Disjunctive Logic Programs with Existential Quantification in Rule Heads

Jia-Huai You[1], Heng Zhang[2], Yan Zhang[2]
[1]University of Alberta, Canada
[2]University of Western Sydney, Australia

August 25, 2013

Horn rules (function-free) with existential variables in rule heads.

$$\exists Y \; father(X, Y) \leftarrow person(X)$$
$$person(Y) \leftarrow father(X, Y)$$

called a tuple generating dependency (TGD) in DBs; QA against universal model by the chase procedure [Fagin et al. 2005].

- Data exchange, incomplete databases, inclusion dependencies in databases, etc.
- Capture and generalize some low complexity description logics [Calì et al. 2008, ...].
- Previous works either assume ontological knowledge is not defeasible [Alviano et al. 2012], or only treat stratified [Cali et al. 2009] or well-founded negation [Gottlob et al 2012, 2013].

Nonmonotonic Reasoning with/about Ontological Knowledge

- Combining ASP with DL
  - DL + rules [Rosati 2006, Lukasiewicz 2010, ...]
  - DL-programs [Eiter et al. 2008]
  - MKNF [Motik and Rosati 2010]
  - ......
- Reasoning with Defeasible Ontologies
  - DLs with circumscription [Bonatti et al. 2006]
  - Defeasible (inheritance-based) DLs [Casini and Straccia 2013]

Disjunctive Programs with Existential Rules (E-Disjunctive Programs)

Finite sets of rules of the form

$$\forall \mathbf{X}\ \exists \mathbf{Y}\ \alpha_1; \ldots; \alpha_m \leftarrow \beta_1, \ldots, \beta_k, \text{not } \gamma_1, \ldots, \text{not } \gamma_n$$

where α_i, β_i and γ_i are atoms, and variables in $\mathbf{Y}$ may appear only in α_i.

- Semantics defined under general stable models; but properties rarely studied
- What can we do with existential variables in rule heads?
- Succinct encoding of knowledge
- Potential in representing defeasible ontological knowledge all in a uniform language.

A Simple Example: K-colorability

$$\exists Y\ set(X, Y) \leftarrow vertex(X)$$
$$\leftarrow edge(X, Y), set(X, C), set(Y, C)$$
$$\leftarrow set(X, Y), \text{not } color(Y)$$

Given a graph and k colors, the graph is k-colorable iff the program has a stable model.

That "a vertex is colored with exactly one color" is by a free ride of minimization.

Example: Strategic Companies

$$\exists Z \quad strat\_from(Z, X) \leftarrow prod\_by(X, Y)$$
$$\leftarrow strat\_from(Z, X), \text{not } prod\_by(X, Z)$$
$$strat(Y) \leftarrow strat\_from(Y, X)$$
$$strat(W) \leftarrow contr\_by(W, X, Y, Z), strat(X), strat(Y), strat(Z)$$

- $prod\_by(X, Y)$: Company Y produces good X
- $strat\_from(Y, X)$: Company Y is strategic because of good X

Example: Representing Defeasible Ontology

Staff members are either users or non-users; users who are not known to present a security threat are given access with some access level. (An example modified from [Bonatti et al. 2011])

$$
\begin{array}{l}
user(X) \vee nonuser(X) \leftarrow staff(X) \\
\exists Y \ accessLevel(X, Y) \leftarrow user(X), \text{not } secThreat(X) \\
secThreat(X) \leftarrow blackListed(X) \\
\leftarrow accessLevel(X, Y), \text{not } secLevel(Y)
\end{array}
$$

Given appropriate facts, one can query, e.g.,

$$Q : \ \exists X \, \exists Y \ accessLevel(X, Y) \wedge blackListed(X)$$

for which we can add a constraint to program Π

$$r_Q : \leftarrow accessLevel(X, Y), blackListed(X)$$

so that $\Pi \models_{SM} Q$ iff $\Pi \cup \{r_Q\}$ has no stable models.

Technical Results of This Paper

- Shows a simple definition of stable models based on the notion of a deductive closure.
- Stable models of these programs can also be characterized by progression [Zhang and Zhou 2011], which, among other benefits, yields an abstract condition ensuring the small predicate property.
- Identify a decidable fragment where the domain for a ∃-variable may be unknown.

A Simple Definition of Stable Model

Definition

Let Π be an E-disjunctive program and $\mathcal{M}$ a structure of $\tau(\Pi)$. $\mathcal{M}$ is a justified stable model of Π if $\mathcal{M}$ is a minimal set X satisfying the condition: for any $r \in \Pi$ and any assignment η of $\mathcal{M}$, if $X \cup \mathcal{M}^{-} \models Body(r)\eta$, then for some assignment ϑ of $\mathcal{M}$ and $\alpha \in Head(r)$, $(\alpha\eta|_{\mathbf{x}})\vartheta \in X$.

Theorem

Let Π be an E-disjunctive program and $\mathcal{M}$ a structure of $\tau(\Pi)$. $\mathcal{M}$ is a justified stable model of Π iff $\mathcal{M}$ is a stable model of $\pi(\Pi)$ (in the sense of [Ferraris et al 2011]).

This is a bit technical but the idea is simple: Given a program Π and a structure $\mathcal{M}$, based on the fixed $\mathcal{M}^-$, we iteratively construct an evolution sequence, $\sigma^0_{\mathcal{M}}(\Pi), \cdots, \sigma^t_{\mathcal{M}}(\Pi), \cdots$, of structures of $\tau(\Pi)$.

Definition

Let S be a set and $\Phi = \{S_1, ..., S_i, ...\}$ a collection of sets such that $S_i \subseteq S$, for all i. A subset $H \subseteq S$ is said to be a hitting set of Φ if for all i, $H \cap S_i \neq \emptyset$. Furthermore, H is said to be a minimal hitting set of Φ if H is a hitting set of Φ and there is no $H' \subset H$ such that H' is also a hitting set of Φ.

Progression

Definition

An evolution sequence of Π based on $\mathcal{M}$, denoted as $\sigma_{\mathcal{M}}(\Pi)$, is a sequence $\sigma^0_{\mathcal{M}}(\Pi), \cdots, \sigma^t_{\mathcal{M}}(\Pi), \cdots$, of structures of $\tau(\Pi)$, defined inductively as follows

1. $\sigma^0_{\mathcal{M}}(\Pi) = \mathcal{E}$, where $\mathcal{E}$ is the structure of $\tau(\Pi)$ in which all interpretations of predicates are the empty set;
2. $\sigma^{t+1}_{\mathcal{M}}(\Pi) = \sigma^t_{\mathcal{M}}(\Pi) \cup H^t$, where there exists $H^t \subseteq \mathcal{M}$ such that it is a minimal hitting set of the collection Φ^t of the following sets:

$$\bigcup_{\theta \in \Psi} (Head(r)\eta|_{\mathbf{X}})\theta|_{\mathbf{Y}} \tag{1}$$

where r is a rule in Π and η an assignment of $\mathcal{M}$ such that $(1) \cap \sigma^t_{\mathcal{M}}(\Pi) = \emptyset$, $\sigma^t_{\mathcal{M}}(\Pi) \models Pos(r)\eta$, and $\mathcal{M} \models Neg(r)\eta$; and $\sigma^{t+1}_{\mathcal{M}}(\Pi) = \sigma^t_{\mathcal{M}}(\Pi)$ if H^t does not exist.

We denote $\sigma^{\infty}_{\mathcal{M}}(\Pi) = \bigcup_{i=0}^{\infty} \sigma^i_{\mathcal{M}}(\Pi)$.

Example

Let Π be

$$h(a) \leftarrow$$
$$\exists Y \; p(X,Y); q(X,Y) \leftarrow \text{not } h(X)$$
$$\exists Y \; p(X,X); q(Y,Y) \leftarrow h(X)$$

Let $\mathcal{M}$ be a structure of $\tau(\Pi)$ where $Dom(\mathcal{M}) = \{1,2\}$, $a^{\mathcal{M}} = 1$, and $\mathcal{M} = \{h(1), q(2,2)\}$. $\mathcal{M}$ is a stable model of Π, which can be constructed by an evolution sequence:

- $\sigma^0_{\mathcal{M}}(\Pi) = \emptyset$,
- $\sigma^1_{\mathcal{M}}(\Pi) = \{h(1), q(2,2)\}$,
- $\sigma^2_{\mathcal{M}} = \sigma^1_{\mathcal{M}}$ ......

Theorem

Let Π be an E-disjunctive program and $\mathcal{M}$ a structure of $\tau(\Pi)$. $\mathcal{M}$ is a stable model of Π iff for all evolution sequences σ of Π based on $\mathcal{M}$, $\sigma^\infty_{\mathcal{M}}(\Pi) = \mathcal{M}$.

An Abstract Class of Programs with Small Predicate Property (SPP)

Definition

A rule is called $\forall$-safe if every $\forall$-variable appearing in its head appears in at least one positive literal of its body. An E-disjunctive program is $\forall$-safe if every rule in it is $\forall$-safe.

Proposition

Let Π be a $\forall$-safe E-disjunctive logic program, where no EQ occurs. Then for any stable model $\mathcal{A}$ of $\tau(\Pi)$, $\mathcal{A} \models \mathbf{SPP}_{c(\Pi)}$.

Corollary

Let Π be a $\forall$-safe E-disjunctive logic program. If Π is EQ-bounded, then for any stable model $\mathcal{A}$ of $\tau(\Pi)$, $\mathcal{A} \models \mathbf{SPP}_{c(\Pi)}$.

K-Colorability: SPP Satisfied

$$\exists Y \; set(X,Y) \leftarrow vertex(X)$$
$$\leftarrow edge(X,Y), set(X,C), set(Y,C)$$
$$\leftarrow set(X,Y), \text{not } color(Y)$$

But SPP may not hold for $\forall$-safe programs. For example,

$$\exists Y \; likes(a,Y) \leftarrow$$
$$\leftarrow likes(a,a).$$

A Decidable Fragment

Definition

Let Π be an E-disjunctive program and $\mathbf{p}$ the tuple of all predicates occurring in Π. Then Π is E-stratified if there is a function ℓ, called an E-level mapping of Π, that maps each predicate in $\mathbf{p}$ to a positive integer such that:

1. if r is a rule in Π, p is a predicate having positive occurrence in the body of r, and q is a predicate occurring in the head, then $\ell(p) \leq \ell(q)$;
2. in the above case, if there is an individual variable occurring in the parameters of q and bounded by an existential quantifier, then $\ell(p) < \ell(q)$.

Definition

An E-disjunctive program Π is safe if it is both $\forall$-safe and E-stratified.

Related Work and Discussion

- The chase procedure for Datalog$^{\exists}$ and Datalog$^{\pm}$.
  - "Chase" in ASP in general requires guesses
- In general, it is an interesting question whether a decidable Datalog$^{\exists}$ fragment can be extended to accommodate negation under stable models.
  - Yes, for the weakly acyclic fragment [Fagin et al 2005].
  - Not possible for sticky sets [Cali et al 2012], nor for the shy fragment [Leone et al 2012].
  - The problem is open for the guarded fragment and its variants [Cali et al 2009; Gottlob et al 2012; Alviano et al 2012].

Summary and Future Work

- E-disjunctive programs is an interesting class of ASP programs.
- The stable model semantics for these programs can be defined simply and intuitively.
- Progression characterization shows that the stable models of these programs are well-supported, and helps discover a decidable class of programs satisfying SPP.
- We identified a decidable fragment without the SPP.

Future Work:

- Semantics
- New decidable classes
- Capture and generalize some defeasible DLs
- Solvers