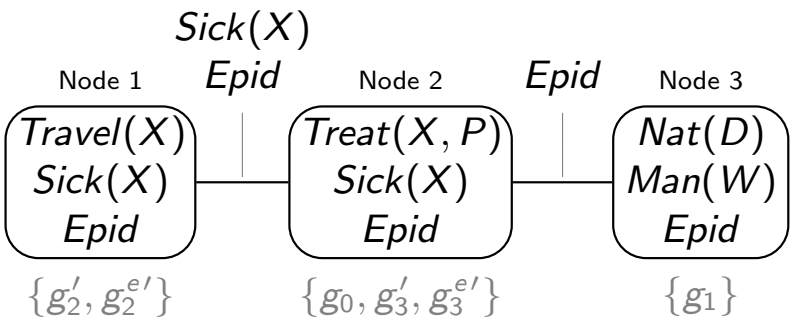
Most probable assignment in a model G given evidence $\mathbf{E}$

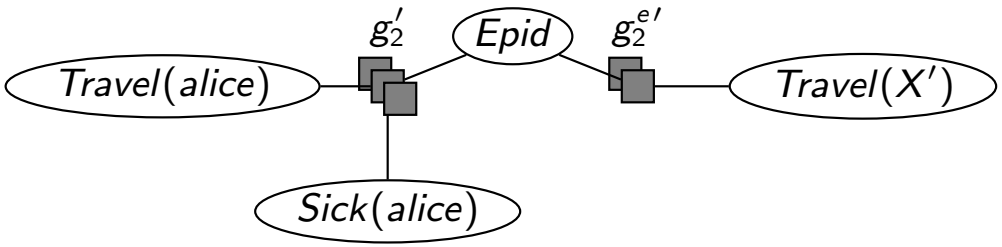
$$\arg\max_{\mathbf{V}} \sum_{\mathbf{T}} P(\mathbf{V}|\mathbf{E})$$

- $\mathbf{V}$ = set of random variables in G without evidence
- $\mathbf{T}$ = set of remaining variables in G without evidence
- $\mathbf{T} = \emptyset$: MPE
- MAP queries hard: $\sum$ and arg max not commutative
  - $\rightarrow$ May prohibit reasonable elimination order

Assignment Queries: MAP

$\arg\max_{Travel(X')} \sum_{\mathbf{T}} P(Travel(X')|Sick(X') = true), X' \in \{eve, bob\}$





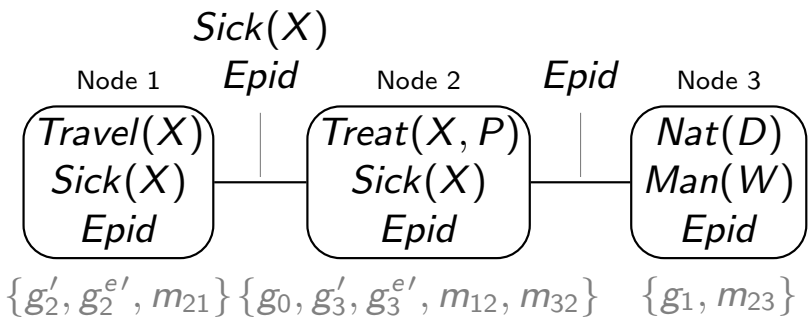
Node 1

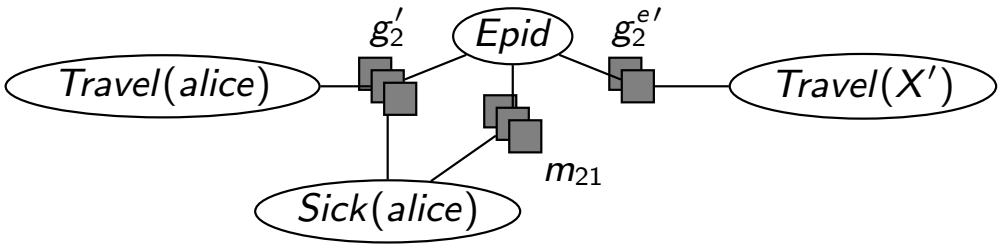
LJT$^{\text{MAP}}$

- Absorb evidence at nodes

Assignment Queries: MAP

$\arg\max_{Travel(X')} \sum_{\mathbf{T}} P(Travel(X')|Sick(X') = true), X' \in \{eve, bob\}$





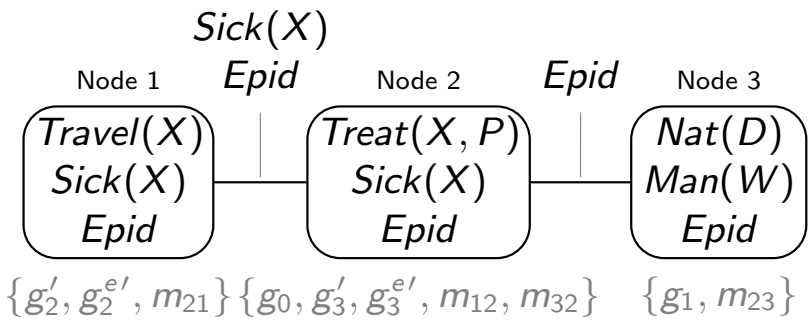
Node 1

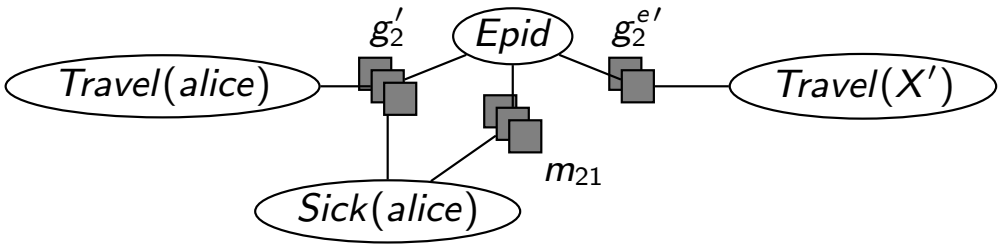
LJT$^{\text{MAP}}$

- Absorb evidence at nodes
- Pass messages with LVE

Assignment Queries: MAP

$\arg\max_{Travel(X')} \sum_{\mathbf{T}} P(Travel(X')|Sick(X') = true), X' \in \{eve, bob\}$





Node 1

LJT$^{\text{MAP}}$

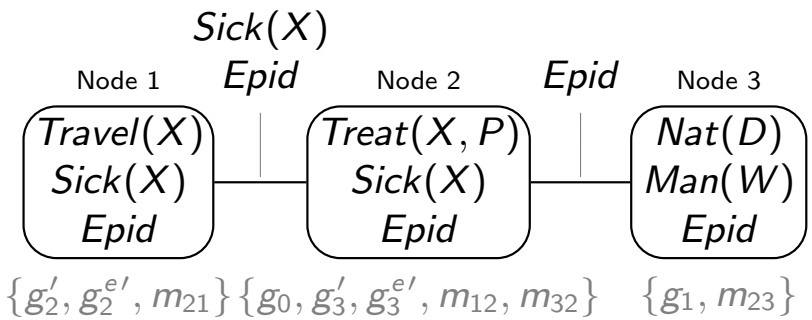
- Absorb evidence at nodes
- Pass messages with LVE
- Answer query
  - Find nodes covering **V**
  - Eliminate remaining variables of **T**
  - Eliminate **V**

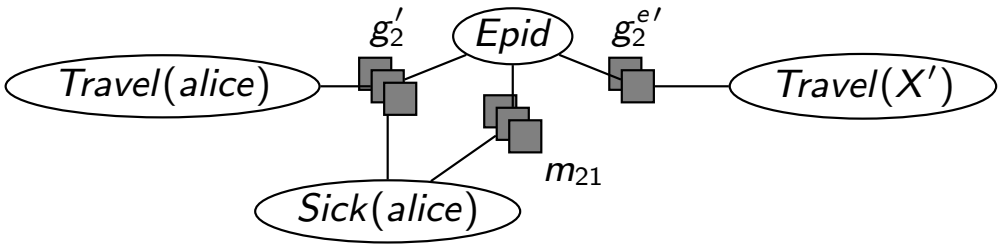
Answer query

- $Travel(eve)$, $Travel(bob)$: node 1

Assignment Queries: MAP

$\arg\max_{Travel(X')} \sum_{\mathbf{T}} P(Travel(X')|Sick(X') = true), X' \in \{eve, bob\}$





Node 1

LJT$^{\text{MAP}}$

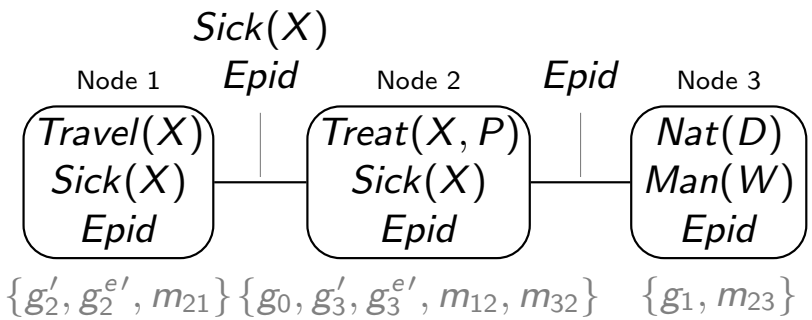
- Absorb evidence at nodes
- Pass messages with LVE
- Answer query
  - Find nodes covering **V**
  - Eliminate remaining variables of **T**
  - Eliminate **V**

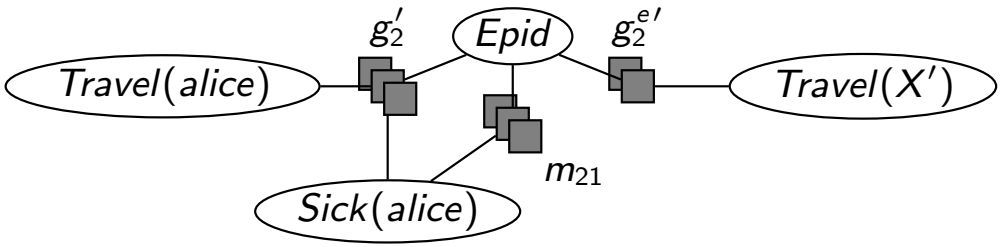
Answer query

- $Travel(eve)$, $Travel(bob)$: node 1
- Eliminate with LVE: $Travel(alice), Sick(alice), Epid$

Assignment Queries: MAP

$\arg\max_{Travel(X')} \sum_{\mathbf{T}} P(Travel(X')|Sick(X') = true), X' \in \{eve, bob\}$





Node 1

LJT$^{\text{MAP}}$

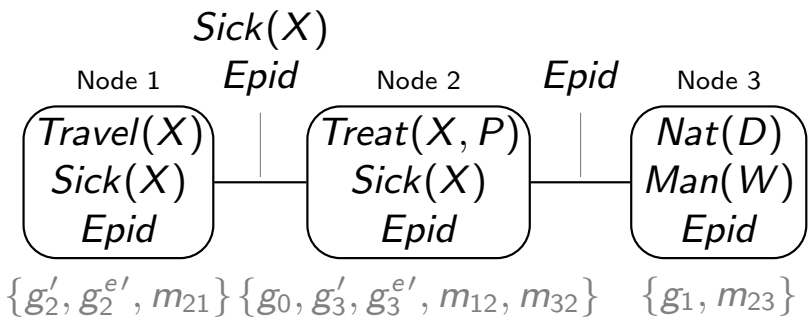
- Absorb evidence at nodes
- Pass messages with LVE
- Answer query
  - Find nodes covering **V**
  - Eliminate remaining variables of **T**
  - Eliminate **V**

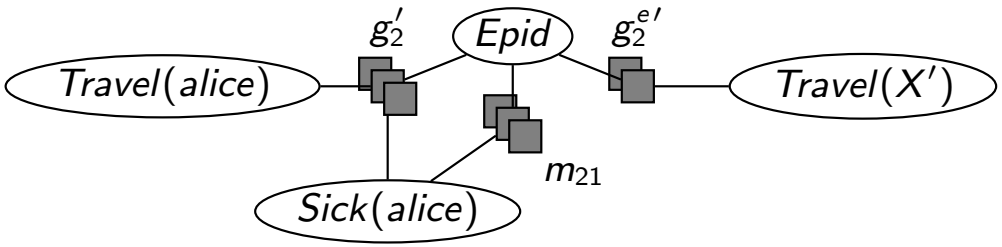
Answer query

- $Travel(eve)$, $Travel(bob)$: node 1
- Eliminate with LVE: $Travel(alice), Sick(alice), Epid$
- Eliminate with LVE$^{\text{MPE}}$: $Travel(X')$

Assignment Queries: MAP

$\arg\max_{Travel(X')} \sum_{\mathbf{T}} P(Travel(X') | Sick(X') = true), X' \in \{eve, bob\}$





Node 1

LJT$^{\text{MAP}}$

- Absorb evidence at nodes
- Pass messages with LVE
- Answer query
  - Find nodes covering **V**
  - Eliminate remaining variables of **T**
  - Eliminate **V**

MAP

- $\forall X' \in \{eve, bob\}, P :$ $Travel(X') = false$

Summary: Adapted LJT

Algorithm steps

1. Construction
2. Evidence entering
3. Message passing
4. Query answering

Reasoning Algorithm

Message passing: Elimination

$\rightarrow$ MPE: arg max (only inward)

$\rightarrow$ All other: $\sum$

Query answering: Elimination

- Probability queries: $\sum_{\text{non-query terms}}$
- MPE: arg max at innermost node
- MAP: $\sum_{\text{non-query terms}}$, $\arg\max_{\text{query terms}}$

Conference Contribution

Extensions to LJT [Braun & Möller 2016]

- Expressivity extension for query language
  - Most probable explanation (MPE)
  - Maximum a posteriori assignment (MAP)
- Adapted LJT for queries of different types
  - Assignment queries: MPE, MAP
  - Probability queries: Marginal, Conditional, Likelihood

Conference Contribution

Extensions to LJT [Braun & Möller 2016]

- Expressivity extension for query language
  - Most probable explanation (MPE)
  - Maximum a posteriori assignment (MAP)
- Adapted LJT for queries of different types
  - Assignment queries: MPE, MAP
  - Probability queries: Marginal, Conditional, Likelihood

Current work

- Incrementally changing models
- Dynamic variant