# StarAI

## Exact Symmetries and Changing Domains in Static PRMs

Tutorial ECAI 2020

Tanya Braun, Marcel Gehrke, Ralf Möller
Universität zu Lübeck

UNIVERSITÄT ZU LÜBECK

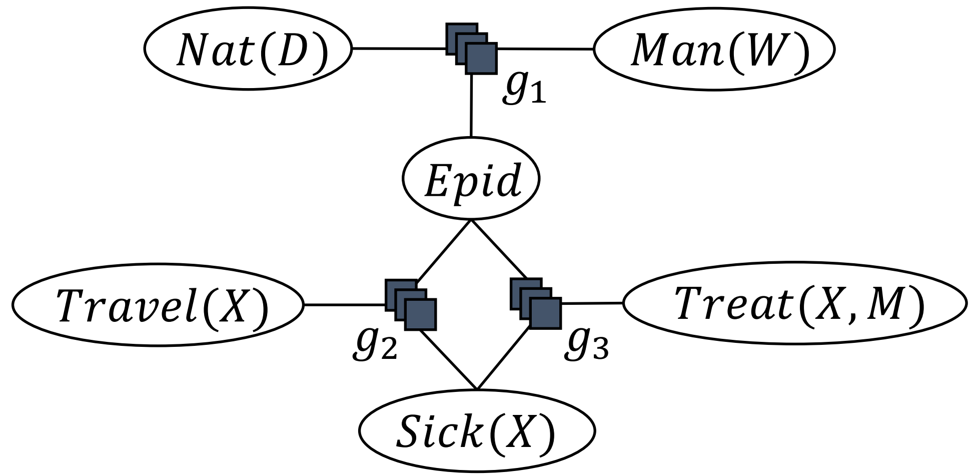
# Agenda

- Probabilistic relational models (PRMs) [Ralf]
- Exact symmetries and changing domains in static PRMs [Tanya]
  - Exact symmetries
  - Changing domains
  - Unknown domains
- Stable inference over time in dynamic PRMs [Marcel]
- Summary [Tanya]

Goal: Overview of central ideas

# Semantics of a PRM

- Joint probability distribution $P_G$ by grounding

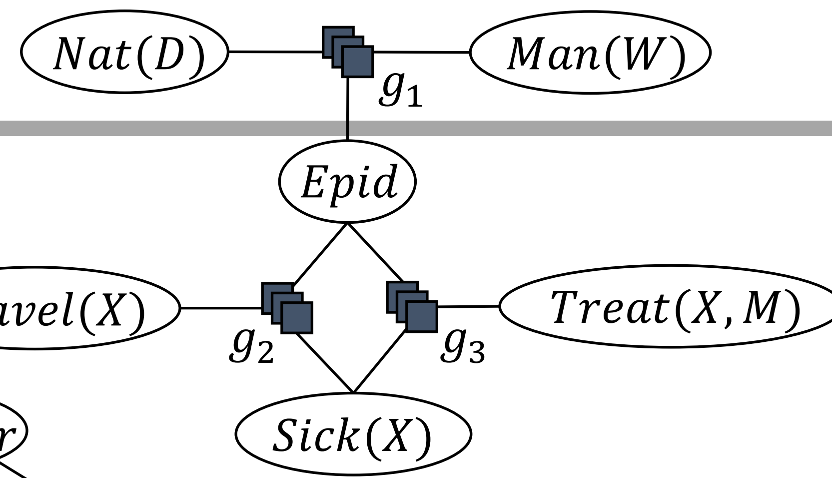



$$P_G = \frac{1}{Z} \prod_{f \in gr(G)} f$$

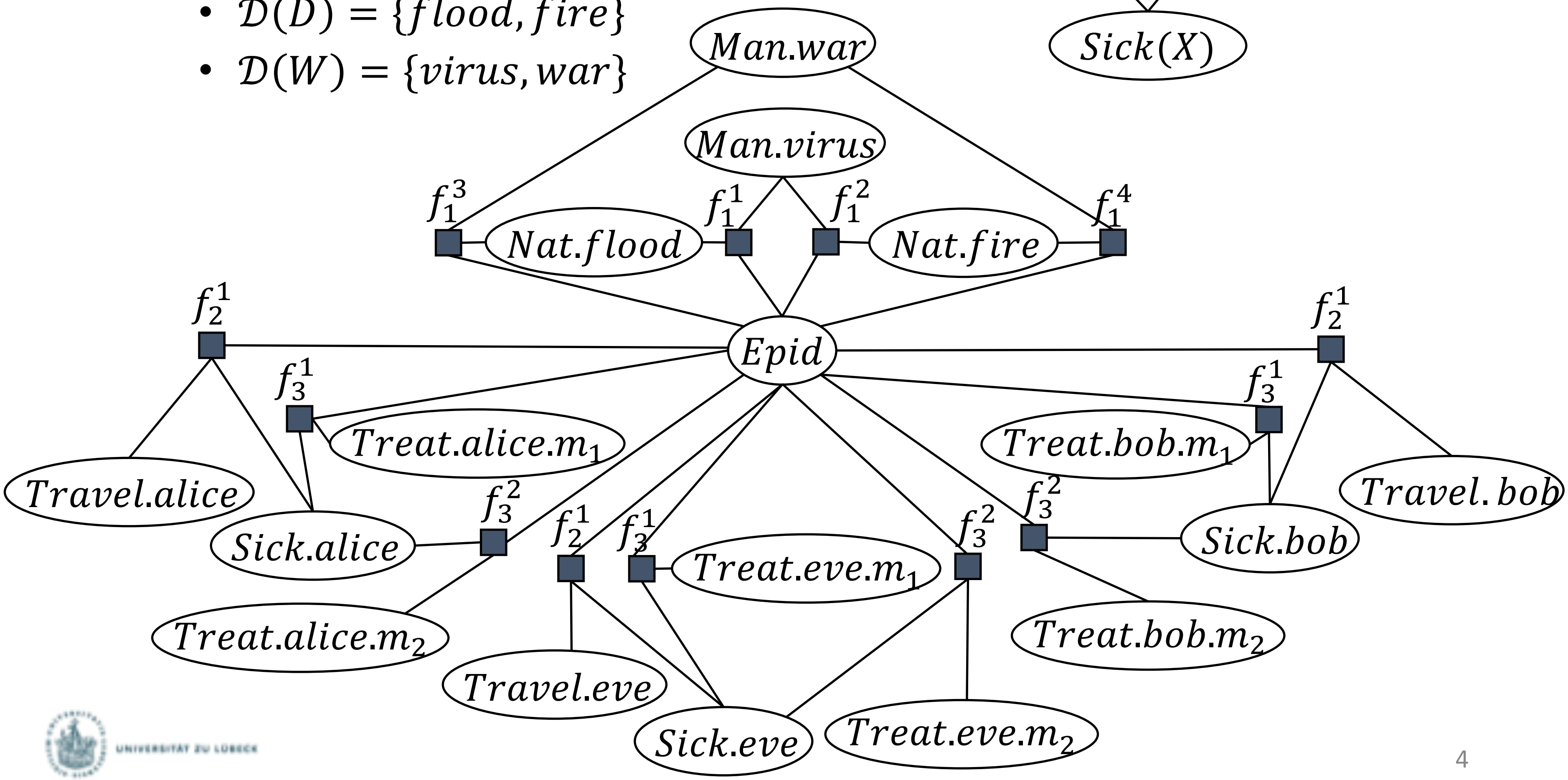
$$Z = \sum_{v \in r(rv(gr(G)))} \prod_{f \in gr(G)} f_i(\pi_{rv(f_i)}(v))$$

$\pi_{variables}(v)$ = projection of $v$ onto *variables*

# Grounded Model




- Given domains
  - $\mathcal{D}(X) = \{alice, eve, bob\}$
  - $\mathcal{D}(M) = \{m_1, m_2\}$
  - $\mathcal{D}(D) = \{flood, fire\}$
  - $\mathcal{D}(W) = \{virus, war\}$

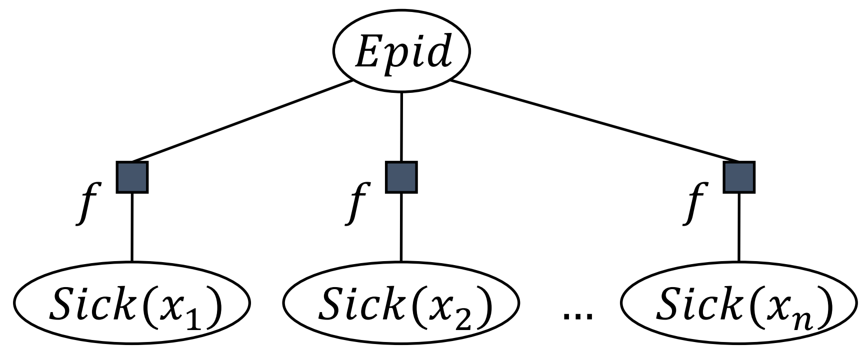


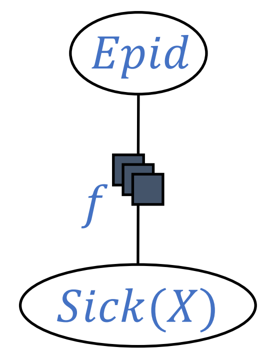
# Complexity, Completeness & Tractability

# Grounding Semantics

- Equivalence of lifted and grounded calculations
  - Never worse than propositional inference

$$P(Epid)$$
$$\propto \sum_{s \in r(Sick(x_1))} f(Epid, Sick(x_1) = s)$$
$$\cdot \sum_{s \in r(Sick(x_2))} f(Epid, Sick(x_2) = s)$$
$$\cdot \cdots$$
$$\cdot \sum_{s \in r(Sick(x_2))} f(Epid, Sick(x_2) = s)$$
$$= \underbrace{f' \cdot f' \cdot \cdots \cdot f'}_{n \text{ times}} = (f')^n$$




$$P(Epid) \propto \left( \sum_{s \in r(Sick(X))} f(Epid, Sick(x) = s) \right)^n = (f')^n$$

# Complexity

Van den Broeck (2011)

- Query answering problem
  - Given a model, ask for probability distribution of a grounded PRV
- Given a model that allows for lifted calculations
  - I.e., no groundings during solving an instance of the problem
- Solving an instance of the problem is possible in time polynomial in domain sizes
  - No longer exponential in domain sizes

→ The query answering algorithm is domain-lifted

# Completeness

- No groundings in all possible models given some characteristic
  - Algorithm is *domain-lifted* in each possible model
- Model characteristics
  - Two logical variables per parfactor
$$g\big(A(X,Y),B(X,Y)\big)$$
$$g(A(X,Y),C(X),C(Y)),X \neq Y$$
$$g(A(X,Y),D(X),E(Y))$$
  - One logical variable per PRV (arbitrarily many logical variables per parfactor)
$$g\big(A(X),B(Y),C(Z)\big)$$
  - Holds for various domain-lifted algorithms, e.g.,
    - Lifted variable elimination Taghipour et al. (2013)
    - Lifted junction tree algorithm B (2020)
    - First-order knowledge compilation Van den Broeck (2011)
- Class of such models called liftable

# Completeness

- Models with other constellations may be computed without groundings but not all possible models
  - E.g., for lifted variable elimination, models with three logical variables

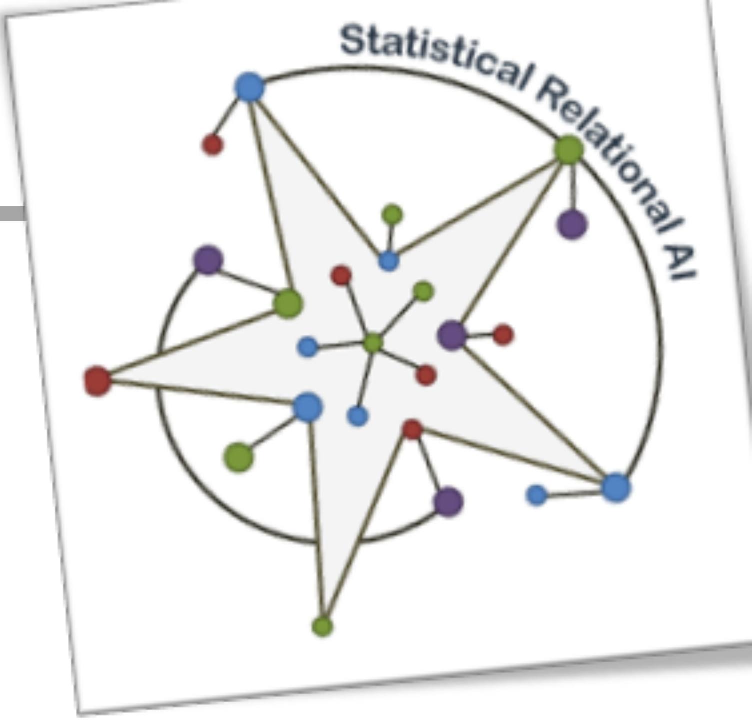
$$g\big(A(X,Y,Z),B(Y),C(Z)\big)\ \rightarrow\ liftable$$

$$g\big(A(X,Y),A(Y,Z),A(X,Z)\big)\ \rightarrow not\ liftable$$

→ Not complete for three logical variables per parfactor

# Tractability

- An query answering problem is tractable
  - when it is solved by an efficient algorithm, running in time polynomial in the number of random variables
- Assume that the number of random variables is characterised by domain sizes
  - Then, solving a query answering problem is tractable under domain-liftability
  - Runtime might still be exponential in other terms
- More general results by Niepert and Van den Broeck (2014)
  - Tractability through Exchangeability

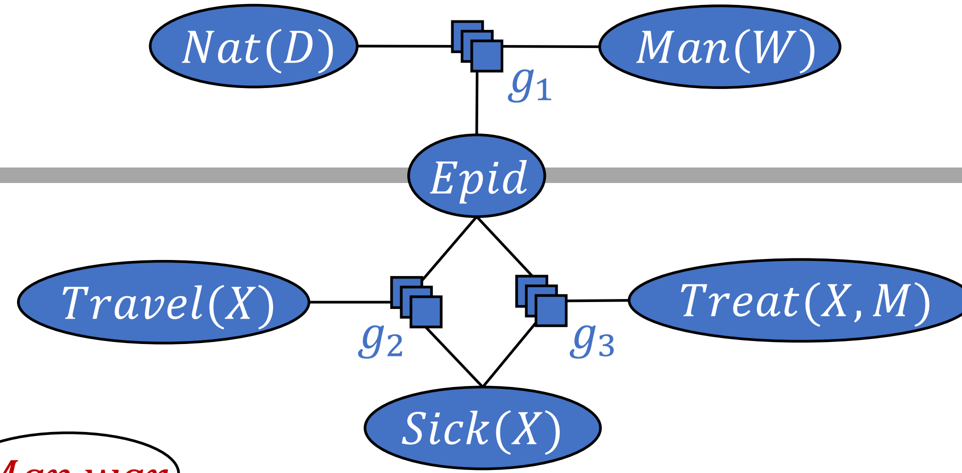



# From a Ground Model to a Lifted Model

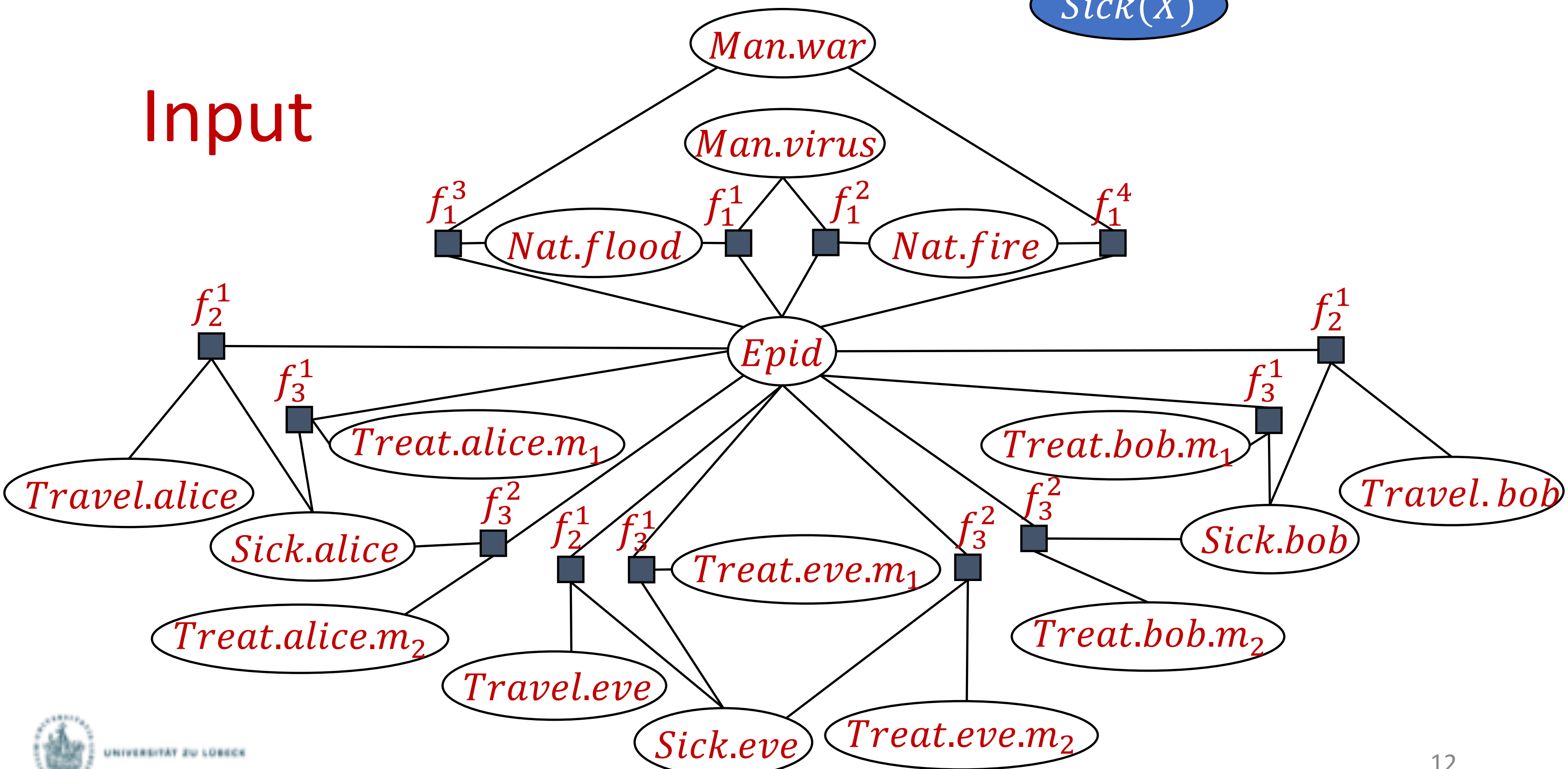
Using exact symmetries in the ground model

# Compression

## Goal



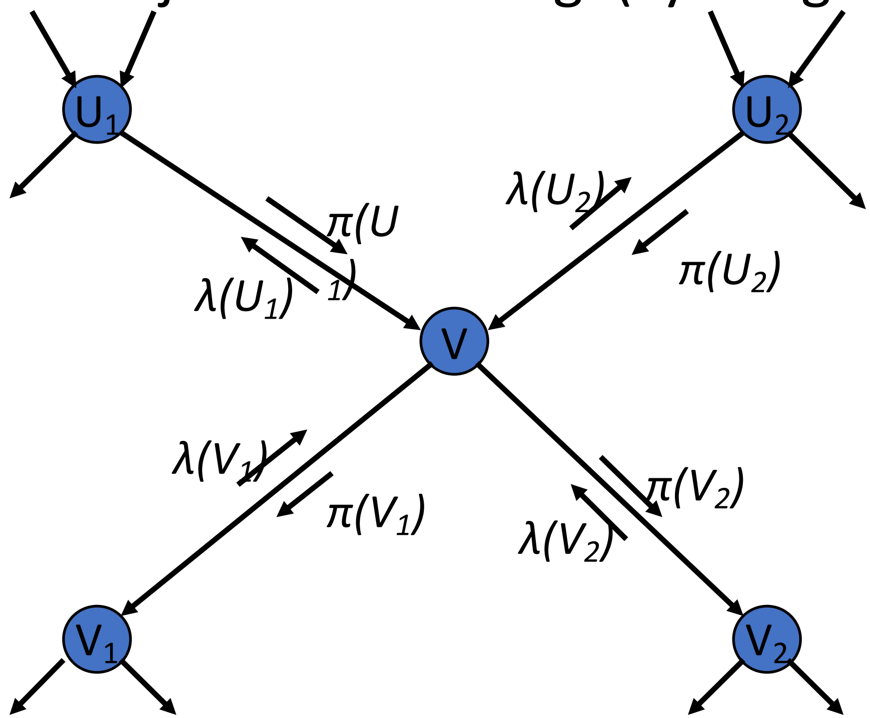

## Input



# A Bit of History…

Pearl (1982), Lauritzen and Spiegelhalter (1988)

- Pearl's Belief propagation
  - Messages on Bayes net
  - Exact for polytrees (no cycles in undirected graph!)
  - Precursor of junction tree alg. (cycles go into clusters)



# Loopy Belief Propagation

Singla and Domingos (2008), Kersting et al. (2009), Ahmadi et al. (2013)

- Pass messages on graph
  - If no cycles: exact
  - Else: approximate

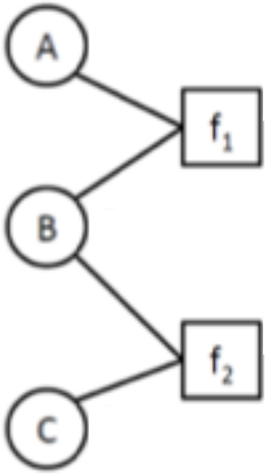
- Lifted (loopy) belief propagation
  - Exploit computational symmetries
  - Compress graph whenever nodes would send identical messages
  - Send messages on compressed graph

→ Colour passing algorithm for compression

# Compression: Pass the colours around*

Singla and Domingos (2008), Kersting et al. (2009), Ahmadi et al. (2013)

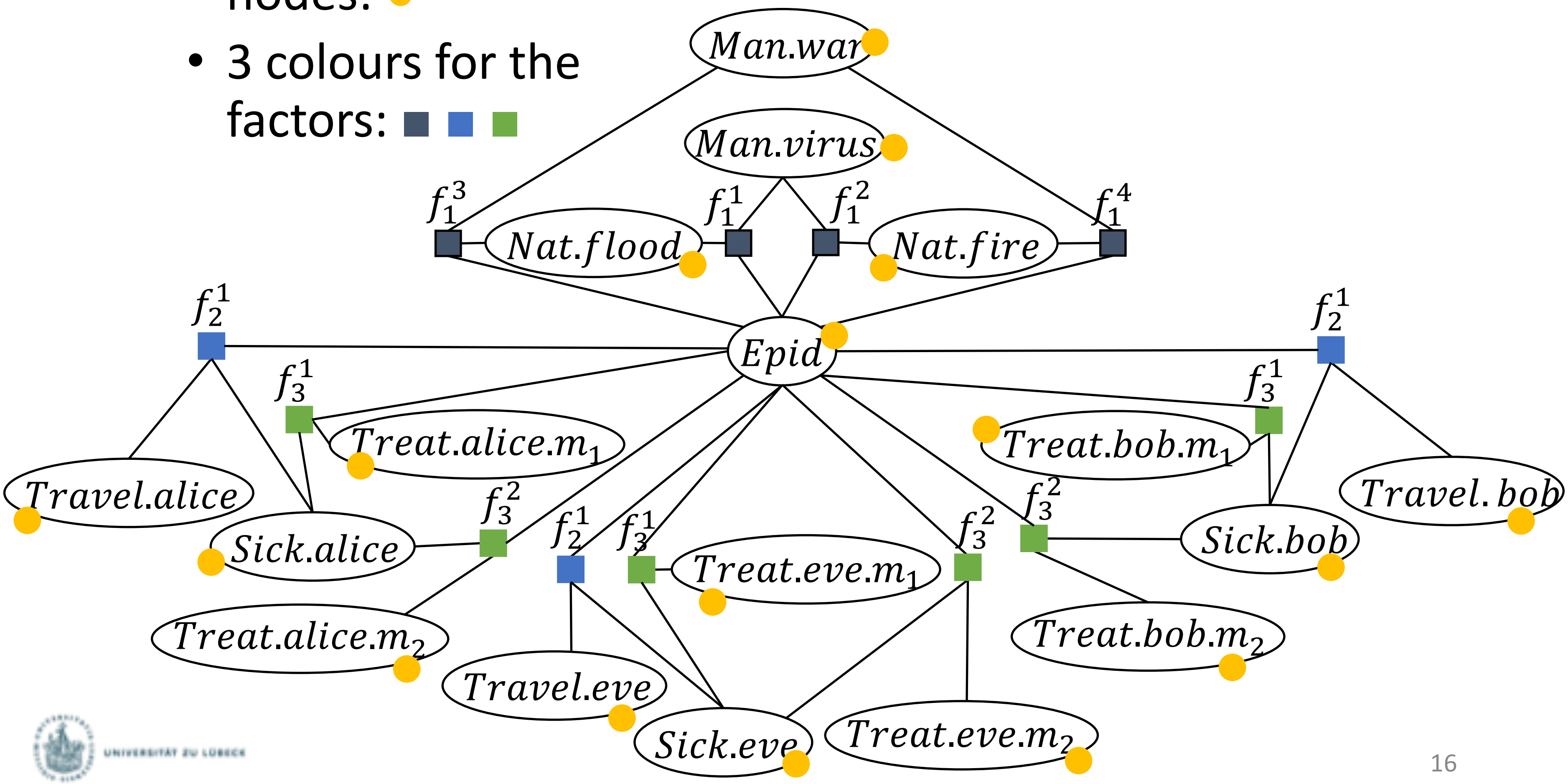
- **Colour nodes according to the evidence you have**
  - No evidence, say **red**
  - State „one“, say **brown**
  - State „two“, say **orange**
  - ...
- **Colour factors distinctively according to their equivalences**
  For instance, assuming $f_1$ and $f_2$ to be identical and B appears at the second position within both, say **blue**




*can also be done at the „lifted“, i.e., relational level

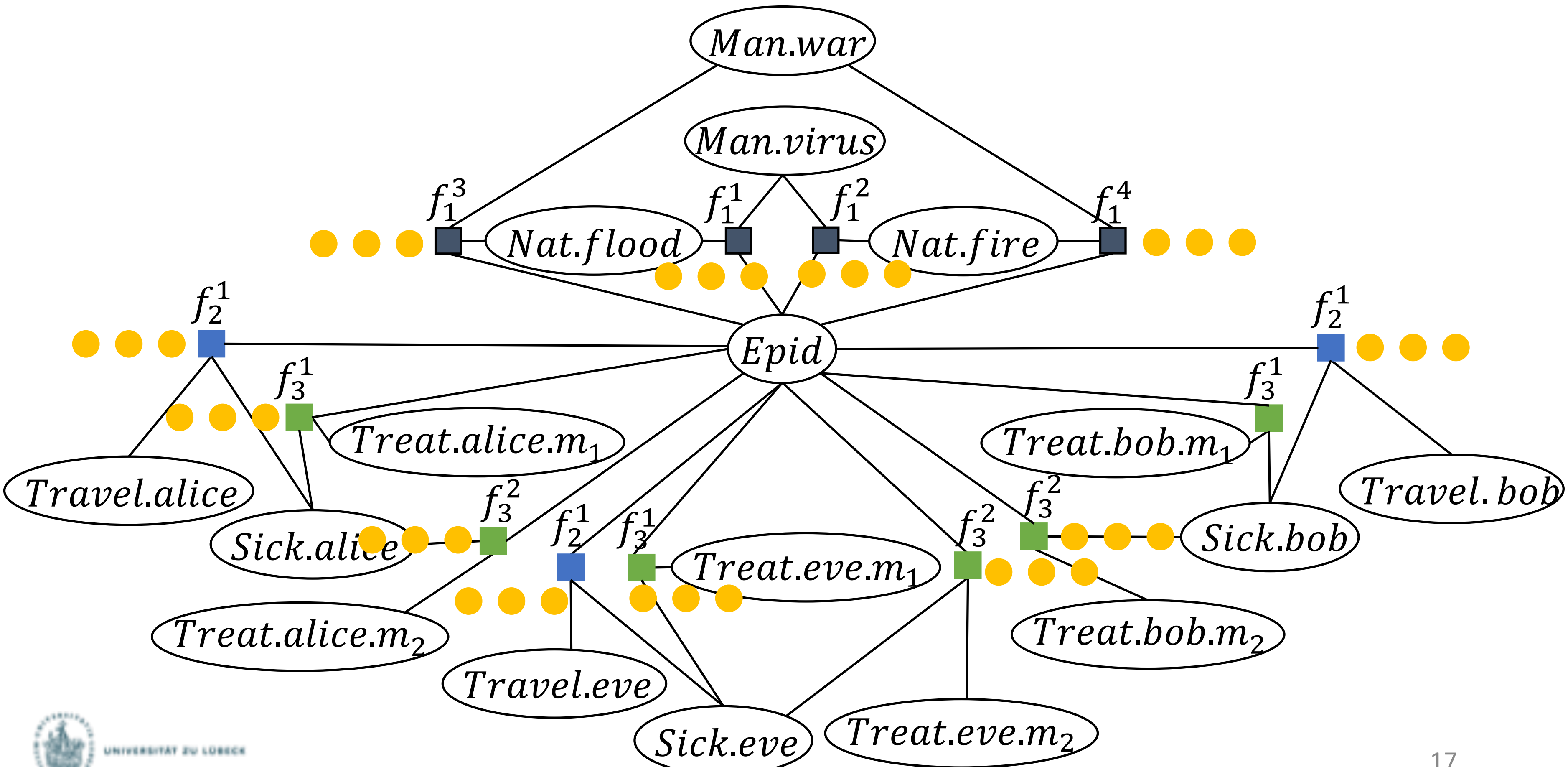
# Compression

1. Colour nodes and factors
   - 1 colour for the nodes:
   - 3 colours for the factors:



# Compression

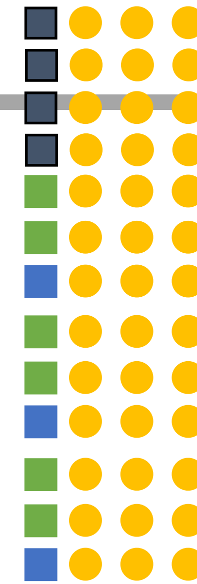
2. Factors collecting colours from nodes, signing their own colours to the collected ones

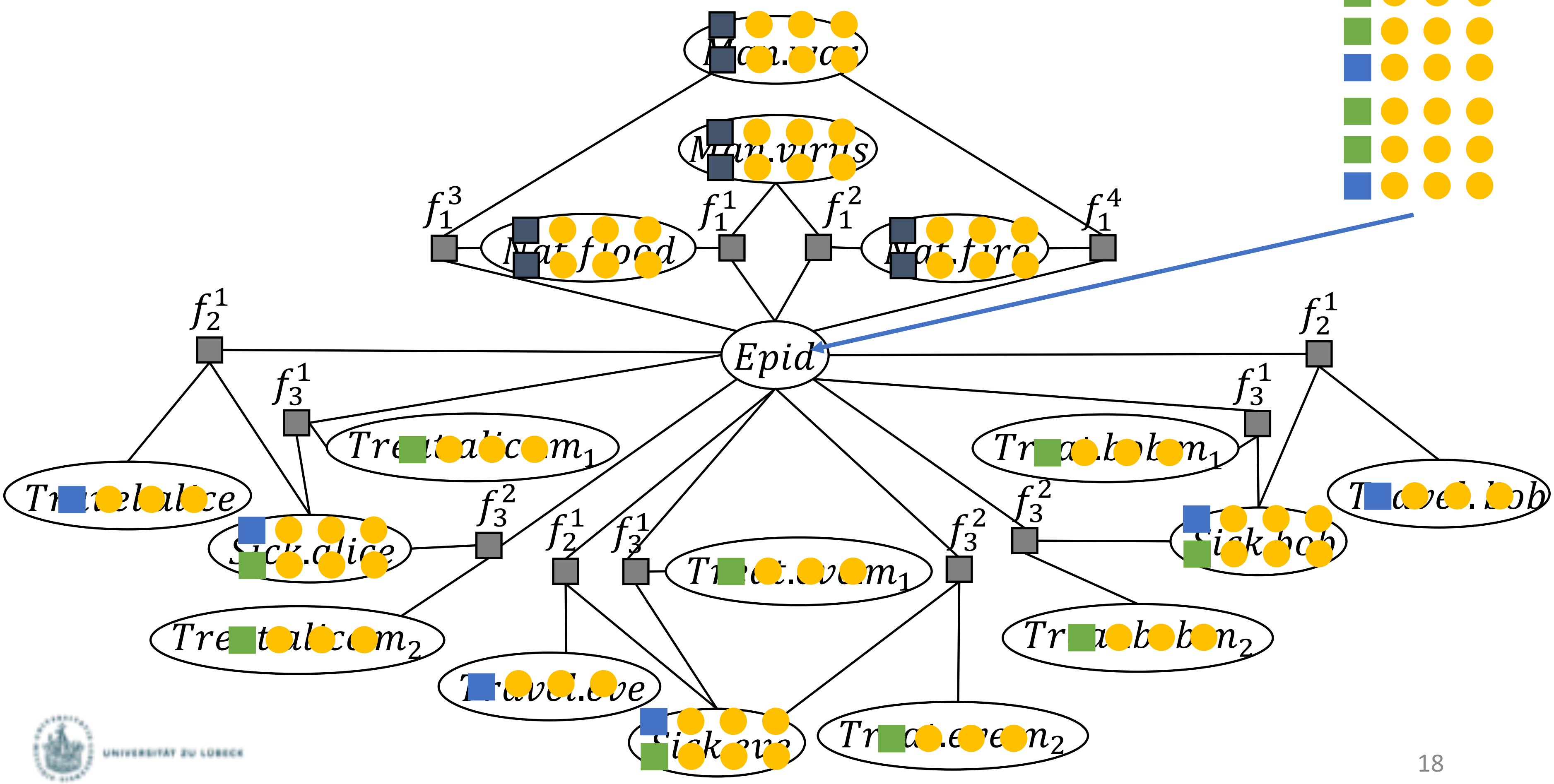



UNIVERSITÄT ZU LÜBECK

# Compression

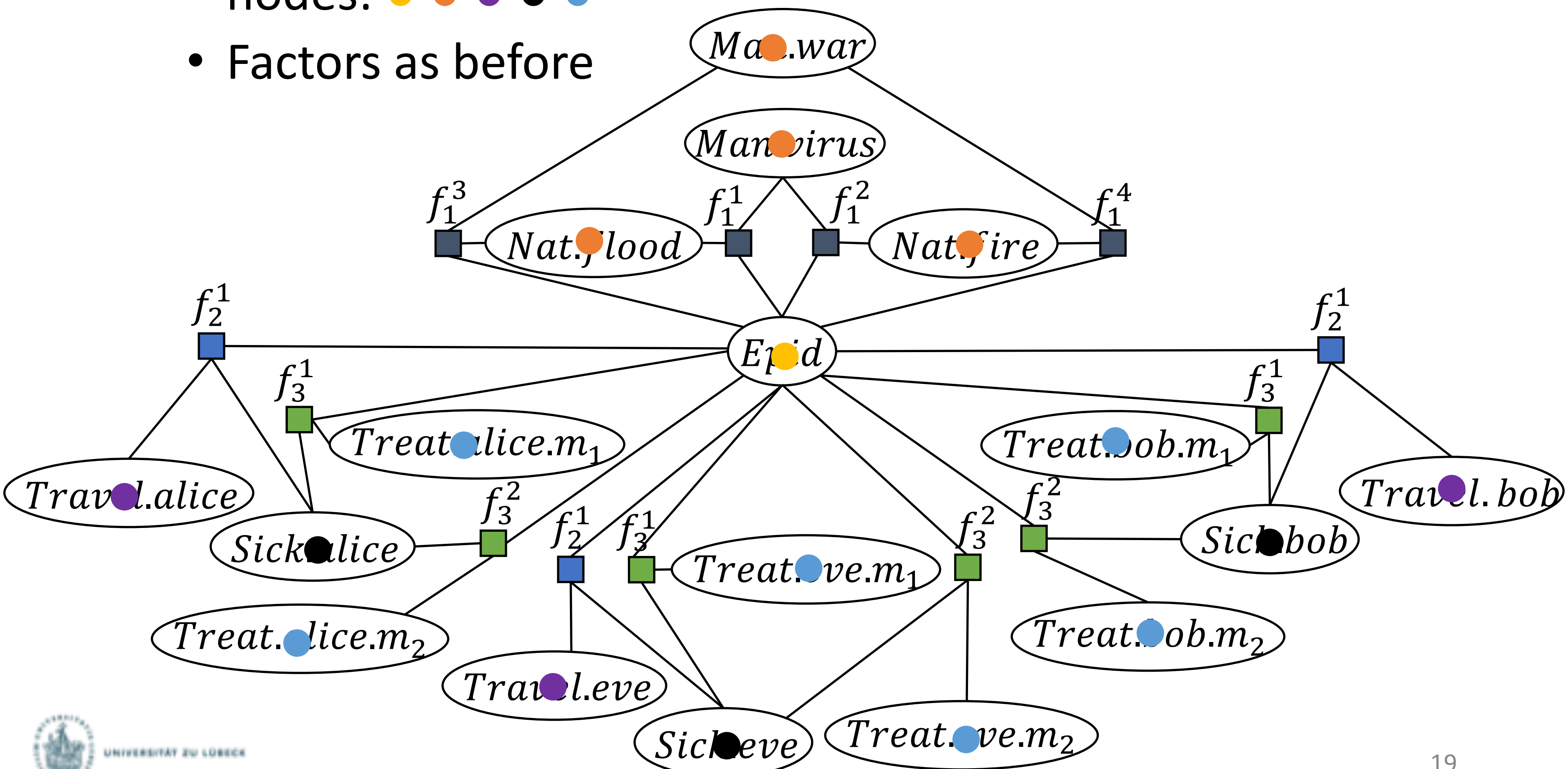
3. Nodes collecting colours from factors





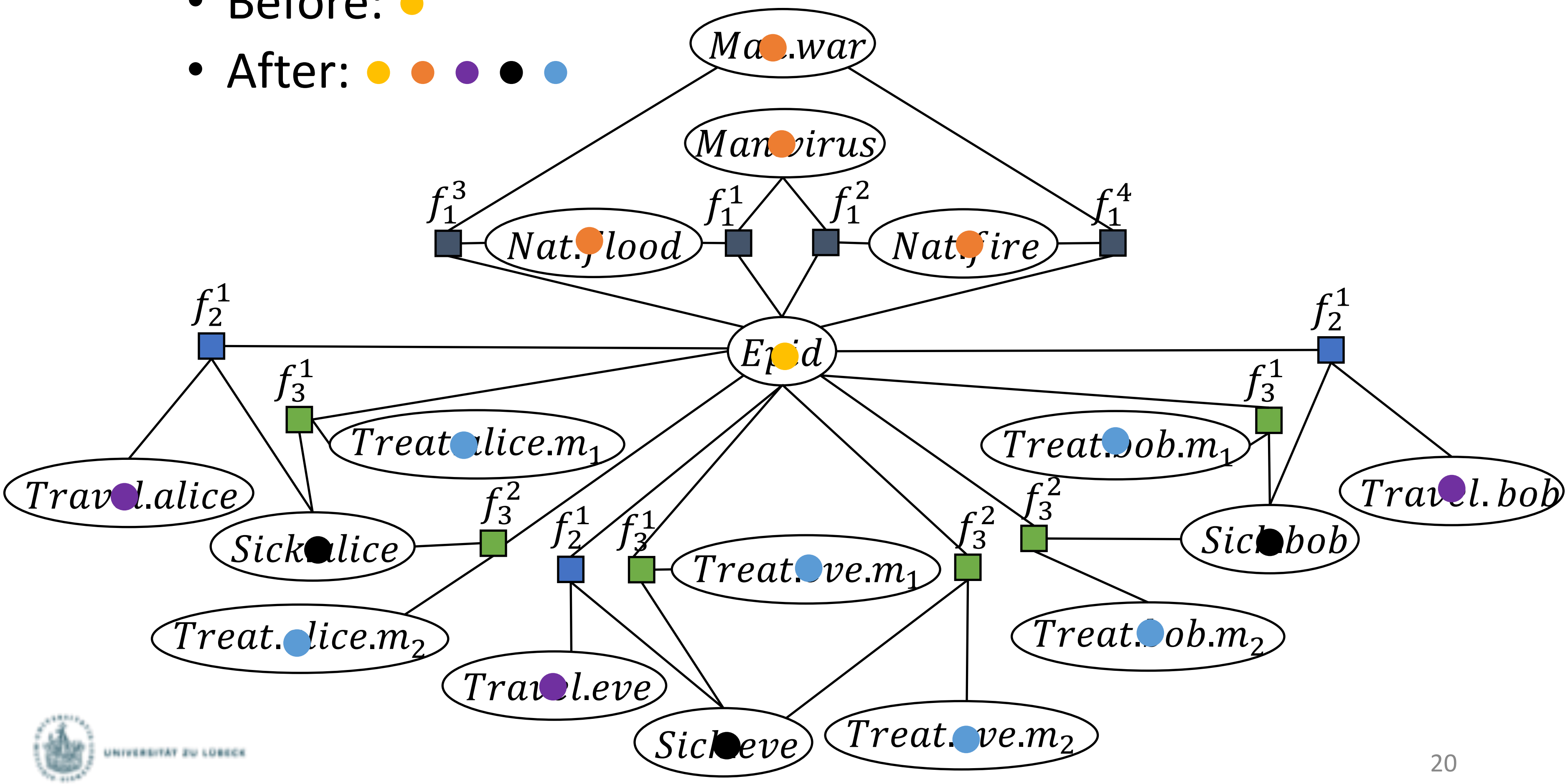
# Compression

4. Recolour nodes based on collected signatures
   - 5 colours for the nodes: ● ● ● ● ●
   - Factors as before



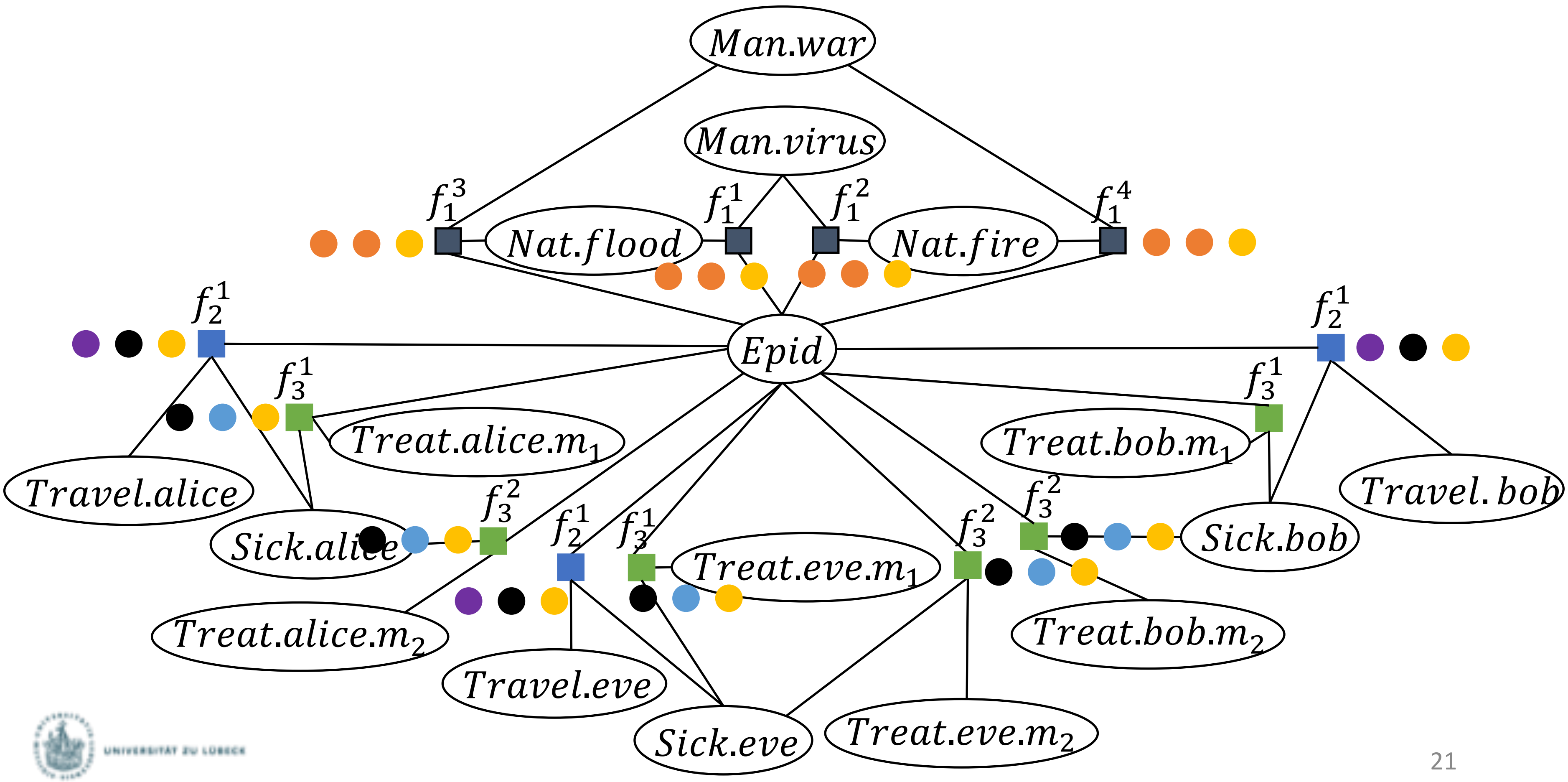
# Compression

5. If no new colour created, stop. Otherwise, pass colours again.
   - Before: 🟡
   - After: 🟡 🟠 🟣 ⚫ 🔵



# Compression

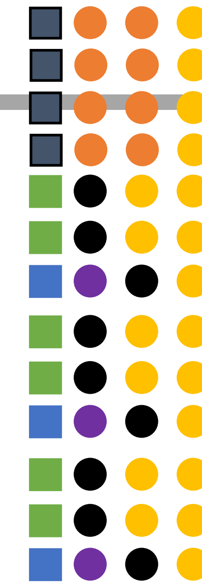
2. Factors collecting colours from nodes, signing their own colours to the collected ones

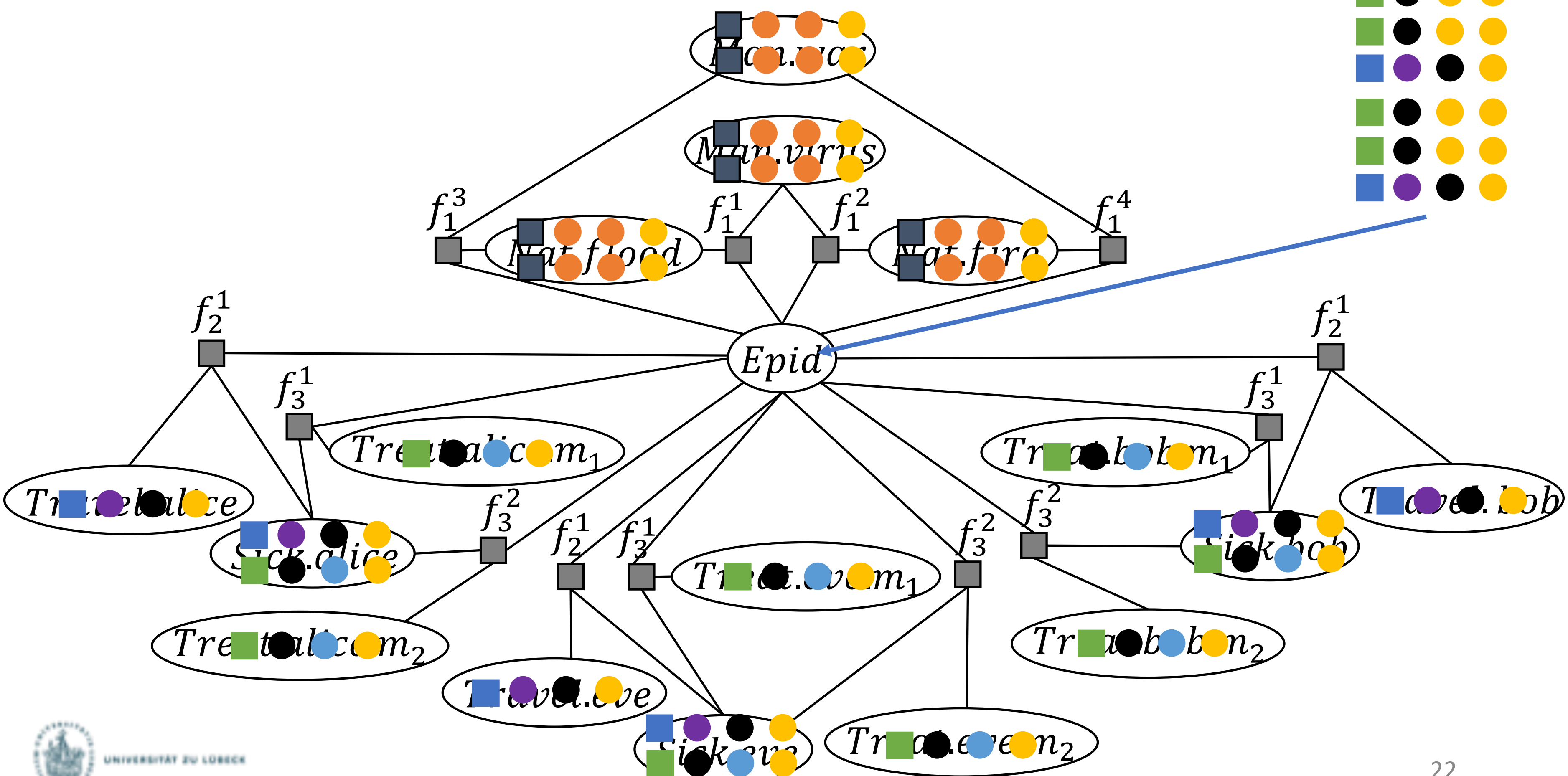



UNIVERSITÄT ZU LÜBECK

# Compression

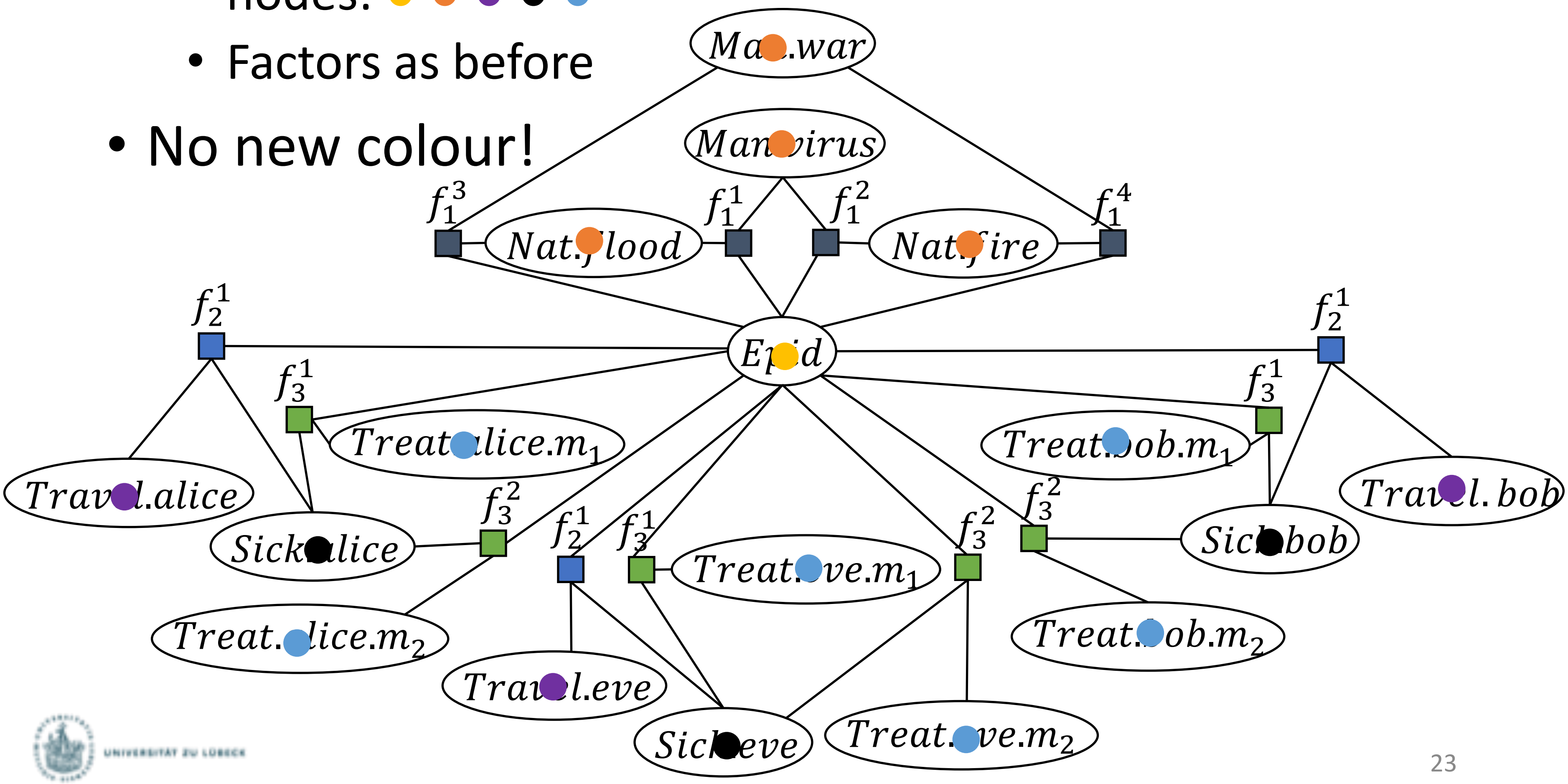
3. Nodes collecting colours from factors





# Compression

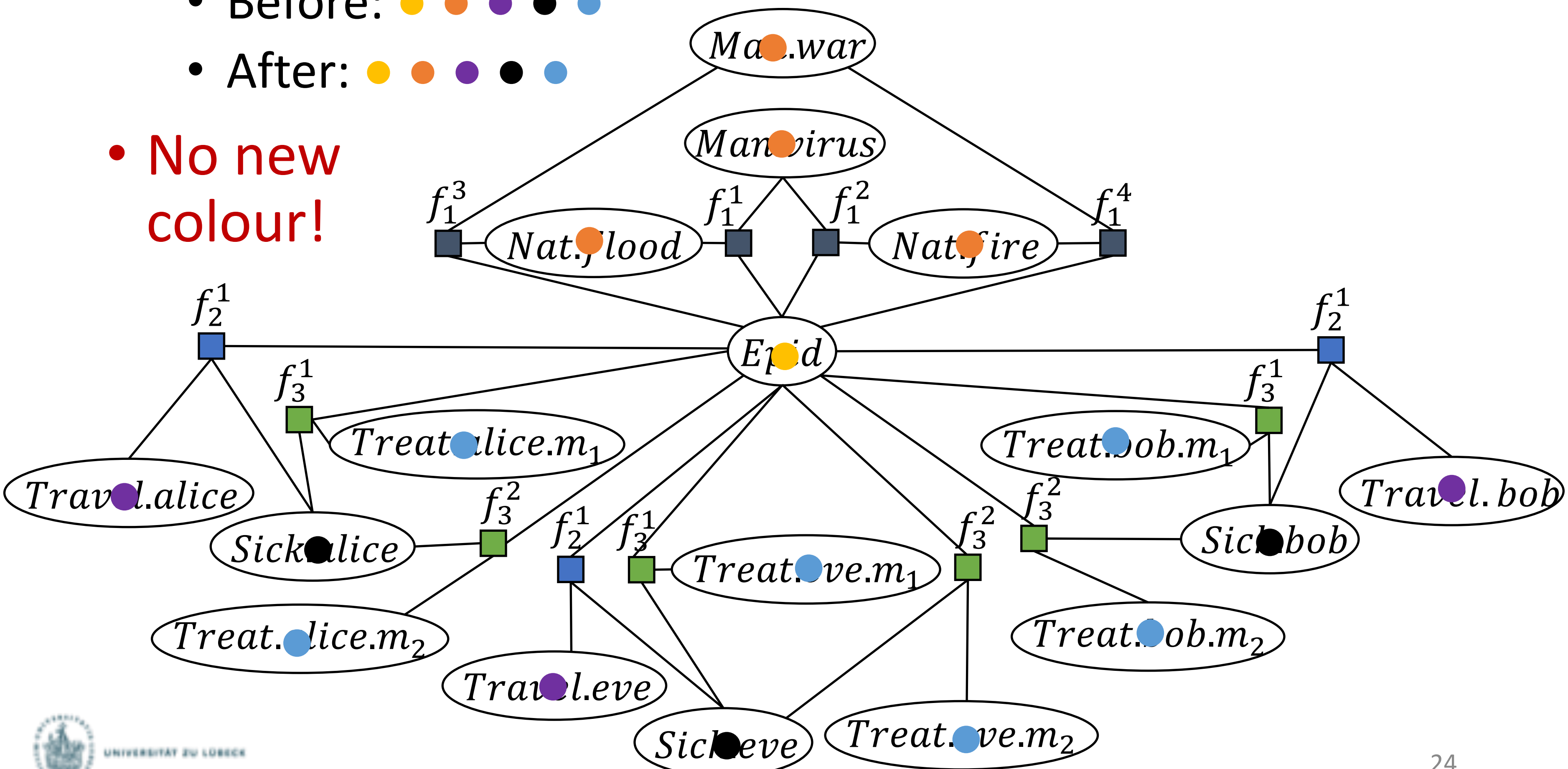
4. Recolour nodes based on collected signatures
   - 5 colours for the nodes: ● ● ● ● ●
   - Factors as before
- No new colour!



# Compression

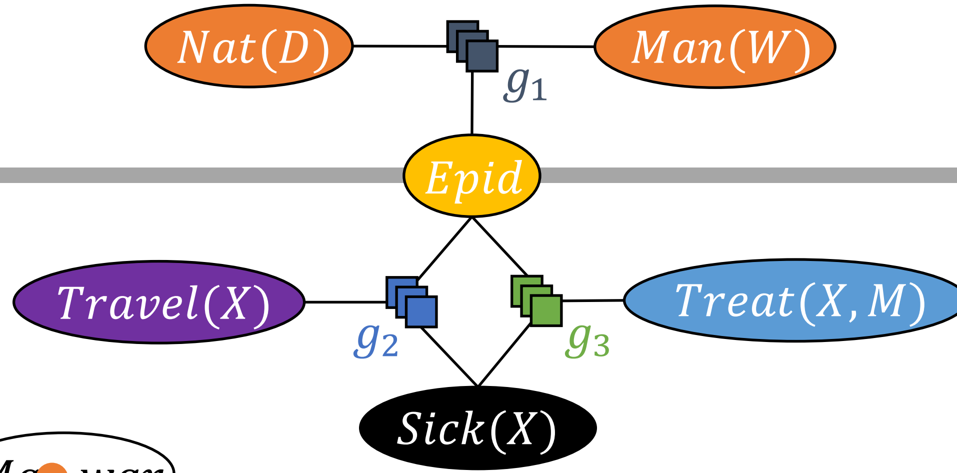
5. If no new colour created, stop. Otherwise, pass colours again.
   - Before:
   - After:

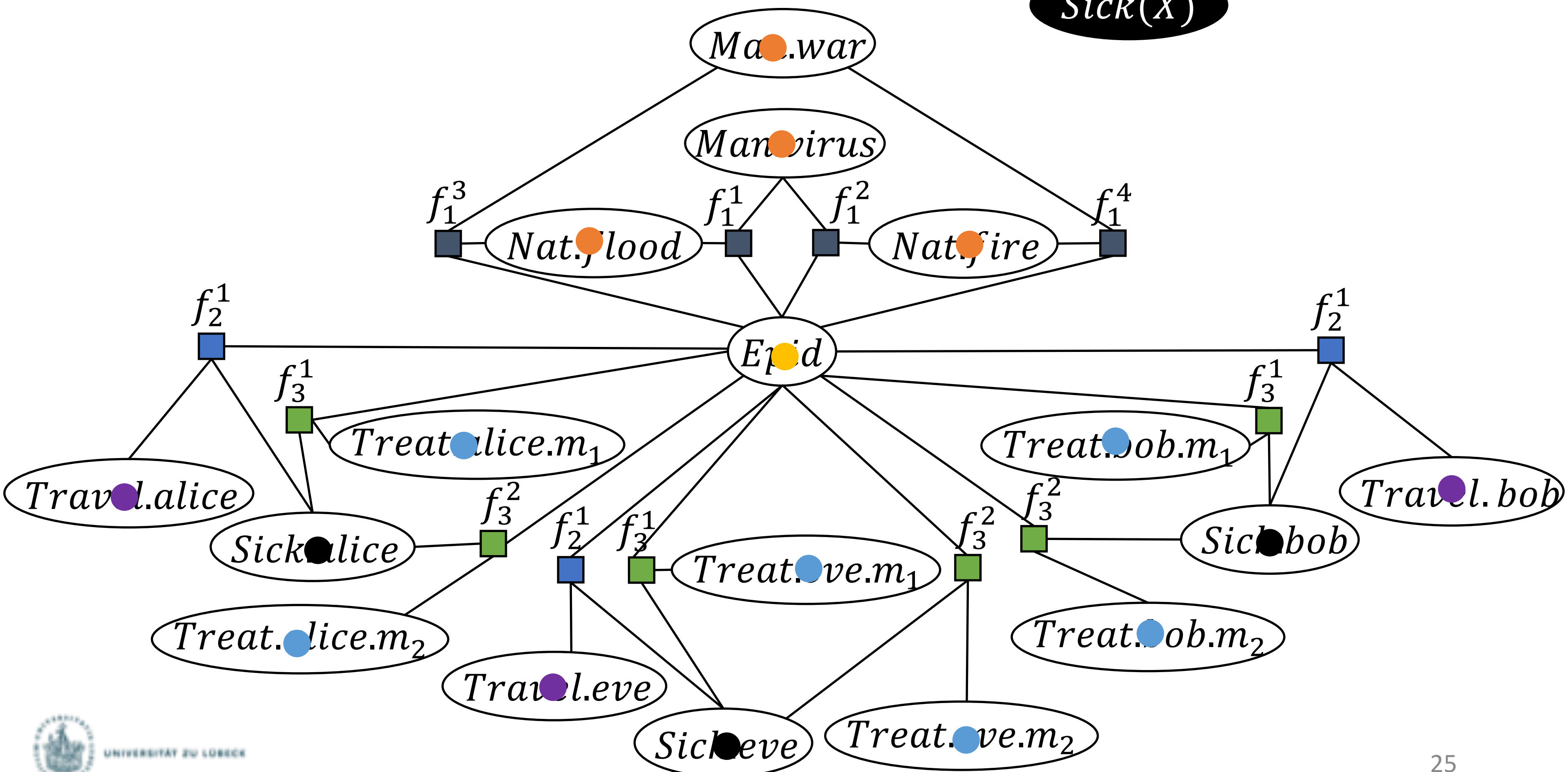
- No new colour!



# Compression

- Compressed graph:



# Colour Passing Compression

- Algorithm:

Singla and Domingos (2008), Kersting et al. (2009), Ahmadi et al. (2013)

1. Each factor collects the colours of its neighbouring nodes
2. Each factor „signs“ its colour signature with its own colour
3. Each node collects the signatures of its neighbouring factors
4. Nodes are recoloured according to the collected signatures
5. If no new colour is created stop, otherwise go back to 1

- Compress a model (lifted or grounded) based on semantics
  - Uses exact symmetries in factors
    - Same colour if factors considered equivalent
  - Ignores syntax
    - E.g., names of randvars

# Exact Symmetries

- Symmetries in (propositional) model allow for compact representation using parameters
  - PRVs for sets of indistinguishable randvars
- If randvars are indistinguishable,
  - what about yielding similar or even indistinguishable observations?

  → Next part!

Have a nice break!

We see each other again in 15 minutes.

# Symmetric Models
# &
# Symmetric Evidence

There are only so many values one can observe

# Symmetric Evidence

Taghipour et al. (2013a)

- Observations for specific randvars of a PRV can be
  - One of the range values
  - Not available
- Example: $Sick(X), r\big(Sick(X)\big) = \{true, false\}$
  - $Sick(x_1) = Sick(x_2) = \cdots = Sick(x_{10}) = true$
  - $Sick(x_{11}) = Sick(x_{12}) = \cdots = Sick(x_{20}) = false$

| $Sick(X^T)$ | $g_e^T$ |
|---|---|
| *false* | 0 |
| *true* | 1 |

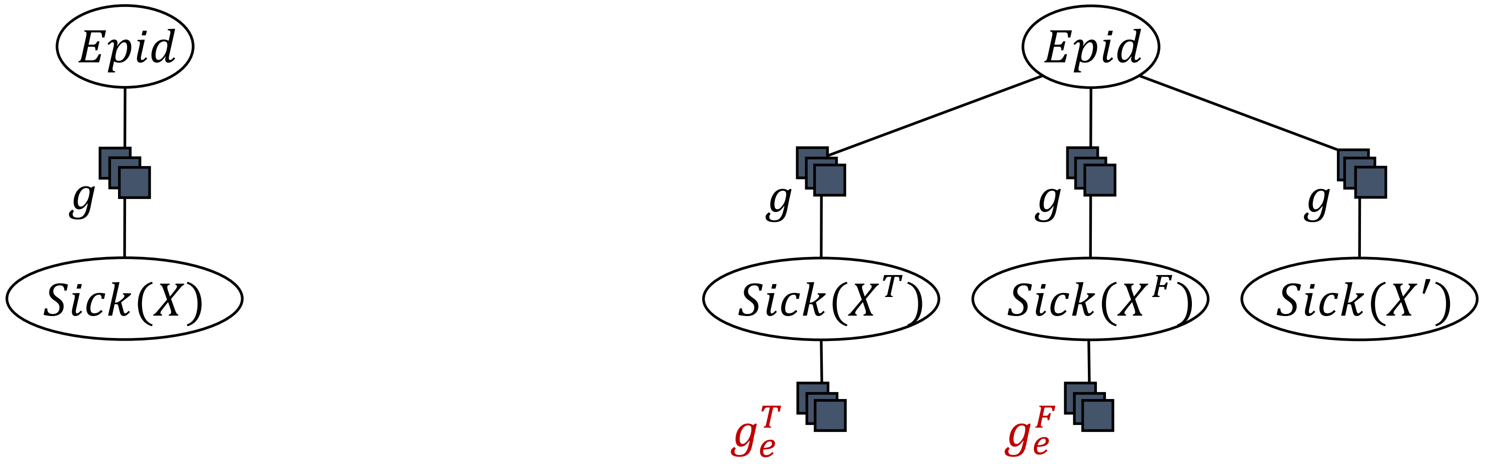
| $Sick(X^F)$ | $g_e^F$ |
|---|---|
| *false* | 1 |
| *true* | 0 |

  - $\mathcal{D}(X^T) = \{x_1, \dots, x_{10}\}, \mathcal{D}(X^F) = \{x_{11}, \dots, x_{20}\}$
  - Observations for $Sick(x_{21})$ … $Sick(x_n)$ not available

# Symmetric Evidence

Taghipour et al. (2013a)

- Evidence: $g_e^T, g_e^F$
  - $\mathcal{D}(X^T) = \{x_1, \dots, x_{10}\}$
  - $\mathcal{D}(X^F) = \{x_{11}, \dots, x_{20}\}$
  - $\mathcal{D}(X') = \{x_{21}, \dots, x_n\}$
- Shattering based on evidence



# Evidence Absorption

Taghipour et al. (2013a)

- Absorb evidence:
  - Set values to 0 where range value ≠ observation
    - Equivalent to multiplying $g$ with $g_e$
  - Possibly eliminate variable
    - Drop lines with values set to 0
    - Drop column of evidence PRV
  - Example
    - $Sick(X^T) = true$
    - $Sick(X^F) = false$

| $Epid$ | $Sick(X^F)$ | $g^F$ |
|---|---|---|
| $false$ | $false$ | 5 |
| $false$ | $true$ | ~~1~~ 0 |
| $true$ | $false$ | 4 |
| $true$ | $true$ | ~~6~~ 0 |

| $Epid$ | $Sick(X^T)$ | $g^T$ |
|---|---|---|
| $false$ | $false$ | ~~5~~ 0 |
| $false$ | $true$ | 1 |
| $true$ | $false$ | ~~4~~ 0 |
| $true$ | $true$ | 6 |

| $Epid$ | $Sick(X^F)$ | $g^F$ |
|---|---|---|
| $false$ | $false$ | 5 |
| $true$ | $false$ | 4 |

| $Epid$ | $Sick(X^T)$ | $g^T$ |
|---|---|---|
| $false$ | $true$ | 1 |
| $true$ | $true$ | 6 |

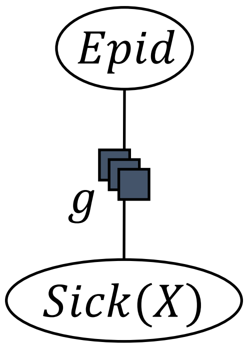
| $Epid$ | $g^F$ |
|---|---|
| $false$ | 5 |
| $true$ | 4 |

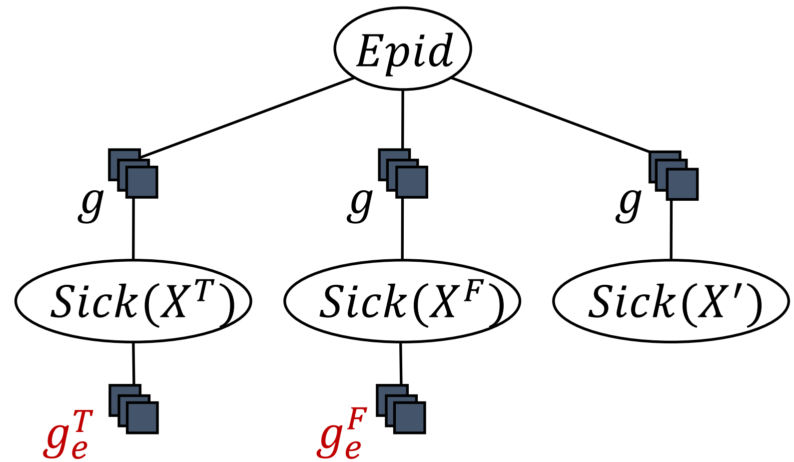
| $Epid$ | $g^T$ |
|---|---|
| $false$ | 1 |
| $true$ | 6 |

# Symmetric Evidence

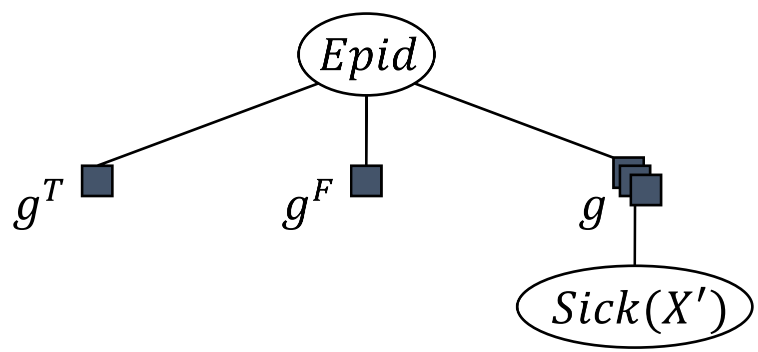
Taghipour et al. (2013a)

- Shattering based on evidence



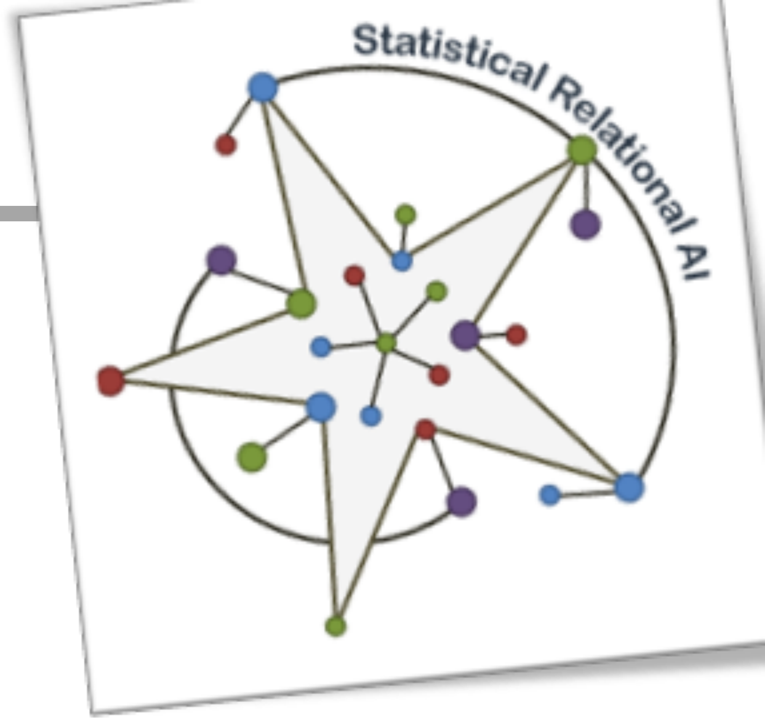

- After absorption



# Lifted Evidence & Completeness

Van den Broeck and Davis (2012)

- Evidence is liftable if observations for
  - Propositional randvars
  - PRVs with one logical variable
    - One set of constants per variable
    - E.g., observations for, e.g., $Travel(X), Sick(X)$
- Evidence for PRVs with two logical variables no longer liftable
  - Liftable cases possible but no guarantee for all possible constellations
  - More by Van den Broeck and Darwiche (2013) on special classes




# Symmetries in Queries

Indistinguishable query terms

Also: a highlight paper here at ECAI 2020!

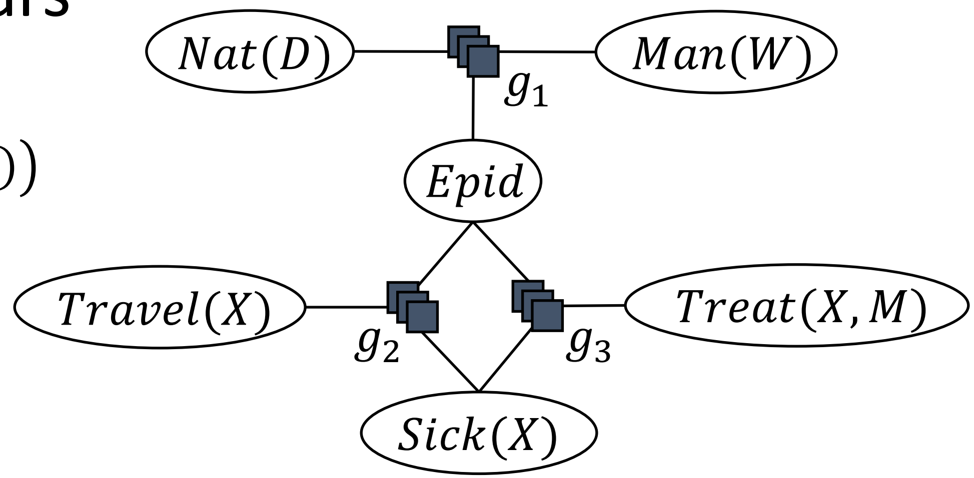
# Indistinguishable Query Terms

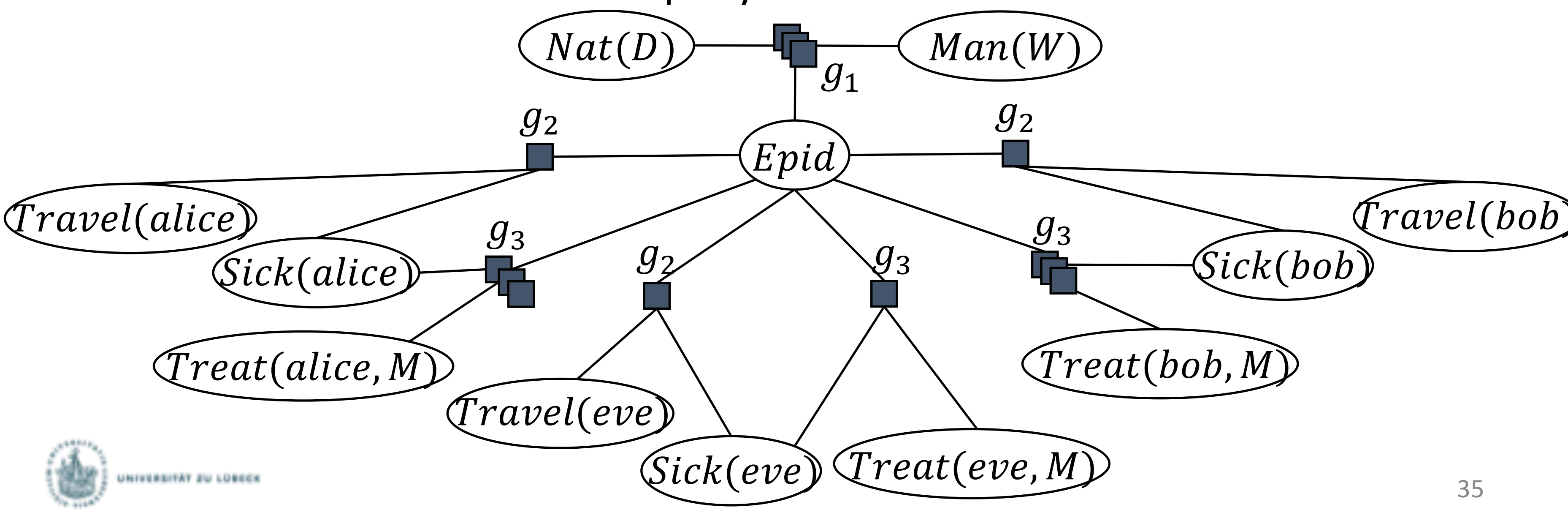
B and Möller (2018)

- Indistinguishable randvars in query:

  $P(Sick(alice), Sick(eve), Sick(bob))$

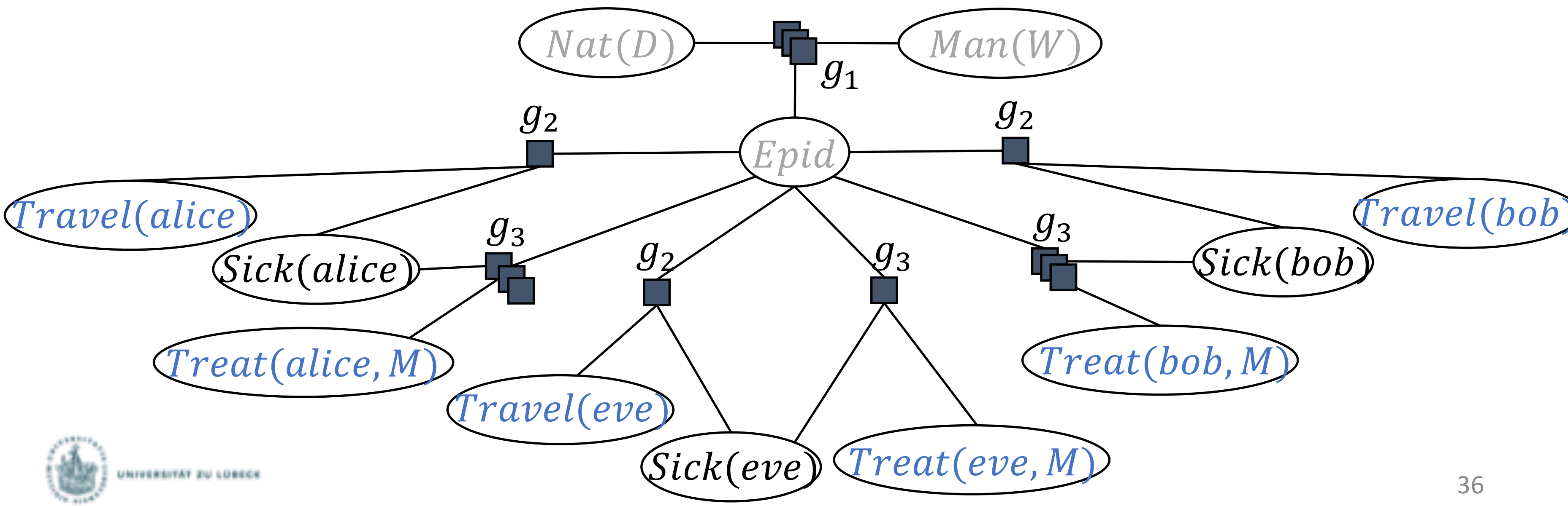
- Standard LVE:
  - Shattering
    - Leads to groundings w.r.t. constants in query



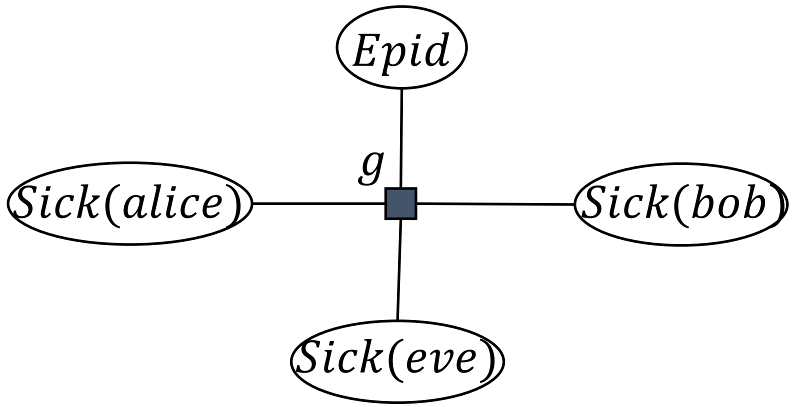
# … And Their Effect

- Query: $P\big(Sick(alice), Sick(eve), Sick(bob)\big)$
- After shattering, eliminate all non-query terms
  - Identical computations during elimination



# … And Their Effect

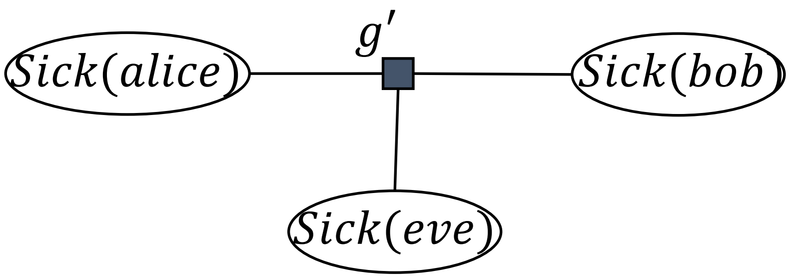
- Query: $P\big(Sick(alice), Sick(eve), Sick(bob)\big)$
- After shattering, eliminate all non-query terms
  - Identical computations during elimination
  - Large intermediate results




UNIVERSITÄT ZU LÜBECK

# … And Their Effect

- Query: $P\big(Sick(alice), Sick(eve), Sick(bob)\big)$
- After shattering, eliminate all non-query terms
- Symmetries in result



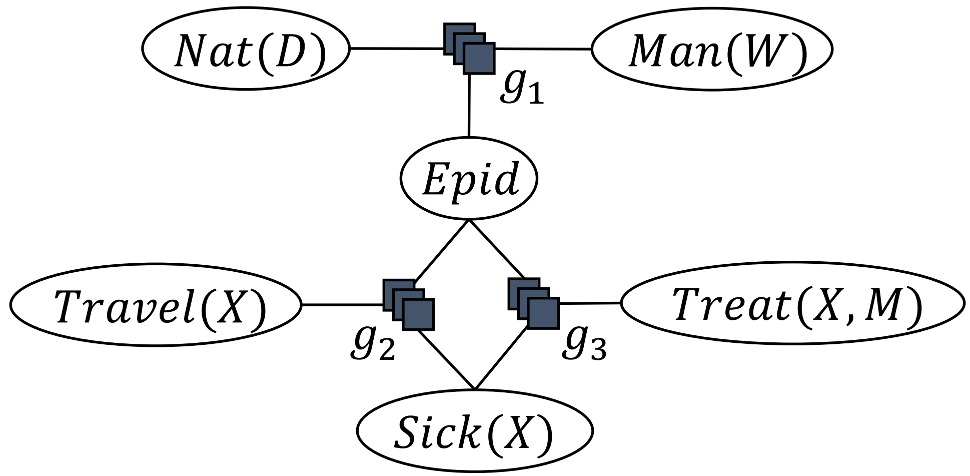

| $\#_X[Sick(X)]$ | $g$ |
|---|---|
| [0,3] | 1 |
| [1,2] | 2 |
| [2,1] | 3 |
| [3,0] | 4 |

| $Sick(alice)$ | $Sick(eve)$ | $Sick(bob)$ | $g'$ |
|---|---|---|---|
| *false* | *false* | *false* | 1 |
| *false* | *false* | *true* | 2 |
| *false* | *true* | *false* | 2 |
| *false* | *true* | *true* | 3 |
| *true* | *false* | *false* | 2 |
| *true* | *false* | *true* | 3 |
| *true* | *true* | *false* | 3 |
| *true* | *true* | *true* | 4 |

# Lifted Queries

B and Möller (2018)

- Parameterised query: $P\big(Sick(X)\big)$
- Standard LVE:
  - Shattering
    - If $X$ references a subdomain, then two groups
  - Elimination
    - Using standard LVE
  - Encode symmetries using so-called counting random variables, which have histograms as range values
    - Using LVE operator called count-conversion
    - If not already a by-product of elimination



# Lifted Queries & Completeness

B (2020)

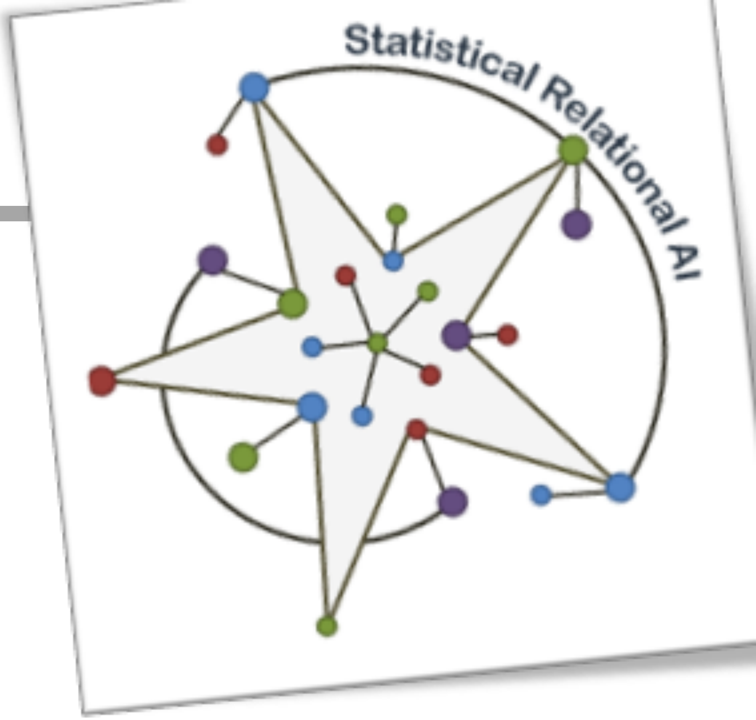
- Given a liftable model and liftable evidence
- Complexity
  - The complexity of LVE for liftable queries is polynomial in domain sizes.
- Completeness
  - Parameterised query terms with only one parameter per term and one set of constants per domain are liftable.
    - Otherwise, groundings may be unavoidable, e.g.,

      Query $P(B(X,Y))$ in model $g(A(X), B(X,Y), C(Y))$
- Corollary
  - Counting random variables compactly represent the result of liftable queries.

# Known Domains

- Grounding semantics is only defined given specific domains for logical variables
  - Evidence for known constants
  - Queries reference known constants
- Also, models usually learned on a specific domain

- What if…
  - domains change?
  - domains are unknown?

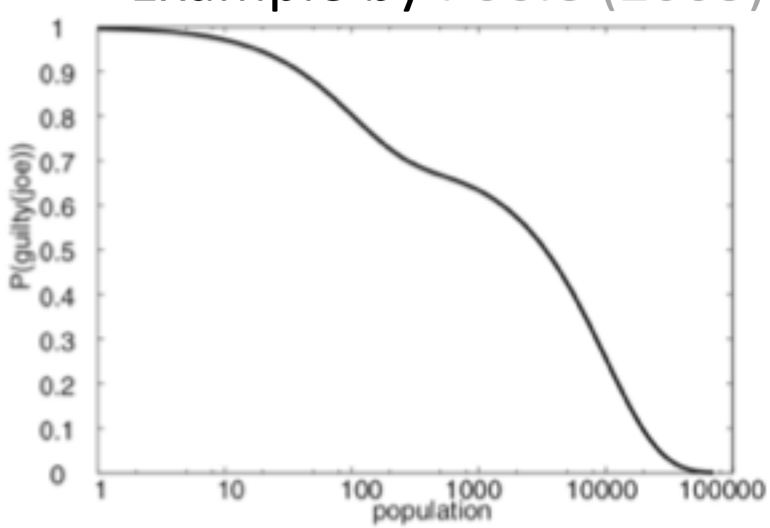


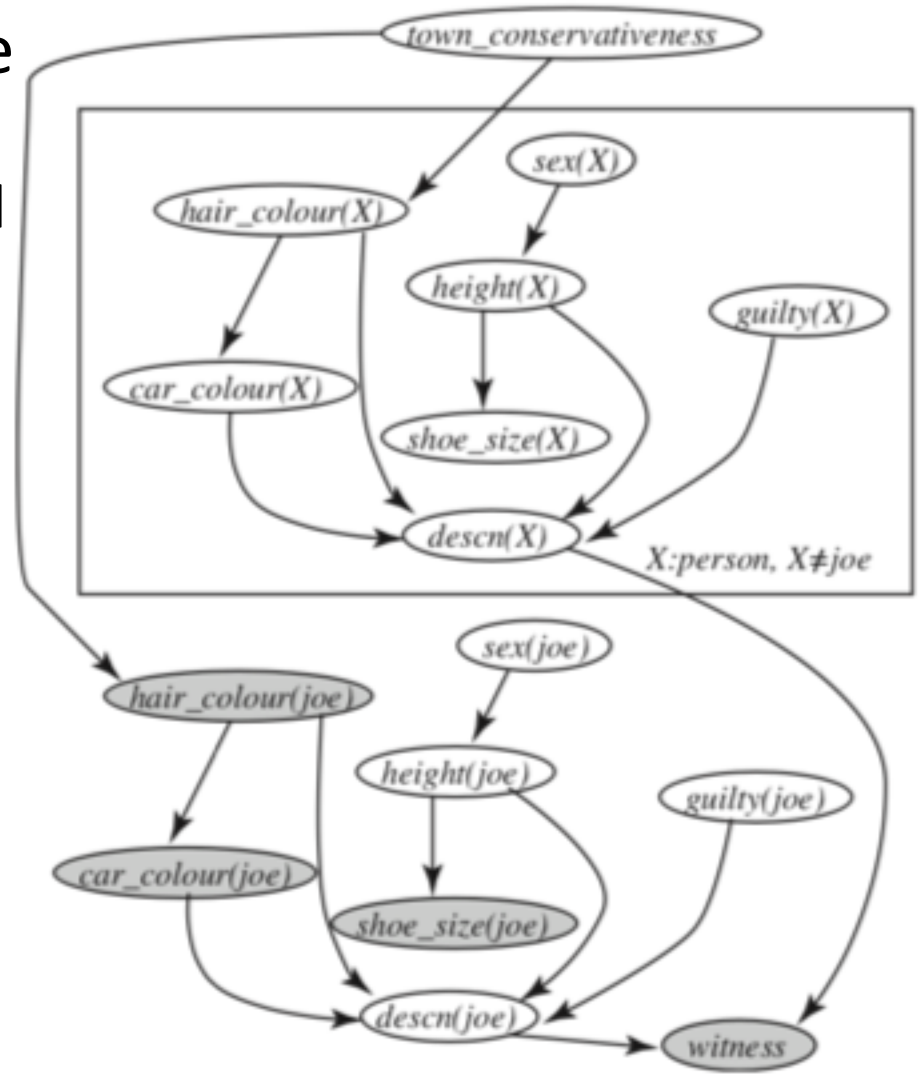

# Leaving a specific domain behind…

What happens if **domains change**?

# Changing Domains

- Keep semantics as before
  - Assume that parfactors accurately describe world
- Posterior probabilities change depending on domain sizes
  - Example by Poole (2003)

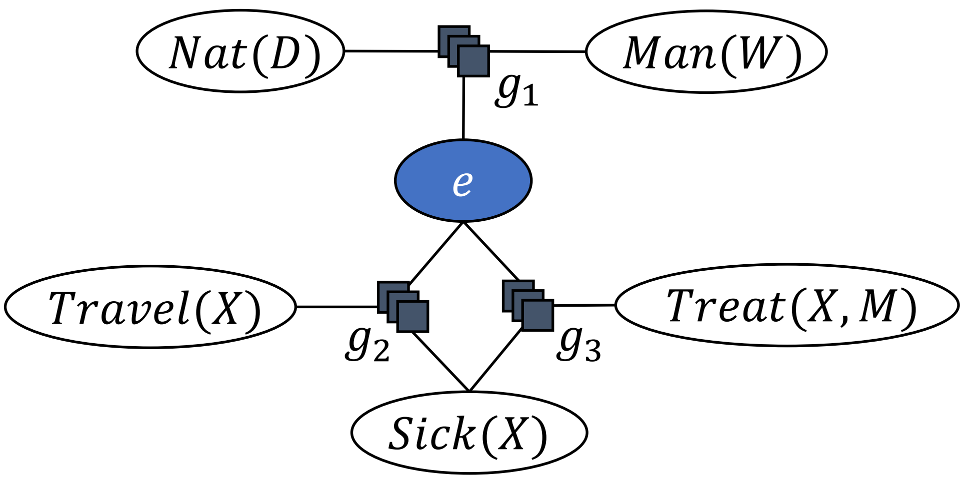



UNIVERSITÄT ZU LÜBECK

# … Without Effects

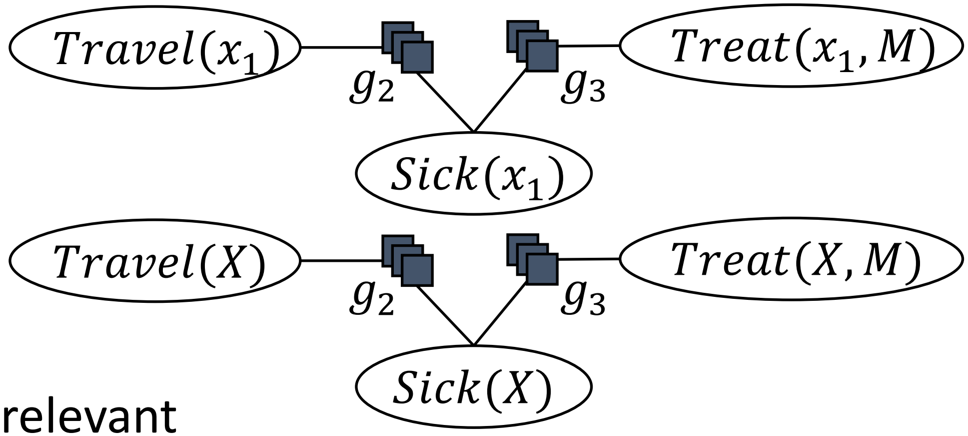
- (Conditional) Independence
  PRVs, containing logical variables $\boldsymbol{X}$, that are (conditionally) independent from query terms $\rightarrow$ domains of $\boldsymbol{X}$ have no influence on query results
- E.g., given $Epid = e$,
  - $\mathcal{D}(D)$ and $\mathcal{D}(W)$ do not matter for queries regarding $Travel$, $Sick$, and $Treat$
  - $\mathcal{D}(X)$ and $\mathcal{D}(M)$ do not matter for queries regarding $Nat$ and $Man$




$\rightarrow$ Partly invariant under increasing domain sizes

# … Without Effects

- A simple case of so-called projectivity
  After shattering, query terms are independent of model parts containing logical variables $\boldsymbol{X}$ → domains of $\boldsymbol{X}$ have no influence on query results
  - Depends on model structure
  - More by Jaeger and Schulte (2018)
- E.g., $P(Sick(x_1))$
  - $\mathcal{D}(X) = \{x_1, \dots, x_n\}$
  - After shattering:
    - $\mathcal{D}(X) = \{x_2, \dots, x_n\}$
    - Upper part independent from lower part; $\mathcal{D}(X)$ irrelevant

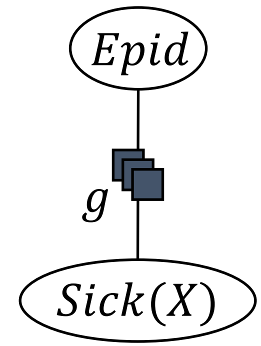



→ Partly invariant under increasing domain sizes

# Growing Domain Sizes

Poole et al. (2014)

- Let domain size $n$ grow
  - With grounding semantics, posteriors change
    - Can lead to extreme behaviour in the posteriors
- Example: $Epid$ gets more and more neighbours with $n$ rising




$$P(Epid) \propto \left( \sum_{s \in r(Sick(X))} g(Epid, Sick(x) = s) \right)^n$$

$$= \left(g'(Epid)\right)^n = g''(Epid) = g^\alpha(Epid)$$

| $Epid$ | $g'$ |
|---|---|
| $false$ | $a$ |
| $true$ | $b$ |

| $Epid$ | $g''$ |
|---|---|
| $false$ | $a^n$ |
| $true$ | $b^n$ |

| $Epid$ | $g^\alpha$ |
|---|---|
| $false$ | $\frac{a^n}{a^n + b^n}$ |
| $true$ | $\frac{b^n}{a^n + b^n}$ |

$\rightarrow \frac{1}{1 + \left(\frac{b}{a}\right)^n}$ (Sigmoid function) $\xrightarrow{n \to \infty}$ $b > a$: $0$; $a > b$: $1$

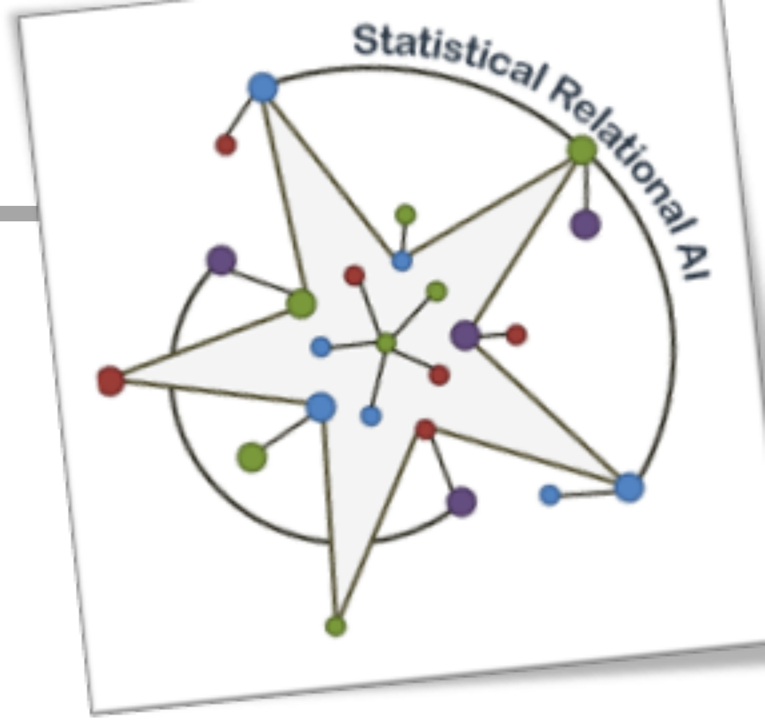
# Growing Domain Sizes

Mittal et al. (2019)

- How to avoid extreme behaviour?

→ Adapt values in model dependent on domain size

- Approach for MLNs: Domain-size aware MLNs
  - Assume predicates $P_1, \dots, P_m$ occur in a first-order formula $F$
    - Count number of connections $c_j$ for each predicate $P_j$ given *new* domains
    - Build a connection vector $[c_1, \dots, c_m]$
    - Choose $\max_{c_i}[c_1, \dots, c_m]$ as scaling-down factor
      - Instead of $\max$, other functions possible
      - Works best if the values in $[c_1, \dots, c_m]$ do not vary that much
  - Given an MLN with a set of formulas $F_i$ with weights $w_i$
    - Rescale each $w_i$ with scaling-down factor $s_i$ computed for $F_i$ as $\frac{w_i}{s_i}$
- Analogous approach possible for parfactors

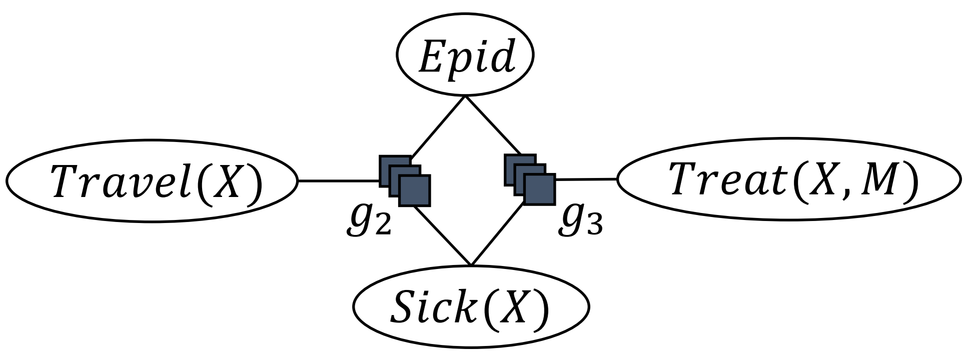



# Leaving a specific domain behind…

What happens if a **domain** is **unknown**?

# Known Domains

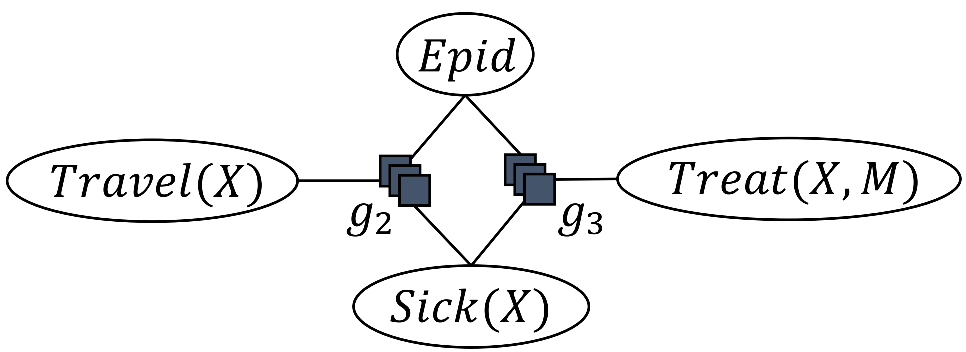
- General domains of logical variables, e.g.,
  - $\mathcal{D}(X) = \{alice, eve, bob\}$ or
  - $\mathcal{D}(X) = \{x_1, \dots, x_n\}$
- Constraint $C_i$ in each parfactor $g_i$, e.g.,
  - $C_3 = \big((X, M), \mathcal{D}(X) \times \mathcal{D}(M)\big)$
- Based on constraints, grounding semantics apply
  - Lifted algorithms work




UNIVERSITÄT ZU LÜBECK

# Unknown Domains

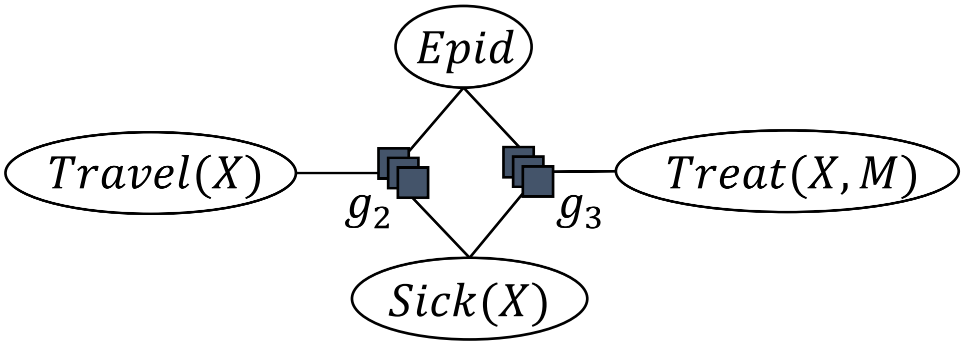
- General domains of logical variables, e.g.,
  - $\mathcal{D}(X) =$ ~~$\{alice, eve, bob\}$ or~~
  - $\mathcal{D}(X) =$ ~~$\{x_1, \ldots, x_n\}$~~
- Constraint $C_i$ in each parfactor $g_i$, e.g.,
  - $C_3 = \big((X, M),$ ~~$\mathcal{D}(X) \times \mathcal{D}(M)$~~$\big)$
- ~~Based on constraints, grounding semantics apply~~
  - ~~Lifted algorithms work~~



# Template Model + Constraint Program

B and Möller (2019)

- Template model:
  Parfactors with empty constraints, e.g.,
  - $C_3 = ((X, M), \bot)$




- Constraint program:
  Generate specific constraints for a template model given a domain, e.g., using probabilistic Datalog:

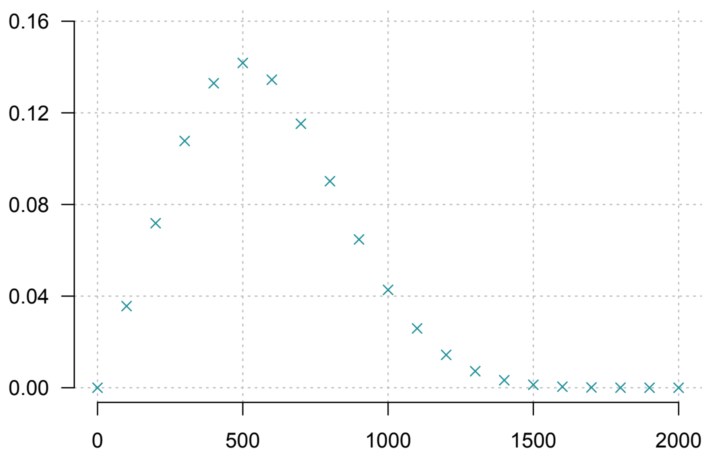
```
element_of_C3(X,Y1) :- linked(X,Y1,Y2).
element_of_C3(X,Y2) :- linked(X,Y1,Y2).
linked(X,Y1,Y2) :- instance_of_X(X) & pair(Y1,Y2).
0.7 pair(t1,t2).
0.2 pair(t2,t3).
0.1 pair(t1,t3)
```

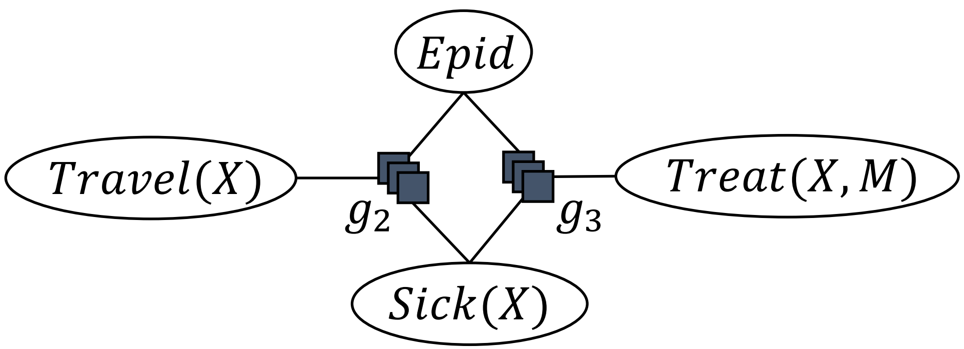
Yields 3 constraint sets per domain

# Domain Worlds

B and Möller (2019)

- Specify or generate possible domains
- Encode assumptions, e.g.,
  - Small domains more likely than large domains
  - Only rough
- For $X$, e.g.,
  - Beta-binomial distribution ($\alpha = 6, \beta = 15$)




Yields 20 domains with probability > 0

# Groundings-based Semantics

B and Möller (2019)

**Inputs**

- Template model
  - Empty constraints
- Constraint program
  - Fill empty constraints given a domain world
  - Can generate a probability distribution over models
- Domain worlds
  - Generate possible worlds as input to constraint program
  - Can be a probability distribution over domains

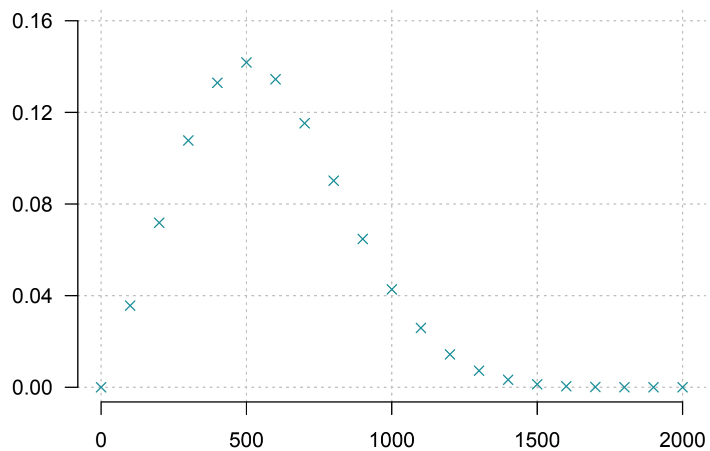
**Approach**

- Generate a set of possible models
  - Can be a probability distribution over possible models
  - Within model: grounding semantics apply
    - Lifted algorithms work again
- Reasoning over possible models
  - New query types

# Interworkings

B and Möller (2019)

- Distribution over domains




Together, they yield 20 · 3 constraint sets, each with probability > 0

- … as input to probabilistic constraint program

```
element_of_C3(X,Y1) :- linked(X,Y1,Y2).
element_of_C3(X,Y2) :- linked(X,Y1,Y2).
linked(X,Y1,Y2) :- instance_of_X(X) & pair(Y1,Y2).
0.7 pair(t1,t2).
0.2 pair(t2,t3).
0.1 pair(t1,t3)
```

# Filtering

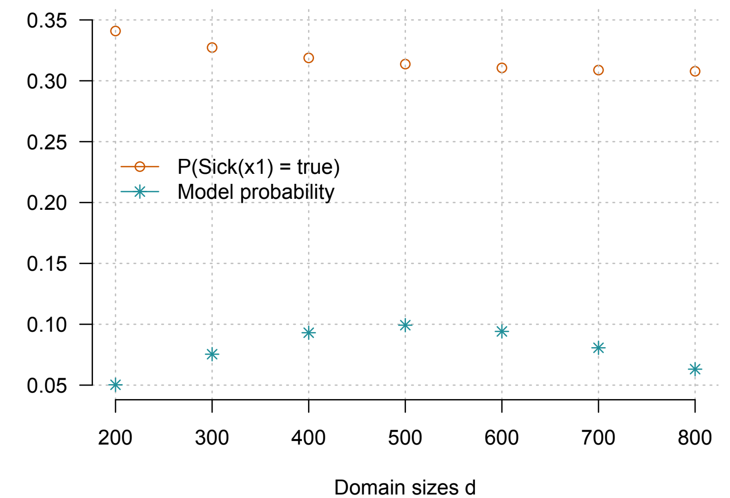
B and Möller (2019)

- Together, they yield $20 \cdot 3$ constraint sets, each with probability > 0
  - Some probabilities very low
- Filtering based on probabilities; e.g.,
  - Threshold $t$
  - Keep only those models whose probabilities make up, e.g., 95% of the distribution around its mean or maximum value
- Cascading filtering
  1. Filter domain worlds
  2. Filter constraint sets resulting from remaining domain worlds

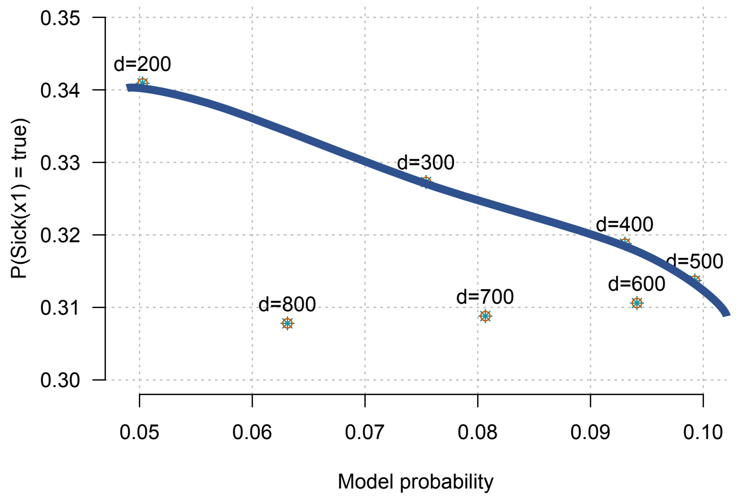
# New Queries Emerging

B and Möller (2019)

- Exploration
  - Model and query probabilities w.r.t.
    - Domain sizes (as in changing domains + grounding semantics)
    - Skyline query



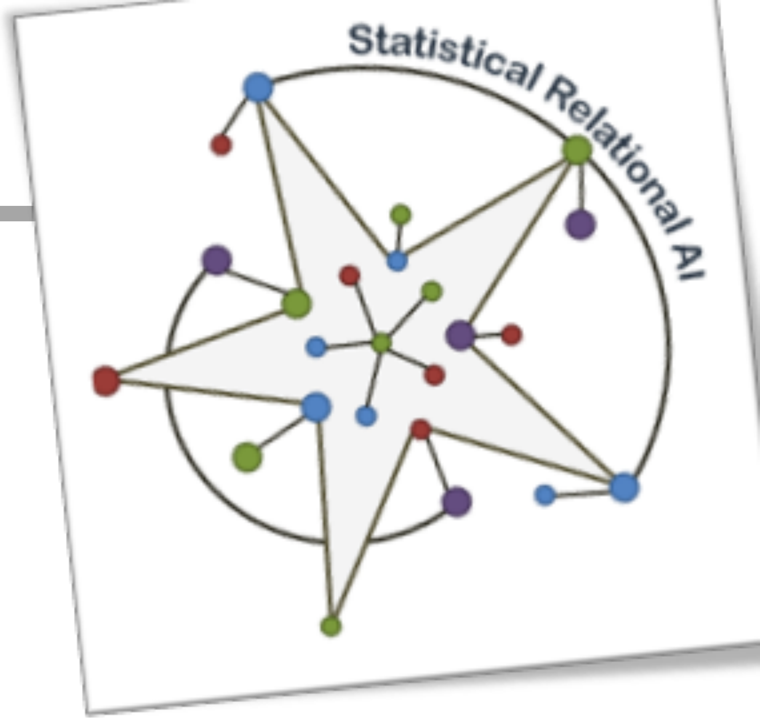

- Model checking
  - E.g., does the probability of
    - an individual being sick decrease with larger domains?
    - an epidemic rise if more people travel?



# Wrap-up Symmetries and Domains

- Exact symmetries in PRMs
  - Grounding semantics
  - Tractability of query answering problem
  - Colour passing for exact compression of models
  - Symmetric evidence for lifted evidence handling
  - Lifted queries for lifted query answering
- Changing domains
  - Models that are invariant under increasing domain sizes
  - Adapting weights to avoid extreme behaviour
- Unknown domains
  - Set of or distribution over universes

Next: Stable inference over time in dynamic PRMs




# Bibliography

Alphabetically sorted

# Bibliography


- Ahmadi et al. (2013)
  Babak Ahmadi, Kristian Kersting, Martin Mladenov, and Sriraam Natarajan. Exploiting Symmetries for Scaling Loopy Belief Propagation and Relational Training. In *Machine Learning*. 92(1):91-132, 2013.
- B (2020)
  Tanya Braun. Rescued from a Sea of Queries: Exact Inference in Probabilistic Relational Models. PhD Thesis, 2020.
- B and Möller (2018)
  Tanya Braun and Ralf Möller. Parameterised Queries and Lifted Query Answering. In *IJCAI-18 Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.
- B and Möller (2019)
  Tanya Braun and Ralf Möller. Exploring Unknown Universes in Probabilistic Relational Models. In *Proceedings of AI 2019: Advances in Artificial Intelligence*, 2019.

# Bibliography


- Jaeger and Schulte (2018)
  Manfred Jaeger and Oliver Schulte. Inference, Learning, and Population Size: Projectivity for SRL Models. In *StaRAI-18 Workshop on Statistical Relational Artificial Intelligence*, 2018.
- Kersting et al. (2009)
  Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting Belief Propagation. In *UAI-09 Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.
- Lauritzen and Spiegelhalter (1988)
  Steffen L. Lauritzen and David J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society. Series B: Methodological*, 50:157–224, 1988.
- Mittal et al. (2019)
  Happy Mittal, Ayush Bhardwaj, Vibhav Gogate, and Parag Singla. Domain-size Aware Markov Logic Networks. In *AISTATS-19 Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

# Bibliography


- Niepert and Van den Broeck (2014)
  Mathias Niepert and Guy Van den Broeck. Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference. In *AAAI-14 Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.
- Pearl (1982)
  Judea Pearl. Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. In *AAAI-82 Proceedings of the 2nd National Conference on Artificial Intelligence*, 1982.
- Poole (2003)
  David Poole. First-order Probabilistic Inference. In *IJCAI-03 Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.
- Poole et al. (2014)
  David Poole, David Buchman, Seyed Mehran Kazemi, Kristian Kersting, and Sriraam Natarajan. Population Size Extrapolation in Relational Probabilistic Modeling. In *SUM-14 Proceedings of the 8th International Conference on Scalable Uncertainty Management*, 2014.

# Bibliography


- Singla and Domingos (2008)
  Parag Singla and Pedro Domingos. Lifted First-order Belief Propagation. In *AAAI-08 Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 2008.
- Taghipour et al. (2013)
  Nima Taghipour, Daan Fierens, Guy Van den Broeck, Jesse Davis, and Hendrik Blockeel. Completeness Results for Lifted Variable Elimination. In *AISTATS-13 Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, 2013.
- Taghipour et al. (2013a)
  Nima Taghipour, Daan Fierens, Jesse Davis, and Hendrik Blockeel. Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *Journal of Artificial Intelligence Research*, 47(1):393–439, 2013.
- Van den Broeck (2011)
  Guy Van den Broeck. On the Completeness of First-order Knowledge Compilation for Lifted Probabilistic Inference. In *NIPS-11 Advances in Neural Information Processing Systems 24*, 2011.

# Bibliography


- Van den Broeck and Darwiche (2013)
  Guy Van den Broeck and Adnan Darwiche. On the Complexity and Approximation of Binary Evidence in Lifted Inference. In *NIPS-13 Advances in Neural Information Processing Systems 26*, 2013.
- Van den Broeck and Davis (2012)
  Guy Van den Broeck and Jesse Davis. Conditioning in First-Order Knowledge Compilation and Lifted Probabilistic Inference. In *AAAI-12 Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

# The End *

*PRMs are a true backbone of AI, and this tutorial emphasized only some central topics. We definitely did not cite all publications relevant to the whole field of PRMs here. We would like to thank all our colleagues for making their slides available (see some of the references to papers for respective credits). Slides or parts of it are almost always modified.