Belief revision theory and its applications: a manifesto

Andreas Herzig
U. of Toulouse and CNRS, IRIT, France

BRA workshop, Ponta Delgada, Feb. 9, 2015

Introduction

- revision operation $* : 2^{\mathtt{Fml}} \times \mathtt{Fml} \longrightarrow 2^{\mathtt{Fml}}$
  - $B * A$ = “belief state after input A is taken into account”
  - $B \in 2^{\mathtt{Fml}}$ = previous belief state
  - A = new piece of information (‘input’)
- belief revision understood in a large sense
  - includes belief update (differences won’t matter here)
  - perhaps better called belief change
- belief change theory = AGM/KM

What belief change theories are about

1. postulates = metalanguage axioms
   - $B * A \not\models \bot$
   - $B * A \models A$ ('success')
   - ...
2. semantics
   - $\texttt{Models}(B * A) = \min_{\leq_B} \texttt{Models}(A)$
   - $\leq_B$ = preorder on the set of valuations, indexed by B
     - comparative possibility
     - epistemic entrenchment
     - ...
     - "relates two imprecise concepts" [Lewis 1973]

What are the applications of belief change theories?

- philosophy:
  - derogation of laws [Alchourrón]
  - scientific theories [Gärdenfors]
  - …
- computer science:
  - databases
  - knowledge representation (ontologies)
  - BDI agents
  - planning
  - program synthesis
  - …

$\Rightarrow$ relevant for virtually any area

What does AGM/KM theory offer to computer science applications?

- “We got a Ferrari and we need a Fiat 500” [Fermé]
  - belief change is only *one* component of an intelligent system
  - in AI we also have to deal with goals and intentions, higher-order beliefs, normative constraints (obligations, permissions), plan generation, argumentation, …
    ⇒ belief change operation should be simple but versatile

What does AGM/KM theory offer to computer science applications? (ctd.)

1. we need one operation and AGM offers many
   - semantics: depends on a total preorder $\leq_B$
   - syntax: postulates don't identify a single $*$ but a family
     - compare to the 'postulates' for $\mathtt{Cn}$ (or $\vdash$) in proof theory: $\exists!$ consequence relation for classical (intuitionistic,...) logic
   - worse:
     - 20+ alternative frameworks [Rott, Hansson, Fermé,...]
     - theories of iterated belief revision [Darwiche&Pearl,...]
     - theories of syntax-based belief revision [Hansson, Nebel,...]
2. we need a simple operation and AGM is complicated
   - represent each total preorder $\leq_B$ on valuations: $2^{2^{\mathrm{card}(\mathtt{Fml})}}$ pairs!
3. AGM is heavily underconstrained
   - even the drastic $*_d$ satisfies the basic AGM postulates

$$B *_d A = \begin{cases} \mathtt{Cn}(A) & \text{if } \neg A \in \mathtt{Cn}(B) \\ \mathtt{Cn}(B \cup \{A\}) & \text{otherwise} \end{cases}$$

4. AGM is for classical propositional calculus
   - epistemic: $B * (p \wedge \neg \mathrm{K} p) \not\models p \wedge \neg \mathrm{K} p$ [Fuhrmann]

Some concrete belief change operations

- build orderings from symmetric difference between valuations

$$\begin{aligned} \mathrm{diff}(V, V_1) &= (V \setminus V_1) \cup (V_1 \setminus V) \\ V_1 <_V V_2 \quad &\text{iff} \quad \mathrm{diff}(V, V_1) \subset \mathrm{diff}(V, V_2) \end{aligned}$$

$\Rightarrow$ Winslett's update operator ('PMA'), Satoh's revision operator

- build orderings from card(diff(V, V′)) ('Hamming distance')

$\Rightarrow$ Forbus's update operator, Dalal's revision operator

- underlying hypothesis: if $p \neq q$ then p and q are independent
  - if $B \models q$ then $B * p \models q$
  - impossible to formulate integrity constraints, such as $p \rightarrow \neg q$ (but more later)
- only defined semantically (no axioms/postulates)
- $*$ is not in the object language

$$* : 2^{\mathrm{Lang(PC)}} \times \mathrm{Lang(PC)} \longrightarrow 2^{2^{\mathrm{Prp}}}$$

A computer science view: change beliefs = execute a program

- logic of (possibly nondeterministic) programs = dynamic logic

  $[\pi]B$ = "B is true after every possible execution of π"

  $\langle\pi\rangle B$ = "B is true after some possible execution of π"

- idea:
  1. associate a update/revision program π_A to A
  2. prove:

$$B * A \models_{\mathrm{PC}} C \quad \text{iff} \quad \models_{\mathrm{DL}} B \to \left[\pi_A\right]C$$
$$\text{iff} \quad \models_{\mathrm{DL}} \left\langle(\pi_A)^{-1}\right\rangle B \to C$$

hence:

$$B * A \quad \equiv \quad \left\langle(\pi_A)^{-1}\right\rangle B$$

An interesting dialect of dynamic logic

- Dynamic Logic of Propositional Assignments DL-PA

  [Herzig et al., IJCAI 2011, Balbiani et al., LICS 2012]

  - propositional assignments $+p$ and $-p$
  - 'DEL-like': reduction to propositional calculus PC
  - good mathematical properties (compact, interpolation, ...)
  - PSPACE complete (just as QBF)
- captures the existing concrete belief change operations

  [Herzig, KR 2014]

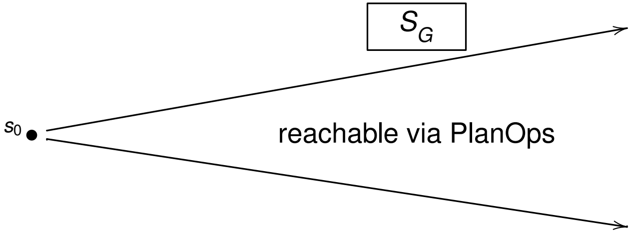
$$B *^{\text{pma}} A = \texttt{Models}\Big(\Big\langle(\pi_A^{\text{pma}})^{-1}\Big\rangle B\Big)$$

$$B *^{\text{forbus}} A = \ldots$$

$$B *^{\text{dalal}} A = \ldots$$

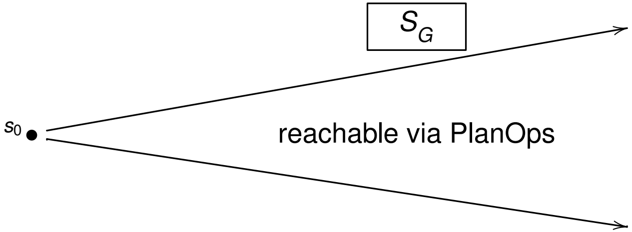
- programs make heavily use of nondeterministic choice, but length is polynomial in A (and, for revision, in B)
- allows to go beyond classical propositional calculus
  - modification of planning tasks
  - modification of abstract argumentation frameworks

Modification of planning tasks



- What if a planning task has no solution?
  1. modify the set of goal states such that it is reachable from s_0
     - 'oversubscribed goals' [Smith, ICAPS 2004, . . .]
  2. modify s_0 such that the goal states is reachable
     - 'finding good excuses' [Göbelbecker et al., ICAPS 2010]
  3. augment the set of planning operators
  4. . . .

Modification of planning tasks, ctd.



- requires revision by a counterfactual statement:

$$s_0 * \text{“}S_G \text{ is reachable”}$$

$$S_G * \text{“}s_0 \text{ can reach me”}$$

- can be captured in DL-PA [Herzig et al., ECAI 2014]

$$\text{“}S_G \text{ is reachable”} = \left\langle \pi_{\text{PlanOps}} \right\rangle S_G$$

$$\text{“}s_0 \text{ can reach me”} = \left\langle (\pi_{\text{PlanOps}})^{-1} \right\rangle s_0$$

where π_{PlanOps} iterates nondeterministic choice of a planning operator

Modification of an abstract argumentation framework

- theory of an argumentation framework:

$$\mathtt{Th}(A,R) = \bigwedge_{(a,b)\in R} Att_{a,b} \wedge \bigwedge_{(a,b)\notin R} \neg Att_{a,b}$$

- logical characterisation of extensions:

$$\mathtt{Stable} = \bigwedge_{a\in A} \Big(In_a \leftrightarrow \neg \bigvee_{b\in A} (In_b \wedge Att_{a,b}) \Big)$$

  (exists for many other semantics [Baroni&Giacomin])
- the programming view: build an extension = execute a program
  - 'generate-and-test':

$$\mathtt{makeExt} = \mathtt{vary}(\{In_a \ : \ a \in A\}); \mathtt{Stable?}$$

  - more sophisticated algorithms can also be recast
  - ...and proved correct in the logic!
- modify (A, R) such that $\mathtt{Goal}$ is true = update by a counterfactual statement [Doutre et al., KR 2014]
  - $\mathtt{Th}(A,R) * \langle \mathtt{makeExt} \rangle Goal$ (credoulous)
  - $\mathtt{Th}(A,R) * [\mathtt{makeExt}] Goal$ (skeptical)

A more informed version of integrity constraints

- old problem in databases: what if a transaction leads to a violation of some integrity constraint *IC*?
  - example: $(\neg p \wedge q) * p = p \wedge q$ violates $IC = p \to \neg q$
  - requires a *repair*
  - much work in the 80ies, but basically still open
- active integrity constraints: guide the repair

  [Flesca, Greco, Zumpano 2004; Caroprese, Truszczynski, Cruz-Filipe,. . .]

$$r = \langle p \to \neg q, \text{-q} \rangle$$

- can be captured in DL-PA [Feuillade&Herzig, JELIA 2014]
  - program $\pi_r = p \wedge q)?; \text{-q}$
  - several semantics: weakly founded, founded, . . .

Conclusion

- AGM/KM too far from computer science applications
- concrete semantics are most useful
  - Winslett, Satoh, Forbus, Dalal
- revise/update = execute a program
- dynamic logic can express revision by counterfactuals
- we can reason about change in the logic