Optimizing the Search for Repairs from Active Integrity Constraints

Luís Cruz-Filipe

Escola Superior Náutica Infante D. Henrique / CMAF / LabMAg

LabMAg Seminar
September 18th, 2013

Outline

1 Active integrity constraints

Outline

1 Active integrity constraints

2 Parallelization

Outline

1. Active integrity constraints
2. Parallelization
3. Stratification

Outline

1. Active integrity constraints
2. Parallelization
3. Stratification
4. Conclusions

Outline

1 Active integrity constraints

2 Parallelization

3 Stratification

4 Conclusions

Active integrity constraints

Motivation

Specify a constraint and propose possible solutions.

Active integrity constraints

Motivation

Specify a constraint and propose possible solutions.

Works both ways:

Active integrity constraints

Motivation

Specify a constraint and propose possible solutions.

Works both ways:

- may express preferences

Active integrity constraints

Motivation

Specify a constraint and propose possible solutions.

Works both ways:

- may express preferences
- may eliminate options

Example: family relations

Integrity constraint

$$\forall x \forall y.((\mathsf{siblingOf}(x,y) \wedge \neg \mathsf{siblingOf}(y,x)) \supset \bot)$$

Example: family relations

Integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x, y) \wedge \neg \mathrm{siblingOf}(y, x)) \supset \bot)$$

Active integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x, y) \wedge \neg \mathrm{siblingOf}(y, x)) \supset +\mathrm{siblingOf}(y, x))$$

Example: family relations

Integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x,y) \wedge \neg \mathrm{siblingOf}(y,x)) \supset \bot)$$

Active integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x,y) \wedge \neg \mathrm{siblingOf}(y,x)) \supset +\mathrm{siblingOf}(y,x))$$

Active integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x,y) \wedge \neg \mathrm{siblingOf}(y,x)) \supset -\mathrm{siblingOf}(x,y))$$

Example: family relations

Integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x,y) \wedge \neg \mathrm{siblingOf}(y,x)) \supset \bot)$$

Active integrity constraint

$$\forall x \forall y.((\mathrm{siblingOf}(x,y) \wedge \neg \mathrm{siblingOf}(y,x)) \supset + \mathrm{siblingOf}(y,x) \mid -\mathrm{siblingOf}(x,y))$$

Active integrity constraints

Definition (Flesca2004)

An *Active integrity constraint* is a formula of the form

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

where $\{\alpha_1^D, \ldots, \alpha_k^D\} \subseteq \{L_1, \ldots, L_m\}$.

Active integrity constraints

Definition (Flesca2004)

An *Active integrity constraint* is a formula of the form

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

where $\{\alpha_1^D, \ldots, \alpha_k^D\} \subseteq \{L_1, \ldots, L_m\}$.

A valid AIC

$$\text{siblingOf}(x, y) \wedge \neg\text{siblingOf}(y, x) \supset +\text{siblingOf}(y, x)$$

Active integrity constraints

Definition (Flesca2004)

An *Active integrity constraint* is a formula of the form

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

where $\{\alpha_1^D, \ldots, \alpha_k^D\} \subseteq \{L_1, \ldots, L_m\}$.

A valid AIC

$$\text{siblingOf}(x, y) \wedge \neg\text{siblingOf}(y, x) \supset +\text{siblingOf}(y, x)$$

An invalid AIC

$$\text{siblingOf}(x, y) \wedge \neg\text{siblingOf}(y, x) \supset -\text{siblingOf}(x, y) \mid +\text{Parent}(x)$$

Intuitive semantics of AICs

A generic AIC

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

Intuitive semantics of AICs

A generic AIC

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

- conjunction on the left (“body”)

Intuitive semantics of AICs

A generic AIC

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

- conjunction on the left ("body")
- disjunction on the right ("head")

Intuitive semantics of AICs

A generic AIC

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

- conjunction on the left ("body")
- disjunction on the right ("head")
- semantics of (normal) implication

Intuitive semantics of AICs

A generic AIC

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

- conjunction on the left ("body")
- disjunction on the right ("head")
- semantics of (normal) implication
- holds iff one of the L_is fails (but...)

Intuitive semantics of AICs

A generic AIC

$$L_1, \ldots, L_m \supset \alpha_1 \mid \ldots \mid \alpha_k$$

- conjunction on the left ("body")
- disjunction on the right ("head")
- semantics of (normal) implication
- holds iff one of the L_is fails (but...)
- $\{\alpha_1^D, \ldots, \alpha_k^D\}$ are *updatable* literals

Repairs

Definition (Caroprese et al., 2006)

Let $\mathcal{I}$ be a database and η be a set of (A)ICs. A *weak repair* for *I* and η is a consistent set $\mathcal{U}$ of update actions such that:

Repairs

Definition (Caroprese et al., 2006)

Let $\mathcal{I}$ be a database and η be a set of (A)ICs. A *weak repair* for *I* and η is a consistent set $\mathcal{U}$ of update actions such that:

- $\mathcal{U}$ consists of essential actions only

Repairs

Definition (Caroprese et al., 2006)

Let $\mathcal{I}$ be a database and η be a set of (A)ICs. A *weak repair* for *I* and η is a consistent set $\mathcal{U}$ of update actions such that:

- $\mathcal{U}$ consists of essential actions only
- $\mathcal{I} \circ \mathcal{U} \models \eta$

Repairs

Definition (Caroprese et al., 2006)

Let $\mathcal{I}$ be a database and η be a set of (A)ICs. A *weak repair* for *I* and η is a consistent set $\mathcal{U}$ of update actions such that:

- $\mathcal{U}$ consists of essential actions only
- $\mathcal{I} \circ \mathcal{U} \models \eta$

(Beware of the notation.)

Repairs

Definition (Caroprese et al., 2006)

Let $\mathcal{I}$ be a database and η be a set of (A)ICs. A *weak repair* for *I* and η is a consistent set $\mathcal{U}$ of update actions such that:

- $\mathcal{U}$ consists of essential actions only
- $\mathcal{I} \circ \mathcal{U} \models \eta$

(Beware of the notation.)

Definition

A *repair* is a weak repair that is minimal w.r.t. inclusion.

Family relations, yet again

Inconsistency

$$\mathrm{siblingOf}(\mathrm{John}, \mathrm{Mary})$$

$$\mathrm{siblingOf}(x, y) \wedge \neg\mathrm{siblingOf}(y, x) \supset +\mathrm{siblingOf}(y, x)$$

Family relations, yet again

Inconsistency

$$\text{siblingOf}(\text{John}, \text{Mary})$$

$$\text{siblingOf}(x, y) \wedge \neg\text{siblingOf}(y, x) \supset +\text{siblingOf}(y, x)$$

A repair

$$+\text{siblingOf}(\text{Mary}, \text{John})$$

Family relations, yet again

Inconsistency

$$\mathsf{siblingOf}(\mathsf{John}, \mathsf{Mary})$$

$$\mathsf{siblingOf}(x, y) \land \neg\mathsf{siblingOf}(y, x) \supset +\mathsf{siblingOf}(y, x)$$

Another repair

$$-\mathsf{siblingOf}(\mathsf{John}, \mathsf{Mary})$$

Family relations, yet again

Inconsistency

$$\text{siblingOf}(\text{John}, \text{Mary})$$

$$\text{siblingOf}(x, y) \wedge \neg\text{siblingOf}(y, x) \supset +\text{siblingOf}(y, x)$$

A weak repair

$$+\text{siblingOf}(\text{Mary}, \text{John}), +\text{Parent}(\text{John})$$

Family relations, yet again

Inconsistency

$$\text{siblingOf}(\text{John}, \text{Mary})$$

$$\text{siblingOf}(x, y) \wedge \neg\text{siblingOf}(y, x) \supset +\text{siblingOf}(y, x)$$

Not a weak repair

$$+\text{siblingOf}(\text{Mary}, \text{John}), -\text{siblingOf}(\text{John}, \text{Mary})$$

Finding repairs

"Algorithm"

1. Choose a set $\mathcal{U}$ of update actions (based on $\mathcal{I}$)
2. Compute $\mathcal{I} \circ \mathcal{U}$
3. Check if all AICs in η hold

Finding repairs

"Algorithm"

1. Choose a set $\mathcal{U}$ of update actions (based on $\mathcal{I}$)
2. Compute $\mathcal{I} \circ \mathcal{U}$
3. Check if all AICs in η hold

Each step can be done in polynomial time on $\mathcal{I}$ and η.

Finding repairs

"Algorithm"

1. Choose a set $\mathcal{U}$ of update actions (based on $\mathcal{I}$)
2. Compute $\mathcal{I} \circ \mathcal{U}$
3. Check if all AICs in η hold

Each step can be done in polynomial time on $\mathcal{I}$ and η.

Finding weak repairs is NP-complete.

Declarative semantics

The notion of repair ignores the head of the AIC.

Declarative semantics

The notion of repair ignores the head of the AIC.

Caroprese et al., 2006/2011

- *founded* repairs take into account the actions in the head

Declarative semantics

The notion of repair ignores the head of the AIC.

Caroprese et al., 2006/2011

- *founded* repairs take into account the actions in the head
- *justified* repairs avoid justification circles

Founded repairs (I)

Intuitively: if $\mathcal{U}$ is founded, then removing an action from $\mathcal{U}$ causes some AIC with that action in the head to be violated.

Founded repairs (I)

Intuitively: if $\mathcal{U}$ is founded, then removing an action from $\mathcal{U}$ causes some AIC with that action in the head to be violated.

Definition

A set of update actions $\mathcal{U}$ is founded w.r.t. $\mathcal{I}$ and η if, for every $\alpha \in \mathcal{U}$, there is a rule $r \in \eta$ such that $\alpha \in \text{head}(r)$ and

$$\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r\,.$$

Founded repairs (I)

Intuitively: if $\mathcal{U}$ is founded, then removing an action from $\mathcal{U}$ causes some AIC with that action in the head to be violated.

Definition

A set of update actions $\mathcal{U}$ is founded w.r.t. $\mathcal{I}$ and η if, for every $\alpha \in \mathcal{U}$, there is a rule $r \in \eta$ such that $\alpha \in \text{head}(r)$ and

$$\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r\,.$$

(This is the original definition, which was abandoned and rediscovered.)

Founded repairs (I)

Intuitively: if $\mathcal{U}$ is founded, then removing an action from $\mathcal{U}$ causes some AIC with that action in the head to be violated.

Definition

A set of update actions $\mathcal{U}$ is founded w.r.t. $\mathcal{I}$ and η if, for every $\alpha \in \mathcal{U}$, there is a rule $r \in \eta$ such that $\alpha \in \mathrm{head}(r)$ and

$$\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r\,.$$

(This is the original definition, which was abandoned and rediscovered.)

Definition

A (weak) repair that is founded is called a founded (weak) repair.

Founded repairs (II)

Example

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, \text{not } b \supset -a \qquad r_3 : a, \text{not } c \supset +c$$
$$r_2 : \text{not } a, b \supset -b \qquad r_4 : b, \text{not } c \supset +c$$

Founded repairs (II)

Example

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, \text{not } b \supset -a \qquad r_3 : a, \text{not } c \supset +c$$
$$r_2 : \text{not } a, b \supset -b \qquad r_4 : b, \text{not } c \supset +c$$

Then $\{+c\}$ is a founded repair,

Founded repairs (II)

Example

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, \text{not } b \supset -a \qquad r_3 : a, \text{not } c \supset +c$$
$$r_2 : \text{not } a, b \supset -b \qquad r_4 : b, \text{not } c \supset +c$$

Then $\{+c\}$ is a founded repair, but so is $\{-a, -b\}$.

Founded repairs (II)

Example

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, \text{not } b \supset -a \qquad r_3 : a, \text{not } c \supset +c$$
$$r_2 : \text{not } a, b \supset -b \qquad r_4 : b, \text{not } c \supset +c$$

Then $\{+c\}$ is a founded repair, but so is $\{-a, -b\}$.

In $\{-a, -b\}$ we have a *circularity of support*.

Founded repairs (II)

Example

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, \text{not } b \supset -a \qquad r_3 : a, \text{not } c \supset +c$$
$$r_2 : \text{not } a, b \supset -b \qquad r_4 : b, \text{not } c \supset +c$$

Then $\{+c\}$ is a founded repair, but so is $\{-a, -b\}$.

In $\{-a, -b\}$ we have a *circularity of support*.

It is a founded repair that is not justified.

Justified repairs

The definition is meant to avoid circularity of support.

Justified repairs

The definition is meant to avoid circularity of support.

It is not intuitive and there is no motivation in the references.

Justified repairs

The definition is meant to avoid circularity of support.

It is not intuitive and there is no motivation in the references.

Too restrictive?

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, b \supset -a \qquad r_2 : a, \text{not } b \supset -a \qquad r_3 : \text{not } a, b \supset -b$$

Justified repairs

The definition is meant to avoid circularity of support.

It is not intuitive and there is no motivation in the references.

Too restrictive?

Take $\mathcal{I} = \{a, b\}$ and

$$r_1 : a, b \supset -a \qquad r_2 : a, \text{not } b \supset -a \qquad r_3 : \text{not } a, b \supset -b$$

Again $\{-a, -b\}$ is a founded repair that is not justified.

Complexity

Deciding whether there is a founded weak repair for $\mathcal{I}$ and η is NP-complete.

Complexity

Deciding whether there is a founded weak repair for $\mathcal{I}$ and η is NP-complete.

Deciding whether there is a founded repair for $\mathcal{I}$ and η is Σ_P^2-complete.

Complexity

Deciding whether there is a founded weak repair for $\mathcal{I}$ and η is NP-complete.

Deciding whether there is a founded repair for $\mathcal{I}$ and η is Σ_P^2-complete.

Deciding whether there is a justified (weak) repair for $\mathcal{I}$ and η is Σ_P^2-complete.

Outline

1 Active integrity constraints

2 Parallelization

3 Stratification

4 Conclusions

Motivation

Reduce the size of the problem by splitting the set of AICs into smaller sets.

Motivation

Reduce the size of the problem by splitting the set of AICs into smaller sets.

Goals:

Motivation

Reduce the size of the problem by splitting the set of AICs into smaller sets.

Goals:

- do not lose repairs;

Motivation

Reduce the size of the problem by splitting the set of AICs into smaller sets.

Goals:

- do not lose repairs;
- efficient combination of results.

Independence (I)

Definition

Two AICs r_1 and r_2 are *independent*, $r_1 \perp\!\!\!\perp r_2$, if the literals in their bodies do not share atoms.

Independence (I)

Definition

Two AICs r_1 and r_2 are *independent*, $r_1 \perp\!\!\!\perp r_2$, if the literals in their bodies do not share atoms.
Two sets of AICs η_1 and η_2 are *independent*, $\eta_1 \perp\!\!\!\perp \eta_2$, if $r_1 \perp\!\!\!\perp r_2$ for every $r_1 \in \eta_1$ and $r_2 \in \eta_2$.

Independence (I)

Definition

Two AICs r_1 and r_2 are *independent*, $r_1 \perp\!\!\!\perp r_2$, if the literals in their bodies do not share atoms.
Two sets of AICs η_1 and η_2 are *independent*, $\eta_1 \perp\!\!\!\perp \eta_2$, if $r_1 \perp\!\!\!\perp r_2$ for every $r_1 \in \eta_1$ and $r_2 \in \eta_2$.

- No attention to the heads of the rules.

Independence (I)

Definition

Two AICs r_1 and r_2 are *independent*, $r_1 \perp\!\!\!\perp r_2$, if the literals in their bodies do not share atoms.
Two sets of AICs η_1 and η_2 are *independent*, $\eta_1 \perp\!\!\!\perp \eta_2$, if $r_1 \perp\!\!\!\perp r_2$ for every $r_1 \in \eta_1$ and $r_2 \in \eta_2$.

- No attention to the heads of the rules.
- Not affected by the database.

Independence (II)

Example

Consider the following AICs:

$$r_1 : a, \text{not } b \supset -a \qquad r_2 : b, c \supset -c \qquad r_3 : d \supset -d$$

Independence (II)

Example

Consider the following AICs:

$$r_1 : a, \text{not } b \supset -a \qquad r_2 : b, c \supset -c \qquad r_3 : d \supset -d$$

Then $\{r_1, r_2\} \perp\!\!\!\perp \{r_3\}$.

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

- *each $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i;*

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

- *each $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i;*
- *if $\mathcal{U}$ is a repair, then so is each $\mathcal{U}_i$;*

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

- *each $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i;*
- *if $\mathcal{U}$ is a repair, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is founded, then so is each $\mathcal{U}_i$;*

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

- *each $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i;*
- *if $\mathcal{U}$ is a repair, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is founded, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is justified, then so is each $\mathcal{U}_i$.*

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

- *each $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i;*
- *if $\mathcal{U}$ is a repair, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is founded, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is justified, then so is each $\mathcal{U}_i$.*

Furthermore, if every action in $\mathcal{U}$ affects a literal in the body of a rule in η, then $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$.

Independence vs. parallelization (I)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and η.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ affecting literals in the bodies of rules in η_i, for $i = 1, 2$.
Then:

- *each $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i;*
- *if $\mathcal{U}$ is a repair, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is founded, then so is each $\mathcal{U}_i$;*
- *if $\mathcal{U}$ is justified, then so is each $\mathcal{U}_i$.*

Furthermore, if every action in $\mathcal{U}$ affects a literal in the body of a rule in η, then $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$. This hypothesis is (very) reasonable in practice.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

For foundedness, take $\alpha \in \mathcal{U}_1$.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

For foundedness, take $\alpha \in \mathcal{U}_1$. Since $\mathcal{U}$ is founded, there is a rule $r \in \eta$ such that $\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r$.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

For foundedness, take $\alpha \in \mathcal{U}_1$. Since $\mathcal{U}$ is founded, there is a rule $r \in \eta$ such that $\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r$. Necessarily $r \in \eta_1$

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

For foundedness, take $\alpha \in \mathcal{U}_1$. Since $\mathcal{U}$ is founded, there is a rule $r \in \eta$ such that $\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r$. Necessarily $r \in \eta_1$ and $\mathcal{I} \circ (\mathcal{U}_1 \setminus \{\alpha\}) \not\models r$.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

For foundedness, take $\alpha \in \mathcal{U}_1$. Since $\mathcal{U}$ is founded, there is a rule $r \in \eta$ such that $\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r$. Necessarily $r \in \eta_1$ and $\mathcal{I} \circ (\mathcal{U}_1 \setminus \{\alpha\}) \not\models r$. Therefore $\mathcal{U}_1$ is founded.

Independence vs. parallelization (II)

Proof (sketch).

Given $r_1 \in \eta_1$ and $r_2 \in \eta_2$, changing the logical values of literals in the body of r_1 cannot affect the semantics of r_2 and vice-versa. This implies that $\mathcal{U}_i$ is a weak repair for $\mathcal{I}$ and η_i.

This also implies that, if $\mathcal{U}$ is minimal, then so must $\mathcal{U}_1$ and $\mathcal{U}_2$ be.

For foundedness, take $\alpha \in \mathcal{U}_1$. Since $\mathcal{U}$ is founded, there is a rule $r \in \eta$ such that $\mathcal{I} \circ (\mathcal{U} \setminus \{\alpha\}) \not\models r$. Necessarily $r \in \eta_1$ and $\mathcal{I} \circ (\mathcal{U}_1 \setminus \{\alpha\}) \not\models r$. Therefore $\mathcal{U}_1$ is founded.

This means that we can parallelize the search for repairs without losing solutions.

Independence vs. parallelization (III)

Example

Let $\mathcal{I} = \{a, b, c, d\}$ and consider the following set of AICs η:

$$r_1 : a, \text{not } b \supset -a \qquad r_2 : b, c \supset -c \qquad r_3 : d \supset -d$$

Independence vs. parallelization (III)

Example

Let $\mathcal{I} = \{a, b, c, d\}$ and consider the following set of AICs η:

$$r_1 : a, \text{not } b \supset -a \qquad r_2 : b, c \supset -c \qquad r_3 : d \supset -d$$

Then $\mathcal{U} = \{-c, -d\}$ is a founded repair for $\mathcal{I}$ and η.

Independence vs. parallelization (III)

Example

Let $\mathcal{I} = \{a, b, c, d\}$ and consider the following set of AICs η:

$$r_1 : a, \text{not } b \supset -a \qquad r_2 : b, c \supset -c \qquad r_3 : d \supset -d$$

Then $\mathcal{U} = \{-c, -d\}$ is a founded repair for $\mathcal{I}$ and η.

Since $\eta_1 = \{r_1, r_2\}$ and $\eta_2 = \{r_3\}$ are independent, the theorem guarantees that $\mathcal{U}_1 = \{-c\}$ and $\mathcal{U}_2 = \{-d\}$ are founded repairs for $\mathcal{I}$ and η_1 or η_2, respectively.

Independence vs. parallelization (IV)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ *with* $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ *be a database; and* $\mathcal{U}_i$ *be weak repairs for* $\mathcal{I}$ *and* η_i, *for* $i = 1, 2$, *such that all actions in* $\mathcal{U}_i$ *affect a literal in the body of a rule in* η_i.

Independence vs. parallelization (IV)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ *with* $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ *be a database; and* $\mathcal{U}_i$ *be weak repairs for* $\mathcal{I}$ *and* η_i*, for* $i = 1, 2$*, such that all actions in* $\mathcal{U}_i$ *affect a literal in the body of a rule in* η_i*.*
Define $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$*.*

Independence vs. parallelization (IV)

Theorem

Let $\eta = \eta_1 \cup \eta_2$ with $\eta_1 \perp\!\!\!\perp \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}_i$ be weak repairs for $\mathcal{I}$ and η_i, for $i = 1, 2$, such that all actions in $\mathcal{U}_i$ affect a literal in the body of a rule in η_i.

Define $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$.

Then:

- *$\mathcal{U}$ is a weak repair for $\mathcal{I}$ and η;*
- *if each $\mathcal{U}_i$ is a repair, then so is $\mathcal{U}$;*
- *if each $\mathcal{U}_i$ is founded, then so is $\mathcal{U}$;*
- *if each $\mathcal{U}_i$ is justified, then so is $\mathcal{U}$.*

Independence vs. parallelization (V)

The proof is similar.

Independence vs. parallelization (V)

The proof is similar.

This means that parallelization of the search does not add “new” (false) solutions.

Computing independent sets of AICs

The previous results generalize to several independent sets of AICs.

Computing independent sets of AICs

The previous results generalize to several independent sets of AICs.

In order to split a set η into independent sets, consider the relation $\not\perp\!\!\!\perp$ of dependence.

Computing independent sets of AICs

The previous results generalize to several independent sets of AICs.

In order to split a set η into independent sets, consider the relation $\not\perp\!\!\!\perp$ of dependence. This relation is reflexive and symmetric.

Computing independent sets of AICs

The previous results generalize to several independent sets of AICs.

In order to split a set η into independent sets, consider the relation $\not\perp\!\!\!\perp$ of dependence. This relation is reflexive and symmetric. Therefore its transitive closure $\not\perp\!\!\!\perp^+$ is an equivalence relation.

Computing independent sets of AICs

The previous results generalize to several independent sets of AICs.

In order to split a set η into independent sets, consider the relation $\not\perp\!\!\!\perp$ of dependence. This relation is reflexive and symmetric. Therefore its transitive closure $\not\perp\!\!\!\perp^+$ is an equivalence relation. The quotient set ${}^{\eta}/_{\not\perp\!\!\!\perp^+}$ is the finest partition of η in independent sets, and can be computed efficiently.

Outline

1. Active integrity constraints
2. Parallelization
3. Stratification
4. Conclusions

Precedence (I)

Definition

AIC r_1 *precedes* AIC r_2, $r_1 \prec r_2$, if some action in the head of r_1 affects a literal in the body of r_2.

Precedence (I)

Definition

AIC r_1 *precedes* AIC r_2, $r_1 \prec r_2$, if some action in the head of r_1 affects a literal in the body of r_2.

- Reflexive relation.

Precedence (I)

Definition

AIC r_1 *precedes* AIC r_2, $r_1 \prec r_2$, if some action in the head of r_1 affects a literal in the body of r_2.

- Reflexive relation.
- $\langle {}^{\eta}/_{\approx}, \preceq \rangle$ is a partial order, where $\preceq$ is the transitive closure of $\prec$ and $\approx$ is the induced equivalence relation.

Precedence (I)

Definition

AIC r_1 *precedes* AIC r_2, $r_1 \prec r_2$, if some action in the head of r_1 affects a literal in the body of r_2.

- Reflexive relation.
- $\langle {}^{\eta}/_{\approx}, \preceq \rangle$ is a partial order, where $\preceq$ is the transitive closure of $\prec$ and $\approx$ is the induced equivalence relation.

(Similar to stratified negation in logic programming. . .)

Precedence (II)

Example

Consider the following set of AICs η.

$$r_1 : a, b \supset -a \qquad r_3 : \text{not } a, c, d \supset -c \mid -d$$

$$r_2 : a, \text{not } b, c \supset +b \qquad r_4 : b, d, e \supset -e$$

Precedence (II)

Example

Consider the following set of AICs η.

$$r_1 : a, b \supset -a \qquad r_3 : \text{not } a, c, d \supset -c \mid -d$$

$$r_2 : a, \text{not } b, c \supset +b \qquad r_4 : b, d, e \supset -e$$

The precedence relation is summarized in the following diagram.

Precedence (II)

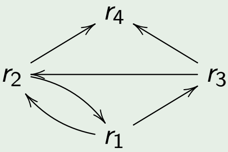
Example

Consider the following set of AICs η.

$$r_1 : a, b \supset -a \qquad r_3 : \text{not } a, c, d \supset -c \mid -d$$

$$r_2 : a, \text{not } b, c \supset +b \qquad r_4 : b, d, e \supset -e$$

The precedence relation is summarized in the following diagram.



Precedence (II)

Example

Consider the following set of AICs η.

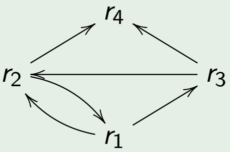
$r_1 : a, b \supset -a$

$r_2 : a, \text{not } b, c \supset +b$

$r_3 : \text{not } a, c, d \supset -c \mid -d$

$r_4 : b, d, e \supset -e$

The precedence relation is summarized in the following diagram.



The equivalence classes are $\eta_1 = \{r_1, r_2, r_3\}$ and $\eta_2 = \{r_4\}$.

Precedence vs. stratification (I)

Theorem

Let $\eta_1, \eta_2 \in {}^{\eta}/_{\approx}$ with $\eta_1 \prec \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.

Precedence vs. stratification (I)

Theorem

Let $\eta_1, \eta_2 \in {}^{\eta}/_{\approx}$ with $\eta_1 \prec \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.
Assume that every action in $\mathcal{U}$ occurs in the head of a rule in $\eta_1 \cup \eta_2$.

Precedence vs. stratification (I)

Theorem

Let $\eta_1, \eta_2 \in {}^{\eta}/_{\approx}$ with $\eta_1 \prec \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.
Assume that every action in $\mathcal{U}$ occurs in the head of a rule in $\eta_1 \cup \eta_2$.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ in the head of a rule in η_i, for $i = 1, 2$.

Precedence vs. stratification (I)

Theorem

Let $\eta_1, \eta_2 \in {}^{\eta}/_{\approx}$ with $\eta_1 \prec \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.
Assume that every action in $\mathcal{U}$ occurs in the head of a rule in $\eta_1 \cup \eta_2$.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ in the head of a rule in η_i, for $i = 1, 2$.
Then:

Precedence vs. stratification (I)

Theorem

Let $\eta_1, \eta_2 \in {}^{\eta}/_{\approx}$ with $\eta_1 \prec \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.
Assume that every action in $\mathcal{U}$ occurs in the head of a rule in $\eta_1 \cup \eta_2$.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ in the head of a rule in η_i, for $i = 1, 2$.
Then:

- *$\mathcal{U}_1$ is a weak repair for $\mathcal{I}$ and η_1 and $\mathcal{U}_2$ is a weak repair for $\mathcal{I} \circ \mathcal{U}_1$ and η_2;*

Precedence vs. stratification (I)

Theorem

Let $\eta_1, \eta_2 \in {}^{\eta}/_{\approx}$ with $\eta_1 \prec \eta_2$; $\mathcal{I}$ be a database; and $\mathcal{U}$ be a weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.
Assume that every action in $\mathcal{U}$ occurs in the head of a rule in $\eta_1 \cup \eta_2$.
Define $\mathcal{U}_i$ as the set of actions in $\mathcal{U}$ in the head of a rule in η_i, for $i = 1, 2$.
Then:

- *$\mathcal{U}_1$ is a weak repair for $\mathcal{I}$ and η_1 and $\mathcal{U}_2$ is a weak repair for $\mathcal{I} \circ \mathcal{U}_1$ and η_2;*
- *if $\mathcal{U}$ is founded/justified, then so is each $\mathcal{U}_i$.*

Precedence vs. stratification (II)

The proof is similar to the above.

Precedence vs. stratification (II)

The proof is similar to the above.

This allows us to sequentialize the search for repairs.

Precedence vs. stratification (II)

The proof is similar to the above.

This allows us to sequentialize the search for repairs.

However, some results do not hold:

Precedence vs. stratification (II)

The proof is similar to the above.

This allows us to sequentialize the search for repairs.

However, some results do not hold:

- it may happen that $\mathcal{U}$ is a repair, but $\mathcal{U}_1$ and/or $\mathcal{U}_2$ are not;

Precedence vs. stratification (II)

The proof is similar to the above.

This allows us to sequentialize the search for repairs.

However, some results do not hold:

- it may happen that $\mathcal{U}$ is a repair, but $\mathcal{U}_1$ and/or $\mathcal{U}_2$ are not;
- there may be a weak (founded, justified) repair $\mathcal{U}_1$ for $\mathcal{I}$ and η_1 that is not a subset of any weak repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$.

Precedence vs. stratification (III)

Example

Let $\mathcal{I} = \emptyset$ and consider the following active integrity constraints.

$$r_1 : \text{not } a \supset +a \quad r_2 : \text{not } b, c \supset +b \quad r_3 : b, \text{not } c \supset +c$$

$$r_4 : a, \text{not } b, \text{not } c, d \supset -d \quad r_5 : a, \text{not } b, \text{not } c, \text{not } d \supset +d$$

Precedence vs. stratification (III)

Example

Let $\mathcal{I} = \emptyset$ and consider the following active integrity constraints.

$$r_1 : \text{not } a \supset +a \quad r_2 : \text{not } b, c \supset +b \quad r_3 : b, \text{not } c \supset +c$$
$$r_4 : a, \text{not } b, \text{not } c, d \supset -d \quad r_5 : a, \text{not } b, \text{not } c, \text{not } d \supset +d$$

Taking $\eta_1 = \{r_1, r_2, r_3\}$ and $\eta_2 = \{r_4, r_5\}$, one has $\eta_1 \prec \eta_2$.

Precedence vs. stratification (III)

Example

Let $\mathcal{I} = \emptyset$ and consider the following active integrity constraints.

$$r_1 : \text{not } a \supset +a \quad r_2 : \text{not } b, c \supset +b \quad r_3 : b, \text{not } c \supset +c$$
$$r_4 : a, \text{not } b, \text{not } c, d \supset -d \quad r_5 : a, \text{not } b, \text{not } c, \text{not } d \supset +d$$

Taking $\eta_1 = \{r_1, r_2, r_3\}$ and $\eta_2 = \{r_4, r_5\}$, one has $\eta_1 \prec \eta_2$. Furthermore, $\{+a\}$ and $\{+a, +b, +c\}$ are weak repairs for $\mathcal{I}$ and η_1, the first of which is a repair.

Precedence vs. stratification (III)

Example

Let $\mathcal{I} = \emptyset$ and consider the following active integrity constraints.

$$r_1 : \text{not } a \supset +a \quad r_2 : \text{not } b, c \supset +b \quad r_3 : b, \text{not } c \supset +c$$
$$r_4 : a, \text{not } b, \text{not } c, d \supset -d \quad r_5 : a, \text{not } b, \text{not } c, \text{not } d \supset +d$$

Taking $\eta_1 = \{r_1, r_2, r_3\}$ and $\eta_2 = \{r_4, r_5\}$, one has $\eta_1 \prec \eta_2$. Furthermore, $\{+a\}$ and $\{+a, +b, +c\}$ are weak repairs for $\mathcal{I}$ and η_1, the first of which is a repair. However, the only repair for $\mathcal{I}$ and $\eta_1 \cup \eta_2$ is $\{+a, +b, +c\}$.

Precedence vs. stratification (IV)

Theorem

Let η_1, η_2 and $\mathcal{I}$ be as before; $\mathcal{U}_1$ be a weak repair for $\mathcal{I}$ and η_1; $\mathcal{U}_2$ be a weak repair for $\mathcal{I} \circ \mathcal{U}_1$ and η_2; such that every action in $\mathcal{U}_i$ occurs in the head of a rule in η_i.

Precedence vs. stratification (IV)

Theorem

Let η_1, η_2 and $\mathcal{I}$ be as before; $\mathcal{U}_1$ be a weak repair for $\mathcal{I}$ and η_1; $\mathcal{U}_2$ be a weak repair for $\mathcal{I} \circ \mathcal{U}_1$ and η_2; such that every action in $\mathcal{U}_i$ occurs in the head of a rule in η_i.
Define $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$.

Precedence vs. stratification (IV)

Theorem

Let η_1, η_2 and $\mathcal{I}$ be as before; $\mathcal{U}_1$ be a weak repair for $\mathcal{I}$ and η_1; $\mathcal{U}_2$ be a weak repair for $\mathcal{I} \circ \mathcal{U}_1$ and η_2; such that every action in $\mathcal{U}_i$ occurs in the head of a rule in η_i.
Define $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$.
Then:

- *$\mathcal{U}$ is a weak repair for $\mathcal{I}$ and η;*
- *if each $\mathcal{U}_i$ is a repair, then so is $\mathcal{U}$;*
- *if each $\mathcal{U}_i$ is founded/justified, then so is $\mathcal{U}$.*

Precedence vs. stratification (IV)

Theorem

Let η_1, η_2 and $\mathcal{I}$ be as before; $\mathcal{U}_1$ be a weak repair for $\mathcal{I}$ and η_1; $\mathcal{U}_2$ be a weak repair for $\mathcal{I} \circ \mathcal{U}_1$ and η_2; such that every action in $\mathcal{U}_i$ occurs in the head of a rule in η_i.
Define $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$.
Then:

- *$\mathcal{U}$ is a weak repair for $\mathcal{I}$ and η;*
- *if each $\mathcal{U}_i$ is a repair, then so is $\mathcal{U}$;*
- *if each $\mathcal{U}_i$ is founded/justified, then so is $\mathcal{U}$.*

The proof is similar.

Outline

1 Active integrity constraints

2 Parallelization

3 Stratification

4 Conclusions

What we achieved. . .

What we achieved. . .

- Split a large problem in several smaller ones

What we achieved. . .

- Split a large problem in several smaller ones
- Possibility of parallelization

What we achieved. . .

- Split a large problem in several smaller ones
- Possibility of parallelization
- Stratification relation

. . . and what we still hope to do

. . . and what we still hope to do

- (More) practical evaluation

. . . and what we still hope to do

- (More) practical evaluation
- Prototype implementation

. . . and what we still hope to do

- (More) practical evaluation
- Prototype implementation
- Generalizations of AICs outside the database world

Thank you.