Towards a (Model) Theory for Probabilistic Logical Models

Manfred Jaeger

Aalborg University

Progic Questions

Progic questions:

Should probability and logic be combined at all?

Progic Questions

Progic questions:

Should probability and logic be combined at all?

Yes.

Progic Questions

Progic questions:

Should probability and logic be combined at all?

Yes.

How can probabilistic networks be used to simplify probabilistic logics?

Progic Questions

Progic questions:

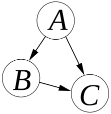
Should probability and logic be combined at all?

Yes.

How can probabilistic networks be used to simplify probabilistic logics?

- How do probabilistic networks relate to probabilistic logic?
- How do “first-order probabilistic networks” relate to first-order logic?
- what **are** “first-order probabilistic networks”?

PL-Models



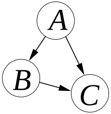

```
b(o).
c(o).
a(X) :- b(X),c(X).
```

+ F-O logic
(Relational data model)
(...)

+ Probabilities

“First-order relational probabilistic logic programming models”

PL-Models




```
b(o).
c(o).
a(X) :- b(X),c(X).
```

+ F-O logic
(Relational data model)
(...)

+ Probabilities

SLP RBN ProbLog
BLOG Prism
CLP(BN)
MLN PRM BLP

"First-order relational probabilistic logic programming models"

⇝ *PL-Models*

The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson,. . .):

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, . . .):

The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson,...):

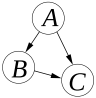
Syntax:

$$P(A \mid B) = 0.4$$
$$P(C \mid \neg A \wedge B) \leq 0.7$$
$$\ldots$$

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:



The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson,...):

Syntax:

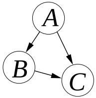
$$P(A \mid B) = 0.4$$
$$P(C \mid \neg A \wedge B) \leq 0.7$$
$$\ldots$$

Semantics:
Set of probability distributions over possible worlds.

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:



Semantics:
Unique probability distribution over possible worlds.

The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson,...):

Syntax:

$$P(A \mid B) = 0.4$$
$$P(C \mid \neg A \wedge B) \leq 0.7$$
$$\ldots$$

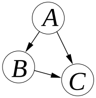
Semantics:
Set of probability distributions over possible worlds.

Inference:
Theorem proving

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:



Semantics:
Unique probability distribution over possible worlds.

Inference:
Model checking

[Halpern,Vardi: *Model-checking vs. Theorem proving: a manifesto*. 1991]

The Propositional Case

Logic:

Propositional probabilistic logic (Boole, Nilsson,...):

Syntax:

$$P(A \mid B) = 0.4$$
$$P(C \mid \neg A \wedge B) \leq 0.7$$
$$\ldots$$

Semantics:
Set of probability distributions over possible worlds.

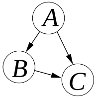
Inference:
Theorem proving

Complexity:
NP-complete

Graphical Models:

Bayesian networks (Lauritzen, Spiegelhalter, Pearl, ...):

Syntax:



Semantics:
Unique probability distribution over possible worlds.

Inference:
Model checking

Complexity:
NP-complete

[Halpern,Vardi: *Model-checking vs. Theorem proving: a manifesto*. 1991]

The First-Order Case

Logic:

First-order probabilistic logic (Halpern, Bacchus, . . .):

Syntax:
$P(\exists x f(x) = a) > 0.6$

Semantics:
Set of probability distributions over possible worlds

Inference:
Theorem proving

Complexity:
Π^1_∞-complete

The First-Order Case

Logic:

First-order probabilistic logic (Halpern, Bacchus, . . .):

Syntax:
$P(\exists x f(x) = a) > 0.6$

Semantics:
Set of probability distributions over possible worlds

Inference:
Theorem proving

Complexity:
Π^1_∞-complete

Graphical Models:

PL-models

Syntax:

Semantics: ?

Inference:
Model checking

Complexity: ?

Terminology

"*Model*"...

in logic:	a possible world
in probabilistic logic:	a probability distribution over possible worlds
in statistics:	a (parametric) class of probability distributions over possible worlds

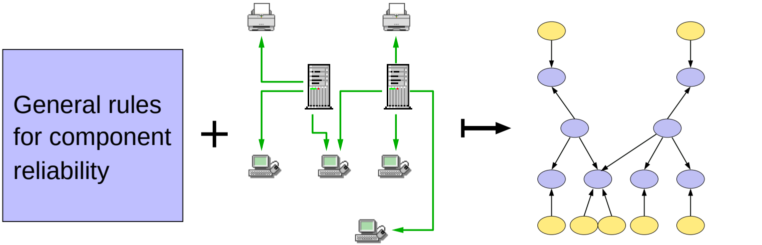
Terminology

"*Model*"...

in logic:	a possible world
in probabilistic logic:	a probability distribution over possible worlds
in statistics:	a (parametric) class of probability distributions over possible worlds

KBMC

"... graphical decision-modeling languages are still quite limited ... while they can describe the relationships among particular event instances, they cannot capture *general* knowledge about probabilistic relationships across classes of events." [Breese,Goldman,Wellman 1994]



General Failure Model   Concrete Network   Graphical Diagnostic Model

KBMC

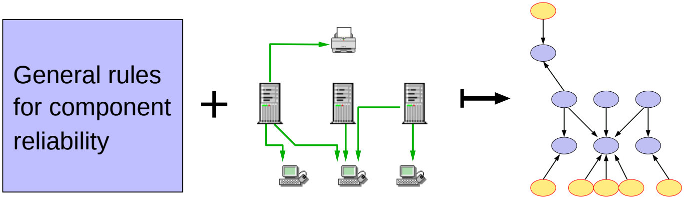
"... graphical decision-modeling languages are still quite limited ... while they can describe the relationships among particular event instances, they cannot capture *general* knowledge about probabilistic relationships across classes of events." [Breese,Goldman,Wellman 1994]

General rules for component reliability $+$ $\longmapsto$

General Failure Model | Concrete Network | Graphical Diagnostic Model

KBMC

"... graphical decision-modeling languages are still quite limited ... while they can describe the relationships among particular event instances, they cannot capture *general* knowledge about probabilistic relationships across classes of events." [Breese,Goldman,Wellman 1994]



General Failure Model

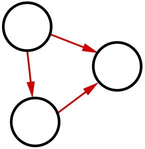
Concrete Network

Graphical Diagnostic Model

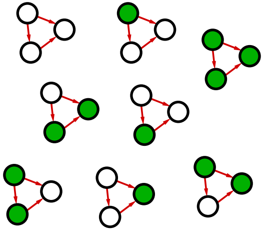
Formal Semantics

Probabilistic − Logic Model

+



⟼

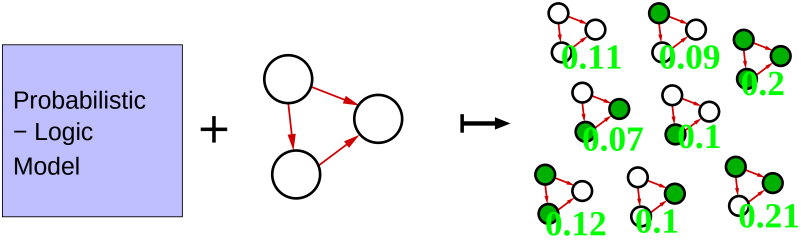


Possible world w

(input domain)

Probability distribution over expansions of w

Formal Semantics



Possible world *w*
(input domain)

Probability distribution over expansions of *w*

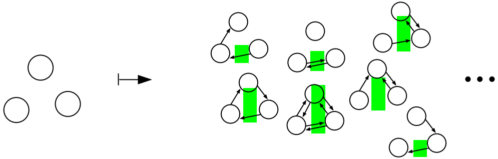
A PL-model is a mapping

$$w \mapsto P$$

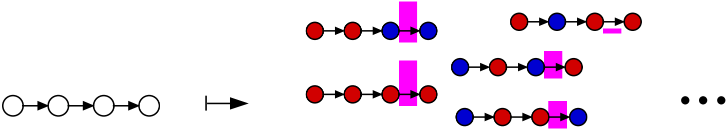
defined for an (infinite) class of input domains w.

Special Cases

Random Graphs

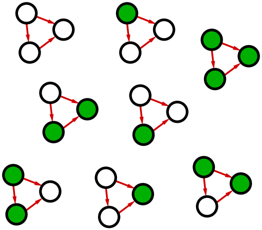


Markov Chains



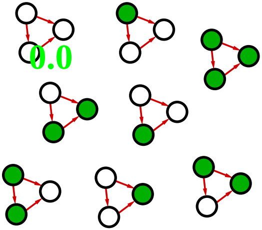
Syntax: MLNs

Markov Logic Networks [Richardson,Domingos 2006]
Software: `http://alchemy.cs.washington.edu/`



Syntax: MLNs

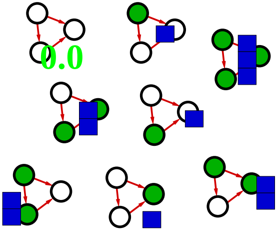
Markov Logic Networks [Richardson,Domingos 2006]
Software: `http://alchemy.cs.washington.edu/`



Hard constraints: $\exists x\, green(x)$

Syntax: MLNs

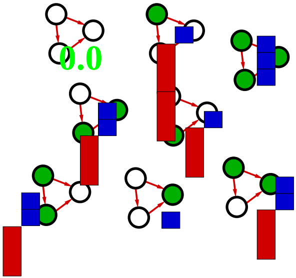
Markov Logic Networks [Richardson,Domingos 2006]
Software: `http://alchemy.cs.washington.edu/`



Hard constraints: $\exists x\ green(x)$
Weighted formulas: $green(x) : 0.5$

Syntax: MLNs

Markov Logic Networks [Richardson,Domingos 2006]
Software: `http://alchemy.cs.washington.edu/`



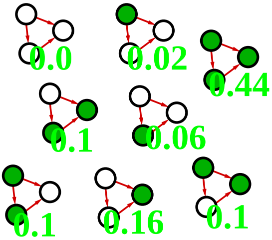
Hard constraints: $\exists x\ green(x)$

Weighted formulas: $green(x)$: 0.5

$green(x) \wedge edge(x,y) \wedge \neg green(y)$: -1.0

Syntax: MLNs

Markov Logic Networks [Richardson,Domingos 2006]
Software: `http://alchemy.cs.washington.edu/`



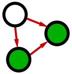
Hard constraints: $\exists x\, green(x)$

Weighted formulas: $green(x)$: 0.5

$green(x) \wedge edge(x,y) \wedge \neg green(y)$: -1.0

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`

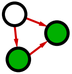


logic definition:

$$\exists y(green(y) \wedge edge(y,x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`

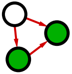


logic definition:

$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



logic definition:

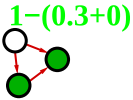
$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y,x)) =: P(green(x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



logic definition:

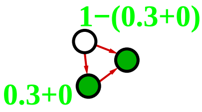
$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y,x)) =: P(green(x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



logic definition:

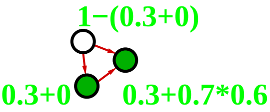
$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y,x)) =: P(green(x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



logic definition:

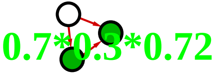
$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y,x)) =: P(green(x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



logic definition:

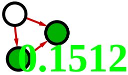
$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y,x)) =: P(green(x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



logic definition:

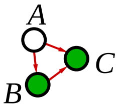
$$\exists y(green(y) \wedge edge(y, x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y, x)) =: P(green(x))$$

Syntax: RBNs

Relational Bayesian Networks [Jaeger 1997]
Software: `http://www.cs.aau.dk/~jaeger/Primula/`



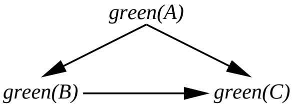
logic definition:

$$\exists y(green(y) \wedge edge(y,x)) \leftrightarrow: green(x)$$

probabilistic definition:

$$0.3 + 0.7 \cdot noisy\text{-}or[0.6].y(green(y) \wedge edge(y,x)) =: P(green(x))$$

Dependency of ground atoms:



RBN/MLN comparison

MLN	RBN
“Declarative” use of F-O formulas	“Definitional” use of F-O formulas
Undirected dependencies between ground atoms	Directed dependencies between ground atoms (acyclicity conditions!)
Best for descriptive/declarative modeling	Best for causal or (incrementally) generative modeling

Formal Framework

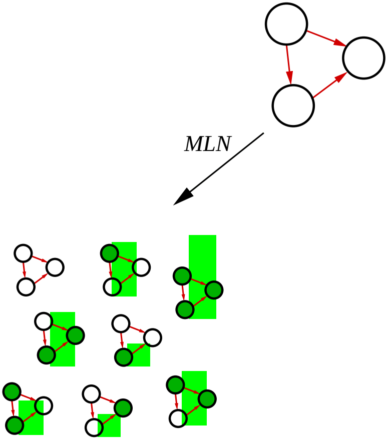
General idea:
$\mathcal{XYZ}$ is at least as expressive as $\mathscr{ABC}$ if every PL-model (i.e. mapping $w \mapsto P$) definable in $\mathscr{ABC}$ is definable in $\mathcal{XYZ}$.

Issues

- Need to fit the semantics of $\mathcal{XYZ}$, $\mathscr{ABC}$ into PL-model framework
- Asking for identity of models is usually too much – only feasible: $\mathscr{ABC}$-models can be *embedded* in $\mathcal{XYZ}$-models.

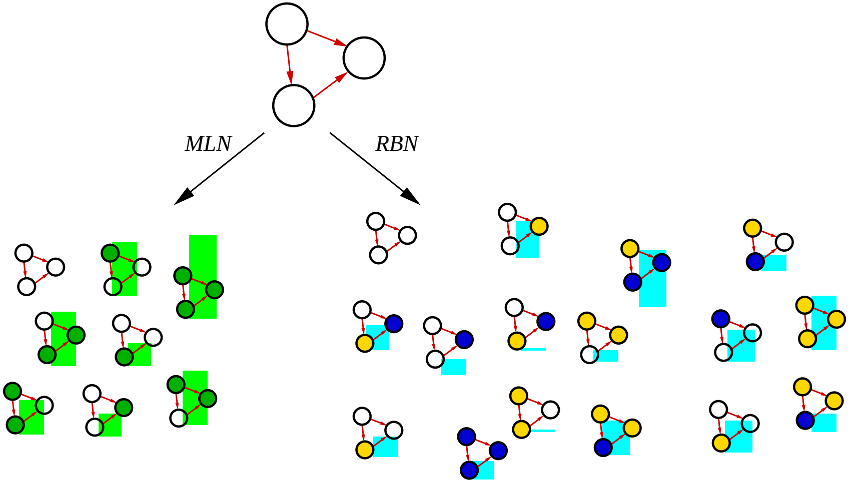
MLN/RBN Comparison

Every MLN-definable model is *conditionally embedded* in a RBN-definable model:



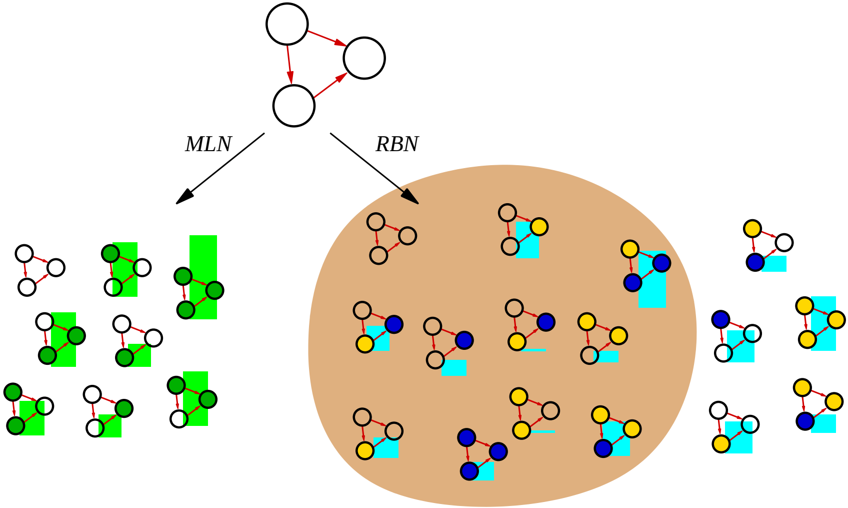
MLN/RBN Comparison

Every MLN-definable model is *conditionally embedded* in a RBN-definable model:



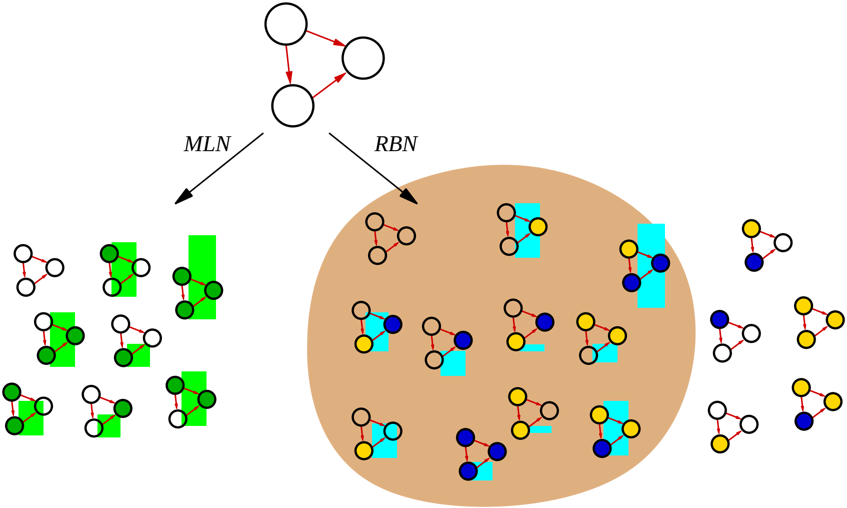
MLN/RBN Comparison

Every MLN-definable model is *conditionally embedded* in a RBN-definable model:



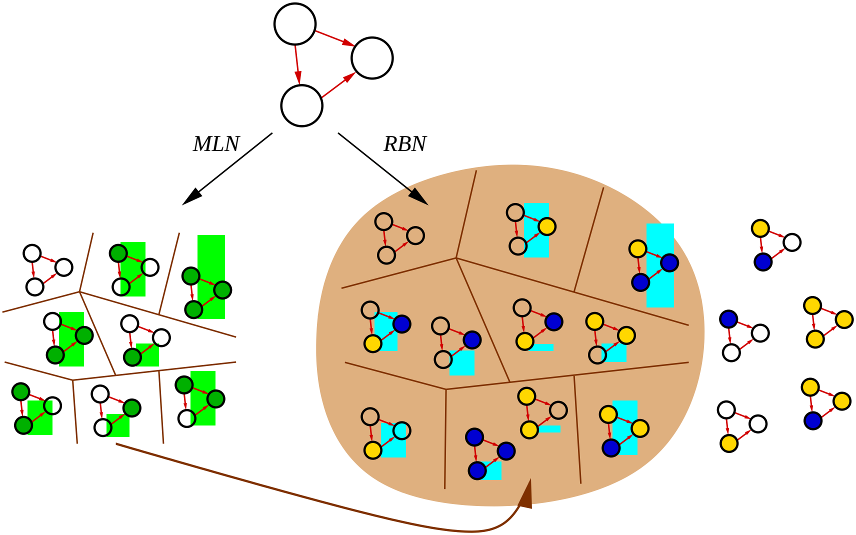
MLN/RBN Comparison

Every MLN-definable model is *conditionally embedded* in a RBN-definable model:



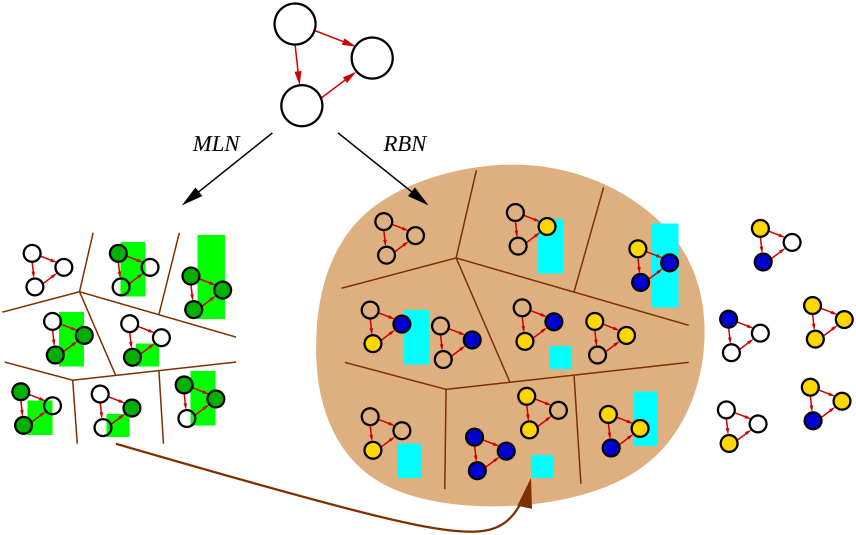
MLN/RBN Comparison

Every MLN-definable model is *conditionally embedded* in a RBN-definable model:



MLN/RBN Comparison

Every MLN-definable model is *conditionally embedded* in a RBN-definable model:



Elementary Inference Problem

Given

- a **model** $\mathcal{M}$
- an (input) **domain** (possible world) D
- truth assignments for ground atoms in expansions of D (**evidence atoms**):

$$r_1(\boldsymbol{d}_1) = \tau_1, \ldots, r_n(\boldsymbol{d}_n) = \tau_n$$

- truth assignments for ground atoms in expansions of D (**query atoms**):

$$r_{n+1}(\boldsymbol{d}_{n+1}) = \tau_{n+1}, \ldots, r_{n+k}(\boldsymbol{d}_{n+k}) = \tau_n$$

compute

$$P(r_{n+1}(\boldsymbol{d}_{n+1}) = \tau_{n+1}, \ldots, r_{n+k}(\boldsymbol{d}_{n+k}) \mid r_1(\boldsymbol{d}_1) = \tau_1, \ldots, r_n(\boldsymbol{d}_n) = \tau_n)$$

for P defined by $\mathcal{M}$ on expansions of D.
(or **decide** whether $P(\ldots \mid \ldots) > 0$).

Example: given that *responding(server003)=false*, what is the probability that *cable_intact(terminal152,server003)=false*?

Complexity of Elementary Inference

Complexity issues:

- complexity in terms of complexity of $\mathcal{M}$
- complexity in terms of size of D

Complexity of Elementary Inference

Complexity issues:

- complexity in terms of complexity of $\mathcal{M}$
- complexity in terms of size of D

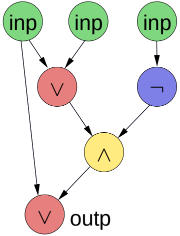
Results:

- Elementary inference (decision version) is *NP*-hard (in size of D) for any modeling language with [expressivity requirement]
- ... *NP*-complete for *MLNs* and *RBNs*

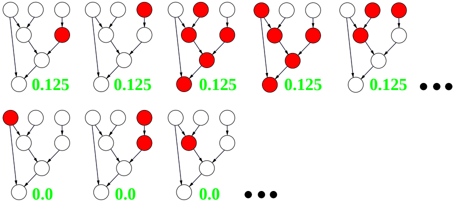
Expressivity Requirement

Boolean satisfiability encoding (cf. [Cooper 1990]):

Boolean function as input domain:



Uniform distribution over consistent value assignments:



$$\text{Formula satisfiable} \leftrightarrow P(\mathit{value}(\mathit{outp}) = \mathit{true}) > 0$$

MLNs and RBNs can encode Boolean satisfiability!

“Verifi cation”

MLN defines a distribution over expansions of D iff

the hard constraints are satisfiable in expansions of D

RBN defines a distribution over expansions of D iff

the dependency relation induced by RBN is acyclic

“Verification” Problem:

Given

- a **model** $\mathcal{M}$
- a class $\mathscr{D}$ of input domains (e.g. axiomatized in some logic)

decide

whether $\mathcal{M}$ defines for all $D \in \mathscr{D}$ a distribution over expansions of D.

Undecidability of Verifi cation

$\mathscr{D}_\emptyset$: finite input domains without relations.

Result for MLNs

It is undecidable whether a MLN defines a distribution for each input $D \in \mathscr{D}_\emptyset$.

Proof: it is undecidable whether a first-order sentence ϕ is satisfiable over all finite cardinalities

Undecidability of Verifi cation

$\mathscr{D}_{\emptyset}$: finite input domains without relations.

Result for MLNs

It is undecidable whether a MLN defines a distribution for each input $D \in \mathscr{D}_{\emptyset}$.

Proof: it is undecidable whether a first-order sentence ϕ is satisfiable over all finite cardinalities

Conjectures for RBNs

$\mathscr{D}_{un}$: finite input domains with only unary relations.

It is decidable whether a RBN defines a distribution for each input $D \in \mathscr{D}_{un}$.

acyclicity of dependencies can be expressed in monadic transitive closure logic

$\mathscr{D}_{bin}$: finite input domains with one binary relation.

It is undecidable whether a RBN defines a distribution for each input $D \in \mathscr{D}_{bin}$.

Global Inference

"Global Inference" Problem:

Given

- a **model** $\mathcal{M}$
- a first-order sentence ϕ

decide

whether $P(\phi) > 0$ for all input domains for which a distribution is defined by $\mathcal{M}$

Global Inference

“Global Inference” Problem:

Given

- a **model** $\mathcal{M}$
- a first-order sentence ϕ

decide

whether $P(\phi) > 0$ for all input domains for which a distribution is defined by $\mathcal{M}$

Results

The global inference problem is undecidable for MLNs and RBNs

Progic Question Again

How can probabilistic networks be used to simplify probabilistic logics?

They can direct our attention to tractable (model checking) problems!

Conclusion

Summary:

To better understand the plethora of probabilistic logical modeling languages:

- introduced semantic concept of PL-models
- obtained first expressivity result
- obtained some complexity results

Next:

- extend expressivity analysis to other languages (*Bayesian Logic Programs*, *Prism*,...)
- investigate *learnability* issues

Acknowledgements:

Kristian Kersting, Luc De Raedt, ..., discussion groups at SRL and Dagstuhl workshops