Multidimensional Dynamic Logic Programs

Adding Explicit Negation

Martin Baláž

balaz@ii.fmph.uniba.sk

Department of Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava

April 3, 2012

Motivation

- Extending the language of multidimensional dynamic logic programs with explicit negation
- Formulating principles on which existing semantics are based
- Understanding design decisions and their consequences

Let I be an interpretation, P be a logic program nad ℓ be a level mapping.

A rule $r \in P$ *supports* a literal L if $H(r) = L$ and $I \models B(r)$.

A rule $r \in P$ *well-supports* a literal L if r supports L and $\ell(L) > \ell(L')$ for each $L' \in B(r)$.

A rule $r \in P$ *globaly well-supports* a literal L if r well-supports L and each literal in $B^+(r)$ is globaly well-supported by a rule in P.

Multidimensional Dynamic Logic Program

$$P_1 = \left\{ \begin{array}{rcl} tv\_on & \leftarrow & \\ watch\_tv & \leftarrow & tv\_on \\ sleep & \leftarrow & \sim tv\_on \end{array} \right\}$$

$$P_2 = \left\{ \begin{array}{rcl} power\_failure & \leftarrow & \\ \sim tv\_on & \leftarrow & power\_failure \end{array} \right\}$$

$$P_3 = \left\{ \begin{array}{rcl} \sim power\_failure & \leftarrow & \end{array} \right\}$$

$$M = \{tv\_on, watch\_tv\}$$

Multidimensional Dynamic Logic Program

$$P_1 = \left\{ \begin{array}{rcl} tv\_on & \leftarrow & \\ watch\_tv & \leftarrow & tv\_on \\ sleep & \leftarrow & \sim tv\_on \end{array} \right\}$$

$$P_2 = \left\{ \begin{array}{rcl} power\_failure & \leftarrow & \\ \sim tv\_on & \leftarrow & power\_failure \end{array} \right\}$$

$$P_3 = \left\{ \begin{array}{rcl} \sim power\_failure & \leftarrow & \end{array} \right\}$$

$$M = \{tv\_on, watch\_tv\}$$

A *generalized extended logic program* is a finite set of rules of the form

$$L_0 \leftarrow L_1, \ldots, L_m$$

where $0 \leq n$ and each L_i, $0 \leq i \leq n$, is a literal (a default literal or a classical literal).

A *dynamic logic program* is a linearly ordered finite set of logic programs.

A *multidimensional dynamic logic program* is a partially ordered finite set of logic programs.

Causal Rejection Principle

$$P_1 = \left\{ \begin{array}{rcl} tv\_on & \leftarrow & \\ watch\_tv & \leftarrow & tv\_on \\ sleep & \leftarrow & \sim tv\_on \end{array} \right\}$$

$$P_2 = \left\{ \begin{array}{rcl} power\_failure & \leftarrow & \\ \sim tv\_on & \leftarrow & power\_failure \end{array} \right\}$$

$$P_3 = \left\{ \begin{array}{rcl} \sim power\_failure & \leftarrow & \end{array} \right\}$$

$$P_1 \prec P_2 \prec P_3$$

$$M = \{tv\_on, watch\_tv\}$$

Causal Rejection Principle

Let $\mathcal{P} = \{P_i \mid i \in (V, \prec)\}$ be a multidimensional dynamic logic program.

An interpretation I *satisfies the causal rejection principle in* $\mathcal{P}$ if for each rule $r \in \mathcal{P}$ not satisfied by I there exists a more preferred rule $r' \in \mathcal{P}$ such that $H(r') \bowtie H(r)$ and r' supports $H(r')$.

Principle of Inertia

Let $\mathcal{P} = \{P_i \mid i \in (V, \prec)\}$ be a multidimensional dynamic logic program.

An interpretation I *satisfies the principle of inertia in* $\mathcal{P}$ if for each rule $r \in \mathcal{P}$ holds:
If there does not exist a more preferred rule $r' \in \mathcal{P}$ such that $H(r') \bowtie H(r)$ and r' supports $H(r')$, then I satisfies r.

Reinstatement Principle

Should rejected rules reject?

$$P_1 = \{a \leftarrow\} \quad P_2 = \{\sim a \leftarrow\} \quad P_3 = \{a \leftarrow a\}$$

$$P_1 \prec P_2 \prec P_3$$

$$M_1 = \emptyset \quad M_2 = \{a\}$$

Let $\mathcal{P} = \{P_i \mid i \in (V, \prec)\}$ be a multidimensional dynamic logic program.

An interpretation I is a *backward stable model of* $\mathcal{P}$ if I is a stable model of $\bigcup_{i \in (V,\prec)} \{r \in P_i \mid I \models r\}$.

An interpretation I is a *stable model of* $\mathcal{P}$ if I is a stable model of $\bigcup_{i \in (V,\prec)} \{r \in P_i \setminus Reject(\mathcal{P}, i, I) \mid I \models r\}$ where

$$Reject(\mathcal{P}, i, I) = \{r \in P_i \mid \exists r' \in P_j : i \prec j, I \models B(r'), \\ H(r') \bowtie H(r)\}$$

Immunity to Tautological Updates

$$P_1 = \{a \leftarrow\} \quad P_2 = \{\sim a \leftarrow\} \quad P_3 = \{a \leftarrow a\}$$

$$P_1 = \{\sim a \leftarrow\} \quad P_2 = \{a \leftarrow\} \quad P_3 = \{\sim a \leftarrow \sim a\}$$

$$P_1 \prec P_2 \prec P_3$$

$$M_1 = \emptyset \quad M_2 = \{a\}$$

Immunity to Tautological Updates

Let $\mathcal{P} = \{P_i \mid i \in (V, \prec)\}$ be a multidimensional dynamic logic program.

An interpretation I is *immune to tautological updates in* $\mathcal{P}$ if for each rule $r \in \mathcal{P}$ not satisfied by I there exists a level mapping ℓ and a more preferred rule $r' \in \mathcal{P}$ such that $H(r') \bowtie H(r)$ and r' globaly well-supports $H(r')$ with respect to ℓ.

Immunity to Cyclic Updates

$$P_1 = \left\{ \begin{array}{lll} a & \leftarrow & \\ b & \leftarrow & \end{array} \right\} \quad P_2 = \left\{ \begin{array}{lll} \sim a & \leftarrow & \\ \sim b & \leftarrow & \end{array} \right\} \quad P_3 = \left\{ \begin{array}{lll} a & \leftarrow & b \\ b & \leftarrow & a \end{array} \right\}$$

$$P_1 \prec P_2 \prec P_3$$

$$M_1 = \emptyset \quad M_2 = \{a, b\}$$

Immunity to Cyclic Updates

Let $\mathcal{P} = \{P_i \mid i \in (V, \prec)\}$ be a multidimensional dynamic logic program.

An interpretation I is *immune to cyclic updates in* $\mathcal{P}$ if there exists a level mapping ℓ such that for each rule $r \in \mathcal{P}$ not satisfied by I there exists a more preferred rule $r' \in \mathcal{P}$ such that $H(r') \bowtie H(r)$ and r' globaly well-supports $H(r')$ with respect to ℓ.

Properties

Dynamic Justified Updates $\equiv$ Stable Models satisfying the causal rejection principle

Backward Dynamic Justified Updates $\equiv$ Backward Stable Models satisfying the causal rejection principle

Well-Supported Models $\equiv$ Stable Models immune to cyclic updates

Backward Well-Supported Models $\equiv$ Backward Stable Models immune to cyclic updates

Properties

Immunity to cyclic updates $\Longrightarrow$ Immunity to tautological updates $\Longrightarrow$ Causal rejection principle

Well-Supported Models $\subseteq$ Dynamic Justified Updates

Backward Well-Supported Models $\subseteq$ Backward Dynamic Justified Updates

Properties

If we restrict the class of logic programs to generalized logic programs or to extended logic programs (if we have only one type of conflict in the heads):

Stable models $\subseteq$ Backward stable models

Dynamic Justified Updates $\subseteq$ Backward Dynamic Justified Updates

Well-Supported Models $\equiv$ Backward Well-Supported Models

Properties

$$P_1 = \{\neg\, a \leftarrow\} \quad P_2 = \{a \leftarrow\} \quad P_3 = \{\sim a \leftarrow\}$$

$$P_1 \prec P_2 \prec P_3$$

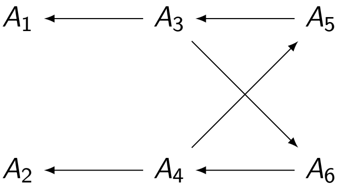
$$M_1 = \emptyset \quad M_2 = \{\neg\, a\}$$

What AF could learn from MDLP

$$P_1 = \left\{ \begin{array}{ccc} & \Rightarrow & a \\ & \Rightarrow & b \end{array} \right\} \quad P_2 = \left\{ \begin{array}{ccc} & \Rightarrow & \neg a \\ & \Rightarrow & \neg b \end{array} \right\} \quad P_3 = \left\{ \begin{array}{ccc} b & \Rightarrow & a \\ a & \Rightarrow & b \end{array} \right\}$$

$$P_1 \prec P_2 \prec P_3$$

$$\begin{array}{llllll} A_1 : & [\Rightarrow a] & A_3 : & [\Rightarrow \neg a] & A_5 : & [[\Rightarrow b] \Rightarrow a] \\ A_2 : & [\Rightarrow b] & A_4 : & [\Rightarrow \neg b] & A_6 : & [[\Rightarrow a] \Rightarrow b] \end{array}$$



What MDLP could learn from AF

$$P_1 = \left\{ \begin{array}{lll} a & \leftarrow & \\ b & \leftarrow & a \end{array} \right\} \quad P_2 = \left\{ \begin{array}{lll} \sim a & \leftarrow & \sim b \end{array} \right\}$$

$$P_1 \prec P_2$$

$$M_1 = \emptyset \quad M_2 = \{a, b\}$$

Conclusion

- We have formalized few principles for MDLP:
  - causal rejection principle
  - principle of inertia
  - reinstatement principle
  - immunity to tautological updates
  - immunity to cyclic updates
- We have characterized several existing semantics of MDLP in terms of those principles
- We have studied how relations between various semantics change if we introduce explicit negation

Thank you.