

Certification of Iterated Belief Changes via Model Checking and its Implementation

Kai Sauerwald¹ Philip Heltweg¹ Christoph Beierle¹

¹Chair of Knowledge-Based Systems - University of Hagen



NMR 2021: November 5th, 2021

Agenda

- ▶ The Certification Problem
- ▶ Encoding as Model-Checking Problem
- ▶ Webtool Alchourron
- ▶ Evaluation and Improvements

The Certification Problem

(Iterated) Belief Change

- ▶ Agents have to adapt their beliefs according to potentially conflicting information
- ▶ Iterated belief change: Modeled by operators over epistemic states¹.
- ▶ Often propositional language \mathcal{L} over finite signature Σ
- ▶ In the following: $\alpha, \beta \in \mathcal{L}$ denote sentences, Ω denotes set of interpretations

¹In contrast, classical belief revision uses belief sets or belief bases as states.

Belief Revision on Epistemic States [1]

- ▶ Epistemic state: abstract entity $\Psi \in \mathcal{E}$, equipped with a deductively closed set of currently held beliefs $\text{Bel}(\Psi)$
- ▶ belief change operator $\circ : \mathcal{E} \times \mathcal{L} \rightarrow \mathcal{E}$
- ▶ we assume syntax-independence for \circ :
 - ▶ if $\alpha \equiv \beta$, then $\Psi \circ \alpha = \Psi \circ \beta$
- ▶ instantiation of \mathcal{E} here: total preorders over Ω that fulfil the faithfulness condition $\text{Mod}(\text{Bel}(\Psi)) = \min(\Omega, \leq)$
- ▶ therefore every $\leq \in \mathcal{E}$ entirely describes an epistemic state

Postulates

- ▶ postulates place restrictions on individual changes or whole operators
- ▶ operators are classified depending on what postulates they fulfil
- ▶ semantic and syntactic postulates
- ▶ for example Darwiche-Pearl postulates for revision [1], here CR1
 - ▶ if $\omega_1, \omega_2 \in \text{Mod}(\alpha)$, then $\omega_1 \leq_{\Psi} \omega_2 \Leftrightarrow \omega_1 \leq_{\Psi \circ \alpha} \omega_2$

Certification Problem

CERTIFICATION-PROBLEM

Given: A belief change operator \circ and a postulate P

Question: Does \circ satisfy the postulate P ?

- ▶ A singular belief change from Ψ to Ψ' by α , i.e.: Does $\Psi \circ \alpha = \Psi'$ hold?
- ▶ A sequence of belief changes $\Psi_1 \circ \alpha_1 = \Psi_2$, and $\Psi_2 \circ \alpha_2 = \Psi_3$, and ...
- ▶ All singular belief changes on a state Ψ , i.e. the set $\{(\Psi_1, \alpha, \Psi_2) \in \circ \mid \Psi = \Psi_1\}$

Encoding as Model-Checking Problem

Approach

- ▶ Define a first-order fragment FO^{TPC} to encode change in epistemic states with new information
- ▶ Build a FO^{TPC} -structure A_C for a concrete belief change $C = (\Psi, \alpha, \Psi')$
- ▶ Load postulate as formula φ and evaluate $A_C \models \varphi$

Language for Postulates

Predicate	Intended meaning	Exemplary appearance
$Mod(w, x)$	w is a model of x	$\omega \in Mod(\Psi), \omega \in Mod(\alpha)$
$LessEQ(w_1, w_2, e)$	$w_1 \leq w_2$ in e	$\omega_1 \leq_{\Psi} \omega_2$
$Int(w)$	w is an interpretation	$\omega \in \Omega$
$ES(e)$	e is an epistemic state	$\Psi \in \mathcal{E}$
$Form(a)$	a is a formula	$\alpha \in \mathcal{L}$
Function	Intended meaning	Exemplary appearance
$op(e_0, a)$	$op(e_0, a)$ is a result of changing e_0 by a	$\Psi \circ \alpha = \Psi'$
$or(a, b)$	propositional disjunction	$Bel(\Psi \circ (\alpha \vee \beta)) = \dots$
$not(a)$	propositional negation	$\neg \alpha \notin Bel(\Psi \circ \alpha)$

$$LogImpl(x, y) := \forall w. Int(w) \rightarrow (Mod(w, x) \rightarrow Mod(w, y))$$

Structure \mathcal{A}_C

Universe	$U^{\mathcal{A}_C} = \Omega \cup \{\Psi_0, \Psi_1\} \cup \mathcal{P}(\Omega)$	
<hr/>		
Predicates		
$Mod^{\mathcal{A}_C}$	$= \{(\omega, x) \mid x \in \mathcal{P}(\Omega) \cup \{\Psi_0, \Psi_1\}, \omega \in \text{Mod}(x)\}$	
$Int^{\mathcal{A}_C}$	$= \Omega$	
$ES^{\mathcal{A}_C}$	$= \{\Psi_0, \Psi_1\}$	
$Form^{\mathcal{A}_C}$	$= \mathcal{P}(\Omega)$	
$LessEQ^{\mathcal{A}_C}$	$= \{(\omega_1, \omega_2, \Psi_i) \mid \omega_1 \leq_{\Psi_i} \omega_2\}$	
<hr/>		
Functions		
$or^{\mathcal{A}_C}$	$= \lambda\alpha_1, \alpha_2. \alpha_1 \cup \alpha_2$	$e_0^{\mathcal{A}_C} = \Psi_0$
$not^{\mathcal{A}_C}$	$= \lambda\alpha_1. \Omega \setminus \alpha_1$	$a^{\mathcal{A}_C} = \text{Mod}(\alpha)$
$op^{\mathcal{A}_C}$	$= (\{(\Psi, \beta, \Psi) \mid \beta \in \mathcal{P}(\Omega), \Psi \in \{\Psi_0, \Psi_1\}\} \setminus \{(\Psi_0, \alpha, \Psi_0)\}) \cup \{(\Psi_0, \alpha, \Psi_1)\}$	

Webtool Alchourron

Implementation

- ▶ Available online at <https://www.fernuni-hagen.de/wbs/alchourron/>
- ▶ Client-Server architecture
- ▶ Backend: Own Java library, Frontend: Browser with web components
- ▶ Loads postulates in TPTP syntax [3] using `scala-tptp-parser` [2]

Postulates in TPTP Syntax

```
fof(
  'CR1',
  postulate,
  ! [W1,W2] : (
    (int(W1) & int(W2) & mod(W1, A) & mod(W2, A))
    =>
    (lesseq(W1, W2, E0) <=> lesseq(W1, W2, op(E0, A)))
  )
).
```



Signature Σ

Create a new propositional syntax here



$\Sigma = \{a,b\}$

CREATE

$\leq \Psi$

Configure the initial state Ψ by defining it's tpo over Ω . Drag the worlds to different layers, the lower layers meaning more plausible.

1				
0	ab	$\bar{a}b$	$a\bar{b}$	$\bar{a}\bar{b}$

α

Configure the input α as a propositional formula over Σ .

Formula

a

$\leq \Psi \circ \alpha$

Configure the next state $\Psi \circ \alpha$ by defining it's tpo over Ω . Drag the worlds to different layers, the lower layers meaning more plausible.

2		
1	$\bar{a}b$	$a\bar{b}$
0	ab	$\bar{a}\bar{b}$

Optional: Enter your own postulate to check here.

Formula

$$C = (\Psi, \alpha, \Psi \circ \alpha)$$

Finally, click "Check Postulates" to verify which postulates are satisfied by the entered belief change.

 CHECK POSTULATES

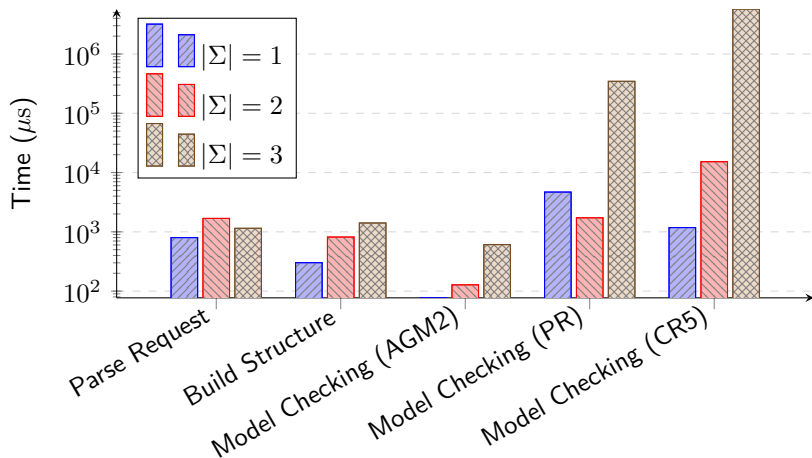
Name	Formula	Satisfied?
Success	FORALL W1. (NOT (Int(W1) AND Mod(W1, op[E0, A])) OR Mod(W1, A))	✓
Vacuity	FORALL W1. (NOT (Int(W1) AND (Mod(W1, E0) AND Mod(W1, A))) OR Mod(W1, op[E0, A]))	✓

Evaluation and Improvements

Performance Questions

- ▶ assumption: size of signature has biggest impact on performance
- ▶ possible bottlenecks: parsing request, building A_C , model-checking postulate
- ▶ method: measure average times for belief change that fulfils all postulates

Measurement Results



- ▶ biggest factor: number of quantifiers in postulate formula

- ▶ better parallelism for postulate evaluation and quantified formula evaluation
- ▶ response time from 12.5s to 3.9s for signature of size three

Pros and Cons

- + Easy to extend with new postulates
- + Completely automated
- + Potentially able to provide counter examples
- Performance

- ▶ Extend approach to more sub-problems (i.e. whole operators)
- ▶ Performance: Improve formula evaluation

Thank you

- ▶ Kai Sauerwald, kai.sauerwald@fernuni-hagen.de
- ▶ Philip Heltweg, pheltweg@gmail.com
- ▶ Try yourself online at
<https://www.fernuni-hagen.de/wbs/alchourron/>

- [1] Adnan Darwiche and Judea Pearl. “On the logic of iterated belief revision”. In: *Artificial Intelligence* 89.1-2 (Jan. 1997), pp. 1–29. DOI: [10.1016/s0004-3702\(96\)00038-0](https://doi.org/10.1016/s0004-3702(96)00038-0). URL: [https://doi.org/10.1016/s0004-3702\(96\)00038-0](https://doi.org/10.1016/s0004-3702(96)00038-0).
- [2] Alexander Steen. *scala-tptp-parser*. Version v1.3. The TPTP syntax specification may be accessed at <http://tptp.org/TPTP/SyntaxBNF.html>. Apr. 2021. DOI: [10.5281/zenodo.4672395](https://doi.org/10.5281/zenodo.4672395). URL: <https://doi.org/10.5281/zenodo.4672395>.
- [3] G. Sutcliffe. “The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0”. In: *Journal of Automated Reasoning* 59.4 (2017), pp. 483–502.