Learning to Assign Degrees of Belief in Relational Domains

Frederic Koriche

LIRMM, Université Montpellier 2, France

Inductive Logic Programming 2007

Outline

1. Learning to Reason
   - The Knowledge Representation Approach
   - The Learning to Reason Approach
2. Exponentiated Gradient Learning to Reason
   - Two Key Ideas
   - The Algorithm
3. Tractable Query Languages
   - Decomposable Queries
   - Hitting Languages
   - Cluster Languages
4. Perspectives

Relational Vocabulary

A finite set of *relation symbols*, and a finite set of *constants*

- Background knowledge: a set $\mathcal{B}$ of ground atoms
- Relational interpretation: a subset I of $\mathcal{B}$

Example

Consider a simple logistic domain

- Constants: 20 objects 5 trucks and 4 cities.
- Relations: In(x, y), At(x, y).

The background knowledge contains 200 ground atoms

Relational Vocabulary

A finite set of *relation symbols*, and a finite set of *constants*

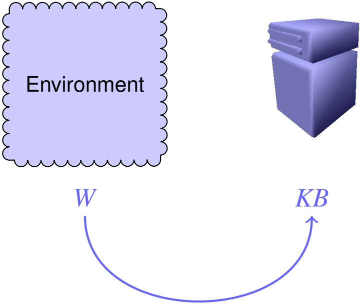
- Background knowledge: a set $\mathcal{B}$ of ground atoms
- Relational interpretation: a subset I of $\mathcal{B}$

Example

Consider a simple logistic domain

- Constants: 20 objects 5 trucks and 4 cities.
- Relations: In(x, y), At(x, y).

The background knowledge contains 200 ground atoms



KR Approach

The reasoning agent is given a description of its environment

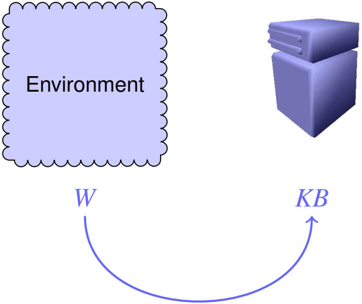
Environment

Distribution W on the space $2^{\mathcal{B}}$ of relational interpretations

Knowledge Base

A description KB of the environment W

- Logical theory
- Bayesian network



KR Approach

The reasoning agent is given a description of its environment

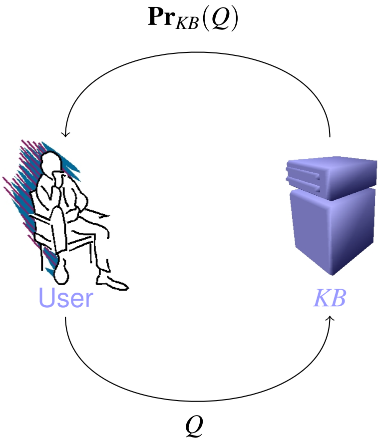
Environment

Distribution W on the space $2^{\mathcal{B}}$ of relational interpretations

Knowledge Base

A description KB of the environment W

- Logical theory
- Bayesian network



KR Approach

The agent is expected to evaluate any query with perfect precision

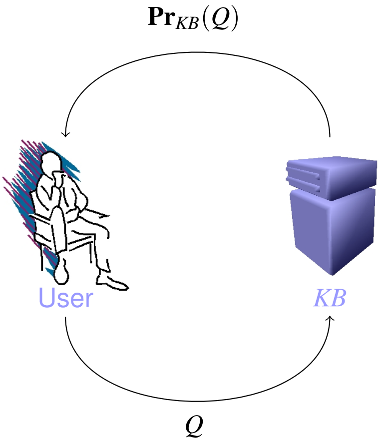
Degree of Belief

For any query Q, the probability of Q according to KB is

$$\mathbf{Pr}_{KB}(Q) = \sum_{I \models Q} \mathbf{Pr}_{KB}(I)$$

Example

$\mathbf{Pr}_{KB}(\mathsf{In}(\mathsf{o}_1, \mathsf{t}_1)) = \frac{3}{4}$ the agent believes that object o_1 is in the truck t_1 with probability $\frac{3}{4}$



KR Approach

The agent is expected to evaluate any query with perfect precision

Degree of Belief

For any query Q, the probability of Q according to KB is

$$\mathbf{Pr}_{KB}(Q) = \sum_{I \models Q} \mathbf{Pr}_{KB}(I)$$

Example

$\mathbf{Pr}_{KB}(\mathsf{In}(\mathsf{o}_1, \mathsf{t}_1)) = \frac{3}{4}$ the agent believes that object o_1 is in the truck t_1 with probability $\frac{3}{4}$

Complexity

The problem of evaluating the degree of belief of any query is #P-Hard

Simple Query Languages

The complexity is unchanged for very simple queries:

- Quantified literals: $\forall x\, \mathrm{In}(x, \mathrm{t}_1)$
- Ground atoms: $\mathrm{At}(\mathrm{c}_1, \mathrm{t}_1)$

Simple Representation Languages

The complexity is unchanged for simple representation languages:

- Horn Theories
- Monotone DNF Theories

Complexity

The problem of evaluating the degree of belief of any query is $\#$P-Hard

Simple Query Languages

The complexity is unchanged for very simple queries:

- Quantified literals: $\forall x\, \mathrm{In}(x, \mathrm{t}_1)$
- Ground atoms: $\mathrm{At}(\mathrm{c}_1, \mathrm{t}_1)$

Simple Representation Languages

The complexity is unchanged for simple representation languages:

- Horn Theories
- Monotone DNF Theories

Complexity

The problem of evaluating the degree of belief of any query is #P-Hard

Simple Query Languages

The complexity is unchanged for very simple queries:

- Quantified literals: $\forall x\, \mathrm{In}(x, \mathrm{t}_1)$
- Ground atoms: $\mathrm{At}(\mathrm{c}_1, \mathrm{t}_1)$

Simple Representation Languages

The complexity is unchanged for simple representation languages:

- Horn Theories
- Monotone DNF Theories

KR Approach

A sharp separation between *knowledge acquisition* and *query evaluation*. Knowledge is given a priori in order to correctly represent an environment

L2R Approach

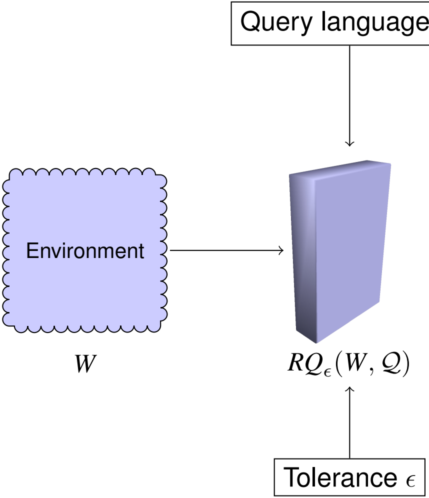
The dependence between *knowledge acquisition* and *query evaluation* is made explicit. Knowledge is acquired a posteriori, by experience, in order to efficiently reason about queries

KR Approach

A sharp separation between *knowledge acquisition* and *query evaluation*. Knowledge is given a priori in order to correctly represent an environment

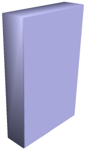
L2R Approach

The dependence between *knowledge acquisition* and *query evaluation* is made explicit. Knowledge is acquired a posteriori, by experience, in order to efficiently reason about queries

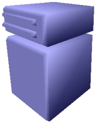


Learning Interface

Help the agent in finding a representation KB of W that is computationally efficient for some target query language $\mathcal{Q}$



$RQ_\epsilon(W, \mathcal{Q})$



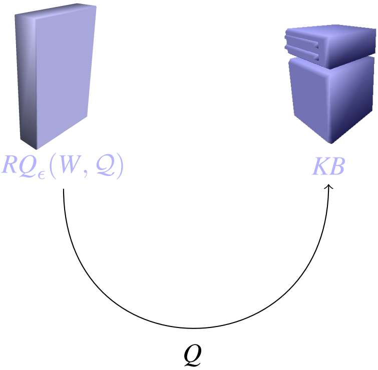
KB

Grace Period

Repeated game between the agent and the interface

1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \mathbf{Pr}_{KB}(Q)$
3. Receive $y = \mathbf{Pr}_W(Q)$

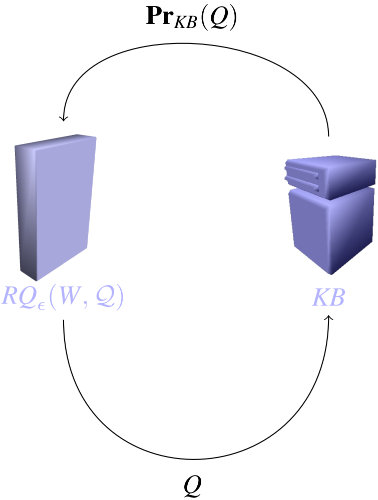
   If $L(y, \hat{y}) > \epsilon$ update KB



Grace Period

Repeated game between the agent and the interface

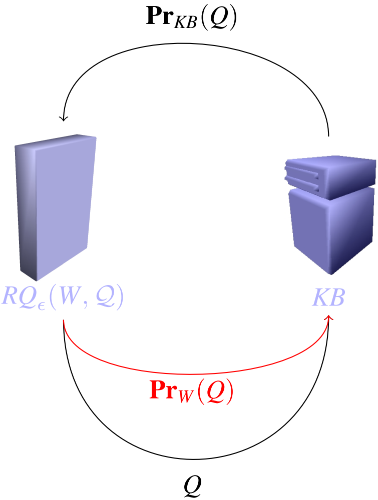
1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \mathbf{Pr}_{KB}(Q)$
3. Receive $y = \mathbf{Pr}_W(Q)$
   If $L(y, \hat{y}) > \epsilon$ update KB



Grace Period

Repeated game between the agent and the interface

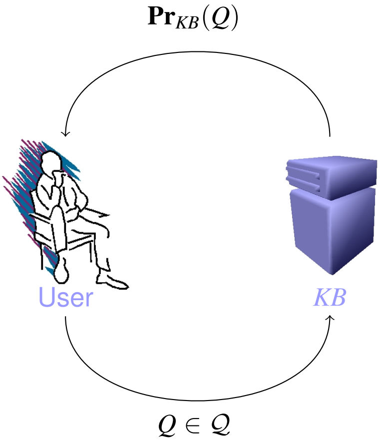
1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \mathbf{Pr}_{KB}(Q)$
3. Receive $y = \mathbf{Pr}_{W}(Q)$
   If $L(y, \hat{y}) > \epsilon$ update KB



Grace Period

Repeated game between the agent and the interface

1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \mathbf{Pr}_{KB}(Q)$
3. Receive $y = \mathbf{Pr}_W(Q)$
   If $L(y, \hat{y}) > \epsilon$ update KB



Operational Period

The reasoning performance of the agent is measured according to

- the same target query language $\mathcal{Q}$
- the same tolerance parameter ϵ

Polynomial Mistake Bound

For any possible sequence of queries in $\mathcal{Q}$, the total number of mistakes made by the L2R algorithm must be $\mathrm{poly}(|\mathcal{B}|, \frac{1}{\epsilon})$

Polynomial Complexity

For any possible query Q in $\mathcal{Q}$, the L2R algorithm must evaluate $\mathbf{Pr}_{KB}(Q)$ in $\mathrm{poly}(|\mathcal{B}|, |Q|, \frac{1}{\epsilon})$ time

Polynomial Mistake Bound

For any possible sequence of queries in $\mathcal{Q}$, the total number of mistakes made by the L2R algorithm must be $\text{poly}(|\mathcal{B}|, \frac{1}{\epsilon})$

Polynomial Complexity

For any possible query Q in $\mathcal{Q}$, the L2R algorithm must evaluate $\mathbf{Pr}_{KB}(Q)$ in $\text{poly}(|\mathcal{B}|, |Q|, \frac{1}{\epsilon})$ time

Outline

1st Idea

Use an exponentiated gradient strategy to update knowledge

2nd Idea

Use a weighted model counting approach to evaluate queries

Weighted Atoms

The vocabulary is extended with a set $\{\mathrm{q}_1, \mathrm{q}_2, \ldots\}$ of weighted atoms

- Standard Atom: $weight(\mathrm{a}) = 1$
- Weighted Atom: $weight(\mathrm{q}) \geq 0$

Weighted Interpretation

An interpretation that possibly contains weighted atoms

$$weight(I) = \prod_{A \in I} weight(A)$$

Weighted Formula

A relational expression F over the extended vocabulary

$$weight(F) = \sum_{I \models_{\min} F} weight(I)$$

Weighted Atoms

The vocabulary is extended with a set $\{\mathrm{q}_1, \mathrm{q}_2, \ldots\}$ of weighted atoms

- Standard Atom: $weight(\mathrm{a}) = 1$
- Weighted Atom: $weight(\mathrm{q}) \geq 0$

Weighted Interpretation

An interpretation that possibly contains weighted atoms

$$weight(I) = \prod_{A \in I} weight(A)$$

Weighted Formula

A relational expression F over the extended vocabulary

$$weight(F) = \sum_{I \models_{\min} F} weight(I)$$

Weighted Atoms

The vocabulary is extended with a set $\{\mathsf{q}_1, \mathsf{q}_2, \ldots\}$ of weighted atoms

- Standard Atom: $\mathit{weight}(\mathsf{a}) = 1$
- Weighted Atom: $\mathit{weight}(\mathsf{q}) \geq 0$

Weighted Interpretation

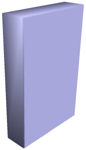
An interpretation that possibly contains weighted atoms

$$\mathit{weight}(I) = \prod_{A \in I} \mathit{weight}(A)$$

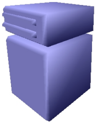
Weighted Formula

A relational expression F over the extended vocabulary

$$\mathit{weight}(F) = \sum_{I \models_{\min} F} \mathit{weight}(I)$$



$RQ_\epsilon(W, \mathcal{Q})$

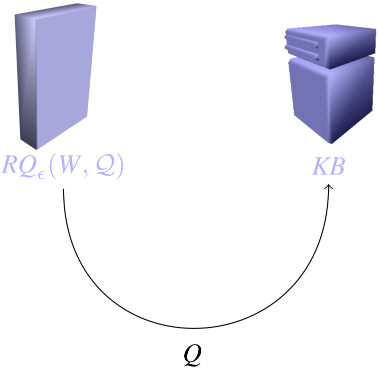


KB

The EG-L2R Algorithm

Start with $KB = \varnothing$. In each trial,

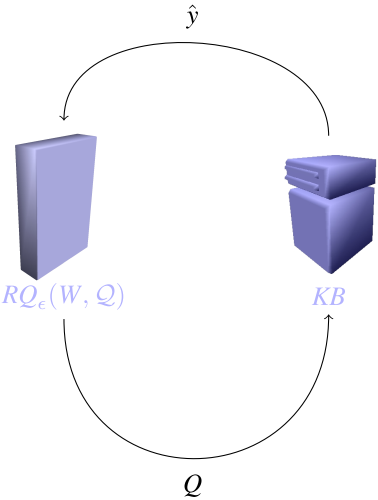
1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \frac{weight(KB \wedge Q)}{weight(KB)}$
3. Receive y. If $L(y, \hat{y}) > \epsilon$ then expand KB with $Q \leftrightarrow$ q where $weight(q) = e^{\eta(y - \hat{y})}$



The EG-L2R Algorithm

Start with $KB = \varnothing$. In each trial,

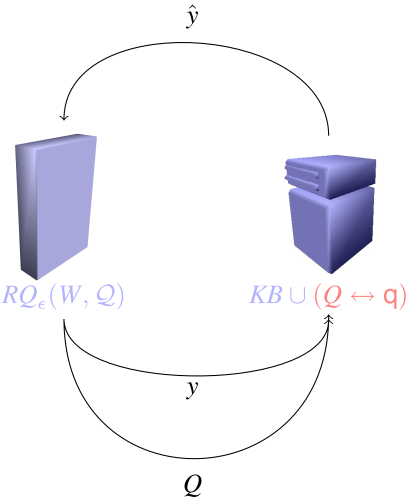
1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \frac{weight(KB \wedge Q)}{weight(KB)}$
3. Receive y. If $L(y, \hat{y}) > \epsilon$ then expand KB with $Q \leftrightarrow$ q where $weight(q) = e^{\eta(y - \hat{y})}$



The EG-L2R Algorithm

Start with $KB = \varnothing$. In each trial,

1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \dfrac{weight(KB \wedge Q)}{weight(KB)}$
3. Receive y. If $L(y, \hat{y}) > \epsilon$ then expand KB with $Q \leftrightarrow$ q where $weight(q) = e^{\eta(y-\hat{y})}$



The EG-L2R Algorithm

Start with $KB = \varnothing$. In each trial,

1. Receive a query $Q \in \mathcal{Q}$
2. Predict $\hat{y} = \dfrac{weight(KB \wedge Q)}{weight(KB)}$
3. Receive y. If $L(y, \hat{y}) > \epsilon$ then expand KB with $Q \leftrightarrow \mathsf{q}$ where $weight(q) = e^{\eta(y-\hat{y})}$

Polynomial Mistake Bound

The total number of mistakes made by EG-L2R is bounded by

$$\frac{|\mathcal{B}|}{2\epsilon}$$

Polynomial Size Representation

Let l be the largest size of any query in $\mathcal{Q}$. Then the size of KB is bounded by

$$\frac{l|\mathcal{B}|}{2\epsilon}$$

Outline

Quantified Atom

Atomic formula where each variable occurs in the scope of a quantifier $\forall$ or $\exists$

$\forall x \, \mathsf{In}(x, \mathsf{t}_1)$ $\qquad$ $\exists y \, \mathsf{At}(y, \mathsf{t}_1))$

Decomposable Query

Conjunction (or disjunction) of pairwise independent quantified literals

$$\forall x \, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y \, \mathsf{At}(y, \mathsf{t}_1)$$

Complexity

The number of models of any decomposable query Q can be evaluated in $O(|\mathcal{B}||Q|)$ time

Quantified Atom

Atomic formula where each variable occurs in the scope of a quantifier $\forall$ or $\exists$

$\forall x \, \mathsf{In}(x, \mathsf{t}_1)$ $\qquad$ $\exists y \, \mathsf{At}(y, \mathsf{t}_1))$

Decomposable Query

Conjunction (or disjunction) of pairwise independent quantified literals

$$\forall x \, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y \, \mathsf{At}(y, \mathsf{t}_1)$$

Complexity

The number of models of any decomposable query Q can be evaluated in $O(|\mathcal{B}||Q|)$ time

Quantified Atom

Atomic formula where each variable occurs in the scope of a quantifier $\forall$ or $\exists$

$\forall x\, \mathsf{In}(x, \mathsf{t}_1)$ $\exists y\, \mathsf{At}(y, \mathsf{t}_1))$

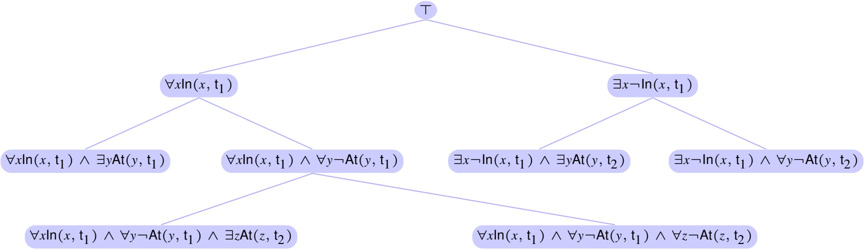
Decomposable Query

Conjunction (or disjunction) of pairwise independent quantified literals

$$\forall x\, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y\, \mathsf{At}(y, \mathsf{t}_1)$$

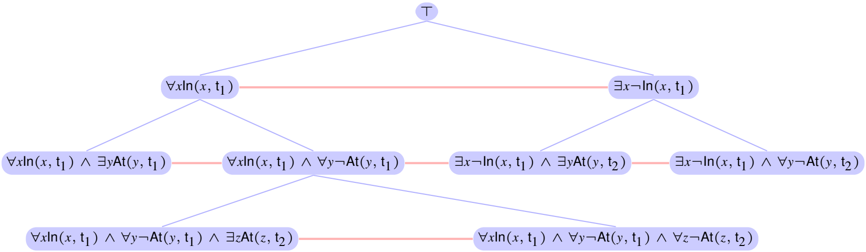
Complexity

The number of models of any decomposable query Q can be evaluated in $O(|\mathcal{B}||Q|)$ time



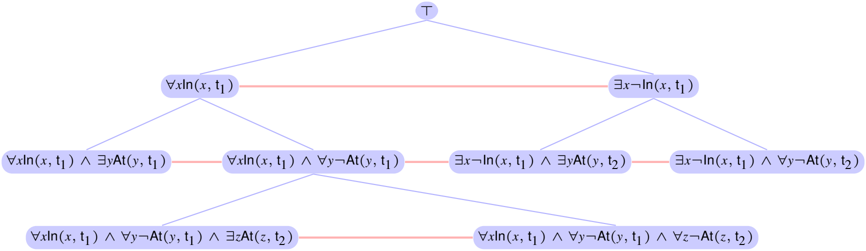
Hitting Language

Set of decomposable queries that are pairwise comparable under entailment or insatisfiable



Hitting Language

Set of decomposable queries that are pairwise comparable under entailment or insatisfiable

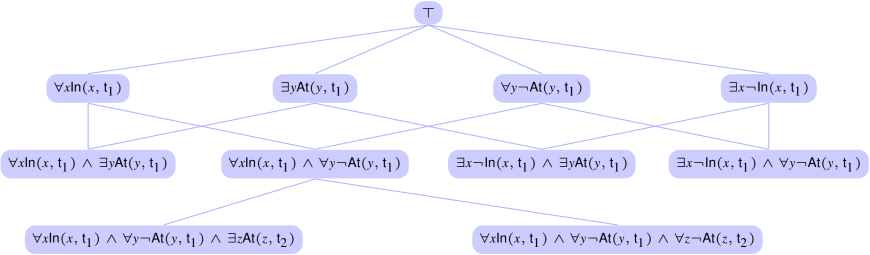


Hitting Language

Set of decomposable queries that are pairwise comparable under entailment or insatisfiable

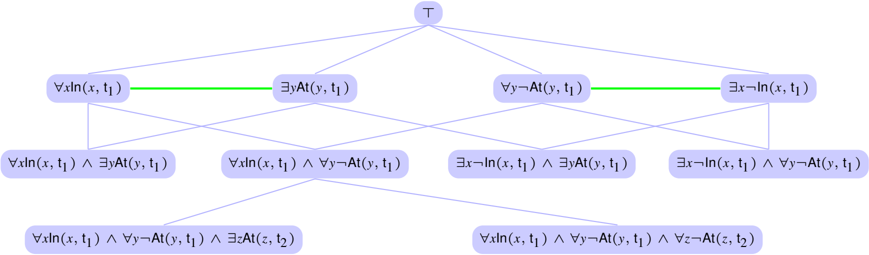
Learnability

There exists an efficient L2R algorithm for any probabilistic reasoning problem $(W, \mathcal{Q})$ where $\mathcal{Q}$ is an hitting query language



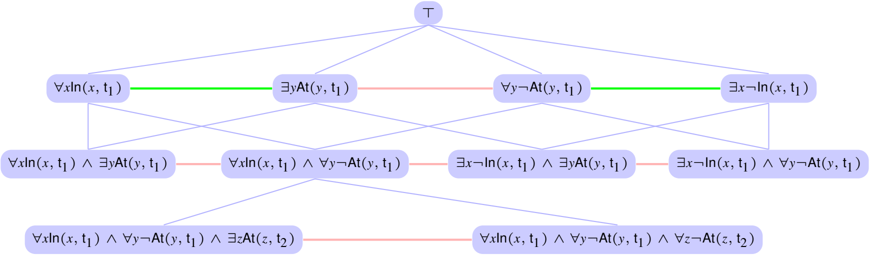
Cluster Language

Set $\mathcal{Q}$ of decomposable queries that are pairwise comparable or independent or insatisfiable



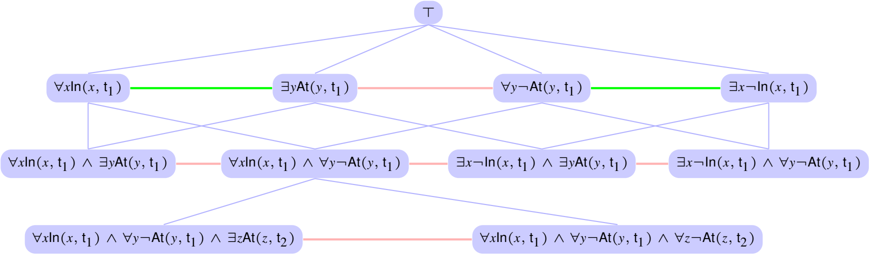
Cluster Language

Set $\mathcal{Q}$ of decomposable queries that are pairwise comparable or independent or insatisfiable



Cluster Language

Set $\mathcal{Q}$ of decomposable queries that are pairwise comparable or independent or insatisfiable



Cluster Language

Set $\mathcal{Q}$ of decomposable queries that are pairwise comparable or independent or insatisfiable

Learnability

There exists an efficient L2R algorithm for any probabilistic reasoning problem $(W, \mathcal{Q})$ where $\mathcal{Q}$ is a cluster query language

Outline

Application

Inductive Knowledge Compilation: Learning a computationally efficient representation of a logical theory (or Bayesian network) for some frequent queries supplied by users

Extensions

- Extending the scope of quantifiers

$$\forall x, y \, (\mathsf{In}(x, y) \rightarrow \mathsf{Truck}(x))$$

- Parameterized Cluster-Width

$$\forall x \, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y \mathsf{At}(y, \mathsf{t}_1) \qquad \exists x \, \mathsf{In}(x, \mathsf{t}_2) \wedge \mathsf{At}(\mathsf{c}_1, \mathsf{t}_1)$$

Application

Inductive Knowledge Compilation: Learning a computationally efficient representation of a logical theory (or Bayesian network) for some frequent queries supplied by users

Extensions

- Extending the scope of quantifiers

$$\forall x, y \, (\mathsf{In}(x, y) \rightarrow \mathsf{Truck}(x))$$

- Parameterized Cluster-Width

$$\forall x \, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y \mathsf{At}(y, \mathsf{t}_1) \qquad \exists x \, \mathsf{In}(x, \mathsf{t}_2) \wedge \mathsf{At}(\mathsf{c}_1, \mathsf{t}_1)$$