Database repair and dynamic logic

Andreas Herzig (CNRS, IRIT)
joint work with Guillaume Feuillade and Christos Rantsoudis

First Workshop on Challenges and Adequacy Conditions for Logics in the New Age of Artificial Intelligence, November 3-6, 2022

Outline

1. Motivation: repair via more informative integrity constraints
2. Active Integrity Constraints: postulates
3. Active Integrity Constraints: revisiting the existing semantics
4. ECA constraints
5. An analysis in dynamic logic
6. From AICs to active TBoxes: a sketch
7. Conclusion

Violations of integrity constraints

- ideally: $\mathcal{D} \models \mathbf{IC}$
- but $\mathcal{D} \not\models \mathbf{IC}$ often happens

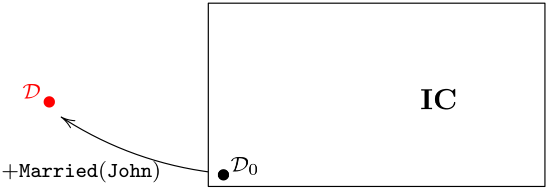
$$\mathbf{IC} = (\forall x)[\mathtt{Bachelor}(x) \wedge \mathtt{Married}(x) \rightarrow \bot]$$

$$\mathcal{D}_0 = \{\mathtt{Bachelor}(\mathtt{John})\}$$

$$\Downarrow \ +\mathtt{Married}(\mathtt{John})$$

$$\mathcal{D} = \{\mathtt{Bachelor}(\mathtt{John}), \mathtt{Married}(\mathtt{John})\}$$

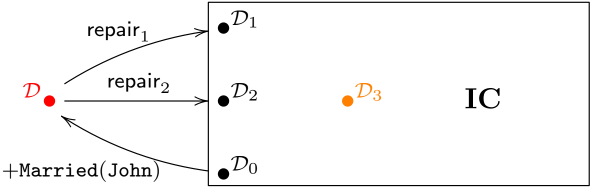
Repairs



Two solutions

1. make $\mathcal{D}$ consistent with $\mathbf{IC}$
   - repair ('data cleaning')
2. live with inconsistent $\mathcal{D}$
   - consistent answer to a query = holds in all possible repairs
   - hypothetical repair

Repairing violations

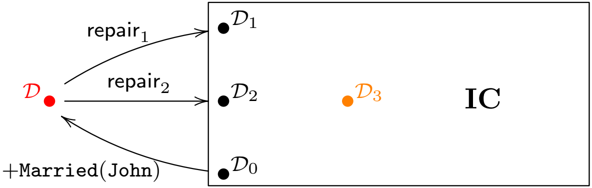


possible repairs of $\mathcal{D} = \{\texttt{Bachelor}(\texttt{John}), \texttt{Married}(\texttt{John})\}$:

$$\mathcal{D}_1 = \{\texttt{Married}(\texttt{John})\}$$
$$\mathcal{D}_2 = \{\texttt{Bachelor}(\texttt{John})\}$$
$$\mathcal{D}_3 = \{\texttt{Bachelor}(\texttt{John}), \texttt{Bachelor}(\texttt{Charles})\}$$
$$\dots$$

- all do the job: $\mathcal{D}_i \models \mathbf{IC}$
  - ... but there are too many
  - ... and some are not intended

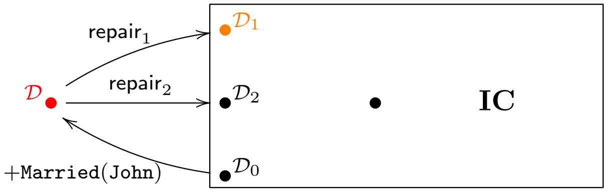
Repairs: minimal change



Minimal repair = a $\mathcal{D}'$ *closest* to $\mathcal{D}$ such that $\mathcal{D}' \models \textbf{IC}$

- intuition: $\mathcal{D}_3$ further away than $\mathcal{D}_1$ and $\mathcal{D}_2$
- definition of closeness?
  - symmetric difference
    - cf. Possible Models Approach PMA [Winslett, AAAI 1988]
    - PMA repairs $\Longrightarrow$ produce PMA updates
  - Hamming distance [Lopatenko&Bertossi, DEXA 2006]

Repairs: minimal change is not enough



- PMA repairs of $\mathcal{D} = \{\texttt{Bachelor}(\texttt{John}), \texttt{Married}(\texttt{John})\}$:

$$\mathcal{D}_1 = \{\texttt{Bachelor}(\texttt{John})\}$$
$$\mathcal{D}_2 = \{\texttt{Married}(\texttt{John})\}$$

  - both closest to $\mathcal{D}$ (for PMA and Hamming distance)
  - but $\mathcal{D}_1$ is unintended
- can we do better by making **IC** more informative?

Active integrity constraints [Flesca et al., PPDP 2004]

"if *condition* holds then do *action*"

- active IC = negation of static IC + update actions

  Static : $(\forall x)[\texttt{Bachelor}(x) \land \texttt{Married}(x) \rightarrow \bot]$

  Active : $(\forall x)[\texttt{Bachelor}(x) \land \texttt{Married}(x), \{-\texttt{Bachelor}(x)\}]$

- intuitions:
  - *"specify for each constraint the actions to be performed to satisfy it"* [Flesca et al., PPDP 2004]
  - *"preferred basic actions to repair [a constraint], if it is violated"* [Caroprese&Truszczynski, TPLP 2011]
- several declarative semantics

Active integrity constraints [Flesca et al., PPDP 2004]

"if *condition* holds then do *action*"

- active IC = negation of static IC + update actions

  Static : $(\forall x)[\mathtt{Bachelor}(x) \wedge \mathtt{Married}(x) \rightarrow \bot]$

  Active : $(\forall x)[\mathtt{Bachelor}(x) \wedge \mathtt{Married}(x), \{-\mathtt{Bachelor}(x)\}]$

- intuitions:
  - *"specify for each constraint the actions to be performed to satisfy it"* [Flesca et al., PPDP 2004]
  - *"preferred basic actions to repair [a constraint], if it is violated"* [Caroprese&Truszczynski, TPLP 2011]
- several declarative semantics
- sometimes still not enough information
  - which AIC for $\mathtt{emp}_{e,d_1} \wedge \mathtt{emp}_{e,d_2} \rightarrow \bot$?

Event-Condition-Action (ECA) rules [Ceri&Widow, 1994]

"if *event* occurs and *condition* holds then trigger *action*"

- now we can implement priority to the input:
  - if $+\mathtt{emp}_{e,d_1}$ occurs and $\mathtt{emp}_{e,d_1} \wedge \mathtt{emp}_{e,d_2}$ holds then $-\mathtt{emp}_{e,d_2}$
  - if $+\mathtt{emp}_{e,d_2}$ occurs and $\mathtt{emp}_{e,d_1} \wedge \mathtt{emp}_{e,d_2}$ holds then $-\mathtt{emp}_{e,d_1}$
- active database = database + set of ECA rules
  - huge literature [Ceri et al., ACM TDS 1994; Widom&Ceri, 1996; Chomicki&Marcinkowski, IC 2005,...]
- problem: chaining repairs may not terminate
- problem: no declarative semantics

  "lack of declarative semantics makes it difficult to understand the behavior of multiple ECAs acting together and to evaluate rule-processing algorithms in a principled way" [Cruz Filipe, 2016]

Aims of talk

1. revisit existing AIC semantics
   - rationality postulates (inspired by belief revision literature)
2. generalise to ECA rules [Herzig et al., FoIKS 2022]
   - analysis in dynamic logic
3. sketch transfer to description logics
   - active TBoxes

Outline

1 Motivation: repair via more informative integrity constraints

2 Active Integrity Constraints: postulates

3 Active Integrity Constraints: revisiting the existing semantics

4 ECA constraints

5 An analysis in dynamic logic

6 From AICs to active TBoxes: a sketch

7 Conclusion

Propositional databases and static integrity constraints

- propositional logic
  - hypothesis: everything is grounded
  - propositional variables $\mathbb{P} = \{p, q \ldots\}$
- databases = sets of propositional variables $\mathcal{D} \subseteq \mathbb{P}$
- update of $\mathcal{D}$ by a set of update actions $A = \{-p, +q, \ldots\}$:

$$\mathcal{D} \circ A = \big(\mathcal{D} \setminus \{p : -p \in A\}\big) \cup \{q : +q \in A\}$$

- static integrity constraints = set of clauses $\mathbf{IC}$

Active Integrity Constraints: syntax

$$r = \langle \mathbf{C}(r), \mathbf{A}(r) \rangle$$

- $\mathbf{C}(r)$ is a conjunction of literals (the negation of an IC)
- $\mathbf{A}(r)$ is a set of update actions ('repair options')
  - each $\alpha \in \mathbf{A}(r)$ makes some literal of $\mathbf{C}(r)$ false:
    - if $+p \in \mathbf{A}(r)$ then $\neg p \in \mathbf{C}(r)$
    - if $-p \in \mathbf{A}(r)$ then $p \in \mathbf{C}(r)$

- database $\mathcal{D}$ + finite set of AICs $R = \{r_1, \ldots, r_n\}$

$$R_1 = \{\langle \texttt{Bachelor} \land \texttt{Married}, \{-\texttt{Bachelor}\}\rangle\}$$

$$R_2 = \{\langle \texttt{Bachelor} \land \texttt{Married}, \{-\texttt{Bachelor}, -\texttt{Married}\}\rangle\}$$

- static constraints associated to R:
  $\mathbf{IC}(R) = \bigwedge\{\neg\mathbf{C}(r) \ : \ r \in R\}$

Active Integrity Constraints: which semantics?

Various semantics

- repairs *tout court*, alias PMA repairs ($\mathbf{A}(r)$ superfluous)
- founded repairs [Caroprese et al., ICLP 2006]
- justified repairs [Caroprese&Truszczynski, TPLP 2011]
- well-founded repairs [Cruz Felipe et al., TASE 2013]
- dynamic repairs [Feuillade&Herzig, JELIA 2013]
- grounded repairs [Bogaerts&Cruz Felipe, AIJ 2018]
- ...

... and each in several versions

- drop minimality requirement $\Longrightarrow$ weak versions
  - for PMA repairs: makes updates drastic
- minimise exceptions
  - preferred update actions are soft constraints, can be violated
  - if static part of R consistent then repair always exists

Active Integrity Constraints: which intuitions?

Permission vs. obligation

when condition $\mathbf{C}(r)$ is violated:

1. permission that the repair contains some $\alpha \in \mathbf{A}(r)$

   "If $\mathcal{D} \models \mathbf{C}(r)$, then $\mathcal{D}$ is inconsistent. It is allowed to repair this inconsistency by executing one or more of the $\alpha_i \in \mathbf{A}(r)$."
   [Bogaerts&Cruz Felipe, AIJ 2018] (notation adapted)

Active Integrity Constraints: which intuitions?

Permission vs. obligation

when condition $\mathbf{C}(r)$ is violated:

1. permission that the repair contains some $\alpha \in \mathbf{A}(r)$

   "If $\mathcal{D} \models \mathbf{C}(r)$, then $\mathcal{D}$ is inconsistent. It is allowed to repair this inconsistency by executing one or more of the $\alpha_i \in \mathbf{A}(r)$."
   [Bogaerts&Cruz Felipe, AIJ 2018] (notation adapted)

   ...but $\mathbf{C}(r)$ might as well be repaired by other AICs

2. obligation that the repair contains some $\alpha \in \mathbf{A}(r)$

   To guarantee that $\mathcal{D} \circ A$ satisfies r, $\mathcal{D} \circ A$ must falsify at least one literal in $\mathbf{C}(r)$. To this end A must contain at least one update action from $\mathbf{A}(r)$."
   [Caroprese&Truszczynski, TPLP 2011] (notation adapted)

Active Integrity Constraints: different intuitions, ctd.

Permission vs. obligation: consequences

when $\mathbf{C}(r)$ is true …

1. 'permission' reading:

$$\langle \mathbf{C}(r), \{+p, +q\}\rangle \text{ equivalent to } \begin{cases} \langle \mathbf{C}(r), \{+p\}\rangle \\ \langle \mathbf{C}(r), \{+q\}\rangle \end{cases}$$

   $\Longrightarrow$ all $\mathbf{A}(r)$ singletons ("R normalised")

2. 'obligation' reading:
   - R cannot be normalised
   - computation more local than 'permission' reading:
     "if $\mathbf{C}(r)$ is true then repair via $\mathbf{A}(r)$ regardless of other AICs"

… but what does "$\mathbf{C}(r)$ is true" mean? Just "$\mathcal{D} \models \mathbf{C}(r)$"?

Active Integrity Constraints: sharpening intuitions by means of abstract examples (1)

Example: one violation, no interaction

$$\mathcal{D} = \emptyset \text{ and } R = \big\{ \langle \neg p \wedge \neg q, \{+p, +q\} \rangle, \\ \langle \neg q \wedge r, \{+q\} \rangle \big\}$$

- PMA repairs are $A_1 = \{+p\}$ and $A_2 = \{+q\}$
- principle:

  if $\mathcal{D} \models \mathbf{C}(r)$ and for all other r',

  $\mathcal{D} \not\models \mathbf{C}(r')$ and r' does not *interact* with r,

  then the repairs are just the update actions in $\mathbf{A}(r)$

Active Integrity Constraints: sharpening intuitions by means of abstract examples (2)

Example: two violations, no interaction

$$\mathcal{D} = \emptyset \text{ and } R = \{\langle \neg p \wedge \neg q, \{+p\}\rangle, \langle \neg p \wedge \neg q, \{+q\}\rangle\ \}$$

- different readings lead to different intuitions
  - 'permission': repairs are $A_1 = \{+p\}$ and $A_2 = \{+q\}$
  - 'obligation': repair is $A = \{+p, +q\}$
    - A is not minimal $\Longrightarrow$ not a PMA repair!
    - active part of R badly designed?

Active Integrity Constraints: sharpening intuitions by means of abstract examples (3)

Example: one violation, with interaction

$$\mathcal{D} = \emptyset \text{ and } R = \big\{\langle \neg p, \{+p\}\rangle, \\ \langle p \wedge \neg q, \{+q\}\rangle\big\}$$

- repair is $A = \{+p, +q\}$
  - $\mathcal{D} \not\models p \wedge \neg q$
  - but $\mathcal{D} \circ \{+p\} \models p \wedge \neg q$!
- hence $\mathcal{D} \not\models \mathbf{C}(r)$ not enough a criterion to trigger an AIC
- in general: membership in A may have to be hypothesised
  - problem: circularity of support (v.i.)

Active Integrity Constraints: sharpening intuitions by means of abstract examples (4)

Example: one violation, with interaction

$$\mathcal{D} = \emptyset \text{ and } R = \{ \langle \neg p \wedge \neg q, \{+p\} \rangle, \langle \neg p \wedge q, \{+p\} \rangle, \langle p \wedge \neg q, \{+q\} \rangle \}$$

- intuitions differ
  1. "circularity of support" $\Longrightarrow$ should have no repair [Caroprese&Truszczynski, TPLP 2011]
  2. we: repair should be $A = \{+p, +q\}$
     reason: identity principle applies to first two AICs
     - disjunction of static parts equivalent to $\neg p$
     - dynamic parts identical

  $\Longrightarrow$ first two AICs should be equivalent to $\langle \neg p, \{+p\} \rangle$

Active Integrity Constraints: summary of intuitions

1. permission reading and obligation reading come with different intuitions
    - obligations more local
        - should lead to a simpler account
    - obligation reading often leads to non-minimal repairs
        - may indicate flawed choices of update actions
2. and a crucial principle: identity
    - cf. revision&update postulates...

The Katsono-Mendelzon (KM) postulates for belief update

$\varphi \diamond \psi =$ update of φ by ψ (φ, ψ: propositional formulas)

(U1) $\varphi \diamond \psi \to \psi$;

(U2) If $\varphi \to \psi$ then $(\varphi \diamond \psi) \leftrightarrow \varphi$;

(U3) $\varphi \diamond \psi \to \bot$ if and only if ($\varphi \to \bot$ or $\psi \to \bot$);

(U4) If $\varphi_1 \leftrightarrow \varphi_2$ and $\psi_1 \leftrightarrow \psi_2$ then $(\varphi_1 \diamond \psi_1) \leftrightarrow (\varphi_2 \diamond \psi_2)$;

(U5) $((\varphi \diamond \psi_1) \wedge \psi_2) \to (\varphi \diamond (\psi_1 \wedge \psi_2))$;

(U6) If $(\varphi \diamond \psi_1) \to \psi_2$ then $(\varphi \diamond (\psi_1 \wedge \psi_2)) \to (\varphi \diamond \psi_1)$;

(U7) If φ is complete then $((\varphi \diamond \psi_1) \wedge (\varphi \diamond \psi_2)) \to (\varphi \diamond (\psi_1 \vee \psi_2))$;

(U8) $((\varphi_1 \vee \varphi_2) \diamond \psi) \leftrightarrow ((\varphi_1 \diamond \psi) \vee (\varphi_2 \diamond \psi))$.

The KM postulates for database repair

R = set of AICs
$\mathbf{IC}(R)$ = static constraints associated to R
$\mathsf{Rep}(\mathcal{D}, R)$ = possible repairs of $\mathcal{D}$ via R

(R1) For every $A \in \mathsf{Rep}(\mathcal{D}, R)$, $\mathcal{D} \circ A \models \mathbf{IC}(R)$;

(R2) If $\mathcal{D} \models \mathbf{IC}(R)$ then $\mathsf{Rep}(\mathcal{D}, R) = \{\emptyset\}$;

(R3) If R is *closed* and $\mathbf{IC}(R)$ is consistent then $\mathsf{Rep}(\mathcal{D}, R) \neq \emptyset$;

(R4) If R_1 and R_2 are *equivalent* then $\mathsf{Rep}(\mathcal{D}, R_1) = \mathsf{Rep}(\mathcal{D}, R_2)$;

(R5) If $A \in \mathsf{Rep}(\mathcal{D}, R_1)$ and $\mathcal{D} \circ A \models \mathbf{C}(R_2)$
then $A \in \mathsf{Rep}(\mathcal{D}, R_1 \cup R_2)$;

(R6) If R_1 is *closed* and $\mathcal{D} \circ A \models \mathbf{C}(R_2)$ for every $A \in \mathsf{Rep}(\mathcal{D}, R_1)$
then $\mathsf{Rep}(\mathcal{D}, R_1 \cup R_2) \subseteq \mathsf{Rep}(\mathcal{D}, R_2)$;

(R7) If $A \in \mathsf{Rep}(\mathcal{D}, R \cup \{\langle \mathbf{C}_1, A_1 \rangle\})$
and $A \in \mathsf{Rep}(\mathcal{D}, R \cup \{\langle \mathbf{C}_2, A_2 \rangle\})$
then $A \in \mathsf{Rep}(\mathcal{D}, R \cup \{\langle \mathbf{C}_1 \cup \mathbf{C}_2, A_1 \cup A_2 \rangle\})$.

Extensionality: when are two sets of AICs equivalent?

Definitions

1. $R \approx_0 R'$ if $R = R'$
2. $R \approx_1 R'$ if $\mathbf{IC}(R) \leftrightarrow \mathbf{IC}(R')$ and
   all AICs have same action set
   (there is an A such that $\mathbf{A}(r) = A$ for every $r \in R \cup R'$)

$$\{\langle \neg p \wedge \neg q, \{+p\}\rangle, \langle \neg p \wedge q, \{+p\}\rangle\} \quad \approx_1 \quad \{\langle \neg p, \{+p\}\rangle\}$$

3. $R \approx_2 R'$ if there are partitions of R and R' with $\approx_1$-equivalent cells
   - for every cell ρ of R there is a cell ρ' of R' such that $\rho \approx_1 \rho'$
   - for every cell ρ' of R' there is a cell ρ of R such that $\rho \approx_1 \rho'$

$$\{\langle p \wedge \neg q, \{+q\}\rangle, \langle \neg p \wedge \neg q, \{+p\}\rangle, \langle \neg p \wedge q, \{+p\}\rangle\} \quad \approx_2 \quad \{\langle p \wedge \neg q, \{+q\}\rangle, \langle \neg p, \{+p\}\rangle\}$$

Outline

1. Motivation: repair via more informative integrity constraints
2. Active Integrity Constraints: postulates
3. Active Integrity Constraints: revisiting the existing semantics
4. ECA constraints
5. An analysis in dynamic logic
6. From AICs to active TBoxes: a sketch
7. Conclusion

Well-Founded Repairs [Bogaerts&Cruz Felipe, AIJ 2018]

Idea

1. choose a violated AIC: an $r \in R$ such that $\mathcal{D} \models \mathbf{C}(r)$
2. update by one of the actions in the active part $\mathbf{A}(r)$
3. iterate until no more violation, avoiding repetitions of assignments

Definition

PMA repair A for $\mathcal{D}$ w.r.t. R is well-founded if $A = \{\alpha_1, \ldots, \alpha_n\}$ (for some ordering) such that for every α_i there is an AIC $r_i \in R$ with

- $\alpha_i \in \mathbf{A}(r_i)$
- $\mathcal{D} \circ \{\alpha_1, \ldots, \alpha_{i-1}\} \models \mathbf{C}(r_i)$ (r_i is violated)

compatible with permission reading and with obligation reading

Founded Repairs [Caroprese et al., ICLP 2006]

Idea

- update actions $\alpha \in A$ should be supported by some active constraint r
  - r would be violated without α

Definition

PMA repair A for $\mathcal{D}$ w.r.t. R is founded if

for every $\alpha \in A$ there is an AIC $r \in R$ such that

- $\alpha \in \mathbf{A}(r)$ and
- $\mathcal{D} \circ (A \setminus \{\alpha\}) \models \mathbf{C}(r)$

permission reading (definition checks that every $\alpha \in A$ is permitted)

Grounded Repairs [Bogaert&Cruz Felipe, AIJ 2018]

Idea

- generalises negative condition of foundedness

$$\mathcal{D} \circ (A \setminus \{\alpha\}) \models \mathbf{C}(r)$$

- hypothesis: all $\mathbf{A}(r)$ are singletons ('all AICs are normal')

Definition

PMA repair A of $\mathcal{D}$ w.r.t. R is grounded if

for every $A' \subset A$ there is a $r \in R$ such that

- $\mathbf{A}(r) \cap (A \setminus A') \neq \emptyset$ and
- $\mathcal{D} \circ A' \models \mathbf{C}(r)$

Properties

- all grounded repairs are well-founded, founded, minimal

Justified repairs [Caroprese&Truszczynski, TPLP 2011]

Definitions

- *non-effect actions* w.r.t. $\mathcal{D}$ and A:

$$\mathtt{neff}_{\mathcal{D}}(A) = \{\alpha : \mathcal{D} \circ \alpha = \mathcal{D} \text{ and } (\mathcal{D} \circ A) \circ \alpha = \mathcal{D} \circ A\}$$

- *non-updatable literals* of r:

$$\mathtt{nup}(r) = \{p \in \mathbf{C}(r) : +p \notin \mathbf{A}(r)\} \cup \{\neg p \in \mathbf{C}(r) : -p \notin \mathbf{A}(r)\}$$

- A is *closed under* R if for each $r \in R$,
  - if $-p \in A$ for every $p \in \mathtt{nup}(r)$ and $+p \in A$ for every $\neg p \in \mathtt{nup}(r)$ (r must be triggered)
  - then $\mathbf{A}(r) \cap A \neq \emptyset$
- A is a *justified action set* if it is a minimal superset of $\mathtt{neff}_{\mathcal{D}}(A)$ closed under R
- PMA repair A of $\mathcal{D}$ w.r.t. R is justified if $A \cup \mathtt{neff}_{\mathcal{D}}(A)$ is a justified action set

Justified repairs

Properties

- obligation reading
- violate identity principle!
    - example [Caroprese&Truszczynski, TPLP 2011]:

$$\mathcal{D} = \emptyset \text{ and } R = \{ \langle \neg p \wedge \neg q, \{+p\} \rangle, \langle \neg p \wedge q, \{+p\} \rangle, \langle p \wedge \neg q, \{+q\} \rangle \}$$

- has no justified repair
- identity principle:

$$R \approx_2 \{\langle \neg p, \{+p\} \rangle, \langle p \wedge \neg q, \{+q\} \rangle\}$$

$$\implies \text{Rep}(\mathcal{D}, R) = \{\{+p, +q\}\}$$

Outline

1. Motivation: repair via more informative integrity constraints
2. Active Integrity Constraints: postulates
3. Active Integrity Constraints: revisiting the existing semantics
4. ECA constraints
5. An analysis in dynamic logic
6. From AICs to active TBoxes: a sketch
7. Conclusion

Event-Condition-Action (ECA) constraints: syntax

$$\kappa = \langle \mathbf{E}(\kappa), \mathbf{C}(\kappa), \mathbf{A}(\kappa) \rangle$$

- $\langle \mathbf{C}(\kappa), \mathbf{A}(\kappa) \rangle$ is an AIC
- $\mathbf{E}(\kappa)$ is a boolean formula built from assignments
  - partial description of last update actions
- database $\mathcal{D}$ + finite sets of ECA constraints $\mathcal{K} = \{\kappa_1, \ldots, \kappa_n\}$

$$\mathcal{K}_{\mathtt{emp}} = \{ \langle +\mathtt{emp}_{e,d_1}, \mathtt{emp}_{e,d_1} \wedge \mathtt{emp}_{e,d_2}, \{-\mathtt{emp}_{e,d_2}\} \rangle, \\ \langle +\mathtt{emp}_{e,d_2}, \mathtt{emp}_{e,d_1} \wedge \mathtt{emp}_{e,d_2}, \{-\mathtt{emp}_{e,d_1}\} \rangle \}$$

- static constraints $\mathbf{IC}(\mathcal{K}) = \bigwedge \{ \neg \mathbf{C}(\kappa) \,:\, \kappa \in \mathcal{K} \}$
  - $\mathcal{D} \models \mathbf{IC}(\mathcal{K})$: all the static constraints hold
  - $\mathcal{D} \not\models \mathbf{IC}(\mathcal{K})$: database has to be repaired

ECA constraints: example

Example (adapted from [Flesca et al., PPDP 2004])

- every manager of a project carried out by a department must be an employee of that department
- if e just became manager of project p or if p was just assigned to d_1 then e should become a member of d_1
- if e has just been removed from d_1 then the project should either be removed from d_1, too, or should get a new manager

$$\mathcal{K} = \mathcal{K}_{\text{emp}} \cup \{ \langle +\mathtt{mgr}_{e,p} \vee +\mathtt{prj}_{p,d_1}, \mathtt{mgr}_{e,p} \wedge \mathtt{prj}_{p,d_1} \wedge \neg\mathtt{emp}_{e,d_1}, \{+\mathtt{emp}_{e,d_1}\}\rangle, \langle -\mathtt{emp}_{e,d_1}, \mathtt{mgr}_{e,p} \wedge \mathtt{prj}_{p,d_1} \wedge \neg\mathtt{emp}_{e,d_1}, \{-\mathtt{mgr}_{e,p}, -\mathtt{prj}_{p,d_1}\}\rangle \}$$

- when last update $+\mathtt{mgr}_{e,p}$ then repair by $\{+\mathtt{emp}_{e,d_1}, -\mathtt{emp}_{e,d_2}\}$

ECA rules: too complicated for a declarative semantics?

- mainly procedural semantics
  - termination problems
- only few declarative semantics
  - logic programming semantics [...]
  - Sitcalc [Bertossi&Pinto, 1999; Kiringa, LICS 2001; Kiringa&Reiter 2003]
- can we import AIC semantics?
  - *"[AIC] formalisms do not allow us to specify triggering events"* [Caroprese&Truszczynski, TPLP 2011]

ECA constraints: models

Adding immediate past events

$$\boxed{\text{model } \Delta = \langle \mathcal{D}, \mathcal{H} \rangle}$$

- $\mathcal{D} \subseteq \mathbb{P}$ database
- $\mathcal{H} \subseteq \{+p : p \in \mathbb{P}\} \cup \{-p : p \in \mathbb{P}\}$
  - most recent update action that brought database into state $\mathcal{D}$
  - if $+p \in \mathcal{H}$ then $p \in \mathcal{D}$
  - if $-p \in \mathcal{H}$ then $p \notin \mathcal{D}$
- semantics
  - $\langle \mathcal{D}, \mathcal{H} \rangle \models p$ if $p \in \mathcal{D}$
  - $\langle \mathcal{D}, \mathcal{H} \rangle \models +p$ if $+p \in \mathcal{H}$
  - $\langle \mathcal{D}, \mathcal{H} \rangle \models -p$ if $-p \in \mathcal{H}$
- import founded and well-founded semantics. . .

Founded ECA repairs

- update action A coherent (no $p \in A$ with $+p, -p \in A$)
- update of history $\mathcal{H}$ by A:

$$\mathcal{H} \circ A = A \cup \{\alpha \in \mathcal{H} \,:\, A \cup \{\alpha\} \text{ is consistent}\}$$

Definition

PMA repair A for $\langle \mathcal{D}, \mathcal{H} \rangle$ w.r.t. $\mathcal{K}$ is founded if for every $\alpha \in A$ there is an ECA rule $\kappa \in \mathcal{K}$ with

- $\alpha \in \mathbf{A}(\kappa)$
- $\mathcal{D} \circ (A \setminus \{\alpha\}) \models \mathbf{C}(\kappa)$
- $\mathcal{H} \circ (A \setminus \{\alpha\}) \models \mathbf{E}(\kappa)$

Founded ECA repairs: example

Example, ctd.

$$\Delta = \langle\{\mathtt{mgr}_{e,p}, \mathtt{prj}_{p,d_1}, \mathtt{emp}_{e,d_2}\}, \{+\mathtt{mgr}_{e,p}\}\rangle$$

$$\mathcal{K} = \mathcal{K}_{\mathtt{emp}} \cup$$

$$\{\, \langle +\mathtt{mgr}_{e,p} \vee +\mathtt{prj}_{p,d_1}, \mathtt{mgr}_{e,p} \wedge \mathtt{prj}_{p,d_1} \wedge \neg\mathtt{emp}_{e,d_1}, \{+\mathtt{emp}_{e,d_1}\}\rangle,$$

$$\langle -\mathtt{emp}_{e,d_1}, \mathtt{mgr}_{e,p} \wedge \mathtt{prj}_{p,d_1} \wedge \neg\mathtt{emp}_{e,d_1}, \{-\mathtt{mgr}_{e,p}, -\mathtt{prj}_{p,d_1}\}\rangle \,\}$$

unique founded ECA repair:

$$A = \{+\mathtt{emp}_{e,d_1}, -\mathtt{emp}_{e,d_2}\}$$

Well-founded ECA repairs

Definition

PMA repair A for $\langle \mathcal{D}, \mathcal{H} \rangle$ w.r.t. $\mathcal{K}$ is well-founded if $A = \{\alpha_1, \ldots, \alpha_n\}$ (for some ordering) such that for every α_i there is an ECA rule $\kappa_i \in \mathcal{K}$ with

- $\alpha_i \in \mathbf{A}(\kappa_i)$
- $\mathcal{D} \circ \{\alpha_1, \ldots, \alpha_{i-1}\} \models \mathbf{C}(\kappa_i)$
- $\mathcal{H} \circ \{\alpha_1\} \circ \ldots \circ \{\alpha_{i-1}\} \models \mathbf{E}(\kappa_i)$

Example, ctd.

well-founded repair of $\langle \{\mathtt{mgr}_{e,p}, \mathtt{prj}_{p,d_1}, \mathtt{emp}_{e,d_2}\}, \{+\mathtt{mgr}_{e,p}\} \rangle$:

$$A = \{+\mathtt{emp}_{e,d_1}, -\mathtt{emp}_{e,d_2}\}$$

can be captured in dynamic logic. . .

Outline

1. Motivation: repair via more informative integrity constraints
2. Active Integrity Constraints: postulates
3. Active Integrity Constraints: revisiting the existing semantics
4. ECA constraints
5. An analysis in dynamic logic
6. From AICs to active TBoxes: a sketch
7. Conclusion

Propositional dynamic logic PDL

[Pratt, 1976; Fischer and Ladner, 1979]

- formulas $\langle\pi\rangle\varphi =$ "there is an execution of π after which φ is true"
- programs $\pi_1;\pi_2$, $\pi_1 \cup \pi_2$, π^*, $\varphi?$
  - capture familiar programming constructions:

    "**if** φ **then** π_1 **else** π_2" and "**while** φ **do** π"
  - atomic programs: abstract (just as propositional variables)

Dynamic Logic of Propositional Assignments DL-PA [Balbiani et al., 2013]

- PDL atomic programs $\Longrightarrow$ *atomic assignments* $+p$, $-p$
- concrete atomic programs:
  - $+p$, $-p$: assignments

Syntax of DL-PA$^{\pm}$

Grammar

$$\varphi ::= p \mid +p \mid -p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\pi\rangle\varphi$$
$$\pi ::= A \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \mid \varphi?$$

where p ranges over $\mathbb{P}$ and A over the set of assignments $\{+p : p \in \mathbb{P}\} \cup \{-p : p \in \mathbb{P}\}$

- $\{+p\}$ is a program: "make p true"
- $+p$ is a formula: "p has just been made true"

Semantics of DL-PA$^{\pm}$

For models $\Delta = \langle \mathcal{D}, \mathcal{H} \rangle$

interpretation of formulas:

$$\begin{aligned}
\Delta &\models p && \text{if} \quad p \in \mathcal{D} \\
\Delta &\models +p && \text{if} \quad +p \in \mathcal{H} \\
\Delta &\models -p && \text{if} \quad -p \in \mathcal{H} \\
\Delta &\models \langle \pi \rangle \varphi && \text{if} \quad \Delta ||\pi|| \Delta' \text{ and } \Delta' \models \varphi \text{ for some } \Delta'
\end{aligned}$$

interpretation of programs:

$$\begin{aligned}
\Delta ||A|| \Delta' &\quad \text{if} \quad A \text{ is consistent and } \Delta' = \langle \mathcal{D} \circ A, \mathcal{H} \circ A \rangle \\
\Delta ||\pi_1 ; \pi_2|| \Delta' &\quad \text{if} \quad \Delta ||\pi_1|| \Delta'' \text{ and } \Delta'' ||\pi_2|| \Delta' \text{ for some } \Delta'' \\
\Delta ||\pi_1 \cup \pi_2|| \Delta' &\quad \text{if} \quad \Delta ||\pi_1|| \Delta' \text{ or } \Delta ||\pi_2|| \Delta' \\
\Delta ||\pi^*|| \Delta' &\quad \text{if} \quad \Delta ||\pi^n|| \Delta' \text{ for some } n \geq 0 \\
\Delta ||\varphi ?|| \Delta' &\quad \text{if} \quad \Delta \models \varphi \text{ and } \Delta' = \Delta
\end{aligned}$$

Properties of DL-PA$^{\pm}$

Semantics

- based on classical valuations
  - no Kripke models needed
- Kleene star can be eliminated (not possible in PDL)

$$\langle \pi^* \rangle \varphi \leftrightarrow \langle \pi^{\leq 2^{\mathsf{card}(\mathbb{P}_\pi)}} \rangle \varphi$$

  - consequence: all dynamic operators can be eliminated

PDL vs. DL-PA$^{\pm}$: complexity of decision problems

	PDL	DL-PA$^{\pm}$
Model checking	PTIME-complete	PSPACE-complete
Satisfiability	EXPTIME-complete	PSPACE-complete

Characterising weak well-founded ECA repairs

- fresh auxiliary propositional variables $\mathsf{D}(\alpha)$: record that assignment α has been executed
- legal repair by assignment α:

$$\mathsf{rep}(\alpha) = \neg\mathsf{D}(\alpha) \land \neg\mathsf{D}(\overline{\alpha}) \land \bigvee_{\kappa\in\mathcal{K}\,:\,\alpha\in\mathbf{A}(\kappa)} (\mathbf{E}(\kappa) \land \mathbf{C}(\kappa))?; \{\alpha, +\mathsf{D}(\alpha)\}$$

- program $-\mathsf{D}(A)$ initialises the auxiliary $\mathsf{D}(\alpha_i)$ and $\mathsf{D}(\overline{\alpha}_i)$ to false

Theorem

The set of assignments $A = \{\alpha_1, \ldots, \alpha_n\}$ is a well-founded weak ECA repair of Δ via $\mathcal{K}$ if and only if the DL-PA$^{\pm}$ *program*

$$\mathsf{rep}^{\mathsf{wwf}}_{\mathcal{K}}(A) = -\mathsf{D}(A); \Big(\bigcup_{\alpha\in A} \mathsf{rep}(\alpha) \Big)^*; \mathbf{IC}(\mathcal{K})?; \bigwedge_{\alpha\in A} \mathsf{D}(\alpha)?$$

is executable at Δ.

Characterising well-founded ECA repairs

- fresh auxiliary propositional variables p': store initial value of p in order to check for minimal change
  - program init(A): initialises all D(α) to false and copies the initial values of p into p'
  - program $\mathtt{undo}(\alpha)$: restores original truth value of the variable assigned in α

Theorem

The set of assignments $A = \{\alpha_1, \ldots, \alpha_n\}$ *is a well-founded ECA repair of* Δ *via* $\mathcal{K}$ *if and only if the* DL-PA$^{\pm}$ *program*

$$\mathsf{rep}^{\mathsf{wf}}_{\mathcal{K}}(A) = \mathsf{init}(A); \Big(\bigcup_{\alpha \in A} \mathsf{rep}(\alpha)\Big)^*; \mathbf{IC}(\mathcal{K})?; \Big(\bigwedge_{\alpha \in A} \mathsf{D}(\alpha)\Big)?;$$

$$\neg \Big\langle \Big(\bigcup_{\alpha \in A} \mathtt{undo}(\alpha)\Big)^+ \Big\rangle \mathbf{IC}(\mathcal{K})?$$

is executable at Δ.

Other decision problems

- founded repairs
- Does there exist a repair of Δ via $\mathcal{K}$?
- Is there a unique repair of Δ via $\mathcal{K}$?
- Does every Δ have a unique repair via $\mathcal{K}$?
- Does every Δ have a repair via $\mathcal{K}$?
- Are $\mathcal{K}_1$ and $\mathcal{K}_2$ equivalent?
  - do they repair any Δ in the same way?
- Can $\mathcal{K}_1$ repair strictly more than $\mathcal{K}_2$?
- May the repair of Δ via $\mathcal{K}$ loop?
  - drop the $\mathsf{D}(\alpha)$ tests from $\mathsf{rep}(\alpha)$
  - check whether $\Delta \models [(\mathsf{urep}_{\mathcal{K}})^*]\langle \mathsf{urep}_{\mathcal{K}} \rangle \top$

$\Longrightarrow$ can all can be expressed in DL-PA$^{\pm}$

Outline

1. Motivation: repair via more informative integrity constraints
2. Active Integrity Constraints: postulates
3. Active Integrity Constraints: revisiting the existing semantics
4. ECA constraints
5. An analysis in dynamic logic
6. From AICs to active TBoxes: a sketch
7. Conclusion

From DBs to description logic KBs

Repairs of inconsistent KBs in the DL literature

- *remove* axioms [Schlobach&Cornet, 2003]
- *weaken* axioms [Troquard et al., 2018]
- main methods: axiom pinpointing, justifications, hitting set, weakening of axioms
- usually no preference between possible repairs considered

Can we import the idea of active constraints?

A simple example

Example TBox

$$\mathcal{T} = \{\mathtt{Father} \sqsubseteq \mathtt{Male} \sqcap \mathtt{Parent}, \\ \mathtt{OnlyChild} \sqsubseteq \forall \mathtt{hasSibling}.\bot\}$$

An ABox inconsistent with $\mathcal{T}$

$$\mathcal{A} = \{\mathtt{John}:\mathtt{Male} \sqcap \mathtt{Father} \sqcap \neg\mathtt{Parent}, \\ \mathtt{Mary}:\mathtt{OnlyChild}, \\ \mathtt{hasSibling}(\mathtt{Mary}, \mathtt{John})\}$$

A simple example

An enhanced TBox extending $\mathcal{T}$

$$\mathtt{a}\mathcal{T}_1 = \{\, \langle \mathtt{Father} \sqsubseteq \mathtt{Male} \sqcap \mathtt{Parent}, \{+\mathtt{Male}, +\mathtt{Parent}\}\rangle, \langle \mathtt{OnlyChild} \sqsubseteq \forall \mathtt{hasSibling}.\bot, \{-\mathtt{OnlyChild}\}\rangle \,\}$$

A repaired ABox consistent with $\mathtt{a}\mathcal{T}$

$$\mathcal{A}_1 = \{\mathtt{John} : \mathtt{Male} \sqcap \mathtt{Father} \sqcap \mathtt{Parent}, \mathtt{Mary} : \neg \mathtt{OnlyChild}, \mathtt{hasSibling}(\mathtt{Mary}, \mathtt{John})\}$$

A simple example

An enhanced TBox extending $\mathcal{T}$

$$\mathtt{a}\mathcal{T}_2 = \{ \langle \mathtt{Father} \sqsubseteq \mathtt{Male} \sqcap \mathtt{Parent}, \{-\mathtt{Father}\}\rangle, \langle \mathtt{OnlyChild} \sqsubseteq \forall \mathtt{hasSibling}.\bot, \{-\mathtt{hasSibling}.\top\}\rangle \}$$

A repaired ABox consistent with a$\mathcal{T}$

$$\mathcal{A}_2 = \{\mathtt{John}:\mathtt{Male} \sqcap \neg\mathtt{Father} \sqcap \neg\mathtt{Parent}, \mathtt{Mary} : \mathtt{OnlyChild}\}$$

Repairs based on active TBoxes: difficulties

Challenges

1. ABoxes have concept constructors & complex concepts
   $\Longrightarrow$ 'atomic' update actions not enough
2. *closed world* semantics vs. *open world* semantics
   $\Longrightarrow$ *satisfiability checking* instead of *model checking*
3. *removing* vs. *forgetting* concepts
   $\Longrightarrow$ choice

Proposals

- [Rantsoudis et al., DL 2017]: syntactic approach
- [Feuillade et al., DL 2018]: semantic approach
  - logic $\mathtt{dyn}\mathcal{ALCO}$
- [Rantsoudis, PhD 2018]

Outline

1. Motivation: repair via more informative integrity constraints
2. Active Integrity Constraints: postulates
3. Active Integrity Constraints: revisiting the existing semantics
4. ECA constraints
5. An analysis in dynamic logic
6. From AICs to active TBoxes: a sketch
7. Conclusion

In summary

- reexamined intuitions behind active integrity constraints
  - permission vs. obligation to choose update action
  - rationality postulates for repair
  - justified repairs violate principle of identity
- generalised two AIC semantics to ECA rules
  - founded & well-founded semantics
- sketched repairs based on active TBoxes

Moreover

- analysis in dynamic logic
  - Kleene star accounts nicely for terminating executions only
  - reasoning about repairs *in the logic*