Linking Open-world Knowledge Bases using Nonmonotonic Rules

Mantas Šimkus (joint work with Thomas Eiter)

Institute of Information Systems,
TU Wien

LPNMR 2015

Lexington, Kentucky, U.S.A.

Motivation

- Integrating knowledge from various sources is a recurring problem in AI
- The problem is often addressed by multi-context systems (MCSs)
- MCSs interlink individual knowledge bases (KBs) (called contexts) with bridge rules
- Existing MCSs provide limited support for KBs with model-based semantics under an open-world view
    - only limited support for Description Logic (DL) ontologies

Example

Assume we have a pair κ_1, κ_2 of contexts, where

- κ_1 is an ontology that describes restaurants and food, and
- κ_2 is a geospatial database.

$$
\begin{aligned}
1:\mathsf{BookingAdvised}(x) \leftarrow\ & 1:\mathsf{Restaurant}(x), \\
& 2:\mathsf{inside}(x, \text{“Manhattan”}), \\
& 1:\ \mathit{not}\ \mathsf{type}(x, \text{“Fast Food”}).
\end{aligned}
$$

Query: return all restaurants for which reservation is advisable.

Contribution

- We introduce knowledge base networks (KBNs)
- A KBN is a collection of
  - KBs with open-world semantics in terms of first-order structures
  - interlinked by non-monotonic bridge rules
- KBNs are equipped with a stable model semantics
- Basic entailment in KBNs is decidable whenever it is in the individual KBs
- For DL ontology networks (ONs) reasoning is reducible to reasoning in dl-programs
  - ONs subsume hybrid MKNF

Interpretations

- We assume disjoint sets
  - Const of constants
  - Rel of relation symbols of *arity* ≥ 0
  - Vars of variables
- We make the standard name assumption (SNA)

Definition (interpretations)

Assume a finite set σ of constants. A σ-interpretation is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where

- $\Delta^{\mathcal{I}} \neq \emptyset$ is a set such that $\sigma \subseteq \Delta^{\mathcal{I}}$, called the domain of $\mathcal{I}$, and
- $\cdot^{\mathcal{I}}$ is a function that assigns to each n-ary relation symbol $R \in$ Rel an n-ary relation $R^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$

Knowledge Bases (Contexts)

- We use a general notion of KBs
- The semantics to each KB is given by associating to it a set of interpretations (called models)

Definition (knowledge bases)

- Let $\mathcal{K}$ denote an infinite set of objects, called knowledge bases (KBs).
- We assume a binary “models” relation $\models$ between interpretations and KBs.
- If $\mathcal{I} \models \kappa$, then $\mathcal{I}$ is called a model of the KB $\kappa \in \mathcal{K}$.

Atoms

- Positive literals (or, atoms) have the form $R(\vec{t})$, where $R \in \mathsf{Rel}$ is n-ary and $\vec{t}$ is an n-tuple of terms
- Negative literals are expressions $not\, B$, where B is an atom
- A literal is ground, if it has no variables

Definition (satisfaction of atoms)

Assume a KB $\kappa \in \mathcal{K}$, a σ-interpretation $\mathcal{I}$, a ground atom $B = R(\vec{t})$, and a set I of ground atoms. We extend the $\models$ relation as follows:

- $\mathcal{I} \models R(\vec{t})$ if $\vec{t} \in R^{\mathcal{I}}$ and $t_i \in \sigma$ for every term t_i in $\vec{t}$;
- $\mathcal{I} \models I$ if $\mathcal{I} \models B$ holds for every $B \in I$;
- $\kappa, I \models B$ if $\mathcal{I} \models B$ for every $\mathcal{I}$ such that $\mathcal{I} \models \kappa$ and $\mathcal{I} \models I$;
- $\kappa \models B$ if $\kappa, \emptyset \models B$.

Bridge Rules

- syntactically similar to rules in logic programming
- indexed literals to refer to different KBs of a KBN

Definition

An indexed literal is an expression $k : L$, where $k \geq 1$ is an integer and L is a literal. A bridge rule ρ is an expression of the form

$$B \leftarrow L_1, \ldots, L_n,$$

where B is an indexed atom, and $L_1, \ldots, L_n$ are indexed literals.

Example

$$\begin{aligned} 1 : \mathsf{BookingAdvised}(x) \leftarrow\ & 1 : \mathsf{Restaurant}(x), \\ & 2 : \mathsf{inside}(x, \text{“Manhattan”}), \\ & 1 : \textit{not}\ \mathsf{type}(x, \text{“Fast Food”}). \end{aligned}$$

Knowledge Base Networks

Definition (knowledge base networks)

A knowledge base network (KBN) is a tuple $\mathcal{N} = (\vec{\kappa}, \mathcal{R}, \sigma)$, where

- $\vec{\kappa} = \langle \kappa_1, \ldots, \kappa_n \rangle$ is a tuple of KBs
- $\mathcal{R}$ is a finite set of bridge rules
- $\sigma \subseteq$ Const is a finite set of constants

If "*not*" does not appear in $\mathcal{R}$, then $\mathcal{R}$ and $\mathcal{N}$ are called positive.

Epistemic Interpretations

- Our semantics assigns to each KB a subset of its models such that:
  - the bridge rules are satisfied
  - models of KBs are maximally preserved
  - "penalizing" violations of default negation
- Sets of interpretations are denoted $\mathbb{I}, \mathbb{I}', \mathbb{I}_1, \mathbb{I}_2$, ...

Definition (epistemic interpretation)

An epistemic interpretation for a KBN $\mathcal{N} = (\vec{\kappa}, \mathcal{R}, \sigma)$ is a tuple $\vec{\mathbb{I}} = (\mathbb{I}_1, \ldots, \mathbb{I}_{|\vec{\kappa}|})$ of nonempty sets of σ-interpretations.

Epistemic Models

Definition (evaluation of atoms)

Assume a set $\mathbb{I}$ of interpretations and a ground atom B.

We write $\mathbb{I} \models B$ if $\mathcal{I} \models B$ for all $\mathcal{I} \in \mathbb{I}$.

We write $\mathbb{I} \models \mathit{not}\, B$ if $\mathbb{I} \not\models B$.

For $\vec{\mathbb{I}} = (\mathbb{I}_1, \ldots, \mathbb{I}_n)$ and a literal $k:L$, we write $\vec{\mathbb{I}} \models k:L$ if $\mathbb{I}_k \models L$.

Definition (epistemic models)

Assume a KBN $\mathcal{N} = (\vec{\kappa}, \mathcal{R}, \sigma)$ and an epistemic interpretation $\vec{\mathbb{I}}$ for $\mathcal{N}$.

We write $\vec{\mathbb{I}} \models \mathcal{N}$ if

- $\vec{\mathbb{I}} \models \mathcal{R}$ and
- $\mathcal{I} \models \kappa_i$ for every $1 \leq i \leq |\vec{\kappa}|$ and $\mathcal{I} \in \mathbb{I}_i$.

If $\vec{\mathbb{I}} \models \mathcal{N}$, then $\vec{\mathbb{I}}$ is an epistemic model of $\mathcal{N}$.

Dealing with Default Negation (the Reduct)

- minimize the truth of atoms that occur negatively in rule bodies
- we adapt for our purposes the Gelfond-Lifschitz reduct [Gelfond and Lifschitz(1988)].

Definition (reduct)

Given a KBN $\mathcal{N} = (\vec{\kappa}, \mathcal{R}, \sigma)$ and an epistemic interpretation $\vec{\mathbb{I}}$ for $\mathcal{N}$, we denote by $\mathcal{R}^{\vec{\mathbb{I}}}$ the set of rules obtained from ground$(\mathcal{R}, \sigma)$ by deleting

- every rule with a body literal $k : \mathit{not}\, B$ s.t. $\vec{\mathbb{I}} \models k : B$;
- every negative literal in the remaining rules.

Then, $\mathcal{N}^{\vec{\mathbb{I}}} = (\vec{\kappa}, \mathcal{R}^{\vec{\mathbb{I}}}, \sigma)$ is called the reduct of $\mathcal{N}$ w.r.t. $\vec{\mathbb{I}}$.

Stable Models

Definition (stable models)

An epistemic interpretation $\vec{\mathbb{I}}$ for a KBN $\mathcal{N}$ is a stable model of $\mathcal{N}$, if

- $\vec{\mathbb{I}} \models \mathcal{N}$ and
- there is no $\vec{\mathbb{I}}'$ such that $\vec{\mathbb{I}} \subset \vec{\mathbb{I}}'$ and $\vec{\mathbb{I}}' \models \mathcal{N}^{\vec{\mathbb{I}}}$.

We say $\mathcal{N}$ is consistent, if $\mathcal{N}$ has a stable model.

Positive Knowledge Base Networks

- In standard ASP, a consistent positive program, i.e. program without an occurrence of "*not*", has a unique stable model
- The same holds for any consistent positive KBN $\mathcal{N}$

Theorem

Every positive KBN $\mathcal{N}$ has at most one stable model.

Proof.

The component-wise union of two epistemic models of $\mathcal{N}$ is again an epistemic model of $\mathcal{N}$. □

Reasoning in Knowledge Base Networks

Problem

INPUT: a KBN $\mathcal{N}$ and a ground indexed atom $k : R(\vec{t})$

QUESTION: $\vec{\mathbb{I}} \models k : R(\vec{t})$ holds for every stable model $\vec{\mathbb{I}}$ of $\mathcal{N}$?

- Stable model may contain infinitely many interpretations
  - we resort to finite representation
  - relatively standard approach (see e.g. [Motik and Rosati(2010)])
- Stable models are represented using finite sets of ground indexed atoms

Decidability

Theorem (decidability)

Atom entailment from KBNs is decidable whenever atom entailment is decidable for individual KBs.

Theorem (general upper bound)

For KBNs over contexts with atom entailment in $\mathcal{C}$, non-entailment of ground atoms in $\textsc{NExp}^{\mathcal{C}}$.

Description Logics Ontologies

- KBNs can be used to intelink ontologies expressed in DLs
- A DL ontology $\mathcal{O}$ can be simply seen as a FOL theory built using
  - constants (a.k.a. individuals),
  - unary and binary relation symbols (a.k.a. concept names role names, respectively).
- Various syntax restrictions on $\mathcal{O}$ give rise to a variety of DLs
  - $\mathcal{SHOIQ}$, $\mathcal{SHIQ}$, $\mathcal{EL}$, DL-Lite, etc.
- The semantics for a DL ontology $\mathcal{O}$ can be given in terms of interpretations

Ontology Networks

Definition

An ontology network (ON) is a KBN $\mathcal{N} = (\vec{\mathcal{O}}, \mathcal{R}, \sigma)$ such that $\vec{\mathcal{O}}$ is a tuple of DL ontologies, and σ is the set of constants in $\vec{\mathcal{O}}$ and $\mathcal{R}$.

Theorem (Complexity (bounded arities / data complexity))

Description Logics	*Positive KBNs*	*Normal KBNs*
DL-Lite, $\mathcal{EL}$	co-NP / P	Σ_2^P / NP
$\mathcal{SHIQ}$	Exp / NP	Exp / Σ_2^p
$\mathcal{SHOIQ}$	NExp / NP	NP^{NExp} / Σ_2^p

Encoding into DL-programs

- The finite representation opens the way to implement reasoning in KBNs by a translation into (extensions of) answer set programming.
- In the paper we show how entailment of ground atoms from a given ON can be reduced to entailment of atoms in a dl-program of Eiter et al.
- Singleton ONs correspond to hybrid MKNF KBs (ground or DL-safe).
- We obtain an embedding of hybrid MKNF to dl-programs.

Conclusion

- We believe that KBNs is a powerful formalism for interlinking open-world first-order KBs
  - general contexts with open-world semantics
  - nonmonotonic rules
  - good decidability properties
- The main challenges for future work:
  - adding closed-world contexts
  - notions of stratification
  - compiling away the bridge rules