Updates of Hybrid Knowledge Bases

Martin Slota

CENTRIA & Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

REDIK Workshop
Bratislava, August 29 – 31, 2011

Part I: What's the Problem?

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Description Logics

- ✓ standard first-order semantics
- ✓ open world assumption
- ✓ expressivity vs. computational complexity
- ✓ implementations
- ✗ poor set of primitives regarding binary predicates (roles)
- ✗ no closed world reasoning
- ✗ hard to express integrity constraints

Logic Programs

- ✓ help resolve the above mentioned problems
- ✓ well-understood declarative semantics
- ✓ good computational properties

Hybrid Knowledge Bases

Integration difficulties

- ✗ semantic issues (OWA vs. CWA)
- ✗ pragmatic problems (decidability)

A number of approaches proposed

- CARIN [Levy and Rousset, 1998]
- $\mathcal{AL}$-log [Donini et al., 1998]
- $\mathcal{DL}$-*log* [Rosati, 1999]
- Description Logic Programs [Grosof et al., 2003]
- dl-programs [Eiter et al., 2004]
- HEX-programs [Eiter et al., 2005]
- Semantic Web Rules Language (SWRL) [Horrocks et al., 2004]
- $\mathcal{DL}$+*log* [Rosati, 2006]
- Description Logic Rules [Krötzsch et al., 2008]
- Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010]
- ...

Hybrid Knowledge Bases

Integration difficulties

✗ semantic issues (OWA vs. CWA)

✗ pragmatic problems (decidability)

A number of approaches proposed

- CARIN [Levy and Rousset, 1998]
- $\mathcal{AL}$-log [Donini et al., 1998]
- $\mathcal{DL}$-*log* [Rosati, 1999]
- Description Logic Programs [Grosof et al., 2003]
- dl-programs [Eiter et al., 2004]
- HEX-programs [Eiter et al., 2005]
- Semantic Web Rules Language (SWRL) [Horrocks et al., 2004]
- $\mathcal{DL}$+*log* [Rosati, 2006]
- Description Logic Rules [Krötzsch et al., 2008]
- Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010]
- ...

Hybrid Knowledge Bases

Integration difficulties

- ✗ semantic issues (OWA vs. CWA)
- ✗ pragmatic problems (decidability)

A number of approaches proposed

- CARIN [Levy and Rousset, 1998]
- $\mathcal{AL}$-log [Donini et al., 1998]
- $\mathcal{DL}$-*log* [Rosati, 1999]
- Description Logic Programs [Grosof et al., 2003]
- dl-programs [Eiter et al., 2004]
- HEX-programs [Eiter et al., 2005]
- Semantic Web Rules Language (SWRL) [Horrocks et al., 2004]
- $\mathcal{DL}$+*log* [Rosati, 2006]
- Description Logic Rules [Krötzsch et al., 2008]
- Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010]
- ...

Hybrid Knowledge Bases

Integration difficulties

- ✗ semantic issues (OWA vs. CWA)
- ✗ pragmatic problems (decidability)

A number of approaches proposed

- CARIN [Levy and Rousset, 1998]
- $\mathcal{AL}$-log [Donini et al., 1998]
- $\mathcal{DL}$-*log* [Rosati, 1999]
- Description Logic Programs [Grosof et al., 2003]
- dl-programs [Eiter et al., 2004]
- HEX-programs [Eiter et al., 2005]
- Semantic Web Rules Language (SWRL) [Horrocks et al., 2004]
- $\mathcal{DL}$+*log* [Rosati, 2006]
- Description Logic Rules [Krötzsch et al., 2008]
- Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010]
- ...

Hybrid Knowledge Bases

Integration difficulties

- ✗ semantic issues (OWA vs. CWA)
- ✗ pragmatic problems (decidability)

A number of approaches proposed

- CARIN [Levy and Rousset, 1998]
- $\mathcal{AL}$-log [Donini et al., 1998]
- $\mathcal{DL}$-*log* [Rosati, 1999]
- Description Logic Programs [Grosof et al., 2003]
- dl-programs [Eiter et al., 2004]
- HEX-programs [Eiter et al., 2005]
- Semantic Web Rules Language (SWRL) [Horrocks et al., 2004]
- $\mathcal{DL}$+*log* [Rosati, 2006]
- Description Logic Rules [Krötzsch et al., 2008]
- Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010]
- ...

Hybrid MKNF Knowledge Bases

Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010] (based on the logic of Minimal Knowledge and Negation as Failure [Lifschitz, 1991]).

- ✓ theoretically neat
- ✓ tight integration between the two distinct formalisms
- ✓ generalise most previous approaches
- ✓ known how to achieve decidability and tractability

Example (Cargo Imports Knowledge Base)

$\text{Admissible}(I) \leftarrow \sim\text{SuspectedBadGuy}(I).$

$\text{Approved}(C) \leftarrow \text{Admissible}(I), \text{RegisteredImporterOf}(I, C).$

$\text{LowRisk} \equiv \text{Approved} \sqcap (\exists\text{From.EUCountry})$

Hybrid MKNF Knowledge Bases

Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010] (based on the logic of Minimal Knowledge and Negation as Failure [Lifschitz, 1991]).

- ✓ theoretically neat
- ✓ tight integration between the two distinct formalisms
- ✓ generalise most previous approaches
- ✓ known how to achieve decidability and tractability

Example (Cargo Imports Knowledge Base)

$\text{Admissible}(I) \leftarrow \sim\text{SuspectedBadGuy}(I).$

$\text{Approved}(C) \leftarrow \text{Admissible}(I), \text{RegisteredImporterOf}(I, C).$

$\text{LowRisk} \equiv \text{Approved} \sqcap (\exists\text{From.EUCountry})$

Hybrid MKNF Knowledge Bases

Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010] (based on the logic of Minimal Knowledge and Negation as Failure [Lifschitz, 1991]).

- ✓ theoretically neat
- ✓ tight integration between the two distinct formalisms
- ✓ generalise most previous approaches
- ✓ known how to achieve decidability and tractability

Example (Cargo Imports Knowledge Base)

$\text{Admissible}(I) \leftarrow {\sim}\text{SuspectedBadGuy}(I).$

$\text{Approved}(C) \leftarrow \text{Admissible}(I), \text{RegisteredImporterOf}(I, C).$

$\text{LowRisk} \equiv \text{Approved} \sqcap (\exists \text{From.EUCountry})$

Hybrid MKNF Knowledge Bases

Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010] (based on the logic of Minimal Knowledge and Negation as Failure [Lifschitz, 1991]).

- ✓ theoretically neat
- ✓ tight integration between the two distinct formalisms
- ✓ generalise most previous approaches
- ✓ known how to achieve decidability and tractability

Example (Cargo Imports Knowledge Base)

$\text{Admissible}(I) \leftarrow \sim\text{SuspectedBadGuy}(I).$

$\text{Approved}(C) \leftarrow \text{Admissible}(I), \text{RegisteredImporterOf}(I, C).$

$\text{LowRisk} \equiv \text{Approved} \sqcap (\exists\text{From.EUCountry})$

Hybrid MKNF Knowledge Bases

Hybrid MKNF Knowledge Bases [Motik and Rosati, 2010] (based on the logic of Minimal Knowledge and Negation as Failure [Lifschitz, 1991]).

- ✓ theoretically neat
- ✓ tight integration between the two distinct formalisms
- ✓ generalise most previous approaches
- ✓ known how to achieve decidability and tractability

Example (Cargo Imports Knowledge Base)

$\text{Admissible}(I) \leftarrow \sim\text{SuspectedBadGuy}(I).$

$\text{Approved}(C) \leftarrow \text{Admissible}(I), \text{RegisteredImporterOf}(I, C).$

$\text{LowRisk} \equiv \text{Approved} \sqcap (\exists \text{From.EUCountry})$

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim\text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim \text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim\text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim\text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

c : (¬LowRiskCommodity)

$\sim$ApprovedImporterOf(i, C) $\leftarrow$ Tomato(C).

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim \text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim\text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Evolution of Hybrid Knowledge Bases

Knowledge evolves with time.

Example (Cargo Import Knowledge Base Updates)

$c : (\neg \text{LowRiskCommodity})$

$\sim\text{ApprovedImporterOf}(i, C) \leftarrow \text{Tomato}(C).$

Overall Goal

Deal with updates of Hybrid Knowledge Bases.

What is out there?

- Belief Update
- Rule Update

Belief Update

Belief Update

- reasoning about action and databases with NULL values

Belief Update

- reasoning about action and databases with NULL values
- models $=$ possible real states of represented domain; updated independently

Belief Update

- reasoning about action and databases with NULL values
- models $=$ possible real states of represented domain; updated independently

$$\boxed{\mathcal{T}}$$

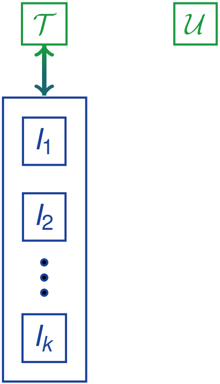
Belief Update

- reasoning about action and databases with NULL values
- models $=$ possible real states of represented domain; updated independently

$$\boxed{\mathcal{T}} \qquad \boxed{\mathcal{U}}$$

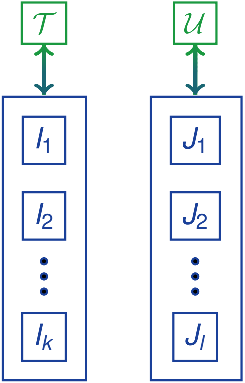
Belief Update

- reasoning about action and databases with NULL values
- models = possible real states of represented domain; updated independently



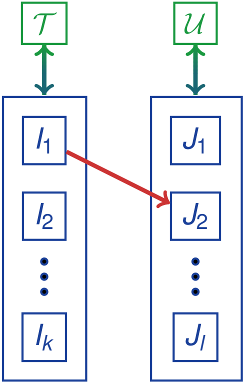
Belief Update

- reasoning about action and databases with NULL values
- models = possible real states of represented domain; updated independently



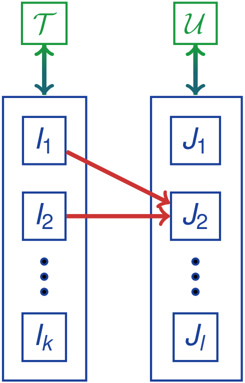
Belief Update

- reasoning about action and databases with NULL values
- models $=$ possible real states of represented domain; updated independently



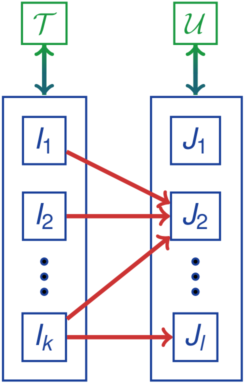
Belief Update

- reasoning about action and databases with NULL values
- models = possible real states of represented domain; updated independently



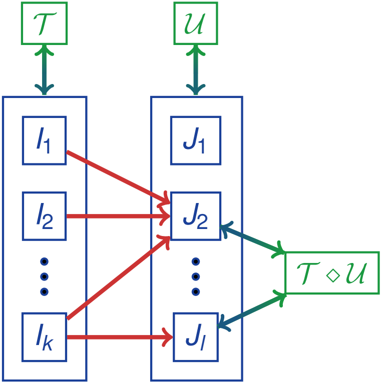
Belief Update

- reasoning about action and databases with NULL values
- models $=$ possible real states of represented domain; updated independently



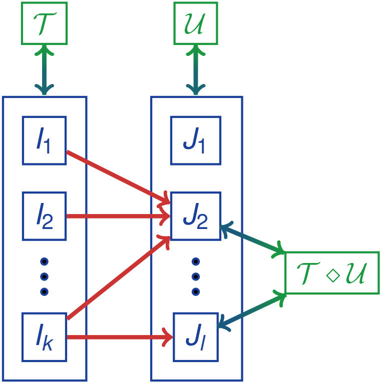
Belief Update

- reasoning about action and databases with NULL values
- models = possible real states of represented domain; updated independently



Belief Update

- reasoning about action and databases with NULL values
- models = possible real states of represented domain; updated independently



- Winslett's operator $\diamond^W$ – minimizes set difference

Belief Update

- reasoning about action and databases with NULL values
- models $=$ possible real states of represented domain; updated independently
- operator class $\Leftrightarrow$ properties $=$ representation theorem

Belief Update

- reasoning about action and databases with NULL values
- models = possible real states of represented domain; updated independently
- operator class $\Leftrightarrow$ properties = representation theorem

Theorem (Representation Theorem Template)

Let $\mathcal{C}$ be a (constructively defined) class of operators, $\mathcal{P}$ a set of properties and $\diamond$ an update operator.
Then $\diamond$ belongs to $\mathcal{C}$ if and only if $\diamond$ satisfies properties from $\mathcal{P}$.

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues

Belief Update on DL Ontologies

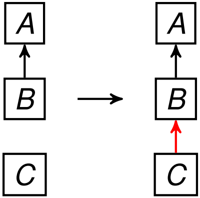
- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues
- issues with TBox updates

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues
- issues with TBox updates

Belief Update on DL Ontologies

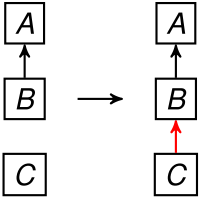
- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues
- issues with TBox updates



Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues
- issues with TBox updates

$$\{ A \sqsupseteq B_1 \} \diamond \{ B_1 \sqsupseteq C \} \models A \sqsupseteq C$$

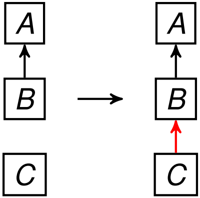


Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues
- issues with TBox updates

$$\{ A \sqsupseteq B_1 \} \diamond \{ B_1 \sqsupseteq C \} \models A \sqsupseteq C$$

$$\{ \neg A \sqsupseteq B_2 \} \diamond \{ B_2 \sqsupseteq C \} \models \neg A \sqsupseteq C$$

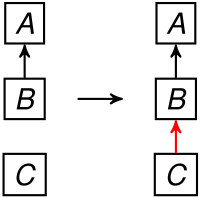


Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity
- existing work on ABox updates with a no or a static TBox addresses these issues
- issues with TBox updates

$$\{ A \sqsupseteq B_1 \} \diamond \{ B_1 \sqsupseteq C \} \models A \sqsupseteq C$$

$$\{ \neg A \sqsupseteq B_2 \} \diamond \{ B_2 \sqsupseteq C \} \models \neg A \sqsupseteq C$$



Theorem (Unsuitability of Belief Update for TBoxes)

It is impossible for an update operator to satisfy (KM 3), (KM 4), (KM 8) and the two properties above.

Belief Update on DL Ontologies

- general problems:
  - ✗ representability
  - ✗ computational complexity

Belief Update is very far from a solution to ontology updates

Theorem (Unsuitability of Belief Update for TBoxes)

It is impossible for an update operator to satisfy (KM 3), (KM 4), (KM 8) and the two properties above.

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1: \quad a. \quad b. \qquad \mathcal{P}_2: \quad a \leftarrow b. \quad b. \qquad \mathcal{P}: \quad \sim b.$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1 : \quad a. \qquad \mathcal{P}_2 : \quad a \leftarrow b. \qquad \mathcal{P} : \quad \sim b.$$
$$\qquad\quad b. \qquad\qquad\quad b.$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1 : \quad a. \quad b. \qquad \mathcal{P}_2 : \quad a \leftarrow b. \quad b. \qquad \mathcal{P} : \quad \sim b.$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1 : \quad a. \quad b. \qquad \mathcal{P}_2 : \quad a \leftarrow b. \quad b. \qquad \mathcal{P} : \quad \sim b.$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1 : \quad a. \quad b. \qquad \mathcal{P}_2 : \quad a \leftarrow b. \quad b. \qquad \mathcal{P} : \quad \sim b.$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1: \quad a. \qquad \mathcal{P}_2: \quad a \leftarrow b. \qquad \mathcal{P}: \quad \sim b.$$
$$\qquad\quad b. \qquad\qquad\qquad b.$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$
\begin{array}{llllll}
\mathcal{P}_1 : & a. & \mathcal{P}_2 : & a \leftarrow b. & \mathcal{P} : & \sim b. \\
 & b. & & b. & &
\end{array}
$$

- alternative: Dynamic Logic Programming

Rule Updates

- belief update approach is problematic
  - (KM 4): equivalent updates of equivalent programs lead to equivalent results
  - (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1: \quad a. \quad b. \qquad \mathcal{P}_2: \quad a \leftarrow b. \quad b. \qquad \mathcal{P}: \quad \sim b.$$

- alternative: Dynamic Logic Programming

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP-hard in general
- polynomial for ABox updates in certain Description Logics

DLP

- most classical postulates turned out unsuitable for them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

Belief Update vs. DLP

Belief Update

- based on rationality postulates
- syntax-insensitive
- inertia applied to atoms
- NP
- po ... Logics

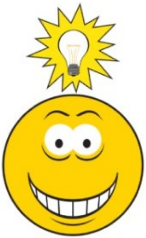
Combining them is difficult!

DLP

- mo ... them
- syntax-sensitive
- inertia applied to rules
- NP-hardness inherited from stable model semantics
- well-founded version stays polynomial

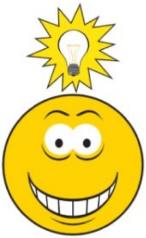
Part II: Addressing the Problem

1. Static Rules and Dynamic ABox



- rules: static queries, policies, business rules, strategies, preferences…
- TBox: static relations between roles and concepts
- a sequence of ABoxes: dynamically changing assertions about individuals in concepts and roles
- we proposed a solution for this scenario and investigated its basic properties

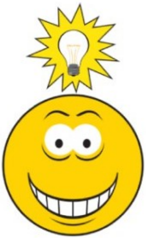
1. Static Rules and Dynamic ABox



- rules: static queries, policies, business rules, strategies, preferences...
- TBox: static relations between roles and concepts
- a sequence of ABoxes: dynamically changing assertions about individuals in concepts and roles
- we proposed a solution for this scenario and investigated its basic properties

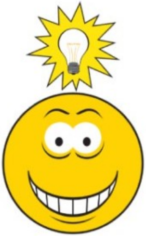
1. Static Rules and Dynamic ABox

- rules: static queries, policies, business rules, strategies, preferences...
- TBox: static relations between roles and concepts



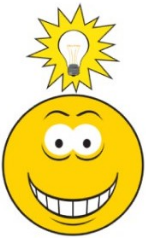
- a sequence of ABoxes: dynamically changing assertions about individuals in concepts and roles
- we proposed a solution for this scenario and investigated its basic properties

1. Static Rules and Dynamic ABox



- rules: static queries, policies, business rules, strategies, preferences...
- TBox: static relations between roles and concepts
- a sequence of ABoxes: dynamically changing assertions about individuals in concepts and roles
- we proposed a solution for this scenario and investigated its basic properties

1. Static Rules and Dynamic ABox



- rules: static queries, policies, business rules, strategies, preferences...
- TBox: static relations between roles and concepts
- a sequence of ABoxes: dynamically changing assertions about individuals in concepts and roles
- we proposed a solution for this scenario and investigated its basic properties

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) =$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{\, H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \,\}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \quad T_{\mathcal{P}}(M)$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{\, H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \,\}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \quad (T_{\mathcal{P}}(M) \cup \mathcal{T})$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \quad (T_{\mathcal{P}}(M) \cup \mathcal{T}) \diamond^{W}$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{\, H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \,\}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \qquad (T_{\mathcal{P}}(M) \cup \mathcal{T}) \diamond^{W} (\mathcal{A}_1 \cup \mathcal{T})$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{\, H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \,\}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \quad (T_{\mathcal{P}}(M) \cup \mathcal{T}) \diamond^{W} (\mathcal{A}_1 \cup \mathcal{T}) \diamond^{W} \cdots \diamond^{W} (\mathcal{A}_n \cup \mathcal{T})$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{ H(r) \mid r \in \mathcal{P} \land M \models B(r) \}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \mathrm{mod}((T_{\mathcal{P}}(M) \cup \mathcal{T}) \diamond^{W} (\mathcal{A}_1 \cup \mathcal{T}) \diamond^{W} \cdots \diamond^{W} (\mathcal{A}_n \cup \mathcal{T}))$$

1. Static Rules and Dynamic ABox (Technicalities)

- we defined semantics for a normal logic program $\mathcal{P}$ updated by an ABox sequence $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ in the presence of a TBox $\mathcal{T}$
- M is a stable model of a normal logic program $\mathcal{P}$ iff M is the least fixed point of $T_{\mathcal{P}^M}$ where $\mathcal{P}^M$ is the Gelfond-Lifschitz reduct of $\mathcal{P}$ w.r.t. M and

$$T_{\mathcal{P}}(M) = \{\, H(r) \mid r \in \mathcal{P} \wedge M \models B(r) \,\}$$

Definition (Minimal Change Dynamic Stable Model)

M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ iff M is the least fixed point of $T_{\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}}$ where

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \mathrm{mod}((T_{\mathcal{P}}(M) \cup \mathcal{T}) \diamond^W (\mathcal{A}_1 \cup \mathcal{T}) \diamond^W \cdots \diamond^W (\mathcal{A}_n \cup \mathcal{T}))$$

1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

Proposition

Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}$ a TBox, $\mathcal{A}$ an ABox and M a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$. Then $M \models \mathcal{A}$.

1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

Proposition

Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}, \mathcal{T}'$ be TBoxes such that $\text{mod}(\mathcal{T}) = \text{mod}(\mathcal{T}')$, *$\mathcal{A}, \mathcal{A}'$ be ABoxes such that* $\text{mod}(\mathcal{A}) = \text{mod}(\mathcal{A}')$ *and M be an MKNF interpretation. Then M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}'} \mathcal{A}'$.*

1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

Proposition

Let $\mathcal{P}$ be a finite ground program. Then M is a stable model of $\mathcal{P}$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\emptyset} \emptyset$.

1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

Proposition

Let $\mathcal{P}$ be a finite ground program containing only facts, $\mathcal{T}$ a TBox, $\mathcal{A}$ a sequence of ABoxes and M an MKNF interpretation. Then M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if and only if M is a minimal change update model of $\mathcal{S}_{\mathcal{P}} \oplus^{\mathcal{T}} \mathcal{A}$ where $\mathcal{S}_{\mathcal{P}} = \{ p \mid \mathbf{K}\, p \in \mathcal{P} \}$.

1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

Proposition

Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}$ be a TBox and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ a sequence of ABoxes (where $n \geq 1$). Let $\mathcal{A}' = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_i, \emptyset, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$ for some $i \in \{0, 1, 2, \ldots, n\}$. Then an MKNF interpretation M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}'$.

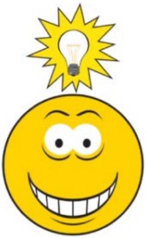
1. Static Rules and Dynamic ABox (Properties)

- ✓ Primacy of new information
- ✓ Syntax-independence w.r.t. to $\mathcal{T}$ and $\mathcal{A}$
- ✓ Generalisation of stable model semantics
- ✓ Generalisation of Winslett's update semantics
- ✓ Immunity to empty updates

Proposition

Let $\mathcal{T}$ be a TBox, $\mathcal{A}$ an ABox and M an MKNF interpretation. Then M is a minimal change dynamic stable model of $\emptyset \oplus^{\mathcal{T}} \mathcal{A}$ if and only if $M = \mathrm{mod}(\mathcal{T} \cup \mathcal{A})$.

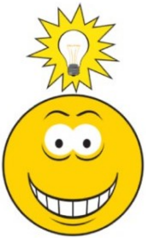
2. Modular Update Semantics



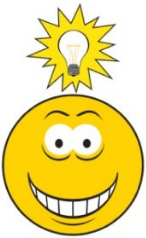
- it is frequently possible to identify distinct ontology layers and rule layers in a hybrid knowledge base
- define a modular update semantics for these cases
- based on a splitting theorem for hybrid MKNF knowledge bases

2. Modular Update Semantics

- it is frequently possible to identify distinct ontology layers and rule layers in a hybrid knowledge base
- define a modular update semantics for these cases
- based on a splitting theorem for hybrid MKNF knowledge bases

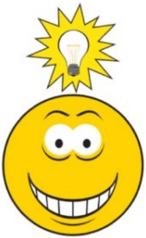


2. Modular Update Semantics



- it is frequently possible to identify distinct ontology layers and rule layers in a hybrid knowledge base
- define a modular update semantics for these cases
- based on a splitting theorem for hybrid MKNF knowledge bases

2. Modular Update Semantics



- it is frequently possible to identify distinct ontology layers and rule layers in a hybrid knowledge base
- define a modular update semantics for these cases
- based on a splitting theorem for hybrid MKNF knowledge bases

2. Modular Update (Splitting Theorem)

2. Modular Update (Splitting Theorem)

$\mathcal{K}$

$$D \sqsubseteq \exists R.E$$

$$q(Z) \leftarrow B(Z), \sim D(Z).$$

$$A \equiv B \sqcap C$$

$$p(Z) \leftarrow \sim A(Z).$$

2. Modular Update (Splitting Theorem)

$\mathcal{K}$

$D \sqsubseteq \exists R.E$

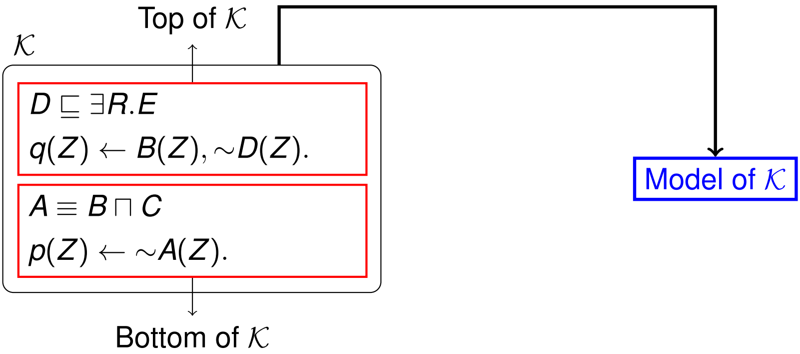
$q(Z) \leftarrow B(Z), \sim D(Z).$

$A \equiv B \sqcap C$

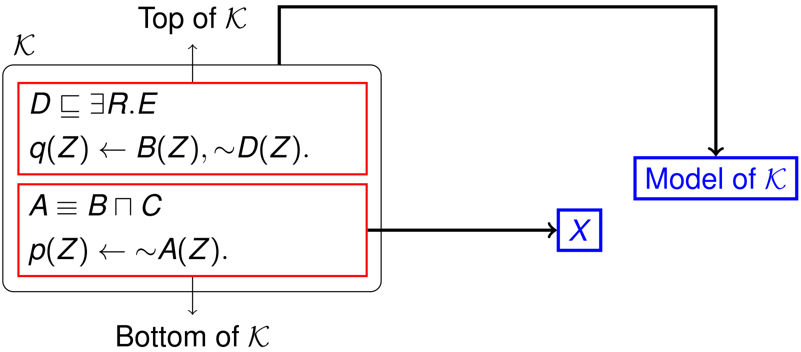
$p(Z) \leftarrow \sim A(Z).$

Model of $\mathcal{K}$

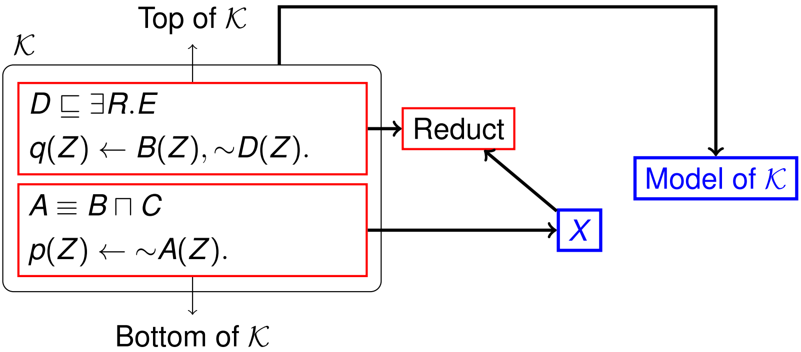
2. Modular Update (Splitting Theorem)



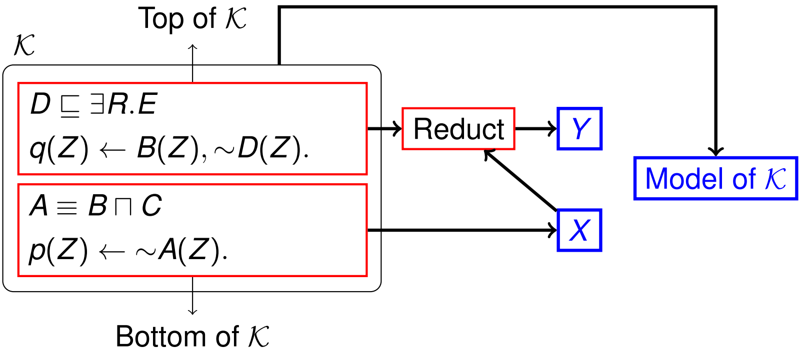
2. Modular Update (Splitting Theorem)



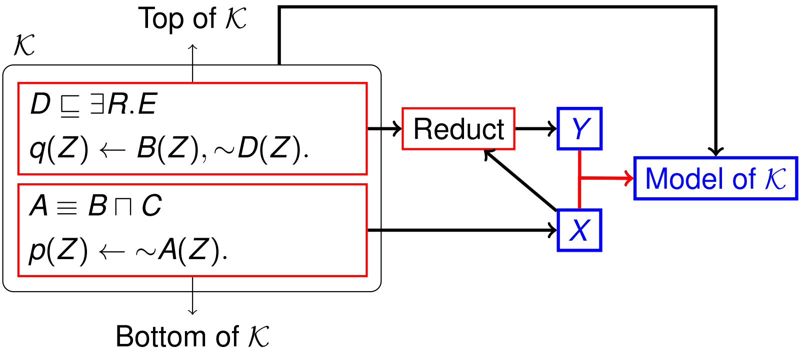
2. Modular Update (Splitting Theorem)



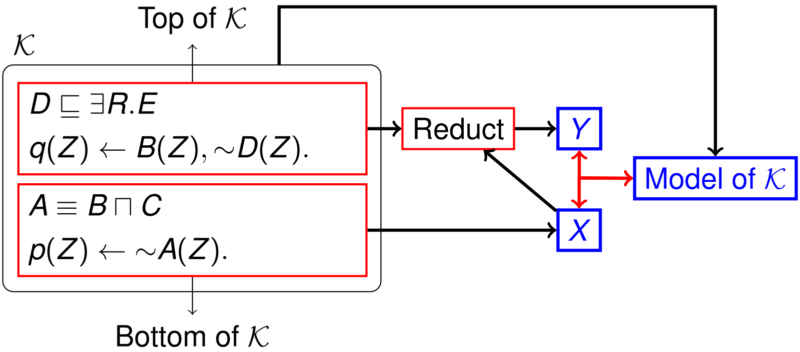
2. Modular Update (Splitting Theorem)



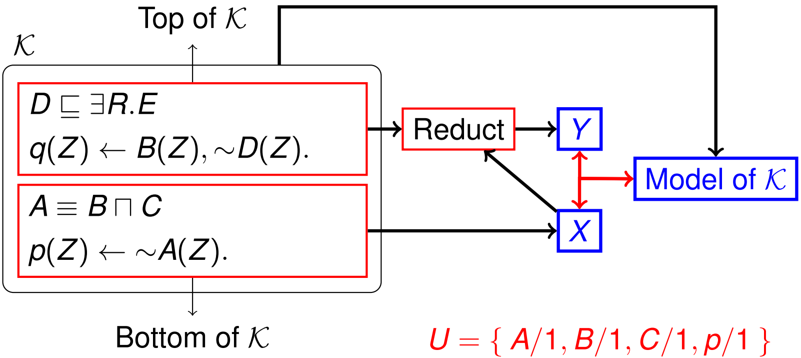
2. Modular Update (Splitting Theorem)



2. Modular Update (Splitting Theorem)

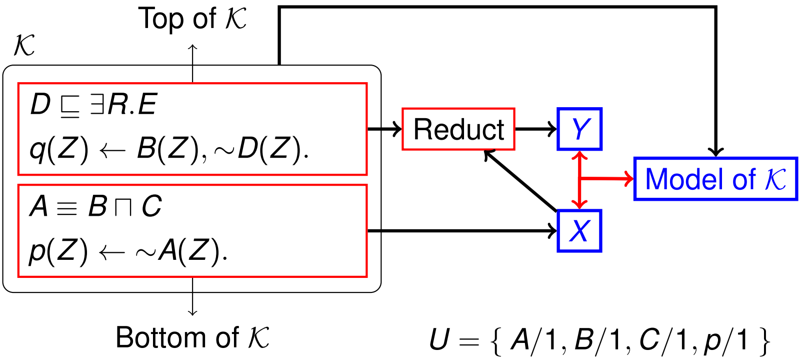


2. Modular Update (Splitting Theorem)



$U = \{ A/1, B/1, C/1, p/1 \}$

2. Modular Update (Splitting Theorem)



$$U = \{\, A/1, B/1, C/1, p/1 \,\}$$

Theorem (Splitting Theorem)

Let U be a splitting set for a hybrid knowledge base $\mathcal{K}$. Then M is an MKNF model of $\mathcal{K}$ if and only if $M = X \cap Y$ for some solution $\langle X, Y \rangle$ to $\mathcal{K}$ w.r.t. U.

2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.

2. Modular Update (Splitting Sequence Theorem)

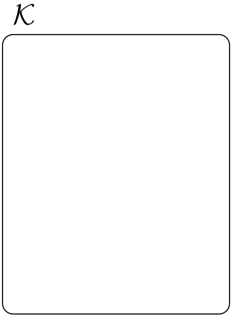
Definition (Splitting Sequence)

A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.

2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

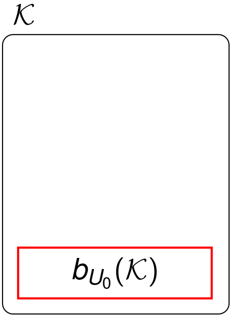
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha < \mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha < \mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

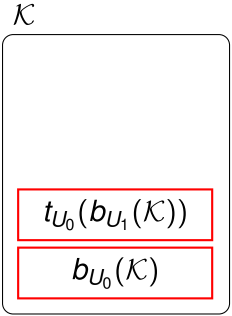
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

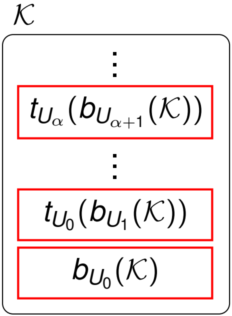
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

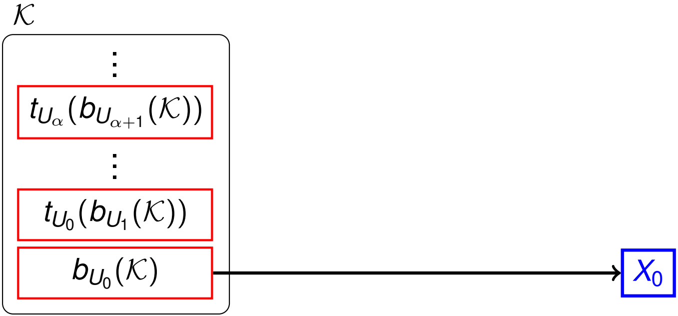
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

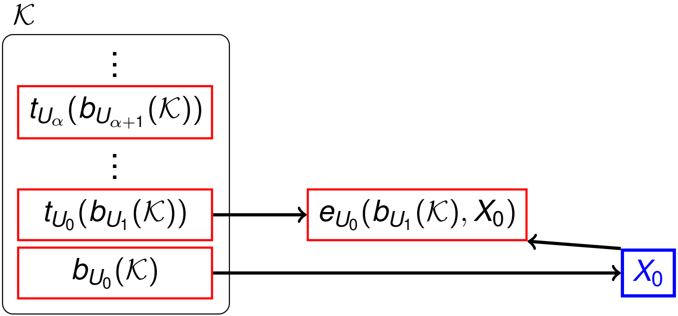
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

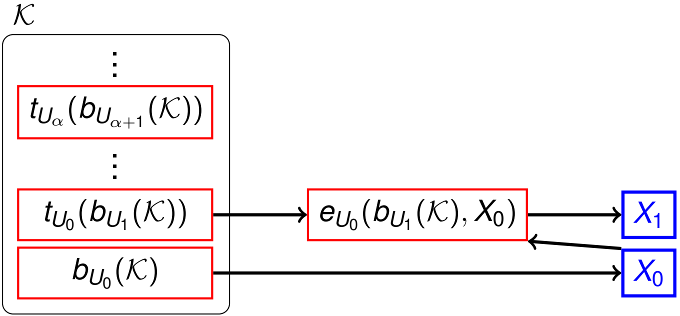
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

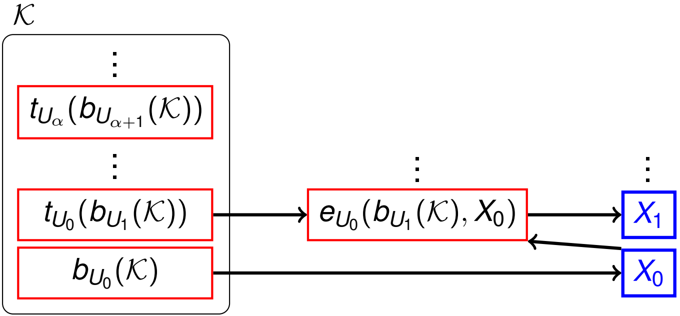
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

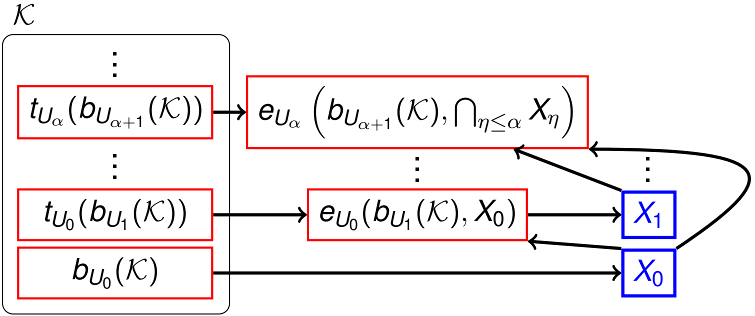
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

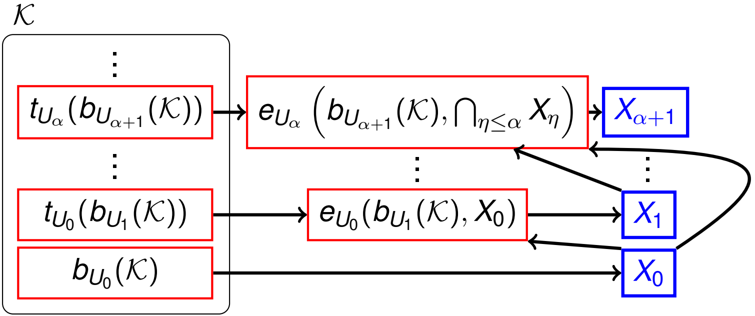
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

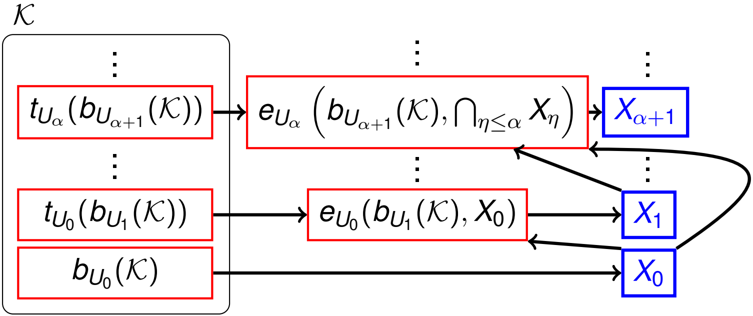
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

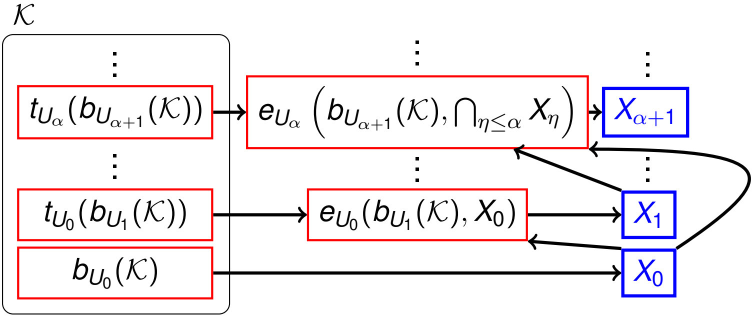
A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.



The sequence $\langle X_\alpha \rangle_{\alpha<\mu}$ is a solution to $\mathcal{K}$ w.r.t. U.

2. Modular Update (Splitting Sequence Theorem)

Definition (Splitting Sequence)

A splitting sequence for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.

Theorem (Splitting Sequence Theorem)

Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a hybrid knowledge base $\mathcal{K}$. Then M is an MKNF model of $\mathcal{K}$ if and only if $M = \bigcap_{\alpha<\mu} X_\alpha$ for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. U.

2. Modular Update Semantics (Operator)

Question

Given

- an update semantics for $\langle \mathcal{O}_i \rangle_{i<n}$ and
- an update semantics for $\langle \mathcal{P}_i \rangle_{i<n}$,

to what type of $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ can we easily assign an update semantics?

2. Modular Update Semantics (Operator)

Question

Given

- an update semantics for $\langle \mathcal{O}_i \rangle_{i<n}$ and
- an update semantics for $\langle \mathcal{P}_i \rangle_{i<n}$,

to what type of $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ can we easily assign an update semantics?

2. Modular Update Semantics (Operator)

Question

Given

- an update semantics for $\langle \mathcal{O}_i \rangle_{i<n}$ and
- an update semantics for $\langle \mathcal{P}_i \rangle_{i<n}$,

to what type of $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ can we easily assign an update semantics?

- $U = \langle U_\alpha \rangle_{\alpha<\mu}$ is called update-enabling if for all X and all α,

$$\left\langle e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}_i), X) \right\rangle_{i<n} \tag{1}$$

  contains either only rules or only ontology axioms

- a sequence $X = \langle X_\alpha \rangle_{\alpha<\mu}$ of dynamic models of (1) is a solution to $\mathcal{K}$ w.r.t. U.

2. Modular Update Semantics (Operator)

Question

Given

- an update semantics for $\langle \mathcal{O}_i \rangle_{i<n}$ and
- an update semantics for $\langle \mathcal{P}_i \rangle_{i<n}$,

to what type of $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ can we easily assign an update semantics?

- $U = \langle U_\alpha \rangle_{\alpha<\mu}$ is called update-enabling if for all X and all α,

$$\left\langle e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}_i), X) \right\rangle_{i<n} \qquad (1)$$

  contains either only rules or only ontology axioms
- a sequence $X = \langle X_\alpha \rangle_{\alpha<\mu}$ of dynamic models of (1) is a solution to $\mathcal{K}$ w.r.t. U.

2. Modular Update Semantics (Operator)

Question

Given

- an update semantics for $\langle \mathcal{O}_i \rangle_{i<n}$ and
- an update semantics for $\langle \mathcal{P}_i \rangle_{i<n}$,

to what type of $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ can we easily assign an update semantics?

Definition

Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be an update-enabling splitting sequence for a dynamic hybrid knowledge base $\mathcal{K}$. We say that an MKNF interpretation M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. U if $M = \bigcap_{\alpha<\mu} X_\alpha$ for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. U.

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- Independence of Splitting Sequence
- Generalisation of MKNF
- Generalisation of Winslett's Update Semantics
- Generalisation of Dynamic Stable Models
- Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

Independence of Splitting Sequence

Generalisation of MKNF

Generalisation of Winslett's Update Semantics

Generalisation of Dynamic Stable Models

Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- ✓ Independence of Splitting Sequence
- Generalisation of MKNF
- Generalisation of Winslett's Update Semantics
- Generalisation of Dynamic Stable Models
- Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

✓ Independence of Splitting Sequence

✓ Generalisation of MKNF

✓ Generalisation of Winslett's Update Semantics

✓ Generalisation of Dynamic Stable Models

✓ Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

✓ Independence of Splitting Sequence

✓ Generalisation of MKNF

✓ Generalisation of Winslett's Update Semantics

✓ Generalisation of Dynamic Stable Models

✓ Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- ✓ Independence of Splitting Sequence
- ✓ Generalisation of MKNF
- ✓ Generalisation of Winslett's Update Semantics
- ✓ Generalisation of Dynamic Stable Models
- ✓ Primacy of New Information

Theorem (Independence of Splitting Sequence)

Let U, V be update-enabling splitting sequences for a dynamic hybrid knowledge base $\mathcal{K}$. Then M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. U if and only if M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. V.

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- ✓ Independence of Splitting Sequence
- ✓ Generalisation of MKNF
- ✓ Generalisation of Winslett's Update Semantics
- ✓ Generalisation of Dynamic Stable Models
- ✓ Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- ✓ Independence of Splitting Sequence
- ✓ Generalisation of MKNF
- ✓ Generalisation of Winslett's Update Semantics
- ✓ Generalisation of Dynamic Stable Models
- ✓ Primacy of New Information

2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- ✓ Independence of Splitting Sequence
- ✓ Generalisation of MKNF
- ✓ Generalisation of Winslett's Update Semantics
- ✓ Generalisation of Dynamic Stable Models
- ✓ Primacy of New Information

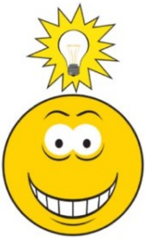
2. Modular Update Semantics (Properties)

Using particular update operators:

1. Winslett's minimal change update operator [Winslett, 1990]
2. Refined dynamic stable models [Alferes et al., 2005]

- ✓ Independence of Splitting Sequence
- ✓ Generalisation of MKNF
- ✓ Generalisation of Winslett's Update Semantics
- ✓ Generalisation of Dynamic Stable Models
- ✓ Primacy of New Information

3. Semantic Rule Updates



- (KM 4) + stable or strong equivalence = catastrophe

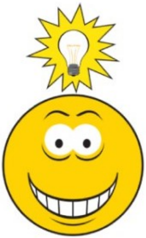
$\mathcal{P}_1: \quad a. \quad b. \qquad \mathcal{P}_2: \quad a \leftarrow b. \quad b. \qquad \mathcal{P}: \quad \sim b.$

- SE-Models: monotonic semantics more expressive than stable models
- view a program as the set of sets of SE-models of its rules
- perform updates on the set of sets of SE-models

3. Semantic Rule Updates

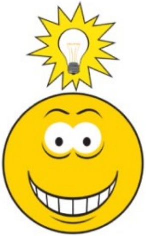
- (KM 4) + stable or strong equivalence
  = catastrophe

$\mathcal{P}_1 : \quad a. \quad b. \qquad \mathcal{P}_2 : \quad a \leftarrow b. \quad b. \qquad \mathcal{P} : \quad \sim b.$



- SE-Models: monotonic semantics more expressive than stable models
- view a program as the set of sets of SE-models of its rules
- perform updates on the set of sets of SE-models

3. Semantic Rule Updates



- (KM 4) + stable or strong equivalence
  = catastrophe

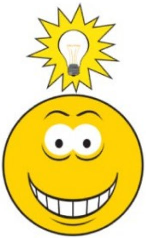
$\mathcal{P}_1 : \quad a. \quad b.$ $\qquad \mathcal{P}_2 : \quad a \leftarrow b. \quad b.$ $\qquad \mathcal{P} : \quad \sim b.$

- SE-Models: monotonic semantics more expressive than stable models
- view a program as the set of sets of SE-models of its rules
- perform updates on the set of sets of SE-models

3. Semantic Rule Updates

- (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1: \quad a. \quad b. \qquad \mathcal{P}_2: \quad a \leftarrow b. \quad b. \qquad \mathcal{P}: \quad \sim b.$$

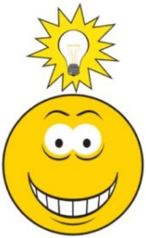


- SE-Models: monotonic semantics more expressive than stable models
- view a program as the set of sets of SE-models of its rules
- perform updates on the set of sets of SE-models

3. Semantic Rule Updates

- (KM 4) + stable or strong equivalence = catastrophe

$$\mathcal{P}_1: \quad a. \quad b. \qquad \mathcal{P}_2: \quad a \leftarrow b. \quad b. \qquad \mathcal{P}: \quad \sim b.$$



- SE-Models: monotonic semantics more expressive than stable models
- view a program as the set of sets of SE-models of its rules
- perform updates on the set of sets of SE-models

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Conclusion

Addressing Hybrid Updates

1. by restricting the dynamic part of the knowledge base
2. by restricting interaction between DL axioms and rules
3. by aiming to create semantic counterparts of the rule update semantics

Open Questions

- DL ontology update
  - principles in line with intuitions
  - operators
  - representability
- rule update
  - logical foundation more expressive than SE-models
  - semantic characterisation of existing operators (or similar)

Thank you!

Thank you!

References I

José Júlio Alferes, Federico Banti, Antonio Brogi, and João Alexandre Leite. The refined extension principle for semantics of dynamic logic programming. Studia Logica, 79 (1):7–32, 2005. URL http://centria.di.fct.unl.pt/~jja/page3/assets/sl05.pdf.

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. AL-log: Integrating datalog and description logics. Journal of Intelligent Information Systems, 10(3): 227–252, 1998.

Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning

References II

(KR2004), pages 141–151, Whistler, Canada, June 2-5 2004. AAAI Press. URL http://www.kr.tuwien.ac.at/staff/eiter/et-archive/kr04-dl_asp.pdf.

Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05), pages 90–96, Edinburgh, Scotland, UK, July 30-August 5 2005. Professional Book Center. ISBN 0938075934.

Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In Proceedings of the 12th International World Wide Web Conference (WWW 2003), pages 48–57, Budapest, Hungary, May 20-24 2003. ACM. ISBN 1-58113-680-3. URL `http://www.cs.man.ac.uk/~horrocks/Publications/download/2003/p117-grosof.pdf`.

Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A semantic web rule language. W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/, 2004. URL `Latestversionavailableathttp://www.w3.org/Submission/SWRL/`.

References IV

Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Description logic rules. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikos Avouris, editors, Proceedings of the 18th European Conference on Artificial Intelligence (ECAI2008), pages 80–84, Patras, Greece, July 2008. IOS Press. URL http://www.aifb.uni-karlsruhe.de/WBS/phi/resources/publications/dlrules-ecai08.pdf.

Alon Y. Levy and Marie-Christine Rousset. Combining horn rules and description logics in CARIN. Artificial Intelligence, 104(1-2):165–209, 1998. URL http://www.cs.washington.edu/homes/alon/files/aij-carin.pdf.

References V

Vladimir Lifschitz. Nonmonotonic databases and epistemic queries. In Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91), pages 381–386, 1991. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.5424&rep=rep1&type=pdf.

Boris Motik and Riccardo Rosati. Reconciling description logics and rules. Journal of the ACM, 57(5):93–154, 2010.

Riccardo Rosati. Towards expressive kr systems integrating datalog and description logics: preliminary report. In Patrick Lambrix, Alexander Borgida, Maurizio Lenzerini, Ralf Möller, and Peter F. Patel-Schneider, editors, Proceedings of the 1999 International Workshop on Description Logics (DL'99), volume 22 of CEUR Workshop Proceedings, Linköping, Sweden, July 30 - August 1 1999. CEUR-WS.org.

Riccardo Rosati. DL+log: Tight integration of description logics and disjunctive datalog. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR2006), pages 68–78, Lake District of the United Kingdom, June 2-5 2006. AAAI Press. ISBN 978-1-57735-271-6. URL http://www.dis.uniroma1.it/~rosati/publications/Rosati-KR-06.pdf.

Martin Slota and João Leite. Towards Closed World Reasoning in Dynamic Open Worlds. Theory and Practice of Logic Programming, 26th Int'l. Conference on Logic Programming (ICLP'10) Special Issue, 10(4-6):547–564, July 2010. URL http://slotik.info/sites/slotik.info/files/Slota2010a.pdf.

References VII

Marianne Winslett. Updating Logical Databases. Cambridge University Press, New York, NY, USA, 1990. ISBN 0-521-37371-9.