On the Completeness and Complexity of the Lifted Dynamic Junction Tree Algorithm

Marcel Gehrke

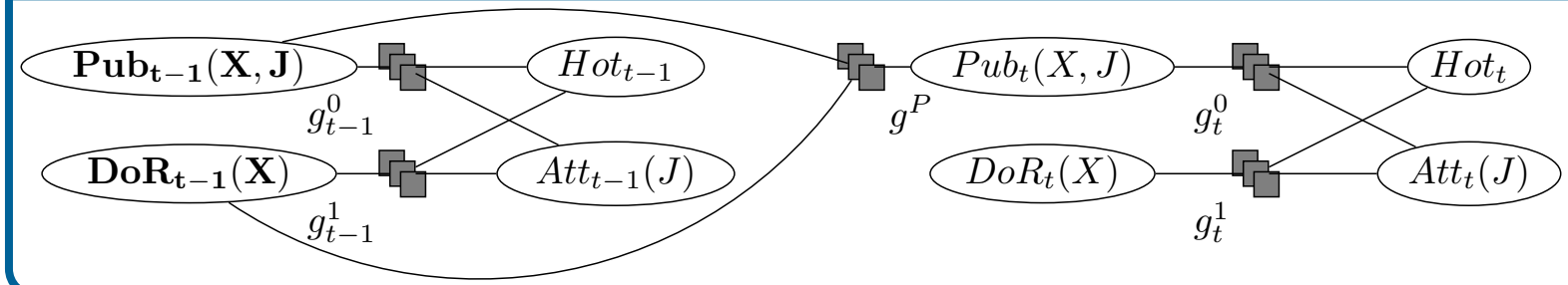
Link to paper (QR code): https://arxiv.org/abs/2110.09197

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

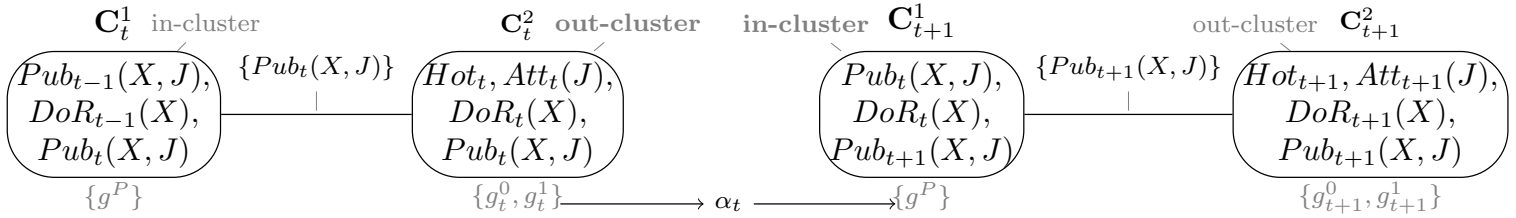
Completeness and Complexity of Lifted Inference

- *Static lifted inference, complete for FO² or 2-logvar fragment*
- *Exact temporal inference algorithms ideally have a complexity linear to the # of time steps*
- How about LDJT?
  - Do the additional constraints on the elimination order of LDJT influence liftability?
  - How do different query types influence LDJT's complexity?

Parameterised Temporal Relational Model (G)



First Order Junction Tree (J_t and J_{t+1})



Completeness of LDJT

- Not complete for the 2-logvar fragment
  - PRV with 2-logvars from time-slice $t-1$ and t in the interface may lead to groundings
  - As long as a PRV with 2-logvars only occurs in 1 time slice of the interface, LDJT computes a lifted solution
- ▶ Additional research w.r.t. count-conversions needed to make LDJT complete for FO²

Complexity of LDJT

- Complexity for filtering queries:
  $O\left(\left(T \cdot n_J + m\right) \cdot \log_2(n) \cdot r^{w_g} \cdot n_{\#}^{w_{\#} \cdot r_{\#}}\right)$
- Complexity + hindsight & prediction queries:
  $O\left(\left((T^2+T) \cdot n_J + m\right) \cdot \log_2(n) \cdot r^{w_g} \cdot n_{\#}^{w_{\#} \cdot r_{\#}}\right)$
- ▶ At least a reduction from exponential to polynomial complexity w.r.t. domain size compared to ground interface algorithm

T is the maximum number of time steps, m is the number of queries, n is the largest domain size among $lv(G)$, $n_{\#}$ is the largest domain size of the counted logvars, r is the largest range size in G, $r_{\#}$ is the largest range size among the PRVs in the counted RVs, and n_J being the number of parclusters in an FOJT. The largest possible factor is given by $r^{w_g} \cdot n^{w_{\#} \cdot r_{\#}}$